

# University of Chester

Faculty of Science, Business and Enterprise  
Department of Computer Science

## Exploring Mixed Reality Level Design Workflows

---

Thesis submitted in accordance with the requirements of  
the University of Chester for the Degree of  
Doctor of Philosophy  
by

**Lee Beever**

Supervisors: Prof. Nigel W. John and Dr Serban R. Pop

March 2023

### **Statement of Originality**

The material being presented for examination is my own work and has not been submitted for an award of this or another HEI except in minor particulars which are explicitly noted in the body of the thesis. Where research pertaining to the thesis was undertaken collaboratively, the nature and extent of my individual contribution has been made explicit.

Student:

A handwritten signature in black ink, appearing to read 'Lee Beever', written in a cursive style.

Lee Beever

# Acknowledgements

I would first like to thank my supervisor, Nigel John for his unwavering support and expert feedback throughout the development of this thesis. I would also like to thank Serban Pop for his continued support as an initial co-supervisor and then as a colleague and friend. Both of their encouragement, belief and, trust in me has been greatly appreciated.

I would like to thank the participants of my studies for their time and contribution, who have provided invaluable feedback on the software and workflows developed throughout this thesis.

Finally, I would like to thank my friends, colleagues, and family for their support throughout these past six years. I would like to especially thank my wife Lauren and dog Tebow, who have been integral in helping me through the challenging times but have also been there to celebrate the successes.

This thesis would not have been possible without the support of all those above.

# Abstract

The past decade has seen a continual increase in quality and capability of augmented reality (AR) and virtual reality (VR) devices. Due to this greater capability, there have been an influx of entertainment and serious games that have been developed for these systems. Yet, the current workflows for developing VR game levels for entertainment or serious games have remained the same, with developers using a game engine presented on a 2D screen with a traditional mouse and keyboard for input.

This thesis explores the use of AR and VR technologies as part of level design workflows used to develop both entertainment and serious VR game levels. Two existing workflows were identified as areas that could be improved by integrating AR and VR technologies as part of the workflow. Whilst a third new workflow was developed which focused on enabling new experiences for players:

**Workflow 1:** This workflow explored using AR to help create a digital map of an existing space to help improve realism and presence of a VR serious game environment. The initial focus was on improving the workflow for developers of serious game levels.

**Workflow 2:** This workflow focused on improving entertainment VR game level creation through the development of a VR level editor. The focus was on improving the entertainment VR level design process for professional level designers.

**Workflow 3:** This workflow enables new experiences by supporting substitutional reality (SR) level design for players through a mix of both AR and VR technologies. It enables players to develop their own entertainment game levels that support SR using consumer technology.

Each of the three workflows are presented in this thesis along with results from multiple studies. Results from the studies show positive outcomes supporting each of the workflows.



# Contents

Acknowledgements .....	ii
Abstract .....	iii
List of Figures .....	x
List of Tables .....	xvii
Chapter 1 Introduction.....	1
1.1 Motivation .....	1
1.2 Research Question.....	5
1.3 Contributions .....	5
1.4 List of Publications.....	6
1.5 Thesis Structure .....	7
Chapter 2 Background .....	9
2.1 Extended Reality .....	9
2.1.1 Augmented Reality.....	9
2.1.2 Immersion and Presence .....	10
2.1.3 Virtual Reality.....	11
2.1.4 Substitutional Reality.....	12
2.2 Augmented Reality State of the Art.....	13
2.2.1 Head-mounted AR .....	14
2.2.2 Smartphone AR .....	17
2.3 Virtual Reality State of the Art.....	22
2.3.1 Tethered VR Headsets.....	27
2.3.2 Standalone VR Headsets .....	29
2.3.3 Mixed Reality VR Headsets.....	31
2.4 Level Design.....	33
2.4.1 Level Design Areas of Focus.....	34
2.4.2 Level Design Process and Workflow .....	35
2.4.3 Serious Games .....	37
2.4.4 Serious Games Level Design .....	38

2.4.5 Virtual Reality Games .....	39
2.4.6 Virtual Reality Level Design .....	41
3.0 Literature Review .....	44
3.1 Augmented Reality Level Design Similar Work.....	44
3.1.1 Augmented Reality Level Editing.....	44
3.1.2 Location Capture and Indoor Modelling.....	46
3.2 Virtual Reality Level Design Similar Work.....	48
3.2.1 World Building in VR .....	48
3.2.2 3D Modelling in VR .....	58
3.3 Substitutional Reality Level Creation Similar Work .....	61
3.3.1 Passive Haptics .....	61
3.3.2 Substitutional Reality Experiences and Benefits.....	62
3.3.3 Non-Substitutional Reality Creation.....	64
3.3.4 Automated or Procedural Substitutional Reality Creation .....	64
3.3.5 Manual/Human-in-the-Loop Substitutional Reality Creation .....	66
3.4 Summary.....	67
3.4.1 Mapping a Real-World Location – Workflow 1 .....	67
3.4.2 Virtual Reality Level Design – Workflow 2 .....	68
3.4.3 Substitutional Reality Level Design – Workflow 3.....	68
Chapter 4 LevelEd AR – Workflow 1 .....	69
4.1 Introduction and Motivation.....	69
4.2 Workflow 1.....	70
4.3 Hypothesis.....	72
4.4 System Overview and Implementation.....	72
4.5 Study 1 - Experiment Methodology .....	75
4.5.1 Experiment Tasks .....	75
4.5.2 Measuring Instruments.....	76
4.5.3 Participants .....	77
4.6 Results .....	77

4.6.1 Measurements .....	77
4.6.2 Time.....	80
4.6.3 System Usability Scale.....	81
4.7 Virtual Environment Workflow Preliminary Study .....	82
4.8 Discussion .....	84
4.9 Conclusion.....	86
Chapter 5 LevelEd VR – Workflow 2 .....	88
5.1 Introduction and Motivation.....	88
5.2 Workflow 2.....	89
5.3 Hypotheses .....	91
5.4 System Overview and Implementation.....	91
5.4.1 Locomotion .....	92
5.4.2 Menu and Interaction .....	94
5.4.3 Level Editing Tools.....	95
5.4.4 Sample Game .....	99
5.4.5 Play Testing .....	100
5.4.6 Saving and Loading .....	100
5.5 Preliminary Study .....	100
5.6 Study 2 - Experiment Methodology .....	102
5.6.1 Experiment Structure .....	102
5.6.2 Participants .....	103
5.7 Results .....	104
5.7.1 System Usability Scale (SUS).....	104
5.7.2 Comparisons of Systems and Workflows .....	105
5.7.3 LevelEd VR Feedback .....	106
5.7.4 Simulator Sickness Questionnaire Results.....	108
5.7.5 Video Coding Analysis .....	110
5.8 Discussion.....	110
5.8.1 Acceptance and Preference .....	110

5.8.2 Minimising Motion Sickness .....	113
5.8.3 Spatial Scripting .....	114
5.8.4 Interaction Issues .....	114
5.9 Conclusion.....	115
Chapter 6 LevelEd SR – Workflow 3 .....	116
6.1 Introduction and Motivation.....	116
6.2 Workflow 3.....	117
6.3 Hypotheses .....	120
6.4 System Overview and Implementation.....	120
6.4.1 Mapping the space in AR .....	121
6.4.2 Registration of the Real and Virtual World.....	122
6.4.3 Substitutional Reality Level Editor .....	123
6.4.4 Locomotion .....	124
6.4.5 Menu and Interaction .....	124
6.4.6 Level Editing Tools.....	125
6.4.7 Sample Game .....	127
6.5 Study 3 - Experiment Methodology .....	129
6.5.1 Study Structure .....	129
6.5.2 Hardware .....	131
6.5.3 Participants .....	131
6.5.4 COVID-19 Safety Procedures .....	132
6.6 Results .....	132
6.6.1 Game Experience Questionnaire (GEQ) .....	132
6.6.2 Spatial Presence Experience Scale (SPES).....	133
6.6.3 Comparison Questionnaires .....	134
6.6.4 System Usability Scale (SUS).....	134
6.6.5 Completion Time.....	136
6.6.6 Simulator Sickness Questionnaire.....	136
6.6.7 LevelEd SR System and Workflow Feedback .....	137

6.7 Discussion .....	139
6.7.1 Mapping Spaces .....	139
6.7.2 Developing Substitutional Reality Levels.....	140
6.7.3 Game Experience and Presence.....	142
6.7.4 Additional Questions .....	143
6.8 Conclusion.....	144
Chapter 7 Conclusions and Future Work.....	145
7.1 Research Question.....	145
7.2 LevelEd AR and Workflow 1 .....	145
7.2.1 Hypothesis .....	145
7.2.2 Limitations.....	146
7.2.3 Actions Identified and Lesson Learnt.....	146
7.3 LevelEd VR and Workflow 2 .....	147
7.3.1 Hypothesis .....	147
7.3.2 Limitations.....	148
7.3.3 Lessons Learnt .....	148
7.3.4 Future Work .....	149
7.4 LevelEd SR and Workflow 3 .....	151
7.4.1 Hypotheses .....	151
7.4.2 Limitations.....	152
7.4.3 Lessons Learnt .....	153
7.4.4 Future Work .....	153
7.5 Final Words .....	155
References .....	156
Appendices .....	179
Appendix A.....	179
Appendix B.....	179
Appendix C.....	179
Appendix D.....	180

Appendix E.....	181
Appendix F.....	182
Appendix G.....	183
Appendix H.....	184
Appendix I.....	185
Appendix J.....	186
Appendix K.....	187
Appendix L.....	192
Appendix M.....	197
Appendix N.....	202
Appendix O.....	205
Appendix P.....	206
Appendix Q.....	207
Appendix R.....	208

# List of Figures

**Figure 1** - (a) E.T. The Extra-Terrestrial released in 1982 for the Atari 2600 (screenshot from Servo, 2003). (b) Pitfall! released in 1982 for the Atari 2600 (screenshot from SirOrlando, 2005). ..... 1

**Figure 2** - (a) Halo Infinite forge editor released in a 2022 update (screenshot from Halo, 2022). (b) Fortnite creative mode released in an update in 2018 (screenshot from Koczwara et al., 2018)..... 3

**Figure 3** - The reality-virtuality continuum outlines the level of inclusion of either the real environment or a fully virtual environment on each category. Adapted from Milgram and Kishino (1994) and Skarbez et al. (2021). ..... 9

**Figure 4** - (a) Microsoft HoloLens headset (Microsoft, 2021). (b) Magic Leap One headset (Greenwald, 2018). ..... 14

**Figure 5** – Trimble Connect allows users to collaboratively view and interact with digital models whilst on site (Trimble, 2022). ..... 16

**Figure 6** - Estimated field of view comparison between the four main devices compared to most VR headsets and human vision. Image constructed from both (Heaney, 2019) and (Ganesan, 2022). ..... 17

**Figure 7** - Example of ARKit detecting the pose (orientation and position) of the smartphone device in the location using sensor data..... 18

**Figure 8** - Computer vision is used to track changes to points of interest (blue, red and orange dots) each frame. Changes to the device pose can be triangulated from these points and their changes over time. .... 18

**Figure 9** - Visual Inertial Odometry utilises computer vision data over time along with up-to-date position and orientation data from device sensors (Apple Inc., 2018). ..... 19

**Figure 10** – (a) Pokemon GO uses ARKit to place digital Pokemon characters into the real-world (Fulton, 2017). (b) IKEA Place lets users view IKEA furniture in their own home (Keller, 2017). (c) Magicplan allows users to create technical floor plans through and AR camera view (Sensopia, 2022). (d) Users can apply AR filters in Instagram that track their face movements (Evans, 2020). ..... 21

**Figure 11** - A high-resolution CAVE system as presented by Visbox (2023). (a) Shows the how the system is configured. (b) Shows the CAVE system in use. .... 23

**Figure 12** - Market share of different VR platforms as of August 2021 (Lang, 2021b)..... 26

**Figure 13** - (a) Oculus Quest CV1 (released 2016) with external tracking. (b) HP Reverb G2 (released 2021) with inside out tracking. (c) HTC Vive (released 2016) with external tracking. (c) PlayStation VR (released 2016) with external tracking via camera..... 27

**Figure 14** - (a) HTC Vive that uses external Lighthouse sensors to create an IR laser beams that sweep on two axes (Allan, 2016). The Vive devices detect these beams and calculate the time it took to receive it. (b) Inside out tracking uses multiple cameras to detect points of interest and calculate their movement over time (Hamilton, 2019). ..... 29

**Figure 15** - (a) Oculus Go (released 2018) with only 3-DoF tracking. (b) Meta (formerly Oculus) Quest 2 (released 2020) with 6-DoF tracking and grayscale passthrough. (c) HTC Vive Focus 3 (released 2021) with 6-DoF tracking and grayscale passthrough. (d) HTC Vive Flow (released 2021) with 6-DoF tracking and is controlled via a smartphone or remote and is meant to be linked to a smartphone..... 30

**Figure 16** – (a) Varjo XR-3 (released 2021) is a tethered MR device with marker-based tracking and 6-DoF via a beta. (b) Meta Quest Pro (released 2022) with MR support via 6-DoF tracking (5 cameras) and colour passthrough. (c) Lynx R-1 (released 2022) focuses primarily on AR but can support VR with 6-DoF tracking via 2 cameras and colour passthrough. (d) HTC Vive XR Elite (released 2023) with MR support via 6-DoF tracking with 4 cameras and colour passthrough. .... 32

**Figure 17** – (a) Multiplayer level blockout for Gears of War 4. (b) The same finished game level (Gears of War, 2017). ..... 36

**Figure 18** – (a) Early blockout prototype of a single player level for Enslaved – Odyssey to the West (b) Final finished version of the same level (Field, 2017)..... 37

**Figure 19** – (a) Blockout prototype of a multiplayer level for Overwatch. (b) Final finished version of the same level (Adams, 2017). ..... 37

**Figure 20** – i.Ge system overview showing how the game is displayed, edited and played on a real-world wall (Oswald et al., 2015). ..... 44

**Figure 21** - Data is displayed digitally along with lines demonstrating how data is transferred. .... 45

**Figure 22** - Images demonstrate virtual items added to a real-world location, as well as how interactions can be programmed (Ng et al., 2018)..... 45

**Figure 23** - Selecting an object to spawn through the UI of Genesys (De Leon, Tavas, Aranzanso & Atienza, 2016)..... 49

**Figure 24** - (a) Bubbles display the available vocal commands for the current context. (b) In the larger scale, the user can have a global view of the environment. (c) In the smaller scale, the user can act on a microscopic level (Barot et al., 2013)..... 50

**Figure 25** - (a) The hardware setup, (b) the floating menu and (c) terrain brush of the HVE world builder (Wang & Lindeman, 2015). ..... 52

**Figure 26** - (a) The guide has a 3/4 top view of the scene and uses the zSpace interactive system. (b) The customer is immersed with a first-person view in the shared virtual world with an HMD and interacts with a Razer Hydra (Le Chenechal et al., 2015)..... 53



<b>Figure 27</b> - A scene with assets from multiple sources (Moreira, 2018).....	54
<b>Figure 28</b> - VR Safari Park. (a) World Tree Connector and Blocks, (b) Tree shape represents the world's structure (Ichikawa et al., 2018).....	55
<b>Figure 29</b> - XRDirector allows users to take on a director, actor or camera role, each with their own method of input/output to collaborative create a prototype animation (Nebeling et al., 2020).....	56
<b>Figure 30</b> - (a) A scene created with VR Designer utilising pre-made assets. (b) A 2D modal window used to apply modifiers displayed as a floating panel in VR. ....	57
<b>Figure 31</b> - A selection of castles created using MakeVR (Jerald et al., 2013).....	58
<b>Figure 32</b> - (a) User extracting curves from the sketch to build a 3D model. (b) The final 3D model rendered in Blender. (c) and (d) participant created boat models (Jackson & Keefe, 2016) .....	58
<b>Figure 33</b> - Mid-air Boolean operations carried out either with (a) gestures detected by a Myo armband or (b) a menu system (Mendes et al., 2017) .....	59
<b>Figure 34</b> - EasyVRModeling demonstrates the use of boolean operations and SDF rendering to create 3D models in VR (Fu et al., 2022).....	60
<b>Figure 35</b> - (a) The pit environment that participants navigate during the test. (b) Passive haptics used during the study. (c) No passive haptics used during the study (Meehan et al., 2002). ....	61
<b>Figure 36</b> - (a) the real-world location. (b) A digital recreation of the real-world. (c) A Sci-Fi version of the space with maximum mismatch. (d) Displays both environments (top) and mixed environments (bottom) (Eckstein et al., 2018).....	63
<b>Figure 37</b> - (a) The real-world location, including some objects that use Vive trackers to track their position. (b) SR version built for the horror game and mapped to the original space (Steinhaeuser and Lugin, 2020).....	63
<b>Figure 38</b> - AffordIt application for adding interaction to models in VR. (a) An object to add interaction to. (b) The user cutting into the mesh to add interaction. (c) The user defining the extents of the rotation interaction. (d) The interaction in action (Masnadi et al., 2020).....	64
<b>Figure 39</b> - DreamWalker system in use. (a) The real-world unseen location traversed by use with VR headset and laptop. (b) The VR environment generated to ensure the user does not collide with obstacles or people. (c) A demonstration of what DreamWalker detects (Yang et al., 2019). ....	65
<b>Figure 40</b> - Reality Skins process. (a) A room is scanned using an RGB-D camera (like Microsoft Kinect). (b) The floor plan is generated, and location of objects located. (c) A medieval tavern theme is generated. (d) A sci-fi hospital theme is generated (Shapira & Freedman, 2016). ....	65

<b>Figure 41</b> - Various demonstrations of RealityCheck. (a) Blending between Waltz of the Wizard and real world. (b) Collision estimates in the Superhot VR game. (c) Flashlight concept that allows users to choose where to reveal real-world. (d). A projection of what the user is viewing for co-located spectatorship (Hartmann et al., 2019).....	66
<b>Figure 42</b> - ARchitect supports SR level creation in AR. (a) The real-world location and obstacles. (b) A user editing the level using smartphone AR. (c) The user adding a tree stump in place of a chair. (d) The final game level (Lin et al, 2020).....	66
<b>Figure 43</b> - Walkthrough of Workflow 1 that enables a user to capture a space using AR to later us in Unity to develop to scale serious game levels. Further details on using the application can be found in section 4.4 and <b>Figure 44</b> .....	71
<b>Figure 44</b> – LevelEd AR workflow from AR to Unity game engine. (a) User is mapping the dividing wall; the shape, scale and location will be captured. (b) User has completed the mapping of the shape, size and location of the table. (c) Model has been generated in Unity engine from the data uploaded by the user. Grid squares are 1m x 1m, with 10cm subsections.....	74
<b>Figure 45</b> – (a) Participant completing Task 4 with measuring tape and paper. (b) Participant completing Task 4 with the Room Capture software with the Structure Sensor. (c) Participant completing Task 2 with the LevelEd AR application. ....	75
<b>Figure 46</b> - Structure and flow of Study 1 .....	76
<b>Figure 47</b> – (a) Task 1 – Area measurement of single wall, (b) Task 2 – Volume measurement of small box, (c) Task 3 – Volume measurement of larger box against a wall, (d) Task 4.1 – Area measurement of 4 continuous walls, (e) Task 4.2 – Volume measurement of small box, (f) Task 4.3 – Gap measurement between Task 4.2 box and starting wall of Task 4.1. ....	78
<b>Figure 48</b> – (a) Task 1 – Time to measure single wall, (b) Task 2 – Time to measure small box, (c) Task 3 – Time to measure larger box against a wall, (d) Task 4 – Time to measure 4 continuous walls, small box, and gap between box and starting wall.....	81
<b>Figure 49</b> – Results from the System Usability Scale questionnaire (Brooke, 1996).....	82
<b>Figure 50</b> – (a) Photo of the real-world location. (b) Model of the location captured by LevelEd AR and downloaded to Unity. (c) Realistic recreation of the location as a serious games virtual environment. (d) Egyptian themed virtual environment created for potential SR. (e) Realistic scene integrated into a powered-wheelchair VR training simulator. (f) View from VR headset when in the realistic scene in the powered-wheelchair VR training simulator. ....	83
<b>Figure 51</b> - Outline of Workflow 2 which supports the creation of VR game levels whilst in VR.....	90

<b>Figure 52</b> - (a) User holds the teleport button and aims where they want to go. (b) User releases teleport button and they are then moved to their destination. ....	93
<b>Figure 53</b> - (a) User reaches out in the direction they want to move. (b) User grips the bottom controller triggers and pulls the world to them. A vignette is used to ease simulator sickness. (c) The user lets go of the triggers when they get to their destination. ....	93
<b>Figure 54</b> - The variable pointer system in use. (a) Short fixed-point set to 10cm. Good for working up close. (b) Medium fixed-point set to 1m. Good for working at arm's length. (c) Infinite beam length will continue until it hits an object. Great for working at large distances or for selecting something at a distance and bringing it closer. ....	94
<b>Figure 55</b> - (a) The options, tools and edit menus as seen in the Unity editor. (b) The Mesh menu with the colour options pop out visible as viewed in VR. (3) The options menu as viewed in VR. The Axis option is greyed out as it is not relevant to the currently selected Edit Faces tool. ....	95
<b>Figure 56</b> - (a) Cube creation tool. (b) Sphere creation tool. (c) Cylinder creation tool. (d) Drawing out initial custom shape with polygon creation tool. (e). Finished custom polygon shape. (f) Another custom polygon shape. (g) A warehouse style level blockout created using the tools. (h) A fantasy style level blockout created using the tools. ....	96
<b>Figure 57</b> - Textures tile correctly across the shapes as they are created. This ensures each sub-square remains 10x10cm. ....	97
<b>Figure 58</b> - (a) User has selected Vertex Edit mode and vertices are made visible. (b) The user has selected some vertices and moved them to make a more unique shape. ....	97
<b>Figure 59</b> - (a) Spatial scripting number nodes attached to the ScoreManager to control time limit. (b) User edits the number node to change the seconds from 13 to 20. (c) The change in seconds to complete the level is visible on the scoreboard. (d) User can quickly test the level to see if it is enough or too much time. ....	98
<b>Figure 60</b> - (a) Normal lock on. Current face can only move on the current normal axis (up and down). (b) Normal lock off. The face can move on all axes. ....	99
<b>Figure 61</b> - (a) Overview of the sample level created using the tools for the study. (b) View from inside the sample level used in the study. ....	100
<b>Figure 62</b> – (a) Previous menu prototype used in preliminary study. (b) New menu used for the main study. ....	101
<b>Figure 63</b> - Structure and flow of Study 2 .....	103
<b>Figure 64</b> – Results from the System Usability Scale questionnaire. Negative outlier marked with a dot and median with horizontal line (Brooke, 1996). ....	105
<b>Figure 65</b> – Comparative questionnaire results based on questions in <b>Table 7</b> . ....	106

<b>Figure 66</b> – Responses to the general feedback questionnaire ( <b>Table 8</b> ) for LevelEd VR. A 7-point Likert scale response was used for each question. Outliers marked with a plus and median marked with horizontal line. ....	107
<b>Figure 67</b> - Sample levels created by participants using LevelEd VR during the study. ....	112
<b>Figure 68</b> - Sample game levels created by the same participants from <b>Figure 67</b> using Unity Desktop during the study. ....	112
<b>Figure 69</b> - Workflow 3 which brings together work on LevelEd AR and LevelEd SR to create a SR level editor and workflow. ....	118
<b>Figure 70</b> - (a) Virtual map captured using LevelEd AR. Controller locations are captured during the session. (b) User puts on their HMD in the real-world location and place the controllers in the same position. (c) User presses the register button, and the virtual world is aligned with the real-world using controller locations as common data points. ....	123
<b>Figure 71</b> - (a) Player has turned on the objects loaded from the <b>Outline</b> data. (b) A player moving a prefab (tree) in a participant created level. (c) A player editing a number scripting node in a participant created level. (d) A participant testing their SR level (the tablet displays what the player is seeing). ....	124
<b>Figure 72</b> - (a) Creation menu. (b) Gameplay menu. (c) Models' menu. ....	125
<b>Figure 73</b> - A demonstration of using the virtual mapping data to create an SR game level. (a) User can use the mesh creation tools to create a mesh that matches the virtual map data. (b) User can use a pre-existing model to replace the virtual map data. (c). User can use the Outline virtual map data and convert it into a game object.....	126
<b>Figure 74</b> - Preview and shortcuts can be used to quickly place same assets with variations. (a) Preview of any object that can be placed in the scene. This helps with placement and live editing. (b) Shortcuts that can be used to scale whilst placing an object. (c) Shortcuts that can be used to rotate whilst placing an object. ....	127
<b>Figure 75</b> - The sample level being played in SR. Left is what the player is seeing whilst interacting on the right. (a) Player starts the game using the start node. (b) Player uses the furniture/Egyptian block to help aim. (c) Player uses the wardrobe/Egyptian tablet for cover from the laser targets. (d) Player runs towards the desk/Egyptian tomb. (d) Player sits on the desk/Egyptian tomb to shoot hidden targets above them. ....	128
<b>Figure 76</b> - Order of tasks carried out by participants during the study. ....	130
<b>Figure 77</b> - The study space consisting of the five pieces of furniture set up to mimic a traditional office.....	130
<b>Figure 78</b> - Comparative questionnaire results. Questions in <b>Table 17</b> . ....	134
<b>Figure 79</b> - Results from the System Usability Scale questionnaire (Brooke, 1996).....	135
<b>Figure 80</b> - Completion time results for both Outline and Meshing.....	136

**Figure 81** - Results from general feedback questions (**Table 20**) regarding the LevelEd SR workflow and individual components. .... 138

**Figure 82** - Four game levels created by four different participants. .... 141

# List of Tables

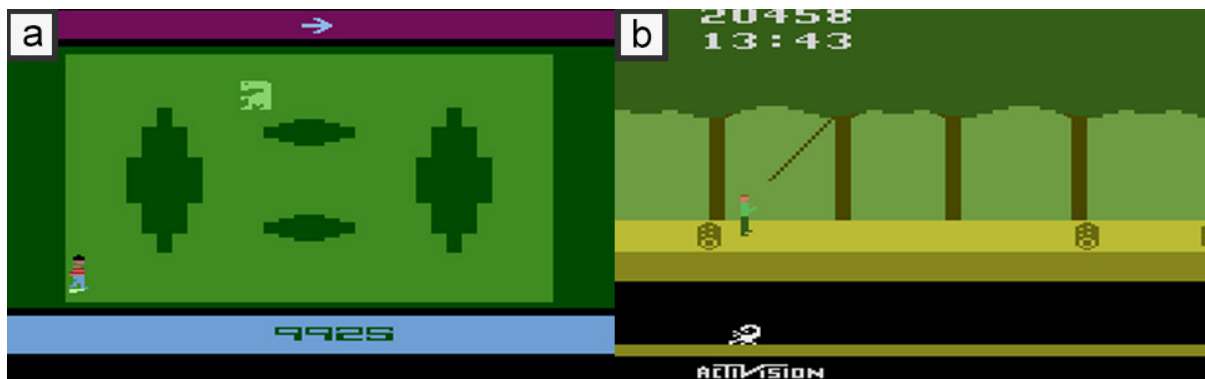
<b>Table 1</b> - Field of Views (FoV) for the four main AR headsets.....	16
<b>Table 2</b> - Common tethered VR headsets from 2016-2023. Sourced from VRCompare (2023). Extended version can be found in <b>Appendix D</b> .....	28
<b>Table 3</b> - Common standalone VR headsets from 2018-2021. Sourced from VRCompare (2023). An extended version can be found in <b>Appendix E</b> .....	31
<b>Table 4</b> - Common standalone XR/MR headsets from 2021 to present. Sourced from VRCompare (2023).....	32
<b>Table 5</b> – Time to complete all four tasks with LevelEd AR across each age group.....	85
<b>Table 6</b> - LevelEd VR Features for Study.....	99
<b>Table 7</b> - Comparative Questionnaire Questions .....	106
<b>Table 8</b> - General Questionnaire Questions for LevelEd VR .....	107
<b>Table 9</b> - Reference SSQ-Evaluation Findings.....	109
<b>Table 10</b> - LevelEd VR Pre SSQ-Evaluation Findings.....	109
<b>Table 11</b> - LevelEd VR Post SSQ-Evaluation Findings .....	109
<b>Table 12</b> - Unity Desktop Pre SSQ-Evaluation Findings .....	109
<b>Table 13</b> - Unity Desktop Post SSQ-Evaluation Findings.....	109
<b>Table 14</b> - LevelEd VR Features Available to Participants. Major new features in bold.....	126
<b>Table 15</b> - Results for the Game Experience Questionnaire (IJsselsteijn et al., 2013). Better scores highlighted in bold.....	133
<b>Table 16</b> - Results for the Spatial Presence Experience Scale (Hartmann et al., 2016) for Self-Location (SL), Possible-Actions (PA) and Total Presence (Total). Higher total scores highlighted in bold. ....	133
<b>Table 17</b> - Comparative Questionnaire Questions. ....	134
<b>Table 18</b> - LevelEd SR Pre SSQ-Evaluation Findings.....	137
<b>Table 19</b> - LevelEd SR Post SSQ-Evaluation Findings .....	137
<b>Table 20</b> - LevelEd SR Workflow General Feedback Questions .....	138
<b>Table 21</b> – Extended table displaying common tethered VR headsets from 2016-2023. Sourced from VRCompare (2023).....	180
<b>Table 22</b> - Extended table displaying common standalone VR headsets from 2018-2022. Sourced from VRCompare (2023).....	181

# Chapter 1 Introduction

This thesis explores how augmented reality (AR), and virtual reality (VR) technologies can be used to improve existing VR level design workflows or enable new workflows and experiences. This chapter is broken down into the motivation for this thesis, the research question that guided the research, contributions of the thesis, a list of publications and a breakdown of the remaining thesis.

## 1.1 Motivation

In modern game development it is typical to find specialist level design roles, however, this has not always been the case. In the early years of the games industry, it was common to find a large proportion of games made by solo developers or very small teams with limited specialisations. In the late 1970s, Atari saw success with their arcade machines and eventually with the Atari 2600 (Amos, 2019, p. 18). Many of the popular games for these systems were developed by small teams, such as the infamous E.T. The Extra-Terrestrial which was designed and programmed solo by Warshaw (1982) and Pitfall! developed solely by Crane (1982) (see **Figure 1**).



*Figure 1 - (a) E.T. The Extra-Terrestrial released in 1982 for the Atari 2600 (screenshot from Servo, 2003). (b) Pitfall! released in 1982 for the Atari 2600 (screenshot from SirOrlando, 2005).*

It was possible for these games to be developed in this way due to the simple graphics and gameplay of that time but also since game design as a discipline was in its infancy. As game hardware evolved and became more powerful, so did the fidelity and complexity of the games. This increase in fidelity required larger teams to build games on new generations of hardware, which resulted in the introduction of more specialist roles. One of the early specialist roles saw the separation of the game design discipline into a game designer and a level designer. This is visible in the credits for Super Mario Bros 3 (Nintendo, 1990) which notes two game designers and four specialist level designers compared to older games where this distinction is not present (Moby Games, 2022).

The separation of game designer and level designer has continued into modern game development and now most large game studios employ level designers to assist in creating a video game. A level designers' role is to take the mechanics developed by the game designers and to construct game spaces that best facilitate these mechanics (Totten, 2014, p. xxiv; Kremers, 2009, p. 18). The level designers spend a lot of their time constructing the layout of these spaces and deciding how best to utilise the mechanics available within these layouts. More importantly, the workflow for level design has not changed much in the last two decades with game levels being developed in a game engine/3D software on a 2D desktop screen with a mouse and keyboard. For example, popular games like Call of Duty: WWII (Sledgehammer, 2017) still make use of this workflow (Pennell, 2019). This workflow has also continued into the development of VR game levels.

Whilst most game levels are created by professional level designers, there are some games that also support the creation of user generated content, through modifications (*mods*) to the game (Totten, 2014, p. xxi; Duan et al., 2022). Early mods were primarily supported on the PC platform both officially (through editors/SDKs) and unofficially (through modifying source files). Most games consoles do not offer access to the game files to modify them, which limits unofficial modifications and very few early games offered official support. Doom (id Software, 1993) and Unreal Tournament (Epic Games, 1999) were early examples of PC games that supported these modifications, in the form of custom game levels.

Despite the popularity of games growing (Wallach, 2020), interest in developing user generated game levels declined due to the rise in popularity of games consoles over PC games (De Jong, 2008). Games consoles are typically cheaper to buy and require less maintenance to run. Games exclusive to a specific console also helped drive players towards consoles and away from PCs, where they could not be played. As a PC was generally required to create modifications, the switch of players from PC to consoles resulted in fewer opportunities to create modifications and fewer players to share them with. However, there has been an increase in popularity of UGC over the past decade with the release of console games that support UGC through official tools and editors. For example, games such as Little Big Planet 1 (Media Molecule, 2008), Minecraft (Mojang Studios, 2011), Roblox (Roblox Corporation, 2006-2015), Fortnite (Epic Games, 2017) and Halo Infinite (343 Industries, 2021) offer tools and UGC on consoles and some also support mobile devices (see **Figure 2**).





**Figure 2** - (a) Halo Infinite forge editor released in a 2022 update (screenshot from Halo, 2022). (b) Fortnite creative mode released in an update in 2018 (screenshot from Koczwara et al., 2018).

These games enable players to create custom content for the game, including game levels, which can then be shared or played with others. Poretski et al. (2019) suggests that value is added to a game when modifications are allowed/encouraged, with players engaging for longer with games with UGC included. This added value could be due to UGC adding more playable content and allowing for more creative risks in its design (Poretski et al., 2019). This added value may explain why game sales are increased when more UGC content is created for a game (Poretski & Arazy, 2017). Whilst some games support unofficial modifications, research shows an increase in modding activity when mods are officially supported by a company, such as with a level editor/SDK (Poretski & Arazy, 2017). An empirical study on VR game trends suggests one of the largest complaints from players was about lack of content in games they played, which could be solved through UGC (Epp et al., 2021). Whilst there are some examples of user generated content in VR games, such as with VRChat (VRChat Inc., 2014), Rec Room (Rec Room Inc., 2016) and Racket: Nx (One Hamsa, 2018), examples over the past few years are limited compared to traditional game platforms. There is potential for this to improve with the creation of intuitive level design tools and requires further research.

Whilst video games in their current form have only really existed since the development of Tennis for Two (Higinbotham, 1958) in the late 1950s, AR and VR enabling concepts have existed for much longer. Their roots can be traced back to the 1830s with stereoscopic images and the stereoscope, then with film-based developments such as the Sensorama in 1950s and computer-based imagery with the Sword of Damocles in 1968 (Jerald, 2016, p. 22). Unlike games though, their popularity failed to move beyond niche/novelty products until more recently, which was often due to costs and technical limitations.

There had been several early attempts at blending video games with VR. For example, Virtuality's range of arcade machines in the early 1990s (Jerald, 2016, p. 61) and with Nintendo's 1995 Virtual Boy console (Boyer, 2009). Whilst Virtuality saw some initial success, the company eventually collapsed in 1997 (Virtual Reality Society, n.d) and the Virtual Boy

failed to sell in significant numbers (Edwards, 2015), this continued the trend of VR remaining as a niche product with consumers.

However, this changed in 2012, with the launch of a Kickstarter campaign for the Oculus DK1. The campaign was hugely successful, raising almost \$2.5 million and arguably kickstarting the current rise in popularity of VR (Mitchell, 2014). The success of the Oculus VR campaign resulted in the purchase of Oculus by Facebook (now Meta) for \$2 billion and the development of several competing devices, such as Sony's PSVR and HTC's Vive (built in partnership with Valve Corporation). The popularity of VR continued to rise but was limited due to the power requirements of PCs needed to run these devices. In 2019, Facebook introduced the Oculus Quest as a standalone device (utilising mobile hardware rather than being connected to a PC) at an attractive price point of \$399. For comparison, the Oculus Rift CV1 launched at \$599 in 2016, but that price did not include the cost of a gaming PC required to use it (see **Table 2**). This lowering of the barrier to entry has resulted in a further increase in popularity of VR with consumers as sales figures have continued to rise, particularly with the release of the Oculus Quest 2 (Lang, 2022a; Hearney, 2022).

AR has also seen a recent increase in popularity with the launch of smartphone AR in 2017, through ARKit (Apple Inc., 2022a) and ARCore (Google Developers, 2022). These technologies allow Apple's iPhone/iPad (via ARKit) and Android smartphone/tablet devices (via ARCore) to support AR without the need of a dedicated head-mounted device. Traditional AR experiences required a dedicated head-mounted device, such as Microsoft HoloLens which launched in 2016 at a cost of \$3000, which created an affordability barrier and required new skills to use. An area where smartphone AR has found recent success is with AR filters on smartphone applications. Photo and video applications, such as Instagram (Meta, 2010) and Snapchat (Snap Inc., 2011) use AR to display masks and other digital content tracked to a user's face through a smartphone camera (Evans, 2020). This removal of the affordability barrier of needing a dedicated device has seen AR reach a wider audience over the past 5 years (Révész, 2021).

The popularity of AR and VR technologies is increasing with consumers, and thus the technologies are becoming more relevant. Level design workflows have also remained stagnant for many years despite VR and AR platforms offering different interaction and visualisation methods compared to traditional platforms. Therefore, there is a potential opportunity to explore the role AR and VR can play in VR level design workflows for developers and players (through UGC) alike. AR and VR technologies require more research to see where they can be applied to assist professional level designers, to enable user generated content

for VR games and enable more novel concepts, such as substitutional reality (SR) game levels (Suzuki et al, 2012; Simeone, 2015).

## 1.2 Research Question

This thesis seeks to explore possible uses for AR and VR technologies for level design workflows. At the outset, an overarching research question was devised to give a general direction to the thesis:

*“Can augmented reality and virtual reality technologies be used to improve current level design workflows or establish new workflows that enable new experiences?”*

After researching the different AR and VR technologies available, understanding the existing level design workflows in detail and searching the current literature around this area (see Chapter 2 and 3), three potential workflows were identified that could make use of AR and VR technologies. These three workflows either look to enhance or improve an existing workflow or to allow for new and novel workflows to be used for designing game levels.

Seven hypotheses were developed to help answer the overall research question by testing if there were potential improvements to existing workflows, what those improvements might be and what new workflows could be established. These hypotheses are outlined at the beginning of each relevant workflow chapter.

## 1.3 Contributions

The three workflows along with their corresponding software implementations and studies form the main contributions of the thesis. The contributions are briefly outlined below and are covered in detail in their corresponding chapters.

**Contribution 1:** Workflow 1 which utilises smart phone AR to assist with capturing a real-world location with the intent to use this data to help create serious game levels that are true and accurate to scale. This workflow was enabled through the creation of a bespoke AR smartphone application called LevelEd AR.

**Contribution 2:** Workflow 2 which was designed to explore if and how developing VR game levels, whilst working in VR, can improve existing VR game level design workflows. This workflow enables users to design VR game levels using a VR level editor. This workflow was enabled through the creation of a custom VR level editor called LevelEd VR.

**Contribution 3:** Workflow 3 which was designed to enable new player experiences through a novel workflow that allows players to create their own SR game levels using consumer hardware. This workflow was enabled through the combination of software developed for contribution 1 and 2.

Several studies were completed and are presented in this thesis to provide evidence of the effectiveness of each of the three workflows and contributions.

## 1.4 List of Publications

The following peer reviewed conference papers were published and presented as a result of the research presented in this thesis:

**Beever, L., Pop, S., & John, N. (2019).** Evaluating LevelEd AR: An Indoor Modelling Application for Serious Games Level Design. *In 2019 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*. <https://doi.org/10.1109/vs-games.2019.8864534>

Contribution of first named author:

- Designed and programmed the LevelEd AR software solution in full.
- Planned and carried out the research study.
- Primary author for the paper, with assistance from Dr Serban Pop on the statistics and results chapter and general feedback from Professor Nigel John.
- Presented paper at Virtual Worlds and Games for Serious Applications in Vienna, Austria.

**Beever, L., Pop, S., & John, N. (2019).** Assisting Serious Games Level Design with an Augmented Reality Application and Workflow. *In 2019 Computer Graphics and Visual Computing (CGVC)*. <https://doi.org/10.2312/cgvc.20191253>

Contribution of first named author:

- Adapted previous paper from a 4-page paper to a full 8-page paper with additional material. Assistance from Dr Serban Pop on the statistics and results chapter and general feedback from Professor Nigel John.
- Carried out an additional preliminary study to compliment the previous study.
- Presented paper at Computer Graphics and Visual Computing in Bangor, UK.

**Beever, L., Pop, S., & John, N. W. (2020).** LevelEd VR: A virtual reality level editor and workflow for virtual reality level design. *In 2020 IEEE Conference on Games (CoG)*. <https://doi.org/10.1109/cog47356.2020.9231769>

Contribution of first named author:

- Designed and programmed the LevelEd VR software solution in full.
- Planned and carried out the research study.

- Primary author of the paper, with some assistance from Dr Serban Pop on creating box plots and general feedback from Professor Nigel John.
- Presented paper remotely at IEEE Conference on Games in Osaka, Japan.

**Beever, L., & John, N. W. (2022).** LevelEd SR: A Substitutional Reality Level Design Workflow. *In 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (pp. 130-138).* IEEE. <https://doi.org/10.1109/vr51125.2022.00031>

Contribution of first named author:

- Designed and programmed the LevelEd SR software solution in full.
- Planned and carried out the research study.
- Primary author of the paper, with general feedback from Professor Nigel John.
- Presented paper remotely at IEEE VR in Christchurch, New Zealand.

## 1.5 Thesis Structure

The remainder of this thesis is broken down as follows:

**Chapter 2** features a detailed background overview of AR, VR and SR concepts, including their definitions. It also features a comprehensive review of past and current AR and VR hardware. Finally, it incorporates a detailed explanation of the level design process and how this differs for VR games.

**Chapter 3** features a literature review that presents a state-of-the-art review of related previous work utilising AR and VR technologies, and SR concepts in level design.

**Chapter 4** first details the implementation of Workflow 1 which utilises AR in the form of a smartphone application entitled LevelEd AR to help map a real-world location. A comparative study was carried out and the methodology of this is discussed in the chapter along with an analysis of the results and a final discussion.

**Chapter 5** covers the implementation of Workflow 2 and details a VR application that enables VR level design called LevelEd VR. A study was devised to evaluate the application and Workflow 2 in comparison to an existing level design workflow. The results of this study are analysed and discussed in this chapter.

**Chapter 6** describes the final study of the project which brings together work from the previous two chapters to enable Workflow 3, which is a SR level design workflow. The system developed is entitled LevelEd SR and the implementation is discussed in this chapter. The chapter also outlines the study that was developed and carried out and the results are analysed and discussed.

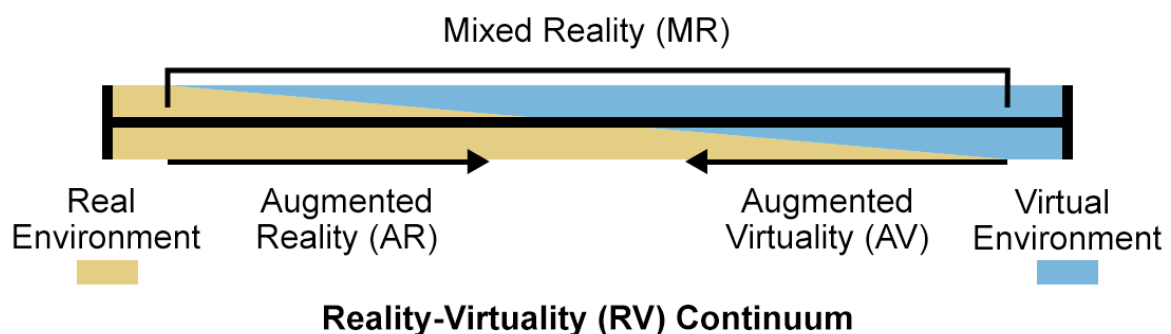
**Chapter 7** concludes the thesis by outlining the key learnings from this exploration of mixed reality (MR) level design workflows. The chapter reviews the research question introduced in this chapter, the hypotheses introduced in each workflow chapter and any additional lessons learnt along with any future work since completed or planned for the future.

# Chapter 2 Background

This chapter begins by defining the key technologies and extended realities explored as part of this thesis. A thorough state of the art review of augmented, virtual, and substitutional reality technologies and hardware was undertaken is then outlined in this chapter. The chapter concludes with an overview of the level design discipline, processes, and workflows.

## 2.1 Extended Reality

Extended reality (XR) is a term that is often used to describe a multitude of different immersive experiences and technologies. Seminal work from Milgram and Kishino (1994) is often cited when discussing extended or mixed reality. Part of their early contribution was the Reality-Virtuality Continuum (see **Figure 3**).



*Figure 3 - The reality-virtuality continuum outlines the level of inclusion of either the real environment or a fully virtual environment on each category. Adapted from Milgram and Kishino (1994) and Skarbez et al. (2021).*

This diagram shows that the term mixed reality covers AR and augmented virtuality (AV), but not VR (which comes under fully virtual environments). However, VR is still part of the Reality-Virtuality Continuum and so this diagram is useful in understanding how extended reality experiences such as AR, augmented virtuality (including SR) and VR are related but also differ. This thesis covers all three of these areas and they are first defined in this section and covered in more detail in later sections.

### 2.1.1 Augmented Reality

The Reality-Virtuality Continuum presents AR to the far left of the diagram, close to the real-environment (see **Figure 3**). This suggests that AR presents the real environment but with the inclusion of virtual artefacts to some degree. A commonly accepted definition of AR was proposed by Azuma (1997) and describes AR has having the following three characteristics:

- Combines real and virtual.
- Interactive in real time.
- Registered in 3D.

This definition suggests therefore that AR takes the real world (whether viewed directly or via video feed) and augments (enhances) it with virtual artefacts. These artefacts should be viewable in real-time with some degree of interactivity and must be registered to the real-world (Schmalstieg & Höllerer, 2016, p. 3). For example, if a system presents a digital character via AR in the real-world, the user should be able to walk around the character and for it to stay anchored in its location as if it is present in the real world. The benefits of AR are the ability to view additional data as part of the real-world. This could be in the form of entertainment, such as game characters/content, like Pokémon GO (Niantic, 2016). There are also more serious uses such as displaying information about a location or item in view, real-time data updates and remote assistance (Microsoft, 2022a). The effectiveness of an AR experience depends on the clarity of the digital artefacts and how believable they are as being present in the real-world. This depends on the quality of display to present the digital artefacts, the quality of the tracking of the user's movement and the degree to which the system understands the real world (such as location of physical surfaces and categorisation of objects in the space). These topics are covered further in the state-of-the-art section along with current technological advancements.

### 2.1.2 Immersion and Presence

Before defining the remaining experiences, it is worth exploring two key terms related to these different experiences:

- Immersion
- Presence

Immersion is “the objective degree to which a VR system and application projects stimuli onto the sensory receptors of users in a way that is extensive, matching, surrounding, vivid, interactive and plot-informing” (Slater & Wilbur, 1997).

These six key areas are described as (Jerald, 2015, p. 45):

- **Extensiveness** – The range of different sensory inputs presented to the player, such as visuals, audio and tactile.
- **Matching** – The agreement between the different sensory inputs and one's own body.
- **Surround** – The degree to which cues are panoramic, such as the range of field of view or surround sound.
- **Vividness** – The quality of the simulation, such as resolution, lighting, framerate, etc.
- **Interactable** – The ability for the user to make changes to the world and objects within it and for the experiences to respond to user input.



- **Plot** – The story presented by the experience through a consistent portrayal of its message and dynamic nature of the sequence of events.

Immersion is therefore the objective ability of a given system to deliver on the above categories. How the above six categories are implemented impacts a device, or experience's, ability to engage a user effectively and convincingly in VR. However, as immersion is more objective, it does not cover how a user subjectively experiences immersion or an experience. This subjectivity is known as presence.

It is difficult to describe presence in words and is often said that it needs to be experienced. However, Jerald (2015) describes presence as “a sense of ‘being there’ inside a space, even when physically located in a different location” (p. 46). A more accepted formal definition is as follows:

Presence is a psychological state or subjective perception in which even though part or all of an individual's current experience is generated by and/or filtered through human-made technology, part or all of the individuals' perception fails to accurately acknowledge the role of the technology in the experience. (International Society for Presence Research, 2000).

The degree of presence in an immersive experience could therefore be described as: to what extent the user perceives they are actually in a virtual space and the ability for them to forget the real-world. A user who might feel highly present in a VR experience might describe it as having visited the place, rather than something merely perceived (Jerald, 2015, p. 46). The ultimate target of immersive experiences is to enable as high a level of presence as possible and to avoid any breaks-in presence. Increasing the immersion of a system or experience, will often lead to a higher potential of presence and could be achieved through increasing the extensiveness (adding more sensory inputs like tactile haptics), improving the vividness (quality of the simulation) or any of the other six categories. These two terms are key components of both VR and AR.

### **2.1.3 Virtual Reality**

Jerald (2015) describes virtual reality as “a computer-generated digital environment that can be experienced and interacted with as if that environment were real” (p. 9). Therefore, VR would be on the opposite end of the Reality-Virtuality Continuum to AR, under virtual environment, as it relates to experiencing a fully virtual environment. An ideal VR system would allow a user to physically walk around objects and touch those objects as if they were real (Jerald, 2015, p. 9). However, technology is not at that level yet, but there are many ways to experience immersive VR, and these are discussed in more detail in section 2.3. At present, the core

ideas behind fully immersive VR technologies are to enable a user to experience a full VR environment, in isolation (free from other visual stimuli) and to allow some degree of interaction. Through these core ideas, VR currently enables several different experiences, such as:

- **Training** – Users can learn how to complete a task or learn a new skill by practicing in an immersive virtual environment. For example, a VR CPR training tool for school children (Rees et al., 2021)
- **Simulation** – Users can experience a simulated environment such as a relaxing forest or a digitally recreated real-world location that would normally require travel to visit. For example, TRIPP, which creates simulated dreamscapes to aid with meditation (TRIPP, Inc, 2019)
- **Entertainment** – Users can play interactive and immersive games that take place in fully virtual environments. Such as the popular rhythm game, Beat Saber, where users slash light sword beams to break blocks to the beat (Beat Games, 2019).

VR experiences are usually measured by their level of immersion and presence, as outlined in the previous section. This is ultimately driven by the hardware device used, but also how the visual quality and degree of interaction are presented by the virtual environment. These experiences are typically created using a game engine, such as Unity or Unreal Engine, both of which support high-fidelity visuals and support for VR devices and interaction. They are also less commonly created with bespoke software such as WorldViz Vizard, or public domain software such as VR Juggler and with web-based technologies through the WebXR specification.

#### **2.1.4 Substitutional Reality**

When reviewing the Reality-Virtuality Continuum there is an area denoted by the term Augmented Virtuality (AV) (see **Figure 3**). This is an area where predominantly virtual environments are augmented by the real-world. This term covers several distinct aspects such as augmenting a virtual environment with passive haptics (from real-world objects), including data from real-world systems, producing scents for olfactory input and potentially even higher fidelity data of human participants, such as additional trackers or motion capture. Substitutional reality focuses on a more specific area of augmented virtuality (AV) and can be described as:

In an SR environment, every physical object and architectural feature is replaced in the Virtual Environment (VE) by a counterpart. This leverages the physicality of the real world to provide tangibility to virtual objects and contributes to a higher sense of presence. (Simeone et al., 2015)

Whilst the term was originally coined by Suzuki et al in 2012, it was refined by Simeone in 2015 and has since been used to group works together that primarily focus on augmenting virtual environments with passive haptics. The main benefit of SR is the potential for increased immersion and therefore the potential for a higher level of presence experienced by the users. Early work by Insko (2001) and Meehan et al. (2002) showed an increase in presence when passive haptics were present in serious virtual experiences. The difficulty with supporting SR is in registering the virtual environment (that includes digital proxies) with the real-world objects.

The following two sections outline the current state of the art AR and VR devices that enable AR, VR and ultimately SR experiences. They are then followed by a section that defines the level design role and process, along with specific differences between VR and non-VR game design concepts.

## 2.2 Augmented Reality State of the Art

One of the earliest digital AR devices is Sutherland's (1968) Sword of Damocles which used see-through optics to display digital images, such as primitive 3D shapes. It is also often cited as one of the earliest VR headsets but is now better categorised as an AR headset. This mis-categorisation may in part be down to the term AR not being coined until 1992 with work from Caudell and Mizell, which demonstrated an AR system used in a Boeing factory to assist in assembling wire bundles.

Developments continued in labs with several different projects until Kato and Billingshurst released an open-source software platform in 1999 called ARToolKit. This platform included a 3D tracking library which could track fiducial markers with standard webcams and opened development to a wider audience. The development and evolution of mobile phones and mobile computers saw a move to develop AR software for mobile devices. Wagner and Schmalstieg (2003) demonstrated an AR system working on a personal digital assistant (PDA). This work ultimately evolved into the popular Vuforia AR platform commonly used on mobile until recent developments from Apple and Google. These developments have all lead to the current state of the art AR technologies and devices.

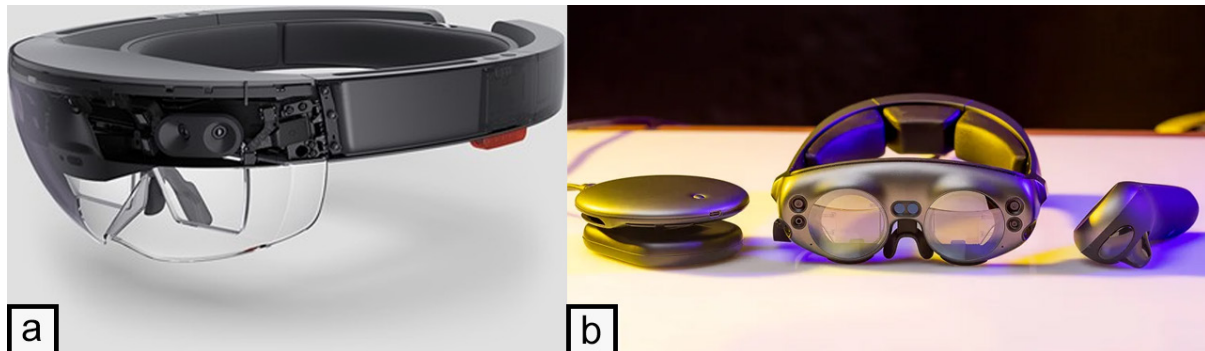
Current state of the art AR devices can be sub-categorised into two areas:

- **Head-mounted AR** – These are devices which are worn on the head with either optical lenses used to display digital images on top of the real world or via passthrough cameras (which are discussed separately in in **2.3.3**).

- **Smartphone AR** – This is enabled by technologies, such as ARKit, ARCore and WebXR (Butcher et al., 2020), that use the camera feed from a smartphone and augments the picture returned to the user.

### 2.2.1 Head-mounted AR

When focusing on head-mounted AR, two of the more popular, but now superseded devices are that of Microsoft’s HoloLens, released in 2016 and the Magic Leap One which launched in 2018 (see **Figure 4**) (Microsoft, 2021; Greenwald, 2018).



*Figure 4 - (a) Microsoft HoloLens headset (Microsoft, 2021). (b) Magic Leap One headset (Greenwald, 2018).*

Neither device targeted consumers initially, with the HoloLens targeting primarily business users (with tools such as remote assist) and the Magic Leap One targeting developers and very early adopters. This was reflected in their starting price points of \$3000 and \$2295 respectively. Both devices are untethered as they do not require a connection to a PC or laptop device. The hardware is fully integrated into the HoloLens headset and a small Compute Pack is provided with the Magic Leap One. This separate Compute Pack benefits the Magic Leap One with a more compact form factor and lighter design (579g – HoloLens vs 316g – Magic Leap One). This is an important advantage when users are expected to wear an AR headset for longer periods of time. As the devices are untethered, they can be used in almost any location without the need for bulky additional hardware or cables restricting movement. Both systems can build a digital map of the space, which gives each headset an understanding of the real-world location in terms of the placement of walls and obstacles within the space. The HoloLens focuses on using hand-tracking with gestures and voice for input, whilst the Magic Leap One offers the same inputs but also an additional hand controller, which allows for more accurate and familiar input for users.

Both devices have since been replaced by 2<sup>nd</sup> generation devices in the form of the HoloLens 2 and the Magic Leap 2. The HoloLens 2 was released in 2019 and continues to target business users, which is again reflected in the \$3500 price point and available software. The system has improved specifications, such as a more powerful processor, more memory (4GB

vs 2GB), a larger field of view (FoV), improved hand tracking and gesture support, and now supports eye tracking (for input and iris authentication) (Microsoft, 2022b). Meanwhile, the Magic Leap 2 was released in 2022 and saw a price increase to \$3299 and a focus on business/enterprise users. As with the HoloLens 2, it saw several specification improvements, such as a decrease in weight (from 316g down to 260g – minus the computer unit), increase in RAM (8GB up to 16GB), improved processing and a larger FoV (Magic Leap, 2022). In both cases, these improvements are primarily iterative, rather than introducing any new major features.

Whilst these are not the only head-mounted AR headsets on the market, they are the two most popular brands. There are a few different alternatives though, such as:

- **Nreal Light** – A compact headset that when connected to a supported smartphone can produce simple MR experiences (Robertson, 2021). It features two cameras for tracking and birdbath optics (OLED screens reflected on mirrors).
- **Lenovo ThinkReality A3** – A similarly compact headset that can be connected to either a compatible smartphone or a PC. The primary use case for the device is for working with multiple virtual PC monitors (Rubin, 2022).
- **Project North Star** – An open-source headset that can be created from off the shelf parts and with a 3D printer, as well as software to run the device. It is made in part with Leap Motion and includes their popular hand-tracking hardware and software (Project North Star, 2022).

A core benefit to head-mounted AR headsets is the ability to augment the world around the user wherever they are looking and doing so hands free. This opens several key use cases. For example, remote support allows a user to get support from an expert remotely, where the expert can see what the user sees (via cameras on the devices) and then augment the users' view (with pointers or additional data) to help solve issues (Microsoft, 2022a). Collaborative experiences can also be supported, for example, multiple users in a meeting or on site (see **Figure 5**) can all be viewing the same digital model which they can interact with, whilst also being able to easily talk with their colleagues in the room or take notes (Trimble, 2022).

The headsets can also be used in medical settings, such as giving surgeons an augmented view of a patient and their data (Pose-Díez-de-la-Lastra, 2022). In each case, these are all improved or only supported due to the hands-free head-mounted design.



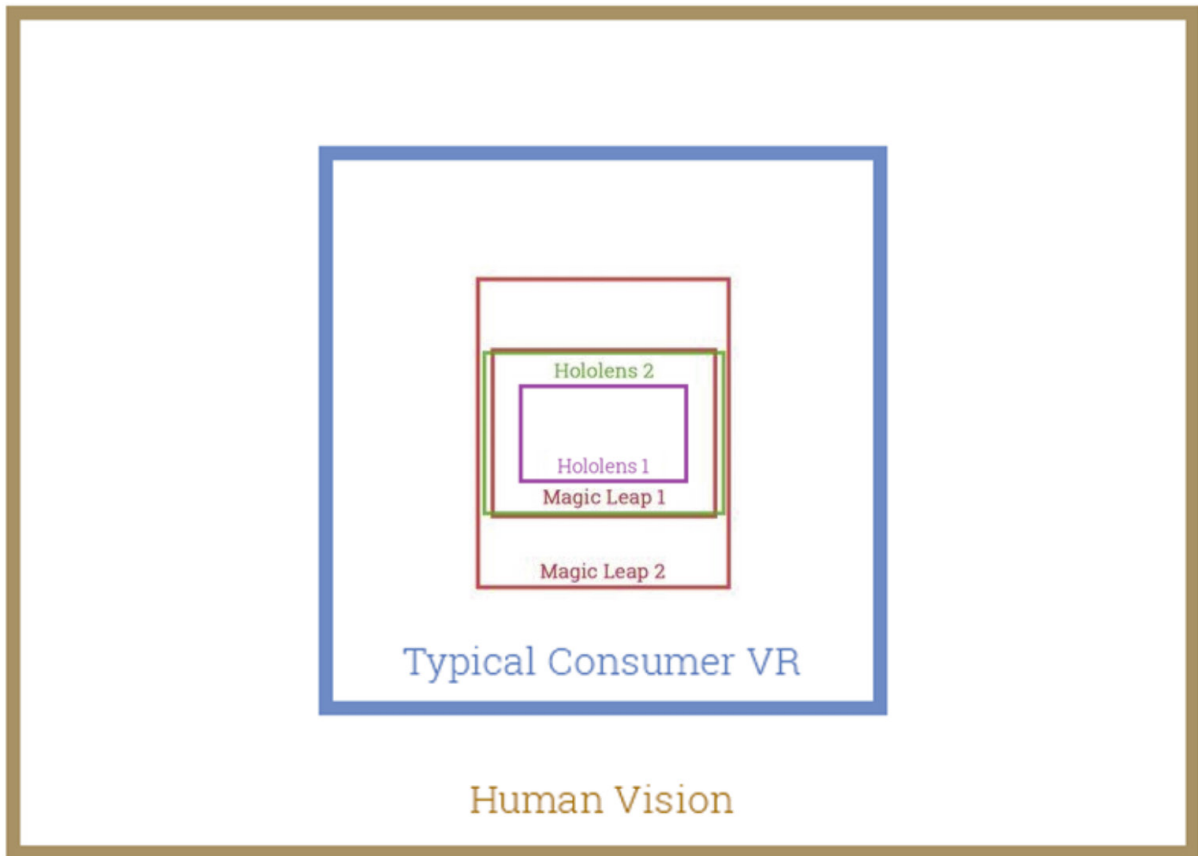
*Figure 5 – Trimble Connect allows users to collaboratively view and interact with digital models whilst on site (Trimble, 2022).*

Whilst all these devices have brought great advancements to the head-mounted AR market, those with optical lenses all suffer from similar limitations in their respective FoVs. The FoV represents how much of the users’ vision can be used to display digital artefacts, the smaller the FoV, the less of the augmented image is visible to the user and therefore limits the effectiveness of the technologies.

*Table 1 - Field of Views (FoV) for the four main AR headsets.*

Device	Field of View (FoV)
<b>HoloLens (2016)</b>	30° x 17.5°
<b>HoloLens 2 (2019)</b>	43° x 29
<b>Magic Leap One (2018)</b>	40° x 30°
<b>Magic Leap 2 (2022)</b>	44° x 53°

It is clear from **Table 1** that improvements have been made to the FoV over time, with Magic Leap 2 displaying the largest FoV of these devices. However, **Figure 6** demonstrates the FoV for all four of these devices compared to the average VR headset and a rough representation of human vision. It is therefore clear that there is still a lot of improvement required to get to a more usable field of view.



*Figure 6 - Estimated field of view comparison between the four main devices compared to most VR headsets and human vision. Image constructed from both (Heaney, 2019) and (Ganesan, 2022).*

The development of head-mounted AR headsets will no doubt continue. For example, Qualcomm (a popular mobile chip manufacturer) is continuing to develop chipsets for AR headsets and are working with Niantic (creator of the popular Pokémon GO AR smartphone game) on an outdoor focused headset (Lang, 2022b). However, the more recent development trend are hybrid MR devices that can deliver both VR and AR content (normally through pass-through cameras). These devices are outlined in **2.3.3**.

### **2.2.2 Smartphone AR**

The second category of AR technologies utilises existing consumer devices in the form of smartphones along with a new set of application programming interfaces (APIs). The two main APIs are Apple's ARKit that works on Apple devices such as the iPhone and iPad and then Google's ARCore, which works on a wide range of Android OS devices. Both technologies were introduced in 2017. They both work using similar technologies and have progressed at a similar rate with regards to new features.

The underlying technologies utilise both computer vision and device sensors to calculate the orientation of a device and any subsequent change in position via movement that a user might make. This is often referred to as visual inertial odometry (VIO) and is made up from two types



of data capture. The first uses the gyroscope and accelerometer on the device to track orientation and motion, which is then matched with the camera image (see **Figure 7**). This is useful for short term movements and can be gathered every frame, but it does not include enough data to maintain accuracy over longer periods of time or movement (Apple Inc. 2018).



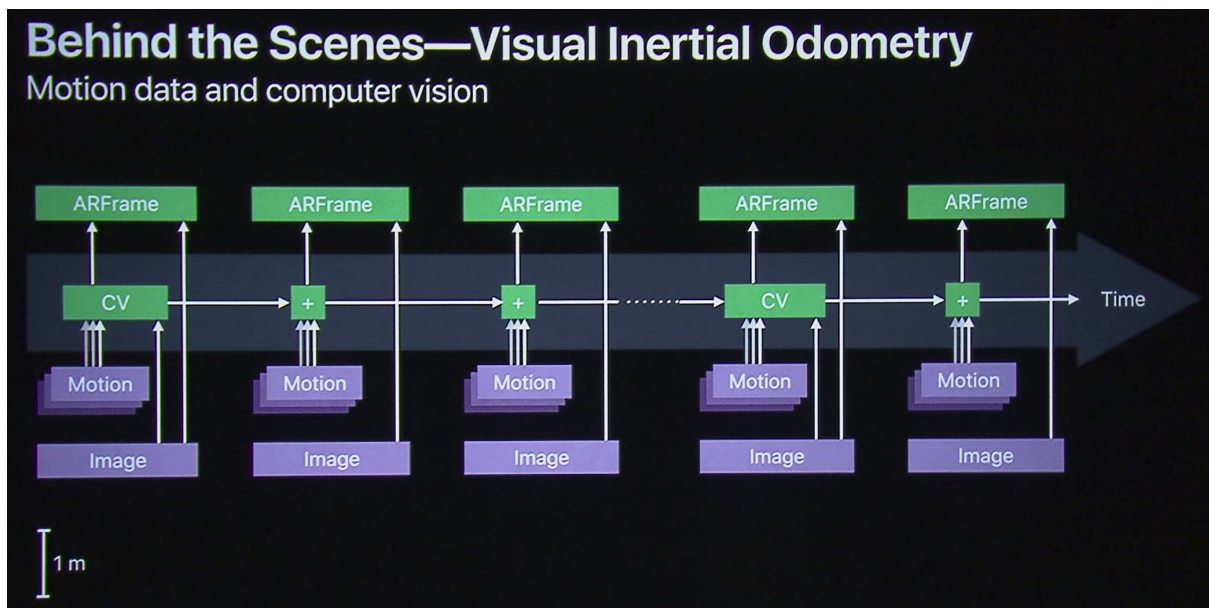
*Figure 7 - Example of ARKit detecting the pose (orientation and position) of the smartphone device in the location using sensor data.*

The second type of data capture uses computer vision to supplement the sensor data. This additional data is generated over several frames (as it is computationally expensive) and is used to detect changes in the image frames in order to track points of interest (see **Figure 8**). The points of interest are triangulated and require movement of the device, such as sideways movements rather than rotation, to function (Apple Inc., 2018).



*Figure 8 - Computer vision is used to track changes to points of interest (blue, red and orange dots) each frame. Changes to the device pose can be triangulated from these points and their changes over time.*





*Figure 9 - Visual Inertial Odometry utilises computer vision data over time along with up-to-date position and orientation data from device sensors (Apple Inc., 2018).*

The data from the sensors and the computer vision is fused to create a system that can track pose changes over longer periods of time, but also deal better with issues such as devices making fast movements (see **Figure 9**).

The system continues to capture data as the device moves in order to build up an understanding of the space (based on points of interest) and can correct some discrepancies and drift when returning back to poses previously captured. This therefore means ARKit utilises a form of Simultaneous Localization and Mapping (SLAM) to create a map of the space (Riisgaard & Blas, 2003). The end result is the ARKit technology can more generically be referred to as a system that uses Visual Inertial SLAM.

Utilising Visual Inertial SLAM and additional software features enabled the first versions of ARKit (and similarly ARCore) to track the pose of the device and its subsequent movement in an environment, to detect flat horizontal surfaces as planes (plane detection), to anchor a virtual object in place (on a plane once detected) and estimate the scene light through light estimation. Light estimation enables ARKit to estimate the brightness of the light in the scene to help match the virtual lighting. This results in virtual models that stand out less as objects appear more naturally lit and more part of the real-world.

Since the initial beta release of ARKit in June 2017, several different features have been added to ARKit to support more immersive AR experiences. In each case, ARKit has gained more understanding of the world around it. Below is a summary of the key developments of ARKit with each new annual release (Bohon, 2021; Apple Inc, 2022b):

- **ARKit 2** – Multiplayer/collaborative support was added along with the ability to track 2D images (image recognition and tracking) and detecting 3D objects known to the system (models had to be pre-scanned and included in the application).
- **ARKit 3** – People occlusion was added to ensure augmented digital artefacts could be occluded by people if they walk in front of them. Body tracking was added, and face tracking also added on devices with a FaceID camera. FaceID tracking enabled support for cheap and accessible facial motion capture in software such as Unreal Engine and Unity (Tanenbaum, 2021).
- **ARKit 4** – LiDAR was added to some Apple devices (such as the iPad Pro) and so depth information could be gathered if compatible hardware was used. The depth information allowed for more accurate tracking and object detection. Location Anchors were also added so objects could be placed at a specific geographical location using Apple Maps.
- **ARKit 5** –The ability to generate procedural meshes of the real world to use for object occlusion and physics was added. It also added APIs to allow scanning of an object to turn into a 3D model (mesh and texture data) and saved as a USDZ file.
- **ARKit 6** – 4K video was added, which results in a much clearer video image. There were improvements to Depth API and LiDAR data and a better overall understanding of the environment by identifying categories of objects.

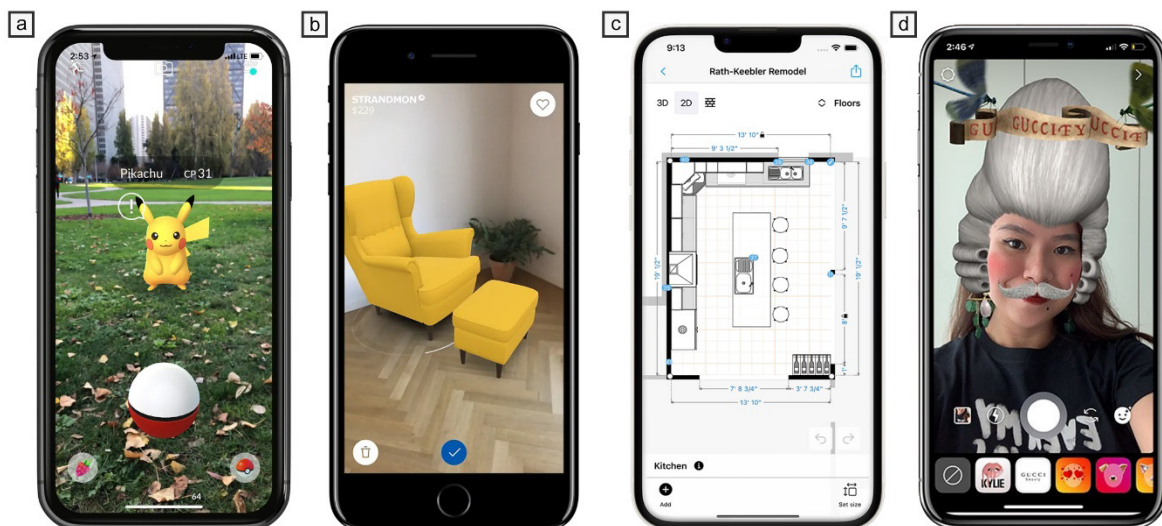
Whilst ARCore has not followed the timeline exactly as above, it has continued to have similar feature parity with ARKit up until now.

Whilst both systems now have similar features sets, supporting both platforms separately can be time consuming, especially if done natively using XCode (ARKit) and Android Studio (ARCore). One solution to this is to use a game engine such as Unity to create AR applications for mobile. Unity is popular for AR application development due to its quick support of new ARKit/ARCore features through developer SDKs, 3D rendering, intuitive editor and pre-existing mobile device support. To further improve the development pipeline, Unity has also added a unified middleware framework called ARFoundation that can be used to support common features between both ARKit and ARCore. The key benefit of this framework is the ability to support both platforms on different mobile devices through a single unified code base. This was initially released in November 2018, but took a while to include the core ARKit and ARCore features but is now regularly updated with new features to both platforms.

One of the core benefits of smartphone-based AR over head-mounted AR headsets is that they make use of consumer technology that a wide audience of users can already access. There is no need to buy additional, expensive hardware in order to participate in AR

experiences as most modern smartphone devices can support either ARKit or ARCore. Sales data for the Magic Leap One and HoloLens aren't easily available, but early estimates suggested only 6000 Magic Leap One devices sold in the first 6-months (Hayden, 2019b) and HoloLens has sold around 300,000 as of 2022 (Giret, 2022). Whereas some estimates suggest there are around 1.5 billion devices compatible with ARKit and ARCore (Révész, 2021). This opens AR experiences up to a much wider audience than head-mounted AR devices. AR technologies can therefore be integrated into existing smartphone applications and can also make use of device input technologies, such as the touch screen, that users will already be familiar with. As the technology is also primarily software based, new features can be added quickly, without new hardware required, as demonstrated with the previous ARKit timeline.

As ARKit and ARCore have evolved, so to have their use cases. Some early examples used AR as part of video games, such as with an update to the popular Pokemon GO game that sees wild Pokemon characters appear in an AR view to make it look as though they are in the real-world location (see **Figure 10a**). IKEA Place utilises ARKit to allow users to try out furniture in their own home with accurately sized models of IKEA furniture including accurate material rendering and light estimation (see **Figure 10b**). Magicplan can be used to generate technical floorplans using an AR view to map out the initial space to record walls, doors and windows (see **Figure 10c**). Finally, AR is now commonly found in applications like Snapchat, TikTok and Instagram to add AR filters and effects to a user's face (see **Figure 10d**).



**Figure 10** – (a) *Pokemon GO* uses ARKit to place digital Pokemon characters into the real-world (Fulton, 2017). (b) *IKEA Place* lets users view IKEA furniture in their own home (Keller, 2017). (c) *Magicplan* allows users to create technical floor plans through and AR camera view (Sensopia, 2022). (d) Users can apply AR filters in *Instagram* that track their face movements (Evans, 2020).

Despite their many benefits and use cases, there are several limitations with smartphone-based AR technologies. As the hardware is typically not head-mounted, it requires users to

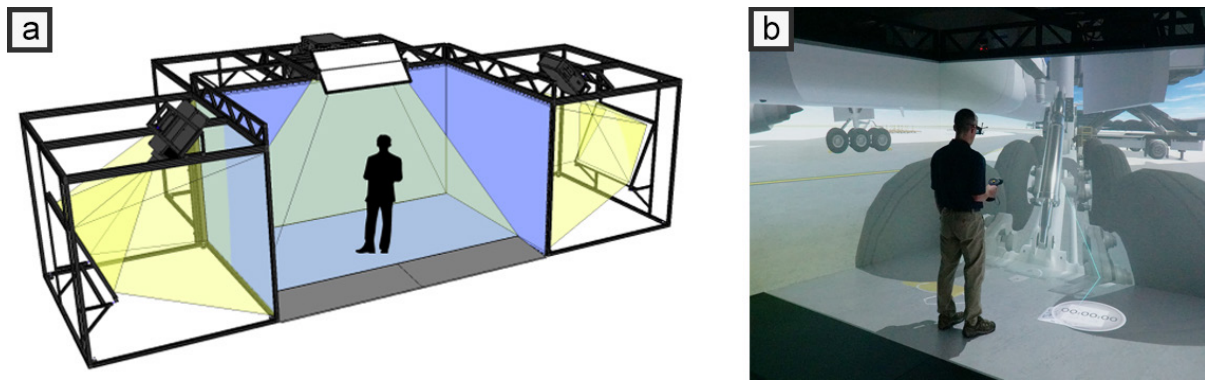
hold the device to move the augmented camera view and to interact (although Zappar (2023) attempts to solve this with Zapbox). This ultimately limits where the technology can be used as users may only have one hand free or in some cases no hands to interact with the real-world.

One of the use cases for head-mounted AR devices is the ability to use them for medical uses, such as surgery. The ability to use smartphone-based AR devices for this is limited due to the user having to hold the device and touch the screen to interact. However, where interaction with the real world is not necessary, it is less of a limitation. The issues of field of view are also similarly present here. The augmented view is only visible through the devices screen via the camera feed. Whilst the camera feed FoV can be quite wide, the actual view is limited to a small screen held in front of the user. This limitation reduces the level of immersion as users will be aware that they are viewing this augmented view on a small screen, rather than in the actual space, such as with head-mounted AR devices or fully enclosed VR headsets.

An alternative method for utilising AR on a smartphone device or standalone AR head-mounted displays is through WebXR. WebXR is a specification developed by the World Wide Web Consortium (W3C) that supports both AR and VR in supported web browsers (W3C, 2023). The specification began with WebVR experiments, which demonstrated that a web specification could work. However, it was deemed necessary that a new specification be created that covers both AR and VR, with a new API (Mozilla, 2023). One key benefit to WebXR is the ease of accessing an application or experience. Users can visit a URL in a supported web browser, which will then begin the application or experience, negating the need for lengthy downloads and installation of software. There is also the benefit of not needing to publish an application on specific stores to support multiple devices. However, supported AR features are currently limited, with support for features like plane detection in an experimental phase and other common features found in ARKit and ARCore, missing entirely (W3C, 2022b).

### **2.3 Virtual Reality State of the Art**

When compared to AR devices, there are many more options to experience VR. The most popular method is through a head mounted display, which when worn, will block out the outside world and create an immersive visual experience for users. However, there are other options which are more expensive and aimed at enterprise users such as world-fixed displays, which encompass systems such as Cave Automatic Virtual Environments (CAVE). CAVE systems often utilise rear-projection screens to present the virtual environment to curved screens or a partially enclosed square room (see **Figure 11**).



**Figure 11** - A high-resolution CAVE system as presented by Visbox (2023). (a) Shows the how the system is configured. (b) Shows the CAVE system in use.

The movement of the user's head is tracked using traditional motion capture technology which enables the synchronisation of the projectors during the experience. Stereoscopic projectors are used along with either passive stereo polarised or active stereo shutter glasses to enable a stereoscopic 3D view. The main issue with CAVE systems (and potentially why their use is limited) is the initial cost of the system, such as the multiple projectors and motion capture tracking system and the space required for the setup.

At present, most consumer VR experiences are delivered through VR head-mounted displays (HMD). The main benefit of VR headsets is their ability to fully immerse the player by blocking the outside world in full. Most modern VR headsets also support full rotation and position tracking of the headset and dual hand controllers. This is often referred to as 6-Degrees of Freedom (6-DoF) tracking and ensures any movement from the user is reflected in the rendered virtual environment. Headsets are usually comprised of two lenses in which the virtual environment is viewed. These two lenses offer two different views of the environment at different angles, and it is due to this, that VR headsets can deliver stereoscopic 3D. This gives the user the sense of depth perception, despite the actual virtual environment being rendered on flat displays behind the lenses. Most modern headsets also support two 6-DoF tracked hand controllers. This allows users to interact with virtual environments more naturally through simply moving their hands. This enables experiences such as fitness applications (like boxing), escape the room games, flight simulators, puzzle games, first person shooters (FPS) and many more where the user can interact with objects in an environment with their hands. Rudimentary finger detection is common (through sensors on the buttons of a controller) which allow users to point, pinch and grab objects for more natural interactions.

Whilst HMD VR devices can offer effective immersive experiences, there are some areas of concern that must be addressed. One key area of concern is that of simulator sickness (also known as cyber sickness). Simulator sickness can cause users of a VR HMD to feel symptoms such as, nausea, sweating, fatigue, headaches and many more (Kennedy et al., 1993). If a

user experiences these symptoms, they will likely want to stop and may avoid VR again in the future. Therefore, simulator sickness must be avoided wherever possible. There is no definitive theory as to what causes simulator sickness, but there is one prevailing theory devised by Reason and Brand (1975) known as Sensory Conflict Theory (Langbehn et al, 2018; Jerald, 2015, p165). This theory suggests that when users experience a change in movement (acceleration sometimes known asvection) and in particular rotation, there is a mismatch between what the user is seeing (in the HMD) and what their otolith organs (within the vestibular system) sense and this mismatch contributes to simulator sickness (Jerald, 2015, p. 164). This is likely to occur where there is motion in a scene, which could be:

- **Intentional scene motion** – Where a user can move around the 3D environment and has been allowed to do so by the software.
- **Unintentional scene motion** – This usually occurs with latency in headset tracking, inaccurate calibration, frame rate of the software, etc.

Unintentional scene motion can be avoided by using a more immersive headset, such as those discussed later in this section. It can also be avoided by ensuring a scene is not too complex to render and maintains a constant framerate/latency (Chang et al., 2020) and in most cases to deliver at least 60 frames per second (fps) but preferably 72-90 fps (Oculus, 2023).

Intentional scene motion can be controlled by the developer of the system by programming how the user can navigate a scene. This is often referred to as a locomotion technique. The locomotion technique selected can have a big impact on the opportunity for simulator sickness but also the quality of the experience (Líndal et al., 2018; Frommel et al., 2017). Some examples of locomotion techniques are (Martinez et al., 2022):

- **Real walking or room scale** – A user can move around their real-world play space whilst being tracked but are limited to only moving within their play space.
- **Redirected walking** – A user can walk in their real-world location but are redirected as they do so to enable them to explore larger spaces.
- **Point-and-teleport** – A user can point at a location using a motion tracked controller and teleport to this location, thus allowing them to explore large distances.
- **Joystick-directed steering** – Based on traditional games that allow for movement with a game pad. The user can control their movement and rotation indirectly from analogue thumb sticks. This typically suffers from higher chance of simulator sickness (Langbehn, et al., 2018).

Many early Oculus Rift VR games either avoided locomotion altogether or tried to use joystick-directed steering with limited success (Mailberg, 2016). Developers then began to incorporate

teleportation to move, as this avoids most of the simulator sickness causes and includes successful examples such as Robo Recall (Epic Games, 2017) and Budget Cuts (Neat Corporation, 2018). More recently games have started to offer multiple options for users (Lang, 2020). This enables players to decide which locomotion they prefer to use as some users appear to be impacted less than others by simulator sickness (whether naturally or through repeated exposure). There are also several additional techniques that have been deployed to alleviate simulator sickness with locomotion techniques, like blurring the peripheral vision whilst rotating or accelerating or using snap rotation (Fernandes & Feiner, 2016). The issue of simulator sickness and locomotion choice must be carefully considered, along with the quality of the technology (discussed in the next section) when designing any VR experience.

The most common tools used to create modern VR applications are game engines, such as Unity and Unreal Engine. It is estimated that 60-70% of games on AR/VR platforms and an estimated 50% of games across other devices are made with Unity (Dillet, 2018; Unity, 2022). Both Unity and Unreal Engine support development of native VR applications for popular headsets, such as the Meta Quest 2, through specific platform SDKs or OpenXR. An alternative method for developing VR applications is with WebXR. As noted in the previous section, this supports VR experience that can be accessed via a URL in a supported web browser. The same benefits apply to VR as they do to AR, namely, the ease of accessing an application or experience, no download or installation and not being required to publish an application through platform specific store front. WebXR support for VR is further along than AR, as it supports not only stereo rendering, but also interaction through 6-DoF headset tracking, a range of 6-DoF controller tracking and early hand tracking (W3C, 2022a).

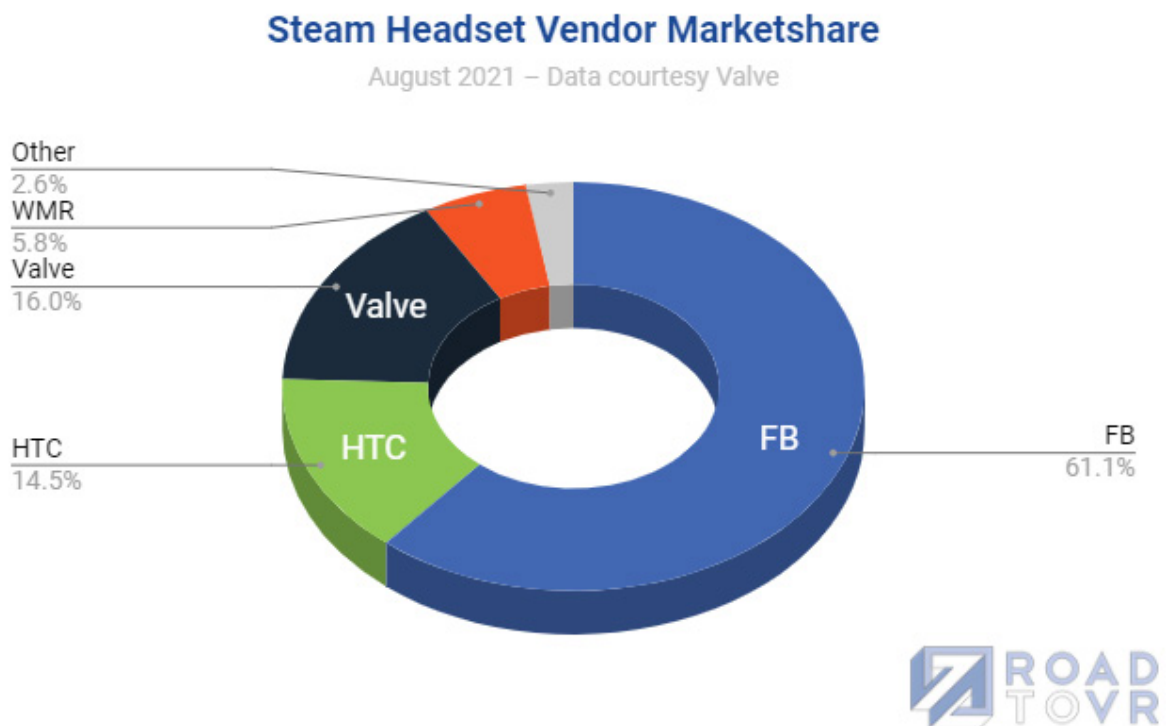
There are many different types of VR head-mounted displays available. VRCompare (2023), an XR comparison site, currently lists 149 different VR headsets, with the vast majority released since 2014. This section will focus on the more popular devices which have had the biggest impact on the VR industry. In general, there are four main popular platforms/manufacturers of VR headsets, these are as follows:

- **Meta (formerly Oculus)** – They have created several popular headsets as well as a full VR operating system, closed app/games store and social systems.
- **HTC** – They started by partnering with Valve Corporation on the HTC Vive, but have since released several of their own headsets, which run on the Vive platform as well as with PCs and the Steam VR 2.0 accessories.
- **Valve Corporation** – After the launch of the HTC Vive they went on to develop their own headset, the Valve Index. It uses their own Steam VR platform and works with Steam VR 2.0 accessories and tracking systems.



- **Windows Mixed Reality (WMR)** – This is more of an open platform that 3<sup>rd</sup> party hardware companies can use to create their own headsets. This includes companies such as Samsung and HP, who both have hardware supporting WMR. The systems utilise Windows as their main OS platform via tethering.

Whilst specific sales data for the different headsets is limited, the popular VR games platform Steam offers data on the percentage of VR headsets connected and used with the platform as well as a breakdown of platforms/devices. As of August 2021, there was roughly 2.5 million headsets connected to PCs running Steam and Meta headsets accounted for most of these headsets (see **Figure 12**) (Lang, 2021b).



*Figure 12 - Market share of different VR platforms as of August 2021 (Lang, 2021b).*

More recent data shows that Meta have continued to increase their market share at the cost of HTC headsets (Valve Corporation, 2023). However, the four VR platforms have consistently held the top market share on Steam for many years.

To better understand the different headsets, they can be sub-categorised into the following categories:

- **Tethered Headsets** – must be connected to an external machine that executes the processing and rendering.
- **Standalone Headsets** – a headset that can execute the processing and rendering solely on the device.



- **Mixed Reality Headset** – a headset that can deliver both AR and VR experiences through one headset.

The pros and cons of these different categories are discussed in the following sections.

### 2.3.1 Tethered VR Headsets

A tethered headset is one that must be connected to an external system to be used (see examples in **Figure 13**). The external system, such as a PC, executes the processing and rendering of a VR experience and this data is usually transmitted to the VR display via a cable. The core benefit of this, is that PCs that are built to run high-end games with dedicated GPUs, can be used to deliver high-fidelity experiences to a VR headset. This additional rendering



**Figure 13** - (a) Oculus Quest CV1 (released 2016) with external tracking. (b) HP Reverb G2 (released 2021) with inside out tracking. (c) HTC Vive (released 2016) with external tracking. (d) PlayStation VR (released 2016) with external tracking via camera.

power plays a big role in the ability to render experiences at a high resolution and thus, deliver a more immersive experience. The visual quality of the games and experiences can also be improved with this additional rendering power.

One of the main disadvantages to tethered headsets is the need to be tethered to a PC to use the headset. This causes several issues. For example, it limits where the headset can be used as you will likely need external power (unless it is tethered to a laptop) to power the machine. If the VR experience needs relocating, then the machine powering the device needs to be moved as well, which reduces the portability of the system. The wire can become tangled when in use and limits the distance a user can move (although some wireless solutions are

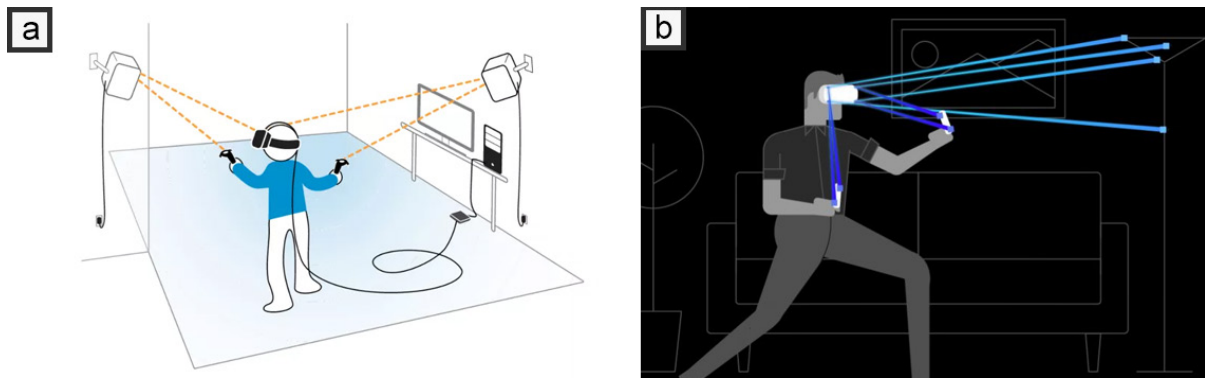
available to limit this issue). Finally, it also adds an additional cost to the VR setup as a powerful machine may need to be bought to use the headset if not already available.

**Table 2** below highlights some of the more popular tethered headsets along with their specifications for comparison. In general, tethered headsets have improved overtime, which is visible in areas such as the FoV, refresh rate and resolution per eye. There has also been a trend to move away from external trackers and towards inside out camera tracking.

*Table 2 - Common tethered VR headsets from 2016-2023. Sourced from VRCompare (2023). Extended version can be found in Appendix D.*

	Oculus Rift	HTC Vive	Playstation VR	Oculus Rift S	Valve Index
<b>Release Date</b>	March 2016	April 2016	October 2016	May 2019	May 2019
<b>Price</b>	\$599	\$799	\$299	\$399	\$999
<b>Horizontal FoV</b>	87°	108°	96°	88°	108°
<b>Refresh Rate</b>	90Hz	90Hz	120Hz	80Hz	144Hz
<b>Resolution (per eye)</b>	1080x1200	1080x1200	960 x 1080	1280x1440	1440 x 1600
<b>Tracking</b>	6-DoF external sensors	6-DoF external sensors (light houses)	6-DoF external sensors (PS Camera)	6-DoF inside out tracking – 5 cameras	6-DoF external sensors (light houses)
<b>Input</b>	Gamepad + 6-DoF Tracked controllers	6-DoF Tracked controllers	Gamepad + 6-DoF Tracked controllers	6-DoF Tracked controllers	6-DoF Tracked controllers
<b>Platform</b>	Oculus	Steam VR	Playstation 4	Oculus	Steam VR

Whilst external trackers were necessary early on to support room-scale VR experiences and offer more comprehensive tracking coverage, they require additional setup and further reduce the portability of a system. Inside out tracking solves this issue by using multiple cameras on the headset to track the movement and rotation of the device in a location (similar to how ARKit functions) and therefore works with minimal set up and no additional hardware (see **Figure 14**). Whilst tethered headsets continue to be released, such as the PlayStation VR2 (Feb 2023), most headsets released now tend to be standalone devices.



**Figure 14** - (a) HTC Vive that uses external Lighthouse sensors to create an IR laser beams that sweep on two axes (Allan, 2016). The Vive devices detect these beams and calculate the time it took to receive it. (b) Inside out tracking uses multiple cameras to detect points of interest and calculate their movement over time (Hamilton, 2019).

### 2.3.2 Standalone VR Headsets

A standalone headset is one which has all necessary processing power for it to be used on board via a mobile chip set. It can therefore be used independent of any PC or laptop, unlike tethered headsets, as all processing and rendering is completed on the device. Most of the first standalone devices utilised adapted smartphone chip sets and modified versions of the Android operating system. They supported only 3-DoF (rotation and not position tracking) through onboard gyroscopes in the headset and handheld remotes. This allowed for basic experiences but could not match those of tethered headset as the user's head or body movement was not tracked. The Oculus Quest was one of the earliest standalone devices to offer true 6-DoF tracking of both the headset and the controllers, which allowed the system to deliver similar experiences to tethered headsets (albeit at lower fidelity). More recent headsets are built on top of Qualcomm's Snapdragon XR platform, which is a chipset built specifically for AR/VR headsets. This platform has enabled several competing devices to launch, as is visible in the short summary of devices in **Table 3** and **Figure 15**.

The main advantage of standalone headsets is portability. As all processing and rendering is completed on the device and as they use inside out tracking for headset and controller tracking, the device can be freely moved to and used in various locations with ease and with minimal set up time. As there are no cables tethering the headset to another machine, there are no issues with getting tangled in the cable and the user is free to roam a larger distance (within limits of tracking capability). The cost of these devices has also been attractive to consumers, specifically with the Meta Quest 2 which launched at \$299 (since increased to \$399), as there is no additional equipment (like a PC) required to use them. These points are some of the reasons behind the success of the Quest 2, which has reportedly sold 15 million units as of June 2022 and is now the best-selling headset (Heaney, 2022).



**Figure 15** - (a) Oculus Go (released 2018) with only 3-DoF tracking. (b) Meta (formerly Oculus) Quest 2 (released 2020) with 6-DoF tracking and grayscale passthrough. (c) HTC Vive Focus 3 (released 2021) with 6-DoF tracking and grayscale passthrough. (d) HTC Vive Flow (released 2021) with 6-DoF tracking and is controlled via a smartphone or remote and is meant to be linked to a smartphone.

The main disadvantage of standalone headsets is their limited processing and rendering power. As all processing and rendering must be carried out on the headset, the processors and GPU must be small and power efficient, as the device must run off a battery and not produce significant heat. This means standalone headsets cannot hit the same levels of fidelity as with tethered headsets connected to a gaming PC. For example, *Half-Life: Alyx* (a popular tethered VR game) is not available on standalone headsets and would require work to get it to run. The battery life is also another issue, as most standalone headsets last between 2-3 hours, whereas a tethered headset does not have this limitation.

As with tethered headsets, the resolution of the screen, FoV, refresh rate and tracking capabilities have improved over time. Some devices, such as the Oculus Quest and Meta Quest 2 now support the ability to tether to another device (either via a cable or streaming over WiFi), like a PC, so they can gain the same benefits as noted in the previous section. This is likely why most recent devices fall into the standalone headset category as they can cater to the needs of both categories. Several more recent headsets have also gained passthrough capabilities. Passthrough utilises the cameras on the devices (usually used for tracking) to give the user a view of the outside world. This was primarily used to help during setup but has since evolved (post LevelEd SR study) into supporting MR experiences. These devices have been separated and discussed in the next section as a new evolving category.

**Table 3** - Common standalone VR headsets from 2018-2021. Sourced from VRCompare (2023). An extended version can be found in **Appendix E**.

	Oculus Go	HTC Vive Focus	Oculus Quest	Meta (previously Oculus) Quest 2	HTC Vive Flow
<b>Release Date</b>	May 2018	April 2019	May 2019	October 2020	Nov 2021
<b>Price</b>	\$199	\$599	\$399	\$399	\$499
<b>Horizontal FoV</b>	89°	110°	93°	97°	100°
<b>Refresh Rate</b>	60Hz	75Hz	72Hz	120Hz	75Hz
<b>Resolution (per eye)</b>	1280 x 1440	1440 x 1600	1440 x 1600	1832 x 1920	1600 x 1600
<b>Tracking</b>	3-DoF (rotation)	6-DoF inside out tracking – 2 cameras	6-DoF inside out tracking – 4 cameras	6-DoF inside out tracking – 4 cameras + Passthrough (grayscale)	6-DoF inside out tracking – 2 cameras + Passthrough (grayscale)
<b>Input</b>	3-DoF remote	3-DoF remote	6-DoF Tracked controllers	6-DoF Tracked controllers + Hand tracking	Input via smartphone + optional remote.
<b>Platform</b>	Oculus	HTC Viveport	Oculus	Oculus	HTC Viveport

### 2.3.3 Mixed Reality VR Headsets

MR or XR headsets are identified by their support for both VR and AR experiences, usually delivered through camera passthrough techniques. These headsets are primarily VR headsets, but by using the camera passthrough and similar techniques to ARKit and ARCore (like VIO SLAM), can support AR experiences such as anchoring digital artefacts into the real-world view (Meta Quest Blog, 2022).

These headsets share the same advantages and disadvantages as the previous standalone category (except for the Varjo XR-3 which is tethered) but with the added benefit that they can enable AR experiences. The key benefit to XR headsets over traditional AR is that they can be selective as to what is shown (real-world camera feed vs digital artefacts) which starts to move into MR, and they offer a much wider FoV (as seen in **Figure 6**) over traditional optical lens based AR headsets. The digital artefacts also look more visually solid as they are created with screens, rather than waveguide techniques as seen on the HoloLens (which creates digital artefacts more akin to holographs).

The disadvantage of XR headsets over more traditional AR devices is the quality of the camera feed is not comparable to human vision. Some systems also use traditional camera feeds



**Figure 16** – (a) Varjo XR-3 (released 2021) is a tethered MR device with marker-based tracking and 6-DoF via a beta. (b) Meta Quest Pro (released 2022) with MR support via 6-DoF tracking (5 cameras) and colour passthrough. (c) Lynx R-1 (released 2022) focuses primarily on AR but can support VR with 6-DoF tracking via 2 cameras and colour passthrough. (d) HTC Vive XR Elite (released 2023) with MR support via 6-DoF tracking with 4 cameras and colour passthrough.

**Table 4** - Common standalone XR/MR headsets from 2021 to present. Sourced from VRCompare (2023).

	Varjo XR-3	Pico 4	Meta Quest Pro	Lynx R-1	HTC Vive XR Elite
<b>Release Date</b>	January 2021	October 2022	October 2022	November 2022	March 2023
<b>Price</b>	\$5495	\$430	\$1500	\$948	\$1099
<b>Horizontal FoV</b>	115°	105°	106°	90°	110°
<b>Refresh Rate</b>	90Hz	90Hz	90Hz	190Hz	90Hz
<b>Resolution (per eye)</b>	2880 x 2720	2160 x 2160	1800 x 1920	1600 x 1600	1920 x 1920
<b>Tracking</b>	Marker based tracking + 6-DoF inside out tracking (beta) + Passthrough (colour) and LiDAR	6-DoF inside out tracking – 4 cameras + Passthrough (colour)	6-DoF inside out tracking – 5 cameras + Passthrough (colour)	6-DoF inside out tracking – 2 cameras + colour Passthrough (dual colour cameras)	6-DoF inside out tracking – 4 cameras + Passthrough (colour)
<b>Input</b>	Eye tracking + Hand tracking	6-DoF Tracked controllers + Hand tracking	6-DoF Tracked controllers + Hand tracking	6-DoF Tracked controllers + Hand tracking	6-DoF Tracked controllers + Hand tracking
<b>Platform</b>	SteamVR	Pico Store	Oculus/Meta	Proprietary	HTC Viveport

(rather than RGB-D) which are distorted to create depth. This limited resolution and distortion limits the immersive nature compared to traditional headsets or even ARKit or ARCore (which now support 4K camera feeds).

There are currently limited applications/games that utilise passthrough, but as these devices continue to grow, these are likely to expand. This is a new category of headsets and one which is likely to grow further as more vendors release devices, such as Apple.

It is clear from the previous state-of-the-art sections, that AR and VR devices continue to improve the technology available to consumers and businesses. Their use in research has also continued to increase along with the availability and release of new devices. Chapter 3 details similar work in level design/world-building across AR, VR and SR techniques. Prior to Chapter 3, this chapter continues with section **2.4**, which defines the level design process and role as well as game/level design differences between VR and non-VR platforms.

## **2.4 Level Design**

Game development is a complex process that requires many different disciplines to work together to create a game. One of these disciplines is that of game design. Game design is the process of designing the many components that when pieced together, form an interactive video game. Game designers must decide upon several components such as, but not limited to (Adams, 2014):

- Genre
- Narrative
- Characters
- Core mechanics/Gameplay
  - Actions
  - Challenges
- Goals
- User interface
- Levels/Game world

Whilst most of these decisions fall under the game designer role, levels and game worlds often require specialist skills and are instead completed by a level designer. Whilst the term level designer and game designer are not interchangeable, the roles are still tightly knit together and must work in tandem. Adams (2014, p. 359) describes level design as “the process of constructing the experience that will be offered directly to the player using components provided by the game designer”. This describes how game designers decide on individual components for a game, such as abilities of the main character and enemies they may face

for example, but that it is the level designer that constructs the experience using these components for the player. Kremers (2009) agrees by defining level design as "...to interpret the game rules, and to translate them into a construct (a level) that best facilitates play. Another way of expressing this is by stating that 'level design is applied game design'". Again, this statement refers to taking the game components and rules created by the game designers and developing a construct that allows the player to experience them. Meanwhile, Totten (2014, p. xxiv) defines level design as "...the thoughtful execution of *gameplay* into *gamespace* for players to dwell in". This further emphasises the need to facilitate the game mechanics in the development of a game level.

Whilst the definition of level design is generally agreed upon, the definition of a game level is not quite so clear due to the large variations of game genres and styles. However, a game level often takes the form of a physical space in which the game character must navigate using their actions to complete challenges to meet a set goal. It is not uncommon for game levels to be described as virtual worlds or virtual environments. Whilst they do share some similarities, they are not the same. Virtual worlds and virtual environments can exist purely as a piece of art in passive mediums such as film, TV, images, whereas the previous definitions make the point that level design is tied into game design and therefore game levels must also include gameplay within their environment. Whilst not all game levels take place in 3D environments (many games are 2D for example), most AAA games (games with large budgets) and many independent games do feature 3D game spaces. Therefore, we can describe a game level for the most common of games as encompassing a 3D space or layout which a player must navigate using their actions to beat challenges set out by the level designer to reach a given goal.

#### **2.4.1 Level Design Areas of Focus**

There are several key areas that a level designer will focus on when they plan and construct a game level. They must first understand what abilities or actions are available to the player at that point or location in the game. It is common for games to drip feed character actions to players to ensure they are not overloaded at the start of the game and to keep them learning and therefore engaged throughout the game (Koster, 2014, p. 46). They must understand what game mechanics the player has already been taught and which mechanics must be taught in the level. In linear games in particular, the player should be taught mechanics before they can be expected to use and master them in open play (Totten, 2014, p. 173). They must also think about how the game level ties into the narrative and where they must advance the narrative (Adams, 2014, p. 360). This often defines the location of the level and will therefore impact the architecture and layout choices available to the level designer.



With the above points in mind, a level designer must then design the layout and geometry of the game space. They must think carefully about the scale and shape of the different areas of the level layout. For example, large open spaces across multiple heights encourage exploration by the player, whereas tight linear corridors leave little choice to the player but are useful to funnel the player from space to space (Totten, 2014, p. 200). The composition of a game space and architectural shapes is also important. Leading lines and eye-catching architecture can be used to draw the attention and gaze from players (Shaver & Yang, 2018, 16:00). Wayfinding and navigation for the player is also an important focus during design. It's important that players can be guided through the level so as not to get lost and frustrated, but not to be done in a way that takes away player agency. Many techniques are available to achieve this (lighting, landmarks, geometry structure, signs, etc.) and each must be considered by the level designer. The design of the layout, geometry and spatial design is therefore one of the core roles of the level designer (Peters, 2020a).

Although the level geometry and layout are important, game levels consist of more than just the geometry and architecture. They exist not in a vacuum but inside an interactive game and must support the core mechanics (actions and challenges) of the game. Therefore, the level designer must also integrate challenges into the game level, making use of the geometry to create environmental challenges as well as integrating the core challenges of a game (such as enemies, puzzles, etc.) into the geometry and level layout.

#### **2.4.2 Level Design Process and Workflow**

Although the level design process and workflow differ between different games, genres and level designers. It is common for level designers to follow a similar process as below:

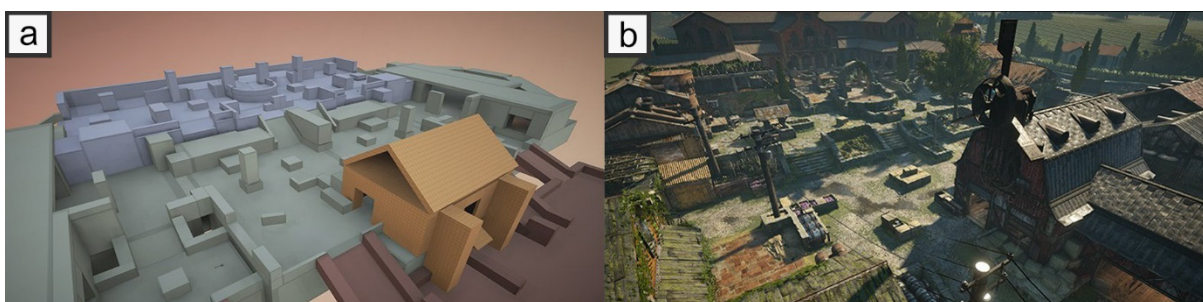
- **Planning** – Key details about the level such as general location, start and finish points, narrative development, mechanics previously taught and new mechanics available are planned out.
- **Sketch or digital 2D level layout** – A 2D plan is mocked up outlining the geometry, location of mechanics, primary route, etc.
- **Blockout (also known as whiteblocking, greyboxing, prototyping or similar)** – A simplistic version of the geometry of the game level is developed in a game engine/level editor.
- **Integrating and prototyping game play** – The challenges, enemies, puzzles and any other game play will need to be implemented in the blockout prototype.
- **Iterating and refining** – The level designer will play test the level many times themselves and with testers in the blockout form to ensure the level is engaging, fun, has clear wayfinding, etc.

- **Hand off to art team** – The art team (3D environment artists in particular) will develop detailed 3D assets once the blockout is unlikely to change. This is the final art the player will experience.

From the above, there are several key areas that most level designers will incorporate into their process. The creation of a blockout level layout, integration of mechanics and gameplay and then the subsequent iteration and refinement of these are universally discussed by level designers as part of their workflow (Grimsby, 2014, p. 198; Pruet, 2014, p. 236; Weldon, 2008, p. 115; Shaver & Yang, 2018; Peters, 2020b). Therefore, any tools that enable level design will need to support these as a priority.

Game levels must be built and tested for balance, difficulty progression, mechanics progression, sight lines and more beyond just their aesthetical appearance. The game level layout will go through many different iterations throughout development and this iteration is a key part of the level design process (Grimsby, 2014, p. 199). Therefore, the level is initially created in a blockout form (prototype) to enable level designers to plan out the level layout, test it with the game mechanics and then iterate many times as quickly and as often as possible (Shaver & Yang, 2018, 1:24; Lee, 2020). **Figure 17a** shows an early blockout version of a multiplayer level for Gears of War 4. The level will be tested in this format with players to ensure it is balanced, engaging and meets the requirements of the level as outlined by the game designers.

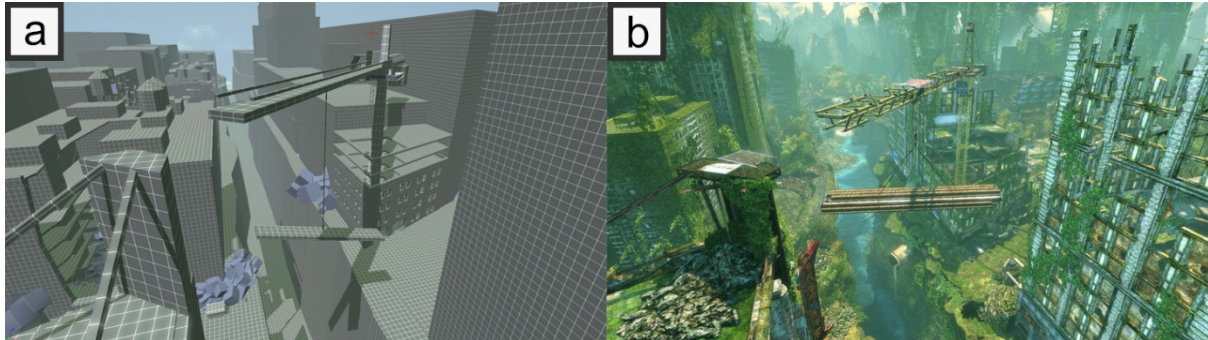
Not only is the geometry and level layout prototyped with the blockout form, but the game mechanics are also integrated into the level. The challenges laid out by the game designers must be integrated along with any scripted events, enemy spawn locations, item placements, or anything else related to the game. It is key that the level is tested with all mechanics present to ensure the level best facilitates the game mechanics and overall goal of the game design.



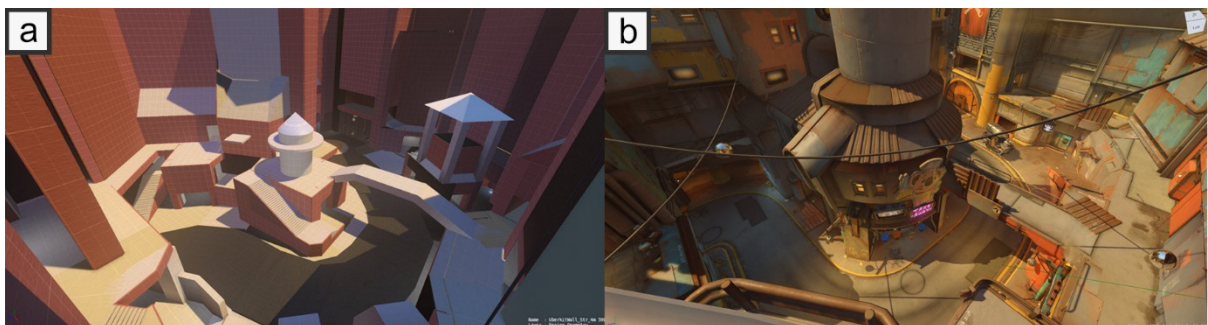
**Figure 17** – (a) Multiplayer level blockout for Gears of War 4. (b) The same finished game level (Gears of War, 2017).

Once the level designer is happy with the level layout and gameplay, they will pass the level over to the art team who will create detailed 3D assets and textures to populate the scene (**Figure 17b**) (Shaver & Yang, 2018, 2:40). The need for the simplistic blockout version is to

ensure the level designer can make quick changes to the simplistic shapes, rather than trying to change the detailed meshes that will have taken many months to create. This breakout form is integral to the level designer workflow and additional examples from different games and studios can be seen in **Figure 18** and **Figure 19**.



**Figure 18** – (a) Early blockout prototype of a single player level for *Enslaved – Odyssey to the West* (b) Final finished version of the same level (Field, 2017).



**Figure 19** – (a) Blockout prototype of a multiplayer level for *Overwatch*. (b) Final finished version of the same level (Adams, 2017).

### 2.4.3 Serious Games

The initial discussion of games thus far has focused on games where their primary purpose is for entertainment. However, there is a category of games that focus not initially on the entertainment of the player, but instead the primary focus is on teaching the player a topic or skill through play. It can be said that these games merge a non-entertaining purpose (*serious*) with a video game structure (*game*) (Djaouti et al, 2011). These games are known as serious games and they can teach, train, and educate (Michael & Chen, 2006). Serious games come in many different forms.

Some serious games teach the player new information implicitly through play. *Dragonbox Elements* (WeWantToKnow, 2014)) demonstrates this by teaching players' geometry through an interactive fantasy puzzle game. The game aims to teach the players without their explicit knowledge that they are learning geometry by masking it inside the puzzle game.

Other serious games teach the player in a more explicit manner and are more akin to digital training systems. *Wheelchair-VR* (John et al., 2018) was developed to train potential users

who were new to using a powered wheelchair with a VR serious game. The game taught users through several bespoke environments and with game like features (score, different levels to complete, etc.). Another example of a serious game that teaches new skills explicitly is vHealthCare (BreakAway Games, 2022). This serious game simulates role-play with virtual patients and team-members working in a healthcare environment. As with all serious games, the game focuses more on realism and the attainment and maintenance of skills over developing a fun and engaging video game.

Despite the above games being serious games, they still often require the development of game levels in which the game and teaching mechanics take place. Therefore, there is still a role for level design in their development.

#### **2.4.4 Serious Games Level Design**

The level design process and workflow for creating levels for serious games share a lot of similarities to entertainment games. Game levels must still be planned out based on the requirements and abilities of the serious game players at that point in the game. They must still design and plan out the layout of the level. They must also add in any scripted functionality before testing and iterating on the design.

However, there are some key differences. One difference is whilst many entertainment games take place in fantasy worlds that were created from the imagination of a level designer or concept artist with little contextual limitations, many serious games are used to train users in real world locations to improve real skills or knowledge. This means the main aim for the level designer is to recreate real world locations closely to better immerse serious games players in the role they are undertaking.

Whilst the levels could still be based on reality, but not a specific location, and created from a constrained imagination. It is felt that these can be improved further for serious games that make use of real-world locations. In these games, rigorous spacing and accurate depiction of distances and gaps play an important role in the simulators' efficiency and usability. In the example of the Wheelchair-VR (John et al., 2018) serious game, affording the user the ability to train in a location based on their home, place of work or regularly visited location could have many benefits to their training. These benefits could be (but not limited to):

- Familiarity of the location could help ease the player into the game, especially in the case of users with a disability (Politis et al., 2017).
- Players will be practicing manoeuvring in a location they will use the actual powered wheelchair giving them more experience for when they use the real powered wheelchair in those locations.

- Issues with the location may present themselves during play (such as doorways not being large enough or location of furniture not being suitable). These issues can then be improved before the players utilise the real powered wheelchair.

Other serious games that may also benefit from this workflow would be virtual evacuation training (Lovreglio et al., 2017) and virtual fire safety training (Chittaro & Ranon, 2009) systems, amongst others as they may receive similar benefits to above. The level of presence and realism also play a key role in the development of skills in serious games (Lovreglio et al., 2018; Chen & Chien, 2022).

This thesis explores different workflows that utilise both AR and VR to aid level designers in developing game levels accurately based on real world locations to enable the above potential improvements.

#### **2.4.5 Virtual Reality Games**

There are several examples of past commercial VR games and systems, such as the Virtual Boy (from Nintendo in 1995). These previous devices were often limited in features, were expensive, didn't work very well in the case of the Virtual Boy, and were mainly unsuccessful at breaking the mass market. With the release of the systems covered in section 2.3 along with a console friendly VR system from Sony (PlayStation VR), VR games have found some success in the market. Meta (previously Oculus) announced that 124 Quest apps have earned over \$1 million in revenue (Lang, 2021a), Beat Saber announced that they had sold over one million copies (Hayden, 2019a) and Sony announced that they had sold 4.2 million PlayStation VR systems (Moon, 2019). This shows the success of VR and how these systems have created a market for VR specific games.

When designing games for immersive VR systems there are several commonalities between non-immersive and immersive VR platforms. For example, both platforms share similar genre tropes (first-person shooters, sports games, action role-playing games, etc.), common player actions and challenges, the ability to support a narrative, to name a few.

However, there are some key differences. For example, in immersive VR the only camera is the one viewed from the players headset (Laurel, 2016). Non-immersive games can utilise multiple different camera views and styles, however, with immersive VR this can only be achieved by moving the player, which will not have the same desired effect.

Developers must also pay attention to how they present the user interface (UI) and heads-up display (HUD). Non-immersive games can present the UI/HUD in screenspace so it appears attached to the lens of the camera and moves wherever the camera moves. However, with VR this is not possible as this would cause binocular-occlusion conflict as the eyes struggle to

focus on an object with no clear depth (Jerald, 2016, p. 173). Therefore, the UI/HUD must be presented in world space such as floating in front of the player, as part of the world geometry or attached to an item in the hand. This ensures that the player's eyes can correctly focus on the UI/HUD without visual conflict.

As noted in section **2.3** the locomotion techniques for immersive VR games require special attention. Traditional 3D games played on a 2D screen allow users to manually move the player and camera to traverse a 3D world with joystick-directed steering. The same techniques used in immersive VR games result in increased levels of simulator sickness (Langbehn, et al., 2018). Therefore, locomotion techniques unique to VR should be used such as, real walking or room scale, redirected walking and point-and-teleport. However, if traditional joystick-directed steering must be used, then simulator sickness mitigations should be utilised, such as limiting the peripheral vision when experiencing linear and angular velocity changes (Buhler, 2018; Fernandes & Feiner, 2016).

Finally, most traditional 3D or 2D games viewed on 2D screens offer indirect actions during game play. For example, if the player wants to swing a sword, jump or open their inventory, they press an arbitrary button to activate these actions. Most immersive VR systems now offer 6-DoF motion tracked controllers. Motion tracked controllers offer developers the ability to support natural mapping of game play actions to real-world actions. For example, if the player wants to swing a sword, they can swing their hand as if they are holding a sword. Research suggests this natural mapping results in higher levels of presence and enjoyment (Skalski et al., 2011). Therefore, where suitable, natural mapping should be considered for player actions.

The initial software releases for recent VR systems were entertainment video games, but serious games have also been popular. This is due to the additional immersion gained from playing a serious game in VR. Academics have long been working on serious games with VR system, but with the release of commercial hardware, the VR serious games market has increased greatly. Wheelchair-VR is a good example of how immersion from VR can improve the ability for users to train for using a real powered wheelchair system (John et al., 2018). FLAIM demonstrates how VR can be used to create immersive training systems for training fire fighters in a safe environment (FLAIM Systems, 2022). Fire fighters can use FLAIM to learn new techniques and to maintain their current skills in an interactive serious game whilst in immersive VR.

With this increase in popularity of VR entertainment and serious games, the need for tools to enable the development of these systems has increased.

## 2.4.6 Virtual Reality Level Design

The current process for developing games for VR usually involves the use of a game engine, such as Unity 3D or Unreal Engine 4/5. These engines both support publishing of games compatible with VR headsets and include many features and APIs to ease development for these platforms from a programming perspective.

The process and workflow of level design for VR games is currently very similar to the processes previously mentioned in **2.4.2**. Whilst this has enabled developers to create quality VR games, there is potential for the workflow and processes to be improved. These workflows work well with existing platforms, such as PC games, console games and even mobile games due to both the development platform and the target platform sharing similar screenspace technologies. PC, console, and mobile are all 2D screenspace systems (the user views a 2D screen with no stereoscopic 3D) as are the systems used to develop these games. This means scales and perspectives are very similar when viewed on the development system in the editor and the final game result.

However, scale and perception in VR are different to 2D screenspace due to the stereoscopic 3D afforded by VR headsets (Silverman, 2017, p. 312), with scale being perceived more accurately in VR (Witsel, 2020). What might look like the correct scale for a room in a 2D screenspace editor (such as Unity), may not feel correct when viewed in VR in first person. Similarly, when working with 3<sup>rd</sup> person games in VR, the environment and characters might look to scale in 2D screenspace but again, this might not be the case when viewed in VR (Bettner, 2015). This mismatch of scale between different screenspaces means the level designer will need to continually check between the 2D editor and through the VR headset to ensure the scale is as expected. This takes time as the designer must stop iterating, put on the headset, pick up the controllers, start the game and then view the area in VR (Bolton, 2020). Therefore, this will impact the designer's ability to iterate quickly on designs which hampers one of the key parts of the level design workflow.

There are also techniques that are used to enhance 2D screenspace based games that don't function quite as well in VR. Many designers will use frameworks to aid with the composition of space. The rule of thirds splits an image or view into three equal horizontal and vertical regions which helps create a pleasing image or draw the eye to specific areas (Caplin, 2008, p. 34). Frameworks like rule of thirds work well in 2D screenspace due to the framing of the device by the bounds of the screen (Silverman, 2017, p. 321). This is not possible in VR due to the user being fully immersed in the game. Therefore, different composition techniques are required that are more like that of the architectural world. Again, these techniques are best

applied when they can be designed and viewed quickly within VR to see where they do or do not work.

Designers will often fake effects, such as the scale of distance objects to create a parallax effect, or to make something look larger than it really is (like mountains in the distance). Techniques like these don't work in the same way in VR as the player has a 3D stereoscopic view (Witsel, 2020). Therefore, these techniques and habits need to be picked up quickly to avoid wasted development time. Viewing these techniques in VR will again avoid designers making these mistakes.

There is also the issue with cutscenes and guidance. In many games, it is common to take control of the camera from the player to display an animated scene between characters, action taking place in an environment or guiding them through the environment. This is ill-advised in VR due to the potential for cyber sickness being increased when visual movement (such as moving and controlling the camera view) do not match what the body is sensing (particularly the ears). Therefore, designers must avoid taking control of the camera for cutscenes (Odeldahl, 2020). They could instead catch and keep the attention of the player to play out animated scenes or guide the player, whilst the user has full control over their camera view. This can be achieved with loud (bright and bold objects) and to have something that blinks (flashes) to catch their attention (Odeldahl, 2020). It is therefore important that designers can quickly and effectively test these scenes in VR to ensure the player's attention would be drawn to these areas and that they aren't distracted by other action in the scene, something that is much easier to do when testing in VR.

Another difference between players on existing platforms and VR is that they are more likely to look up in VR than on a PC, console, or mobile game (Silverman, 2017, p. 316). This is partly due to fixed camera angles on some 2D screenspace platforms such as 3<sup>rd</sup> person perspective that don't allow the player to look up, but even for first person games with full movement, players rarely look completely up. This results in some level designers not making the most of the ceiling in their designs or simply not taking the time to design them in detail. Players in VR are more likely to look around them in the space, including above them partly due to the novelty of using a fully tracked headset but also because it is human nature to explore the architecture of a space when entered. Again, working and designing in VR will help highlight areas of a level that has detail missing that might not be seen when building the game on a 2D screen.

Based on the above issues and difficulties with designing levels for VR games, one of the main focuses of this thesis is to explore improvements to the VR level design workflow through the design of levels from within a VR level editor. One aim is to validate any possible



improvements to the workflow, such as improved iteration time, better scale and perspective awareness and avoiding mistakes of using incompatible existing frameworks.

Designing levels in VR may not only improve the existing workflows and solve some of the previously mentioned issues but may also offer several possible benefits to level designers. More natural interactions through motion tracked controllers may allow level designers to place objects more naturally. Currently rotating assets using a mouse and keyboard (or 3D mouse with a rotation knob) on a 2D screen is unnatural. By giving the designers the ability to pick up objects and position like they would with their hands in the real world allows for quicker and more natural placement of objects. When working on a 2D screen it is often difficult to perceive the exact depth of objects in view (Conway et al., 2000). Improved depth perception from the stereoscopic 3D view may help with the with the placement of assets and location/shape/size of geometry in the space. These improvements and possible benefits are all explored and validated in this thesis.

## 3.0 Literature Review

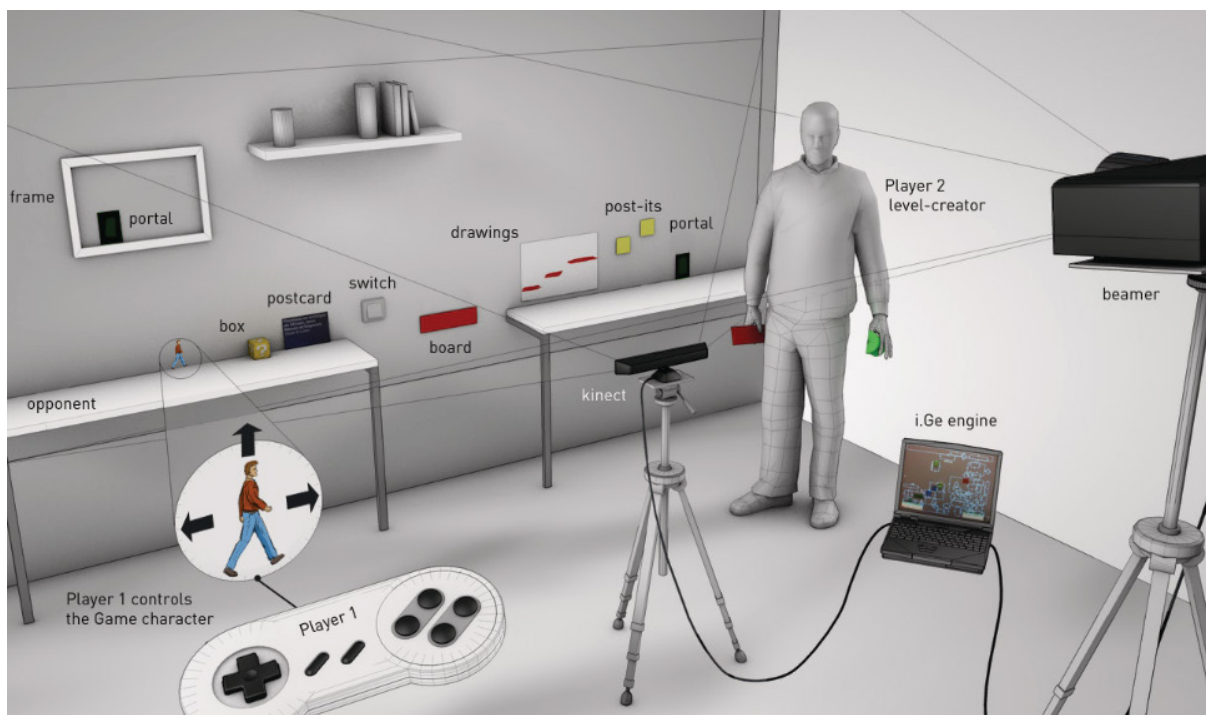
This chapter presents an analysis of similar work that relates to AR, VR and SR concepts and their use in level design workflows or adjacent topics. The chapter concludes by summarising the key outcomes and gaps in knowledge.

### 3.1 Augmented Reality Level Design Similar Work

Two routes were identified when starting to explore how AR technologies could be used to aid level design workflows. The first is using AR technologies, such as working in an AR view, to allow a level designer to design and construct game levels. The second is to look at how AR technologies or similar can be used to help capture a digital map of a real-world location. The map could then be used to construct a traditional or VR game level that is accurate to scale.

#### 3.1.1 Augmented Reality Level Editing

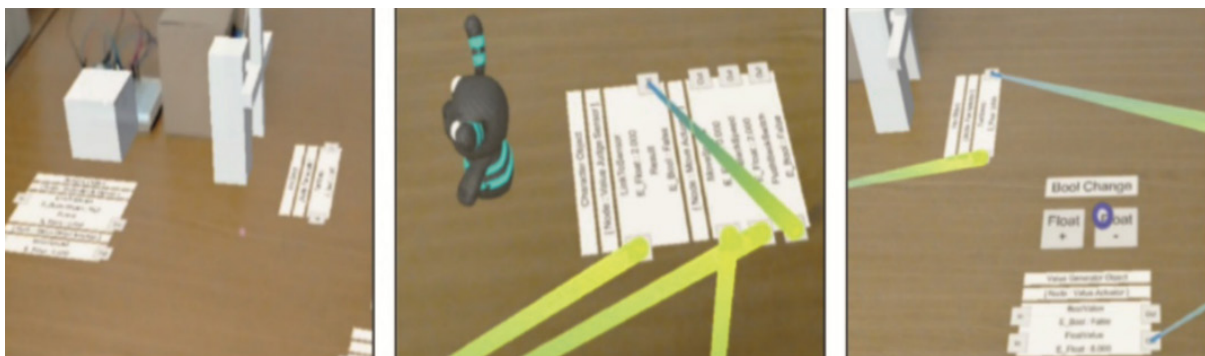
There are a number of works that have explored AR level design. i.Ge (Oswald et al., 2015) uses a projector and Microsoft Kinect device to allow for level design and editing using the real world as a level location (see **Figure 20**). The game is projected onto the real world and the game character can be moved with a game controller in a 2D platformer style game. The platforms and obstacles can be created digitally, but also from existing real-world objects, like signs on the wall, sticky notes or specific-coloured pieces of paper (which can trigger gameplay mechanics).



**Figure 20** – i.Ge system overview showing how the game is displayed, edited and played on a real-world wall (Oswald et al., 2015).

The game level can be constructed before play starts or edited as it is being played. This is an interesting use of the technologies but does limit the level design to that one specific location, rather than a more generalised level. The technologies are also quite bulky and not suitable for easily moving to various locations. This could potentially be solved with more recent smartphone AR technologies that were not available when this work was created, such as ARKit (Apple Inc., 2022a) and ARCore (Google Developers, 2022).

Work by Park et al. (2018) utilises the Microsoft HoloLens and Unity 3D, along with markers and paper interfaces to create AR interactions. The system displays connections between real-world and digital objects with visual node graphs and can utilise IoT (internet-of-things) devices to interact with digital objects, such as a fan (real) and a game character (digital) (**Figure 21**).



*Figure 21 - Data is displayed digitally along with lines demonstrating how data is transferred.*

Whilst not level design, *per se*, this shows how game mechanic and interactions could be programmed and displayed. The real-time connections between objects visualised by lines are a potentially useful way of showing the user how the data will be passed from object to object as it uses the spatial dimension for additional context.

Ng et al. (2018) utilise the HoloLens to allow for level creation using the real-world space and includes the placement of virtual objects and scripting (see **Figure 22**).



*Figure 22 - Images demonstrate virtual items added to a real-world location, as well as how interactions can be programmed (Ng et al., 2018).*

The system enables *in-situ* level design and creation whilst in a real-world location. Players can place virtual game objects in the world, as well as scripting objects like triggers and

buttons. As with the previous work, virtual lines are drawn between triggers and objects their acting upon to good effect.

These works all focus on creating content or game levels for AR experiences. Some lessons are transferable, but they focus mainly on creating levels for real world environments rather than for fully immersive digital environments. This could be explored further as a new workflow, but there are other ways AR could be used to help improve existing level design workflows, such as capturing a digital map of a real-world location for use in a serious game.

### **3.1.2 Location Capture and Indoor Modelling**

There are several different techniques that enable capturing or creating a digital map of a real-world location. Systems built around laser scanning and photogrammetry have existed for many years. These systems have been used for autonomous robot navigation (Hoppen et al., 1990) as well as building information modelling (BIM) which is used to aid with construction projects and renovations (Arayici, 2007). This early work helped contribute to the development of industrial laser scanning systems, which are now commonly used in construction to create point cloud models of a site or indoor space (Volk et al., 2014). However, these point cloud models are not suitable for use in serious games due to their complexity, lack of polygonal mesh data and inclusion of every object in that space present at the time of the scan.

Whilst the raw data from laser scanning systems would be difficult to work with for a serious game, various academic works explored generating simpler and cleaner models from the laser scanned point cloud data. Turner and Zakhor (2014), developed a system that initially generates 2D floor plans from complex point cloud data and then extrudes a simplified 3D model from the floor plan. Monszpart et al. (2015), presents RAPter, which uses point cloud data to rebuild a scene using a regular arrangement of planes. Their system analyses the point cloud data locally and generates planes where necessary with completion times ranging from seconds to hours, depending on complexity of the scene. Ochmann et al. (2016), analyses the point cloud data to create volumes that represent rooms and can detect wall depth by analysing nearby volumes. This creates a model that shows relationships between walls to produce room segments. These systems are mostly effective at developing indoor models that are more suitable for serious game virtual environments than raw 3D point cloud data. However, they still require the initial collection of 3D point cloud data from laser scanners, which are costly; and these systems focus only on recreating walls and not furniture or objects within the space.

Another technique for modelling indoor spaces in real-time is through the use of infrared (IR) depth sensors. Kintinuous utilises the Microsoft Kinect IR depth sensor to allow for real-time unbounded modelling of spaces (Newcombe et al., 2011). A complex mesh is created that

encompasses walls and objects in the space. Kalantari and Nechifor (2016), developed a custom application that utilises Occipital's Structure sensor attached to an iPad to model indoor spaces in real-time by scanning the area. Meanwhile, Angladon et al. (2018) developed a system that utilises a Google Project Tango device (a tablet with a built-in RGB-D camera) that can scan a room and allows users to differentiate between walls and clutter to produce a 2D floor plan. ComNSense (Abdelaal et al., 2018) utilises 3D point cloud data from RGB-D enabled devices and compresses the point cloud data locally using indoor grammars before uploading to a server. The uploaded data is then extracted into a 2D floor plan. All four systems demonstrate some issues though. Kintinuous (Newcombe et al., 2011) produces complex and noisy mesh indoor models that would not be suitable for use in serious game virtual environments. The Structure Sensor application often produced models that suffered from walls collapsing inwards when multiple walls were mapped (Kalantari & Nechifor, 2016). Whilst the work from Angladon et al. (2018) and Abdelaal et al. (2018) only produce 2D floor plans that would require additional steps to produce a virtual environment. All four systems require additional or specific hardware to function that can be costly and complex for users to learn.

There are also several works that utilize computer vision with monocular camera systems or static images. This solves the problem of requiring complex and costly additional equipment which is a frequent problem with many of the previously discussed academic works. Rent3D (Liu et al., 2015) reconstructs rental apartments from a set of monocular images and a 2D floor plan. The 2D floor plan is extruded into 3D and the system analyses the images to determine which parts of the space they encompass and add the images as textures to the correct walls. The system requires a 2D floor plan to begin with, which limits its effectiveness. Hedau et al. (2010), developed a system that analyses static images to detect the estimated size and shapes of walls along with the extents of large furniture in the room. The system is offline, and results are currently not stitched into a full scene. Wall and object scales are estimated and display some errors. LayoutNet (Zou et al., 2018) can reconstruct a room layout in 3D based on a single RGB panoramic image using a convolutional neural network (CNN). The system is effective for standard shaped rooms but struggles with irregular rooms and only maps the walls. These systems all show promise but as noted, they all suffer from different problems that mean they are not currently ideal for serious games level design.

Prior to the development and completion of the LevelEd AR workflow and study, ARKit was identified as a potential technology that could be used to help create digital maps of real-world locations using consumer devices. However, due to ARKit being in its infancy, there were no academic works demonstrating it being used as part of an indoor modelling system. Similar commercial systems did exist though, such as Magicplan (Sensopia, 2022), but these mainly focused on the floor plan and not size and placement of objects within the space. There are,

however, several academic works utilising ARKit to develop systems in other domains. For example, Fusco and Coughlan (2018) utilised ARKit to develop an indoor localisation system for users with visual impairments. The system was developed to help visually impaired users navigate a location with the system using visual-inertial odometry (VIO) via ARKit paired with sign recognition (based on prior sign locations on a 2D floorplan) to estimate their location. Their later work improved this system with full turn-by-turn directions (Fusco et al., 2020). Dilek and Erol (2018) produced an educational system for generating position-time graphs in real-time using ARKit as a foundation. They attached an iPhone 7 to a cart that could be moved along a single axis that allowed students to experiment in producing position-time graphs. Both works demonstrate the effectiveness for ARKit to track movement of a device in a space. However, they also report issues with tracking inaccuracies (also known as drift), in spaces that have limited detail on surfaces, where the device was moved too quickly or in some cases too slowly straight after initialisation. In all these cases, the studies were produced before LiDAR was available on iPhone devices or part of ARKit and so demonstrate the ability of VIO without additional tracking sensors.

## **3.2 Virtual Reality Level Design Similar Work**

The initial area of focus for using VR as part of the level design process was exploring how much of the level design process could be completed whilst in VR and what the potential benefits may be. As the literature search below shows, there are several different papers in this area, but none that focuses purely on level design, which includes level layout, game play editing and testing. Several similar works are covered in this following section.

### **3.2.1 World Building in VR**

There are several works that look at world building in VR for games or adjacent disciplines. One of the more relevant examples of VR world building is *Genesys* (De Leon et al., 2016). This system allows users to construct a virtual scene in VR. The system utilised an Oculus Rift for visual output and head tracking along with a Leap Motion device for hand tracking input and was built in Unreal Engine 4 (UE4). Users can select from a range of built in models or download additional models from a server that can be placed into the world to build up a virtual scene (see **Figure 23**). Users can manipulate the models by moving, rotating and scaling them to suit their needs. Once the scene is complete, users can upload their scenes to a central database for viewing by other users.



*Figure 23 - Selecting an object to spawn through the UI of Genesys (De Leon, Tavas, Aranzanso & Atienza, 2016).*

A simple scene composition study was carried out to test the system in a controlled test rather than testing the full system with participants. Participants were tasked with completing tasks based on block play for children, focusing on Picking, Lining Up, Bridging, Enclosures and Construction (Morrison, 2012 as cited in De Leon et al., 2016). Participants completed these tasks first using actual physical blocks and then within the immersive world building system. This was repeated three times with each participant. The results showed that whilst working with actual blocks was much quicker, participants did improve their times over the three tests with the immersive world builder. As participants were able to complete the tasks and improve over time, it does suggest an immersive system can be used to create virtual worlds in VR, but it is not conclusive. The study is quite limited and focuses more on evaluating the input mechanism (comparing Leap Motion hand tracking to real world hand interactions) over testing the more complex world building system. A study that focuses more on either evaluating their full system or comparing their system to other VR world building systems or desktop-based systems would have given more insight to the effectiveness of their system.

The aim for *Genesys* was to create a system that was simple to use for non-games developers to create virtual scenes that could eventually be used for animation and storytelling. This has been somewhat achieved, as users can create simple scenes with the built-in assets. However, the input mechanism (hand tracking) selected has not been proven to be the most suitable system, as the work discusses several issues involving user's inability to successfully rotate assets. The team had to remove the rotation mechanic from the system during the user study due to issues observed. Hand tracking has the potential to deliver more natural



interactions when manipulating objects (Bowman & Hodges, 1997), but in this case and with many others, if they are not implemented effectively, they can be a hindrance. The system offers a limited set of features for world building and focuses more on allowing users to place premade assets to create a visual scene rather than a game level. Therefore, the *Genesys* system does not support or validate a full level design workflow.

Slightly earlier work on immersive world building resulted in the creation of *The wonderland builder...* (Barot et al., 2013). Similarly, this system focused primarily on giving users the ability to create virtual scenes or worlds whilst in VR. The system was developed prior to the Oculus Rift but still made use of a motion tracked Nintendo Wiimote for pointing and manipulating objects, a Sony HMZ-T1 stereoscopic headset which was motion tracked for visual output and head tracking as well as a microphone for voice commands. Barot et al. (2013) targeted the system at younger users by wrapping the experience in a fantasy theme, including the menu systems and the accessories (hat and wand) to create dream-like state for users. Users can place models in the scene from a selection of built in models. There is no mention of how users move around the scene which suggests they are static. However, they have implemented the ability for users to scale up and down in size against the environment to get a bird's eye view of the scene or to get closer (see **Figure 24**). This is an important feature to consider in a system as whilst it is useful to get a bird's eye view, the user can no longer easily gauge scales of the scene against themselves.



**Figure 24** - (a) Bubbles display the available vocal commands for the current context. (b) In the larger scale, the user can have a global view of the environment. (c) In the smaller scale, the user can act on a microscopic level (Barot et al., 2013)

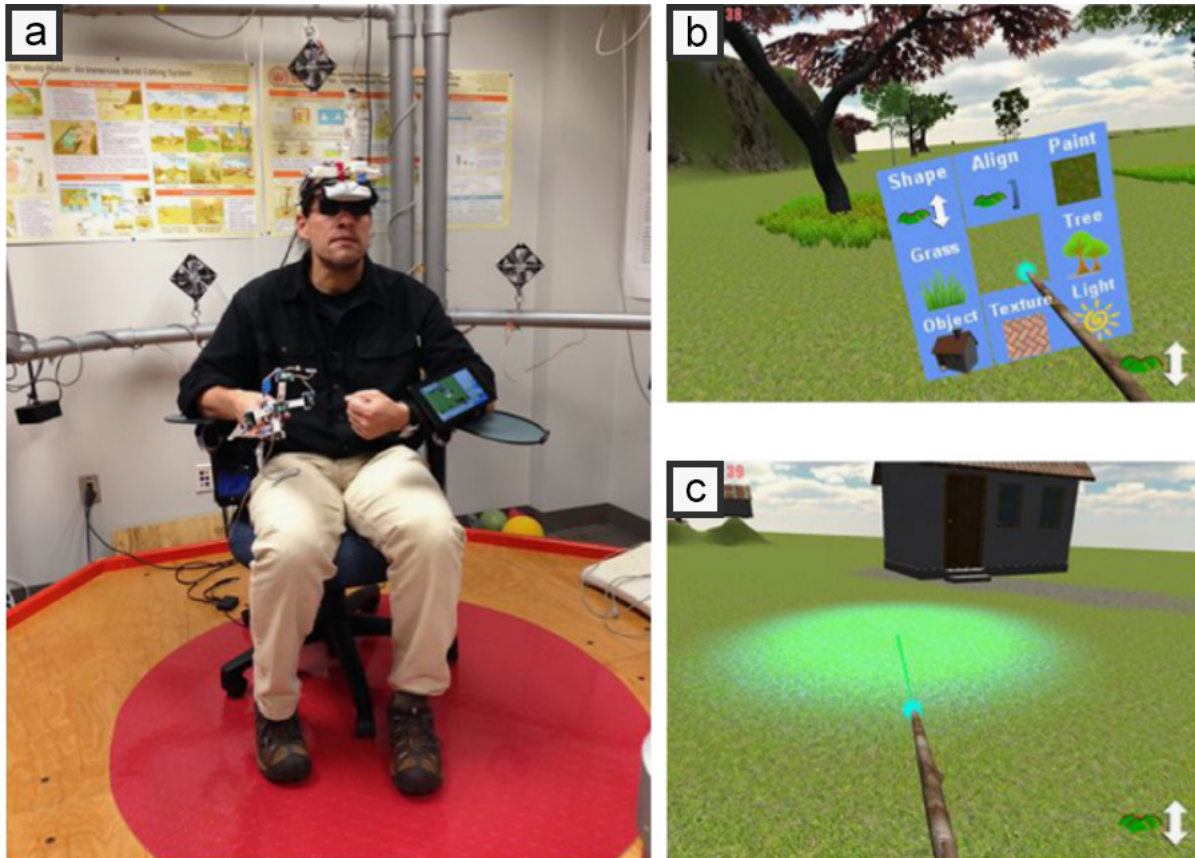
The work not only contributes a basic world builder, but also a hybrid input method by using both a motion tracked controller for placing and manipulating objects but also voice commands for selecting objects to create and selecting manipulation tools. The use of voice commands could be an effective way for allowing users to quickly change tools in a complex system as they do not need to navigate through a menu, but they do offer their own issues. Voice commands often suffer from limited recognition capability, commands are not always obvious (may need prompting), they can be difficult to select from a continuous scale, background noise can cause issues as well as the variability of speakers (accent) and they can be a distraction to other individuals (McMahan et al., 2014). Giving the user an option to use voice



and a menu system could be one solution to avoiding users not being able to use the system due to the above issues.

The above work also presents a guided tutorial system that steps the users through moving around the scene, then placing models and finally manipulating the models. The system clearly takes queues from game tutorials and more modern onboarding processes of applications and services to ease the user into the system. The evaluation of *The wonderland builder* focuses on the impact of the guided tutorial on users of the system, rather than on the system itself. The results showed that users completed more object manipulations (considered a more advanced interaction than merely placing models) with the guided system than without. An understandable outcome as the final part of the guided system focused the user on manipulating models. It is clear from this and as evidenced by games that an onboarding process is beneficial to any complex system rather than leaving users to figure it out on their own. Whilst evaluation shows the system to be usable and the system shows promise, it again focuses purely on world building and not the key level design workflow steps outlined earlier and therefore it does not validate a full level design workflow.

Wang et al. (2013) and Wang and Lindeman (2015) produced a more sophisticated world building system called *DIY World Builder*. This system afforded users the ability to place models in a scene and manipulate them as with previously discussed world builders. However, this system also allowed users to complete basic terrain creation, editing (height maps) and texture painting (see **Figure 25**). This is a more advanced form of world building and would allow for some game level blockout ability for a level designer, depending on the type of level being created (outdoors rather than indoors). The system used a novel input/output system in the form of a hybrid HMD + tracked wand and an interactive tablet system. The HMD used was an eMagin Z800 which does not include a fully immersive blocked out view and therefore supports the ability for the user to interact with a tablet whilst using the HMD. The wand and HMD are tracked via an external motion tracking system (see **Figure 25**). Users could complete tasks using the HMD and motion tracked controller and the interactive tablet with the user being in an egocentric mode for the HMD and motion tracked controller and an exocentric mode for the tablet (Jerald, 2016, p. 291).



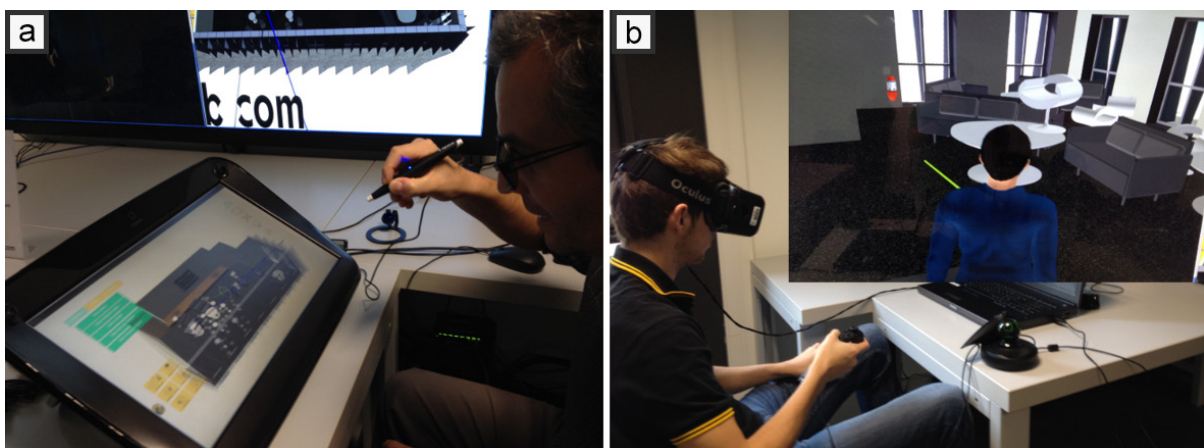
**Figure 25** - (a) The hardware setup, (b) the floating menu and (c) terrain brush of the HVE world builder (Wang & Lindeman, 2015).

The above work presents an evaluation of the input system and looked at how users made use of each of the two input methods. The focus of the evaluation was also on whether the two systems should be automatically synchronised in regard to tasks such as selecting an object, moving to a new location and making changes to the scene. The results show that having both synchronised automatically benefitted the users more by allowing them to complete tasks quicker and score higher on the evaluation tasks (Wang & Lindeman, 2015). This is likely due to the synchronised version requiring less cognitive load to use, such as not needing to remember the location of the user in both the HMD vs tablet version, for example.

Whilst the system showed benefits for a hybrid input system, in its current form it would not be viable for current consumer devices. This is mostly due to current VR HMDs focusing on immersing the user fully in a system and therefore using any external tablet or screen is not possible without breaking in and out of that immersive experience. It could be achieved with more recent MR options on headsets (like Meta Quest's passthrough), but these views are typically low fidelity and not suitable for viewing a small external screen. To make the most of the immersive nature of current HMD systems it is likely more effective to keep the user inside the system as much as possible. The evaluation of the system focused more on the input, rather than the benefits of using an immersive world building system. Although the system

presents a more sophisticated world building system than previously discussed works, it also does not cater to level designers with a full level design workflow that includes game play editing and testing.

A system developed by Le Chenechal et al. (2015) demonstrates the ability for users to layout spaces in VR (Oculus Rift DK2 and Razer Hydra controllers) collaboratively with a 3D zSpace interactive system. The use case presented primarily focuses on real estate agents who could work with clients to demonstrate the placement of furniture in a space. The real estate agent would use the 3D zSpace version, whilst the client can view the space in an immersive fashion in VR. Both users could add furniture to the scene, manipulate the layout of the objects and make measurements. Whilst only the real estate agent could highlight objects to catch the attention of the user and to send help to the user (see **Figure 26**). It is not clear how users can communicate in the system, other than moving objects or highlighting them. The work makes use of raycasts and object movement through the movement of the Hydra controller or zSpace pen, a common method of interaction in many works.

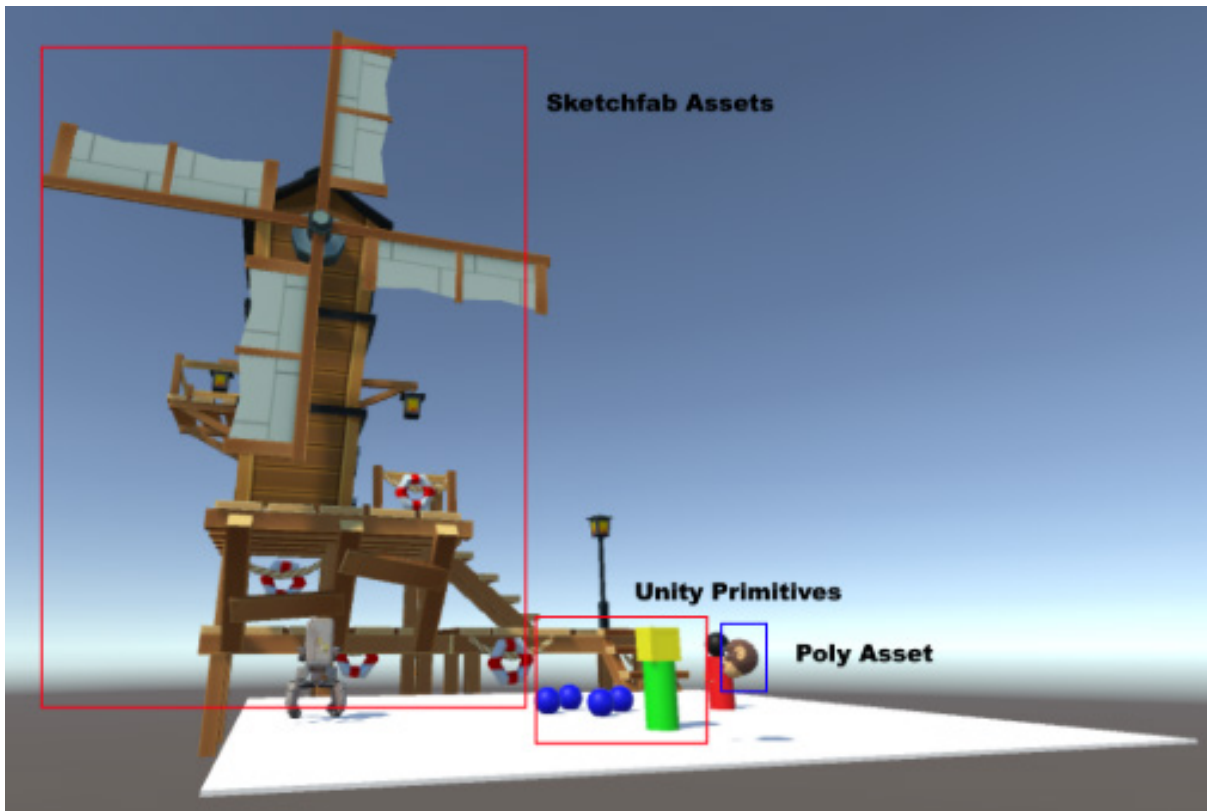


*Figure 26 - (a) The guide has a 3/4 top view of the scene and uses the zSpace interactive system. (b) The customer is immersed with a first-person view in the shared virtual world with an HMD and interacts with a Razer Hydra (Le Chenechal et al., 2015)*

The work demonstrates the possibility of asymmetric collaboration for world building systems, although it does not effectively show why an asymmetric version would be more effective than both users being in immersive VR, as both users have similar feature sets. The work is a good example of remote working that is enabled by the technology and system but does not offer an evaluation of the system or any form of study. As with the previous works, it only allows users to place models in a scene to fill up a space and does not cover sufficient steps of the level design workflow.

Work by Moreira (2018) contributes a world building tool that enables users to interact using voice commands to either place primitive Unity models in a scene or to find them on an online repository (Google Poly and Sketchfab) and place them into the scene (see **Figure 27**). The

system was built with modern commercial VR hardware, in the form of the Oculus Rift and Touch controllers (including 6-DoF tracking). The voice command system is much more robust than 'The wonderland builder' (Barot et al., 2013) as it allows users to use more natural speech. For example, users could say 'add 5 red cubes there' and it will create five red cubes and place them at the location of the user's pointer. Users could also carry out certain interactions using the Touch controller. The system makes use of WIT.ai for language processing and Microsoft Speech to Text for displaying spoken word as text in the application.



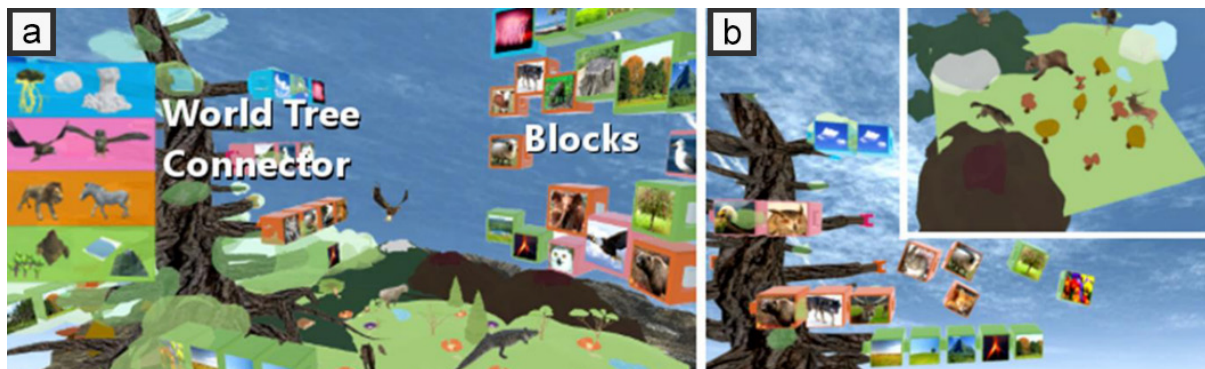
*Figure 27 - A scene with assets from multiple sources (Moreira, 2018).*

The above work also contributes an evaluation of the system by looking at participants' perception of the voice commands vs Touch controller interactions. Participants generally preferred the use of both as an option rather than relying solely on one interaction method. Participants seemed to find the voice commands useful, and this should be explored further as an option for input in an immersive level editor whilst considering previously mentioned drawbacks. Moreira (2018) also evaluated the system by having users' complete similar tasks in the 2D desktop-based Unity editor. The results here were inconclusive but did suggest that users found creating scenes in VR allowed them to focus better on the task and get a better idea of the result. However, the core focus of this work was on the input method rather than on developing and testing a competent VR system. The resulting system is again more of a world builder that allows users to add models to a scene and make basic changes to them. It



does not support all features required to support a level design workflow in VR as it does not support blocking out, scripting of gameplay and testing/iterating the level.

Similarly, Ichikawa et al. (2018) created VR Safari Park. This system aimed to create a very user-friendly world builder that would be much simpler than using traditional content creation tools. Users are presented with a tree structure which is both physical and abstract from a data structure perspective (see **Figure 28**). Users grab blocks that represent animals, the environment or weather and attach these to the tree on the branches. This allows users to build up layers of blocks on the tree that is then generated into a scene. The scene is built on rules, so certain animals will go towards certain environmental areas. However, the system does not allow for the manipulation of the scene, so the user is stuck essentially swapping out features rather than designing how the environment will look. Whilst the system is very user friendly, it does not afford users much freedom to create an original scene due to these restrictions.

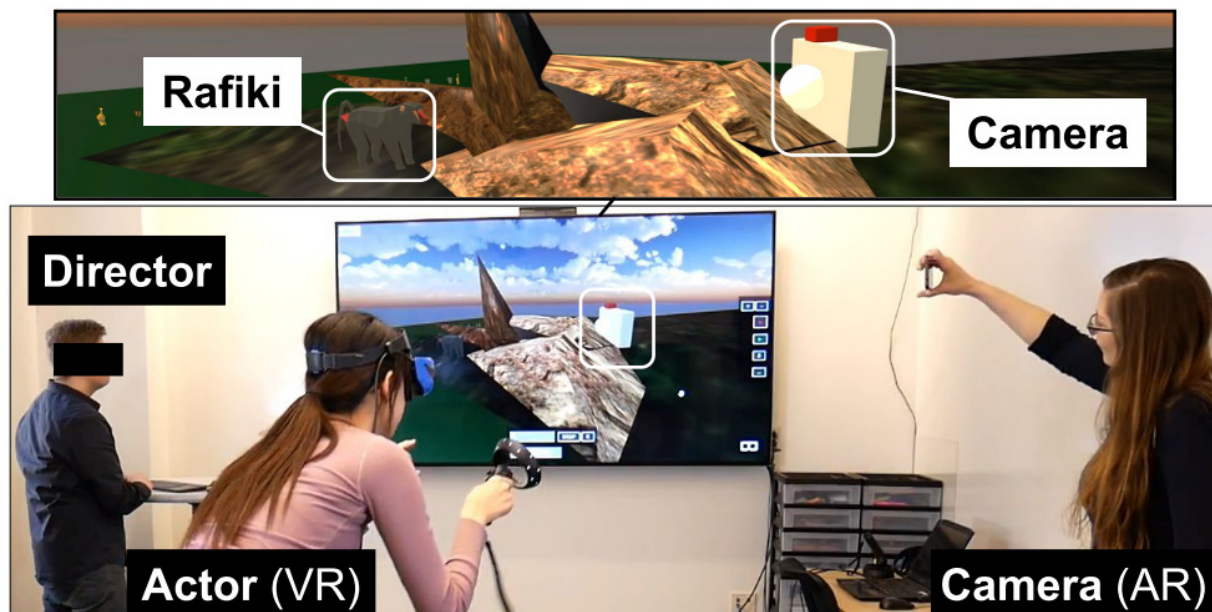


*Figure 28 - VR Safari Park. (a) World Tree Connector and Blocks, (b) Tree shape represents the world's structure (Ichikawa et al., 2018).*

The system received a positive usability score of 77.2 on the System Usability Scale (SUS) questionnaire (Brooke, 1996). Digging deeper participants found it easy to interact with the blocks and the world tree, with most quickly understanding that attaching blocks to the tree added objects to the scene. It is worth noting that these results, along with their previous work, suggests that physical and spatial input and menu systems work effectively in immersive environments (Ishikawa et al., 2016). Avoiding replicating complex Windows, Icons, Menus and Pointer (WIMP) interfaces in VR and building a system better suited to VR, making use of spatial depth perception would be advisable in an immersive level editor system. Whilst the work contributes a novel input and scene structure it is far too restrictive to be used for any real content creation. It therefore fails to meet the needs of a full level design workflow.

More recent work published since the creation of LevelEd VR demonstrates the ability to create an animated scene in a collaborative MR environment. To do this, Nebeling et al. (2020) developed a system that supports multiple different design roles, such as a director, actor and

camera operator. These can all be controlled by different devices, such as a traditional 2D desktop with 3D view, using immersive VR headsets and motion controllers and using a smartphone as an AR device. The result is all users can participate in developing an animated scene collaboratively in a Wizard of Oz style system where actors are moved like puppets.



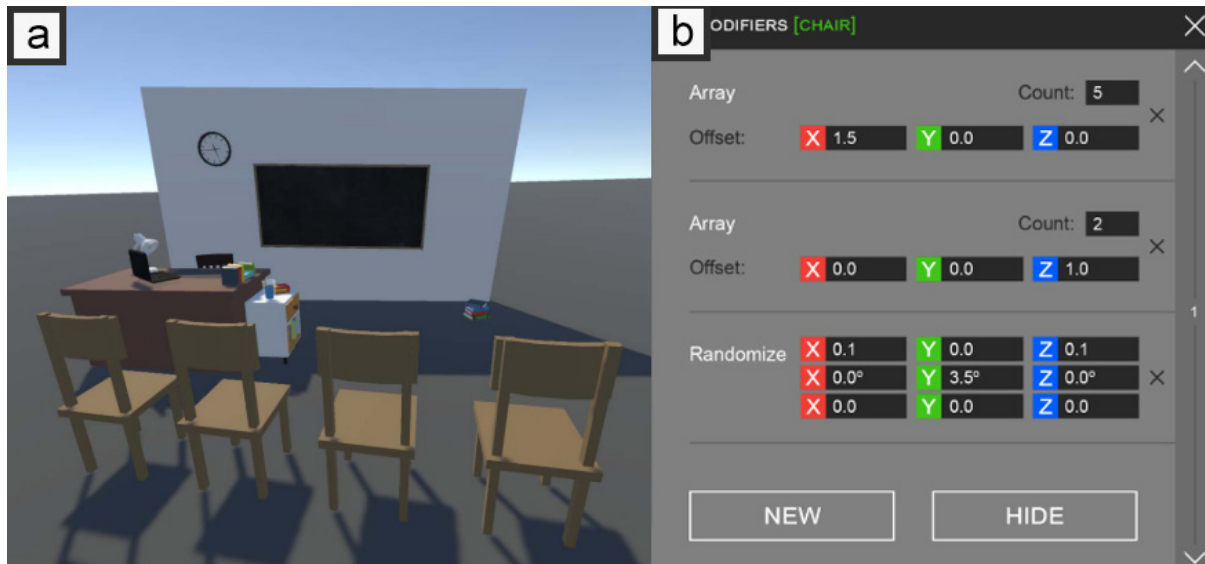
*Figure 29 - XRDirector allows users to take on a director, actor or camera role, each with their own method of input/output to collaborative create a prototype animation (Nebeling et al., 2020).*

The XRDirector system can be used to create many different types of animated prototypes, from animated scenes/movies to prototypes of games or applications. The end result is not an interactive product, but an animated demonstration of a game being played, application being used or a basic animated scene.

XRDirector was evaluated by four groups of three users who were asked to create prototype scenes of a movie and two games. They utilised different combinations of devices and roles during the tests, but the ultimate preferences from participants was the director using the 3D view on a 2D desktop, the actors using the immersive VR headset and controllers to puppeteer the characters and the camera using the smartphone with an AR view. There were a number of challenges reported during the tests. Such as keeping the different systems in sync, communication issues between the director and those in VR/AR as they could only describe what they wanted and not show them and the issue of the physical space (knocking into objects or other users). The end result is a useful insight into utilising different systems for prototyping animated sequences/scenes, but it does not allow for editing of the space or creation of interactive game play.

Another more recent work demonstrates VR Designer, a tool to create 3D environments utilising voice commands to import models from external repositories as well as additional

editing tools whilst in VR (Ferreira et al., 2021). Users start by adding either basic primitive objects or pre-made models to the environment. They can then move them using a motion controller or by using the modal 2D window presented as a floating panel. Users have additional tools, such as the ability to create arrays of objects or to randomly place them in the scene (see **Figure 30**).



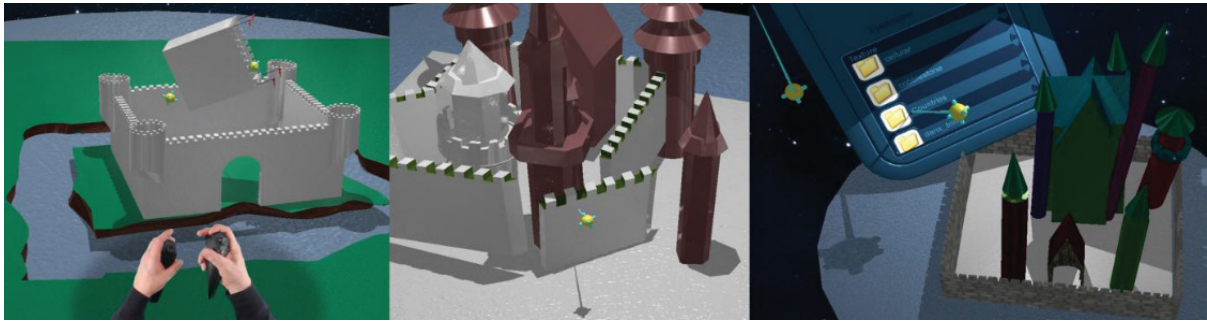
*Figure 30 - (a) A scene created with VR Designer utilising pre-made assets. (b) A 2D modal window used to apply modifiers displayed as a floating panel in VR.*

The system was evaluated by eight users of varying 3D and VR experience. Additionally, a control group was formed by three 3D modelling software users who completed the task using traditional 3D modelling software. Participants were grouped by their VR and/or 3D modelling experience. They were tasked with recreating three different scenes and their completion times were compared against the control. When all three scene completion times were taken as a whole, all participants using VR Designer were quicker than the control. This is likely because the system is simpler in design than general purpose 3D modelling software and so better suited to quickly placing assets in a scene. It is noted that the study had a small sample size, nevertheless, the work does suggest that voice commands could be a useful input method and that the inclusion of tools such as array or random scattering could be useful for creating or blocking out an environment more quickly. The ability to create more flexible blockouts is a limitation of the system though and the system does not demonstrate any ability to edit game play or test it.

Whilst all the above systems demonstrate some form of world building (a large part of the level design process), they rarely allowed for full flexible blockout/layout modelling, which limits their usefulness for level design, and none looked at creating and testing gameplay in VR.

### 3.2.2 3D Modelling in VR

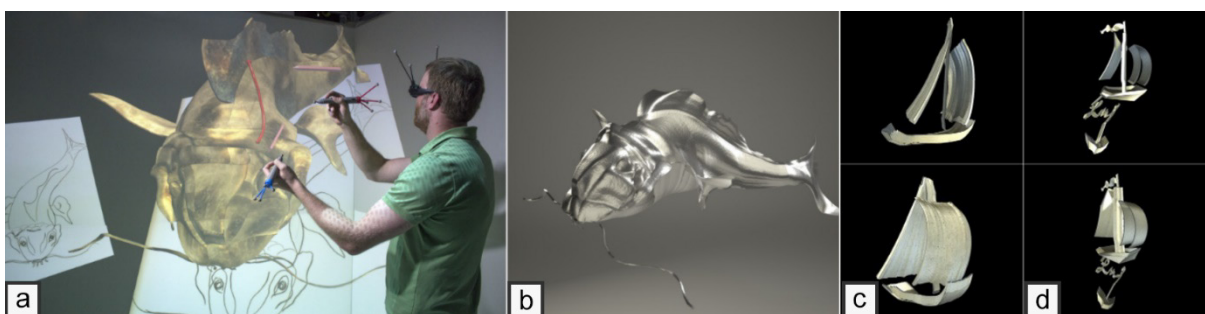
Whilst the previous works focused on creating virtual worlds and environments, a number of other works have focused more on the 3D modelling and creation aspect in VR. For example, MakeVR allows users to create new 3D models with similar functionality to those found in commercial 3D modelling applications (Jerald et al., 2013). Users can create more complex objects by combining different shapes, but they also have the ability to stretch and cut objects to suit their needs and to also assign different textures (see **Figure 31**).



*Figure 31 - A selection of castles created using MakeVR (Jerald et al., 2013).*

The study showed the potential for this intuitive input system but chose to test the system with the four participants using a 2D display, rather than a fully immersive VR headset (although it did support it) and participants also received extensive training of up to four hours before using the system. However, the results were positive and show that motion controls could be an intuitive input system.

Lift-Off utilises a CAVE system to allow users to add a 2D sketch or reference image into the scene and use bi-manual interactions to create 3D models based on the sketch (Jackson & Keefe, 2016). Users can select specific curves from the sketch, which are identified from the underlying pixel data of the image. These can be pulled away from the image into 3D space. Additional curves can be added in the 3D space and then used to create surfaces along the wireframe. The result is a 3D model created based on the original sketch.

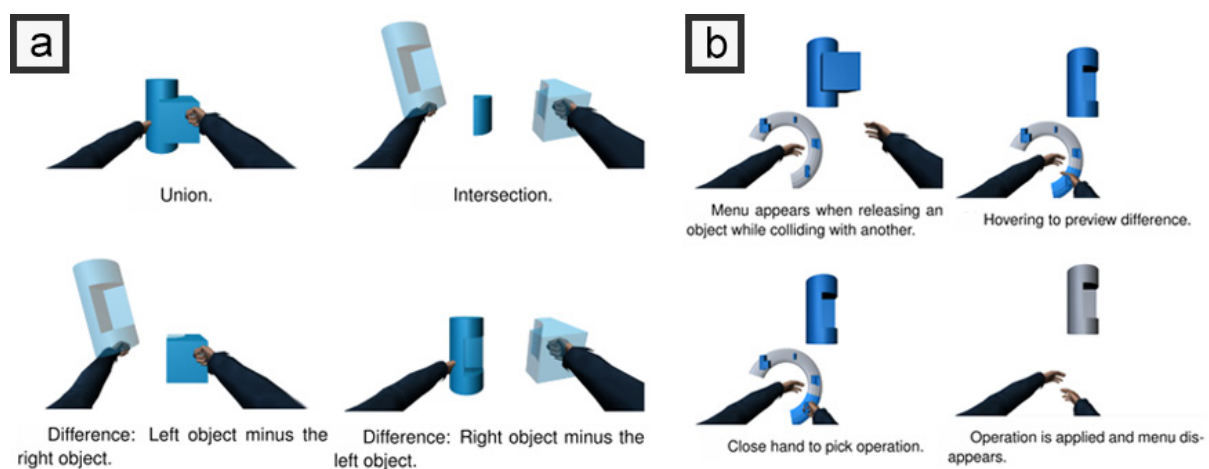


*Figure 32 - (a) User extracting curves from the sketch to build a 3D model. (b) The final 3D model rendered in Blender. (c) and (d) participant created boat models (Jackson & Keefe, 2016)*



They found some success with this approach, with the technique of lifting curves from a previous sketch resulting in an interesting modelling paradigm. However, time to complete simple models (such as a boat) took between 21-54 minute to complete, which is time consuming considering the final output and might not be suitable for quick prototyping of game levels (see **Figure 32**). The system was also built around utilising a CAVE, so it is not clear if the same lessons can be applied to immersive VR headsets.

Mendes et al. (2017) looked at supporting Boolean operations (cutting meshes where they intersect) whilst in VR using hand and arm tracking. The arm tracking was done using a Microsoft Kinect camera for position and Myo armbands for detecting gestures. A menu based approached was also implemented as an alternative to the gestures (see **Figure 33**).

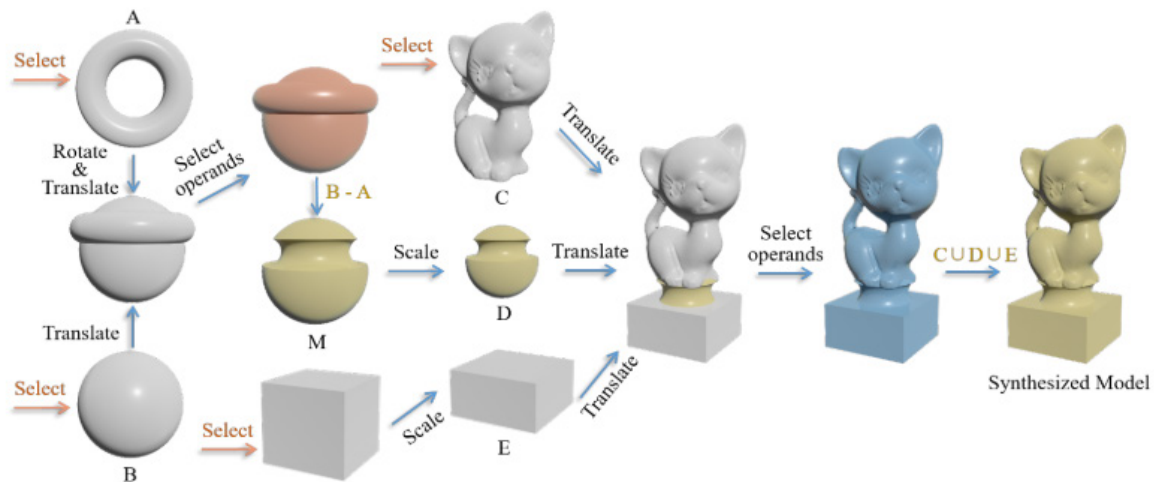


**Figure 33** - Mid-air Boolean operations carried out either with (a) gestures detected by a Myo armband or (b) a menu system (Mendes et al., 2017)

A study was carried out that asked 24 participants to recreate a model that required the use of all three Boolean operations. This was compared to a baseline technique that utilised physical buttons on a controller. They tested this with both the gesture and menu approach, as well as a full-bodied avatar representation versus simple hand representation. They found that the Boolean operations worked well, but that participants struggled with the input method when using gestures. This was primarily down to issues with gesture recognition from the Myo bands, but also because participants found it easier to recall and use the menu options, compared to remembering gestures. Interestingly, completion time was negatively influenced by the full-bodied avatar representation. This is potentially due to the avatar arms occluding the workspace, which suggests a simpler representation might be more suitable.

More recent work by Fu et al. (2022) presents EasyVRModeling, which allows users to create 3D models using a library of pre-existing shapes and combine them together using several different operations. The system uses signed distance function (SDF) that runs on the GPU and supports constructive solid geometry (CSG) operations such as union, intersect and

subtract to create a single model. The system runs on an HTC Vive, using the motion controllers to select, manipulate and interact with the components.



*Figure 34 - EasyVRModeling demonstrates the use of boolean operations and SDF rendering to create 3D models in VR (Fu et al., 2022).*

A user study was carried out with 34 participants and results were positive for the system. However, there was a potential for lag if users tried to carry out too many operations at once, and so the system would benefit from a limit being placed on the length of chained operations. The system shows that users found it useful to be able to carry out Boolean operations like subtract, but offering operations like this would likely need a level design system to support non-traditional mesh creation (such as SDF) or a CSG-solver, which could be complex to implement.

These works demonstrate potential for more detailed blockout/layout creation in VR, but there is more of a focus on creating single more detailed models, rather than larger spaces and they also don't demonstrate creating or testing gameplay in VR.

Finally, commercial developers of game engines such as the Unreal Engine and Unity have been building VR editors for their engines for some years now. However, Unity Editor XR lacks robust geometry creation/editing tools and the ability to script game play functionality and whilst the Unreal Engine VR mode is more robust it also lacks a runtime component. Unreal Engine VR is also built as an extension of the 2D editor, which requires prior knowledge of the engine and features many 2D modal windows rather than interfaces developed specifically for VR. Ekströmer et al. (2019) utilised Unreal Engine's VR mode to enable design ideation of lights in truck cabins. They showed potential for these tools for design but focused only on manipulation of the lights, rather than exploring the tools for more complex level design processes. Existing VR games with level editors are limited in number but do exist. For example, Rec Room; a social experience based around mini games (Rec Room Inc. 2016),

Ratchet: Nx; a racket and ball arcade game (One Hamsa, 2018), and more recently Horizon Worlds; a social meetup application (Facebook, 2021). Those that do exist, such as these examples, all have different levels of depth to their feature sets, and all implement features in different ways. Therefore, better understanding the overall benefits and how well specific feature implementations work and if they are accepted by users would benefit the development community.

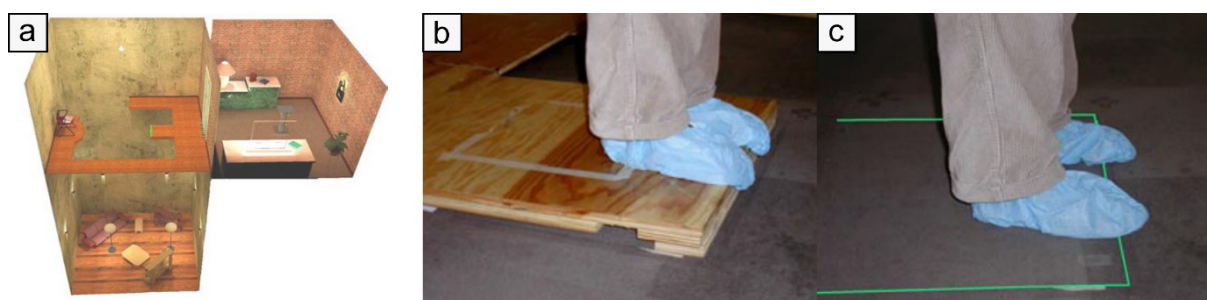
Overall, it is clear from this review of the literature that there is a lack of research focusing on exploring a full level design workflow and editing in VR, which encompasses gameplay and testing rather than just world building or 3D modelling.

### 3.3 Substitutional Reality Level Creation Similar Work

There are several areas of research related to SR. This section reviews similar work around passive haptics, SR experiences and benefits, non-SR creation tools, automated or procedural SR creation and manual/human-in-the-loop SR creation.

#### 3.3.1 Passive Haptics

Previous work into passive haptics has demonstrated benefits to VR experiences. Early work from Insko (2001) and Meehan et al. (2002) showed that including low-fidelity passive haptics in a virtual experience resulted in participants experiencing increased presence and demonstrating better spatial knowledge transfer. Two studies were completed to test the passive haptics. The first asked participants to stand on a virtual ledge looking down a pit, whilst in the real world they were stood on planks of wood with only a small drop (passive haptics) for the first test and then without the wood for the second, with the tests carried out on separate days (see **Figure 35**).



**Figure 35** - (a) The pit environment that participants navigate during the test. (b) Passive haptics used during the study. (c) No passive haptics used during the study (Meehan et al., 2002).

Participants' heart rates were monitored throughout the test and results showed that when using passive haptics, participants heart rates were increased by a statistically significant amount. The second study placed participants in a maze-like environment that they had to explore, and the study looked at cognitive mapping and training effectiveness. Participants

completed a between-subjects study that saw them trained either with passive haptics or without. Participants were then asked to map out the virtual environment that they had just trained in on paper, before then going to a physical location (with the same layout) and were asked to navigate it blindfolded as fast and safely as possible. Time to complete the task was shown to be statistically significant in the favour of those who had trained in the virtual space with passive haptics. This suggests participants gained a better spatial awareness and mapping when using passive haptics. Both works are considered seminal works that have influenced much of the research into passive haptics and VR.

Other works have looked at passive haptics through small physical props, rather than larger environmental spaces. In 1998, Hoffman reported an increase in realism in virtual environments when users interacted with virtual props that had a corresponding real-world object registered to them (for example a plate). Garcia-Palacios et al. (2002) also showed the benefits of VR serious applications and passive haptics (using a fake spider model) by demonstrating improvement in patients with spider phobias. A more recent study by Joyce and Robinson (2017) showed an increase in throughput for interactions with a system when passive haptics was used. These works all show improvements gained via passive haptics, but all focus primarily on serious games or applications.

### **3.3.2 Substitutional Reality Experiences and Benefits**

The following similar works are all focused on the term SR. This is a recent active research area and term used to describe experiences that integrate real world objects into a fully virtual environment with varying levels of accuracy. The term was used by Simeone (2015) to group together research around virtual environments and passive haptics. Following on from that initial paper, Simeone et al. (2015) looked at what changes can be made to digital proxies compared to the physical object and how this impacts their ease of use and suspension of disbelief. They found that objects with a different shape or temperature detracted most from their suspension of disbelief. However, they focused mainly on small props manipulated by fingers and hands, rather than larger objects, such as furniture that may have interactions with other parts of a user's body. The change in shape could be less noticeable when users are not directly interacting with an object with their hands, a concept that is worth investigating further.

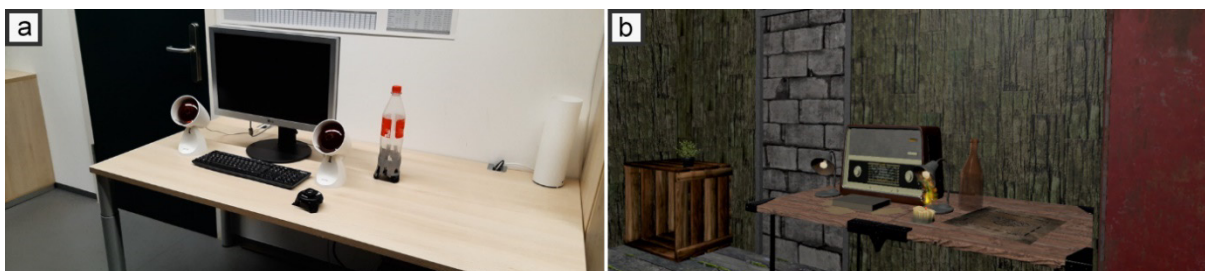
Eckstein et al. (2018; 2019) took SR further by adding a smart component (using Internet-of-Things devices) to add additional feedback to their users, such as sensors (temperature, brightness, and air quality) and remote-controlled devices (fan and lights) (see **Figure 36**).



**Figure 36** - (a) the real-world location. (b) A digital recreation of the real-world. (c) A Sci-Fi version of the space with maximum mismatch. (d) Displays both environments (top) and mixed environments (bottom) (Eckstein et al., 2018)

They found an increased level of presence through the addition of both SR and smart devices. They found no significant difference between the levels of mismatch (how accurately a digital proxy matches a physical object) and spatial presence. However, there was a significant difference with an increase in enjoyment of those in the Sci-Fi setting, despite this setting detracting most from the real world. This bodes well for enabling SR in situations where matching physical objects accurately would be difficult. Both examples focus more on serious games and applications than on entertainment games.

Steinhausser and Lugrin (2020) developed a smart substitutional reality (SSR) horror game (see **Figure 37**) built on the previous framework developed by Eckstein et al. (2019). Whilst there was significant difference in presence between a standard desktop experience with both VR and SSR, there was no significant difference between VR and SSR.



**Figure 37** - (a) The real-world location, including some objects that use Vive trackers to track their position. (b) SR version built for the horror game and mapped to the original space (Steinhausser and Lugrin, 2020).

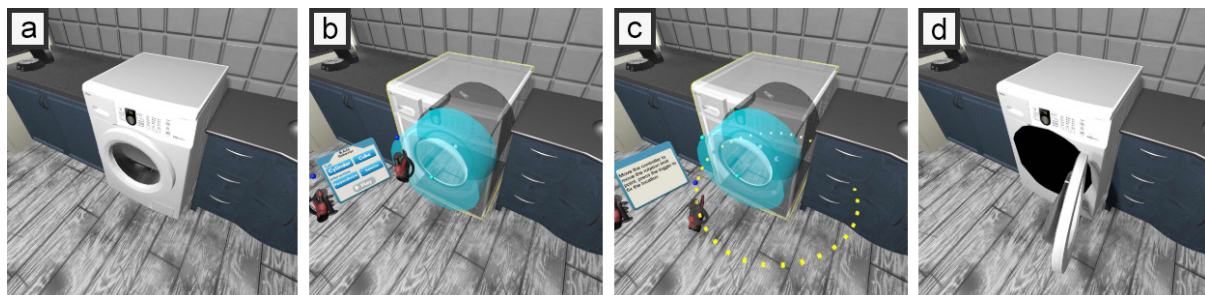


This contradicts previous results and leaves room for questions about the benefits of SR in different types of entertainment games.

In all three examples, the environments were created using 3D modelling software and manually aligned with the real world, which is a time-consuming workflow.

### 3.3.3 Non-Substitutional Reality Creation

The following works focus purely on creating VR experiences and not SR experiences that benefit from passive haptics. VRFromX by Ipsita et al. (2021) uses the LiDAR scanner on an iPad Pro to capture a 3D mesh of a space which can then be loaded into desktop VR. Users can use artificial intelligence (AI) assisted tools to implement basic interactions for serious games and applications. Participants found the system easy to use (albeit with scanned data gathered previously by the researchers) and were able to create their own interactive experiences. However, due to the use of AI tools, the system needs to run on a powerful external machine, rather than locally on the VR headset. Masnadi et al. (2020) created AffordIt! which allows users to take static meshes of objects (such as a washing machine) and to add interactive movement or rotational elements (see **Figure 38**). Users can segment parts of the mesh to separate them and then add behaviours, such as making the washing machine drum rotate or a door that can open.

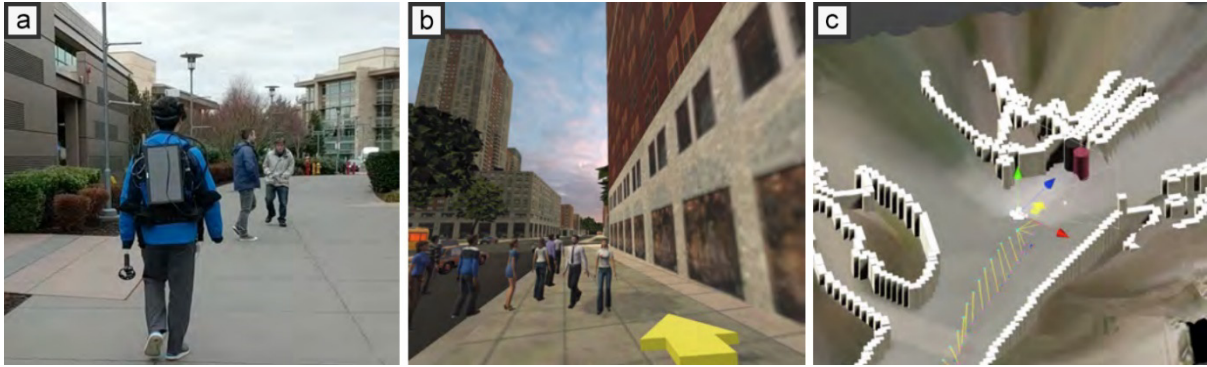


**Figure 38** - AffordIt! application for adding interaction to models in VR. (a) An object to add interaction to. (b) The user cutting into the mesh to add interaction. (c) The user defining the extents of the rotation interaction. (d) The interaction in action (Masnadi et al., 2020).

Again, users found the ability to create behaviours in VR accessible. Both show positive results for editing functionality in VR but do not incorporate SR experiences through mapped digital proxies.

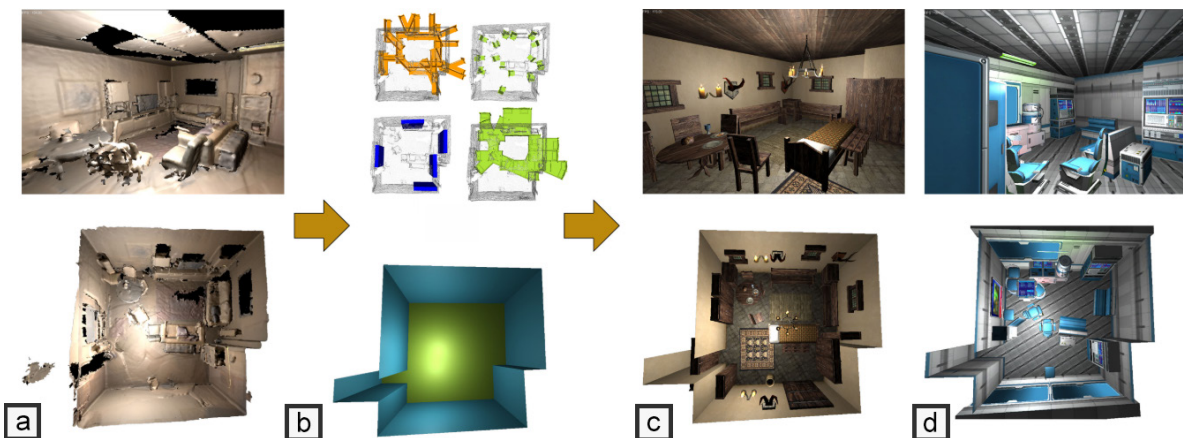
### 3.3.4 Automated or Procedural Substitutional Reality Creation

There are many examples of work that explore the creation of SR environments through automated and procedural techniques. Yang et al. (2019) developed DreamWalker, a system that generates VR experiences whilst walking between locations and builds on their similar VR Roamer work (Cheng et al., 2019).



**Figure 39** - DreamWalker system in use. (a) The real-world unseen location traversed by use with VR headset and laptop. (b) The VR environment generated to ensure the user does not collide with obstacles or people. (c) A demonstration of what DreamWalker detects (Yang et al., 2019).

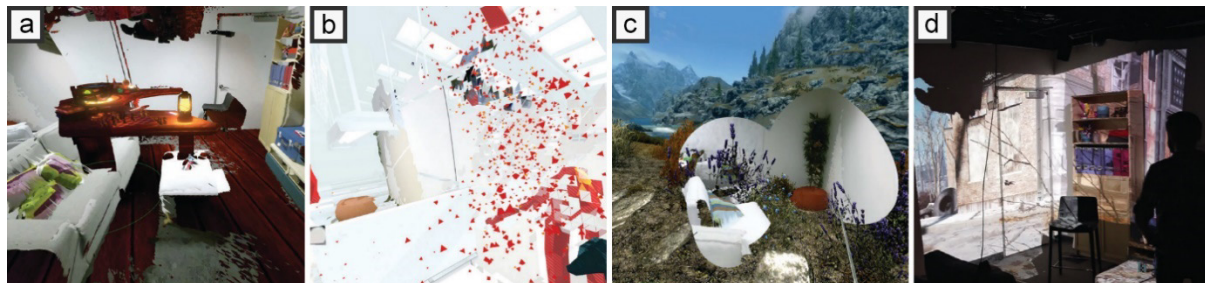
It partially uses redirected walking and an RGB-D camera to map the space on the fly and create a substituted virtual version (see **Figure 39**). This allows users to effectively walk between locations by ensuring they avoid obstacles but is not aimed at creating passive haptics. Sun et al. (2016) also looked at using redirected walking in a known space to allow for continuous locomotion with generated virtual environments. They utilise a 2D floor plan, rather than a depth camera to facilitate this. Reality Skins, developed by Shapira and Freedman (2016) aims at procedurally generating SR environments for a single room, rather than for walking between locations and to enable passive haptics. It utilises pre-scanned room data to generate a new digital proxy for the scene which is processed and executed offline on a PC. It produces effective virtual environments but no game play. The level of mismatch between shape and location of digital proxies can also be quite large, which could result in immersion breaking and dangerous situations (see **Figure 40**).



**Figure 40** - Reality Skins process. (a) A room is scanned using an RGB-D camera (like Microsoft Kinect). (b) The floor plan is generated, and location of objects located. (c) A medieval tavern theme is generated. (d) A sci-fi hospital theme is generated (Shapira & Freedman, 2016).

Oasis developed by Sra et al. (2016; 2017) again uses an RGB-D camera (a Google Tango enabled tablet) to scan a space and to produce a virtual environment. The scanning and

processing are done offline on a PC, but the immersive experience runs standalone on an HMD that uses the Tango tablet. The system produces effective floor plans but makes minimal use of objects in the space. Hartmann et al. developed RealityCheck in 2019. This system utilises an Intel RealSense camera to capture the real-world live and then intercepts any game utilising OpenVR to create a composite image (see **Figure 41**).



**Figure 41** - Various demonstrations of RealityCheck. (a) Blending between *Waltz of the Wizard* and real world. (b) Collision estimates in the *Superhot* VR game. (c) Flashlight concept that allows users to choose where to reveal real-world. (d). A projection of what the user is viewing for co-located spectatorship (Hartmann et al., 2019).

This can be used to create a more effective chaperone system that is more advanced than what is currently available on some consumer headsets or within bespoke games. The system runs on a high-spec PC and the solution can be quite demanding (impacting framerate), meaning it is unlikely to run on a standalone system in its current state. Finally, Yixian et al. (2020) created a unique system called ZoomWalls, which features autonomous robotic wall shaped props that move around the scene to reconfigure themselves in accordance with the virtual environment. The system effectively increased presence in their participants but does still require a large open space and robotic props. All these systems show promising results but do require in most cases specialist equipment and expensive, technical set ups. They also primarily focus on creating a virtual environment but no game play implementation.

### 3.3.5 Manual/Human-in-the-Loop Substitutional Reality Creation

The following examples allow for the creation of SR environments through manual creation methods, sometimes known as ‘human-in-the-loop’ techniques. Lin et al. (2020) developed ARchitect, which takes the form of an AR smartphone application that allows users to construct



**Figure 42** - ARchitect supports SR level creation in AR. (a) The real-world location and obstacles. (b) A user editing the level using smartphone AR. (c) The user adding a tree stump in place of a chair. (d) The final game level (Lin et al, 2020).



an environment in AR first, by placing proxy models in place of the real-world objects (see **Figure 42**). Users then play the game using the smartphone as a VR headset. ARchitect shows promise as users can design their game level as they move around the real world. However, it does not allow for more complex interactions due to lack of 6-DoF controller input or more complex level editing (only placement of existing models). The levels are also created entirely in an AR view, which some users found difficult to interact with accurately due to having to rely on the touch screen and gestures.

Work by Garcia et al. (2018) utilised an HTC Vive to test two techniques for mapping a real-world space. One technique used the HTC Vive's monoscopic camera to give the HMD user a view of the outside world as they used the controller to mark points in the space to create volumes around objects. The second technique used a tablet as a virtual camera instead of the HMD, but users still used the Vive controller to map the space. They found that the HMD system was preferred and more accurate. Their technique is promising but was designed to work on the Vive as it needs developer accessible cameras and fixed tracking units to align the virtual map with the real world. The Meta Quest does now support passthrough camera developer support (post the creation of LevelEd SR), so this system is now more viable, but would not be compatible with headsets that do not support passthrough. The system also only went as far as adding digital proxy models to a scene. Both systems show promise but have limitations that need solving to ensure SR game levels can be created using popular consumer VR hardware, be as portable as possible and allow for more complex level editing.

It is clear from this review that whilst SR is an active research area, there is limited work exploring the benefits of SR on entertainment games and workflows that enable players to create their own SR game levels and experiences using consumer technology.

### **3.4 Summary**

The literature search has revealed several gaps in the knowledge surrounding the use of AR and VR technologies to assist with current serious or entertainment game level design workflows or new workflows that enable new experiences.

#### **3.4.1 Mapping a Real-World Location – Workflow 1**

Current workflows for generating digital maps of real-world locations for possible use in VR serious games often requires expensive or non-consumer focused technologies. There are very limited examples of research using smartphone AR to help generate maps of real-world locations for use in serious games. The design and implementation of LevelEd AR and Workflow 1 was created to explore the use of smartphone AR in generating digital maps of real-world locations. This is detailed in the **Chapter 4 LevelEd AR – Workflow 1**.

### **3.4.2 Virtual Reality Level Design – Workflow 2**

Current workflows for creating entertainment-based VR game levels normally require developers to work on a 2D display to create a level and then check the level by putting on a VR headset whenever it needs testing. It is felt that this could be improved through developing VR game levels whilst in VR. Whilst there are examples of research exploring world building in VR, these primarily focus solely on building an environment and often with pre-made assets. This ignores the full level design workflow that requires blocking out a game level, prototyping game play and testing the game level. The design and implementation of LevelEd VR and Workflow 2 has been created to explore how effective a VR level editor is and the potential benefits to the user. This is detailed in **Chapter 5 LevelEd VR – Workflow 2**.

### **3.4.3 Substitutional Reality Level Design – Workflow 3**

Finally, there is little evidence from the literature of workflows that utilise consumer technology, such as a smartphone and portable HMD, to create and experience SR game levels. The design and implementation of LevelEd SR and Workflow 3 has been developed to further explore SR game experiences and a potential workflow to enable them. This is detailed in **Chapter 6 LevelEd SR – Workflow 3**.

# Chapter 4 LevelEd AR – Workflow 1

The following chapter outlines the motivation for Workflow 1, along with the design of the LevelEd AR software that enables this workflow. A study was devised and carried out to evaluate the workflow and LevelEd AR software and the results of this study are presented here along with a discussion.

## 4.1 Introduction and Motivation

LevelEd AR was developed as part of the exploration of how AR could assist or improve current level design workflows for developers (see **Figure 43**). An area that was highlighted early on was that of serious games level design. When developing serious games, designers often create virtual worlds from scratch that facilitate the user and the intended experience. However, a serious game virtual world, which employs a personal space or addresses prior user knowledge, may benefit from being based on a real-world location rather than imagination.

One example is a wheelchair driving simulator used to train new powered wheelchair users in VR (John et al., 2018). This system may benefit from users training in a real-world location, such as their home or place of work. Using such spaces could help train users to navigate commonly visited locations and potentially reveal accessibility issues within these spaces. Serious games for virtual evacuation training (Lovreglio et al., 2017) and virtual fire safety training (Chittaro & Ranon, 2009) could also benefit from being based on a real-world location by allowing users to train in the actual location of a potential emergency evacuation. In all these cases, rigorous spacing and accurate depiction of distances and gaps play a particularly key role in the simulators' efficiency and usability. Research suggests creating environments as close to reality benefits learning transfer, specifically for those with disabilities (Politis et al., 2017) and that the level of presence and realism of the environment play key roles in the development of skills in a serious game (Lovreglio et al., 2018; Chen & Chien, 2022).

There is also a benefit for entertainment and serious games applications that support passive haptics (Insko, 2001) and SR (Simeone, 2015; Ekstein et al., 2018). These are forms of augmented virtuality (Milgram, 1994) which blend full digital worlds with real-world locations. Users of these applications can move around a real-world space and interact with walls and physical objects whilst the VR headset displays a digital world on top of the space. LevelEd AR was therefore developed with this future use case in mind.

To enable the complex task of indoor modeling a multitude of techniques have been used in the past.

- Static and mobile laser scanners can be used to create complex point cloud virtual models which are commonly used in construction (Volk et al., 2014).
- Mapping systems that utilize infra-red (IR) depth sensors to model an indoor space are sometimes used by interior designers and builders (Kalantari & Nechifor, 2016).
- Manual capture of measurements with a tape measure on a floor plan are often used for DIY projects.

These techniques, whilst effective can be time consuming and are not always accessible due to cost and technical ability required for use. The resulting model/data produced is often not suitable for use in a serious game or SR game without significant adjustments due to their high fidelity.

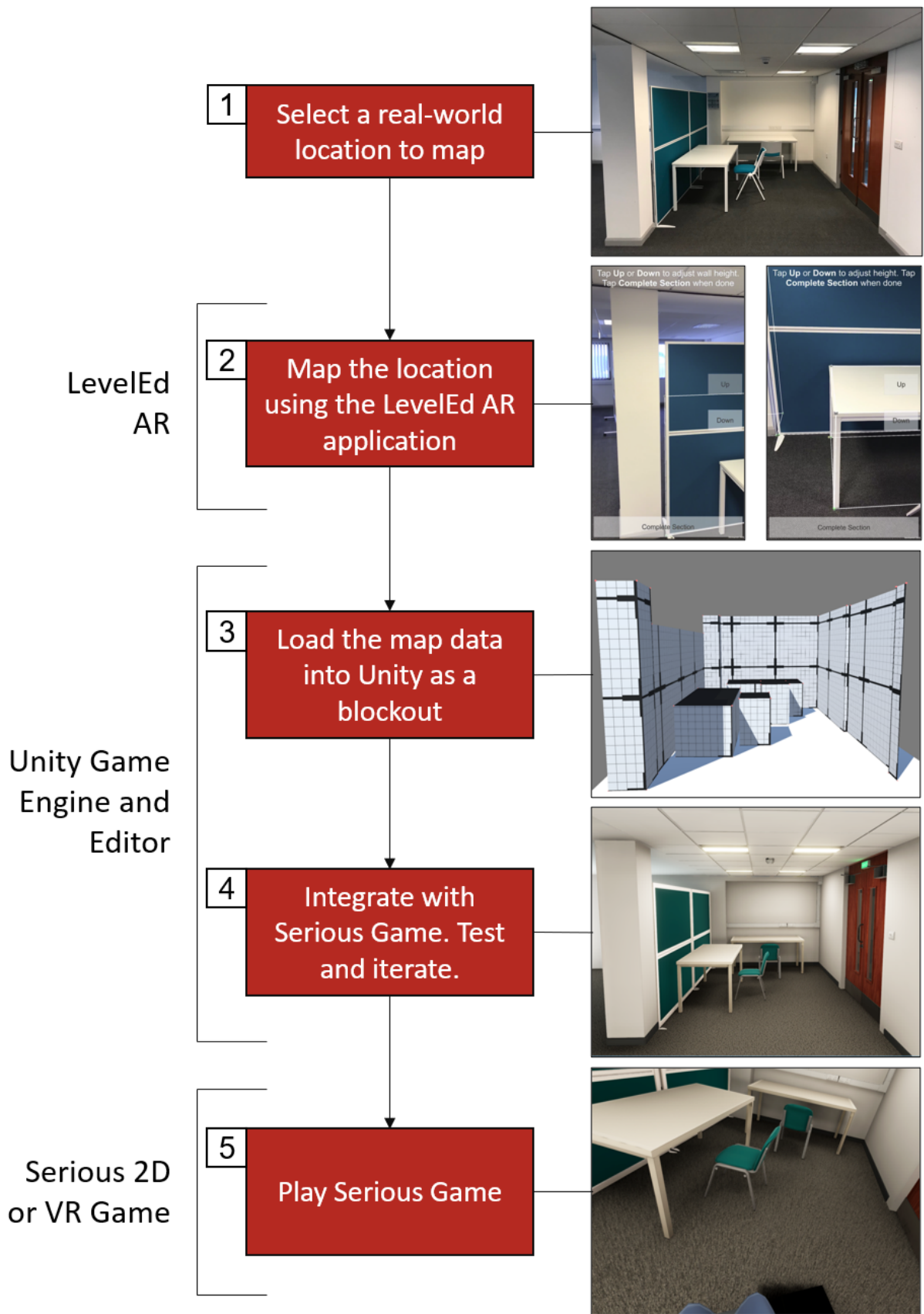
## 4.2 Workflow 1

Workflow 1 was developed to allow users to capture a model of a real-world location that is suitable for use in a game engine such as Unity, for serious games level design, with an eye to potential future use in enabling SR with Workflow 3. LevelEd AR, a bespoke smartphone AR application, was created to enable this workflow. The study outlined later in this chapter compares the precision of the system to existing techniques and evaluates the usability of the system.

Based on a review of the literature, an indoor modelling system that was suitable for serious game level design had not been developed and investigated. LevelEd AR was designed as a proof of concept that could be evaluated in a user study with the following goals:

- Produce a simplified mesh model that can be used directly in a serious game virtual environment or as a guide during development.
- Model both walls and objects within the space, accurately placed with respect to their location, where necessary, including irregular rooms.
- Give the user the choice as to what parts of a space are modelled or not modelled.
- Be accessible to users without the requirement of additional costly and complex hardware and thereby opening it up to a larger range of users.

A walkthrough of Workflow 1 can be seen below in **Figure 43**:



*Figure 43 - Walkthrough of Workflow 1 that enables a user to capture a space using AR to later use in Unity to develop to scale serious game levels. Further details on using the application can be found in section 4.4 and Figure 44*

To achieve the above, LevelEd AR supports the following features:

- Wall mapping – Users can map the location, shape and height of walls within a room.
- Object mapping – Users can map the location and general shape and size of objects within the space, such as furniture.
- Saving – The data is saved to the device.
- Uploading – The data is uploaded to a webserver.
- Downloading and Recreating – The data can be downloaded to a Unity project and recreated within a scene.

A video demonstration of the system can be found at **Appendix A1** and **Appendix A2** and access to the code via **Appendix A3i** and **Appendix A3ii**.

### 4.3 Hypothesis

The following hypothesis was devised to address the ability of Workflow 1, which uses AR technology, to improve an existing workflow for developers:

*H1: An augmented reality smartphone application is an acceptable digital alternative (in terms of a trade-off between speed and ease of use) for capturing accurate room scale data.*

The study outlined in section 4.5 was devised to test the hypothesis, with the results presented and discussed later in this chapter.

### 4.4 System Overview and Implementation

The system was developed starting in May 2017 for iOS devices that support ARKit (iPhone 6S and newer) using the game engine, Unity 2017.1 (Unity Technologies, San Francisco, CA). One of the aims for LevelEd AR was to ensure it is widely accessible by making use of readily available consumer technology. Apple's ARKit 1.0 beta (officially released September 2017) was selected for this project due to the wide availability of compatible devices. As of July 2017, there were an estimated 380 million ARKit compatible smartphone devices with this growing to around 1.5 billion devices including those that support ARCore by 2020 (Révész, 2020). To keep development straightforward, only ARKit was supported, but in later versions (see **Chapter 6**) it was updated to also support ARCore. This application is therefore accessible to many users without the need for the purchase or learning of specialist equipment. WebXR was not considered for this project as it was not available at that time.

LevelEd AR presents users with an AR view of the real-world location they are modelling with transparent overlay UI that guides them through the process (see **Figure 44**). Users first confirm the floor location by pointing the device down, scanning the ground and then selecting

the surface by tapping the screen. The user moves around the real-world location whilst aiming a yellow marker in the centre of their smartphone screen at key locations (such as room corners or around objects). By tapping the screen, the user can place an AR marker into the scene. This marker stays in place as the user continues to model the rest of the space. Users have two modes available to them:

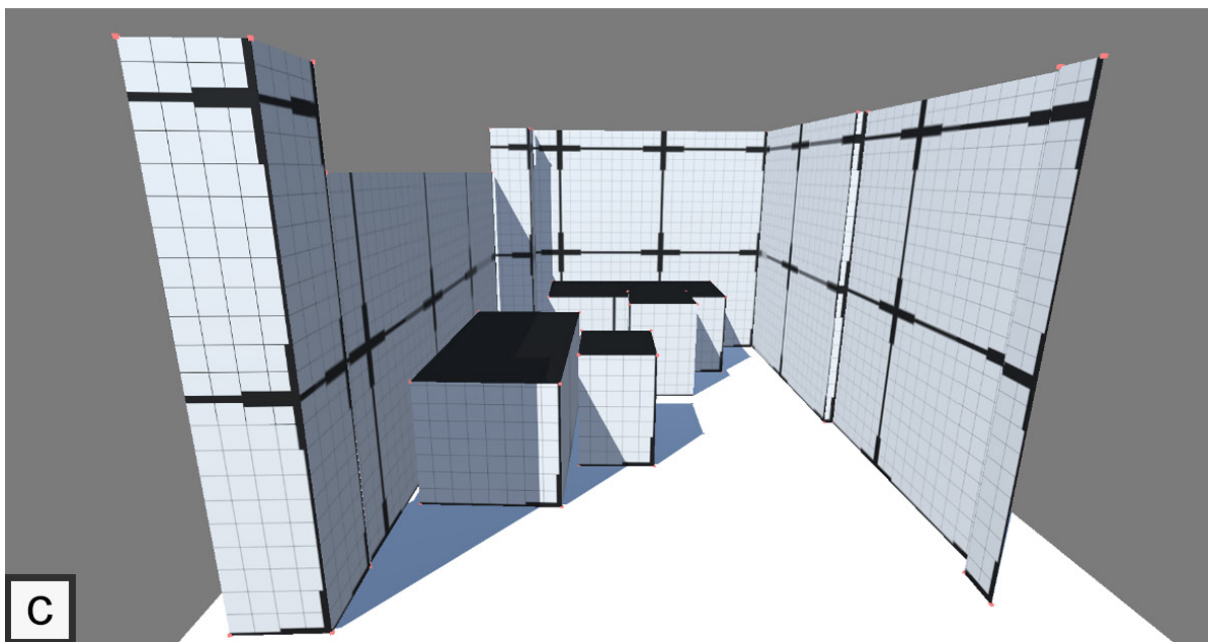
- **Modelling walls:** users place markers at intersections of walls within a room to map out the base of the walls. They can then raise up a second set of markers for the height of the walls (see **Figure 44a**).
- **Modelling 3D objects:** users place markers to surround the object. This can be an object of any number of sides but in the study, it was set to four. Once the base markers are in place, the user can raise up a second set of markers for the object height (see **Figure 44b**).

The above modes enable users to model a space effectively, including not only the walls but also the obstacles and furniture (general shape for more complex furniture) within the space. The shape, location, and scale of these are all captured by LevelEd AR. Users are also free to choose what will be included in the model. This avoids the modelling of temporary objects found in a space (such as rubbish or people), a problem which automated indoor modelling systems can suffer from.

LevelEd AR makes use of several key ARKit functions, such as the ability to detect horizontal planes and key points-of-interest. The system works by casting a ray into the scene from the centre of the screen (filled with the AR camera view). A marker object tracks the raycast hit location and can then be anchored in place with a tap of the screen. The marker locations are used to create data in the form of wall objects (a series of planes) or 3D objects (of any number of sides) (see **Appendix N1**).

Once the user has captured all the wall and/or objects they require, they can then save the data. The data is serialized to a file and then uploaded to a webserver for accessing later in Unity (see **Appendix N2**). It is interesting to note that in Unity/C# Vector3 data types are not serializable in a file and therefore several custom data structures were created to store this data (see **Appendix N3**).

In Unity, the data can then be downloaded, and a model of the environment generated (**Figure 44c**) from the data to be used as part of the level design process as a guide or in some cases such as walls, used in the final version of the level/virtual environment (**Figure 50d**).



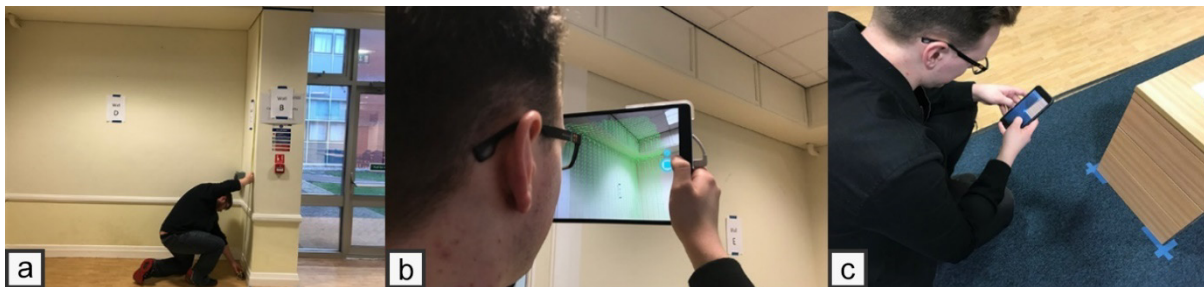
**Figure 44** – LevelEd AR workflow from AR to Unity game engine. (a) User is mapping the dividing wall; the shape, scale and location will be captured. (b) User has completed the mapping of the shape, size and location of the table. (c) Model has been generated in Unity engine from the data uploaded by the user. Grid squares are 1m x 1m, with 10cm subsections.



## 4.5 Study 1 - Experiment Methodology

The experiment (completed January 2018) was primarily developed to evaluate the accuracy of LevelEd AR for indoor modelling of real-world locations. However, the underlying technology, ARKit, and its ability to accurately track the movement of a user and maintain the position of objects in a space over time was also evaluated consequently. There was a lack of data on ARKit accuracy at the time of the study. The experiment was also designed to explore the usability of AR as a means for enabling indoor modelling on consumer devices. Ethical approval was granted by the FSE Research Ethics Committee (see **Appendix F**).

### 4.5.1 Experiment Tasks



**Figure 45** – (a) Participant completing Task 4 with measuring tape and paper. (b) Participant completing Task 4 with the Room Capture software with the Structure Sensor. (c) Participant completing Task 2 with the LevelEd AR application.

To evaluate the accuracy and usability, participants were asked to complete four separate measuring tasks (**Figure 45**). Participants were informed that the tasks all focused on measuring objects and/or the space to enable a digital recreation. These tasks were as follows:

- **Task 1:** Measuring a single wall – the length and height were captured.
- **Task 2:** Measuring a small cupboard with open space around all sides – the length, depth and height of the small cupboard were captured
- **Task 3:** Measuring a large cupboard against a wall – the length, depth and height of the large cupboard were captured.
- **Task 4:** Measuring four consecutive walls and a small filing cabinet – measurements recorded were the length and height of each wall (Task 4.1), the length, depth, and height of the small filing cabinet (Task 4.2) and the distance of the cabinet from the first wall (Task 4.3).

The above tasks allowed for the evaluation of the accuracy of the different measuring techniques under controlled circumstances. The first three tasks check the accuracy of completing measurements of singular common objects whilst the fourth task was included to

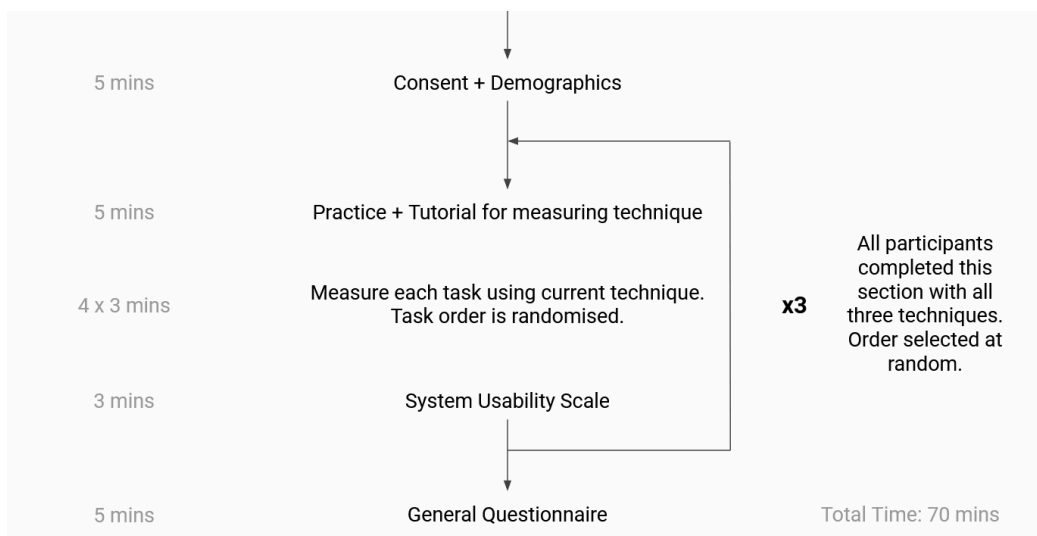
replicate a more complex task such as modelling a whole room including furniture. The time taken to capture the measurements for each of the above tasks was also recorded.

#### 4.5.2 Measuring Instruments

Participants completed the measurement of the four tasks using three different measuring instruments (see **Figure 45**). These instruments were selected based on their similarity in cost and accessibility to the proposed system. The instruments were:

- **Measuring tape and paper:** users manually measured the tasks using a tape measure and recorded the measurements on a sheet of paper provided. The time taken was recorded from when the participant started until the final measurement was captured on the sheet of paper.
- **Room Capture application and Structure Sensor:** users used an iPad Pro 10.5" with a Structure Sensor attachment along with the Occipital Room Capture software to scan the task locations. The time taken was recorded from when the scan was started until the scan was completed. Once the participant was happy with the scan, they were then asked to use the Room Capture built in measurement tool to acquire the requested measurements from the scanned model.
- **LevelEd AR application:** users used an iPhone 7 Plus and the LevelEd AR application to model the tasks in AR. The time was taken from when the users started the application and until they had modeled the task and uploaded the file to the webserver.

Participants utilized all three measuring instruments to complete all four measuring tasks. A randomized crossover design was used for both the order of measuring instruments utilized and the order of measuring tasks completed (see **Figure 46**).



**Figure 46** - Structure and flow of Study 1

### 4.5.3 Participants

18 participants completed the study, three identifying as female and 15 as male, ranging from 18-59 years of age. 27.8% were between the ages of 50-59, 5.6% between the ages of 40-49, 22.2% between the ages of 30-39 and 44.4% between the ages of 18-29. Prior experience of AR was mixed with 16.7% having no prior experience, 38.9% rating themselves as novices, 27.8% rating themselves as intermediate and 16.7% rating themselves as advanced.

Participants were selected through convenience sampling and consisted of students and staff in the Faculty of Science and Engineering at the University of Chester. Participants were recruited via an email sent out to staff and students in the faculty (see **Appendix K**). One challenge of this study was a limitation on how long the room could be booked and kept set up for the study. Therefore, convenience sampling was selected to maximise participant numbers whilst remaining cost and time effective; a key benefit to convenience sampling (Etikan et al., 2016). The demographic of this convenience sample demonstrates an imbalance in gender (more male identifying participants) and age (participants skewed to the younger age group). This limits the ability for the results to be generalised to the wider population and should be viewed through the lens of being comparative to similar populations of engineering faculties.

## 4.6 Results

Throughout this section the measuring techniques used are called instruments and denoted with *Tape* for measuring tape and paper (which was also used as the ground truth), *LevelEd* for the AR application and *Structure* for the Structure Sensor and Occipital Room Capture application. Considering the measurements required, the fourth task incorporated three different values and has been split into Tasks 4.1, 4.2 and 4.3.

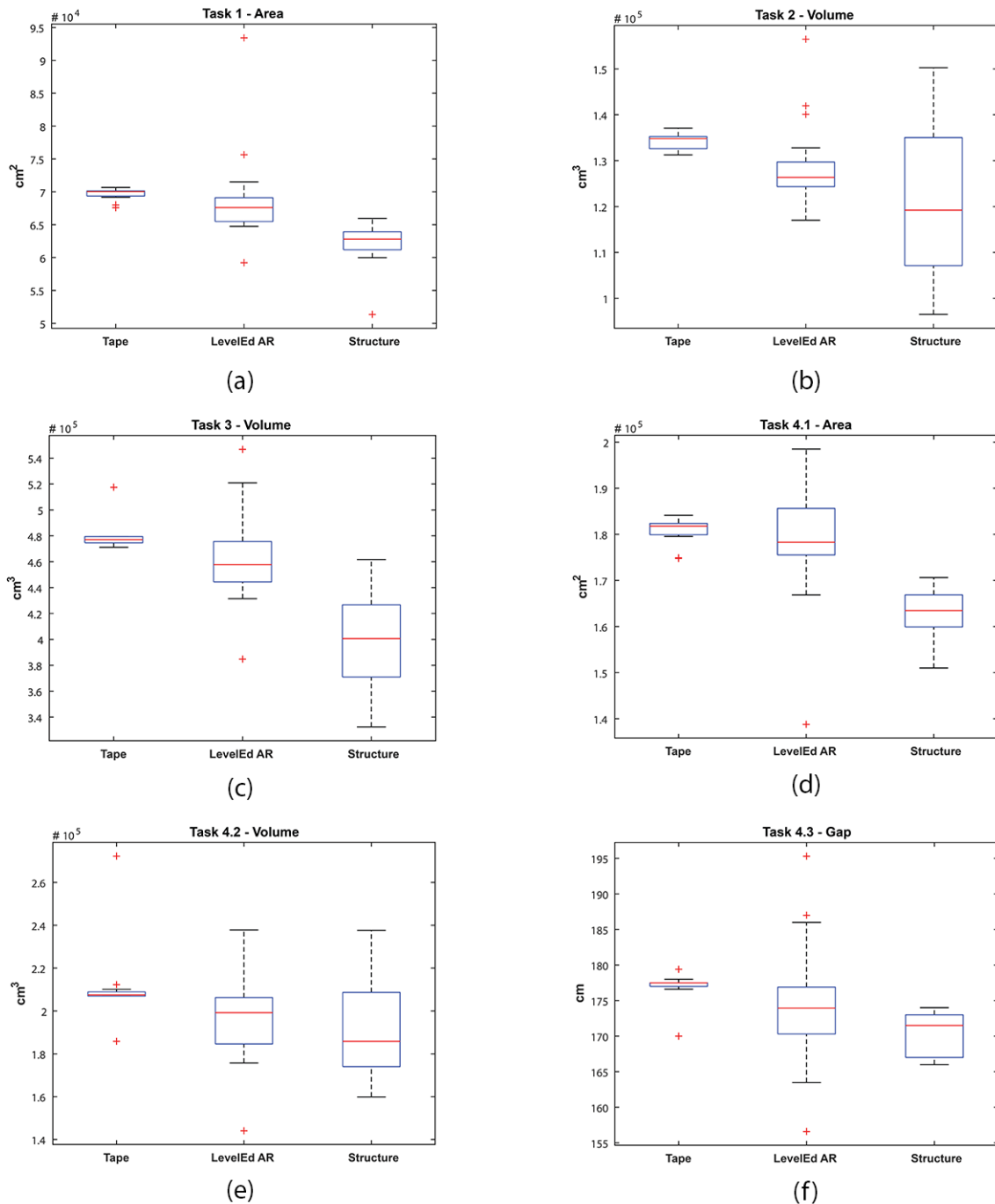
The significance was tested by employing a two-way repeated measures ANOVA for both measurements and time, a method supported by the large effect sizes observed throughout. The degrees of freedom were adjusted to the lower bound estimate according to the result of the sphericity test.

### 4.6.1 Measurements

The results show that the instruments significantly differ from each other in terms of performance overall ( $F(2,34) = 73.89, p < .001, \eta^2 p = .813$ ). The same effect was observed for the tasks in all cases, which suggests the tasks vary significantly in complexity ( $F(1,17) = 4533.90, p < .001, \eta^2 p = .996$ ). With respect to the interaction between instruments and tasks, we observed that each instrument performs significantly stronger on some of the tasks but weakly on others ( $F(1,17) = 20.14, p < .001, \eta^2 p = .542$ ), an important result which needed to

be investigated further. The significant interaction was followed up with six separate one-way ANOVAs. The results were plotted to identify and visualize significant trends, which will help characterise the interaction between instrument and task.

The first task (**Figure 47a**) required an area measurement, and using the Tape as the ground truth, it is shown that using *Tape* or *LevelEd* produces no significant difference. Planned



**Figure 47** – (a) Task 1 – Area measurement of single wall, (b) Task 2 – Volume measurement of small box, (c) Task 3 – Volume measurement of larger box against a wall, (d) Task 4.1 – Area measurement of 4 continuous walls, (e) Task 4.2 – Volume measurement of small box, (f) Task 4.3 – Gap measurement between Task 4.2 box and starting wall of Task 4.1.

contrasts showed that for Task 1, the *Tape* value was significantly larger than *Structure*'s measurement ( $F(1,17)=88.47$ ,  $p<0.001$ ,  $\eta^2p = 0.839$ ). The lack of complexity in this task brought no difference regarding traditional point-by-point measurements, however *Structure*'s under reported measurements could be due to collapsing walls shortening the distances recorded as previously experienced by others who used a similar technique (Kalantari & Nechifor, 2016).

Unlike the first task, the second task showed no difference between *LevelEd* and *Structure*, however, both were significantly separated by the ground truth. Hence, for Task 2 (**Figure 47b**), the *Tape* measurement is significantly different from both *LevelEd* ( $F(1,17)=7.06$ ,  $p=0.017$ ,  $\eta^2p = 0.293$ ) and *Structure*'s outputs ( $F(1,17)=13.94$ ,  $p=0.002$ ,  $\eta^2p = 0.450$ ). In this task, the complexity increased, and the *LevelEd* results supported by several large outliers were not significantly different to the large variation in the *Structure* measurements.

The next two tasks were of a higher complexity than the previous ones and in both, the *Structure* sensor showed a significant loss in accuracy in comparison with the other two instruments (**Figure 47c & Figure 47d**), hence, the *Structure*'s accuracy was significantly lower than both *Tape* and *LevelEd*. The *Tape* instrument had larger values than *Structure*'s (Task 3 -  $F=86.01$ ,  $p<0.001$ ,  $\eta^2p =0.835$ ; Task 4.1 -  $F=212.13$ ,  $p<0.001$ ,  $\eta^2p =0.926$ ); and *LevelEd*'s values followed the same pattern (Task 3 -  $F=36.59$ ,  $p<0.001$ ,  $\eta^2p = 0.683$ ; Task 4.1 -  $F=37.70$ ,  $p<0.001$ ,  $\eta^2p = 0.689$ ). Although Task 3 is similar to Task 2 due to both measuring the volume of a box, Task 3 featured a much larger box situated next to a wall. This increased the complexity of the task and decisions required from the participant. Task 4 required participants to move the iPad more significantly whilst completing the task with *Structure* than previous tasks. This often resulted in walls shortening and collapsing in on themselves in the scanned model (Kalantari & Nechifor, 2016), as reported above for Task 1. This was not as pronounced with *LevelEd*, although there were some over and under recording of distances.

Results recorded for Task 4.2, a volume measurement, showed the same pattern as for Task 2 where the same type of measurement was required (**Figure 47e**). Hence, the *Tape* measurement significantly differs to both *LevelEd* ( $F(1,17)=5.66$ ,  $p=0.029$ ,  $\eta^2p = 0.250$ ) and *Structure* ( $F(1,17)=4.74$ ,  $p=0.044$ ,  $\eta^2p = 0.218$ ). At this task, both instruments employed failed to show differences, providing in the process a loss in accuracy and larger variations over *Tape*. Some of the factors responsible for this result were the task's limited complexity (a box similar in shape and size to Task 2), the order completed within Task 4, and subsequent exhaustion of the participants. Another aspect for *LevelEd* with Task 4.2 is the potential for drift (tracking inaccuracies) to occur, increasing over time.

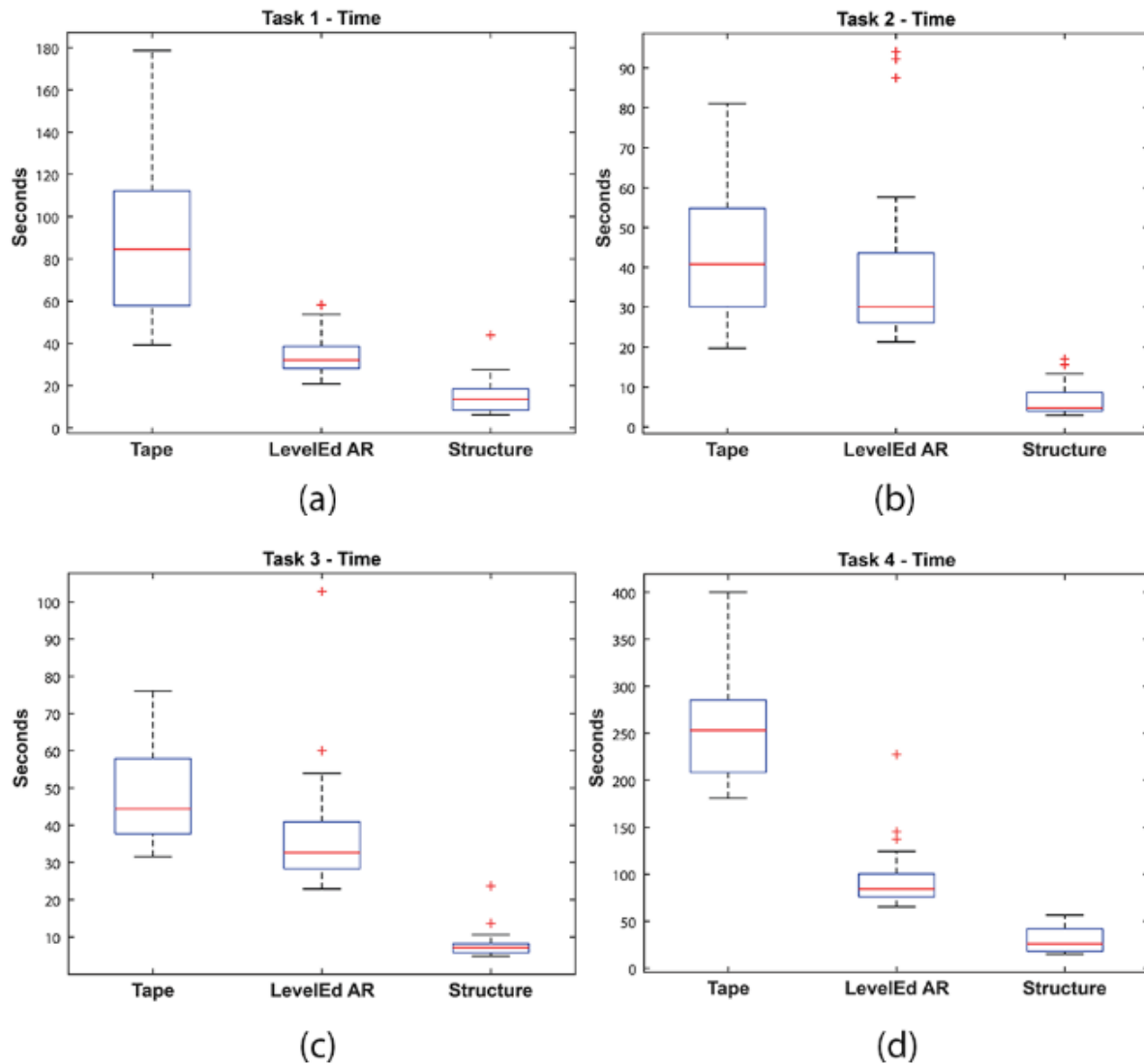
Finally, at Task 4.3, only a straightforward gap measurement was required (between the starting wall of Task 4.1 and the box in Task 4.2). As expected, the Tape measurement was larger than *Structure*'s ( $F=85.63$ ,  $p<0.001$ ,  $\eta^2p =0.834$ ) with *LevelEd* being no different than the ground truth (see **Figure 47f**). However, the larger variation in the measurements of *LevelEd* may be explained due to the potential for drift to occur more frequently over time with markerless AR (Fusco & Coughlan, 2018; Dilek & Erol, 2018).

#### 4.6.2 Time

Similarly, for the time (measured in seconds), significant differences were observed throughout the test between the choices of the instrument ( $F(2,34) = 116.99$ ,  $p<.001$ ,  $\eta^2p = .873$ ). The same effect was observed for the individual tasks, hence their significant difference in complexity was preserved. Moreover, in this case, there were only four tasks as the duration of the sub-tasks of Task 4 were summed up ( $F(1,17) = 249.58$ ,  $p<.001$ ,  $\eta^2p = .936$ ). We observed that each instrument performs significantly different with each completed task ( $F(1,17) = 119.36$ ,  $p<.001$ ,  $\eta^2p = .875$ ).

Following the same pattern, it can be observed that for the first task (**Figure 48a**) the time required to perform the measurements differed significantly between all three instruments, with the *Structure* instrument being the fastest. Planned contrasts (four separate one-way ANOVAs) showed that for Task 1, the time participants spent using the *Tape* instrument was significantly larger than using *LevelEd* ( $F(1,17)=46.75$ ,  $p<0.001$ ,  $\eta^2p = 0.733$ ) and *Structure* ( $F(1,17)=80.23$ ,  $p<0.001$ ,  $\eta^2p = 0.825$ ). Moreover, the time using *LevelEd* was subsequently higher than using the *Structure* ( $F(1,17)=40.44$ ,  $p<0.001$ ,  $\eta^2p = 0.704$ ). The results of Task 2 (**Figure 48b**) showed that the time spent using *Structure* was significantly lower than both other instruments (vs. *Tape* -  $F(1,17)=84.41$ ,  $p<0.001$ ,  $\eta^2p = 0.832$ ; vs. *LevelEd* -  $F(1,17)=31.15$ ,  $p<0.001$ ,  $\eta^2p = 0.647$ ). The same pattern was observed for Task 3 (**Figure 48c**), where using the *Structure* proved to be the most efficient time-wise (vs. *Tape* -  $F(1,17)=93.36$ ,  $p<0.001$ ,  $\eta^2p = 0.846$ ; vs. *LevelEd* -  $F(1,17)=21.18$ ,  $p<0.001$ ,  $\eta^2p = 0.587$ ). The similar time between *LevelEd* and *Tape* could be due to the boxes being sufficiently small enough for the participant to easily capture the three measurements required to calculate the volume with Tape.

A similar behaviour to Task 1 was registered for the last task, the most time consuming one (**Figure 48d**). Planned contrasts showed that for Task 4, the time participants spent using the *Tape* instrument was significantly larger than using *LevelEd* ( $F(1,17)=103.81$ ,  $p<0.001$ ,  $\eta^2p = 0.859$ ) and *Structure* ( $F(1,17)=286.01$ ,  $p<0.001$ ,  $\eta^2p = 0.944$ ). Moreover the time using

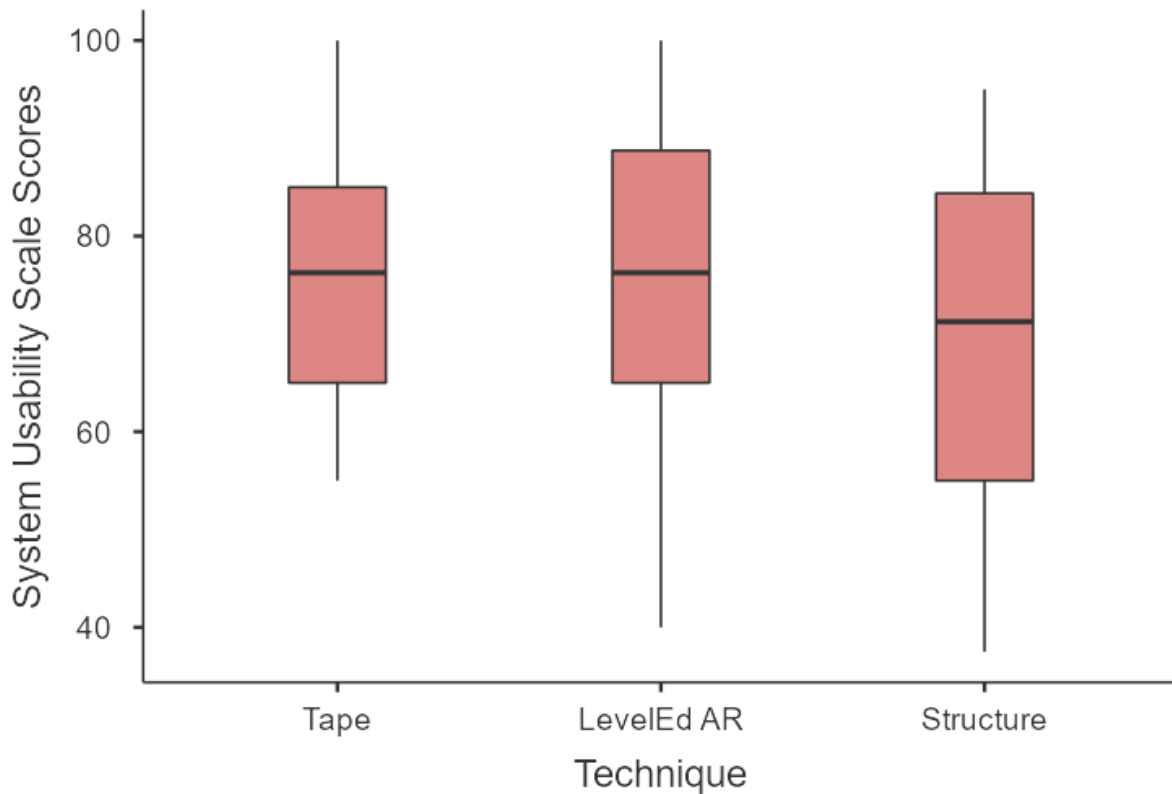


**Figure 48** – (a) Task 1 – Time to measure single wall, (b) Task 2 – Time to measure small box, (c) Task 3 – Time to measure larger box against a wall, (d) Task 4 – Time to measure 4 continuous walls, small box, and gap between box and starting wall.

*Structure* was overall lower than all the others (vs. *LevelEd* -  $F(1,17)=78.78$ ,  $p<0.001$ ,  $\eta^2p = 0.823$ ). As the tasks increased in complexity, the gap between *Tape* and the other two instruments appears to have increased.

#### 4.6.3 System Usability Scale

Participants were also asked to complete a System Usability Scale (SUS) (Brooke, 1996) questionnaire after completing the tasks with each instrument. All three instruments met the SUS usability threshold of 68 (see **Figure 49**).



*Figure 49 – Results from the System Usability Scale questionnaire (Brooke, 1996).*

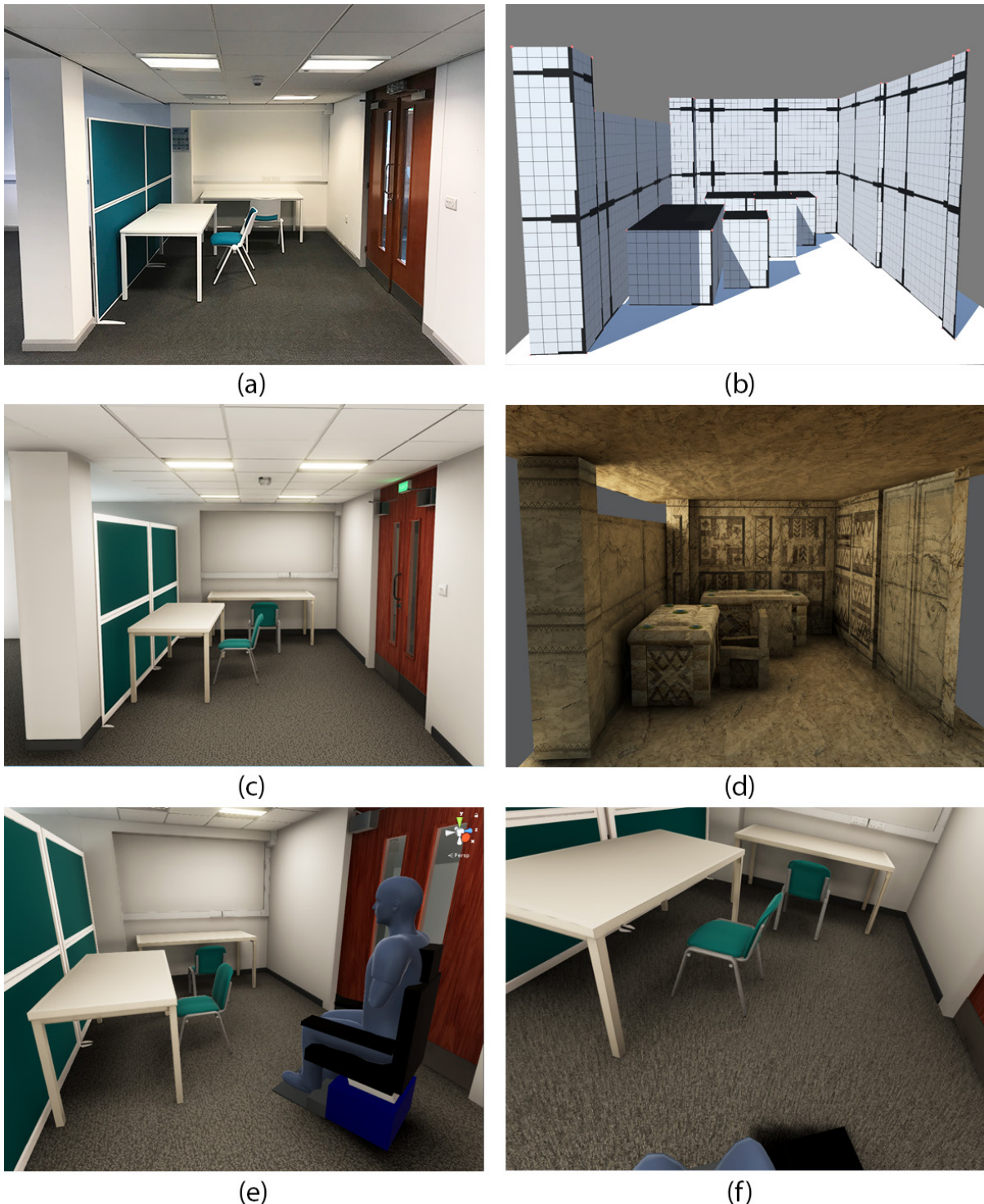
Structure was the least favoured with a mean score of 70, standard deviation of 17, minimum of 38 and maximum of 95. *Tape* was similarly accepted as *LevelEd* with a mean score of 76, standard deviation of 13, minimum of 55 and maximum of 100. Whilst *LevelEd* had a mean score of 74, standard deviation of 18, minimum of 40 and maximum of 100. This is a positive result and suggests that *LevelEd* is more accessible than the *Structure* system for indoor modelling and participants were almost as comfortable using *LevelEd* as the traditional *Tape* instrument.

#### **4.7 Virtual Environment Workflow Preliminary Study**

Whilst the experiment evaluated LevelEd AR for accuracy, time efficiency and usability of the application; a preliminary case study to understand how LevelEd AR can be used effectively as part of a workflow to create detailed virtual environments was also carried out. Three participants were asked to develop two virtual environments which were both based on a real-world location. All three participants had prior skills in 3D modelling. A real-world location was chosen that was more complex than the tasks used in the experiment to effectively test the workflow. The real-world location was captured by the author using the LevelEd AR application and took a total of 3 minutes and 41 seconds to capture and upload to the webserver. The indoor model captured was downloaded in Unity and saved to a scene. Participants were provided with the Unity project with the indoor model ready in the scene (see **Figure 50b**)



along with one photograph of the original real-world location (see **Figure 50a**). Participants were not given access to the location. Ethical approval for this additional study was granted by the FSE Research Ethics Committee (see **Appendix G**).



**Figure 50** – (a) Photo of the real-world location. (b) Model of the location captured by LevelEd AR and downloaded to Unity. (c) Realistic recreation of the location as a serious games virtual environment. (d) Egyptian themed virtual environment created for potential SR. (e) Realistic scene integrated into a powered-wheelchair VR training simulator. (f) View from VR headset when in the realistic scene in the powered-wheelchair VR training simulator.

One participant was asked to recreate the space virtually as close as possible to the original scene so it could be used as part of a serious game virtual environment (see **Figure 50c**). The other two participants were asked to create an Egyptian tomb themed virtual environment that would fit within the confines of the space and map to the objects included in the area (even if this will break the context somewhat). This virtual environment could potentially be used as part of a SR (Simeone, 2015) experience (see **Figure 50d**). Participants spent between 1-3 days creating their highly detailed and game ready environments from the blockout. In the case of the realistic environment, subsequent environments could be created much quicker since the models can be reused.

The serious game scene (**Figure 50c**) was then integrated into a powered wheelchair VR serious games application (John et al., 2018) (**Figure 50e** and **Figure 50f**). The scale of the scene appeared, subjectively, to be realistic and correct with the scene requiring no additional work as it was already built to match the real-world dimensions. This scene could now be reused and integrated into any serious games application that utilises real world scales.

Participants demonstrated the potential for LevelEd AR to be used to capture spaces and to provide a guide (through the generated blockout) from which to develop a virtual environment that is sufficiently spatially accurate for serious games. Participants completed a questionnaire and provided valuable feedback on the workflow. Participants noted that having a captured model to work from in Unity meant they did not need to estimate distances or sizes from the image provided. Instead, the image was used only to gauge colour and texture. Participants suggested the ability to export objects from the captured model that could be imported into 3DS Max and in-scene object measurements would have sped up the workflow. This is something that could be added in future iterations of the workflow.

## 4.8 Discussion

Although the limited sample size is acknowledged, the results of the study show that LevelEd AR application measurements are closer to the *Tape* measure (which is considered to be the ground truth) than the *Structure* sensor and Room Capture application in most tasks. For many tasks, especially the ones of increased complexity (such as Task 4), the AR application proves to be more accurate than the Structure sensor and requires less time than the tape measure. This is a major usability and accessibility benefit which enables the users to acquire fast and reliable geometrical information of their environment using consumer technology. This level of accuracy is important as a more recent study has shown that the scale of the room and large furniture within it are the two most important factors in the plausibility of a space (Skarbez et al., 2022). The SUS scores also present a positive outcome for *LevelEd AR* as it scored

favourably with the *Tape* and with a higher acceptability score than the *Structure* (see **Figure 49**).

Despite the overall positive accuracy and usability of *LevelEd*, there were some tasks that either demonstrated minor inaccuracies (**Figure 47b** – Task 2) or variations (**Figure 47f** – Task 4.3) with measurements. There are several factors that could have caused these inaccuracies/variations:

- The age of participants and their perception of AR.
- Potential for drift to occur over time.
- Fatigue of users over time.

Participants in the 40-59 age group spent a total of between 159 and 478 seconds completing all four tasks whereas users in the 18-39 age group spent a total of between 142 and 205 seconds (see **Table 5** for individual age ranges). It was observed during the experiment that this increase in time was due to some of those in the 40-59 age group struggling to control the marker on screen and with the perception of dealing with a 3D object being presented in a camera feed on a 2D screen. No significance statistical test was performed on this data due to the small number of participants in this age group, however one explanation can be the users' perception towards AR and mobile technology in general. The lack of useful non-games applications (at the time of the test) which used AR might have contributed to this perception. Additional training and tutorials for the application may help alleviate these issues.

*Table 5 – Time to complete all four tasks with LevelEd AR across each age group.*

<b>Age Group</b>	<b>Average Time (seconds)</b>	<b>Min Time (seconds)</b>	<b>Max Time (seconds)</b>
<b>18-29</b>	165	142	205
<b>30-39</b>	184	149	205
<b>40-49</b>	305	305	305
<b>50-59</b>	298	159	478

Another major factor in the accuracy of *LevelEd* AR is the problem of drift. Drift occurs when the device loses track of its position in the real world and becomes out of sync with the virtual world. This can cause measurements to become inaccurate as the system may under or over report the distance it has moved since the last marker was placed. This is a common issue with markerless techniques and in particular with ARKit, as was previously reported by both Fusco and Coughlan (2018) and Dilek and Erol (2018). This study further confirms this is a potential problem.

Common reasons for drift are numerous. It could be attributed to the user moving the device too quickly, causing the camera image to blur and computer vision systems becoming unable to track key points frame to frame. It could be due to some surfaces lacking sufficient points of interest which again causes the computer vision system issues when trying to track key points between frames. The fourth task was intentionally designed to study the above effect and it could be that the longer the amount of time that is spent on a mapping task then the larger the potential for drift to impact the accuracy, as drift and inaccuracies accumulate over time. This is evident in **Figure 47f** – Task 4.3 which shows larger variations in measurements for LevelEd. This was the longest and most complex task as the box was measured after the four walls, thereby increasing opportunities for drift.

Potential solutions to the issue of drift could come from improvements to the computer vision algorithm used in ARKit. The version of ARKit used at the time maintains only a partial map of the scanned area and so this makes recalibration difficult over longer more complex tasks. ARKit has improved from this early version and the latest ARKit 5 does appear to fix some of the issues. However, improvements could also be addressed through the user experience of the application. The system could warn users when they are moving too fast or when drift has potentially happened, prompting users to try again.

Another reason that could explain drift is the participants' fatigue. Users seem to reach exhaustion faster when moving around the room with the attention and focus oriented towards making measurements through the AR device. Others have noted the *gorilla arm* effect of holding a device out for extended periods of time affecting the use of AR (Grubert et al., 2016). This is easily observable in longer tasks as a comparison with the time spent by participants using the Structure app where their immersion and the physical requirement is limited. As reaching physical and mental exhaustion is a process that builds up over time, together with the technical aspects of AR, this can be a reasonable explanation for an increase in drift over time as users rush to finish the task.

## 4.9 Conclusion

This chapter has presented a prototype application that enables users to capture indoor models of real-world locations for use in serious games and entertainment games that could also support passive haptics (Insko, 2001) and SR (Simeone, 2015; Eckstein et al., 2018). The system was developed with consumer technology in the form of a smartphone and Apple's ARKit framework. Users were presented with an AR view of the world whereby they could place markers for walls or objects to capture their location, size and then to define the height. The data captured was then available to download within the Unity engine to generate an indoor model suitable for use directly in a serious game, to be used as a guide whilst

developing a virtual environment, or eventually converted to collision meshes once more complex assets have been added.

In section **4.2 Workflow**, some key features that the system must address were outlined. The software needed to:

- Produce a simplified mesh model that can be used directly in a serious game virtual environment or as a guide during development.
- Model both walls and objects within the space, accurately placed with respect to their location, where necessary, including irregular rooms.
- Give the user the choice as to what parts of a space are modelled or not modelled.
- Be accessible to users without the requirement of additional costly and complex hardware and thereby opening it up to a larger range of users.

These have all been met as the system produces a simplified mesh suitable for direct use in a serious game virtual environment, it allows users to model both walls and objects within the space including irregular rooms, it gives the users a choice as to what should be modelled and finally, it is accessible to users due to the utilisation of ARKit and the support for a large number of pre-existing smartphone devices. Whilst the prototype has been successful at meeting the above aims, there were still minor issues that needed to be addressed that have been highlighted previously in this chapter.

Overall, the study shows that AR can be used to assist in the serious games level design workflow and enhance it with more accurately scaled levels based on real-world locations. The work in this chapter also laid the groundwork for Workflow 3, which is detailed in **Chapter 6**.

## Chapter 5 LevelEd VR – Workflow 2

The following chapter outlines the motivation for Workflow 2, along with the design of the LevelEd VR software that enables this workflow. A study was devised and carried out to evaluate the workflow and LevelEd VR software and the results of this study are presented here along with a discussion.

### 5.1 Introduction and Motivation

Workflow 2 is designed to explore if and how developing VR game levels, whilst working in VR, can improve existing VR game level design workflows. The workflow was developed to allow level designers to design a blockout game level, test it and iterate on the design without leaving the VR level editor.

Before constructing any tools, it was important to understand how level design differs to more generic environmental design or world building. Level design is an area of game development that focuses on not only the construction of the level layout, but also the implementation of game play based on the game mechanics devised by the game designers (Kremers, 2009, p. 16). Level design therefore differs to world building or environment design (which has been researched before) in that level designers also construct game play within a space (Totten, 2014, p. 77). As previously noted, a common design technique used by level designers is to create prototype layouts called blockouts (see **Figure 17**, **Figure 18** and **Figure 19**) in order to ensure they retain the ability to quickly edit and iterate on designs (Weldon, 2008, p. 115; Pruett, 2014, p. 236). Gameplay is then constructed and tested within the blockout and confirmed before any final art is added (Grimsby, 2014, p. 198). Therefore, the tools must support not only the creation of the level layout, but also support the creation or implementation of existing game mechanics that can then be tested within the tool.

The current workflow for creating game levels for VR entertainment or serious games is to use a game engine such as Unity or Unreal Engine. Users construct a level in the complex editor on their 2D screen and then are required to test the game in a VR headset. Whilst working on a 2D screen works well for games that are also played on a 2D screen, there are key differences to games played in VR. Designers can no longer rely on traditional framing or compositional techniques to help with scale and attention (Silverman, 2017, p. 312). Binocular vision (supported by stereoscopic 3D HMDs) along with occlusion, oculomotor depth cues and linear perspective also differ (Jerald, 2016, p. 120-124). There are possible benefits and efficiencies to moving the VR level design workflow into VR. These include a better sense of scale and a better understanding of how a level will be perceived in stereoscopic 3D in a headset whilst building the prototype blockout level geometry and implementing gameplay.

There was also potential for improvements to the iteration of levels due to less time required to move between editing and testing.

## 5.2 Workflow 2

LevelEd VR was developed to explore Workflow 2 (**Figure 51**) and therefore how and if VR tools could be used to build game levels for VR games. It would also offer the ability to evaluate any benefits this workflow may have over a traditional workflow that uses a 2D screen. The primary target of this workflow is developers of VR games. However, user generated content has continued to gain in popularity with successful games such as, Minecraft, Little Big Planet, Fortnite and Roblox (Business Wire, 2020). Some game developers also use the same level editor released to users, to develop the game, such as Volume (Bithell, 2015) and Divinity: Original Sin (Larian Studios, 2017). Research also suggests an increase in modding activity when mods are officially supported by a company, such as with a level editor/SDK, which can lead to higher game sales (Poretski & Arazy, 2017). It would appear then that there are benefits to a game that includes a level editor to enable UGC. Therefore, an additional aim was to also allow developers to include the level editing tools in a game for players to create custom content. To enable this, the tool needed to be suitable for the task, but also intuitive to use. There would then be the potential to combine this tool with LevelEd AR to create SR game levels, but at this stage, that was not the primary focus.

To better understand Workflow 2, a generic runtime VR level editor that enables users to develop and design game levels in VR was created, called LevelEd VR. To meet the needs of a level designer the system needed to:

1. Enable the quick creation of prototype blockout layouts with robust geometry editing tools with accurate depth, scale, and perspectives for VR.
  - a. Primitive shape creation, such as Cube, Cylinder, Polygon, etc.
  - b. Common editing tools, such as Move, Copy, Rotate, etc.
2. Enable the ability to place and script gameplay elements to edit basic gameplay in the editor.
3. Enable the ability to quickly play test the level for a rapid and agile iterative workflow without leaving the VR level editor.

To ensure the system can also be used for user generated content by players and users it also needed to:

1. Be a runtime system that does not require a traditional desktop operating system so that it can run on standalone devices such as the Oculus Quest.
2. Not require prior knowledge of or make use of the game engine/editor interfaces.



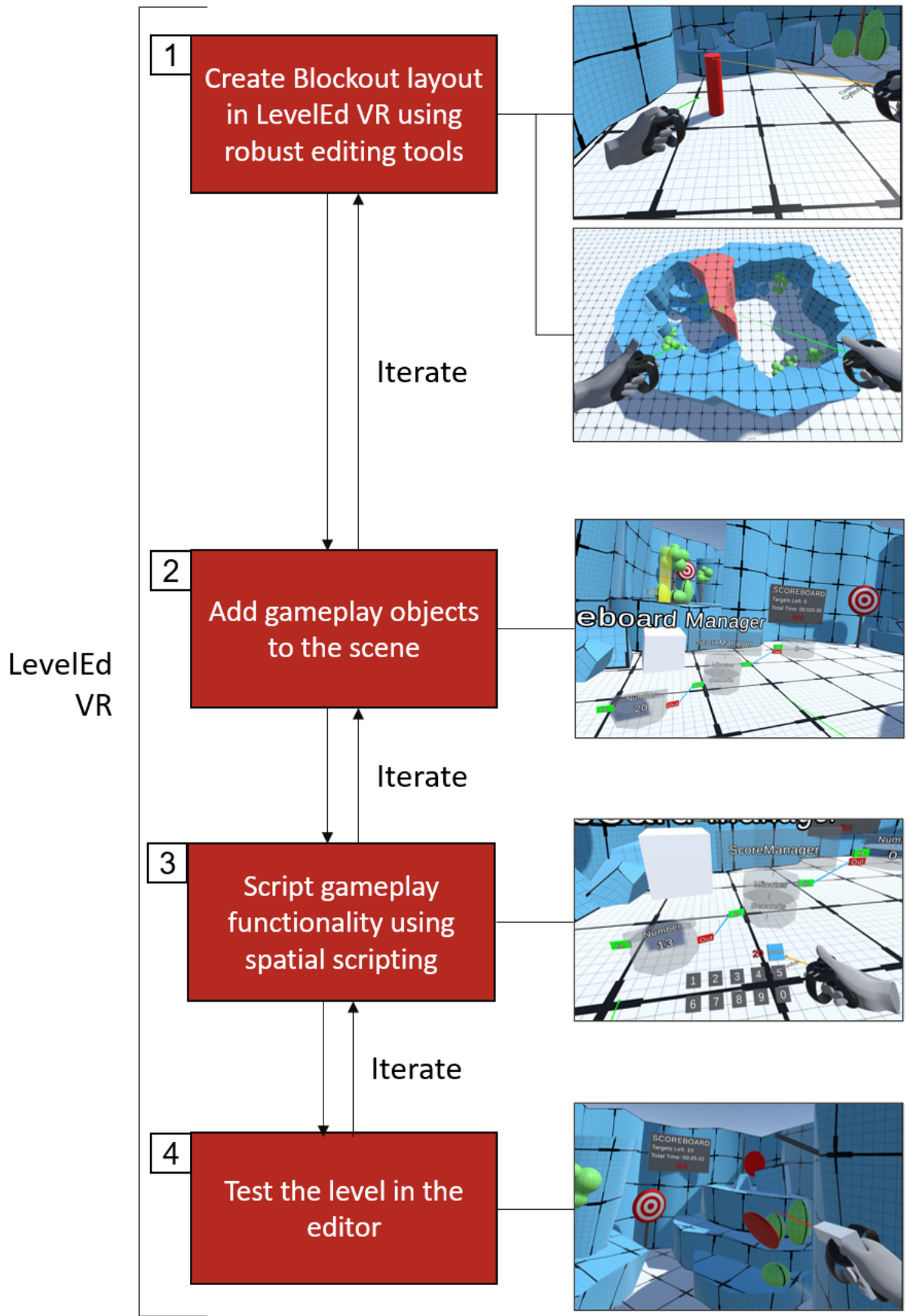


Figure 51 - Outline of Workflow 2 which supports the creation of VR game levels whilst in VR.



LevelEd VR would therefore be suitable for testing and evaluating Workflow 2 (**Figure 51**), where the level design blockout process is completed in the VR headset and tool. Two example videos of the system in use can be found at **Appendix B1** and **Appendix B2**. The Unity project can be accessed via **Appendix B3i** and **Appendix B3ii**.

### 5.3 Hypotheses

The following three hypotheses were devised to address the ability of Workflow 2, which uses VR technology, to improve an existing workflow for developers:

*H2: A virtual reality-based level design workflow is accepted (based on the System Usability Scale) as an alternative for developing VR game levels by experienced participants.*

*H3: A virtual reality-based level design workflow is preferred (based on a comparative questionnaire) as an alternative for developing VR game levels by experienced participants.*

*H4: A virtual reality-based level design workflow is shown to deliver an improved sense of scale awareness (based on a comparative questionnaire).*

The study outlined in section **5.6** was devised to test all three hypotheses, with the results presented and discussed later in this chapter.

### 5.4 System Overview and Implementation

It is clear from the previous analysis of the literature that there is a lack of research focusing on exploring a full level design workflow and editing in VR, that encompasses gameplay rather than solely world building. As noted in **2.4.2**, level design encompasses not just world building (creating a space), but also the creation of gameplay within a space. The level designer must consider both the gameplay and the level layout together, and so need the ability to implement and test both whilst designing a level. To advance this area of research the contributions of this chapter and Workflow 2 are as follows:

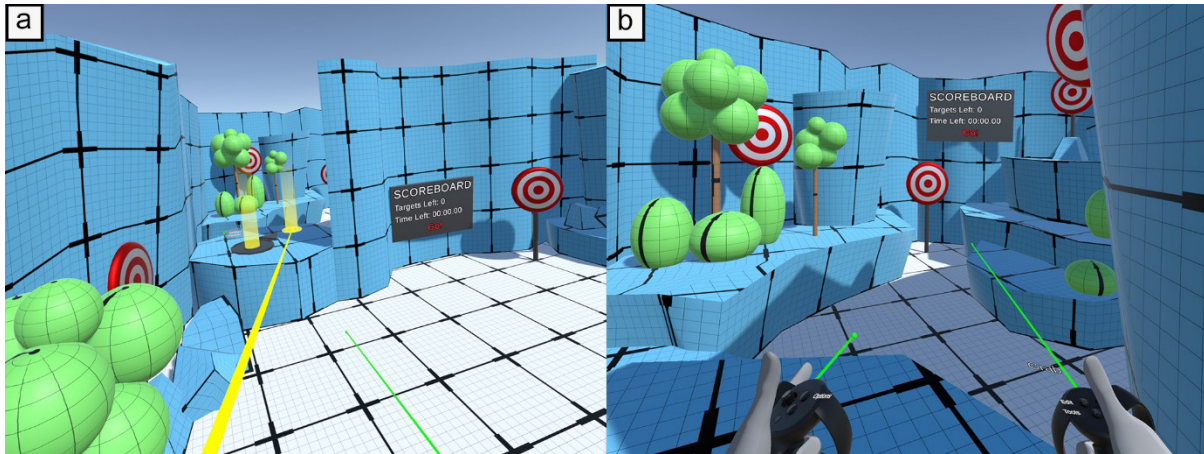
1. The design and implementation of a prototype generic runtime VR level editor called LevelEd VR.
2. An evaluation of the system by participants with prior game development and Unity engine experience.
3. A comparison to an existing level design workflow that utilises Unity on desktop by participants with prior experience.
4. Best practices and lessons learnt from the development and testing of the system.

LevelEd VR allows users to create levels for VR games whilst in VR. It has been built as a runtime editor so that it can run on devices, free from the Windows operating system or the Unity Editor. The version used in the study focuses on the blockout process where levels created are serialized to a file and regenerated at runtime. Users can also choose to regenerate the level in the Unity Editor to continue traditional development. Support for developing the full level in LevelEd VR or even bundling LevelEd VR with their game to support user generated content was only available in a limited capacity at the time of the study. LevelEd VR was designed from the ground up for VR and spatial computing, avoiding traditional 2D windows or interface paradigms to make the most of 6-DoF interactions and the spatial nature of VR.

LevelEd VR was built with Unity and supports games built in this engine. Unity (version 2018.2.21f1) was selected due to the popularity of the system for entertainment games, Riccitiello estimates between 60-70% of games on VR and AR platforms are built with Unity (Dillet, 2018). It is also a popular engine for serious games and research with a review of immersive serious games papers showing that 50% were built with Unity, with Unreal Engine (the next closest engine) being used for around 5% (Checa & Bustillo, 2019). VRTK (2022a) was used for some components, such as input detection to allow for easy porting between devices. LevelEd VR ran on the Oculus Rift (CV1) with Touch controllers for the evaluation study described later in section 4.3. WebXR was not considered at this time as few commercial games are built for WebXR and this would have limited the potential pool of games that could utilise LevelEd VR.

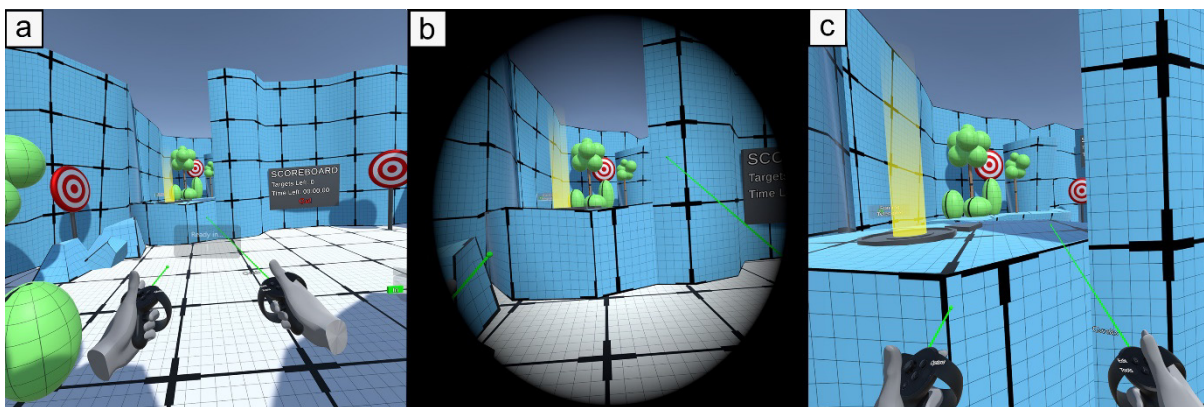
#### **5.4.1 Locomotion**

One key design choice for a creation tool like LevelEd VR, is how users can move around the level blockout they are creating. This is commonly referred to as locomotion. Many recent VR games and applications, such as Half-Life: Alyx (Lang, 2020) offer multiple options for locomotion so that the user can decide for themselves based on their preference and comfort levels (Bolton, 2020). This was considered with LevelEd VR and so three forms of locomotion are supported, and each have been designed to support different use cases and comfort levels. The first method is teleporting; this allows users to instantly move from one location to another and is aimed at moving over large distances (see **Figure 52**). Level blockouts can become quite large and therefore it was important to allow users to quickly move around the level as needed. This method of locomotion has also been shown to reduce the potential for simulator sickness compared to full movement techniques (Líndal et al., 2018; Frommel et al., 2017).



*Figure 52 - (a) User holds the teleport button and aims where they want to go. (b) User releases teleport button and they are then moved to their destination.*

The second is the Drag World (VRTK, 2022b) method where users can grip their controllers to move or rotate (but not scale) the world around them (see **Figure 53**) and is sometimes referred to as Scene-in-Hand locomotion (Martinez et al., 2022). This allows for precise movement over short and medium distances. A similar technique was shown to cause simulator sickness over large distances (Coomer et al, 2018). To help alleviate this potential simulator sickness which can be caused by both movement and rotation (Chang et al., 2020), a vignette is used to limit peripheral vision based on velocity and angular velocity (Buhler, 2018; Fernandes & Feiner, 2016).



*Figure 53 - (a) User reaches out in the direction they want to move. (b) User grips the bottom controller triggers and pulls the world to them. A vignette is used to ease simulator sickness. (c) The user lets go of the triggers when they get to their destination.*

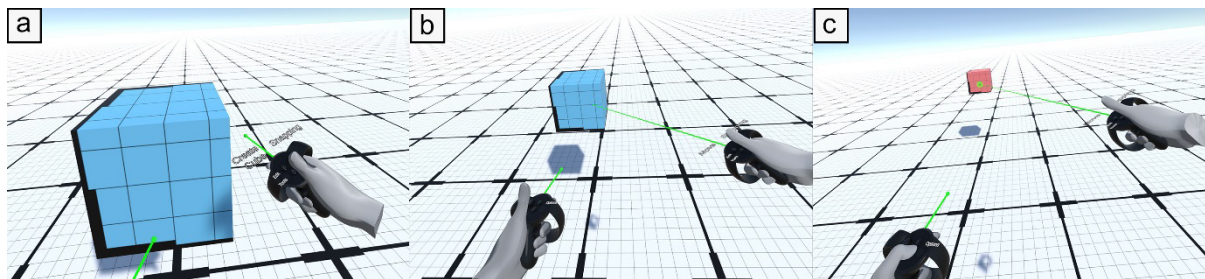
Finally, within physical space limitations, users can move and turn in the real world for small distances or to quickly look around. This was enabled by the room-scale tracking of the Oculus Rift CV1 and would potentially help reduce potential simulator sickness that is often brought on by the mismatch of movement (in the virtual world vs the real world) that is seen in other locomotion techniques (Chang et al., 2020). It was anticipated that users would use a mix of

the different techniques throughout the development of a level, rather than picking and using only one.

More traditional locomotion techniques could have been used, such as moving the user with the left analogue stick and rotating with the right analogue stick. However, this technique was not incorporated into LevelEd VR. At many points during development there may not be a concept of the 'ground' and so moving with an analogue stick would not make as much sense here and would also require additional buttons to deal with moving up and down in 3D space. The use of these additional buttons as well as the analogue sticks would reduce the number of input options available for interaction in the editor. It was therefore felt that the previous three locomotion techniques were better suited to this type of system.

### 5.4.2 Menu and Interaction

An early decision in the design of the system was to utilise the spatial nature of VR, rather than relying on existing 2D interface paradigms, such as 2D lists or windows. In LevelEd VR, interaction is supported by a variable pointer beam. Users select objects by pointing at them and can create objects at the tip of the beam. The length can be changed to fixed points (short, medium or infinite) with an analogue stick click or the length fine-tuned with up and down movements on the analogue stick (**Figure 54**). A beam solution was utilised as it would allow the user to work at larger distances, compared to direct hand interaction.



**Figure 54** - The variable pointer system in use. (a) Short fixed-point set to 10cm. Good for working up close. (b) Medium fixed-point set to 1m. Good for working at arm's length. (c) Infinite beam length will continue until it hits an object. Great for working at large distances or for selecting something at a distance and bringing it closer.

The menu appears physically in the virtual world and is interacted with by either pointing at or by intersecting with the beam. It is radial in shape and appears anywhere the controller is currently located when accessed (**Figure 55b** and **Figure 55c**). As the menu appears physically in the world the layout can be customised (in Unity) by the user/developer before they use the tool (**Figure 55a**).

The menu system has also been developed to allow developers to add their own custom tools without having to edit the existing editor menu code or tool code (see **Appendix B3i**).



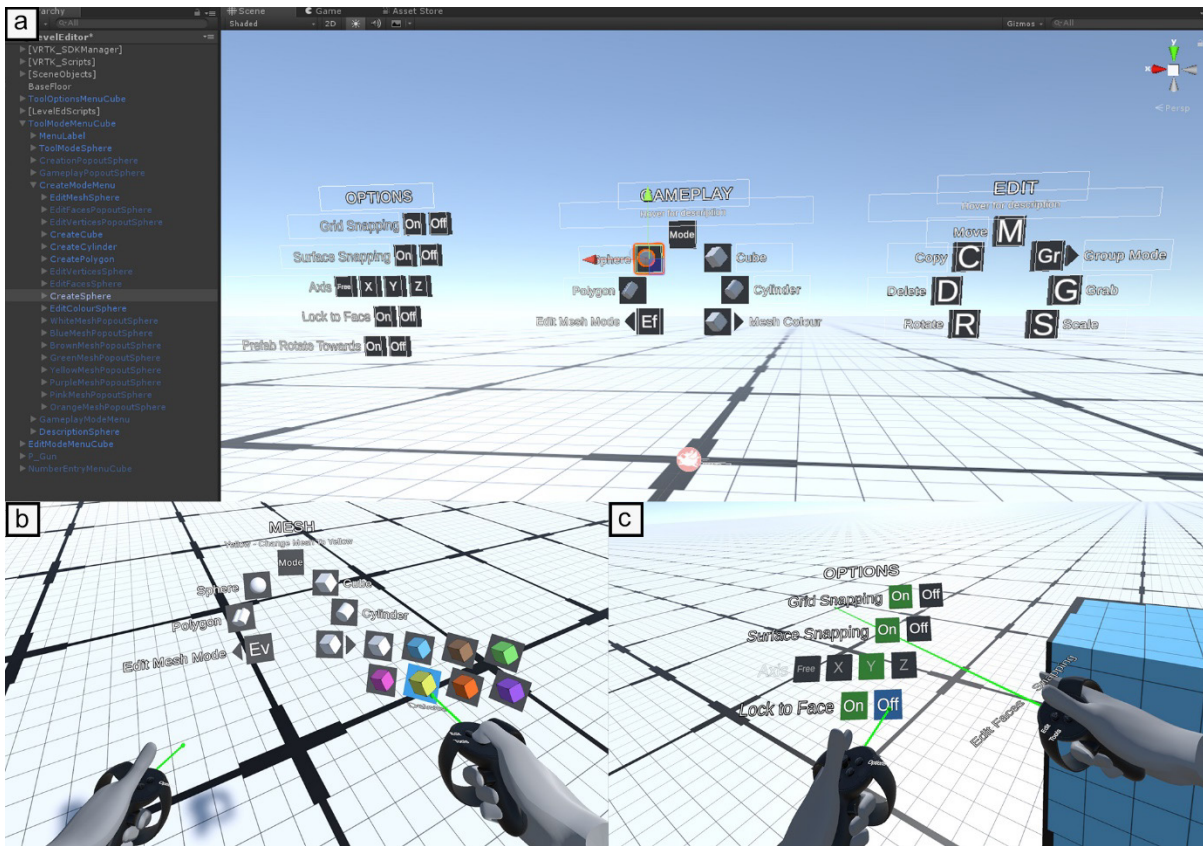


Figure 55 - (a) The options, tools and edit menus as seen in the Unity editor. (b) The Mesh menu with the colour options pop out visible as viewed in VR. (3) The options menu as viewed in VR. The Axis option is greyed out as it is not relevant to the currently selected Edit Faces tool.

To do so, they need to:

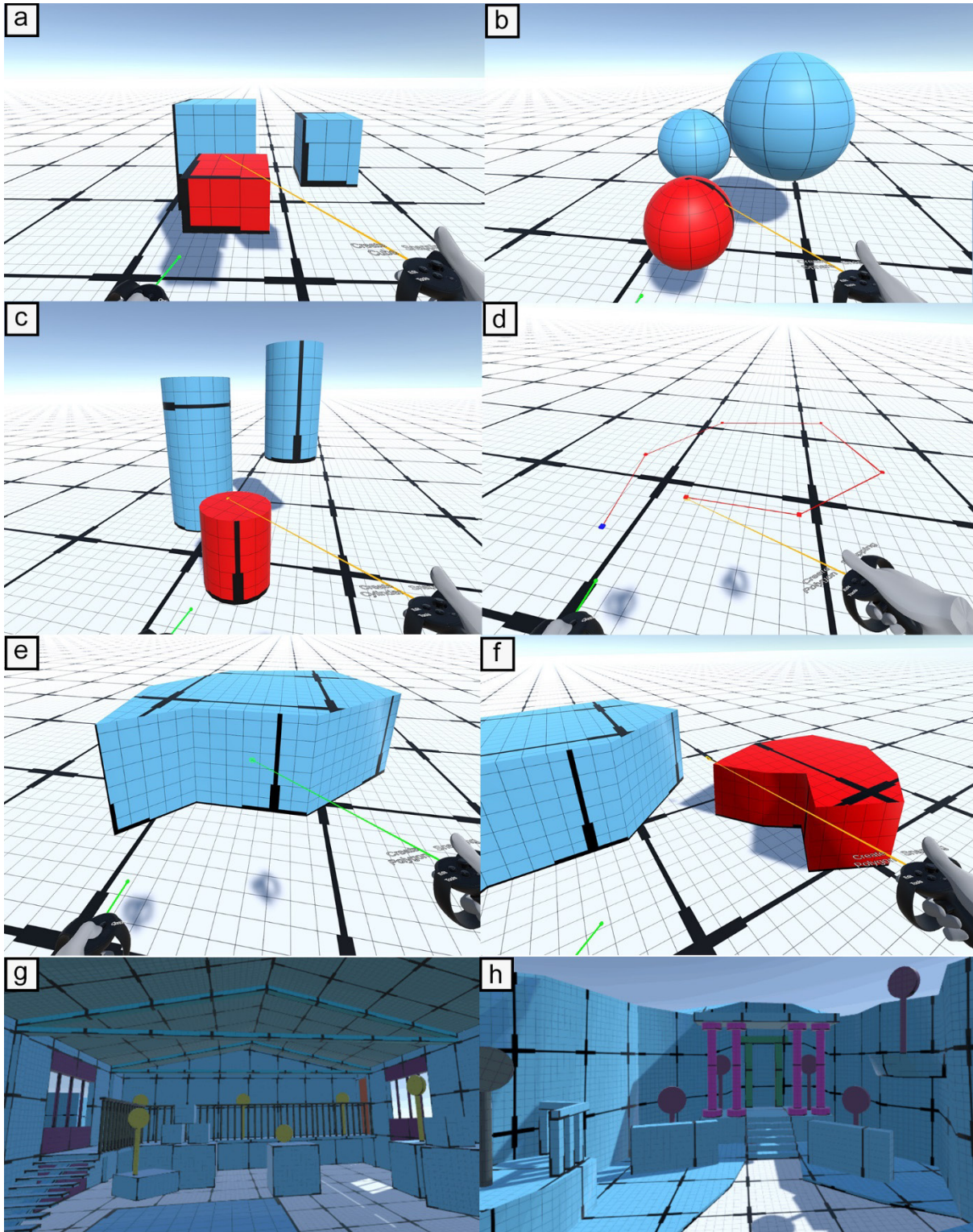
- Write the new tool code following the common tool template based on existing tools (see **Appendix O**)
- Add in a new menu node based on existing nodes as a template (see **Figure 55a**).

The user can then select the new tool in the menu and use the new tool in the VR level editor.

### 5.4.3 Level Editing Tools

Several different editing tools were implemented to enable users to create their level layouts. These focused on creating and editing the layout, adding prefabs (existing models or objects with game mechanics) and a basic scripting system. **Table 6** summarises the robust set of tools featured in LevelEd VR for the study and a demo video can be found at **Appendix B1 and 2**, but the main features are detailed below:

**Geometry/Layout:** These tools enable a level designer to create procedural static meshes to blockout a level layout. They can create meshes, such as cubes, spheres, cylinders and custom polygons (see **Figure 56a-f**). Sample layouts created using these tools can be seen in **Figure 56g-h**.

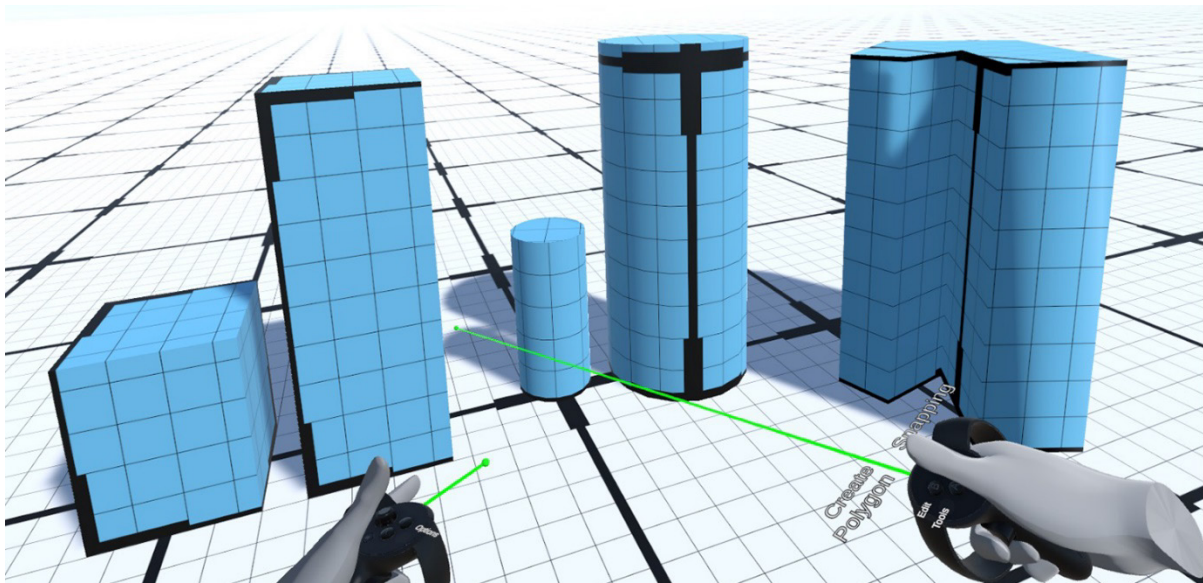


**Figure 56** - (a) Cube creation tool. (b) Sphere creation tool. (c) Cylinder creation tool. (d) Drawing out initial custom shape with polygon creation tool. (e). Finished custom polygon shape. (f) Another custom polygon shape. (g) A warehouse style level blockout created using the tools. (h) A fantasy style level blockout created using the tools.

The procedural static meshes are created with accurate UV layouts so that textures will tile correctly with a mesh. The texture used to prototype marks out 10x10cm squares as

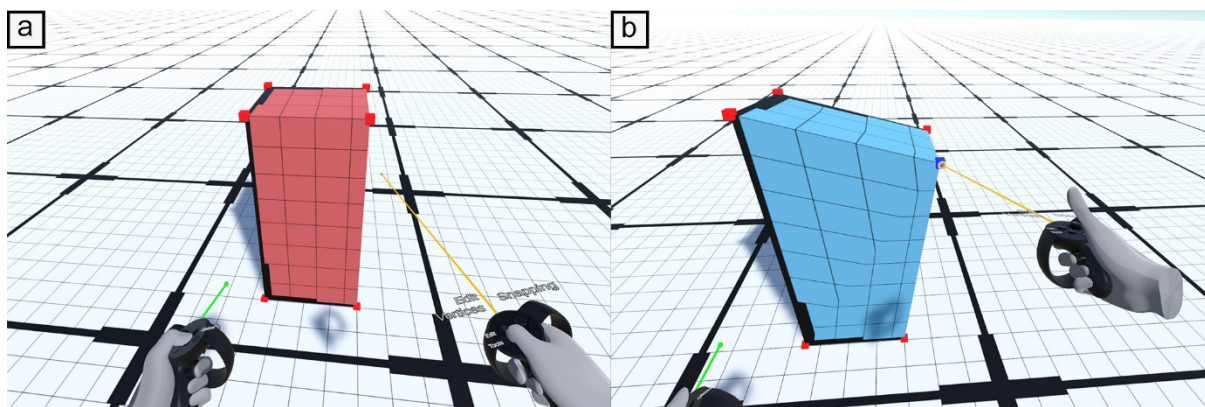


subsections of larger 1x1m squares (See **Figure 57**). Using accurate tiling ensures users can create consistently sized assets.



*Figure 57 - Textures tile correctly across the shapes as they are created. This ensures each sub-square remains 10x10cm.*

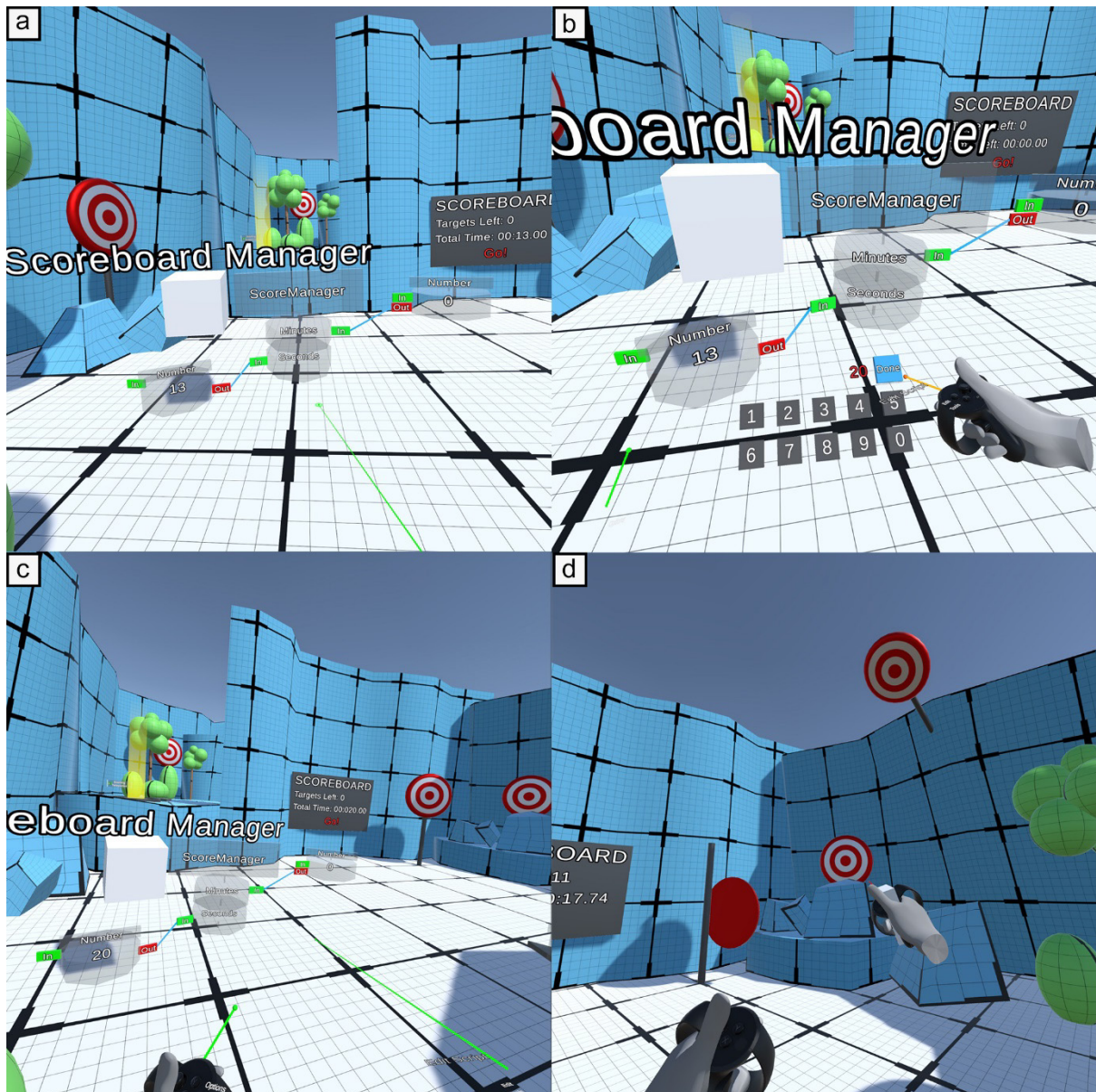
The user can then edit vertices/faces of a static mesh to create more unusual shapes and change its colour (see **Figure 58** and **Figure 60**).



*Figure 58 - (a) User has selected Vertex Edit mode and vertices are made visible. (b) The user has selected some vertices and moved them to make a more unique shape.*

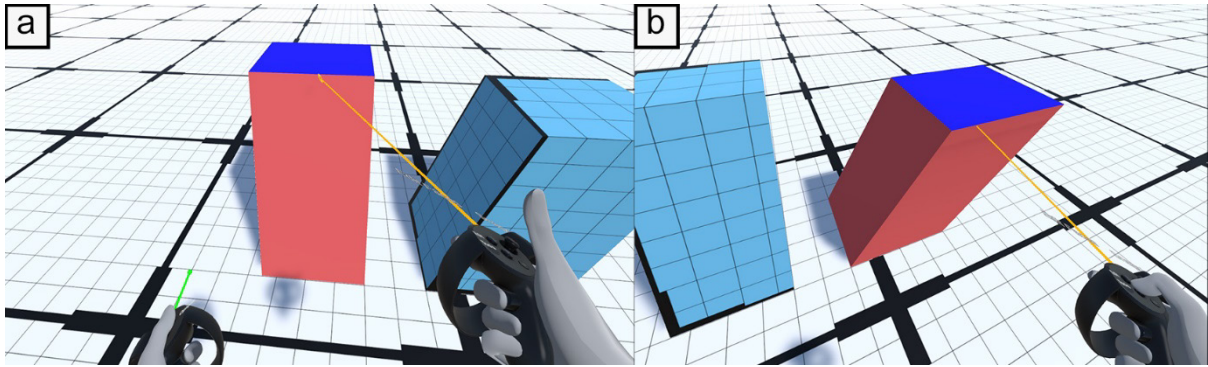
**Scripting/Prefabs:** Users can add prefabs (models, interactive objects, etc) to the level. They can also add spatial scripting nodes directly into the level. The spatial scripting system utilises a flow-based scripting concept that connects nodes to one another to define the flow of data (**Figure 59**). Building on the spatial nature of VR it was decided to put them in the level so that users benefit from the extra spatial dimension, such as the relationship of a node to an object in the level (proximity and line connection). They can edit the values and node flow with the Edit Script tool.





*Figure 59 - (a) Spatial scripting number nodes attached to the ScoreManager to control time limit. (b) User edits the number node to change the seconds from 13 to 20. (c) The change in seconds to complete the level is visible on the scoreboard. (d) User can quickly test the level to see if it is enough or too much time.*

**Editing/Options:** Users can make use of several editing tools (see **Table 6 - Editing**) which can be used on any instance of an object in the level. These feature common editing tools that are found in most creation tools, such as copy, move, rotate, delete, etc. There are also several options available to users. For example, they can choose to snap existing assets to a grid as they are moved or create new static meshes on a grid. This is useful when trying to create consistently sized components in the layout. They can also choose to lock or not to lock the editing of a new static mesh to a normal direction (see **Figure 60**). For example, when editing the face of a cube, you may only want to edit along the current normal axis to lengthen the cube, but not change its shape.



**Figure 60** - (a) Normal lock on. Current face can only move on the current normal axis (up and down). (b) Normal lock off. The face can move on all axes.

Surface snapping makes it easier to snap the pointer to a surface, which is useful when creating new static meshes, but also when editing objects. Sometimes this needs to be disabled so meshes can be placed inside one another.

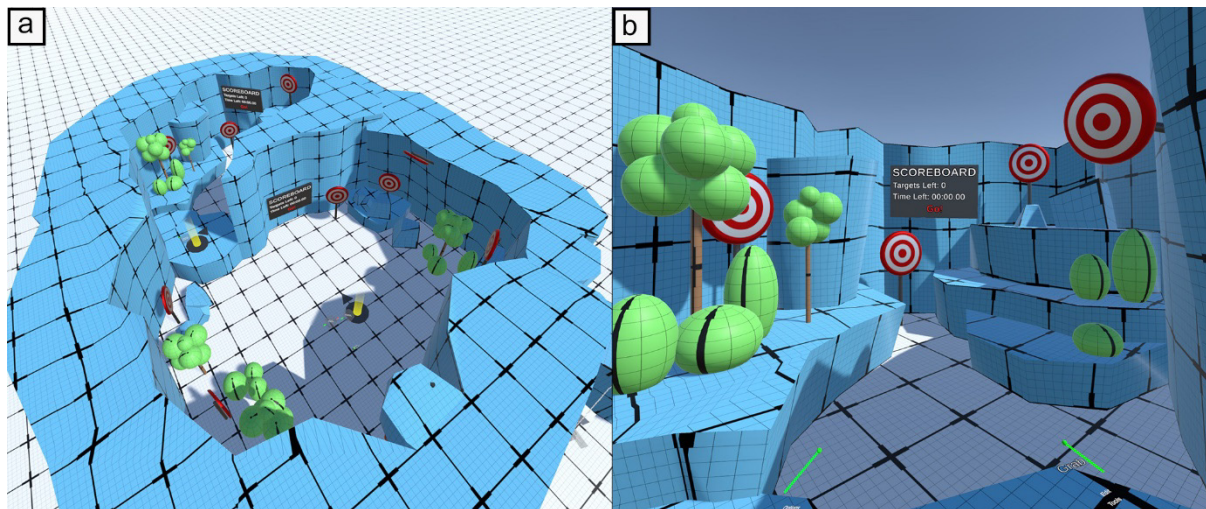
**Table 6** - LevelEd VR Features for Study

<b>Geometry</b>	<b>Script/Prefab</b>	<b>Editing</b>	<b>Options/Modes</b>
Create Cube	Number Node	Move	Enter Play Mode
Create Sphere	Bool Node	Copy	Exit Play Mode
Create Cylinder	Edit Scripts	Delete	Grid Snapping
Create Polygon	Add Prefabs	Rotate	Current Axis
Edit Vertices	Add Models	Scale	Normal Lock
Edit Faces		Grab	Surface Snapping
Edit Colour		Group/Ungroup	

#### 5.4.4 Sample Game

A sample game was created to appropriately test the VR level design workflow from geometry creation to gameplay tweaking/scripting, and then play testing. The game took the form of a gallery shooting game (see **Figure 61**). Players use a gun (tracked by Oculus Touch controllers) to shoot the targets. The player must shoot all the targets within a time limit (displayed on the scoreboard), they can move around the level via fixed teleporter nodes.





*Figure 61 - (a) Overview of the sample level created using the tools for the study. (b) View from inside the sample level used in the study.*

Level designers can control the geometry/layout, the location and number of targets, scoreboards, and teleporters. They can also select a starting teleporter and set the time limit for the level using the spatial scripting system.

#### **5.4.5 Play Testing**

Users can choose to play test their level at any time by pressing in the two analogue sticks to toggle between edit mode and play testing mode. This ensures users can quickly test the level and continue editing without delay. This is to encourage iterative changes that can be made quickly and effectively by allowing users to move between modes instantaneously (**Figure 59d**).

#### **5.4.6 Saving and Loading**

After a user has completed their level or finished editing a level in a session, they can close the application which in turn saves the level to a file. As this is a runtime editor, when the user opens the application, they first load the level from the file, and they can then continue to edit the level.

### **5.5 Preliminary Study**

There were two studies carried out on LevelEd VR and Workflow 2. The first was a preliminary study aimed at getting early feedback on the system, the second was a thorough evaluation and comparative study.

The aim of the preliminary study was to get early feedback on the system to ensure development was going in the right direction and to get direct feedback on the current features as well as additional features or changes that would improve the system. Two participants were recruited to this early study. Both participants worked in the games industry and had

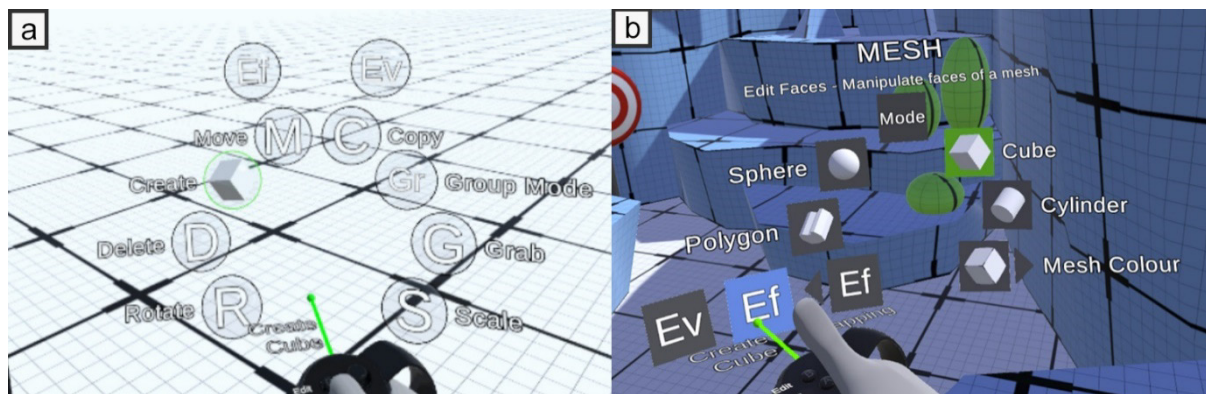
experience of using Unity (an existing tool for game development and level design) and of the level design discipline and workflows. This study took place in July 2019 and ethical approval was granted by the FSE Research Ethics Committee (see **Appendix H**).

Participants were introduced to the system with a brief tutorial, they were then given time to use the tools and attempt to create a level for the sample game outlined in section 4.1.4. They then completed a general feedback questionnaire about the system. The consensus from the participants was that the system was intuitive and would benefit level designers by helping with scale, perspective and quick iterative changes to a game level during prototyping. There were no serious issues highlighted with the study.

The participants noted a few minor issues and feature requests for example:

- The ability to snap back down to the floor.
- The ability to teleport between locations.
- Change the menu to be opaquer and to use icons where possible, rather than letters.

The ability to snap to the floor was implemented through a new teleport system outlined in section 4.1.1. The menu used in this preliminary study used translucent spheres and text (see **Figure 62a**). This was changed based on the feedback to the current version as outlined in section 4.1.2 and **Figure 62b**. This had the benefit of being more legible, but also being more optimised by not using transparency and a reduction in triangles by moving to cubes from spheres. The menu was also split to separate out Edit and Creation tools.



*Figure 62 – (a) Previous menu prototype used in preliminary study. (b) New menu used for the main study.*

This was an especially useful study to carry out as it confirmed that the system design was going in the right direction and that development could continue. The system was improved in a number of ways based directly on feedback from the participants.

## 5.6 Study 2 - Experiment Methodology

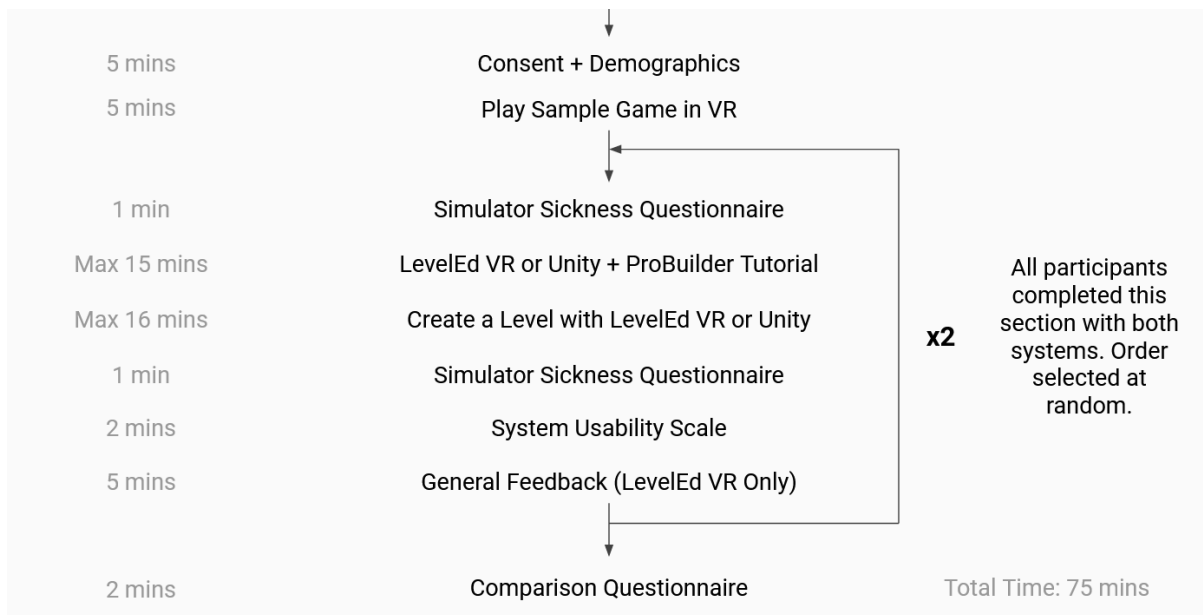
The main study (Study 2) was designed to evaluate the LevelEd VR system for usability and effectiveness at the task of enabling level design in VR. It focused on the blocking out of layout and gameplay by level designers for the sample game. The study was then extended to compare the LevelEd VR system and Workflow 2 against current industry workflows using Unity for a desktop system. In the latter case, the users design the level on the desktop system but use the VR headset when testing the level. The study was carried out in January 2020 (see **Figure 63**) and ethical approval was granted by FSE Research Ethics Committee (see **Appendix I**).

### 5.6.1 Experiment Structure

Once consent and demographic information were obtained, participants began by spending five minutes playing the VR target shooting sample game to ensure they understood the aim of the game and the potential mechanics available to them. Participants were then asked to complete two tasks consisting of creating a game level for the sample game with the LevelEd VR system (Workflow 2) and another level using Unity on desktop with the aid of the ProBuilder package (referred henceforth as Unity Desktop). Participants were told they needed to create a game level that included all the required components for the sample game, such as:

- several targets.
- at least one teleporter to act as the initial spawn location.
- at least one scoreboard.
- the scoreboard manager so the game can function.
- number nodes to dictate the time limit for the level.

The order in which they completed either task was randomised, with eight completing the task with LevelEd VR first and another eight completing the task with Unity Desktop first. Before each task, participants were given a maximum of fifteen minutes' tutorial on the system. They were then given sixteen minutes to complete the task. There were no design requirements for the level (except that it should include the components above to ensure it is playable) and they could choose to create a different level using each system.



**Figure 63** - Structure and flow of Study 2

To help compare levels of simulator sickness between each system, participants completed a Simulator Sickness Questionnaire (SSQ) (Kennedy et al., 1993) before each task. During each task participants' screens were recorded and video captured of them completing the task. After each task they completed the SSQ again (for post results) and a System Usability Scale questionnaire (SUS) (Brooke, 1996). After the LevelEd VR task, they were also issued a general questionnaire for feedback on the system including 7-point Likert scale questions and some open questions (see **Table 8**). Finally, after both tasks were completed, participants filled out a comparative questionnaire that asked them to select which system (or either) they preferred, or they felt better demonstrated the given statements.

### 5.6.2 Participants

Participants were selected through homogeneous convenience sampling and consisted of students and staff in the Department of Computer Science at the University of Chester. Participants were recruited via an email sent out to staff and students in the department (see **Appendix L**). As this study aimed to explore how the LevelEd VR tool could improve the VR level design workflow for developers, participants were required to have prior game development and Unity game engine experience. This ensured that the participants had sufficient prior experience to effectively evaluate both systems. Whilst convenience sampling limits generalisability compared to probability sampling. Homogeneous convenience sampling is considered by some to offer improved generalisability (compared to conventional convenience sampling) as it is likely to represent a narrower population more accurately (Jager et al., 2017).

A total of 16 participants took part in the study. Participants were split with 75% identifying as male and 25% identifying as female. Participants all had prior game development and Unity experience, with all but one having between 1.5-2.5 years' experience. 94% of the participants were between the age of 18-29, with the remaining 6% aged between 50-59. 69% of participants stated they had little or no experience with VR with the rest stating intermediate experience.

The demographic of this convenience sample demonstrates an imbalance in gender (more male identifying participants) and age (participants skewed to the younger age group). However, this is closer to the demographic makeup of the games development industry in the UK, which differs to the wider workforce. A census carried out by UKIE (2022) reveals that 30% of the games industry are female and 67% are male, compared to 48% of the general workforce being female. For age, the census shows that 61% of the games industry are aged 35 or younger, compared to the overall workforce which is 33%. So, whilst the participant sample is skewed away from the general workforce, it is closer to the games industry in the UK. This still limits the ability for the results to be generalised to the wider population and therefore should be viewed through the lens of being comparative to similar populations of computer science staff and students, with game development experience and those working in the games industry.

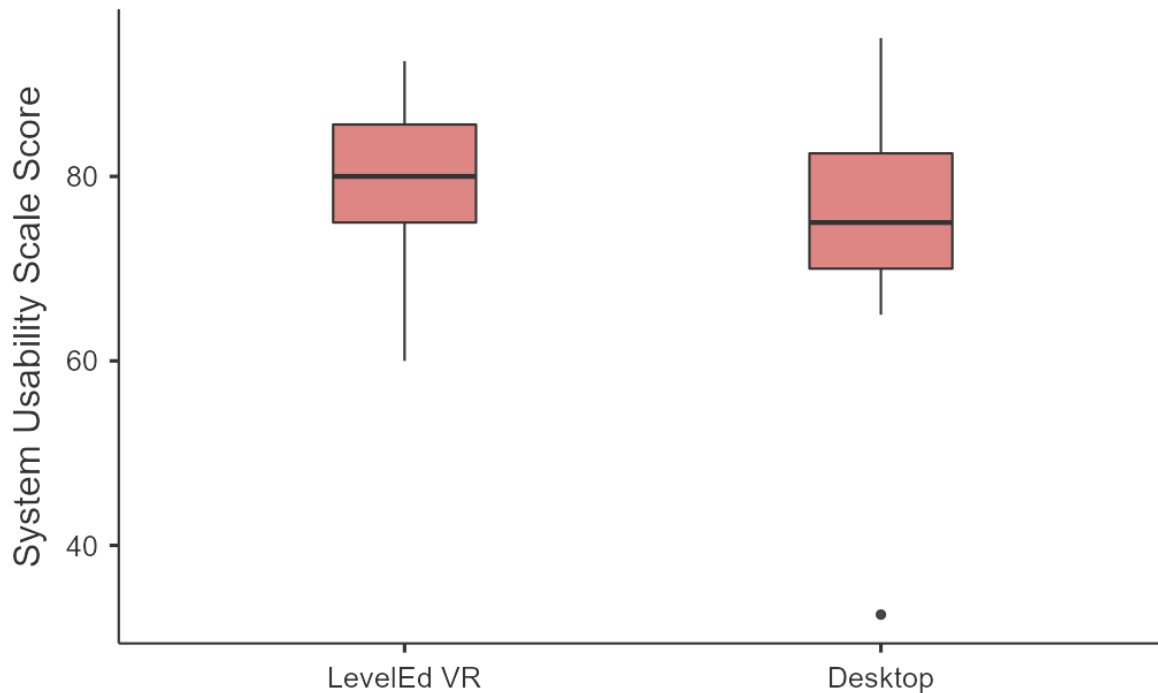
## **5.7 Results**

### **5.7.1 System Usability Scale (SUS)**

The SUS questionnaire was used to gather an acceptability rating for LevelEd VR and to also compare it to that of the existing Unity Desktop workflow. It consists of ten questions, each with five response options. Each completed questionnaire is converted into a score out of 100, where any score of 68 or more is considered above average (Brooke, 1996). The results (summarised in **Figure 64**) show that LevelEd VR was accepted by the majority of participants with a mean SUS score of 79 and standard deviation of 9. Only two of the 16 participants gave it a score below 68 with a minimum score of 60 and a maximum score of 93. According to Bangor et al. (2009), the mean score of 79 results in an adjective rating of Good for the system.

When compared to the existing Unity Desktop workflow, LevelEd VR compares favourably. Unity Desktop scored a mean of 75, a standard deviation of 15, minimum score was 33, and the maximum score was 95. Unity Desktop is also considered an acceptable system (as the mean exceeds the 68-score threshold) and has an adjective rating of Good.





**Figure 64** – Results from the System Usability Scale questionnaire. Negative outlier marked with a dot and median with horizontal line (Brooke, 1996).

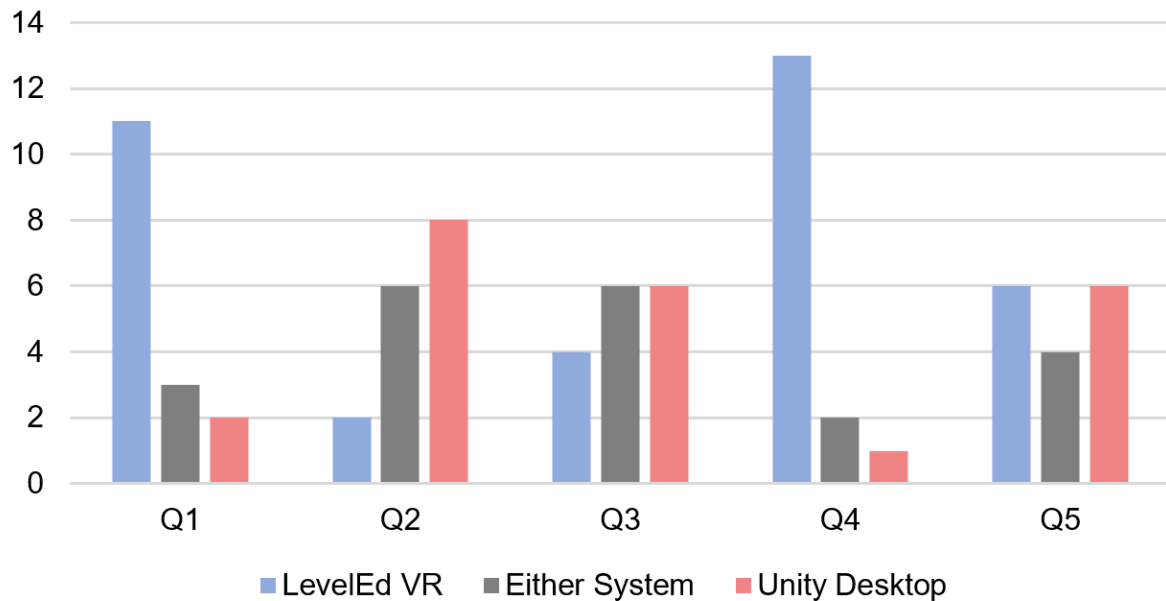
A t-test does not show any statistical significance between the two sets of results (p-value = 0.3533). However, the mean and boxplots (**Figure 64**) show a slightly higher usability score for LevelEd VR when compared to the Unity Desktop.

### 5.7.2 Comparisons of Systems and Workflows

A comparative questionnaire was also used to ascertain participants' preferences or experiences of each system (see **Table 7**) with the results presented in **Figure 65**. The results show that 68.75% of participants preferred to use LevelEd VR when building game levels for a VR game (Q1), with 18.75% preferring either system and only 12.50% preferring the Unity Desktop. However, when asked which system they preferred to use for creating game levels for games other than VR (Q2) 50.00% of participants preferred to use Unity Desktop, with 37.50% preferring either system and only 12.50% preferring to use LevelEd VR. This suggests the benefits of working in VR are more apparent for VR games than non-VR games.

When asked about adding scripted gameplay to a VR game level (Q3) the results were 37.50% preferring Unity Desktop, 37.50% preferring either system and 25% preferring LevelEd VR. With Q4, 81.25% of participants overwhelmingly found they were able to create game levels with a good sense of scale in LevelEd VR, with 12.50% choosing either system and only 6.25% selecting Unity Desktop. Finally, participants answered evenly when asked which workflow they found more efficient for developing VR game levels, with 37.50% preferring both LevelEd VR and Unity Desktop and 25.00% finding either system to be efficient.

## Comparative Questionnaire Results



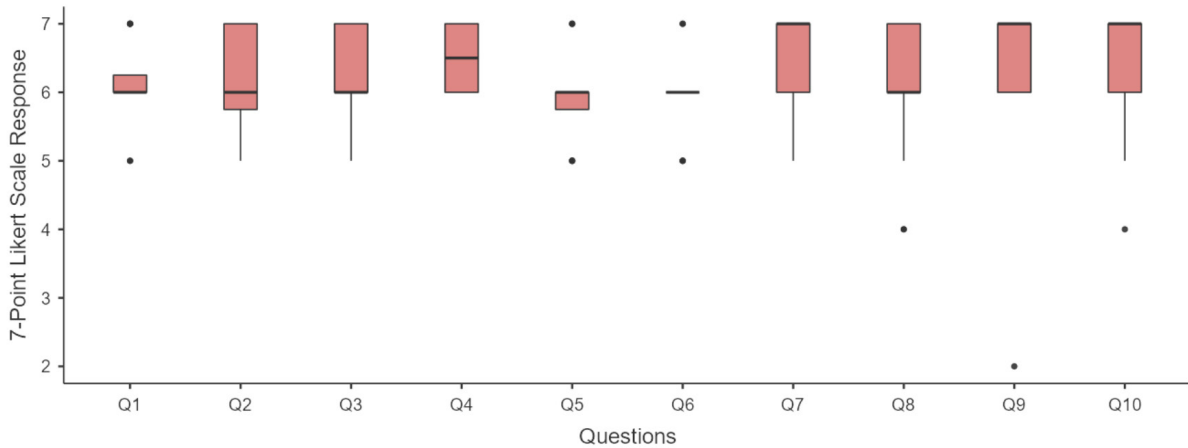
*Figure 65 – Comparative questionnaire results based on questions in Table 7.*

*Table 7 - Comparative Questionnaire Questions*

	<b>Questions</b>
<b>Q1</b>	When creating a level blockout for a virtual reality game I would prefer to use the following system
<b>Q2</b>	When creating a level blockout for any type of game I would prefer to use the following system
<b>Q3</b>	When adding scripted game play to a virtual reality game level I would prefer to use the following system
<b>Q4</b>	I found creating virtual reality game levels with a good sense of scale easier in the following system
<b>Q5</b>	I found my workflow was quicker/more efficient for developing virtual reality game levels in the following system

### 5.7.3 LevelEd VR Feedback

To better understand participants' experiences with LevelEd VR a feedback questionnaire (**Table 8**) was given to participants after completing the LevelEd VR task with the results



**Figure 66** – Responses to the general feedback questionnaire (**Table 8**) for LevelEd VR. A 7-point Likert scale response was used for each question. Outliers marked with a plus and median marked with horizontal line.

**Table 8** - General Questionnaire Questions for LevelEd VR

	<b>Questions</b>
<b>Q1</b>	I found the locomotion methods easy to use once trained.
<b>Q2</b>	I was able to navigate the environment successfully to create a level.
<b>Q3</b>	The menu system was easy to understand.
<b>Q4</b>	The menu system allowed for quick changes between tools.
<b>Q5</b>	I found the prototyping/geometry tools easy to use.
<b>Q6</b>	The prototyping/geometry tools were sufficient for me to build the level I had envisioned.
<b>Q7</b>	I found the scripting system easy to use.
<b>Q8</b>	I found having the scripting system in the scene helped with spatial awareness of what script nodes will impact specific objects in the scene.
<b>Q9</b>	I found working in VR helped with gauging the scale or positioning of objects in the level.
<b>Q10</b>	Creating and play testing in VR allowed for quick iteration of the level.

shown in **Figure 66**. A 7-point Likert scale was used ranging from *strongly disagree* (1) to *strongly agree* (7). The questionnaire consisted of 10 items and the value for Cronbach’s Alpha for the questionnaire was  $\alpha = .745$ . According to George and Mallery (2003), this is considered an *Acceptable* level of reliability.

Responses to Q1 and Q2 show that participants were confident in using the locomotion system and were able to effectively navigate the level to construct their game level. This was a good result as participants were unfamiliar with the drag world mechanic. Q3 and Q4 related to the

menu system and again, it is clear that users found the menu system easy to understand and that it allowed them to quickly change between tools. Q5 and Q6 focused on the prototyping and geometry tools. The results show that whilst participants found them easy to use in general with eleven participants selecting agree, some participants felt they could have been easier to use with four participants choosing somewhat agree and only one participant selecting strongly agree.

Q7 and Q8 both focused on the spatial scripting system. Despite the system having the potential to be challenging to learn, participants appear to have understood it and found it easy to use, with more people choosing strongly agree (9) than those selecting agree (4) and somewhat agree (3). Participants also found having the scripting system in the 3D space helped with understanding the relationship of scripting nodes to objects in the space. However, there were two participants (marked as outliers) who neither agreed nor disagreed on this benefit.

Finally, Q9 and Q10 focused on the overall benefits of the system. All participants except for one (marked as an outlier) positively agreed (eleven strongly agreed and four agreed) that they found working in VR helped with gauging the scale or positioning of objects in the level. One explanation for the outlier who disagreed with the statement is that they were unable to use their glasses with the Oculus Rift headset and this may have impeded their vision and therefore depth perception. This was not an issue for other glasses wearing participants as they could either use their glasses with the Oculus Rift or had a sufficient level of vision without. The final question suggests that participants were able to quickly iterate on their level using LevelEd VR. The majority of participants strongly agreed (10) or agreed (4) with one participant agreeing somewhat and one neither agreeing nor disagreeing with the statement.

#### **5.7.4 Simulator Sickness Questionnaire Results**

To gauge if there were any simulator sickness effects resulting from using either LevelEd VR or Unity Desktop, the Simulator Sickness Questionnaire devised by Kennedy et al. (1993) was administered. Reference results for the different degrees of symptoms and the score ranges for each of the three categories and total score can be found in **Table 9. Table 10-Table 13** include the pre and post results for each system.

Both systems/workflows demonstrated some levels of simulator sickness. In all categories Unity Desktop had less of an effect on simulator sickness for participants than LevelEd VR. With Unity Desktop, there were five participants who suffered some slight symptoms. Common symptoms for Unity Desktop were sweating (reported by three participants), whilst two participants reported symptoms of fatigue, eye strain and difficulty concentrating. No participants suffered from any symptoms on a moderate or severe level.

*Table 9 - Reference SSQ-Evaluation Findings*

	<b>Nausea</b>	<b>Oculomotor</b>	<b>Disorientation</b>	<b>Total Score</b>
<b>None</b>	0	0	0	0
<b>Slight</b>	66.8	53.1	97.4	78.5
<b>Moderate</b>	133.6	106.1	194.9	157.1
<b>Severe</b>	200.3	159.2	292.3	235.6

*Table 10 - LevelEd VR Pre SSQ-Evaluation Findings*

	<b>Nausea</b>	<b>Oculomotor</b>	<b>Disorientation</b>	<b>Total Score</b>
<b>Mean</b>	4.77	5.21	6.09	6.08
<b>SD</b>	6.97	10.62	11.33	9.93
<b>Min</b>	0.00	0.00	0.00	0.00
<b>Max</b>	19.08	37.90	27.84	29.92

*Table 11 - LevelEd VR Post SSQ-Evaluation Findings*

	<b>Nausea</b>	<b>Oculomotor</b>	<b>Disorientation</b>	<b>Total Score</b>
<b>Mean</b>	11.93	6.63	5.22	9.35
<b>SD</b>	12.80	7.77	11.22	8.85
<b>Min</b>	0.00	0.00	0.00	0.00
<b>Max</b>	38.16	22.74	41.76	26.18

*Table 12 - Unity Desktop Pre SSQ-Evaluation Findings*

	<b>Nausea</b>	<b>Oculomotor</b>	<b>Disorientation</b>	<b>Total Score</b>
<b>Mean</b>	4.77	2.37	4.35	4.21
<b>SD</b>	6.03	5.34	11.04	6.24
<b>Min</b>	0.00	0.00	0.00	0.00
<b>Max</b>	19.08	15.16	41.76	18.70

*Table 13 - Unity Desktop Post SSQ-Evaluation Findings*

	<b>Nausea</b>	<b>Oculomotor</b>	<b>Disorientation</b>	<b>Total Score</b>
<b>Mean</b>	4.77	3.79	3.48	4.68
<b>SD</b>	8.53	9.18	9.51	9.01
<b>Min</b>	0.00	0.00	0.00	0.00
<b>Max</b>	28.62	30.32	27.84	26.18

LevelEd VR had eleven participants show some slight symptoms of simulator sickness. Three users reported a single moderate symptom, one of which was vertigo the other two were sweating. The symptoms reported by the remaining eight participants were all slight. Common symptoms for LevelEd VR were sweating (reported by seven participants), general discomfort (reported by five participants) and eye strain (reported by four participants). No participants suffered from any symptoms on a severe level.

Participants of neither system reported an overall score of moderate or severe symptoms in any of the three categories or total score and no one reached the threshold for slight based on the reference totals. The order of task completion does not appear to have impacted sickness levels for either system.

### **5.7.5 Video Coding Analysis**

The video recordings of each task for each participant were analysed to better understand any differences between the two workflows. One key difference revealed is the number of times a participant play tested their level. There was a total of 61 play tests using LevelEd VR with a mean of 3.81, standard deviation of 1.97, minimum of one and maximum of seven. Unity Desktop resulted in 34 play tests with a mean of 2.12, standard deviation of 1.14 and minimum of one and a maximum of four. This resulted in 75.0% of participants completing more play tests with LevelEd VR than when they used Unity Desktop, 12.5% completing the same amount and 12.5% completing more with Unity Desktop. The time it took for participants to transition from editing in Unity Desktop to testing in VR and then the transition back after testing to editing again was also analysed. Participants using the Unity Desktop workflow took on average 31 seconds to transition.

The results of the study show many positives for LevelEd VR and the acceptance of VR level design and editing tools.

## **5.8 Discussion**

### **5.8.1 Acceptance and Preference**

The LevelEd VR system and Workflow 2 scored an average of 79 on the SUS (see **Figure 64**). This result shows that participants considered Workflow 2 as an acceptable workflow for designing levels for VR games. A preference for the LevelEd VR system and Workflow 2 is shown in the results from the comparative questionnaire, with 69.75% preferring to use LevelEd VR versus Unity Desktop at 12.50% (see **Figure 65 – Q1**). This is a positive result as all but one of the participants had 1.5-2.5 years' prior experience of using the Unity Desktop workflow for game and level development. This suggests that whilst they had more experience

with Unity Desktop and the traditional workflow, they saw sufficient benefits in LevelEd VR and Workflow 2.

The results suggest that the key benefit to working with a VR workflow for level design is being able to more effectively judge scale and positioning of objects and the level layout whilst creating the blockout and gameplay (see **Figure 65** – Q4 and **Figure 66** – Q9). This means users can spend more time iterating on the gameplay and general layout rather than checking the scale and positioning of objects is correct. This could be the reason participants preferred to use Unity Desktop over LevelEd VR when working on non-VR games (see **Figure 65** – Q2). Working on 2D screens to create a game for 2D screens (non-VR) is likely to cause less of an issue with scale and perception than working on 2D screens for a game that will be viewed in stereoscopic 3D. Anecdotally, there were several participants who found the scale of their levels were not what they had expected and that areas were simply not visible from the spawn points when they expected them to be, when using the Unity Desktop workflow.

Another potential benefit to having the full workflow in VR is that testing might be easier and more frequent. Ease of testing helps users to test often, which supports the general agile and iterative game development methodologies used across the industry. The video coding results show that participants tested more frequently with LevelEd VR than with Unity Desktop. This difference could be explained due to the average time it takes to transition from editing in Unity Desktop to testing the game in VR and then transitioning back to editing after testing taking 31 seconds. This length of time may have put people off testing as much. Several users noted this issue in their written feedback for LevelEd VR with comments such as *“Being able to immediately edit and play test really improved my experience with the workflow”* and *“Using this system was overall much more pleasant than the usual method of swapping between VR and desktop”*. This break from editing and testing could also break focus and distract the users. Further analysis of this is required over longer sessions (similar to a working day) to see if the results are the same and if the breaks cause issues for users.

When analysing the specific SUS question responses there were some clear trends. For example, participants responded more favourably for LevelEd VR when asked to respond to the statements “I think I would like to use the system frequently” and “I would imagine that most people would learn to use this system very quickly”. This further supports the suggestion that participants preferred to use LevelEd VR over Unity Desktop. However, they did also show a stronger response to the following questions “I think that I would need the support of a technical person to be able to use this system” and “I needed to learn a lot of things before I could get going with this system”. This suggests that they felt they needed support to understand and learn the system, but once they had the initial support, they would like to use



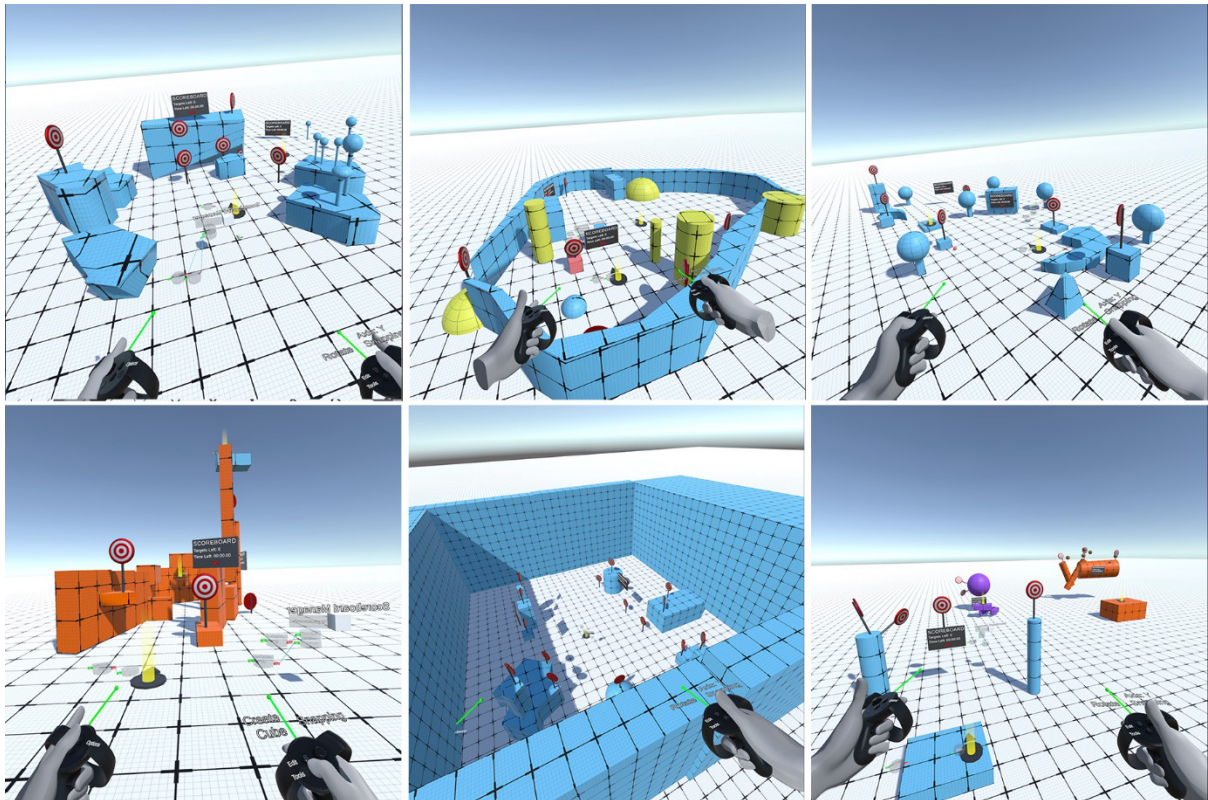


Figure 67 - Sample levels created by participants using LevelEd VR during the study.

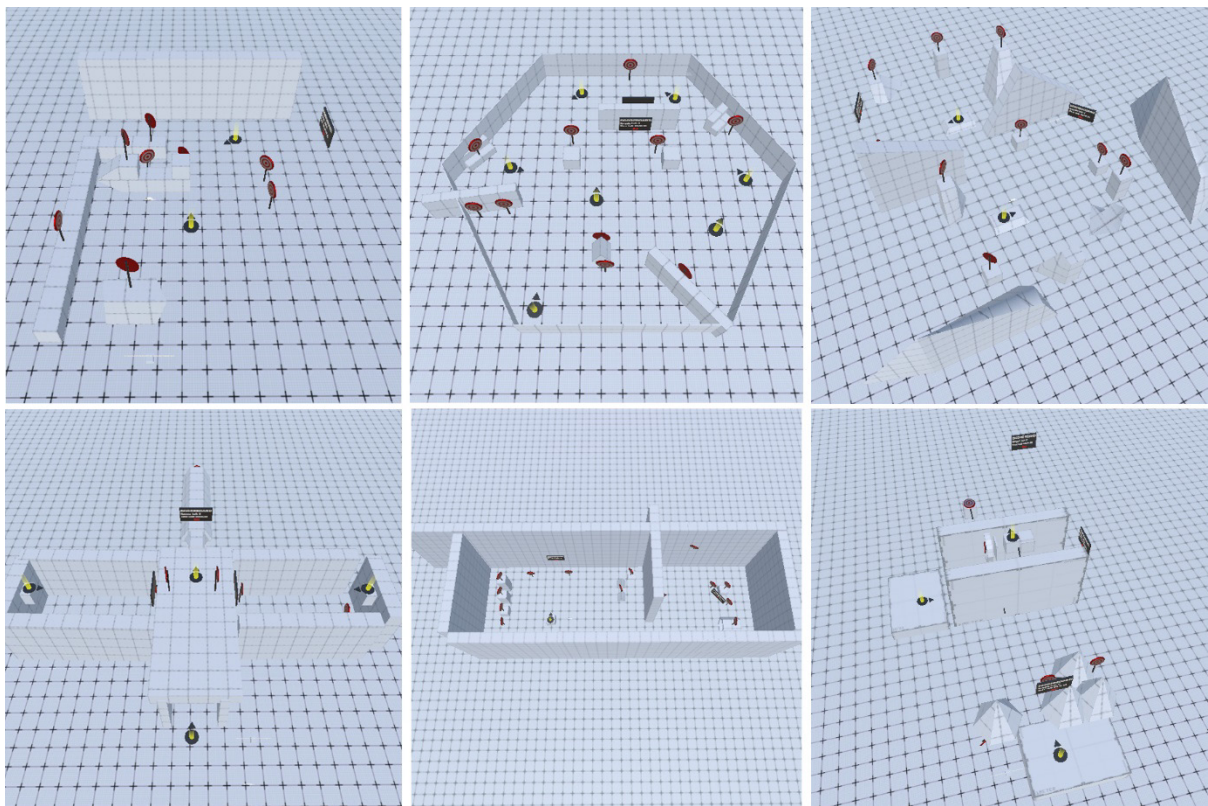


Figure 68 - Sample game levels created by the same participants from Figure 67 using Unity Desktop during the study.



the system for level design for VR games and felt that others would learn quickly. This is somewhat expected, as participants were primarily new to VR and spatial computing whereas all but one of the participants had between 1.5-2.5 years' experience with Unity Desktop and traditional 2D workflows. This highlights the importance of quality tutorials for LevelEd VR and any VR system as most users will be new to spatial computing and will need clear guidance on new interaction paradigms. One issue the participants encountered was activating triggers/grips unintentionally during the tutorial section. Locking out features and slowly introducing them to the users would likely have solved this issue.

Sample levels from the same six participants created using the LevelEd VR workflow can be seen in **Figure 67** and those created with the Unity Desktop workflow in **Figure 68**. Those created in Unity Desktop tended to be slightly further developed, which is expected due to the participants prior experience and having used LevelEd VR for a limited time before creating their levels. However, levels created with Unity Desktop tended to be more rigid in their design, compared to those created with LevelEd VR.

### **5.8.2 Minimising Motion Sickness**

It appears as though both systems/workflows have the ability to cause some slight simulator sickness symptoms, with LevelEd VR causing more simulator sickness symptoms than Unity Desktop. Whilst there are slight simulator sickness symptoms reported, the values are very low for both systems. When comparing the mean values for each cluster and total score for LevelEd VR against the reference values, they fail to reach the slight reference values and are at worst 6 times less and at best 19 times less depending on the category.

This is seen as a positive result considering the main locomotion method and the one that appeared to be used most often by participants was the drag world mechanic, which allowed participants to move and rotate themselves in the virtual world. As rotating in the virtual world and not in the real world can commonly cause sickness (Bonato et al., 2008, Chang et al., 2020), there was a potential for the system to cause large amounts of simulator sickness, but this does not appear to be the case. As well as the SSQ results, this is also supported by the high scores shown in the general feedback results (**Figure 66** – Q1 and Q2) and the lack of any comments regarding simulator sickness in the written comments for locomotion.

Simulator sickness was potentially minimised due to several factors. First is the vignette that reduced participants peripheral vision when moving, which was adjusted based on their velocity and angular velocity. Work exploring limiting peripheral vision suggests there could be a reduction in simulator sickness when the peripheral vision is limited (Buhler, 2018; Fernandes & Feiner, 2016). The potential benefit of limiting peripheral vision is confirmed again more recently and results suggest that similar reductions in simulator sickness could be

gained from only limiting one side of the peripherals when turning (Wu & Rosenberg, 2022). This could be potential improvement to the system as it would reduce the impact of the potential occlusion from the vignette. Participants could also move over large distances using the teleport mechanic, this reduced the amount of time participants had to use the drag world mechanic, opting instead to use drag world for short/medium distances and adjustments. Finally, users could rotate their head and body in the real world when needing to look at something or to face something. This again, reduced the need to always use the drag world mechanic (although many still did). These factors working together appear to have stymied the effect of simulator sickness normally associated withvection and in particular rotating in VR. Participants chose to use a mixture of these three locomotion techniques, and it is recommended that when building systems like LevelEd VR that users are given different options for locomotion as no one size fits all situations.

### 5.8.3 Spatial Scripting

One of the key differences between a world builder and a level editor is the ability to script and test gameplay. Therefore, the accessibility and usability of the spatial scripting system is important for the success of LevelEd VR as a level design tool. The results show that users found the spatial scripting system easy to use once shown how it works (**Figure 66** – Q7). The majority of participants also understood the benefits of having the scripting system in the level, such as the relationship between script nodes and objects in the scene (**Figure 66** – Q8). This suggests that the spatial scripting system is worth pursuing further, which more recent work appears to agree with (Zhang & Oney, 2020).

However, there were some issues reported by users such as the ‘in’ and ‘out’ tags (**Figure 59**) being “*a little small*” and they had “*difficulty selecting them at times*”. Increasing the size and adding a highlight to the tag they are trying to attach to should help solve this. One user also commented that they “*felt as if the script elements polluted the scene*”, which is a valid point that should be addressed in future versions as levels get more complex. Potential solutions are to give users the ability to hide/reveal scripting nodes when necessary and possibly to allow nested nodes when complex script node chains are created. More research is certainly required on different solutions for programming/scripting in VR.

### 5.8.4 Interaction Issues

One of the key design goals for the system was to build interactions from the ground up for VR and avoid using 2D windows/paradigms by default. This appears to have worked well with users finding the menu system and structure easy to understand and quick to change tools (**Figure 66** – Q3 and Q4). However, the usability of some interactions and tools suffered due to this thinking. For example, the rotation tool received several complaints or improvement

requests. This was due to how a gesture-based system was used rather than relying on the standard rotation XYZ widgets commonly found in 2D desktop modelling applications. Participants found it difficult to know which way the object would rotate and often tried to use a gesture relating to the direction they wanted, rather than up and down for all interactions as the axis was fixed from the options menu. This suggests that either more feedback is required for users to understand how the gesture works, or that some 2D interaction techniques may still be viable in spatial computing.

## **5.9 Conclusion**

This chapter has presented and discussed a generic runtime VR system and Workflow 2 that supports the level design and level creation process for VR games with the initial version presented here focusing on supporting the blockout process. Participants with prior game development and Unity experience were accepting of the system and when compared to an existing 2D workflow using Unity, the trends suggest that the majority of participants preferred to use LevelEd VR and therefore Workflow 2. The key benefit to the system appears to be the improvement in judging scale and positioning of objects/gameplay, which allows users to spend more time iterating elsewhere, such as on gameplay with the spatial scripting system.

There are also potential efficiencies gained from working with LevelEd VR such as being able to instantly change between editing and testing, rather than swapping between desktop and VR system as is the current process (which takes time). As the system is a runtime editor, it can also run on standalone systems, such as the Oculus Quest, opening the ability to work away from a desktop or laptop PC. With the current popularity of user generated content, developers can also choose to release LevelEd VR as part of their game to allow for user generated content by players for entertainment games or users and practitioners for serious games.

Whilst the system has been successful in meeting the aims, there are areas for improvement, such as with the scripting system, natural and gesture-based interactions and work must continue to reduce the effects of simulator sickness further to ensure the system is accessible to all.

The following chapter shows some of the lessons learnt from this study implemented and how both LevelEd AR and LevelEd VR can be combined to create an entirely new workflow (Workflow 3) that supports SR level design.

## Chapter 6 LevelEd SR – Workflow 3

The following chapter outlines the motivation for Workflow 3, along with the design of the LevelEd SR software that enables this workflow. A study was devised and carried out to evaluate the workflow and LevelEd SR software and the results of this study are presented here along with a discussion.

### 6.1 Introduction and Motivation

LevelEd SR is the culmination of work on both LevelEd AR - enabling players to capture a virtual model of a real-world location and LevelEd VR – allowing players to create custom VR levels whilst working in VR. This combination is defined as Workflow 3 (**Figure 69**), which could be considered an entirely new and novel workflow when compared to Workflow 1 and 2 which instead improve upon existing industry workflows. The result is a workflow that enables players to experience SR by augmenting a virtual game level with objects from the real-world location. The workflow was specifically created to ensure it could be achieved with consumer devices and be used by game players of differing experiences. This was important to ensure this system would not be limited to specialist locations due to the need for specialist hardware or skill sets.

There are several potential benefits to SR games, over just VR games. One key benefit of SR is the passive haptics delivered by the inclusion of real-world objects within the virtual world. For example, instead of an office desk in their real world, a player could be presented with a virtual sci-fi control deck that they could interact with physically. Controllers with vibration functionality (such as Oculus Touch controllers) or haptic gloves help by offering tactile feedback, but are unable to restrict a player's movement, like leaning on a table. An increase in presence and immersion has been demonstrated in previous studies looking at passive haptics (Garcia-Palacios et al., 2002; Hoffman, 1998, Meehan et al., 2002) and SR (Eckstein et al., 2018; Simeone et al., 2015), but little work has looked specifically at entertainment games as outlined in the literature review.

Another benefit to SR and thus enabled by Workflow 3 is a better use of the player's real-world space. The last few years has seen continual growth in the VR market, with the release of standalone headsets, such as the Oculus Quest 1 and 2 (Lang, 2022a). Currently, room-scale VR experiences require the user to find an empty space so that players do not knock into furniture. This can be difficult in some spaces, for example bedrooms, where furniture such as the bed often takes up most of the room space. Players could get around these issues by visiting location-based entertainment (LBE) centres. For example, centres such as those that support SpringboardVR's Tower Tag (Dickinson, 2019), which often feature large open spaces

and sometimes integrate real-world objects into the virtual world. However, a solution that works in a player's home or chosen location would improve the VR user experience. SR is one possible solution for the lack of space in room-scale experiences as a game could incorporate existing furniture within the virtual world without breaking immersion of the player by revealing to them the real world, such as in mixed or augmented reality. This will increase the possible player space making VR usage potentially more accessible.

Games that feature user generated content (UGC) have also grown in popularity over the past decade (Sherr, 2020). Games such as Minecraft, the Halo series and Fortnite offer in-game tools for content creation. These games give creative freedom to their players and allow them to create their own game content/game levels for personal pleasure or to share with others. A small number of VR games have also started to implement VR based UGC tools, such as Angry Birds VR: Isle of Pigs, Racket: NX and Rec Room. Workflow 3 therefore enables developers to include creation tools in their games that players can use to create custom content. This is of benefit to developers as games that have official modding support through tools such as level editors/SDKs result in increased modding activity, which research suggests leads to higher game sales (Poretski & Arazy, 2017).

## 6.2 Workflow 3

There is currently an opportunity and potential benefit for creating tools and workflows that enable user generated content creation, in the form of game levels (not just virtual environments), that also support SR. It is clear from analysis of the literature that there is limited research into systems that support user generated SR experiences, particularly those that support game play rather than just world building. There are also limited examples discussing the potential benefits of SR/passive haptics for entertainment games over more serious games or applications. LevelEd SR and Workflow 3 was developed with this in mind, and this chapter presents:

1. A comparative study of a game level played in SR and VR that explores both game experience and presence.
2. The design and implementation of a prototype SR level editor and workflow called LevelEd SR.
3. An evaluation of the LevelEd SR system and workflow by participants.

LevelEd SR was developed to see if it was possible to create a workflow to support SR game levels on consumer devices. It could also be used to compare SR and VR levels for game experience and presence. The LevelEd SR workflow for developing a new SR game level is as follows (see **Figure 69**):

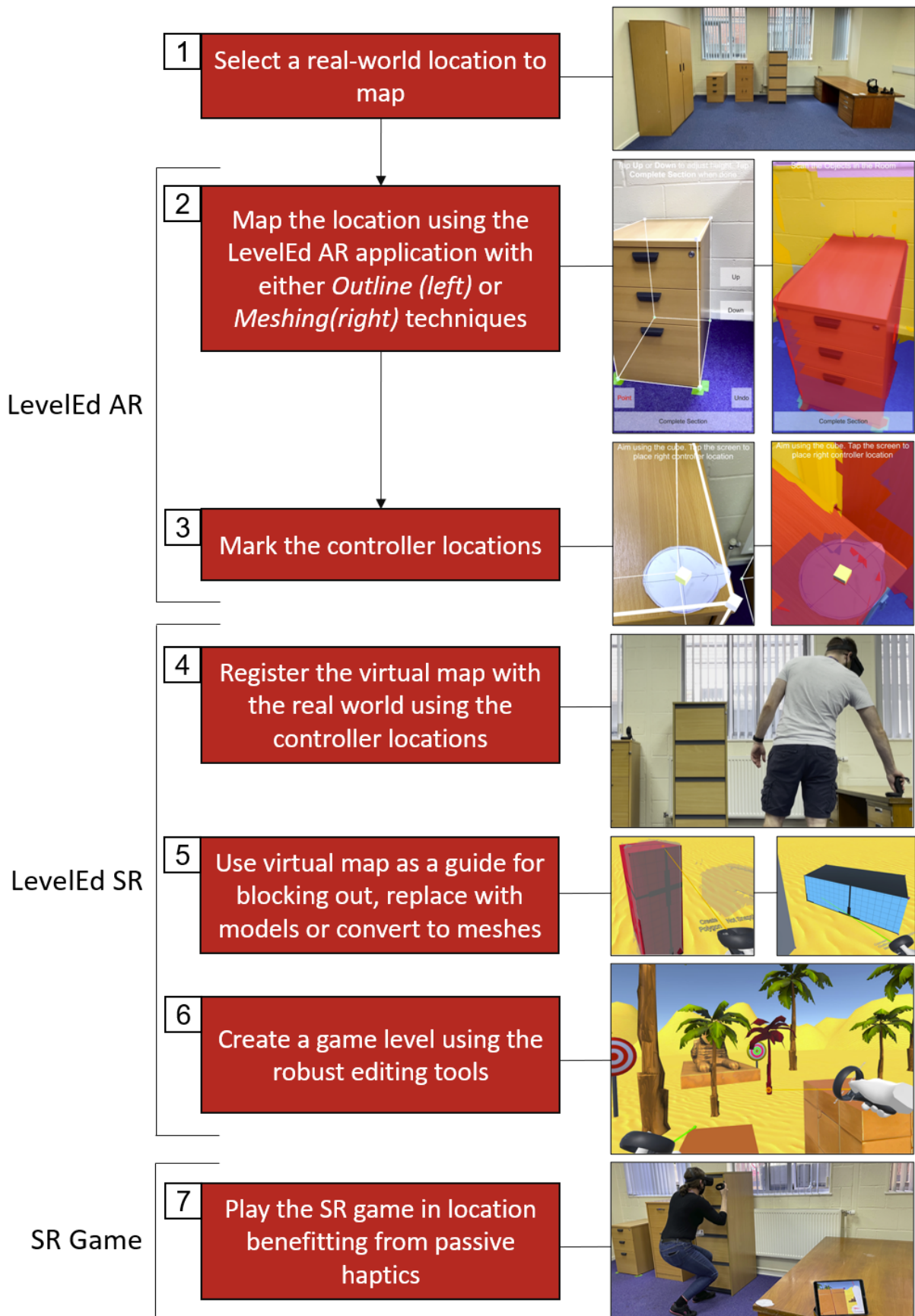


Figure 69 - Workflow 3 which brings together work on LevelEd AR and LevelEd SR to create a SR level editor and workflow.

1. Player selects a real-world location that they wish to use for a SR game level.
2. Player maps a chosen space using the LevelEd AR smartphone application using either the *Outline* or *Meshing* techniques.
3. Once the space is mapped, players mark out controller locations on the newly acquired virtual world data. The virtual world data (including controller locations) is uploaded to a webserver.
4. Player enters the LevelEd SR application on a VR headset. The player places their controllers at the same locations as marked on the AR capture. This is used to register the virtual world data with the real-world. The data is downloaded from the server and virtual and real-world are registered.
5. Players use the virtual world data to help build a game level mapped to the real-world location. The virtual world data is either replaced with generated meshes, existing models or converted in some cases.
6. Players create a game level by placing decorative models, interactive objects and editing basic gameplay variables using spatial scripting.
7. Players play their SR game level that now supports passive haptics.

When a player wants to play an existing game level, they simply enter the LevelEd SR application, place their controllers in the mapped location and the level will be downloaded and registered to the real world and they can then play the SR level.

Workflow 3 achieves the above by supporting the following features, which are discussed in more detail later in section **6.4**:

**LevelEd AR software includes:**

- Wall mapping – Users can map the location, shape and height of walls within a room.
- Object mapping – Users can map the location and general shape and size of objects within the space, such as furniture.
- Saving – The data is saved to the device.
- Uploading – The data is uploaded to a webserver.
- Downloading and Recreating – The data can be downloaded to a Unity project and recreated within a scene.

**LevelEd SR software includes:**

- Support for downloading and integrating virtual mapping data from LevelEd AR and registering it to the real world.
- Support showing and hiding virtual mapping data generated from LevelEd AR. Users can also convert *Outline* data into game objects that can be used in the SR game.



- Tools to enable the quick creation of prototype blockout layouts with robust geometry editing tools with accurate depth, scale, and perspectives for VR.
  - a. Primitive shape creation, such as Cube, Cylinder, Polygon, etc.
  - b. Common editing tools, such as Move, Copy, Rotate, etc.
- The ability to place and script gameplay elements to edit basic gameplay in the editor.
- The ability to add models quickly and efficiently to the scene.
- The ability to quickly play test the level for a rapid and agile iterative workflow without leaving the VR level editor.

An example video demonstrating LevelEd SR in action can be found at **Appendix C1**. The Unity projects for the VR editor and AR companion application can be found at **Appendix C2** and **Appendix C3** respectively.

### 6.3 Hypotheses

The following three hypotheses were devised to address the ability of Workflow 3, which uses AR and VR technology, to create a new experience for players through substitutional reality level design:

**H5:** *Substitutional reality game levels offer an improved level of presence (based on the Spatial Presence Experience Scale) over virtual reality.*

**H6:** *Substitutional reality game levels offer an improved game experience (based on the Game Experience Questionnaire) over virtual reality.*

**H7:** *An AR to VR workflow is accepted (based on the System Usability Scale) by consumers to enable user generated substitutional reality game levels.*

The study outlined in section **6.5** was devised to test all three hypotheses, with the results presented and discussed later in this chapter.

### 6.4 System Overview and Implementation

LevelEd SR (**Figure 69**) was developed to allow players to experience SR using consumer hardware, with the aim to offer wider access to the technology. The system allows players to map a space, such as an office or bedroom, considering the location, size, and general shape of objects (e.g., furniture), which can then be used in tandem with a VR headset to create a SR game level and experience. Workflow 3 is comprised of LevelEd AR for mapping and an updated version of LevelEd VR (now called LevelEd SR) for editing and playing levels on a VR headset. All components of Workflow 3 were built using Unity (Unity 2021.1.9f1 and Unity 2020.1.7f1), primarily due to its proven history in AR and VR development, but also due to its

popularity, with 60-70% of games on AR/VR platforms and an estimated 50% of games across other devices are made with Unity (Dillet, 2018; Unity, 2022).

#### 6.4.1 Mapping the space in AR

The AR smartphone application builds on previous work on LevelEd AR (Beever et al., 2019a; Beever et al., 2019b). It features two techniques for mapping a space, an improved technique called LevelEd AR: *Outline* and a new addition in the form of LevelEd AR: *Meshing*. These will be referred to as *Outline* and *Meshing* throughout the rest of the thesis. These techniques can be used in tandem or separately as they do not rely on one another.

**Outline** – This technique was improved over the version used in Workflow 1 and the related study. This version was rebuilt on Unity’s AR Foundation (Unity, 2020) and is therefore compatible with any smartphone that supports Apple’s ARKit or Google’s ARCore. This means it is potentially compatible with around 1.5 billion devices (Révész, 2021). Porting the software to WebXR was considered, but the standard did not support the requisite features outlined in the rest of this section.

The system presents players with an AR view and asks them to place marker points around the base of an object, before lifting up the volume to specify height of an object (**Error! Reference source not found.**2). Volumes of any number of sides can be created (limited to one height segment) and composite volumes used to map more complex shapes. The system continues to use a mixture of plane detection and points-of-interest along with raycasting to allow for the placement of markers that are anchored in place. Users can now switch between detection type depending on the situation as whilst plane detection is preferred, in some situations a plane cannot be generated and so points-of-interest can be used (**Appendix P**). This technique has also been improved with the inclusion of an undo system that allows users to roll back any marker points placed to tweak their mapping. The system supports the ability to map walls as planes and objects of any number of sides. Players move around the space mapping objects that need to be included in the SR experience. Players are advised not to map objects that will move between sessions, for example, an office chair.

**Meshing** – This technique was a new addition compared to Workflow 1 and is again built on top of Unity’s AR Foundation (Unity, 2020), but only supports devices that are compatible with ARKit and that feature a LiDAR sensor, such as the iPad Pro and iPhone 12+ Pro devices. This means the system is much less widely accessible than the *Outline* technique but does offer additional benefits. The system makes use of the LiDAR sensor to automatically generate a polygonal mesh as the player moves the phone camera around the space (**Figure 69.2**). This allows the player to quickly map a space and to capture more intricate details of the objects. The system uses ARKit Scene Geometry to segment different types of objects. In the

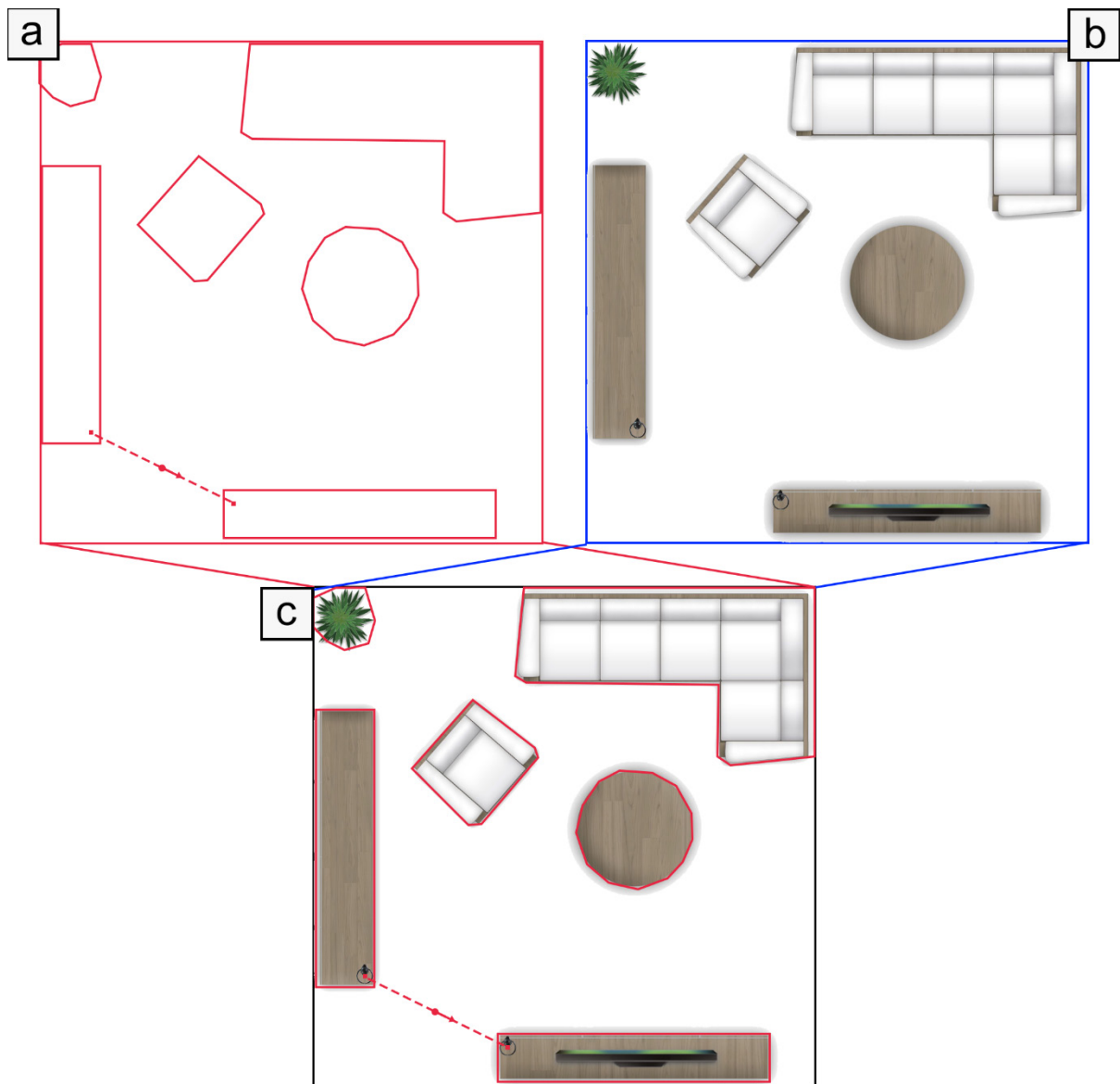
current iteration, tables, chairs and other obstacles are segmented and captured whereas walls, ceilings and floors are stripped out as the VR headset's on-device chaperone will deal with those and it will also reduce clutter within the virtual environment (**Appendix Q**).

#### **6.4.2 Registration of the Real and Virtual World**

Once players have captured a space using either *Outline* or *Meshing*, they must then mark out the controller locations. Marking the controller locations in the AR captured environment ensures there are common data points between the real world (actual controller locations) and the virtual world data (marker controller locations) (**Figure 69.3**). This is an important step as it is used to register both the location and rotation of the virtual world when in VR. Marking the controller locations uses the same marker placing technique as *Outline*. Paper controller location markers are provided (see **Figure 69.3**) to aid with this step and to ensure players can remember the location of the markers for future sessions. Once complete, the data is then saved and uploaded to a webserver (**Appendix R**).

To load the map, the user enters the VR level editor (named LevelEd SR) on their VR HMD whilst in the real-world location. They place the controllers in the same location as where they were mapped and press the register button. The virtual map is then automatically loaded and aligned to the real-world location along with the furniture within it. This controller method was selected as it allows for consistent results in positioning and aligning the virtual world with the real world, but also ensures the map is aligned not only in position, but rotational angle as well. A single controller could have been used, but that would have placed more importance on the exact rotation and placement of the controller, by using two controllers the position and rotation can be calculated from the midpoint between the controllers and corresponding vector. The midpoint is saved as the pivot of the virtual map and the forward direction is set along the vector of the two points. When the player presses the register button the midpoint and rotation between the marker locations and the real-world controller locations are aligned (**Figure 70**).

Other techniques, such as marking the start position of the HMD when mapping the space could have been used, but this does not guarantee positional or rotational accuracy as the users might forget where they started, not stand in the exact position or face the exact direction. Users could have also been given the ability to manually position the virtual environment whilst in the headset, but this felt cumbersome and added additional work for the user.

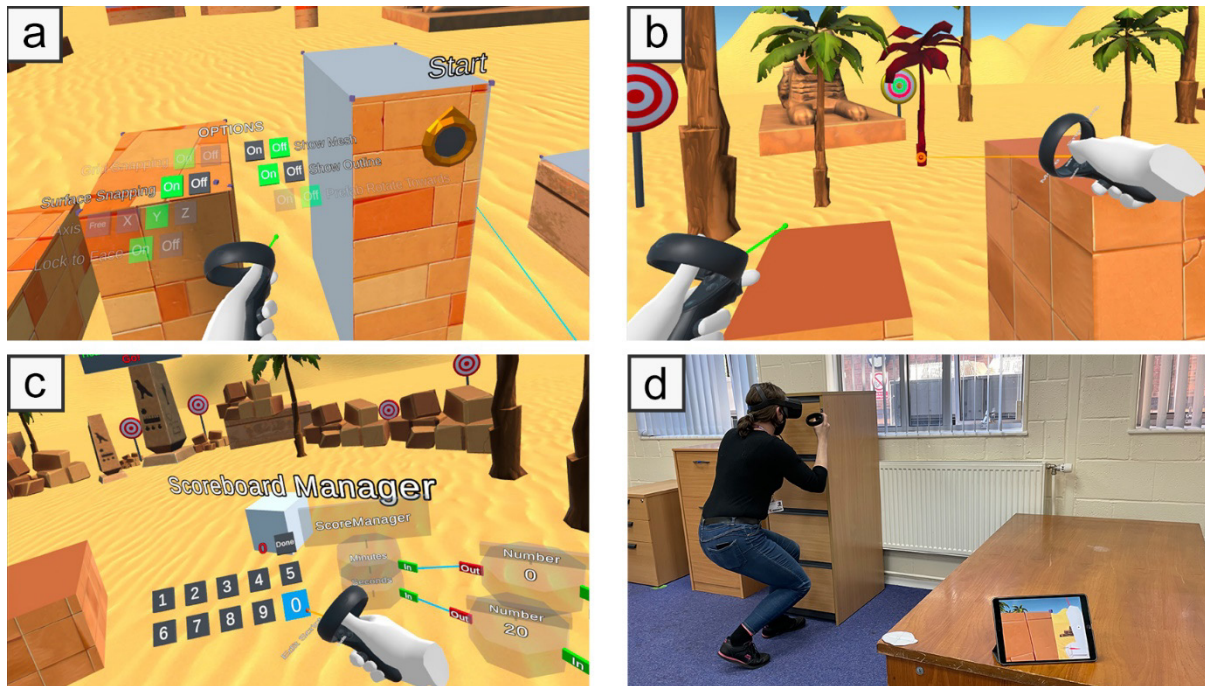


**Figure 70** - (a) Virtual map captured using LevelEd AR. Controller locations are captured during the session. (b) User puts on their HMD in the real-world location and place the controllers in the same position. (c) User presses the register button, and the virtual world is aligned with the real-world using controller locations as common data points.

### 6.4.3 Substitutional Reality Level Editor

The SR level editor component is built on an improved version of LevelEd VR (Beever et al., 2020) which now accepts and integrates data created with LevelEd AR (Beever et al., 2019a). Additional features and tools to support SR level editing were also added to the LevelEd SR system. The system supports the creation and editing of game levels whilst using a VR headset and 6-DoF controllers. The Meta Quest (a standalone device) was selected as it is more convenient to use in different places due to its compact size, and minimal setup required. Meta Quest devices were also selected due to their popularity and considerable market share (Lang, 2022a). Whilst the workflow also supports the Meta Quest 2, the Meta Quest 1 was

chosen for the study to demonstrate that the system works effectively on more constrained hardware after additional optimisations were made. At the time of the study in July 2021, there was no developer access to the Passthrough feature (which allows for the use of onboard cameras in the headset) for the Meta Quest devices (Oculus, 2022). The workflow devised was partly due to that limitation, but also knowing that it would support additional devices that also do not offer similar Passthrough features. VRTK (2022a) was used for some components, such as input detection, to support porting between devices.



**Figure 71** - (a) Player has turned on the objects loaded from the **Outline** data. (b) A player moving a prefab (tree) in a participant created level. (c) A player editing a number scripting node in a participant created level. (d) A participant testing their SR level (the tablet displays what the player is seeing).

#### 6.4.4 Locomotion

Unlike the version of LevelEd VR used in Workflow 2, there is only one option for locomotion around a scene. Locomotion is achieved by the player moving around in the real world only. If they were allowed to teleport around the scene or use the Drag World method, then objects would no longer align between the real and digital world which could cause confusion or worse create a dangerous situation.

#### 6.4.5 Menu and Interaction

As with the original version of LevelEd VR, interaction is supported by a variable pointer beam. Players can select objects by pointing at them and can create objects at the beam end point. The length of the beam can be adjusted to fixed points with an analogue stick click to 10cm, 1m and infinite, to enable work close up, at arm's length and greater distances respectively or fine-tuned by pushing up or down on the analogue stick (**Figure 71b**).

The menu appears in front of the controller when requested and options are selected by either pointing at or intersecting the 3D buttons with the pointer (**Figure 72**). The tools menu was adapted to include sections on:

- **Creation:** This includes the cube, cylinder, sphere, edit vertices, etc. tools
- **Gameplay:** This includes prefabs for gameplay items, in the case of this game, the targets, scripting nodes, etc.
- **Models:** This includes an array of 3D models that players can add to their scenes

Shortcuts were also added so that players could quickly change between menus using the analogue stick (left and right), rather than having to only use the mode selection buttons.

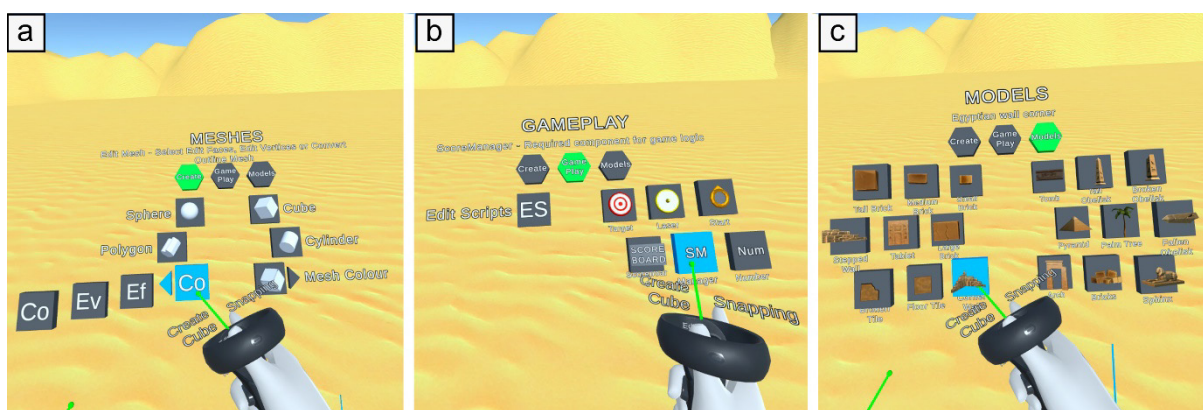


Figure 72 - (a) Creation menu. (b) Gameplay menu. (c) Models' menu.

### 6.4.6 Level Editing Tools

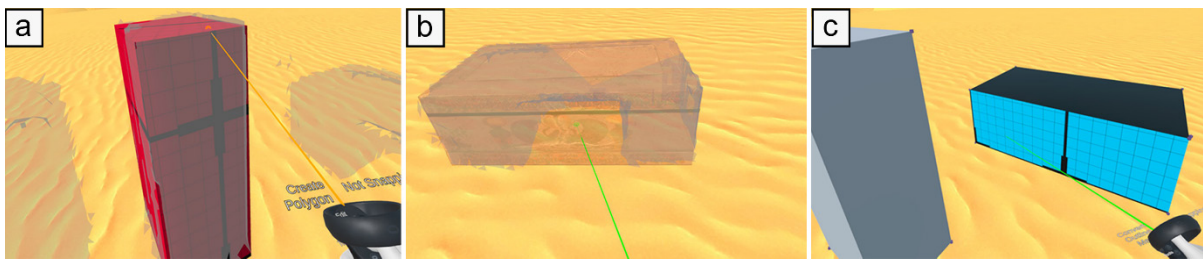
Creation and editing tools available to participants are listed in **Table 14** but key categories are described here in more detail.

- **Geometry/Layout:** These tools enable a player to procedurally create static meshes that can be combined to create level geometry, such as walls, pillars, tables, etc. Players can also choose to convert *Outline* data to be used in game (**Figure 73c**) or create meshes to match physical objects (**Figure 73a**). They can then change materials on these newly created meshes with Edit Colour.
- **Prefabs/Scripting:** Prefabs (models, interactive objects, etc.) can be added to a level by a player (**Figure 71a**). They can also place and edit models to match physical objects in the space (such as a tomb in place of a desk) (**Figure 73b**) or to add them outside the direct play space. They can also add spatial scripting nodes directly into the level (**Figure 71c**). The spatial scripting system continues to utilise a flow-based node system that allows players to connect 3D nodes together to change values or create functionality. Scripting nodes exist in the game level to enable benefits such as



additional spatial information from the proximity of nodes to objects and connections between them.

- **Editing/Options:** Players have access to many common 3D modelling and level editing tools to help manipulate objects in a level as well as several options (see **Table 14 – Editing and Options/Modes**). Some new additions are the ability to show or hide either the *Outline* or *Meshing* objects loaded from the data captured with LevelEd AR. This gives players the ability to keep them visible whilst they design the level but can then hide them once they've created digital proxies.



**Figure 73** - A demonstration of using the virtual mapping data to create an SR game level. (a) User can use the mesh creation tools to create a mesh that matches the virtual map data. (b) User can use a pre-existing model to replace the virtual map data. (c). User can use the Outline virtual map data and convert it into a game object.

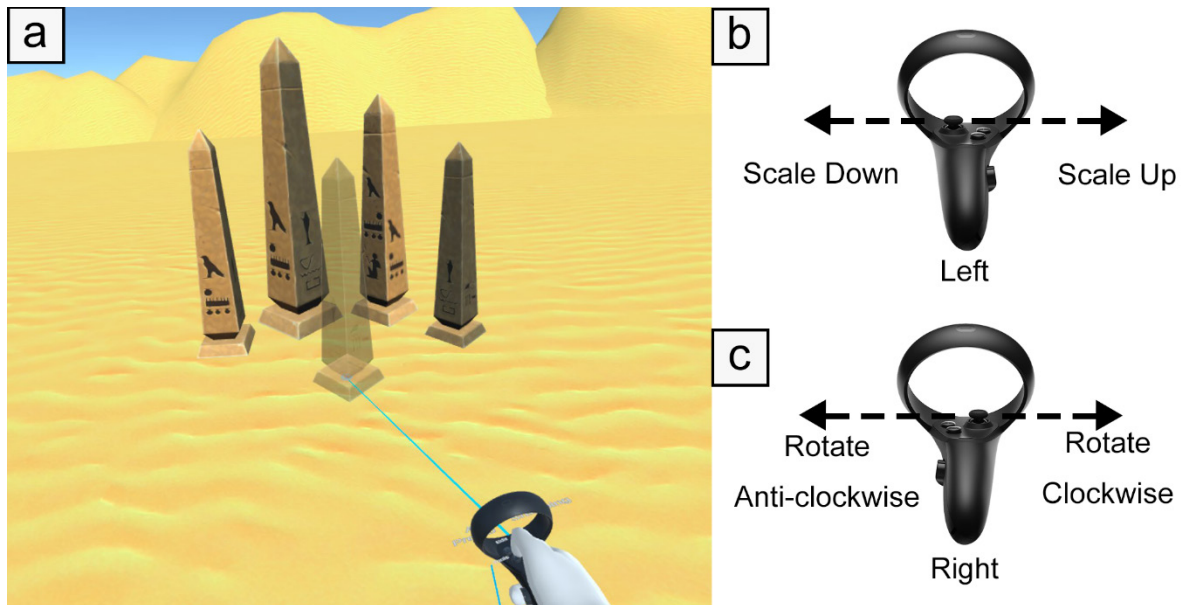
There were many improvements made over the previous version based on feedback from the previous study and from personal testing. For example, in the last iteration of LevelEd VR, to place and edit a prefab required the user to use the Add Prefab tool to place a prefab in the scene but with no preview. To edit it, the user would need to swap to the individual tools such as Rotate, Move and Scale. This was slow and cumbersome. Therefore, shortcuts and a model preview were added to several tools, such as the Add Prefab or Add Model tools. Now as a player adds a model, they see a preview of the model so they can move it around and

**Table 14** - LevelEd VR Features Available to Participants. Major new features in bold.

<b>Geometry</b>	<b>Script/Prefab</b>	<b>Editing</b>	<b>Options/Modes</b>
Create Cube	Number Node	Move	Enter Play Mode
Create Sphere	<b>ScoreManager</b>	Copy	Exit Play Mode
Create Cylinder	Edit Scripts	Delete	Grid Snapping
Create Polygon	Add Prefabs	Rotate	Current Axis
Edit Vertices	<b>Add Models</b>	Scale	Normal Lock
Edit Faces		Grab	Surface Snapping
Edit Colour		Group	<b>Hide/Show Mesh</b>
<b>Convert Outline</b>		Ungroup	<b>Hide/Show Outline</b>



using shortcuts on the analogue stick they can rotate and scale all using the same tool. This decreases the time spent when adding multiple objects to the scene as there is no need to swap tools (see **Figure 74**)



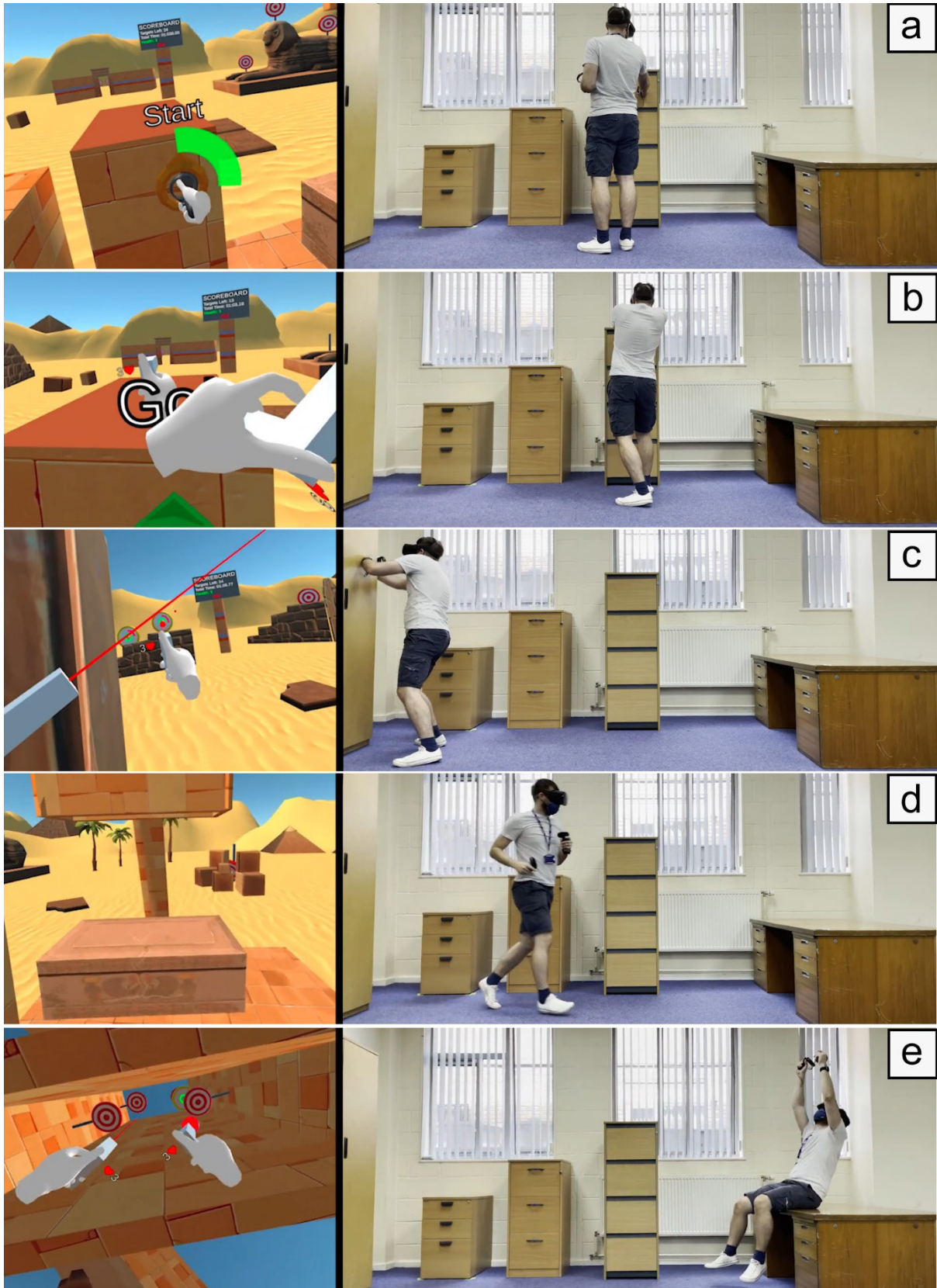
**Figure 74** - Preview and shortcuts can be used to quickly place same assets with variations. (a) Preview of any object that can be placed in the scene. This helps with placement and live editing. (b) Shortcuts that can be used to scale whilst placing an object. (c) Shortcuts that can be used to rotate whilst placing an object.

#### 6.4.7 Sample Game

A sample game was developed to enable participants to not only play an SR game level for the comparison, but also so they can create their own experiences. The game was built on top of the game used in Workflow 2 but with some changes and new features. The game is a target shooting game set in Egypt, which utilises modified assets from Total Game Assets (2018) and 3DBear (2018). Players still use a gun to shoot at targets within the environment, but now have a gun in each hand. As well as the time limit, players now have three lives that they must preserve to complete the level. These are all shown on the scoreboard and via the heart icon next to the guns. There are now two types of targets:

- A single shot simple red and white target (**Figure 75a**) as previously used in Workflow 2.
- A new laser target (**Figure 75d**) that shoots at the player and takes four shots to disable.

The laser target charges up over time whilst the player is in view. A shot is fired if they do not take cover in time and if hit, they lose a life. Use of the physical environment was encouraged through hard to reach targets (such as requiring players to sit on physical objects – see **Figure 75c** and **Figure 75d**), through the laser target requiring players to take cover behind objects



**Figure 75** - The sample level being played in SR. Left is what the player is seeing whilst interacting on the right. (a) Player starts the game using the start node. (b) Player uses the furniture/Egyptian block to help aim. (c) Player uses the wardrobe/Egyptian tablet for cover from the laser targets. (d) Player runs towards the desk/Egyptian tomb. (e) Player sits on the desk/Egyptian tomb to shoot hidden targets above them.

and the start node, which requires the player to place their controller in the slot (placed next to physical objects) to start the level (**Figure 75a**). Movement is only possible through natural real-world locomotion.

The study sample level was made entirely with the LevelEd SR workflow before the study began. A VR only version of the level was created that featured the exact same level layout as the SR version. The only difference between the two versions is the lack of physical objects in the real-world space in the VR version, whilst the digital obstacles remained.

In their own created levels, players can choose how to place pre-created decorative models, geometry created with the creation tools, interactive prefabs like the scoreboards/targets, as well as set the time limit for the level using the spatial scripting system. Whilst building their own levels, players can choose to quickly toggle between Play mode and Edit mode by pressing in the two analogue sticks (**Figure 71d**). This was done to ensure players can quickly test and iterate on their levels.

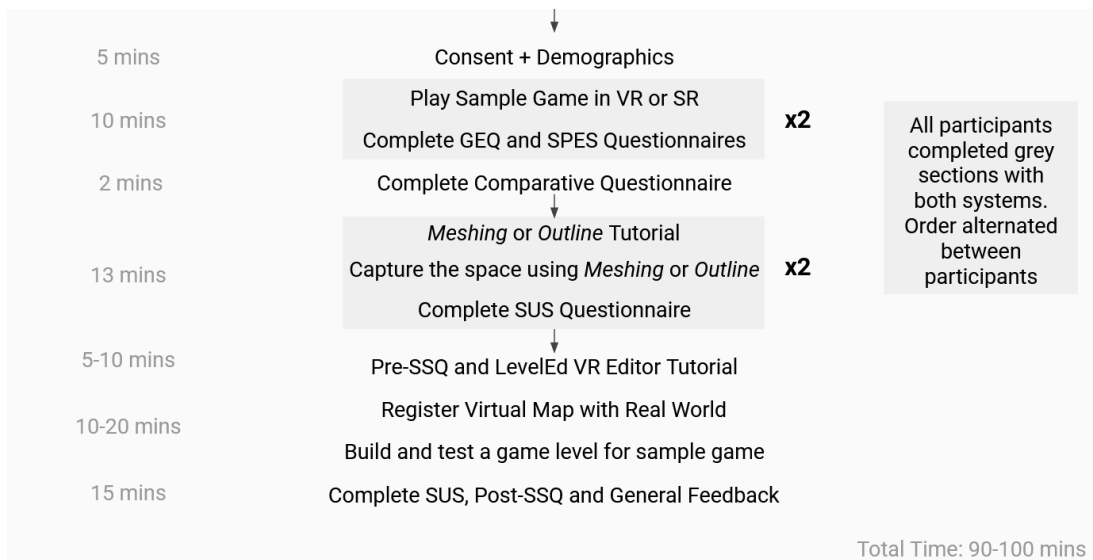
## **6.5 Study 3 - Experiment Methodology**

The study was designed to not only evaluate the LevelEd SR system and workflow, but to also investigate the differences between playing an action game in SR with passive haptics compared to only VR. The SR game took place in a space with physical objects, which were incorporated into the game, whereas the VR version took place in the same sized space, but without any physical objects. The level layout and content were the same between both games.

### **6.5.1 Study Structure**

Ethical approval was obtained for the study from the FSE Research Ethics Committee (see **Appendix J**). Participants took between 90-100 minutes to complete all the tasks in a within-subjects design (see **Figure 76**). Each participant started alternatively in either SR or VR to mitigate potential learning effects. The sample game (see **6.4.7**) featured a single level that was played twice consecutively under each condition and took around a minute to complete. The starting condition was alternated between each participant with nine playing SR first and eight playing VR first. After each condition, participants completed both a Game Experience Questionnaire (GEQ) (IJsselsteijn et al., 2013) and Spatial Presence Experience Scale (SPES) (Hartmann et al., 2016) survey. The GEQ was selected due to its popularity in games research with 660+ citations on Google Scholar, however, it is understood that it has not been fully validated (Law et al., 2018). After both had been played, they also completed a comparative questionnaire to gauge their preference (see **Table 17**). Once complete, participants moved onto creating their own SR game level using the LevelEd SR





**Figure 76** - Order of tasks carried out by participants during the study.

system and workflow. Their first task was to map the space using the improved LevelEd AR smartphone application. Participants mapped the space using both the *Outline* and *Meshing* techniques. As before, the order was alternated between participants with nine participants using *Outline* first and eight using *Meshing* first. The space included five pieces of furniture to map along with the paper controller location markers (see **Figure 77**). After using each mapping technique, participants completed a System Usability Scale (SUS) (Brooke, 1996) questionnaire. The time taken to complete the task with each technique was also recorded.



**Figure 77** - The study space consisting of the five pieces of furniture set up to mimic a traditional office.

Once the participants had captured their own data, they were then introduced to the LevelEd SR system on the headset. The researcher gave a five to ten-minute demonstration of the system before participants used the system. The participants' previously captured data was loaded and registered to use in the game level. Participants were then given an opportunity to create their own SR game level. Participants were not told to create a specific type of level, but they were given verbal guidance by the researcher on the basic requirements of a level and were given some suggestions to ensure they sampled as many of the tools and pre-defined library of models/prefabs as possible. Participants completed a Simulator Sickness Questionnaire (SSQ) (Kennedy et al., 1993) before and after they used the LevelEd VR system. They also completed a final SUS questionnaire about the editor and a general feedback questionnaire on the editor and workflow (see **Table 20**).

### **6.5.2 Hardware**

The mapping task using the LevelEd AR software for both *Outline* and *Meshing* were completed on an iPhone 12 Pro device. The LevelEd VR software ran on a Meta Quest 1 device. An iPad Pro with the Oculus application installed was used to broadcast footage from inside the VR headset to the researcher and participants.

### **6.5.3 Participants**

There were 17 participants, with 14 identifying as male, two as female and one as non-binary. Nine participants were between the ages of 18-29, three between 30-39, four between 40-49 and one between 50-59. Due to the Covid-19 pandemic it was a challenge to recruit participants. To improve the potential sample size, participants were selected through convenience sampling, and they were made up from staff and students from the University of Chester and several alumni. Participants were recruited via an email sent out to staff and students and alumni were recruited with a message on a Discord server (see **Appendix M**).

There was a mixture of VR experience between participants with 18% having no experience, 24% rating themselves as novice, 12% as intermediate, 35% as advanced and 12% as experts. There was a similar range across participants' game development experience, with 18% having no experience, 24% rating themselves as a novice, 41% as intermediate, 12% as advanced and 6% as expert. All participants had played some form of video game before with 25% playing several times a year, 13% several times a month, 57% several times a week and 23% every day.

The demographic of this convenience sample demonstrates an imbalance in gender (more male identifying participants) and age (participants skewed to the younger age group). Whilst the sample population drawn upon was wider than study 1, most participants still came from a background of game development either as a student or from industry. This therefore limits

the ability for the results to be generalised to the wider population and should be viewed through the lens of being comparative to similar populations of those with prior game development experience.

#### **6.5.4 COVID-19 Safety Procedures**

As the study took part during the COVID-19 pandemic the following safety procedures were implemented to reduce the risk of transmission to participants and researchers. Participants were asked not to attend if they were currently self-isolating. Hand sanitiser was used upon entry and during the study. All participants and researchers wore a mask throughout the study; however, it was necessary for some to pull down the mask below the nose whilst in the VR headset to avoid fogging. VR equipment was not shared between participants and the researcher. VR equipment and surfaces were cleaned after each use, including the use of a Cleanbox (2022) to further sanitise the VR headset and controllers. VR equipment was enhanced with rubberised mask covers to aid with cleaning between participants on the same day. Windows and doors were kept open, and a pedestal fan was used to improve ventilation.

### **6.6 Results**

Three data capture techniques were used to gauge the difference between the game played in SR and VR. The data capture techniques looked at the difference in game experience, levels of presence and preference between each technique by participants. The software jamovi 1.6.23.0 (The Jamovi project, 2021; R Core Team, 2020; Kerby, 2014; Revelle, 2019) was used throughout except for the non-parametric tests which used SPSS 26. A statistical significance of  $\alpha = 0.05$  was used.

#### **6.6.1 Game Experience Questionnaire (GEQ)**

The Game Experience Questionnaire (Core Module) (IJsselsteijn et al., 2013) consists of 33 statements that participants rate using a 5-Point Likert scale (0-4) ranging from “not at all” to “extremely”. The statements are grouped and give a score over seven categories of game experience. The GEQ can be used to compare the experience between multiple games or the same game with differences. **Table 15** displays the results for the GEQ. Paired Sample T-Tests were carried out for each category. Shapiro-Wilk analysis showed that Competence, Tension/Annoyance and Negative Affect were not normally distributed and so the Wilcoxon Signed-Rank Test was applied, whilst the rest utilised Student’s t-test. Although the small sample size is acknowledged, we found the SR version performed better than the VR version with significant differences and medium to large effect sizes for *Sensory and Imaginative Immersion* ( $t(16) = 3.55, p = 0.003, \text{Cohen's } d = 0.861$ ), *Flow* ( $t(16) = 3.26, p = 0.005, \text{Cohen's } d = 0.791$ ), *Challenge* ( $t(16) = 2.16, p = 0.046, \text{Cohen's } d = 0.524$ ) and *Positive Affect* ( $t(16) = 4.81, p \leq .001, \text{Cohen's } d = 1.167$ ), but not for *Competence* ( $z = -1.102, p = 0.270, \text{Cohen's } d$

= 0.385), *Tension/Annoyance* ( $z = -0.816$ ,  $p = 0.414$ , Cohen's  $d = 0.283$ ) and *Negative Affect* ( $z = -1$ ,  $p = 0.317$ , Cohen's  $d = 0.348$ ).

**Table 15** - Results for the Game Experience Questionnaire (Ijsselstein et al., 2013). Better scores highlighted in bold.

	<b>Substitutional Reality (SR)</b>		<b>Virtual Reality (VR)</b>	
	<b>M</b>	<b>SD</b>	<b>M</b>	<b>SD</b>
<b>Competence</b>	<b>2.78</b>	0.77	2.69	0.74
<b>Sensory + Immersion</b>	<b>2.68</b>	0.72	2.31	0.79
<b>Flow</b>	<b>3.06</b>	0.64	2.56	0.75
<b>Tension/Annoyance</b>	<b>0.04</b>	0.11	0.06	0.18
<b>Challenge</b>	<b>1.61</b>	0.63	1.36	0.64
<b>Negative Affect</b>	<b>0.07</b>	0.12	0.13	0.20
<b>Positive Affect</b>	<b>3.65</b>	0.42	3.32	0.44

### 6.6.2 Spatial Presence Experience Scale (SPES)

The Spatial Presence Experience Scale (SPES) (Hartmann et al., 2016) was used to evaluate the difference in presence between participants playing the SR and VR versions of the sample game. The SPES is split into two categories, Self-Location (how the player felt in the game space) and Possible-Actions (what the player felt they could do in the game space). There are four statements for each category which are scored from *strongly disagree* to *strongly agree* (1-5). These generate separate scores for each category but also result in a total score for presence. The results can be found in **Table 16** and show an increased level of presence in both categories and the total score for the SR version of the game. Paired Sample t-tests showed significant differences and large effect sizes for *Self-Location* ( $t(16) = 4.89$ ,  $p \leq .001$ , Cohen's  $d = 1.19$ ), *Possible Actions* ( $t(16) = 4.50$ ,  $p \leq .001$ , Cohen's  $d = 1.09$ ) and *Total Presence* ( $t(16) = 5.24$ ,  $p \leq .001$ , Cohen's  $d = 1.27$ ).

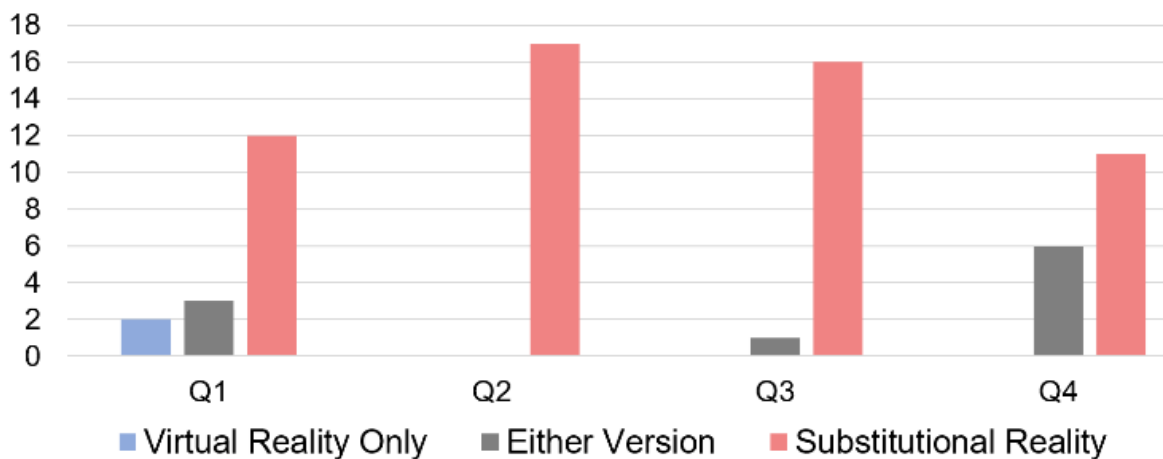
**Table 16** - Results for the Spatial Presence Experience Scale (Hartmann et al., 2016) for Self-Location (SL), Possible-Actions (PA) and Total Presence (Total). Higher total scores highlighted in bold.

	<b>Substitutional Reality (SR)</b>			<b>Virtual Reality (VR)</b>		
	<b>SR First</b>	<b>VR First</b>	<b>Total</b>	<b>SR First</b>	<b>VR First</b>	<b>Total</b>
<b>SL</b>	4.42	4.50	<b>4.46</b>	3.28	4.06	3.65
<b>PA</b>	4.31	4.38	<b>4.34</b>	3.06	3.69	3.35
<b>Total</b>	4.36	4.44	<b>4.40</b>	3.17	3.88	3.50



### 6.6.3 Comparison Questionnaires

After participants had played both versions of the game they were given a comparative questionnaire to complete (see **Table 17**) with the results presented in **Figure 78**. They show that for all statements, participants preferred or agreed more with the statement for the SR version of the game. When asked which version they found more fun (Q1) 70.6% found SR more fun, 17.6% found either version fun but only 11.8% found the VR version to be more fun. The results show that 100% of participants found the SR version to be more engaging (Q2). The results also show that 94.1% of participants felt they had a greater feeling of presence in the SR version (Q3), with 5.9% feeling either version delivered the same level of presence. Finally, when asked which version of the game they would prefer to play (Q4), 64.7% of participants would prefer to play SR again, whilst 35.3% were happy to play either version again.



*Figure 78 - Comparative questionnaire results. Questions in Table 17.*

*Table 17 - Comparative Questionnaire Questions.*

	<b>Questions</b>
<b>Q1</b>	I found the following version of the game more fun to play.
<b>Q2</b>	I found the following version of the game more engaging to play.
<b>Q3</b>	I felt like I had more presence (feeling of being in the environment) in the following version.
<b>Q4</b>	I would prefer to play the following version of the game.

### 6.6.4 System Usability Scale (SUS)

The SUS consists of ten usability questions each with five response options. Each completed questionnaire is converted into a score of 100, where a score of 68 or more is considered above average and an acceptable system (Brooke, 1996). The results (summarised in **Figure**

79) show no significant differences ( $t(16) = 1.94, p = 0.070, \text{Cohen's } d = 0.471$ ) between *Meshing* and *Outline* and both were accepted by the participants. *Meshing* scored an average of 81 and a standard deviation of 13. Three participants scored it lower than the acceptability score of 68, with a minimum of 50 and maximum of 100. *Outline* scored an average of 69 with a standard deviation of 20. Six participants scored it lower than the score of 68, with a minimum of 23 and a high of 98. Whilst both techniques were accepted, *Outline* was clearly more polarised in its responses and this is supported by the adjective ratings from Bangor et al. (2009), which show that *Meshing* would be described as Good, whilst *Outline* as OK. The SUS scores for the VR level editor component of LevelEd SR were positive with an average score of 77 and a standard deviation of 16. Only three participants scored it lower than the score of 68, with a minimum of 45 and a maximum of 100. The adjective rating for the system would be described as Good.

Whilst there is not enough data to demonstrate statistical significance, when looking at the ages of participants, the *Outline* SUS scores drop as the ages of participants increase, with those aged 18-39 scoring it an average and median of 74 and those aged 30-59 scoring it an average of 60 and median of 68. This could be the result of younger participants having more experience and confidence with smartphone AR, than those in the older category, but there is not enough data to definitively confirm this.

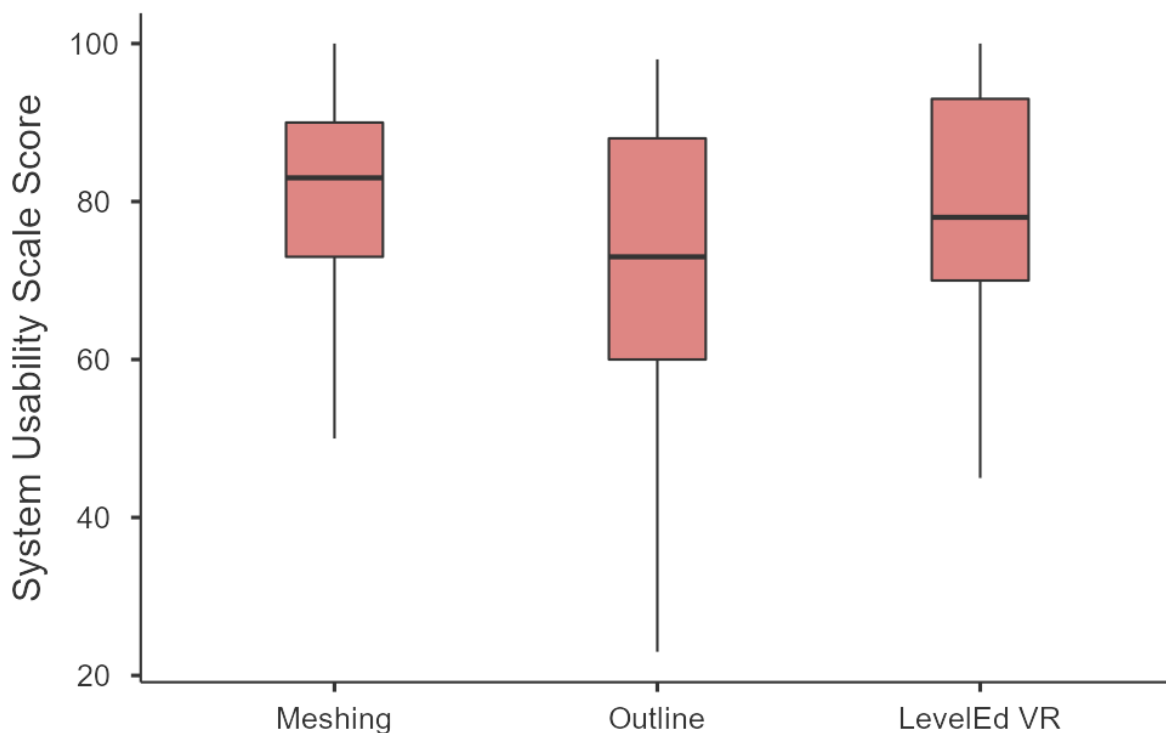
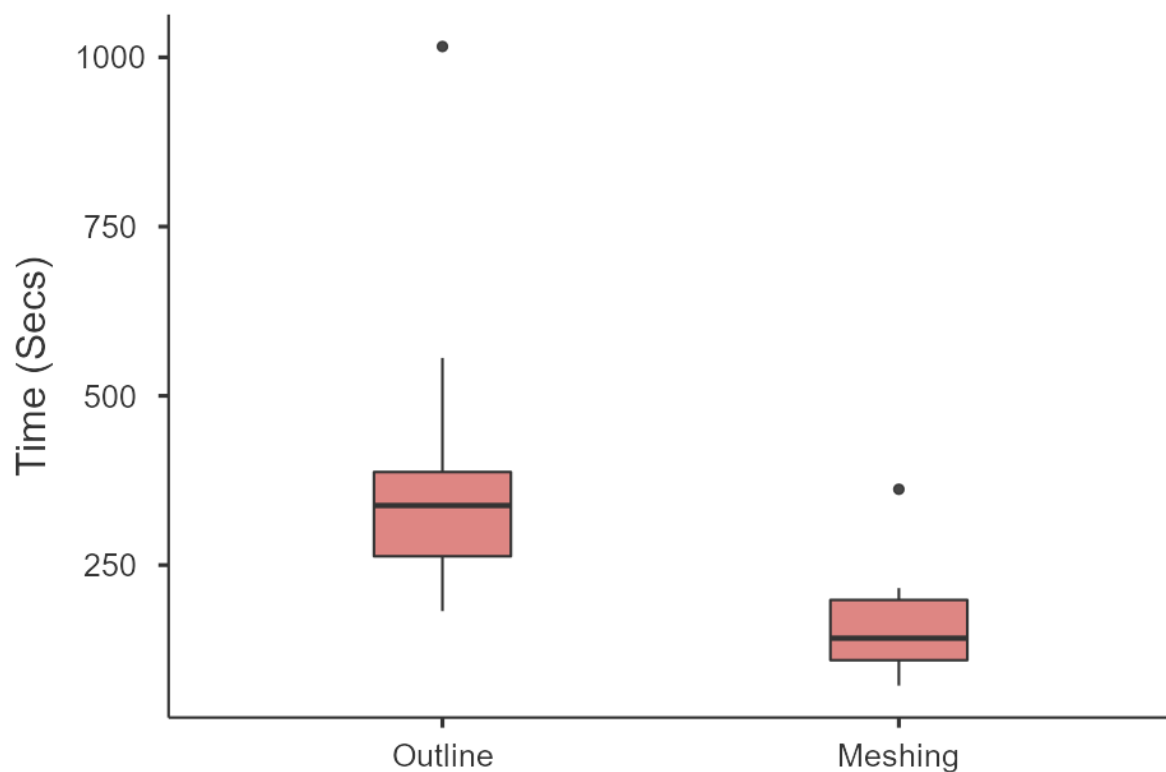


Figure 79 - Results from the System Usability Scale questionnaire (Brooke, 1996).

### 6.6.5 Completion Time

The completion time for mapping the space with both *Outline* and *Meshing* was also recorded by the application. Times were averaged for 15 of the participants as the first two participants' times failed to record, this was fixed for the later participants. The results (summarised in **Figure 80**) show that on average *Outline* (6 minutes and 16 seconds) took just over twice as long to complete the task than *Meshing* (2 minutes and 39 seconds). When removing outliers, *Outline* averaged 46 seconds quicker and *Meshing* averaged 14 seconds quicker. Shapiro-Wilk analysis showed that the times were not normally distributed and so the Wilcoxon Signed-Rank Test was applied. The times were significantly different between both techniques ( $z = -3.351$ ,  $p \leq .001$ ), however, both were still quick at completing the task.



*Figure 80 - Completion time results for both Outline and Meshing.*

### 6.6.6 Simulator Sickness Questionnaire

The SSQ (Kennedy et al., 1993) was administered to see if there were any simulator sickness effects resulting from using the VR level editor. The results show on average low post-completion scores across the categories (**Table 19**) and no one reached the threshold for slight based on the reference totals (**Table 9**) suggesting the effects of simulator sickness was limited. The most reported symptoms were sweating (reported by 6 participants), eye strain (5) and fatigue (4). It is worth pointing out the pre-test results were of a similar level (**Table 18**).

Table 18 - LevelEd SR Pre SSQ-Evaluation Findings

	<b>Nausea</b>	<b>Oculomotor</b>	<b>Disorientation</b>	<b>Total Score</b>
<b>Mean</b>	6.73	7.13	7.37	8.14
<b>SD</b>	8.10	9.08	13.13	9.92
<b>Min</b>	0.00	0.00	0.00	0.00
<b>Max</b>	28.62	30.32	41.76	29.92

Table 19 - LevelEd SR Post SSQ-Evaluation Findings

	<b>Nausea</b>	<b>Oculomotor</b>	<b>Disorientation</b>	<b>Total Score</b>
<b>Mean</b>	7.86	7.58	12.28	10.12
<b>SD</b>	9.68	10.38	16.24	11.20
<b>Min</b>	0.00	0.00	0.00	0.00
<b>Max</b>	28.62	37.90	41.76	37.40

### 6.6.7 LevelEd SR System and Workflow Feedback

To better understand participants' experience with the workflow and individual components, a feedback questionnaire (**Table 20**) was given to participants at the end of the study with results shown in **Figure 81**. A 7-point Likert scale was used ranging from *strongly disagree* (1) to *strongly agree* (7). The questionnaire consisted of 12 items and the value for Cronbach's Alpha for the questionnaire was  $\alpha = .763$ . According to George and Mallery (2003), this is considered an *Acceptable* level of reliability.

Questions 1 to 4 related specifically to the VR level editor. Responses to Q1 show that participants found the menu system easy to use and understand. Five participants strongly agreed with the statement and a further ten agreed. Only two participants felt it could be improved by selecting somewhat agree. Q2 and Q3 show that participants were also happy with the usability of the prototyping/geometry/model tools and found they had sufficient functionality to create the level they wanted. This is a good result as the participants had a mixed range of previous experience with VR. Results for Q4 showed that most participants found the spatial scripting tools easy to use with eight participants selecting strongly agree and three also agreeing. However, five participants selected somewhat agree and one selected neither agree nor disagree. This is still a positive result as some participants had no or little game development experience. Question 5 and 6, shows a similar result to the SUS for the *Outline* and *Meshing* tools with participants agreeing that *Meshing* ( $M = 6.06$ ) was easier to use than *Outline* ( $M = 5.24$ ).

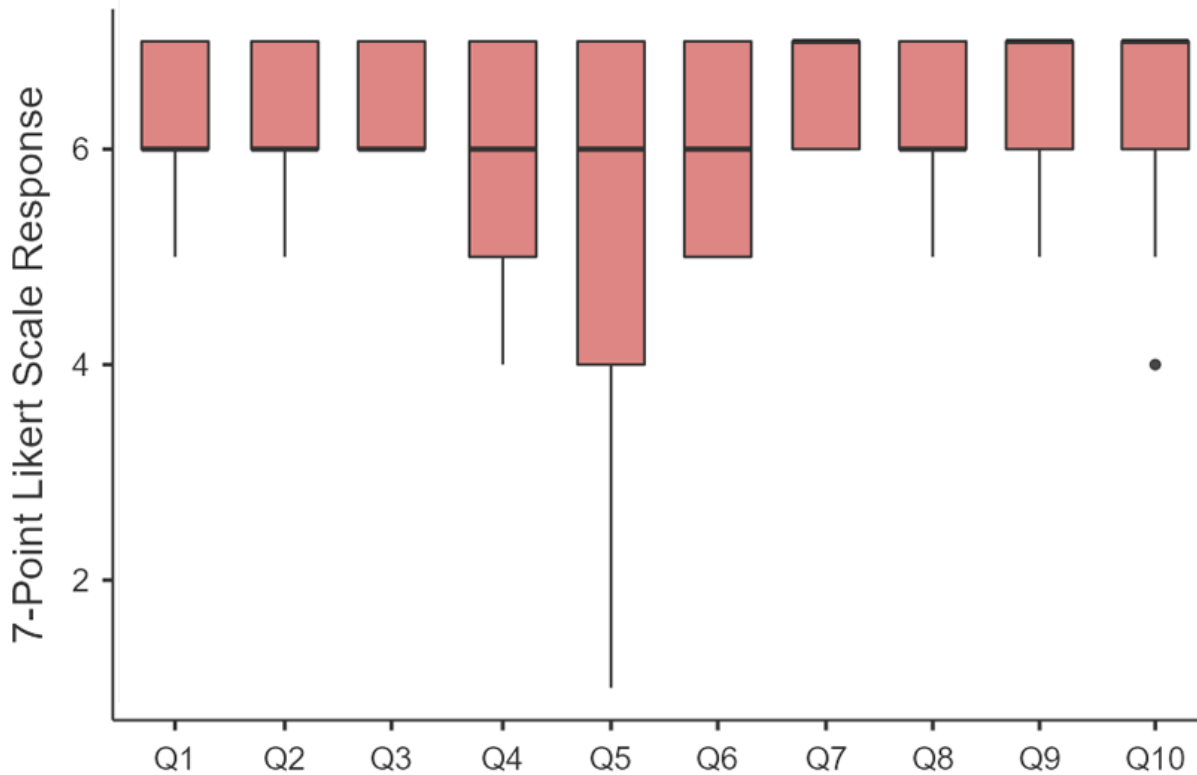


Figure 81 - Results from general feedback questions (Table 20) regarding the LevelEd SR workflow and individual components.

Table 20 - LevelEd SR Workflow General Feedback Questions

	Questions
Q1	The menu system was easy to understand.
Q2	I found the prototyping/geometry/model tools easy to use.
Q3	The prototyping/geometry/model tools were sufficient for me to build the level I had envisioned.
Q4	I found the scripting system easy to use.
Q5	I found the LevelEd AR application without LiDAR scanner (manual shape creation) easy to use (Outline).
Q6	I found the LevelEd AR application with LiDAR scanner (automatic mesh scanning) easy to use (Meshing).
Q7	I found the substitutional reality capture of the real-world space was accurate enough to play the game.
Q8	I found the LevelEd SR workflow (from room scanning to a playable level) easy to use.
Q9	I would like to use this system again to create substitutional reality game levels.
Q10	I found playing my substitutional reality game levels fun and engaging.
Q11	I would prefer to use the following system for capturing a space for substitutional reality games.
Q12	If you did not have a device with a LiDAR scanner, would you be happy to use the LevelEd AR version of the application that requires manual shape creation in order to take part in substitution reality?

Question 7 to 10 relate to the SR components specifically and the LevelEd SR workflow in general. Q7 shows that participants found their mapping data to be accurate enough for them to create a level and play the game with nine strongly agreeing and seven agreeing. One participant's response was removed as they inadvertently used the researchers room mapping data. Responses to Q8 show that all participants found the general SR workflow (from scanning a room to creating a playable level) easy to use with six strongly agreeing, nine agreeing and two somewhat agreeing. Q9 showed that participants were happy to use the workflow again to create SR game levels and Q10 showed that participants found the levels they created fun and engaging to play. These are all good results as very few participants had used a workflow like this before.

The final two questions relate to using the *Outline* and *Meshing* techniques. Q11 asked which system participants would prefer to use to map a space. Eight participants preferred *Meshing*, with 6 preferring *Outline* and three happy to use either. This is surprising after seeing the lower SUS and feedback scores for *Outline*, but this question was asked after the participants had viewed and used their data to create their level and they may have found the *Outline* data easier to work with when in the VR editor.

Finally, it was explained to participants that *Outline* was more widely available to use due to not requiring a LiDAR sensor. Q12 shows that sixteen out of seventeen participants were happy to use the *Outline* technique if *Meshing* was unavailable to them.

## 6.7 Discussion

### 6.7.1 Mapping Spaces

The first step of the LevelEd SR workflow is to create a digital map of a space. Two versions of the mapping application LevelEd AR were developed to enable players to map a space. The results from the SUS show that players found the *Meshing* tool more accessible and usable, than the *Outline* tool. However, both did manage to meet the 68-score threshold for an acceptable system (Brooke, 1996). This is not unexpected as the *Outline* system requires more engagement and interaction from players and requires a better sense of perspective in an AR view. The *Meshing* system, meanwhile, is more passive as players sweep the phone camera around the room to map the objects. Q11 of the general feedback questionnaire showed that 35.3% preferred to use the *Outline* tool, 17.7% were happy to use either and 47.1% preferred the *Meshing* tool for mapping a space. It was not expected that so many people preferred to use *Outline* or were happy to use either after looking at the SUS results. This could be because Q11 was asked after participants had used the data from *Outline* in the VR level editor and found the data to be useful. This matches written comments from participants such as "*The LiDAR scanning was much faster to use but I preferred the manual*

*shape creation because it gave more accurate results” and “System with LiDAR scanner showed greater potential but I felt more ‘confident’ in the manual system due to clearer and more consistent visual feedback, as the objects were created.” as well as “I definitely preferred using the manual mesh tool because I felt it gave greater control and accuracy with the shape of the meshes”. The lack of complex furniture in the test space will have had an impact on this analysis and is something that should be explored further.*

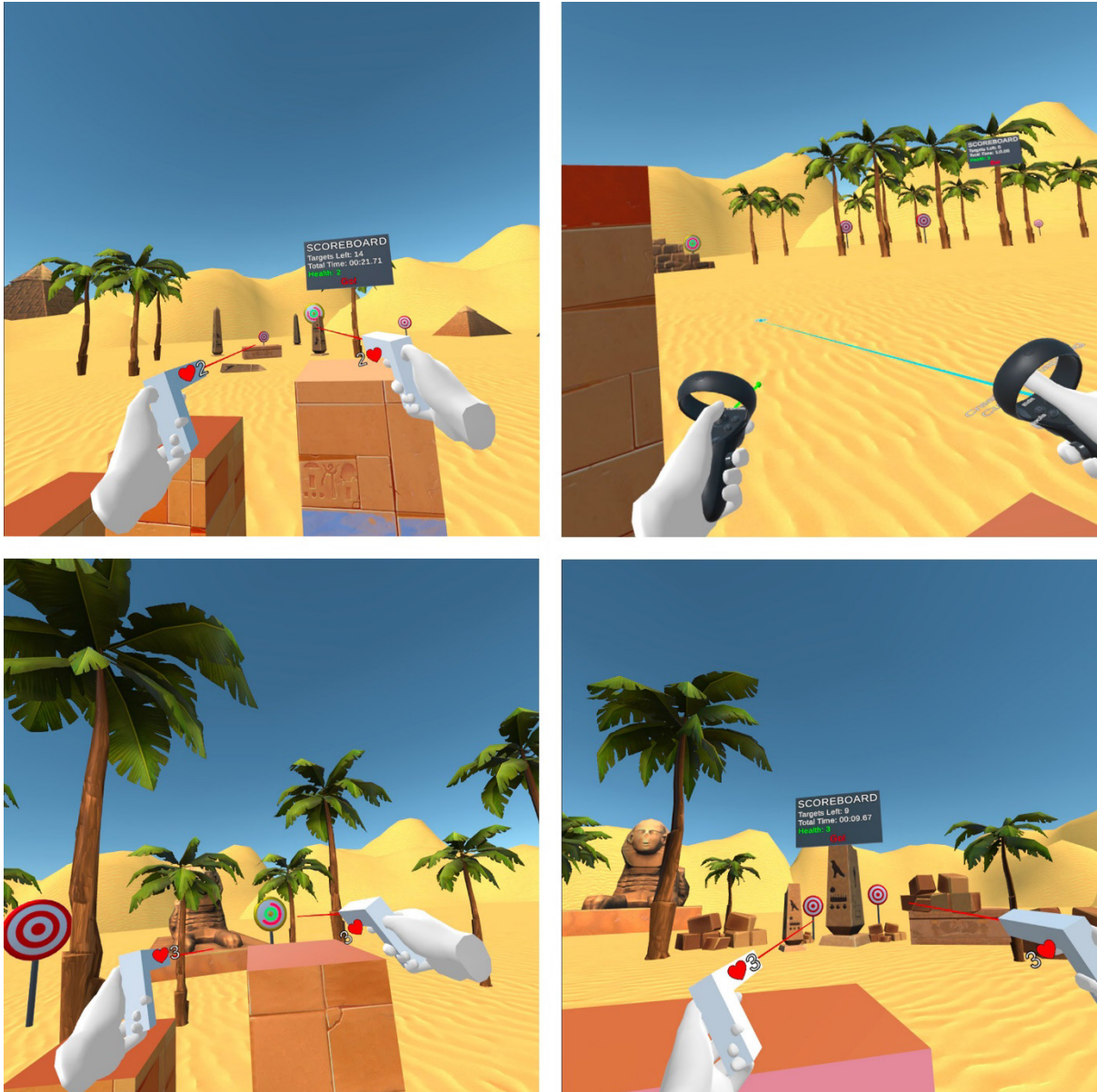
Despite these comments, there is clearly an issue with the usability of *Outline* for some participants. Several participants struggled to position the marker point at their intended location, and this led to frustration and longer completion times. The marker is positioned by a raycast out from the camera forward vector, those that struggled with the system instead tended to try to move the marker by tilting the device, rather than aiming the camera. Q8 of the SUS supports this, which states “*I found the system very cumbersome to use*” which scored poorly for *Outline* ( $M = 2.23$  out of 5) compared to *Meshing* ( $M = 1.58$ ). Participants were given a brief demonstration of the system, but a clearer on-device tutorial may help with this misunderstanding. One possible solution could be to use the experimental Passthrough API (Oculus, 2022) on the Meta Quest to enable a similar technique to *Outline* but directly on the device. However, the Passthrough API is new and experimental, and this would also limit the system to only supporting Meta Quest 2 in the future rather than any other standalone systems with 6-DoF controllers which would be compatible with our current workflow.

### **6.7.2 Developing Substitutional Reality Levels**

The second step of the LevelEd SR workflow is to register the mapped data with the real-world location and then to create an SR level through the LevelEd VR system. The results from the SUS show that participants found the system to be acceptable and would have been rated Good (Bangor et al., 2009). This is confirmed by positive ratings on the feedback questionnaire (**Table 20** and **Figure 81**), with questions Q1 ( $M = 6.18$ ), Q2 ( $M = 6.24$ ), Q3 ( $M = 6.47$ ) and Q4 ( $M = 6.06$ ) all scoring highly in regards to the menus, level creation tools and the spatial scripting system. This shows that participants were confident using the VR level editor. This is a good result as the participants had mixed experience of VR and game development.

Participants spent up to 16 minutes creating their own game level. During this time, they were able to acquaint themselves with the tools and see how they could be used to create a game level. **Figure 82** presents some example game levels created by four different participants and shows the variation in their level designs despite the limited selection of assets available to them.





*Figure 82 - Four game levels created by four different participants.*

The results show that participants were happy with the overall LevelEd SR workflow. Participants were happy with the accuracy of the data they captured, evidenced with Q7 (M = 6.59) of the general feedback questionnaire (**Table 20** and **Figure 81**). This is a key part of the workflow to enable SR and an encouraging result. Further exploration of this component with more varied environments is an important next step. The overall workflow was rated positively with Q8 (M = 6.24), Q9 (M = 6.47) and importantly participants found their levels fun to play with Q10 (M = 6.47). This bodes well for further development of LevelEd SR. The workflow has been shown to not only improve the player experience, but it also helps towards solving the issue of lack of space for room-scale experiences. Using this workflow, players can bring objects into the game world that would normally restrict their play space, such as a bed in a bedroom. It is envisioned that not all games would need to utilise the level editor

aspect as some games could use the data captured from *Outline* or *Meshing* to create procedural content.

The results of the SSQ show that there was minimal sickness from participants, which is a good result (see **Table 18** and **Table 19**). However, they were slightly higher than what was expected for a system that only affords locomotion through natural walking. This could be a result of the test conditions due to participants having to wear a mask throughout the study and the hot weather at the time. The pre-test (see **Table 18**) shows similar values and the highest reported symptom was sweating. Researchers completing studies that require participants to wear a face covering may find these values useful going forwards.

Whilst LevelEd VR was rated positively, there are a few areas that need improving. One example is when observing players, it was clear that it was sometimes difficult for them to work accurately at greater distances. The current system only affords movement within the players real-world space and one participant stated they would like to “*Move around further than possible on foot*”. There is potential to allow players to move outside of that space (via teleportation/drag-world) so they can edit closer up, but this would present challenges on how to ensure players know what is physical and what is not. This could be dealt with by leaving behind a silhouette of physical objects, but it would need some testing to ensure players do not get confused between physical and digital objects.

### **6.7.3 Game Experience and Presence**

The results suggest that there is an improved game experience when playing the action target shooting game in SR over VR. The SR version shows an improvement and significant difference (all with a medium or large effect size) in sensory and imaginative immersion, flow, challenge, and overall positive affect (see **Table 15** and section **5.3.1.1**). The positive affect category was very encouraging as the SR version scored 3.65 out of a possible 4.00 with VR scoring 3.32. This is a good result and shows that the game experience was not hampered by the inclusion of objects in the space but was in fact enhanced by it. This is further backed up by the comparative questions which all favoured the SR experience. In particular Q1 showed that the majority of participants (70.6%) found SR more fun (see **Figure 78 – Q1**) and Q2 which showed that all participants found the SR version more engaging (see **Figure 78 – Q2**).

The results also show an increased level of perceived presence by participants in the SR version over the VR version. This was present in all categories of the SPES, such as Self-Location, Possible-Actions and the Total Presence score (see **Table 16**) with all showing significant differences.

This is further emphasised when looking at differences between which version was completed first. There was a minimal increase for SR scores when it was completed second, additionally VR scores were reduced when SR was completed first. This suggests that those completing VR first were generally impressed with the experience but may not have scored VR as high if they had played it after SR. However, the sample size for these sub-groups is too small to be significant, but a trend can be observed, and it may also suggest any learning effect was minimal.

The increase in presence was also supported by Q3 of the comparative questionnaire with 94.1% finding SR to have more presence (see **Figure 78** – Q3) and the improved score on the GEQ for sensory and imaginative immersion. This agrees with earlier research findings around presence and passive haptics (Eckstein et al., 2019; Meehan, 2002), however, these results show that presence is increased for entertainment games as well as the serious games and applications shown in those studies. This is also confirmed in more recent work that looked at ‘bizarre’ experiences and showed a trend towards higher self-location when visual changes were made to the digital proxies (Denzer et al, 2022).

This a result that somewhat contradicts recent work that looked at smart SR entertainment games and presence, which showed that smart SR and passive haptics did not significantly increase the perceived presence over a VR experience (Steinhaeusser & Lugin, 2020). It is unclear why these results differ, but it could be due to the genre and type of game. The Horror Laboratory and Forest Cabin (Steinhaeusser & Lugin, 2020) appear to be slower paced and focus more on participants touching objects with their hands or additional simulations such as heat lamps and fans. The sample game in this study was more action orientated, which required quick movements and less precise interactions with the furniture for passive haptics. Whilst most players briefly tested the accuracy of the digital proxies with their controller/hand before starting the game, once playing the game, most interactions were with body parts other than the fingers as players sat on or leant against objects. Whilst findings are mixed on the impact of levels of mismatch (LoM) on presence (Eckstein et al., 2019; Simeone et al., 2015), these interactions may have reduced the impact of any mismatch.

Both the improved game experience and increased presence suggest that there are benefits to passive haptics delivered through SR. This shows that there is good reason to continue to explore SR workflows. Q4 on the comparative questionnaire confirms this with 64.7% of participants preferring to play the SR version and 35.3% happy to play either.

#### **6.7.4 Additional Questions**

When developing a SR system, it is important to think about the privacy of the player and their home. It is vital that players are informed about how their mapped space data will be used and

stored. As this system is a proof of concept, there are no systems in place to deal with this and the data is stored on a webserver. This is particularly pertinent to tools that capture detailed meshes of a space (like the *Meshing* tool) as that could include more detail about a player's home than a tool such as *Outline*. Taking a system like this further should include processes to deal with this.

Safety is another aspect that should be thought about carefully. As LevelEd SR relies on the player to create an accurate map of the space, there is the potential for injury if they have not mapped it accurately enough or missed out objects. The same can be said for automated processes that may miss something during mapping. Detailed safety warnings, tutorials and asking players to carefully check the space could assist with this, but it is not a complete solution. Designers of SR level design systems, such as LevelEd SR, should be advised to discourage players from including objects within a player space (more so on the ground) that do not have corresponding real-world objects to avoid the issue of players thinking something should be there and falling/injuring themselves.

## 6.8 Conclusion

This chapter shows that there are potential benefits to playing entertainment games using SR that enable passive haptics. To do this, an action target shooting game was developed that could be played in both SR and VR. The SR version incorporated physical objects in the space within the virtual environment through digital proxies, whilst the VR version was in a similarly sized empty space. The LevelEd SR system and workflow that enables SR level design for players using widely available consumer hardware was also presented. This workflow allows players to experience and create their own SR game levels without the need of a PC or specialist game development software.

A study comparing the two versions of the game was conducted and presented in this thesis. Participants reported that they had an improved game experience and an increased perceived level of presence in the SR version of the game over VR. The LevelEd SR system and workflow was also evaluated with participants from a mixed VR and game development background. Participants created their own SR game levels from mapping their own space to creating an SR game level in the VR level editor. Participants were accepting of the components in the workflow and overall rated the system positively with a desire to use the system again.

# Chapter 7 Conclusions and Future Work

## 7.1 Research Question

The landscape of AR and VR technologies is constantly evolving. Since the recent resurgence of VR with the initial Oculus Rift Kickstarter campaign, there have been significant improvements in the technology. Although VR has yet to enter the mainstream, it is closer than ever before, as evidenced by the considerable sales of devices such as the Meta Quest (Heath, 2023) and the financial success of many game titles released for the system (Lang, 2021a). Whilst head mounted AR technologies have not grown in popularity at a similar rate, smartphone-based AR has quickly been embraced by many as the technology has been integrated into smartphone applications. At the onset of this thesis, it was apparent that AR and VR technologies would continue to improve, but there was limited research into how these technologies could be utilised and what potential benefits they could bring to the level design process and workflow. This thesis has explored this further to answer the following research question:

*Can augmented reality and virtual reality technologies be used to improve current level design workflows or establish new workflows that enable new experiences?*

To answer this research question, this thesis contributes three different workflows, along with seven hypotheses which were tested with three original software implementations. The results of these have been summarised below.

## 7.2 LevelEd AR and Workflow 1

Workflow 1 was planned and developed to explore how AR technology could be used to create VR game levels that were true and accurate to scale based on real world locations. LevelEd AR was developed to enable this workflow and used smartphone AR technology to capture a space for use in a VR game level (see **Figure 43**).

### 7.2.1 Hypothesis

The following hypothesis was devised and LevelEd AR was used to test this hypothesis in Study 1:

**H1:** *An augmented reality smartphone application is an acceptable digital alternative (in terms of a trade-off between speed and ease of use) for capturing accurate room scale data.*

The results from Chapter 3 show that **H1** was proven true. Participants were able to use LevelEd AR to create reasonably accurate captures of real-world locations, quicker than using a tape measure (see **Figure 48**), more accurately than the Structure Sensor (see **Figure 47**)

and scoring well in the SUS questionnaire (average score of 74, which is above the 68-usability threshold). These results show that a smartphone AR based system, such as LevelEd AR is an acceptable digital alternative to a tape measure and a similar system such as the Structure Sensor.

The main study above focused purely on the usability of LevelEd AR to create the initial digital maps of a real-world location. However, the virtual environment workflow preliminary study shows that there is potential to use the captured digital maps to create detailed serious games experiences (see **Figure 50**). Whilst this type of workflow is niche, this preliminary study demonstrates a usable and accessible way to achieve an accurate to scale virtual environment for use in serious games. This early work also inspired the final trajectory of this thesis, which investigates substitutional reality level design and a suitable workflow to enable it.

### **7.2.2 Limitations**

One limitation of the study used to evaluate LevelEd AR was the low complexity of the environment that was measured by participants. This was done intentionally to ensure measurements could be captured across all three instruments and that results were consistent between participants and tasks. However, further work should explore the accuracy of the system in more natural locations and with more complex furniture, such as a users' bedroom or place of work, such as an office. There is also a limitation due to the limited sample size of participants. Increasing this sample size (including more balanced age groups) would allow for deeper analysis of how different age groups interact with the system.

There are currently several limitations with the software presented in Chapter 4, such as current marker creation being limited to the initial ground plane, so areas like stairs or ramps are not currently supported. Objects can not currently be mapped on top of previously mapped objects, such as mapping a computer screen on top of a desk. Finally, the version presented in Chapter 4 does not support undoing the placement of any markers, this means users must either keep mistakes or restart the mapping process. To take this system beyond a prototype, these features would need to be added, some of which have been completed and are noted below.

### **7.2.3 Actions Identified and Lesson Learnt**

Several additional features were added to improve LevelEd AR for use within the LevelEd SR system and Workflow 3. One of the largest improvements was the inclusion of an undo button that could roll back all changes made for a particular object currently being mapped. The previous version required users to start from the beginning if they went wrong, which was not ideal. The ability to tag objects as they were created was also added. Tags could be selected from a common list of furniture and users could also add their own with a text box. These tags

were added as the name of the GameObjects when generated in Unity, however, it would also be useful to use these as labels in LevelEd SR to help users remember what each digital map object represents. Another new feature added was the ability to generate a detailed mesh of a real-world location using the LiDAR scanner. This creates a more detailed digital map of a space but could be improved by allowing users to manually select polygons to either keep or remove from the capture and by also capturing the texture and colour detail of the objects. Whilst the fidelity is unlikely to match the bespoke or sourced models seen in **Figure 50**, it could allow for a more automated process for generating digital maps from real-world locations for use within serious games.

Whilst LevelEd AR was shown to be usable in both Study 1 and Study 3, there were clear issues for some users with how to aim the marker points when using the *Outline* technique. Several participants in both studies struggled to place the marker in the desired location. This was primarily down to how they tried to aim the marker by tilting the device to move it, rather than aiming it with the centre of the camera. As noted, there were some improvements made to LevelEd AR *Outline* before Study 3, but one area that was not changed was the design of the AR marker which was still represented by a simple yellow cube. A future change would be to explore if the design of the marker could help with users' understanding of how to aim the marker. This could be through a different marker shape or even a broader screen-based reticule for aiming.

## 7.3 LevelEd VR and Workflow 2

Workflow 2 was planned and developed to explore how VR technology could be used to improve existing VR game level design workflows. It achieves this by allowing VR game levels to be built whilst working in VR (see **Figure 51**). Several potential areas of improvement were initially identified, such as a better sense of scale awareness and improved iteration times prior to development of the system.

### 7.3.1 Hypothesis

LevelEd VR was developed to enable level designers to create game levels whilst in VR and to test the following three hypotheses:

**H2:** *A virtual reality-based level design workflow is accepted (based on the System Usability Scale) as an alternative for developing VR game levels by experienced participants.*

**H3:** *A virtual reality-based level design workflow is preferred (based on a comparative questionnaire) as an alternative for developing VR game levels by experienced participants.*



**H4:** *A virtual reality-based level design workflow is shown to deliver an improved sense of scale awareness (based on a comparative questionnaire).*

The results from Chapter 5 show that **H2** was proven true as experienced participants tested LevelEd VR and Workflow 2 and accepted the system with an average score of 79 (vs. 75 for Unity Desktop) on the SUS. **H3** was also proven true as participants found LevelEd VR and Workflow 2 preferable to existing non-VR based workflows based on a comparative questionnaire (68.75% preferred and 18.75% were happy to use either). Participants also agreed that they could build levels with a better sense of scale using LevelEd VR based on a comparative questionnaire (82.25% agreed), proving **H4** also to be true.

These results suggests that a VR-based level design workflow and editor are accepted and preferred to a desktop-based system and that it includes additional benefits such as improved sense of scale awareness.

### **7.3.2 Limitations**

One limitation of this study is that whilst care was taken to try to ensure the only independent variable was that one system was desktop-based editor and the other an immersive VR editor, the different editing tools may introduce confounding variables. Whilst the desktop and immersive VR editor shared a similar toolset, the way they are used and interacted with are different. This could introduce compounding variables beyond the initial independent variable. This is somewhat to be expected as the study was developed to compare two different workflows, which includes the different tools, but it should still be taken into consideration when viewing the study results.

It's worth noting that there are additional limitations with the study regarding the limited sample size, short session time, and the simple design of the sample game. Longer testing sessions and looking at more complex games (including non-first-person games) may help to reveal additional benefits but may also highlight unforeseen issues, such as an increase in simulator sickness. A longer testing session would also help to understand possible fatigue from using a VR system over longer periods of time. There is potential for fatigue due to the more physical nature of working in VR (bigger arm/body movements than a mouse and keyboard). Eye fatigue could also be a problem over longer periods of time. Future testing should also look at testing with a larger sample size and possibly with inexperienced users, such as potential players.

### **7.3.3 Lessons Learnt**

Overall, the LevelEd VR software has been successful in meeting the requirements of Workflow 2. Whilst it is not yet feature complete, several additional lessons have been learnt

whilst creating and testing the LevelEd VR system. For example, there were concerns surrounding simulator sickness early on in development of the system. Several features were included to attempt to reduce the chance of simulator sickness such as multiple movement schemes and limiting peripheral vision based on linear and angular velocity changes. These appear to have limited the effect of simulator sickness and would therefore be recommendations for any VR system that requires substantial movement around a scene.

It is clear from both Study 2 and 3 that there is a need for extensive tutorials for inexperienced users to successfully experience complex software, like LevelEd VR. In both studies, participants had a period of training delivered by the author to ensure they understood the system before using it. Without this support it is unlikely participants would have been as comfortable with the system as they were. This is evident in the responses to the SUS questionnaire where in Study 2, Q4 resulted in a higher response (out of five, where lower is better) for LevelEd VR (1.75) than Unity Desktop (1.56) and the VR level editor scored higher again when used by participants with a wider range of game development experience in Study 3, with it scoring 2.4. It is therefore advisable that any complex system like LevelEd VR be launched with clear step by step tutorials outlining how to use the system, with these ideally delivered in an immersive way.

Whilst it is clear from the results that participants preferred to use the LevelEd VR system to develop VR game levels, this was not the case for more traditional games. Here, 37.50% would prefer to use either system, with only 12.50% outright preferring to use LevelEd VR compared to 50% preferring to use Unity Desktop. It was anticipated that users may find the tools and natural motion-controlled interactions compelling enough to use for general level design, but this does not appear to be the case. This would suggest that the key advantage to the LevelEd VR editor is the improved sense of scale awareness and potentially the lower friction and time taken to get into testing. There is a potential for this to change as the public start to use VR at an earlier age and grow up with the technology. This is evident with touch screen devices, such as smartphones and tablets, where there is now a generation of users who have grown up with these devices and now find them integral to their lives and their preferred way to consume and create content. If VR use continues to grow, we may start to see a similar trend whereby we see a generation of VR native users.

### **7.3.4 Future Work**

The positive results discussed above suggest that continuing development on LevelEd VR would be worthwhile. As a development tool, the system is currently suited to the prototyping stage, rather than taking a game level from concept, right through to the finished article. As it stands, there are several areas that LevelEd VR does not yet cover and supporting all use

cases is not trivial. As the system clearly benefits the sense of scale awareness and iteration time, using it to initially blockout a level and test it (with some gameplay), could be a useful tool for level designers of VR game levels. Once past that blockout stage they could utilise their game engine, like Unity, to continue development. The current version is available as a proof-of-concept that can be accessed on GitHub (**Appendix B3**). However, if LevelEd VR were to be taken further forward, beyond proof-of-concept, the following improvements would be made.

- The ability to transition from editing a level in LevelEd VR, taking that work back into a game engine (like Unity) so additional work can be done, before taking it back into LevelEd VR would be a useful addition. Currently, it is possible to load a level created in LevelEd VR into a Unity scene so work can continue there, but the ability to move back and forward would require additional work.
- The ability to automatically detect potential user editable variables to expose to the scripting system for a prefab would make the system more flexible and quicker to setup. As it stands, developers must identify variables they want to be able to edit in the VR editor and use the template scripting nodes to create a customisable prefab. It would be ideal if this can be automated which would mean more flexibility for developers and less time setting up custom prefabs. Some early exploratory work has been done on this, but nothing implemented.
- It would also be useful if the ability to add prefabs to the system that have interactive gameplay properties could be included without additional work. Currently, these prefabs need to be aware of which mode the scene is in (editor or play mode) with additional code. As with the point above, it would be useful if this can be automated. Again, some early exploratory work has been done on this, but nothing implemented.
- The system would also need to include the ability to add sounds, lights, include additional scripting nodes, terrain editing and the ability to generate navigation meshes for AI.

Adding the above features would result in a more robust VR level editor for professional level designers. One of the other aims of LevelEd VR was to also enable user generated content for players. There have been several level editors recently announced for new or existing VR games, such as Population One (Hayden, 2022b) and Guardians Frontline (Hayden, 2022a). These new releases suggest there is still an audience for user generated content tools, like LevelEd VR. Therefore, the continued development of new features should also keep in mind the need to support players as well as professional level designers.

There are also several opportunities to continue researching this workflow in more depth. Whilst the study took a total of 75 minutes to complete, the total time participants had constructing a level with the tools was 16 minutes. A study that looks at building on this and asking participants to create a larger level over a longer period of time could reveal additional impacts of a level design workflow in VR. Biener et al. (2022) looked at the effects of working in VR over a working week (8-hours a day for 5-days). They found that participants rated working in VR worse than a standard 2D desktop system and many suffered from simulator sickness or similar symptoms. However, participants in this study were carrying out traditional office-based tasks using VR, rather than using fully immersive software, such as LevelEd VR. Therefore, an exploration of a more feature complete LevelEd VR used over a longer period of time would be a good candidate for future research.

## 7.4 LevelEd SR and Workflow 3

The final workflow was planned and developed to explore how both AR and VR could be used to enable a new workflow and experience for players. Both technologies were utilised in this workflow to enable substitutional reality level design by players. This workflow built on the past two workflows and studies. It brought together LevelEd AR, which enables a player to capture their current location and turn it into a digital map and LevelEd VR to enable the creation of game levels that supported substitutional reality *in-situ* (see **Figure 69**).

### 7.4.1 Hypotheses

The software to enable Workflow 3 was named LevelEd SR, and this was used to answer the following three hypotheses:

**H5:** *Substitutional reality game levels offer an improved level of presence (based on the Spatial Presence Experience Scale) over virtual reality.*

**H6:** *Substitutional reality game levels offer an improved game experience (based on the Game Experience Questionnaire) over virtual reality.*

**H7:** *An AR to VR workflow is accepted (based on the System Usability Scale) by consumers to enable user generated substitutional reality game levels.*

The results from Chapter 5 show that **H5** was proven true. When comparing SR to VR game levels, the results from the SPES questionnaire showed significant differences in favour of the SR game level for *Self-Location* (**4.46** vs 3.65), *Possible-Actions* (**4.34** vs 3.35) and *Total Presence* (**4.40** vs 3.50) compared to the VR game level.

**H6** was also proven true as the SR game level also performed higher in *Sensory and Imaginative Immersion*, *Flow*, *Challenge* and *Positive Affect* with significant differences in the

GEQ (see **Table 15**). This is further backed up by the results of the comparative questions, which highly favoured the experience of the SR game level over the VR game level (see **Figure 78**).

The workflow itself was also shown to be accepted by the participants with a SUS average score of 81 for the *Meshing* technique, 69 for *Outline* (which form parts of LevelEd AR) and 77 for the LevelEd SR component (see **Figure 79**). This result confirms **H7** was proven true. The overall LevelEd SR system was also accepted and scored positively in the general feedback questionnaire (see **Figure 81**).

#### 7.4.2 Limitations

There are however several limitations to this work. The small sample size in this study weakens the claims that can be made for statistical significance. Nevertheless, strong trends have been identified. The results suggest that not only does SR have benefits, but that the workflow presented here, which was used to create the sample game experience, is sufficiently effective at creating SR game levels to deliver benefits.

There is also potential for a learning effect for participants between each test completion. Mitigation was attempted by alternating which version of the level (SR or VR) was completed first between participants. Due to the quick completion time of each version along with the mitigation in place, any learning effect should have had limited effect. The SPES results in **Table 16** also suggest that the learning effect was limited for presence.

There are also limitations due to the static nature of the capture technique and no support for moving objects. The location of obstacles remains the same (risking fatigue) unless the player captures the scene again (if something has changed) or moves spaces. This limits the layout of a game level to what is physically available. Whilst levels can be developed beyond the physical obstacles and themed (**Figure 71b** and **Figure 71c**) it is still a limitation that will impact genres that often use many varied environments (RPGs, platformers, etc.).

Developing narrative games is also more difficult for an SR system, as the narrative must work with unseen environments and therefore cutscenes cannot be choreographed based on the environment layout. AR games suffer from a similar issue as they deal with unseen environments. Therefore, techniques suitable for AR games in general and with narratives (such as *Spirit Camera: The Cursed Memoir* (Tecmo Koei, 2012)) could be used in SR, for example story pickup items or artificial intelligence characters.

SR systems that focus on larger passive objects (such as LevelEd SR) are best suited to games with limited narrative focus and that utilise real world objects as an obstacle/barrier. For example, action/casual games like shooters (static targets/moving targets like zombies),

physics games (using surfaces to bounce/reflect/impede), mini golf (furniture as course obstacles), hidden object games (key objects hidden around obstacles) or systems-based games that do not rely on set environments but may benefit from tactile surfaces and larger play spaces (construction/factory simulators or strategy games).

### **7.4.3 Lessons Learnt**

In addition to the above results, several lessons have also been learnt and confirmed from previous studies in this thesis. For example, the *Outline* component of LevelEd AR continues to be divisive, with some participants using it with no issues (with the highest scoring it 98 on the SUS), but some finding it difficult to use (with the lowest scoring it 23). There is also potential for age to play a part in the difficulties experienced by some users, but more research is required to confirm this.

It was also positive to see users ranging from no experience of game development right through to those who consider themselves experts, rating the VR level editor component high on the SUS questionnaire (average of 77) and positively with the general feedback questionnaire responses (see **Figure 66**). Whilst Study 2 showed that users with prior game development experience found LevelEd VR usable, it was promising to see these positive results continue in Study 3 for those with limited or no experience in game development or level design. This bodes well for LevelEd VR and SR as tools to enable user generated content for players in VR or SR games.

The lines between AR and VR devices are also starting to blur, such as with the release of headsets like the Meta Quest Pro, HTC Vive XR Elite and Pico 4. These devices are primarily VR headsets that can also support AR via passthrough cameras and therefore MR experiences. Lessons learnt from developing SR game levels will certainly be useful when developing MR experiences. There is commonality in the need for both SR and MR experiences to make use of the real-world within the game experience (surfaces, furniture, etc.), albeit in different ways and between the tools required to create these experiences for or by players. Therefore, lessons learnt in the development of LevelEd SR will be relevant to those wanting to create MR experiences and tools.

### **7.4.4 Future Work**

The results for LevelEd SR and SR experiences are positive and suggest that creating more SR experiences would be worthwhile to pursue. To enable these experiences there are several updates to the LevelEd SR software and workflow that could be made.

Since the completion of the LevelEd SR study, Meta have enabled developer access to the passthrough cameras to enable MR experiences in their games and applications on Quest

devices (Oculus, 2022). Workflow 3 was developed partially because there was no way to access the passthrough cameras on Quest devices at that time. Therefore, LevelEd AR was used to capture a digital map of a real-world location that could be used to create SR game levels in LevelEd SR. One future improvement would be to utilise the passthrough cameras to enable the capture of the real-world location and obstacles directly using the Quest device. This could make use of the existing geometry creation tools in the VR level editor to map the obstacles in the real-world location that can then be used in the SR game level. This would streamline the workflow as there would be less need to use the separate smartphone AR application, as it can all be done on the HMD.

LevelEd AR does still offer some additional benefits though. For example, LevelEd AR still offers a higher quality presentation from the smartphone camera (4K image) that might be preferred, compared to the lower quality passthrough cameras that suffer from some distortions and artefacts (including the more recent Meta Quest Pro) (Robertson, 2022). There are still VR devices that do not enable camera passthrough access to developers, and so the LevelEd AR software would continue to enable SR experiences on these devices. Finally, the LevelEd AR software could be used at a location by one user to capture a real-world location to enable another user to create SR experiences remotely before they arrive at the location. Therefore, this capture method would be included as an additional option, rather than a replacement for LevelEd AR.

There is also potential to utilise LevelEd SR to help with the ideation and creation of AR experiences. Qian et al. (2022) demonstrate the potential for developing AR experiences whilst in VR. Their system uses a HoloLens 2 (AR HMD) to capture the real-world location which can then be viewed in VR to allow for the creation of bespoke AR experiences. There is potential to adapt LevelEd SR to enable this type of workflow but with the real-world location captured by the more accessible LevelEd AR software.

Finally, the study presented in this thesis shows the promise and potential of SR game levels and the LevelEd SR workflow to enable them. However, a more in-depth study that explores the use of LevelEd SR in differing conditions would be a clear next step. The current study utilised the same location and furniture set up for each participant to create consistent comparable conditions between participants. The next step would be to allow users to test LevelEd SR and the workflow in their own homes as this would help to further evaluate the system and reveal any potential issues not already picked up as well as additional useful feature requests.



## 7.5 Final Words

This thesis set out to answer the following research question:

*Can augmented reality and virtual reality technologies be used to improve current level design workflows or establish new workflows that enable new experiences?*

In total, three pieces of software were developed to enable three different level design workflows. Each workflow made use of either augmented reality and/or virtual reality technologies. Each of these workflows has been evaluated by participants and in total, seven hypotheses were tested.

**Workflow 1** shows that augmented reality can be used to improve a current level design workflow by supporting the creation of a virtual map of a space that can be used in serious games VR level design. This workflow was tested with **H1**, which was proven true through **Study 1**. This supports the first part of the research question relating to improving current level design workflows.

**Workflow 2** shows that virtual reality can be used to improve a current level design workflow by enabling developers to create VR game levels using a VR level editor. This workflow was tested with **H2**, **H3** and **H4**, all of which were proven true through **Study 2**. This supports the first part of the research question relating to improving current level design workflows.

Finally, **Workflow 3** shows that both augmented reality and virtual reality can be used to enable new experiences in the form of substitutional reality level design. This workflow was tested with **H5**, **H6** and **H7**, all of which were proven true through **Study 3**. This supports the final part of the research question relating to enabling new experiences.

The seven hypotheses have been proven to be true and therefore the research question has been answered as both existing workflows have been improved and a new workflow and experience successfully created. As both AR and VR technologies continue to improve and as the public become more familiar with them, it is expected that these workflows will become more common for both developers and players in the years ahead. The contributions of this thesis are therefore of interest and use to anyone looking to develop VR level editors for professional level designers or players, indoor modelling tools, and substitutional reality game experiences.

# References

- 343 Industries. (2021). *Halo Infinite* [Video game]. Xbox Game Studios.
- 3DBear. (2018, December 7). *Low Poly Egypt Desert Fantasy Environment* [Assets]. Unity Asset Store. <https://assetstore.unity.com/packages/3d/environments/historic/low-poly-egypt-desert-fantasy-environment-134729>
- Abdelaal, M., Reichelt, D., Dürr, F., Rothermel, K., Runceanu, L., Becker, S., & Fritsch, D. (2018). Comnsense: Grammar-driven crowd-sourcing of point clouds for automatic indoor mapping. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1), 1-26. <https://doi.org/10.1145/3191733>
- Adams, D. [DaveMAdams]. (2017, October 4). *Early blackout and final art for the middle section of OW Junkertown. #Blocktober #Overwatch #junkertown #leveldesign* [Tweet]. Twitter. <https://twitter.com/DaveMAdams/status/915684159552577536>
- Adams, E. (2014). *Fundamentals of Game Design* (3rd ed.). New Riders.
- Allan, D. (2016, February 5). *Thinking of getting an HTC Vive? Check that your room is this size first* [Image]. TechRadar. <https://www.techradar.com/news/wearables/thinking-of-getting-a-htc-vive-check-that-your-room-is-this-size-first-1314450>
- Amos, E. (2019). *The Game Console*. No Starch Press, Inc
- Angladon, V., Gasparini, S., & Charvillat, V. (2018, February). Room floor plan generation on a project tango device. *In International Conference on Multimedia Modeling* (pp. 226-238). Springer, Cham. [https://doi.org/10.1007/978-3-319-73600-6\\_20](https://doi.org/10.1007/978-3-319-73600-6_20)
- Ante, J. (2012, January 10). *Reverse Normals*. Unify Community. <https://web.archive.org/web/20201112015209/https://wiki.unity3d.com/index.php/ReverseNormals>
- Apple Inc. (2022a). *ARKit*. <https://developer.apple.com/documentation/arkit/>
- Apple Inc. (2022b). *More to explore with ARKit 6*. <https://developer.apple.com/augmented-reality/arkit/>
- Arayici, Y. (2007). An approach for real world data modelling with the 3D terrestrial laser scanner for built environment. *Automation in construction*, 16(6), 816-829. <https://doi.org/10.1016/j.autcon.2007.02.008>
- Azuma, R. T. (1997). A survey of augmented reality. *Presence: teleoperators & virtual environments*, 6(4), 355-385. <https://doi.org/10.1162/pres.1997.6.4.355>

- Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3), 114-123.  
<https://dl.acm.org/doi/abs/10.5555/2835587.2835589>
- Barot, C., Carpentier, K., Collet, M., Cuella-Martin, A., Lanquepin, V., Muller, M., ... & Wagrez, K. (2013, March). The wonderland builder: Using storytelling to guide dream-like interaction. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 201-202). IEEE. <https://doi.org/10.1109/3dui.2013.6550248>
- Beat Games. (2019). *Beat Saber* [Video game]. Beat Games.
- Beever, L., Pop, S., & John, N. (2019a, September). Evaluating LevelEd AR: An Indoor Modelling Application for Serious Games Level Design. In *2019 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)* (pp. 1-4). IEEE. <https://doi.org/10.1109/vs-games.2019.8864534>
- Beever, L., Pop, S. R., & John, N. W. (2019b, September). Assisting Serious Games Level Design with an Augmented Reality Application and Workflow. In *CGVC* (pp. 9-17). <https://doi.org/10.2312/cgvc.20191253>
- Beever, L., Pop, S., & John, N. W. (2020, August). LevelEd VR: A virtual reality level editor and workflow for virtual reality level design. In *2020 IEEE Conference on Games (CoG)* (pp. 136-143). IEEE. <https://doi.org/10.1109/cog47356.2020.9231769>
- Bérenger. (2014, October 26). *Procedural Primitives*. Unify Community.  
<https://web.archive.org/web/20210828182001/https://wiki.unity3d.com/index.php/ProceduralPrimitives>
- Bettner, P. (2015, October 1). *How devs deal with 4 problem areas in VR game design*. Interview by F, Bryant [Written Interview]. Game Developer.  
<https://www.gamedeveloper.com/design/how-devs-deal-with-4-problem-areas-in-vr-game-design>
- Biener, V., Kalamkar, S., Nouri, N., Ofek, E., Pahud, M., Dudley, J. J., ... & Grubert, J. (2022). Quantifying the effects of working in VR for one week. *IEEE Transactions on Visualization and Computer Graphics*, 28(11), 3810-3820.  
<https://doi.org/10.1109/tvcg.2022.3203103>
- Bithell, M. [Mike Bithell]. (2015, May 11). *Volume - Level Editor* [Video]. YouTube.  
<https://www.youtube.com/watch?v=0PFySU14wZg>

- Bohon, C. (2021, June 9). *Apple's ARKit: Cheat sheet*. Tech Republic.  
<https://www.techrepublic.com/article/apples-arkit-everything-the-pros-need-to-know/>
- Bolton, G. (2020, April 2). *The best practices and design principles of VR development*. Interview by M. Dealessandri [Written Interview]. Gamesindustry.biz.  
<https://www.gamesindustry.biz/the-best-practices-and-design-principles-of-vr-development>
- Bonato, F., Bubka, A., Palmisano, S., Phillip, D., & Moreno, G. (2008). Vection change exacerbates simulator sickness in virtual environments. *Presence: Teleoperators and Virtual Environments*, 17(3), 283-292. <https://doi.org/10.1162/pres.17.3.283>
- Boyer, S. (2009). A virtual failure: Evaluating the success of Nintendo's Virtual Boy. *The Velvet Light Trap*, (64), 23-33. <https://doi.org/10.1353/vlt.0.0039>
- Bowman, D. A., & Hodges, L. F. (1997). An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *SI3D*, 97, 35-38.  
<https://doi.org/10.1145/253284.253301>
- BreakAway Games. (2022). *vHealthCare*. <http://www.breakawaygames.com/vhealthcare/>
- Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7. <https://doi.org/10.1201/9781498710411-35>
- Buhler, H., Misztal, S., & Schild, J. (2018, March). Reducing vr sickness through peripheral visual effects. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 517-9). IEEE. <https://doi.org/10.1109/vr.2018.8446346>
- Business Wire. (2020, March 23). *The 7th Annual Bloxy Awards Brings 4 Million Concurrent Players Together on Roblox*.  
<https://www.businesswire.com/news/home/20200323005741/en/7th-Annual-Bloxy-Awards-Brings-4-Million>
- Butcher, P. W., John, N. W., & Ritsos, P. D. (2020). VRIA: A web-based framework for creating immersive analytics experiences. *IEEE Transactions on visualization and computer graphics*, 27(7), 3213-3225. <https://doi.org/10.1109/tvcg.2020.2965109>
- Caplin, S. (2008). *Art and Design in Photoshop*. Focal Press
- Chang, E., Kim, H. T., & Yoo, B. (2020). Virtual reality sickness: a review of causes and measurements. *International Journal of Human-Computer Interaction*, 36(17), 1658-1682. <https://doi.org/10.1080/10447318.2020.1778351>

- Checa, D., & Bustillo, A. (2020). A review of immersive virtual reality serious games to enhance learning and training. *Multimedia Tools and Applications*, 79(9), 5501-5527. <https://doi.org/10.1007/s11042-019-08348-9>
- Chen, S. Y., & Chien, W. C. (2022). Immersive virtual reality serious games with DL-assisted learning in high-rise fire evacuation on fire safety training and research. *Frontiers in psychology*, 13. <https://doi.org/10.3389/fpsyg.2022.786314>
- Cheng, L. P., Ofek, E., Holz, C., & Wilson, A. D. (2019, March). Vroamer: generating on-the-fly VR experiences while walking inside large, unknown real-world building environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 359-366). IEEE. <https://doi.org/10.1109/vr.2019.8798074>
- Chittaro, L., & Ranon, R. (2009, March). Serious games for training occupants of a building in personal fire safety skills. In *2009 Conference in Games and Virtual Worlds for Serious Applications* (pp. 76-83). IEEE. <https://doi.org/10.1109/vs-games.2009.8>
- Circonia Studio. (2021, January 26). *Free Double Sided Shaders*. Unity Asset Store. <https://assetstore.unity.com/packages/vfx/shaders/free-double-sided-shaders-23087>
- Cleanbox. (2022). About Covid-19. Retrieved October 05, 2022, from <https://cleanboxtech.com/about-covid-19/>
- Conway, M., Audia, S., Burnette, T., Cosgrove, D., & Christiansen, K. (2000, April). Alice: lessons learned from building a 3D system for novices. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 486-493). ACM. <https://doi.org/10.1145/332040.332481>
- Coomer, N., Bullard, S., Clinton, W., & Williams-Sanders, B. (2018, August). Evaluating the effects of four VR locomotion methods: joystick, arm-cycling, point-tugging, and teleporting. In *Proceedings of the 15th ACM symposium on applied perception* (pp. 1-8). <https://doi.org/10.1145/3225153.3225175>
- Crane, D. (1982). *Pitfall!* [Video game]. Activision.
- Cyclops. (2010, April 26). *Determine whether or not model is "inside out"*. Unity. <https://answers.unity.com/questions/15978/determine-whether-or-not-model-is-inside-out.html>
- De Leon, J. D. O., Tavas, R. P., Aranzanso, R. A., & Atienza, R. O. (2016, November). Genesys: A Virtual Reality scene builder. In *2016 IEEE Region 10 Conference (TENCON)* (pp. 3708-3711). IEEE. <https://doi.org/10.1109/tencon.2016.7848751>

- Denzer, S., Diezig, S., Achermann, P., Koenig, T., & Mast, F. W. (2022). BizarreVR: Dream-like bizarreness in immersive virtual reality induced changes in conscious experience of reality while leaving spatial presence intact. *Consciousness and cognition*, 99, 103283. <https://doi.org/10.1016/j.concog.2022.103283>
- Dickinson, W. (2019, October 22). *Tower Tag Announces VR Arcade Tournament & Launch on the SpringboardVR Marketplace*. Springboard VR. <https://www.springboardvr.com/blog/tower-tag-announces-vr-arcade-tournament-and-launch-on-the-springboardvr-marketplace>
- Dilek, U., & Erol, M. (2018). Detecting position using ARKit II: generating position-time graphs in real-time and further information on limitations of ARKit. *Physics Education*, 53(3), 035020. <https://doi.org/10.1088/1361-6552/aaadd9>
- Dillet, R. (2018, September 5). *Unity CEO says half of all games are built on Unity*. Tech Crunch. <https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/>
- Djaouti, D., Alvarez, J., & Jessel, J. P. (2011). Classifying serious games: the G/P/S model. In *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches* (pp. 118-136). IGI Global. [https://doi.org/10.1007/978-3-319-94274-2\\_13](https://doi.org/10.1007/978-3-319-94274-2_13)
- Duan, H., Huang, Y., Zhao, Y., Huang, Z., & Cai, W. (2022, August). User-generated content and editors in video games: Survey and vision. In *2022 IEEE Conference on Games (CoG)* (pp. 536-543). IEEE. <https://doi.org/10.1109/cog51982.2022.9893717>
- Eckstein, B., Krapp, E., & Lugin, B. (2018, September). Towards serious games and applications in smart substitutional reality. In *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)* (pp. 1-8). IEEE. <https://doi.org/10.1109/vs-games.2018.8493444>
- Eckstein, B., Krapp, E., Elsässer, A., & Lugin, B. (2019). Smart substitutional reality: Integrating the smart home into virtual reality. *Entertainment Computing*, 31, 100306. <https://doi.org/10.1016/j.entcom.2019.100306>
- Edwards, B. (2015, October 21). *Unraveling The Enigma Of Nintendo's Virtual Boy, 20 Years Later*. Fast Company. <https://www.fastcompany.com/3050016/unraveling-the-enigma-of-nintendos-virtual-boy-20-years-later>
- Ekströmer, P., Wever, R., Andersson, P., & Jönsson, J. (2019, July). Shedding light on game engines and virtual reality for design ideation. In *Proceedings of the Design Society*:

- International Conference on Engineering Design* (Vol. 1, No. 1, pp. 2003-2010). Cambridge University Press. <https://doi.org/10.1017/dsi.2019.206>
- Epic Games. (1999). *Unreal Tournament* [Video game]. GT Interactive.
- Epic Games. (2017). *Robo Recall* [Video game]. Epic Games.
- Epic Games. (2017). *Fortnite* [Video game]. Epic Games.
- Epp, R., Lin, D., & Bezemer, C. P. (2021). An empirical study of trends of popular virtual reality games and their complaints. *IEEE Transactions on Games*, 13(3), 275-286. <https://doi.org/10.1109/tg.2021.3057288>
- Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics*, 5(1), 1-4. <https://doi.org/10.11648/j.ajtas.20160501.11>
- Evans, B. (2022, May 4). *How to use Instagram AR filters for marketing*. We Are Hydrogen. <https://www.wearehydrogen.com/post/How-to-use-Instagram-AR-filters-for-marketing>
- Facebook. (2021). *Horizon Worlds Beta* [Video game]. Facebook.
- Fernandes, A. S., & Feiner, S. K. (2016, March). Combating VR sickness through subtle dynamic field-of-view modification. In *2016 IEEE symposium on 3D user interfaces (3DUI)* (pp. 201-210). IEEE. <https://doi.org/10.1109/3dui.2016.7460053>
- Ferreira, J., Mendes, D., Nóbrega, R., & Rodrigues, R. (2021, March). Immersive multimodal and procedurally-assisted creation of vr environments. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (pp. 30-37). IEEE. <https://doi.org/10.1109/vrw52623.2021.00012>
- Field, P. [Peter\_\_Field]. (2017, October 3). OK so #Blocktober is my favorite thing right now. Here are some shots from the first game I ever worked on. Enslaved - Odyssey to the West [Tweet]. Twitter. [https://twitter.com/Peter\\_Field/status/915321037327388678](https://twitter.com/Peter_Field/status/915321037327388678)
- FLAIM Systems. (2022). *FLAIM Systems: Train Virtually, Experience Reality*. <https://www.flaimsystems.com/>
- Frommel, J., Sonntag, S., & Weber, M. (2017, August). Effects of controller-based locomotion on player experience in a virtual reality exploration game. In *Proceedings of the 12th international conference on the foundations of digital games* (pp. 1-6). <https://doi.org/10.1145/3102071.3102082>



- Fu, Z., Xu, R., Xin, S., Chen, S., Tu, C., Yang, C., & Lu, L. (2022). EasyVRModeling: Easily Create 3D Models by an Immersive VR System. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1), 1-14.  
<https://doi.org/10.1145/3522613>
- Fulton, W. (2017, December 20). 'Pokémon Go' levels up its augmented reality abilities with Apple's ARKit. Digital Trends. <https://www.digitaltrends.com/gaming/pokemon-go-arkit/>
- Fusco, G., & Coughlan, J. M. (2018, July). Indoor localization using computer vision and visual-inertial odometry. In *International conference on computers helping people with special needs* (pp. 86-93). Springer, Cham. [https://doi.org/10.1007/978-3-319-94274-2\\_13](https://doi.org/10.1007/978-3-319-94274-2_13)
- Fusco, G., Cheraghi, S. A., Neat, L., & Coughlan, J. M. (2020, September). An indoor navigation app using computer vision and sign recognition. In *International Conference on Computers Helping People with Special Needs* (pp. 485-494). Springer, Cham. [https://doi.org/10.1007/978-3-030-58796-3\\_56](https://doi.org/10.1007/978-3-030-58796-3_56)
- Ganesan, K. [@karanganesan]. (2022, January 30). Woah @Lynx\_xr 🤖 FOV Comparison between HoloLens 2, Magic Leap 1/2 and Lynx R1 (ofc comparing a pass through AR... [Tweet]. Twitter.  
<https://twitter.com/karanganesan/status/1487820724433604618>
- Garcia, J. F., Simeone, A. L., Higgins, M., Powell, W., & Powell, V. (2018, May). Inside looking out or outside looking in? An evaluation of visualisation modalities to support the creation of a Substitutional Virtual Environment. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (pp. 1-8).  
<https://doi.org/10.1145/3206505.3206529>
- Garcia-Palacios, A., Hoffman, H., Carlin, A., Furness III, T. A., & Botella, C. (2002). Virtual reality in the treatment of spider phobia: a controlled study. *Behaviour research and therapy*, 40(9), 983-993. [https://doi.org/10.1016/s0005-7967\(01\)00068-7](https://doi.org/10.1016/s0005-7967(01)00068-7)
- Gears of War. [GearsofWar]. (2017, October 5). #GearsofWar4 #Blocktober Pt. 2! Reclaimed was originally an MPLD's design test for a job. From there, it evolved into what you play today! [Tweet]. Twitter.  
<https://twitter.com/GearsofWar/status/916028361180250112>
- George, D., & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference. 11.0 update* (4th ed.). Allyn & Bacon.

- Giret, L. (2022, October 26). *Microsoft Has Sold 300,000 HoloLens Units According to Analysts*. Thurrott. <https://www.thurrott.com/mobile/275228/microsoft-hololens-300000-units-sold>
- Google Developers. (2022). *Overview of ARCore and supported development environments*. <https://developers.google.com/ar/develop/>
- Greenwald, W. (2018, September 06). *Magic Leap One*. PCMag UK. <https://uk.pcmag.com/virtual-reality/117270/magic-leap-one>
- Grimsby, G. (2014). *An Architectural Approach to Level Design* (1st ed.). Interview by C. W. Totten [Written Interview]. Boca Raton, FL: CRC Press.
- Grubert, J., Langlotz, T., Zollmann, S., & Regenbrecht, H. (2016). Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE transactions on visualization and computer graphics*, 23(6), 1706-1724. <https://doi.org/10.1109/tvcg.2016.2543720>
- Halo. [Halo]. *Halo Infinite | Forge Fundamentals – Building* [Video]. YouTube. <https://www.youtube.com/watch?v=SAwEbD5qNG8>
- Hamilton, I. (2019, August 22). *Oculus Insight: Facebook Details Quest's Inside Out Tracking System* [Image]. UploadVR. <https://uploadvr.com/oculus-insight-details-quest/>
- Hartmann, J., Holz, C., Ofek, E., & Wilson, A. D. (2019, May). Realitycheck: Blending virtual environments with situated physical reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-12). <https://doi.org/10.1145/3290605.3300577>
- Hartmann, T., Wirth, W., Schramm, H., Klimmt, C., Vorderer, P., Gysbers, A., ... & Sacau, A. M. (2016). The Spatial Presence Experience Scale (SPES): A short self-report measure for diverse media settings. *Journal of Media Psychology: Theories, Methods, and Applications*, 28(1), 1. <https://doi.org/10.1027/1864-1105/a000137>
- Hayden, S. (2019a, March 16). *'Beat Saber' Sells Over One Million Copies*. Road To VR. <https://www.roadtovr.com/beat-saber-sells-one-million-copies/>
- Hayden, S. (2019b, December 10). Report: Magic Leap Sees Sluggish Sales Numbers Amidst Latest Funding Round. Road To VR. <https://www.roadtovr.com/report-magical-leap-sales-funding-round/>

- Hayden, S. (2022a, September 22). 'Guardians Frontline' is Like 'Halo' Meets 'Starcraft', Coming to Quest 2 & SteamVR Early 2023. Road To VR. <https://www.roadtovr.com/guardians-frontline-quest-2-halo-starcraft/>
- Hayden, S. (2022b, December 13). VR's Most Popular Battle Royale is Getting Sandbox Creation Tools This Week. Road To VR. <https://www.roadtovr.com/population-one-sandbox-quest-2-steam-release/>
- Heaney, D. (2019, February 25). HoloLens 2's Field of View Revealed. UploadVR. <https://uploadvr.com/hololens-2-field-of-view/>
- Heaney, D. (2022, June 07). IDC Estimates Quest 2 Has Sold Almost 15 Million Units. UploadVR. <https://uploadvr.com/quest-2-sold-almost-15-million-idc/>
- Heath, A. (2023, March 1). This is Meta's AR / VR hardware roadmap for the next four years. The Verge. <https://www.theverge.com/2023/2/28/23619730/meta-vr-oculus-ar-glasses-smartwatch-plans>
- Hedau, V., Hoiem, D., & Forsyth, D. (2010, September). Thinking inside the box: Using appearance models and context based on room geometry. In European Conference on Computer Vision (pp. 224-237). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-15567-3\\_17](https://doi.org/10.1007/978-3-642-15567-3_17)
- Higinbotham. (1958). *Tennis for Two* [Video game]. William Higinbotham.
- Hoffman, H. G. (1998, March). Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments. In *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No. 98CB36180)* (pp. 59-63). IEEE. <https://doi.org/10.1109/vrais.1998.658423>
- Hoppen, P., Knieriemen, T., & Von Puttkamer, E. (1990, May). Laser-radar based mapping and navigation for an autonomous mobile robot. In *Proceedings., IEEE International Conference on Robotics and Automation* (pp. 948-953). IEEE. <https://doi.org/10.1109/robot.1990.126113>
- Ichikawa, S., Takashima, K., Tang, A., & Kitamura, Y. (2018, November). VR safari park: a concept-based world building interface using blocks and world tree. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology* (p. 6). ACM. <https://doi.org/10.1145/3281505.3281517>
- id Software. (1996). *Doom* [Video game]. id Software.

- IJsselsteijn, W. A., de Kort, Y. A. W., & Poels, K. (2013). *The Game Experience Questionnaire*. Technische Universiteit Eindhoven.  
<https://research.tue.nl/en/publications/the-game-experience-questionnaire>
- Insko, B. E. (2001). *Passive haptics significantly enhances virtual environments [PhD Thesis]*. The University of North Carolina at Chapel Hill.  
<https://www.proquest.com/openview/2b43afef0d0711642d36bf836a207a5a/1?pq-origsite=gscholar&cbl=18750&diss=y>
- International Society of Presence Research. (2000). *Presence defined*.  
<https://ispr.info/about-presence-2/about-presence/>
- Ipsita, A., Li, H., Duan, R., Cao, Y., Chidambaram, S., Liu, M., & Ramani, K. (2021, May). VRFromX: from scanned reality to interactive virtual experience with human-in-the-loop. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-7). <https://doi.org/10.1145/3411763.3451747>
- Ishikawa, M., Hagiwara, T., Takashima, K., & Kitamura, Y. (2016, November). Viblock: block-shaped content manipulation in VR. In *SIGGRAPH ASIA 2016 VR Showcase* (p. 1). ACM. <https://doi.org/10.1145/2996376.2996391>
- Jackson, B., & Keefe, D. F. (2016). Lift-off: Using reference imagery and freehand sketching to create 3d models in vr. *IEEE transactions on visualization and computer graphics*, 22(4), 1442-1451. <https://doi.org/10.1109/tvcg.2016.2518099>
- Jager, J., Putnick, D. L., & Bornstein, M. H. (2017). II. More than just convenient: The scientific merits of homogeneous convenience samples. *Monographs of the Society for Research in Child Development*, 82(2), 13-30.  
<https://doi.org/10.1111/mono.12296>
- Jerald, J., Mlyniec, P., Yoganandan, A., Rubin, A., Paullus, D., & Solotko, S. (2013, March). Makevr: A 3d world-building interface. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 197-198). IEEE. <https://doi.org/10.1109/3dui.2013.6550246>
- Jerald, J. (2016). *The VR book: Human-centered design for virtual reality*. San Rafael, CA: Morgan & Claypool
- John, N. W., Pop, S. R., Day, T. W., Ritsos, P. D., & Headleand, C. J. (2018). The implementation and validation of a virtual environment for training powered wheelchair manoeuvres. *IEEE transactions on visualization and computer graphics*, 24(5), 1867-1878. <https://doi.org/10.1109/tvcg.2017.2700273>

- Joyce, R. D., & Robinson, S. (2017). Passive haptics to enhance virtual reality simulations. In *AIAA Modeling and Simulation Technologies Conference* (p. 1313).
- Kalantari, M., & Nechifor, M. (2016). Accuracy and utility of the Structure Sensor for collecting 3D indoor information. *Geo-spatial information science*, 19(3), 202-209.  
<https://doi.org/10.1080/10095020.2016.1235817>
- Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3), 203-220.  
[https://doi.org/10.1207/s15327108ijap0303\\_3](https://doi.org/10.1207/s15327108ijap0303_3)
- Kenney. (2023). Thousands of completely free game assets for you to use [Audio].  
<https://www.kenney.nl/>
- Kerby, D. S. (2014). The simple difference formula: An approach to teaching nonparametric correlation. *Comprehensive Psychology*, 3, 2165–2228.  
<https://doi.org/10.2466/11.it.3.1>
- Koczwara, M., Smith, A., & Gutierrez, L. (2018). *How to Build in Fortnite Creative Mode* [Image]. IGN.  
[https://www.ign.com/wikis/fortnite/How to Build in Fortnite Creative Mode](https://www.ign.com/wikis/fortnite/How_to_Build_in_Fortnite_Creative_Mode)
- Koster, R. (2014). *A Theory of Fun for Games Design*. (2nd ed.) Sebastopol, CA: O'Reilly Media
- Kremers, R. (2009). *Level Design: Concept, Theory and Practice*. Boca Raton, FL: CRC Press.
- Lang, B. (2020, March 10). *A Complete List of Headsets Compatible with 'Half-Life: Alyx'. Road To VR.* <https://www.roadtovr.com/half-life-alyx-which-vr-headset-support-compatible-oculus-quest-faq/>
- Lang, B. (2021a, April 11). *Meta: 124 Quest Apps Have Earned More Than \$1M, 8 Have Earned More Than \$20M.* Road to VR. <https://www.roadtovr.com/oculus-quest-store-revenue-stats-data-apps-earning-1-million-meta-gdc/>
- Lang, B. (2021b, September 7). *Quest 2 Sees Its First Loss in Share on Steam But It's Hardly Done Growing.* Road To VR. <https://www.roadtovr.com/quest-2-loss-steamvr-hardware-software-survey-august-2021/>

- Lang, B. (2022a, August 8). *Meta's PC VR Dominance Continues as Quest 2 Passes New Milestone*. Road To VR. <https://www.roadtovr.com/meta-quest-2-majority-vr-headset-on-steam/>
- Lang, B. (2022b, November 21). *Niantic Reveals Snapdragon AR2-powered Headset Reference Design for Outdoor Use*. Road To VR. <https://www.roadtovr.com/niantic-ar2-headset-outdoor-announcement/>
- Langbehn, E., Lubos, P., & Steinicke, F. (2018, April). Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking. In *Proceedings of the Virtual Reality International Conference-Laval Virtual* (pp. 1-9). <https://doi.org/10.1145/3234253.3234291>
- Larian Studios. (2017, September 19). *Larian guides*. [https://docs.larian.game/Larian\\_guides](https://docs.larian.game/Larian_guides)
- Laurel, B. (2016). *What is virtual reality?*. Medium. <https://medium.com/@blaurel/what-is-virtual-reality-77b876d829ba>.
- Law, E. L. C., Brühlmann, F., & Mekler, E. D. (2018, October). Systematic review and validation of the game experience questionnaire (GEQ) - implications for citation and reporting practice. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play* (pp. 257-270). <https://doi.org/10.1145/3242671.3242683>
- Le Chenechal, M., Lacoche, J., Martin, C., & Royan, J. (2015, March). Laying out spaces with virtual reality. In *2015 IEEE Virtual Reality (VR)* (pp. 337-338). IEEE. <https://doi.org/10.1109/vr.2015.7223433>
- Lee, S. (2020, May 22). *From pen-and-paper to blockouts: How level designers craft iconic stealthy encounters*. Interview by J. Yarwood [Written Interview]. Game Developer. <https://www.gamedeveloper.com/design/from-pen-and-paper-to-blockouts-how-level-designers-craft-iconic-stealthy-encounters>
- Lin, C. E., Cheng, T. Y., & Ma, X. (2020, April). Architect: Building interactive virtual experiences from physical affordances by bringing human-in-the-loop. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-13). <https://doi.org/10.1145/3313831.3376614>
- Líndal, P. J., Jóhannsdóttir, K. R., Kristjánsson, U., Lensing, N., Stühmeier, A., Wohlan, A., & Vilhjálmsson, H. H. (2018, September). Comparison of teleportation and fixed track driving in VR. In *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)* (pp. 1-7). IEEE. <https://doi.org/10.1109/VS-Games.2018.8493414>



- Liu, C., Schwing, A. G., Kundu, K., Urtasun, R., & Fidler, S. (2015). Rent3d: Floor-plan priors for monocular layout estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3413-3421).  
<https://doi.org/10.1109/cvpr.2015.7298963>
- Lovreglio, R., Gonzalez, V., Amor, R., Spearpoint, M., Thomas, J., Trotter, M., & Sacks, R. (2017, July). The need for enhancing earthquake evacuee safety by using virtual reality serious games. In *Lean & Computing in Construction Congress* (pp. 1-9).  
<https://doi.org/10.24928/jc3-2017/0058>
- Lovreglio, R., Gonzalez, V., Feng, Z., Amor, R., Spearpoint, M., Thomas, J., ... & Sacks, R. (2018). Prototyping virtual reality serious games for building earthquake preparedness: The Auckland City Hospital case study. *Advanced Engineering Informatics*, 38, 670-682. <https://doi.org/10.1016/j.aei.2018.08.018>
- Magic Leap. (2022). *Magic Leap 2 – The most immersive enterprise AR device*.  
[https://www.magicleap.com/hubfs/docs/Magic-Leap-2\\_Product-Spec-Sales-Sheet\\_09.30.2022\\_Version-English-4.0.pdf?hsLang=en](https://www.magicleap.com/hubfs/docs/Magic-Leap-2_Product-Spec-Sales-Sheet_09.30.2022_Version-English-4.0.pdf?hsLang=en)
- Maiberg, E. (2016, April 17). *Virtual Reality's Locomotion Problem*. Vice.  
<https://www.vice.com/en/article/d7y38q/virtual-realities-locomotion-problem>
- Masnadi, S., González, A. N. V., Williamson, B., & LaViola, J. J. (2020, March). Affordit!: A tool for authoring object component behavior in VR. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (pp. 740-741). IEEE. <https://doi.org/10.1109/vrw50115.2020.00221>
- Martinez, E. S., Wu, A. S., & McMahan, R. P. (2022, March). Research Trends in Virtual Reality Locomotion Techniques. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 270-280). IEEE.  
<https://doi.org/10.1109/vr51125.2022.00046>
- McMahan, R. P., Kopper, R., and Bowman, D. A. (2014). Principles for Designing Effective 3D Interaction Techniques. In K. M. Stanney and K. S. Hale (Eds.), *Handbook of Virtual Environments* (2nd ed., pp. 285–311). Boca Raton, FL: CRC Press.
- Media Molecule. (2008). *Little Big Planet* [Video game]. Sony Computer Entertainment.
- Meehan, M., Insko, B., Whitton, M., & Brooks Jr, F. P. (2002). Physiological measures of presence in stressful virtual environments. *Acm transactions on graphics (tog)*, 21(3), 645-652. <https://doi.org/10.1145/566654.566630>



- Meta. (2010). *Instagram* (1.0) [Mobile app]. App Store.  
<https://apps.apple.com/gb/app/instagram/id389801252>
- Meta Quest Blog. (2022, December 19). *Introducing Meta Reality: A Look at the Technologies Necessary to Convincingly Blend the Virtual and Physical Worlds*. Meta. <https://www.meta.com/en-gb/blog/quest/mixed-reality-definition-passthrough-scene-understanding-spatial-anchors/>
- Metastruct. (2013). *Simple C# FTP Class*. Code Project.  
<https://www.codeproject.com/Tips/443588/Simple-Csharp-FTP-Class>
- Michael, D. and Chen, S. (2006) *Serious Games: Games That Educate, Train and Inform*. Boston, MA: Thomson.
- Microsoft. (2021, November 23). *HoloLens (1<sup>st</sup> gen) hardware*.  
<https://learn.microsoft.com/en-us/hololens/hololens1-hardware>
- Microsoft. (2022a, October 13). *About HoloLens 2*. <https://learn.microsoft.com/en-us/hololens/hololens2-hardware>
- Microsoft. (2022b, October 13). *HoloLens 2 with Remote Assist*.  
<https://learn.microsoft.com/en-us/hololens/hololens2-options-remote-assist>
- Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12), 1321-1329.
- Mitchell, R. (2014, March 28). *Oculus Rift: From \$2.4 million Kickstarter to \$2 billion sale*. Engadget. <https://www.engadget.com/2014-03-28-oculus-rift-from-2-4-million-kickstarter-to-2-billion-sale.html>
- Moby Games. (2022). *Super Mario Bros. 3 Credits (NES)*.  
<https://www.mobygames.com/game/nes/super-mario-bros-3/credits>
- Mojang Studios. (2011). *Minecraft* [Video game]. Mojang Studios.
- Monszpart, A., Mellado, N., Brostow, G. J., & Mitra, N. J. (2015). RAPter: rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.*, 34(4), 103-1.  
<https://doi.org/10.1145/2766995>
- Moon, M. (2019, March 26). *Sony has sold 4.2 million PlayStation VR headset*. Engadget.  
<https://www.engadget.com/2019/03/26/sony-4-2-million-playstation-vr-headset-sales/>

- Moreira, J. F. M. (2018). *Mixed interactive/procedural content creation in immersive VR environments* [Thesis]. Retrieved from <https://repositorio-aberto.up.pt/bitstream/10216/114109/2/277705.pdf>
- Mozilla. (2023, Feb 19). *Fundamentals of WebXR*. [https://developer.mozilla.org/en-US/docs/Web/API/WebXR\\_Device\\_API/Fundamentals](https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API/Fundamentals)
- Nebeling, M., Lewis, K., Chang, Y. C., Zhu, L., Chung, M., Wang, P., & Nebeling, J. (2020, April). Xrdirector: A role-based collaborative immersive authoring system. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-12). <https://doi.org/10.1145/3313831.3376637>
- Neat Corporation. (2018). *Budget Cuts* [Video game]. Neat Corporation.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., ... & Fitzgibbon, A. (2011, October). Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality* (pp. 127-136). IEEE. <https://doi.org/10.1109/ismar.2011.6092378>
- Ng, G., Shin, J. G., Plopski, A., Sandor, C., & Saakes, D. (2018, March). Situated game level editing in augmented reality. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction* (pp. 409-418). <https://doi.org/10.1145/3173225.3173230>
- Niantic. (2016). *Pokémon GO* [Video game]. Niantic.
- Nintendo. (1990). *Super Mario Bros. 3* [Video game]. Nintendo.
- Ochmann, S., Vock, R., Wessel, R., & Klein, R. (2016). Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics*, 54, 94-103. <https://doi.org/10.1016/j.cag.2015.07.008>
- Oculus. (2012, August 1). *Oculus rift: Step Into the Game*. Kickstarter. <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>
- Oculus. (2022). *Passthrough API Overview*. <https://developer.oculus.com/documentation/unity/unity-passthrough/>
- Oculus. (2023). *VRC.Quest.Performance.1*. <https://developer.oculus.com/resources/vrc-quest-performance-1/>
- Odeldahl, E. (2020, April 2). *The best practices and design principles of VR development*. Interview by M. Dealessandri [Written Interview]. Gamesindustry.biz.

<https://www.gamesindustry.biz/the-best-practices-and-design-principles-of-vr-development>

One Hamsa. (2018). *Racket: Nx* [Video game]. One Hamsa.

Oswald, P., Tost, J., & Wettach, R. (2015, January). i. Ge: Exploring new game interaction metaphors with interactive projection. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (pp. 733-738). <https://doi.org/10.1145/2677199.2687910>

Park, J., Son, B., Seo, J., & Park, C. (2018, July). Development of Authoring Tool for Extended Interaction in Mixed Reality Environments. In *International Conference on Human-Computer Interaction* (pp. 304-310). Springer, Cham. [https://doi.org/10.1007/978-3-319-92279-9\\_41](https://doi.org/10.1007/978-3-319-92279-9_41)

Pennell, S. [SLEDGEHAMMER GAMES]. (2019, October 30). *Call of Duty Developer Profile: Level Designer, Sean Pennell* [Video]. YouTube. <https://www.youtube.com/watch?v=75yvWP-0XWk>

Peters, Z. (2020a, May 22). *From pen-and-paper to blockouts: How level designers craft iconic stealthy encounters*. Interview by J. Yarwood [Written Interview]. Game Developer. <https://www.gamedeveloper.com/design/from-pen-and-paper-to-blockouts-how-level-designers-craft-iconic-stealthy-encounters>

Peters, C. (2020b, April 2). *The best practices and design principles of VR development*. Interview by M. Dealessandri [Written Interview]. Gamesindustry.biz. <https://www.gamesindustry.biz/the-best-practices-and-design-principles-of-vr-development>

Politis, Y., Robb, N., Yakkundi, A., Dillenburger, K., Herbertson, N., Charlesworth, B., & Goodman, L. (2017). People with disabilities leading the design of serious games and virtual worlds. *International Journal of Serious Games*, 4(2), 63-73. <https://doi.org/10.17083/ijsg.v4i2.160>

Poretski, L., & Arazy, O. (2017, February). Placing value on community co-creations: A study of a video game 'modding' community. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (pp. 480-491). <https://doi.org/10.1145/2998181.2998301>

Poretski, L., Zalmanson, L., & Arazy, O. (2019, December). The Effects of Co-Creation and Word-of-Mouth on Content Consumption-Findings from the Video Game Industry. In *ICIS*. [https://aisel.aisnet.org/icis2019/crowds\\_social/crowds\\_social/22](https://aisel.aisnet.org/icis2019/crowds_social/crowds_social/22)

- Pose-Díez-de-la-Lastra, A., Moreta-Martinez, R., García-Sevilla, M., García-Mato, D., Calvo-Haro, J. A., Mediavilla-Santos, L., ... & Pascau, J. (2022). HoloLens 1 vs. HoloLens 2: Improvements in the New Model for Orthopedic Oncological Interventions. *Sensors*, 22(13), 4915. <https://doi.org/10.3390/s22134915>
- Pranckevicius, A. (2015) MeshSerializer2. Unify Community. <https://web.archive.org/web/20200223132129/https://wiki.unity3d.com/index.php/MeshSerializer2>
- Project North Star (2022). *Project North Star – The future of spatial computing deserves to be open*. <https://docs.projectnorthstar.org/>
- Pruett, C. (2019). *An Architectural Approach to Level Design* (2nd ed.). Interview by C. W. Totten [Written Interview]. Boca Raton, FL: CRC Press.
- R Core Team (2020). *R: A Language and environment for statistical computing*. (Version 4.0) [Computer software]. <https://cran.r-project.org>. (R packages retrieved from MRAN snapshot 2020-08-24).
- Reason, J., & Brand, J. J. (1975). *Motion Sickness*. Academic Press.
- Rec Room Inc. (2016). *Rec Room* [Video game]. Rec Room Inc.
- Rees, N., Beever, L., Vaughan, N., Powell, C., Fletcher, A., & John, N. (2021). Virtual reality training in cardiopulmonary resuscitation in schools. *Journal of Paramedic Practice*, 13(9), 365-372. <https://doi.org/10.12968/jpar.2021.13.9.365>
- Revelle, W. (2019). *psych: Procedures for Psychological, Psychometric, and Personality Research*. [R package]. R Project. <https://cran.r-project.org/package=psych>
- Révész, R. (2021, May 20). *How many AR supported phones are in the world? – 2020*. AR On Platform. <https://www.aronplatform.com/mobile-ar-penetration-2020/>
- Riisgaard, S., & Blas, M. R. (2003). SLAM for Dummies. *A Tutorial Approach to Simultaneous Localization and Mapping*, 22(1-127), 126. [http://ocw.abu.edu.ng/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam\\_blas\\_repo.pdf](http://ocw.abu.edu.ng/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf)
- Robertbu. (2013, August 10). *How to find the center of three or more objects (number of objects varies)*. Unity. <https://answers.unity.com/questions/511841/how-to-make-an-object-move-away-from-three-or-more.html>

- Robertson, A. (2021, November 22). *Nreal Light review: Hardware is only half the battle*. The Verge. <https://www.theverge.com/22791981/nreal-light-augmented-mixed-reality-glasses-review>
- Roberston, A. (2022, November 22). *Meta Quest Pro review: get me out of here*. The Verge. <https://www.theverge.com/23451629/meta-quest-pro-vr-headset-horizon-review>
- Roblox Corporation. (2006-2015). *Roblox* [Video game]. Roblox Corporation.
- Rocket5tim. (2009, December 18). *transform.LookAt or Quaternion.LookRotation on 1 axis only?*. Unity. <https://forum.unity.com/threads/transform-lookat-or-quaternion-lookrotation-on-1-axis-only.36377/>
- Rubin, R. (2022, April 18). *Lenovo ThinkReality A3: Mixed reality hits the big screen*. ZDNET. <https://www.zdnet.com/article/lenovo-thinkreality-a3-mixed-reality-hits-the-big-screen/>
- runevision. (2018). *Triangulator*. Unify Community. <https://web.archive.org/web/20210622183655/https://wiki.unity3d.com/index.php?title=Triangulator>
- Wu, F., & Rosenberg, E. S. (2022, March). Asymmetric Lateral Field-of-View Restriction to Mitigate Cybersickness During Virtual Turns. *In 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 103-111). IEEE. <https://doi.org/10.1109/vr51125.2022.00028>
- Schmalstieg, D., & Höllerer, T. (2016). *Augmented reality: principles and practice*. Addison-Wesley.
- Sensopia. (2022). *Magicplan* (2022.15.0) [Mobile app]. App Store. <https://apps.apple.com/us/app/magicplan/id427424432>
- Servo. (2003). *E.T. The Extra-Terrestrial (Atari 2600) screenshot* [Image]. Moby Games. <https://www.mobygames.com/game/8874/et-the-extra-terrestrial/screenshots/atari-2600/53293/>
- Shapira, L., & Freedman, D. (2016, September). Reality skins: Creating immersive and tactile virtual environments. *In 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (pp. 115-124). IEEE. <https://doi.org/10.1109/ismar.2016.23>

- Shaver, D., & Yang, R. (2018). *Level Design Workshop: Invisible Intuition: Blockmesh and Lighting Tips to Guide Players and Set the Mood* [Video]. GDC Vault.  
<https://www.gdcvault.com/play/1025179/Level-Design-Workshop-Invisible-Intuition>
- Sherr, I. (2020, May 18). *Minecraft play spikes, tops 126M builders a month amid COVID-19*. CNET. <https://www.cnet.com/culture/amid-covid-19-minecraft-sees-sharp-spikes-over-126m-players-a-month/>
- Silverman, H. (2017). The Rule of 27s: A Comparative Analysis of 2D Screenspace and Virtual Reality Environment Design. In C. W. Totten (Ed.), *Level Design: Processes and Experiences* (pp. 309-322). Boca Raton, FL: CRC Press.
- Simeone, A. L. (2015, March). Substitutional reality: Towards a research agenda. In *2015 IEEE 1st Workshop on Everyday Virtual Reality (WEVR)* (pp. 19-22). IEEE.  
<https://doi.org/10.1109/wevr.2015.7151690>
- Simeone, A. L., Velloso, E., & Gellersen, H. (2015, April). Substitutional reality: Using the physical environment to design virtual reality experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 3307-3316).  
<https://doi.org/10.1145/2702123.2702389>
- SirOrlando. (2005). *Pitfall! (Atari 2600) screenshot* [Image]. Moby Games.  
<https://www.mobygames.com/game/8793/pitfall/screenshots/atari-2600/134595/>
- Skalski, P., Tamborini, R., Shelton, A., Buncher, M., & Lindmark, P. (2011). Mapping the road to fun: Natural video game controllers, presence, and game enjoyment. *New Media & Society*, 13(2), 224-242. <https://doi.org/10.1177/1461444810370949>
- Skarbez, R., Gabbard, J. L., Bowman, D. A., Ogle, J. T., & Tucker, T. (2022). Virtual replicas of real places: Experimental investigations. *IEEE Transactions on Visualization and Computer Graphics*, 28(12), 4594-4608. <https://doi.org/10.1109/tvcg.2021.3096494>
- Skarbez, R., Smith, M., & Whitton, M. C. (2021). Revisiting Milgram and Kishino's reality-virtuality continuum. *Frontiers in Virtual Reality*, 2, 647997.  
<https://doi.org/10.3389/frvir.2021.647997>
- Slater, M., & Wilbur, S. (1997). A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators & Virtual Environments*, 6(6), 603-616.  
<https://doi.org/10.1162/pres.1997.6.6.603>
- Sledgehammer Games. (2017). *Call of Duty: WWII* [Video game]. Activision.

- Sanp Inc. (2011). *Snapchat* (1.0) [Mobile app]. App Store.  
<https://apps.apple.com/gb/app/snapchat/id447188370>
- Sra, M., Garrido-Jurado, S., Schmandt, C., & Maes, P. (2016, November). Procedurally generated virtual reality from 3D reconstructed physical space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (pp. 191-200).  
<https://doi.org/10.1145/2993369.2993372>
- Sra, M., Garrido-Jurado, S., & Maes, P. (2017). Oasis: Procedurally generated social virtual spaces from 3d scanned real spaces. *IEEE transactions on visualization and computer graphics*, 24(12), 3174-3187. <https://doi.org/10.1109/tvcg.2017.2762691>
- Steinhaeusser, S. C., & Lugin, B. (2020, November). Horror laboratory and forest cabin-a horror game series for desktop computer, virtual reality, and smart substitutional reality. In *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play* (pp. 80-85). <https://doi.org/10.1145/3383668.3419924>
- Sun, Q., Wei, L. Y., & Kaufman, A. (2016). Mapping virtual and physical reality. *ACM Transactions on Graphics (TOG)*, 35(4), 1-12.  
<https://doi.org/10.1145/2897824.2925883>
- Suzuki, K., Wakisaka, S., & Fujii, N. (2012). Substitutional reality system: a novel experimental platform for experiencing alternative reality. *Scientific reports*, 2(1), 459.  
<https://doi.org/10.1038/srep00459>
- Tanenbaum, M. [marc\_tanenbaum]. (2021, May 17). *Virtual Camera & Face Capture apps: details and documentation*. Unity. <https://forum.unity.com/threads/virtual-camera-face-capture-apps-details-and-documentation.1111255/>
- Tecmo Koei. (2012). *Spirit Camera: The Cursed Memoir* [Video game]. Nintendo.
- The jamovi project (2021). *jamovi*. (Version 1.6) [Computer Software].  
<https://www.jamovi.org>
- Total Game Assets. (2018, January 23). *Stylized Egypt Texture Pack* [Assets]. Unity Asset Store. <https://assetstore.unity.com/packages/2d/textures-materials/stylized-egypt-texture-pack-107305>
- Totten, C. W. (2019). *An Architectural Approach to Level Design* (2nd ed.). Boca Raton, FL: CRC Press.



- Trimble. (2022). *Trimble Connect MR*. <https://fieldtech.trimble.com/en/product/trimble-connect-mr>
- TRIPP, Inc. (2019). *TRIPP* [Video game]. TRIPP, Inc.
- Turner, E., & Zakhor, A. (2014, January). Floor plan generation and room labeling of indoor environments from laser range data. *In 2014 international conference on computer graphics theory and applications (GRAPP)* (pp. 1-12). IEEE.  
<https://doi.org/10.5220/0004680300220033>
- UKIE. (2022). *UK Games Industry Census 2022* [PDF].  
<https://ukie.org.uk/download/4nxb3qvjfh5yvwqpqsaf4chjx9/0>
- Unity. (2020). *About AR Foundation*.  
<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.2/manual/index.html>  
!
- Unity. (2022). *Unity for Games*. Retrieved December 16, 2022, from  
<https://unity.com/solutions/game>
- Valve Corporation. (2023). Steam Hardware & Software Survey: January 2023. Steam Powered. Retrieved February 2, 2023, from  
<https://store.steampowered.com/hwsurvey>
- Virtual Reality Society. (n.d.). *Virtuality – A New Reality of Promise, Two Decades Too Soon*. Retrieved November 16, 2022, from <https://www.vrs.org.uk/dr-jonathan-walden-virtuality-new-reality-promise-two-decades-soon/>
- Visbox. (2023). *VisCube™ C4 CAVE Immersive 3D Display* [Image].  
<http://www.visbox.com/products/cave/viscube-c4/>
- Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings—Literature review and future needs. *Automation in construction*, 38, 109-127. <https://doi.org/10.1016/j.autcon.2013.10.023>
- VRChat Inc. (2014). *VRChat* [Video game]. VRChat Inc.
- VRCompare. (2023). *Compare Virtual Reality Headsets*. <https://www.vr-compare.com/vr>
- VRTK. (2022a). *Welcome to VRTK*. <https://vrtoolkit.readme.io/docs>
- VRTK. (2022b). *VRTK\_DragWorld*. [https://vrtoolkit.readme.io/docs/vrtk\\_dragworld](https://vrtoolkit.readme.io/docs/vrtk_dragworld)
- W3C. (2022a, April 19). *WebXR Hand Input Module - Level 1*. <https://www.w3.org/TR/webxr-hand-input-1/>

- W3C. (2022b, October 27). *WebXR Plane Detection Module*. <https://immersive-web.github.io/real-world-geometry/plane-detection.html>
- W3C. (2023, June 21). *WebXR Device API*. <https://www.w3.org/TR/2023/CRD-webxr-20230621/>
- Wallach, O. (2020, November 23). *50 Years of Gaming History, by Revenue Stream (1970-2020)*. Visual Capitalist. <https://www.visualcapitalist.com/50-years-gaming-history-revenue-stream/>
- Wang, J., Leach, O., & Lindeman, R. W. (2013, March). DIY World Builder: an immersive level-editing system. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 195-196). IEEE. <https://doi.org/10.1109/3dui.2013.6550245>
- Wang, J., & Lindeman, R. (2015). Coordinated hybrid virtual environments: Seamless interaction contexts for effective virtual reality. *Computers & Graphics*, 48, 71-83. <https://doi.org/10.1016/j.cag.2015.02.007>
- Warshaw, H. S. (1982). *E.T. the Extra-Terrestrial* [Video game]. Atari, Inc.
- Weimann, J. (2017, October 8). *Simple Unity3D Snap to Grid System*. Unity3D College. <https://unity3d.college/2017/10/08/simple-unity3d-snap-grid-system/>
- Weldon, A. (2008). *The Hows and Whys of Level Design* (2nd ed.). Interview by S. De Jong [Written Interview]. Raleigh, NC: Lulu Press.
- WeWantToKnow (2014). *Dragonbox Elements*. [iOS game]. Oslo, Norway: WeWantToKnow
- Witsel, N. (2020, April 2). *The best practices and design principles of VR development*. Interview by M. Dealessandri [Written Interview]. Gamesindustry.biz. <https://www.gamesindustry.biz/the-best-practices-and-design-principles-of-vr-development>
- Yang, J., Holz, C., Ofek, E., & Wilson, A. D. (2019, October). Dreamwalker: Substituting real-world walking experiences with a virtual reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (pp. 1093-1107). <https://doi.org/10.1145/3332165.3347875>
- Zhang, L., & Oney, S. (2020, October). Flowmatic: An immersive authoring tool for creating interactive scenes in virtual reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (pp. 342-353). <https://doi.org/10.1145/3379337.3415824>

Yixian, Y., Takashima, K., Tang, A., Tanno, T., Fujita, K., & Kitamura, Y. (2020, October). Zoomwalls: Dynamic walls that simulate haptic infrastructure for room-scale vr world. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (pp. 223-235). <https://doi.org/10.1145/3379337.3415859>

Zappar. (2023). Zapbox. <https://www.zappar.com/zapbox/>

Zou, C., Colburn, A., Shan, Q., & Hoiem, D. (2018). Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2051-2059). <https://doi.org/10.1109/cvpr.2018.00219>

# Appendices

## Appendix A

1. LevelEd AR (Workflow 1) Walkthrough Video: [https://youtu.be/IOd8JQjQs\\_U](https://youtu.be/IOd8JQjQs_U)
2. LevelEd AR (Workflow 1) Complex Scene Mapping: <https://youtu.be/K6x6RwyRpw4>
3. LevelEd AR Code (Unity 2018.3.6f1):
  - i. <https://github.com/TheBeev/LevelEd-AR>
  - ii. <https://www.dropbox.com/s/7i7I7qeb2h2q92w/LevelEd%20AR%20Original.zip?dl=0>

## Appendix B

1. LevelEd VR (Workflow 2) Short Example Video with commentary: <https://youtu.be/MXr9paO8Da0>
2. LevelEd VR (Workflow 2) Long Example Video with no commentary: <https://youtu.be/lgrr0bxEsLI>
3. LevelEd VR Code (Unity 2018.2.21f):
  - i. <https://github.com/TheBeev/LevelEd-VR>
  - ii. <https://www.dropbox.com/s/lockoegbexphbv2/LevelEd%20VR.zip?dl=0>
4. Desktop Version for study: <https://www.dropbox.com/s/z00qn58qf39o33e/LevelEd%20VR%20Desktop%20Version.zip?dl=0>

## Appendix C

1. LevelEd SR (Workflow 3) Example Video: <https://youtu.be/1qa9LHxre88>
2. LevelEd SR Code (Unity 2021.1.9f1):
  - i. <https://github.com/TheBeev/LevelEd-SR>
  - ii. <https://www.dropbox.com/s/3iq6abbf108ylur/LevelEd%20SR.zip?dl=0>
3. LevelEd AR to SR smartphone application code (Unity 2020.1.7f1):
  - i. <https://github.com/TheBeev/LevelEd-AR-to-SR>
  - ii. <https://www.dropbox.com/s/du3e7111614jzyy/LevelEd%20AR%20to%20SR.zip?dl=0>

## Appendix D

Table 21 – Extended table displaying common tethered VR headsets from 2016-2023. Sourced from VRCompare (2023).

	Oculus Rift	HTC Vive	Playstation VR	HP VR1000	Pimax Vision 8K+	Oculus Rift S	Valve Index	HP Reverb G2 Omnicept	HTC Vive Pro 2	Playstation VR 2
<b>Release Date</b>	March 2016	April 2016	October 2016	2017	Feb 2019	May 2019	May 2019	May 2021	June 2021	Feb 2023
<b>Price</b>	\$599	\$799	\$299	\$399	\$1449	\$399	\$999	\$1249	\$1399	\$549
<b>Horizontal FoV</b>	87°	108°	96°	95°	154°	88°	108°	98°	116°	110°
<b>Refresh Rate</b>	90Hz	90Hz	120Hz	90Hz	120Hz	80Hz	144Hz	90Hz	120Hz	120Hz
<b>Resolution (per eye)</b>	1080x1200	1080x1200	960 x 1080	1440 x 1440	3840 x 2160	1280x1440	1440 x 1600	2160 x 2160	2448 x 2448	2000 x 2040
<b>Tracking</b>	6-DoF external sensors	6-DoF external sensors (light houses)	6-DoF external sensors (PS Camera)	6-DoF inside out tracking – 2 cameras	6-DoF external sensors (light houses)	6-DoF inside out tracking – 5 cameras	6-DoF external sensors (light houses)	6-DoF inside out tracking – 4 cameras	6-DoF external sensors (light houses)	6-DoF inside out tracking – 4 cameras
<b>Input</b>	Gamepad + 6-DoF Tracked controllers	6-DoF Tracked controllers	Gamepad + 6-DoF Tracked controllers	6-DoF Tracked controllers	6-DoF Tracked controllers	6-DoF Tracked controllers	6-DoF Tracked controllers	6-DoF Tracked controllers + Eye tracking, heartrate, face camera and pupillometry	6-DoF Tracked controllers	6-DoF Tracked controllers + Eye tracking
<b>Platform</b>	Oculus	Steam VR	Playstation 4	Windows Mixed Reality	Steam VR	Oculus	Steam VR	Windows Mixed Reality	Steam VR + HTC Viveport	Playstation 5

## Appendix E

Table 22 - Extended table displaying common standalone VR headsets from 2018-2022. Sourced from VRCompare (2023).

	Oculus Go	Pico G2	HTC Vive Focus	Oculus Quest	Pico New 2 Eye	Meta (previously Oculus) Quest 2	HTC Vive Focus 3	HTC Vive Flow	iQIYI Qiyu Dream	Lenovo Legion VR700
<b>Release Date</b>	May 2018	July 2018	April 2019	May 2019	May 2020	October 2020	June 2021	Nov 2021	Dec 2021	August 2022
<b>Price</b>	\$199	\$249	\$599	\$399	\$899	\$399	\$1300	\$499	\$300	\$440
<b>Horizontal FoV</b>	89°	92°	110°	93°	101°	97°	116°	100°	90°	Unknown
<b>Refresh Rate</b>	60Hz	90Hz	75Hz	72Hz	75Hz	120Hz	90Hz	75Hz	90Hz	90Hz
<b>Resolution (per eye)</b>	1280 x 1440	1440 x 1600	1440 x 1600	1440 x 1600	2048 x 2160 + foveated rendering	1832 x 1920	2448 x 2448	1600 x 1600	1280 x 1440	1832 x 1920
<b>Tracking</b>	3-DoF (rotation)	3-DoF (rotation)	6-DoF inside out tracking – 2 cameras	6-DoF inside out tracking – 4 cameras	6-DoF inside out tracking – 2 cameras	6-DoF inside out tracking – 4 cameras + Passthrough (grayscale)	6-DoF inside out tracking – 4 cameras + Passthrough (grayscale)	6-DoF inside out tracking – 2 cameras + Passthrough (grayscale)	6-DoF inside out tracking – 4 cameras	6-DoF inside out tracking – 4 cameras + Passthrough (grayscale)
<b>Input</b>	3-DoF remote	3-DoF remote	3-DoF remote	6-DoF Tracked controllers	6-DoF Tracked controllers + Eye tracking	6-DoF Tracked controllers + Hand tracking	6-DoF Tracked controllers + Hand tracking	Input via smartphone + optional remote.	6-DoF Tracked controllers	6-DoF Tracked controllers
<b>Platform</b>	Oculus	Pico Store + HTC VivePort	HTC Viveport	Oculus	Pico Store + HTC Viveport	Oculus	HTC Viveport	HTC Viveport	iQIYI / Dream XR	iQUT

## Appendix F



*Faculty of Science and Engineering  
Research Ethics Committee*

Lee Beever  
Dept. of Computer Science  
University of Chester.

21st February 2018

Dear Lee,

**Study title:** Evaluating the accuracy of LevelD AR, an ARKit enabled augmented reality spatial mapping application.  
**FSE-REC reference:** 095/18/LB/CS

**Version number:** 2

Thank you for uploading the amended documentation relating to your application for ethical approval, and for addressing the action points specified in our letter of 20<sup>th</sup> February.

I am pleased to confirm ethical approval for the above research, provided that you comply with the conditions set out in the attached documents, and adhere to the processes described in your application form and supporting documentation.

Please note that this approval is given in accordance with the requirements of English law only. For research taking place wholly or partly within other jurisdictions (including Wales, Scotland and Northern Ireland), you should seek further advice from the Committee Chair/Secretary or the Research and Knowledge Transfer Office and may need additional approval from the appropriate agencies in the country (or countries) in which the research will take place.

Yours sincerely,

A handwritten signature in black ink, appearing to read 'A. Lupton', written over a light blue horizontal line.

**Angela Lupton**  
Chair, Faculty of Science and Engineering Research Ethics Committee

cc. Supervisor

Enc. Research Governance Handbook, Standard Conditions of Approval



## Appendix G



Lee Beever  
Dept. of Computer Science  
University of Chester.  
via email

18<sup>th</sup> January 2019

Dear Lee,

**Study title:** Evaluating the use of data captured by LevelEd AR to create an effective 3D environment for serious and entertainment games.

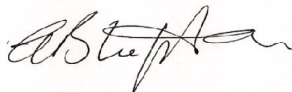
**FSE-REC reference:** 110/18/LB/CS

**Version number:** 2

Thank you for responding to the point raised in my letter of 17<sup>th</sup> January. I am satisfied that the above research project will now meet the necessary criteria, and am therefore pleased to confirm ethical approval, provided that you comply with the conditions set out in the attached documents and adhere to the processes described in your application form and supporting documentation.

Please note that this approval is given in accordance with the requirements of English law only. For research taking place wholly or partly within other jurisdictions (including Wales, Scotland and Northern Ireland), you should seek further advice from the Committee Chair/Secretary or the Research and Knowledge Transfer Office and may need additional approval from the appropriate agencies in the country (or countries) in which the research will take place.

Yours sincerely,



**Angela Lupton**

Chair, Faculty of Science and Engineering Research Ethics Committee

cc. Supervisor

Enc. Research Governance Handbook, Standard Conditions of Approval

## Appendix H



Lee Beever  
Dept. of Computer Science  
University of Chester.  
via email

15<sup>th</sup> May 2019

Dear Lee,

**Study title: User Feedback Study - A study to gather preliminary user feedback on the LevelEd VR system.**  
**FSE-REC reference: 126/19/LB/CS**  
**Version number: 1**

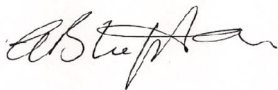
Thank you for the above application which was considered by the Faculty Research Ethics Committee at the meeting held on Tuesday 14<sup>th</sup> May 2019. The Committee is conscious of the work involved in the preparation of such research proposals and is grateful for your attention to this.

**Decision: Approved**

I am pleased to confirm ethical approval, provided that you comply with the conditions set out in the attached documents and adhere to the processes described in your application form and supporting documentation.

Please note that this approval is given in accordance with the requirements of English law only. For research taking place wholly or partly within other jurisdictions (including Wales, Scotland and Northern Ireland), you should seek further advice from the Committee Chair/Secretary or the Research and Knowledge Transfer Office and may need additional approval from the appropriate agencies in the country (or countries) in which the research will take place.

Yours sincerely,



**Angela Lupton**  
Chair, Faculty of Science and Engineering Research Ethics Committee

cc. Supervisor

Enc. Research Governance Handbook, Standard Conditions of Approval

## Appendix I

As the study was very similar to the preliminary feedback study approved above, a whole new application was not required, and it was approved based on the previous application:

### Application

**Title**

LevelEd VR Comparative Study - Comparing LevelEd VR to existing level design techniques

**Applicant**

[Lee Beever](#)

**Status**

Approved

**Last Updated**

11/26/2019 3:15:54 PM

**Submitted**

11/7/2019 10:43:37 AM

**Last Reviewer**

[Pauline Morton](#)

Edit

[Pauline Morton](#)

3 years ago

Hi Lee,

At this morning's Ethics Committee meeting, no ethical issues were identified in your latest application, so no separate approval is required.

Our reviewer did point out, however, that the project end date given on this application was March 2019 (should it be 2020?), and also wondered whether you had allowed yourself enough time to make the necessary arrangements to run your next round of tests in December.

Best wishes,  
Pauline

---

[Lee Beever](#)

3 years ago 

This is fairly similar to the one submitted in April 2019.

## Appendix J



Lee Beever  
via online system

9<sup>th</sup> February 2021

Dear Lee,

**Study title: LevelEd SR Evaluation Study – Evaluating the substitutional reality game experience and the LevelEd SR system which enables substitutional reality games with consumer hardware**

**FSE-REC reference: 165/21/LB/CS**

**Version number: 1**

Thank you for the above application which was considered by the Faculty Research Ethics Committee at the meeting held on Wednesday 3<sup>rd</sup> February 2021. The Committee is conscious of the work involved in the preparation of such research proposals and is grateful for your attention to this.

Thank you also for answering questions regarding your application at the meeting. It is noted that you have agreed to the following:


- (i) that you will ensure your testing location is well-ventilated, in accordance with the measures put in place by the University to combat the spread of Covid19;
- (ii) that you will provide masks (ideally FFP3 style) which offer an effective level of protection to yourself and your participants;
- (iii) that your write-up will mention that the wearing of masks might have affected the results of your tests.

### **Decision: Approved**

I am satisfied that the above research project will meet the necessary criteria, and am therefore pleased to confirm ethical approval, provided that you comply with the conditions set out in the University's Research Governance Handbook (<https://portal1.chester.ac.uk/rkto/PublishingImages/pages/rge-research-governance-handbook/2020%20Research%20Governance%20Handbook%203.4.pdf>) and adhere to the processes described in your application form and supporting documentation.

Please note that this approval is given in accordance with the requirements of English law only. For research taking place wholly or partly within other jurisdictions (including Wales, Scotland and Northern Ireland), you should seek further advice from the Committee Chair/Secretary or the Research and Knowledge Transfer Office and may need additional approval from the appropriate agencies in the country (or countries) in which the research will take place.

Yours sincerely,



**Dr Mark Tomás McAuley** PGCE, SFHEA  
Chair, Faculty of Science and Engineering Research Ethics Committee

cc. Supervisor (Nigel John)

## Appendix K

Dear Sir/Madam,

You have been invited to take part in a research study in the Department of Computer Science at the University of Chester.

The aim of the study is to compare the accuracy and speed of our LevelED AR application for acquiring real world spatial data against a traditional measuring tape and depth sensor enabled application. We will be comparing the accuracy of measurements of 4 measuring tasks and the time taken to complete these measurements for all 3 aforementioned techniques.

The tasks will require some light exercise as you will need to walk around the room to gather measurements. We anticipate the study will take around 45 minutes to complete.

I have attached a Participant Information Sheet that goes into more depth about the study. If you are interested in taking part you must meet the following requirements:

- You are a student/staff at University of Chester
- You must be confident in your ability to physically complete tasks outlined in the “What will happen to me if I take part?” section on the attached Participant Information Sheet.
- You can read, write and speak English.
- 18+ years old. ID may be requested.

The study is expected to run on **05/03/2018 – 09/03/2018** and available slots can be viewed here: . If you would like to take part, please confirm by replying to this email along with your available timeslots.

Kind regards,

Lee Beever



## **Participant information sheet**

### **Evaluating the accuracy of LevelED AR, an ARKit enabled augmented reality spatial mapping application.**

You are being invited to take part in a research study. Before you decide, it is important for you to understand why the research is being done and what it will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like more information. Take time to decide whether or not you wish to take part.

Thank you for reading this.

#### **What is the purpose of the study?**

The project is to evaluate the accuracy of LevelED AR, an ARKit enabled augmented reality smartphone application for spatially mapping and measuring real world environments.

The study will compare the LevelED AR application against 2 alternative techniques for measuring and mapping real world environments. Technique 1 is a tape measure and paper to record measurements. Technique 2 is a depth sensor called Structure. The Structure sensor can spatially map a room using infrared sensors to produce a scale 3D model.

#### **Why have I been chosen?**

You have been chosen because you are an able-bodied adult able to communicate in English.

#### **Do I have to take part?**

It is up to you to decide whether or not to take part. If you decide to take part you will be given this information sheet to keep and be asked to sign a consent form. If you decide to take part you are still free to withdraw at any time and without giving a reason. A decision to withdraw at any time, or a decision not to take part, will not affect you in any way.

#### **What will happen to me if I take part?**

As a participant you will be involved in a single session that should last between 30-45 minutes. Before completing the study you will be presented with a consent form to sign. You must sign the consent form in order to continue.

The study will begin with the completion of a demographics questionnaire. You will then be presented with 4 tasks to complete. Each task requires you to measure a specific object or area. The 4 tasks are as follows:

1. Measure a single wall.
2. Measure a box with open space around all sides.
3. Measure a box against a wall.
4. Measure 4 consecutive walls and box.

You will complete these 4 tasks a total of 3 times using different measuring techniques. The order of task and techniques will be randomised. A 10 minute tutorial will be given before each task is completed to ensure you understand how to use each of the techniques. The 3 measuring techniques are as follows:

**1. Measuring tape and paper.**

You will be expected to complete each of the above measuring tasks by moving around the room to measure them using a traditional measuring tape and recording the measurements on the Tape Measure Data Capture form provided.

**2. LevelED AR application.**

This application runs on a smart phone and requires you to move around the room to place markers in the real world environment using augmented reality. The application can be used in one hand. This will allow you to measure walls and 3D objects. You will be guided through the process using the application. The markers are then saved to a file and uploaded for later viewing and measuring by the researchers

**3. Room Capture application developed by Occipital Inc. (2016).**

This application runs on an iPad Pro with a Structure Sensor attached and requires two hands to hold it. It will allow you to scan the objects for each task by moving the provided iPad camera around the objects. The application will use the attached depth sensor to create a 3D model of the objects. You will use the built in measuring tool to capture the measurements and record them on the Room Capture Data Capture form provided.

As well as the measuring data produced during the tasks, each task will also be timed, to help evaluate the efficiency of each technique. You will have only 1 attempt at each of the tasks with each technique so you should not rush and aim for accuracy of measurements/mapping. You will be told when to start and you will need to let the researcher know when you have completed each task.

For each of the techniques and after the 4 tasks have been completed you will be presented with a System Usability Scale questionnaire to complete. This will allow the usability of each technique to be evaluated.



No one will be identifiable in the final report.

### **What are the possible disadvantages and risks of taking part?**

There are no disadvantages for taking part but there are some minor risks:

All 3 of the techniques require you to move around the room to measure and map the given targets for each task. Whilst focusing on the task there is a risk of tripping. However, the research team has made every possible effort to reduce this risk by ensuring a clear area is available for the tasks and a researcher will monitor you throughout the tasks to guide you away from any potential obstacles.

A first aider will be on site in the unlikely event of any accidents.

### **What are the possible benefits of taking part?**

The benefits for taking part are your contribution to the body of knowledge through the research and the further development and improvement of LevelED AR.

### **What if something goes wrong?**

If you wish to complain or have any concerns about any aspect of the way you have been approached or treated during the course of this study, please contact the Executive Dean of the Faculty of Science and Engineering.

### **Will my taking part in the study be kept confidential?**

All information which is collected about you during the course of the research will be kept strictly confidential so that only the researcher carrying out the research will have access to such information. All data both raw and final, generated by the study will remain under the custodianship of the University of Chester securely on a networked server.

### **What will happen to the results of the research study?**

The results may be written up into an academic paper and/or presented at an academic conference. The results will also be written up in the final PhD thesis. Individuals who participate will not be identified in any subsequent report or publication.

### **Who is organising the research?**

The research is conducted as part of an MPhil/PhD in the field of Mixed Reality Level Design and World Building within the Department of Computer Science at the University of Chester. The study is organised with supervision from the department, by Lee Beever, an MPhil/PhD student.

**Who may I contact for further information?**

If you would like more information about the research before you decide whether or not you would be willing to take part, please contact:

*Lee Beever*: email redacted

*Nigel John (Principle Supervisor)*: email redacted

*Serban Pop (Co-supervisor)*: email redacted

**Retention and publication of participant's data**

Participants should note that data collected from this project may be retained and published in an anonymised form. By agreeing to participate in this project, you are consenting to the retention and publication of data.

**Thank you for your interest in this research.**

## Appendix L

Dear Sir/Madam,

You have been invited to take part in a research study in the Department of Computer Science, Electronics and Electrical Engineering at the University of Chester.

The study aims to evaluate the effectiveness of our LevelEd VR system compared to existing level design workflows and techniques. The LevelEd VR system allows users to design levels for virtual reality games whilst in virtual reality. You will be designing games for a target shooting game, which uses non-graphic targets and non-realistic guns.

The tasks will require some light exercise as you will need to use motion controllers to interact with the system, however, you may be seated for this. The study is expected to take around 65-75 minutes to complete.

I have attached a Participant Information Sheet that goes into more depth about the study. If you are interested in taking part, you must meet the following requirements:

- You have experience developing games using Unity 3D.
- You must be confident in your ability to use motion controllers.
- You are happy to play a target shooting game.
- You can read, write and speak English.
- 18+ years old. ID may be requested.

The study will run January 6<sup>th</sup>-24<sup>th</sup> 2020. Available timeslots can be found here:

If you would like to take part, please confirm by replying to this email along with your preferred timeslot.

Kind regards,

Lee Beever



## **Participant information sheet**

### **LevelEd VR Comparative Study – Comparing LevelEd VR to existing level design techniques.**

You are being invited to take part in a research study. Before you decide, it is important for you to understand why the research is being done and what it will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like more information. Take time to decide whether or not you wish to take part.

Thank you for reading this.

#### **What is the purpose of the study?**

The LevelEd VR system allows users to plan, design and develop game levels for virtual reality (VR) games whilst in VR using a head mounted display (HMD) and motion controllers. The study looks to compare the LevelEd VR system against existing level design techniques to evaluate its effectiveness as a virtual reality game level design tool.

#### **Why have I been chosen?**

You have been chosen because you are an adult able to communicate in English and have prior experience developing games in Unity.

#### **Do I have to take part?**

It is up to you to decide whether or not to take part. If you decide to take part, you will be given this information sheet to keep and be asked to sign a consent form. If you decide to take part, you are still free to withdraw at any time and without giving a reason. A decision to withdraw at any time, or a decision not to take part, will not affect you in any way. Choosing to take part or not, will have no impact on your studies at University of Chester if you are currently a student.

#### **What will happen to me if I take part?**

As a participant you will be involved in a single session that should last around 75-minutes. Before completing the study, you will be presented with a consent form to sign. You must sign the consent form in order to continue.

The study will begin with the completion of a demographics questionnaire and a simulator sickness questionnaire. You will then be asked to play a simple example level that demonstrates the game you will be creating a level for. The game requires you to shoot at a number of targets (round bullseye targets) whilst being timed. Movement around the level takes place by teleporting to specific nodes. There will be no graphic content in the game, but if you are uncomfortable shooting a virtual gun then you should not consent to participate in the study.

You will then complete the following tasks; the order of completion will be randomly selected.

1. You will be introduced to the LevelEd VR system with a 10-minute tutorial. This will introduce the locomotion method, menu system and basic feature set of the system. Once you are satisfied with the basic functions of the system you will be asked to spend 15-minutes (or less) creating a level for the aforementioned game using the tools. If you wish to terminate your evaluation of the system you may do so at any point. Telemetry data of how you use the system will be captured and stored along with the final level created. Once complete, you will be asked to complete a simulator sickness questionnaire again, a system usability scale questionnaire and a general feedback questionnaire which will allow you to give feedback on the system for future improvements.
2. You will be introduced to the Sample Game Unity project files. You will be given a 10-minutes to become accustomed to the game prefabs and ProBuilder. You will then be given 15-minutes (or less) to create a level using the Unity game engine, ProBuilder and the supplied prefabs. If you wish to terminate your evaluation of the system you may do so at any point. Telemetry data of how you use the system will be captured and stored along with the final level created. Once complete, you will be asked to complete a simulator sickness questionnaire again and a system usability scale questionnaire.

Once both tasks are completed you will be asked to complete the Comparative Questionnaire.

Video of the screen and you will also be recorded to allow for detailed analysis of your workflow and processes after the study is complete. Images from this footage will not be published or displayed elsewhere without additional consent from you.

All feedback and questionnaire data will be anonymised. Data (questionnaires and final levels) may be used in a future publication or as part of the PhD thesis, however, no one will be identifiable in the final report.

### **What are the possible disadvantages and risks of taking part?**

There are no disadvantages for taking part but there are some minor risks:

1. There is potential for participants to feel discomfort and/or varying degrees of simulator sickness as a result of using the system. If at any point the discomfort becomes too

great, participants will be asked to remove the headset immediately. There will be seating and water available for any, unlikely extreme cases.

2. There is a limited risk for damage (to the equipment) or personal injury to occur during the study. The system uses a built-in chaperone system to help avoid the issue of users walking into or hitting walls/objects around them. However, the facilitator will also monitor the user to ensure they don't ignore the chaperone system.

A first aider will be on site in the unlikely event of any accidents.

### **What are the possible benefits of taking part?**

The benefits for taking part are your contribution to the body of knowledge through the research and the further development and improvement of LevelEd VR system.

### **What if something goes wrong?**

If you wish to complain or have any concerns about any aspect of the way you have been approached or treated during the course of this study, please contact the Executive Dean of the Faculty of Science and Engineering, University of Chester.

### **Will my taking part in the study be kept confidential?**

All information which is collected about you during the course of the research will be kept strictly confidential so that only the researcher carrying out the research will have access to such information. All data both raw and final, generated by the study will remain under the custodianship of the University of Chester securely on a networked server.

### **What will happen to the results of the research study?**

The results may be written up into an academic paper and/or presented at an academic conference. The results will also be written up in the final PhD thesis. Individuals who participate will not be identified in any subsequent report or publication.

### **Who is organising the research?**

The research is conducted as part of a PhD in the field of Mixed Reality Level Design Workflows within the Department of Computer Science at the University of Chester. The study is organised with supervision from the department, by Lee Beever, an PhD student.

### **Who may I contact for further information?**

If you would like more information about the research before you decide whether or not you would be willing to take part, please contact:

*Lee Beever:* redacted email

*Nigel John (Principle Supervisor): redacted email*

*Serban Pop (Co-supervisor): redacted email*

**Retention and publication of participant's data**

Participants should note that data collected from this project may be retained and published in an anonymised form. By agreeing to participate in this project, you are consenting to the retention and publication of data.

**Thank you for your interest in this research.**



## Appendix M

Dear Sir/Madam,

You have been invited to take part in a research study in the Department of Computer Science at the University of Chester.

The study explores substitutional reality and the impact on the game experience along with the evaluation of a substitutional reality editor and workflow. Substitutional reality in the context of this study takes the form of virtual reality that has been enhanced by passive haptics (inclusion of physical objects in the player space).

The tasks will require some light exercise as you will need to use motion controllers to interact with the system and move around the room. The study is expected to take around 90-100 minutes to complete.

I have attached a Participant Information Sheet that goes into more depth about the study. If you are interested in taking part, you must meet the following requirements:

- You must be confident in your ability to use motion controllers.
- You are happy to play a target shooting game.
- You can read, write and speak English.
- 18+ years old. ID may be requested.

The study will run July 20<sup>th</sup> – August 13<sup>th</sup> on the Exton Park (Parkgate Road) campus in CBB012. Available timeslots can be found here:

<https://docs.google.com/spreadsheets/d/1qXUQVrXWYZ6U46idE1wEcSukgSrf-UfAmcMtOf5Qxa4/edit?usp=sharing>

If you would like to take part, please sign up at the following location:

<https://forms.office.com/r/rPQSvKhxjx>

Kind regards,

**Lee Beever**

**University of Chester**

**Department of Computer Science, Electronics and Electrical Engineering**



## **Participant information sheet**

### **LevelEd SR Evaluation Study – Evaluating the substitutional reality game experience and the LevelEd SR system which enables substitutional reality games with consumer hardware**

You are being invited to take part in a research study. Before you decide, it is important for you to understand why the research is being done and what it will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like more information. Take time to decide whether or not you wish to take part.

Thank you for reading this.

#### **What is the purpose of the study?**

The LevelEd SR system allows users to create their own substitutional reality (SR) levels for an existing game. Substitutional reality in this context refers to the inclusion of passive haptics (static objects that you can touch like furniture) into the virtual reality world. The study is looking to explore the impacts of SR on the game experience and to evaluate our LevelEd SR system which includes the ability to capture a 3D virtual model of a room and the ability to create and test game levels from an Oculus Quest device.

#### **Why have I been chosen?**

You have been chosen because you are an adult able to communicate in English.

#### **Do I have to take part?**

It is up to you to decide whether or not to take part. If you decide to take part, you will be given this information sheet to keep and be asked to sign a consent form. If you decide to take part, you are still free to withdraw at any time and without giving a reason. A decision to withdraw at any time, or a decision not to take part, will not affect you in any way. Choosing to take part or not, will have no impact on your studies at University of Chester if you are currently a student.

#### **What will happen to me if I take part?**

As a participant you will be involved in a single session that should last around 90-100 minutes. Before completing the study, you will be presented with a consent form to sign. You must sign the consent form in order to continue.

The study will take place over two parts.

The first part will be as follows and the order you will complete this will be randomised:

1. You will play a VR target shooting game (no gore, only non-humanoid targets) that involves no passive haptics.
2. You will play a SR target shooting game (no gore, only non-humanoid targets) that will involve the use of passive haptics to create an SR experience.

The second part will be as follows. Point 1 and 2 below will have a randomised order:

1. You will be given a tutorial before capturing a real-world location (the room) using our LevelEd AR software on a provided smartphone using a manual mesh creation system.
2. You will be given a tutorial before capturing a real-world location (the room) using our LevelEd AR software on a provided smartphone using a LiDAR powered mesh generation system.
3. You will then get a tutorial on how to use the LevelEd SR level editor which will run in VR using the Oculus Quest headset to create a substitutional reality level for the aforementioned target shooting game.
4. Finally, you will develop your own substitutional reality level for the aforementioned target shooting game using the LevelEd SR level editor and play test it using the headset.

Throughout the above you will complete several different questionnaires in order for us to capture data about your experience.

### **What are the possible disadvantages and risks of taking part?**

There are no disadvantages for taking part but there are some minor risks:

1. There is potential for participants to feel discomfort and/or varying degrees of simulator sickness as a result of using the system. If at any point the discomfort becomes too great, participants will be asked to remove the headset immediately. There will be seating available for any, unlikely extreme cases.
2. There is a limited risk for damage (to the equipment) or personal injury to occur during the study. The system uses a built-in chaperone system to help avoid the issue of users walking into or hitting walls/objects around them. However, the facilitator will also monitor the user to ensure they don't ignore the chaperone system.

A first aider will be on site in the unlikely event of any accidents.

We will also adhere to the current COVID-19 government guidelines and take the following precautions:

The following procedures are proposed to ensure the study is COVID secure:

- Participants and researchers will be required to wear face coverings at all times (the researcher will also use a visor).
- The room will be ventilated as well as possible (windows opened).
- Participants will be asked if they have any COVID-19 symptoms or if they should be self-isolating before being allowed into the room.
- Participants and the researcher will use hand sanitizer continually throughout the study.
- Equipment and surfaces will be sanitized after the study using chemical cleaners/wipes.
- The VR HMD and controllers will also be sanitized using the research teams' Cleanbox (<https://cleanboxtech.com/>). This system uses UVC light to sanitize devices including COVID-19 with 99.999 +% efficacy.
- Where possible, only one study will run per day.

### **What are the possible benefits of taking part?**

The benefits for taking part are your contribution to the body of knowledge through the research and the further development and improvement of LevelEd SR system.

### **What if something goes wrong?**

If you wish to complain or have any concerns about any aspect of the way you have been approached or treated during the course of this study, please contact the Executive Dean of the Faculty of Science and Engineering, University of Chester.

### **Will my taking part in the study be kept confidential?**

All information which is collected about you during the course of the research will be kept strictly confidential so that only the researcher carrying out the research will have access to such information. All data both raw and final, generated by the study will remain under the custodianship of the University of Chester securely on a networked server.

### **What will happen to the results of the research study?**

The results may be written up into an academic paper and/or presented at an academic conference. The results will also be written up in the final PhD thesis. Individuals who participate will not be identified in any subsequent report or publication.

### **Who is organising the research?**

The research is conducted as part of a PhD in the field of Mixed Reality Level Design Workflows within the Department of Computer Science at the University of Chester. The study is organised with supervision from the department, by Lee Beever, a PhD student.

### **Who may I contact for further information?**

If you would like more information about the research before you decide whether or not you would be willing to take part, please contact:

*Lee Beever*: redacted email

*Nigel John (Principal Supervisor)*: redacted email

*Serban Pop (Co-supervisor)*: redacted email

**Retention and publication of participant's data**

Participants should note that data collected from this project may be retained and published in an anonymised form. By agreeing to participate in this project, you are consenting to the retention and publication of data.

**Thank you for your interest in this research.**

# Appendix N

## LevelEd AR Code Samples

1. Example code that uses a raycast against a plane or point-of-interest to create a new (or move an existing) marker point.

```
while (bPlacingMarkers)
{
    var screenPosition = mainCamera.ScreenToWorldPoint (new Vector3 (Screen.width / 2, Screen.height / 2,
mainCamera.transform.position.z));

    ARPoint point = new ARPoint {
        x = screenPosition.x,
        y = screenPosition.y
    };

    ARHitTestResultType[] resultTypesPrioritised = {
        ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent,
        ARHitTestResultType.ARHitTestResultTypeExistingPlane,
        ARHitTestResultType.ARHitTestResultTypeHorizontalPlane,
        ARHitTestResultType.ARHitTestResultTypeFeaturePoint
    };

    Vector3 hitLocation;

    foreach (ARHitTestResultType resultType in resultTypesPrioritised)
    {
        if (HitTestWithResultType (point, resultType, out hitLocation))
        {
            spawnLocation = new Vector3 (hitLocation.x, markerHeight, hitLocation.z);

            if (guideMarker == null)
            {
                guideMarker = Instantiate (markerPrefab, spawnLocation, Quaternion.identity);
                guideMarker.GetComponent<Renderer> ().material.color = guideColour;
            }
            else
            {
                guideMarker.SetActive (true);
                guideMarker.transform.position = spawnLocation;
                if (guideLineRendererComponent && markerGroupCount == 1)
                {
                    guideLineRendererComponent.SetPosition (1,
                    guideMarker.transform.position);
                }
            }
            break;
        }
        else
        {
            guideMarker.SetActive (false);
        }
    }

    if (Input.touchCount > 0 && !UnityEngine.EventSystems.EventSystem.current.IsPointerOverGameObject
(Input.GetTouch (0).fingerId))
    {
        var touch = Input.GetTouch (0);
        if (touch.phase == TouchPhase.Began)
        {
            CreateMarker (new Vector3 (spawnLocation.x, spawnLocation.y, spawnLocation.z));
        }
    }

    if (markerGroupCount >= 2)
    {
        bPlacingMarkers = false;
    }

    yield return null;
}
```

## 2. Example code demonstrating the save system for LevelEd AR.

```
/// <summary>
/// Function takes the data provided, extracts the locations of the markers and stores them in an
/// object.
/// The object is serialised to a file before being uploaded to a webserver
/// </summary>
/// <param name="freeformList">stores markers for objects</param>
/// <param name="continuousList">stores markers for walls</param>
/// <param name="participantNumber">the participant number in the study</param>
/// <param name="testType">what part of the study task does the data relate to</param>

public void SaveFile(List<FreeformListG> freeformList, List<ContinuousListG> continuousList, string
participantNumber, string testType)
{
    fileName = "/P_" + participantNumber + "_" + testType + ".dat";
    BinaryFormatter bf = new BinaryFormatter ();
    FileStream saveFile = File.Create (Application.persistentDataPath + fileName);
    WorldSaveData data = new WorldSaveData ();

    if (freeformList != null && testType == "Task2Box" || testType == "Task3BoxWall" || testType
    == "Task4Walls")
    {
        for (int i = 0; i < freeformList.Count; i++)
        {
            FreeformListV tempVectorList = new FreeformListV();
            data.freeformPointLocationList.Add (tempVectorList);
            for (int j = 0; j < freeformList[i].freeformList.Count; j++)
            {
                PositionV pos = new PositionV
                (freeformList[i].freeformList[j].transform.position.x,
                freeformList[i].freeformList[j].transform.position.y,
                freeformList[i].freeformList[j].transform.position.z);

                data.freeformPointLocationList[i].freeformList.Add (pos);
            }
        }
    }

    if (continuousList != null && testType == "Task4Walls" || testType == "Task1Plane")
    {
        for (int i = 0; i < continuousList.Count; i++)
        {
            ContinuousListV tempVectorList = new ContinuousListV();
            data.continuousPointLocationList.Add (tempVectorList);
            for (int j = 0; j < continuousList[i].continuousGList.Count; j++)
            {
                PositionV pos = new PositionV
                (continuousList[i].continuousGList[j].transform.position.x,
                continuousList[i].continuousGList[j].transform.position.y,
                continuousList[i].continuousGList[j].transform.position.z);

                data.continuousPointLocationList[i].continuousList.Add(pos);
            }
        }
    }

    bf.Serialize (saveFile, data);
    saveFile.Close ();
    UploadFile ();
}

void UploadFile()
{
    string location = Application.persistentDataPath + fileName;
    ftpClient.upload(fileName, @location);
}
}
```



### 3. Example code demonstrating the data structures used to store the mapped objects/walls.

```
//Stores Vector3 data separately
[System.Serializable]
public class PositionV
{
    public float x = 0;
    public float y = 0;
    public float z = 0;

    public PositionV(float posX, float posY, float posZ)
    {
        x = posX;
        y = posY;
        z = posZ;
    }
}

//A list of vertices used to create objects
[System.Serializable]
public class FreeformListV
{
    public List<PositionV> freeformList = new List<PositionV>();
}

//A list of vertices used to create walls
[System.Serializable]
public class ContinuousListV
{
    public List<PositionV> continuousList = new List<PositionV> ();
}

//A list of marker GameObjects for objects
[System.Serializable]
public class FreeformListG
{
    public List<GameObject> freeformList = new List<GameObject>();
}

//A list of marker GameObjects for walls
[System.Serializable]
public class ContinuousListG
{
    public List<GameObject> continuousGList = new List<GameObject>();
}

//creates a new class called WorldSaveData to store all scene data
[System.Serializable]
class WorldSaveData
{
    //Each mapped object has its vertices stored in a list.
    //Each list is then stored in a list below
    public List<ContinuousListV> continuousPointLocationList = new List<ContinuousListV> ();
    public List<FreeformListV> freeformPointLocationList = new List<FreeformListV>();
}
}
```

## Appendix O

### LevelEd VR Code Samples

1. A template to demonstrate how each tool is constructed using a state system.

Further example code can be seen in Appendix B3i.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using VRTK;
public class SCR_TemplateTool : MonoBehaviour, ITool {

    private enum ToolStates { State1, State2 };
    private ToolStates currentState = ToolStates.State1;

    [SerializeField] private VRTK_ControllerEvents.ButtonAlias activationButton =
VRTK_ControllerEvents.ButtonAliases.TriggerPress;
    [SerializeField] private VRTK_ControllerEvents controllerEvents;
    [SerializeField] private string toolName;

    bool bBusy;
    public bool Busy { get { return bBusy; } }

    private void OnEnable()
    {
        //set this as the current tool and enable/disable grid snapping, surface snapping, etc.
        //get a reference to the variable pointer and subscribe to the appropriate controller events
    }

    private void OnDisable()
    {
        //unsubscribe from controller events
    }

    void DoActivationButtonPressed(object sender, ControllerInteractionEventArgs e)
    {
        if (SCR_LevelEditorManager.instance.CurrentEditorState == EditorState.LevelEditor)
        {
            //deal with button press when in level editor mode only
        }
    }
    // Use this for initialization
    void Start()
    {
        //get a reference to the variable pointer object
    }
    // Call the current function for the give state of the tool. Could be more than two.
    void Update()
    {
        switch (currentState)
        {
            case ToolStates.State1:
                State1Code();
                break;
            case ToolStates.State2:
                State2Code();
                break;
            default:
                break;
        }
    }
    void State1Code()
    {
        //Add code here that should run during this tool state
    }
    void State2Code()
    {
        //Add code here that should run during this tool state
    }
}
```

## Appendix P

### LevelEd AR Code Samples

1. Example code (edited for brevity) used to place the marker point by raycasting against either an AR plane or point-of-interest using AR Foundation SDK.

```
List<ARRaycastHit> arHits = new List<ARRaycastHit>();
ARRaycastManager m_RaycastManager;
TrackableType trackableTypeToUse;

//Coroutine to deal with making 3D objects, not walls
IEnumerator FreeformPlaceMarkers()
{
    if (!bFreeformUndoFromVertical)
    {
        currentFreeformList = null;
        currentFreeformList = new FreeformListG();
        markerFreeformList.Add(currentFreeformList);
        freeformPlacedMarkerCountCurrent = 0;
        freeformObjectCount++;
        freeformPanelConfirmButton.SetActive(false);
        undoButtonFreeform.SetActive(true);
    }
    else
    {
        bFreeformUndoFromVertical = false;
    }

    bCreatingFreeformBase = true;
    freeformDescription.text = "Aim using the cube. Tap screen to place each marker cube.";

    while (bCreatingFreeformBase)
    {
        Vector2 screenPosition = new Vector2(Screen.width / 2, Screen.height / 2);

        if (m_RaycastManager.Raycast(screenPosition, arHits, trackableTypeToUse))
        {
            var hitLocation = arHits[0].pose;
            spawnLocation = new Vector3 (hitLocation.position.x, markerHeight,
            hitLocation.position.z);

            if (guideMarker == null)
            {
                guideMarker = Instantiate (markerPrefab, spawnLocation,
                Quaternion.identity);
                guideMarker.GetComponent<Renderer> ().material.color = guideColour;
            }
            else
            {
                guideMarker.SetActive (true);
                guideMarker.transform.position = spawnLocation;
                if (guideLineRendererComponent &&
                freeformPlacedMarkerCountCurrent != 0)
                {
                    guideLineRendererComponent.SetPosition
                    (freeformPlacedMarkerCountCurrent,
                    guideMarker.transform.position);
                }
            }
        }
        else
        {
            guideMarker.SetActive (false);
        }
    }
}
```

## Appendix Q

### LevelEd AR Code Samples

1. Example code that shows only the required meshes being extracted from the complete list of generated meshes. Checking the material type was found to be the most reliable technique.

```
/// <summary>
/// Takes the meshes generated by ARKit (via AR Foundation) and extracts only the meshes identified as
/// - Seat/chair
/// - Table
/// - None (any other obstacle that is not the floor, ceiling or wall)
/// The mesh data is serialised using a script by Pranckevicius (2015)
/// </summary>
/// <param name="listOfMeshes">The meshes generated by ARKit and the LiDAR sensor</param>
public void SaveFileQuestMeshing(List<MeshFilter> listOfMeshes)
{
    meshingData = new WorldSaveDataMesh();

    if (listOfMeshes != null)
    {
        foreach (var item in listOfMeshes)
        {
            string materialName =
                item.gameObject.GetComponent<MeshRenderer>().sharedMaterial.name;

            if (materialName == "MeshMaterial_None" || materialName == "MeshMaterial_Seat"
                || materialName == "MeshMaterial_Table")
            {
                MeshList mesh = new MeshList();
                mesh.meshData = SCR_MeshSerializer.WriteMesh(item.mesh, true);
                meshingData.listOfMeshData.Add(mesh);
            }
        }
    }
}
```

# Appendix R

## LevelEd AR Code Samples

### 1. Example code the shows the data being saved for the Meshing.

```
/// <summary>
/// After the required meshes are extracted above, this function is called and goes through the
/// process of saving the data. It saves the meshes, the controller locations and the time to complete
/// (for the study) as a binary file before uploading to a server.
/// </summary>
/// <param name="participantNumber">The number of the current participant completing study</param>
/// <param name="testType">Is this Meshing or Outline</param>
/// <param name="controllerLocations">Location of the two controllers used for syncing</param>
/// <param name="timeToComplete">How long it took the participant to complete the task</param>
public void SaveFileQuestMeshingFinal(string participantNumber, string testType, ControllerListT
controllerLocations, float timeToComplete)
{
    fileName = "/P_" + participantNumber + "_" + testType + ".dat";
    BinaryFormatter bf = new BinaryFormatter();
    FileStream saveFile = File.Create(Application.persistentDataPath + fileName);

    if (meshingData == null)
    {
        meshingData = new WorldSaveDataMesh();
    }

    if (controllerLocations != null)
    {
        meshingData.controllerPositions.leftControllerLocation =
            controllerLocations.leftControllerLocation;
        meshingData.controllerPositions.rightControllerLocation =
            controllerLocations.rightControllerLocation;
    }
    else
    {
        print("controllers are empty");
    }

    meshingData.timeToComplete = timeToComplete;
    bf.Serialize(saveFile, meshingData);
    saveFile.Close();
    UploadFile();
}

void UploadFile()
{
    string location = Application.persistentDataPath + fileName;
    ftpClient.upload(fileName, @location);
}
```