

cdr

LevelEd VR: A virtual reality level editor and workflow for virtual reality level design

| | |
|---------------|---|
| Item Type | Conference Contribution |
| Authors | Beever, Lee;Pop, Serban W.;John, Nigel W. |
| Citation | Beever, L., Pop, S. W. & John, N, W. (2020, August 24-27). LevelEd VR: A virtual reality level editor and workflow for virtual reality level design (pp. 136-143). 2020 IEEE Conference on Games, Osaka, Japan. https://doi.org/10.1109/CoG47356.2020.9231769 |
| DOI | 10.1109/CoG47356.2020.9231769 |
| Publisher | IEEE |
| Rights | Attribution-ShareAlike 4.0 International |
| Download date | 2026-05-11 09:39:59 |
| Item License | http://creativecommons.org/licenses/by-sa/4.0/ |
| Link to Item | http://hdl.handle.net/10034/623487 |

LevelEd VR: A virtual reality level editor and workflow for virtual reality level design

Lee Beever

Department of Computer Science,
Electronics and Electrical Engineering
University of Chester
Chester, United Kingdom
l.beever@chester.ac.uk

Serban Pop

Department of Computer Science,
Electronics and Electrical Engineering
University of Chester
Chester, United Kingdom
s.pop@chester.ac.uk

Nigel W. John

Department of Computer Science,
Electronics and Electrical Engineering
University of Chester
Chester, United Kingdom
nigel.john@chester.ac.uk

Abstract— Virtual reality entertainment and serious games popularity has continued to rise but the processes for level design for VR games has not been adequately researched. Our paper contributes LevelEd VR; a generic runtime virtual reality level editor that supports the level design workflow used by developers and can potentially support user generated content. We evaluated our LevelEd VR application and compared it to an existing workflow of Unity on a desktop. Our current research indicates that users are accepting of such a system, and it has the potential to be preferred over existing workflows for VR level design. We found that the primary benefit of our system is an improved sense of scale and perspective when creating the geometry and implementing gameplay. The paper also contributes some best practices and lessons learned from creating a complex virtual reality tool, such as LevelEd VR.

Keywords—virtual reality, level design, level editor

I. INTRODUCTION

With the continued rise in popularity of virtual reality (VR) entertainment and serious games, the level design workflow and processes for developing VR game levels requires further research. 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 [1]. 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 [2]. A common design technique used by level designers is to create prototype layouts called blockouts (see Fig. 1a) in order to ensure they retain the ability to quickly edit and iterate on designs [3][4]. Gameplay is then constructed and tested within the blockout and confirmed before any final art is added [5].

The current workflow for creating game levels for virtual reality entertainment or serious games is to use a game engine such as Unity or Unreal Engine. Users construct a level in the editor on their 2D screen and then are required to test the game in a virtual reality 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 [6]. Binocular vision (supported by stereoscopic 3D HMDs) along with occlusion, oculomotor depth cues and linear perspective also differ [7]. We believe there are benefits and efficiencies to moving the VR level design workflow into virtual reality. 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.

User generated content in games has also continued to gain in popularity with successful games such as, Minecraft, Little Big Planet, Fortnite and Roblox [8]. We see an opportunity for VR entertainment and serious games to continue this trend. However, designing and implementing a level editor for a game is time consuming and costly. A level editor that can also be bundled with a game would help solve this issue.

To better understand virtual reality level design workflows we have created a generic runtime virtual reality level editor that enables users to develop and design game levels in VR. To meet the needs of a level designer the system needs to: 1) Enable the quick creation of prototype blockout layouts with robust geometry editing tools with accurate depth, scale and perspectives for VR. 2) Enable the ability to place and script gameplay elements. 3) Enable the ability to quickly play test the level for a rapid and agile iterative workflow. To ensure the system can also be used for user generated content by players and users it also needs 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. To meet these needs we developed LevelEd VR.

II. RELATED WORK

To our knowledge there are no academic works focusing specifically on exploring or evaluating level design and level editing in virtual reality. However, there are some works that look at components or areas of this process.

There are several works that look at world building in virtual reality for games or adjacent disciplines. Genesys [9] is a world builder that utilises premade assets and focuses on building worlds for animation and storytelling. Their work focuses initially on evaluating the leap motion input system against real life for block manipulation. The wonderland builder [10] made use of a pointing device and voice to interact and allow the placement of premade objects in a scene and is focused primarily on children. Wang and Lindeman's work on the DIY World Builder [11] produced a more sophisticated world builder with terrain creation, object placement and manipulation, however, the focus was more on testing different interaction techniques and there was no gameplay or play testing. Work by Moreira [12] looked at

world building in VR using voice commands to generate either primitive or premade assets from Google Poly/Sketchfab. They showed success with utilising both the voice commands and manual manipulation. VR Safari Park [13] created a novel way for creating worlds through the placement of topic blocks on a tree like structure. The placement of these blocks generates a world for the user. The interface was easy to use but limits what users can create. Whilst not level design or games focused, Ekströmer et al. [14] 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 on manipulation of the lights. Whilst all of these systems demonstrate some form of world building (a large part of the level design process), they rarely allowed for full blockout/layout modelling, which limits their usefulness for level design, and none looked at creating and testing gameplay in VR.

Other works have focused more on the 3D modelling and creation aspect in virtual reality. MakeVR [15] allows users to create new 3D models with similar functionality to those found in commercial 3D modelling applications. Lift-Off [16] 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. They found some success with this approach. Mendes et al. [17] looked at supporting Boolean operations (cutting meshes where they intersect) whilst in VR using hand and arm tracking. The Boolean operations worked well, but they struggled with their input method. These works demonstrate potential for blockout/layout creation in VR, but do not demonstrate creating or testing gameplay in VR.

More research appears to have been completed in the augmented reality space in regard to creating gameplay. i.Ge [18] uses a projector and Kinect to allow for level design and editing using the real world as a location to good effect. Work by Park, Son, Seo and Park [19] utilises the HoloLens along with markers and paper interfaces to create AR interactions. Ng, Shin, Plopski, Sandor and Saakes utilise the HoloLens to allow for level creation using the real world space and includes the placement of virtual objects and scripting [20]. Some lessons are transferable, but they focus on creating levels for real world environments and not for fully immersive digital environments.

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 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.

It is clear from our analysis of the literature that there is a lack of research focusing on exploring a full level design workflow and editing in virtual reality, that encompasses gameplay rather than just world building. To advance this area of research the contributions of this paper are as follows:

1. The design and implementation of a prototype generic runtime virtual reality 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.

III. SYSTEM OVERVIEW

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 is only available currently in a limited capacity. 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 (degrees of freedom) interactions and the spatial nature of VR.

LevelEd VR was built with Unity and supports games built in this engine. Unity 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 [21]. 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% [22]. VRTK [23] 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 below.

A. Locomotion

There are three forms of locomotion supported in LevelEd VR. The first is teleporting; this allows users to instantly move from one location to another and is aimed at moving over large distances. The second is the Drag World [23] method where users can grip their controllers to move or rotate (but not scale) the world around them. This allows for precise movement over short and medium distances. A vignette is used to limit peripheral vision based on velocity and angular velocity with the aim to reduce simulator sickness. Finally, within space limitations, users can move and turn in the real world for small distances or to quickly look around.

B. Menu and Interaction

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 (Fig. 1b). We opted for a beam solution as we felt this offered more precision. 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 (Fig. 1c).

C. Level Editing Tools

Table I summarises the robust set of tools featured in LevelEd VR.

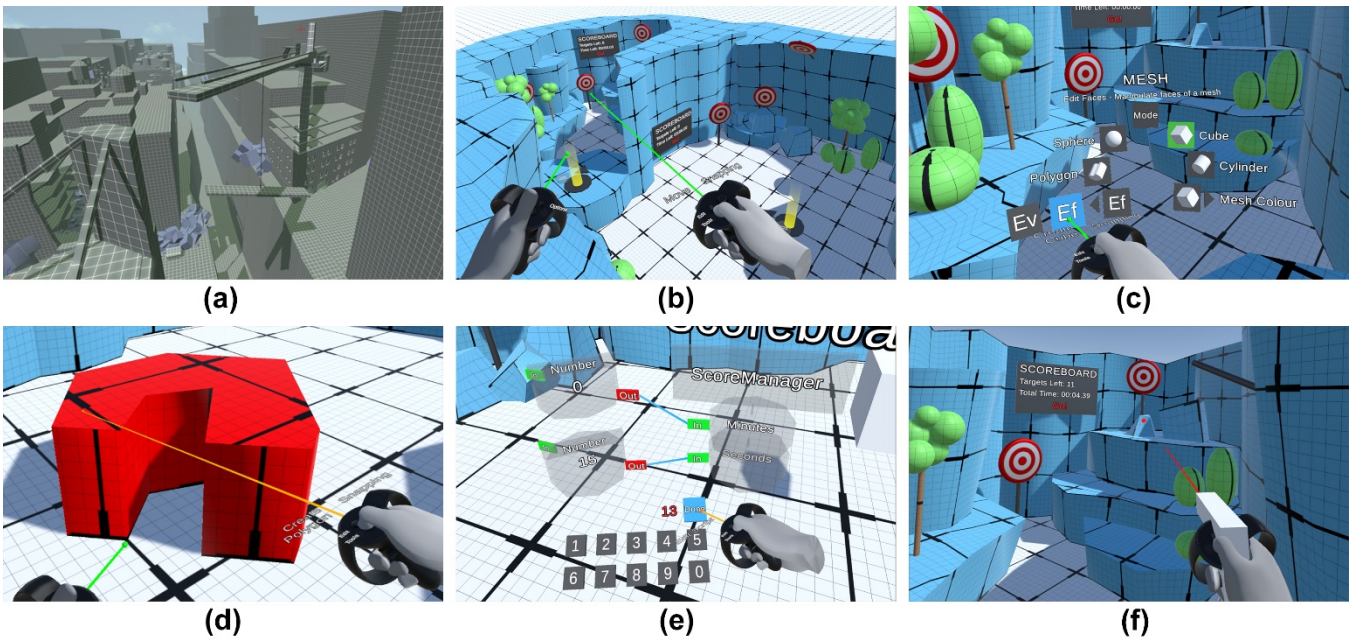


Fig. 1. (a) Example of a blockout level created for the game Enslaved (2010). (b) Example of a blockout level created for the Sample Game using LevelEd VR. (c) The spatial menu system. (d) The Polygon creation tool in use. (e) The spatial scripting system and a number node value being changed. (f) The level being play tested.

Geometry/Layout: These tools enable a level designer to create procedural static meshes to blockout a level layout. They can create meshes, edit their vertices/faces and change their colour (see Fig. 1b and 1d).

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 (Fig. 1e). We have opted 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

Editing/Options: Users can make use of several editing tools (see Table I) which can be used on any instance of an object in the level. There are also several options available such as locking to a grid or not, or which axis to scale/rotate on.

D. Sample Game

A sample game was created in order 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 Fig. 1f). 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. 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.

E. 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 structure 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 (Fig 1f).

IV. EXPERIMENT METHODOLOGY

A study was designed to evaluate the LevelEd VR system for usability and effectiveness at the task of enabling level design in virtual reality. 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 VR level design workflow 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.

A. Experiment Structure

Participants began by spending five minutes playing the virtual reality 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 of creating a game level for the sample game with the LevelEd VR system (complete VR workflow) and another level using Unity on desktop with the aid of the ProBuilder package (referred henceforth as Unity Desktop). 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 requirements for the level (except that it should be playable) and they could choose to create a different level using each system.

TABLE I. LEVELLED VR FEATURES FOR STUDY

| Geometry | Script/Prefab | Editing | Options/Modes |
|-----------------|---------------|---------------|------------------|
| 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 | |

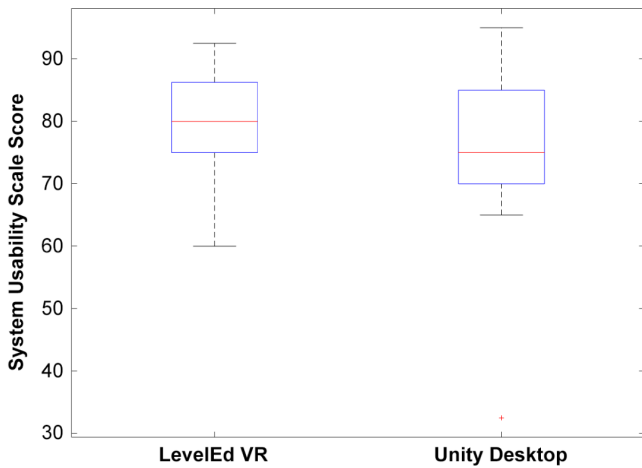


Fig. 2. Results from the System Usability Scale questionnaire. Negative outlier marked with a cross and median with horizontal line [25].

Before each task participants completed a Simulator Sickness Questionnaire (SSQ) [24]. 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) [25]. 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. 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.

B. Participants

Participants all had prior game development and Unity experience (all but one had between 1.5-2.5 years) and were selected to ensure that they could effectively evaluate both systems. A total of 16 participants took part in the study. Participants were split with 75% identifying as male and 25% identifying as female. 69% of participants stated they had little or no experience with virtual reality with the rest stating intermediate experience.

V. EXPERIMENT RESULTS

A. 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 [25]. The results (summarised in Fig. 2) 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, Kortum and Miller, the mean score of 79 results in an adjective rating of Good for the system [26]. This is a positive outcome and bodes well for future development.

When compared to the existing Unity Desktop workflow, LevelEd VR compares favourably. Unity Desktop scored a mean of 75 with standard deviation of 15. The minimum score was 33, and the maximum score was 95. Unity Desktop

Comparative Questionnaire Results

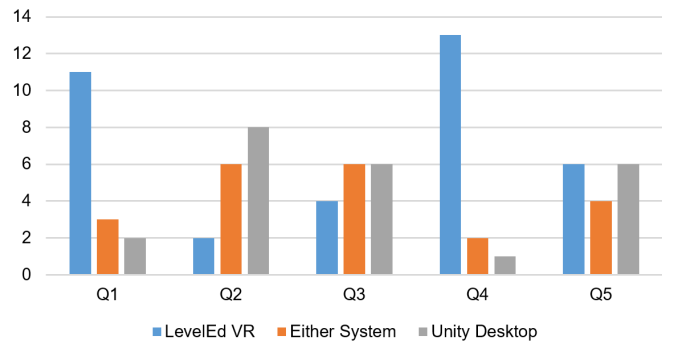


Fig. 3. Comparative questionnaire results based on questions in Table II.

is also considered an acceptable system (as the mean exceeds the 68-score threshold) and has an adjective rating of Good.

A t-test does not show any statistical significance between the two sets of results (p -value = 0.3533). The mean and boxplots (Fig. 2) show a slightly better usability score for LevelEd VR when compared to the Unity Desktop.

B. Comparison of Systems and Workflow

A comparative questionnaire was also used to ascertain participants' preferences or experiences of each system (see Table. II) with the results presented in Fig. 3. The results show that 68.75% of participants preferred to use LevelEd VR when building game levels for a virtual reality 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 virtual reality 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 virtual reality game levels, with 37.50% preferring both LevelEd VR and Unity Desktop and 25.00% finding either system to be efficient.

TABLE II. COMPARATIVE QUESTIONNAIRE QUESTIONS

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

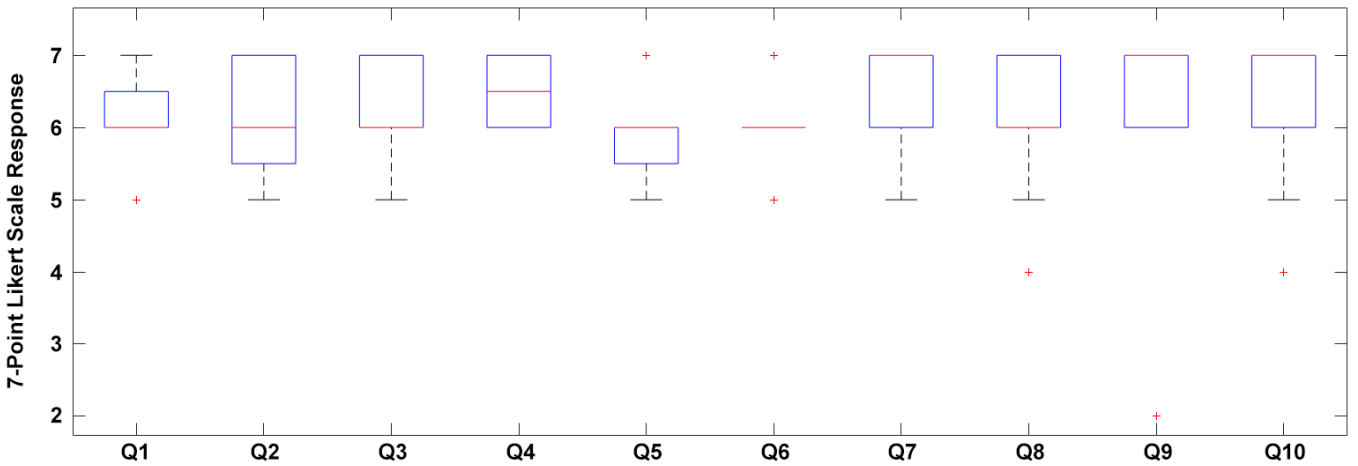


Fig. 4. Responses to the general feedback questionnaire (Table III) for LevelEd VR. A 7-point Likert scale response was used for each question. Outliers marked with a cross and median marked with horizontal line.

C. LevelEd VR Feedback

To better understand participants' experiences with LevelEd VR a feedback questionnaire (Table III) was given to participants after completing the LevelEd VR task with the results shown in Fig. 4. A 7-point Likert scale was used ranging from strongly disagree (1) to strongly agree (7).

Responses to Q1 and Q2 show that participants were confident in using the locomotion system and were able to effectively navigate the level in order 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.

D. 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. [24] 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 IV. Tables V-VIII include the pre and post results for each system.

Both systems/workflows demonstrated some levels of simulator sickness. In all categories it is clear that 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.

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

TABLE III. GENERAL QUESTIONNAIRE QUESTIONS FOR LEVELED VR

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

TABLE IV. REFERENCE SSQ-EVALUATION FINDINGS

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

TABLE V. LEVELEd VR PRE SSQ-EVALUATION FINDINGS

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

TABLE VI. LEVELEd VR POST SSQ-EVALUATION FINDINGS

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

TABLE VII. UNITY DESKTOP PRE SSQ-EVALUATION FINDINGS

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

TABLE VIII. UNITY DESKTOP POST SSQ-EVALUATION FINDINGS

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

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.

E. Video Coding Analysis

The video recordings of each task for each participant was analysed to better understand any differences between the two workflows. One key difference revealed is the amount of times that 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. We also looked at 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. We found that participants took on average 31 seconds to transition.

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

VI. DISCUSSION

A. Acceptance and Preference

It is clear from the SUS, general feedback and comparative results that the participants accepted the LevelEd VR system and VR workflow for level design for VR games. The SUS results were not statistically significant but nevertheless the trend observed was a preference for using LevelEd VR when making levels for VR games. This was reinforced by results from the comparative questionnaire (see Fig. 3 – Q1 with 69.75% versus 12.50%). This is a very 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.

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 Fig. 3 – Q4 and Fig. 4 – 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 why participants preferred to use Unity Desktop over LevelEd VR when working on non-VR games (see Fig. 3 – 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.

We believed another of the benefits to having the full workflow in VR is that testing would 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. We believe 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 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 the 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 our 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.

B. Minimising Simulator Sickness

It is clear that 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.

We see this 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 [27] 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 (Fig. 4 – Q1 and Q2) and the lack of any comments regarding simulator sickness in the written comments for locomotion.

We believe the simulator sickness was 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 a reduction in simulator sickness when the peripheral vision is limited [28][29]. 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 we would recommend that when building systems similar to LevelEd VR that users are given different options for locomotion as no one size fits all situations.

C. 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 (Fig. 4 – 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 (Fig. 4 – Q8). This suggests that the spatial scripting system is worth pursuing further.

However, there were some issues reported by users such as the ‘in’ and ‘out’ tags (Fig. 1e) being “*a little small*” and they had “*difficulty selecting them at times*”. We believe increasing the size and adding a highlight to the tag they are trying to attach to will 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.

D. 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 (Fig. 4 – 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 the way we opted for a gesture-based system 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.

VII. CONCLUSION

In this paper we have presented a generic runtime VR system and workflow that supports the level design and level creation process for VR games and the initial version presented here focuses on supporting the blackout 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 our system. 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 our 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 our aims, there are areas for improvement, such as with the scripting system, natural and gesture based interactions and work must continue on reducing the effects of simulator sickness further to ensure the system is accessible to all.

There is further work to be completed in investigating additional benefits of working in VR with a system like LevelEd VR for the level design process. It's worth noting that there are limitations with the study in regard to the 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. Future testing should also look at testing with inexperienced users, such as potential players. We intend to analyse the videos further to see if there are any additional differences between the workflows. We also intend to compare the quality of levels produced by each system to see if there are any design differences between levels built in the different workflows.

Finally, future work should look at comparing our runtime editor to that of the Unity Editor XR and Unreal Engine VR Mode. We will also evaluate how both LevelEd VR and our previous system LevelEd AR [30] can work together to allow for the development of spatially accurate spaces entirely on devices without the need for a desktop/laptop or traditional game engine/editor programme.

REFERENCES

- [1] R. Kremers, *Level Design: Concept, Theory and Practice*, Boca Raton, FL: CRC Press, 2009
- [2] C. W. Totten, *An Architectural Approach to Level Design* (1st ed.), Boca Raton, FL: CRC Press, 2014
- [3] A. Wheldon, Interview by S. De Jong [Written Interview] in *The Hows and Whys of Level Design* (2nd ed.), Raleigh, NC: Lulu Press, 2008
- [4] C. Pruet Interview by C. W. Totten [Written Interview] in *An Architectural Approach to Level Design* (1st ed.), Boca Raton, FL: CRC Press, 2014
- [5] G. Grimsby Interview by C. W. Totten [Written Interview] in *An Architectural Approach to Level Design* (1st ed.), Boca Raton, FL: CRC Press, 2014
- [6] H. Silverman, *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, 2017.
- [7] J. Jerald, *The VR book: Human-centered design for virtual reality*, San Rafael, CA: Morgan & Claypool, 2015
- [8] Business Wire, "The 7th Annual Bloxy Awards Brings 4 Million Concurrent Players Together on Roblox", Retrieved March 26, 2020 from <https://www.businesswire.com/news/home/20200323005741/en/7th-Annual-Bloxy-Awards-Brings-4-Million>
- [9] J. D. O. De Leon, R. P. Tavas, R. A. Aranzanso, and R. O. Atienza, "Genesys: A Virtual Reality scene builder," 2016 IEEE Region 10 Conference (TENCON), Nov. 2016.
- [10] C. Barot, K. Carpentier, M. Collet, A. Cuella-Martin, V. Lanquepin, M. Muller, E. Pasquier, L. Picavet, A. Van Ceulen, and K. Wagrez, "The Wonderland Builder: Using storytelling to guide dream-like interaction," 2013 IEEE Symposium on 3D User Interfaces (3DUI), Mar. 2013.
- [11] J. Wang and R. Lindeman, "Coordinated hybrid virtual environments: Seamless interaction contexts for effective virtual reality," *Computers & Graphics*, vol. 48, pp. 71–83, May 2015.
- [12] J. F. M. Moreira, *Mixed interactive/procedural content creation in immersive VR environments*, in Doctorate Thesis, 2018, Retrieved March 01, 2020 from <https://repositorio-aberto.up.pt/handle/10216/114109>
- [13] S. Ichikawa, K. Takashima, A. Tang, and Y. Kitamura, "VR safari park," *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology - VRST '18*, 2018.
- [14] P. Ekströmer, R. Wever, P. Andersson, and J. Jönsson, "Shedding Light on Game Engines and Virtual Reality for Design Ideation," *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, pp. 2003–2010, Jul. 2019.
- [15] J. Jerald, P. Mlyniec, A. Yoganandan, A. Rubin, D. Paullus, and S. Solotko, "MakeVR: A 3D world-building interface," 2013 IEEE Symposium on 3D User Interfaces (3DUI), Mar. 2013.
- [16] B. Jackson and D. F. Keefe, "Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 4, pp. 1442–1451, Apr. 2016.
- [17] D. Mendes, D. Medeiros, M. Sousa, R. Ferreira, A. Raposo, A. Ferreira, and J. Jorge, "Mid-air modeling with Boolean operations in VR," 2017 IEEE Symposium on 3D User Interfaces (3DUI), 2017.
- [18] P. Oswald, J. Tost, and R. Wettach, "i.Ge," *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14*, 2015.
- [19] J. Park, B. Son, J. Seo, and C. Park, "Development of Authoring Tool for Extended Interaction in Mixed Reality Environments," *Communications in Computer and Information Science*, pp. 304–310, 2018.
- [20] G. Ng, J. G. Shin, A. Plopski, C. Sandor, and D. Saakes, "Situating Game Level Editing in Augmented Reality," *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '18*, 2018.
- [21] R. Dillet, "Unity CEO says half of all games are built on Unity", Sept. 2018, Retrieved March 01, 2020 from <https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/>
- [22] D. Checa and A. Bustillo, "A review of immersive virtual reality serious games to enhance learning and training," *Multimedia Tools and Applications*, vol. 79, no. 9–10, pp. 5501–5527, Dec. 2019.
- [23] VRTK, "VRTK - Virtual Reality Toolkit", Retrieved March 01, 2020 from <https://vrtktoolkit.readme.io/>
- [24] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *Int. J. Aviation Psychology*, vol. 3, no. 3, pp. 203–220, 1993.
- [25] J. Brooke, "SUS - A quick and dirty usability scale," in *Usability evaluation in industry*, Edited by: P. W. Jordan, B. Thomas, B. A. Weerdmeester and A. L. McClelland, London: Taylor & Francis, 1996, pp. 189-194.
- [26] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," in *Journal of Usability Studies*, vol. 4, no. 3, pp. 113-123, 2009.
- [27] F. Bonato, A. Bubka, S. Palmisano, D. Phillip, and G. Moreno, "Vection Change Exacerbates Simulator Sickness in Virtual Environments," *Presence: Teleoperators and Virtual Environments*, vol. 17, no. 3, pp. 283–292, Jun. 2008.
- [28] H. Buhler, S. Misztal, and J. Schild, "Reducing VR Sickness through Peripheral Visual Effects", *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2018. doi:10.1109/VR.2018.8446346
- [29] A. S. Fernandes and S. K. Feiner, "Combating VR sickness through subtle dynamic field-of-view modification," 2016 IEEE Symposium on 3D User Interfaces (3DUI), Mar. 2016. doi:10.1109/3dui.2016.7460053
- [30] L. Beever, S. Pop and N. W. John, "Assisting Serious Games Level Design with an Augmented Reality Application and Workflow", *Computer Graphics and Visual Computing*, 2019. doi: 10.2312/cgvc.20191253