

Exploring Mixed Reality for Level Design Workflows

Lee Beever



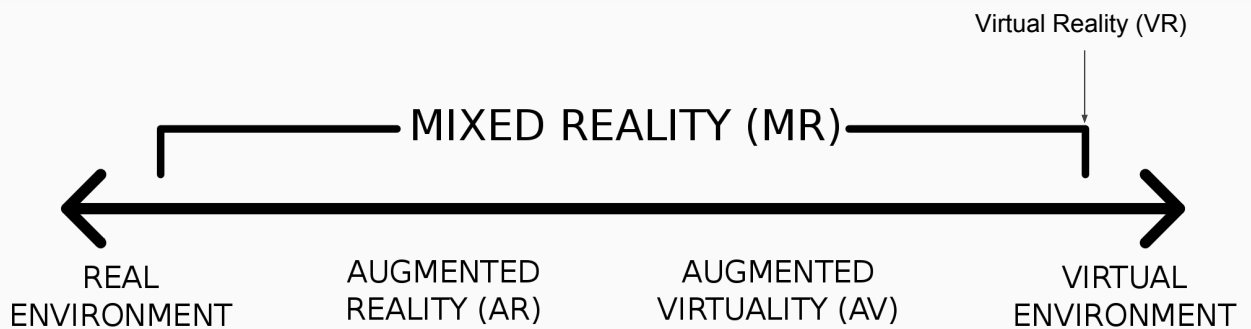
Topics

- Mixed Reality
- Level Design
- Research Area
- LevelEd AR Study

Exploring **Mixed Reality** for Level Design Workflows

What is mixed reality?

Mixed Reality



[Milgram & Kishino, 1994]

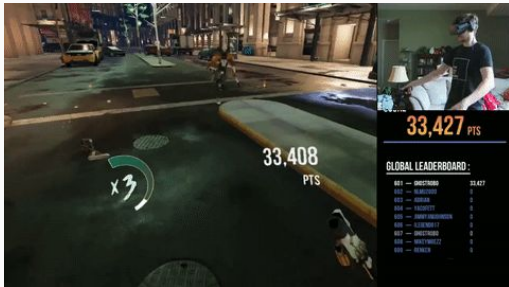
What is mixed reality?

Milgram & Kishino developed the virtuality continuum in 1994.

1. **Real Environment** is the real world with no technology interference.
2. **Augmented Reality** is where digital artifacts are overlaid or appear in the real world.
3. **Augmented Virtuality** is where real world objects are brought into a virtual environment. Quite often this is seen with virtual control centres that are fed real time data and viewed in virtual reality.
4. **Virtual Environments** are artificially created without capturing any content from the real world.

Virtual Reality sits between AV and VE. My research takes me from AR up to VE in relation to level design workflows.

VR



Virtual reality headsets like the Oculus Rift and HTC Vive allow for full 6 degrees of freedom (DOF) tracking of the head and now also hand controllers.

The user sees a virtual world in the headset and can move around and interact with the environment. Environments are completely artificial.

Motion controllers allow users to interact more naturally with scenes and objects.

AR



Augmented Reality devices on the other hand allow you to view the real world and then overlay digital images on top.

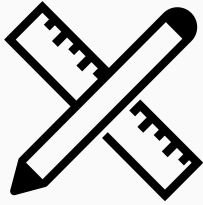
HoloLens is a standalone headset that allows you to view 'holograms' in the world and also allow these 'holograms' to interact with the world. Hand gestures are used for interaction.

Smartphone AR uses the camera to feed the real world onto the screen where digital images are inserted into the scene. These are anchored in place so you can move around them.

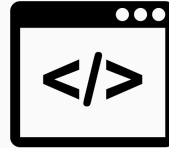
Exploring Mixed Reality for **Level Design** Workflows

What is level design?

Game Development Disciplines



Design



Programming



Art



Production

To understand the level design role you need to first understand the core game development disciplines.

1. **Design** - Generate the initial ideas for the game and then must define the specific game mechanics for implementation and to continue to refine throughout the project.
2. **Art** - Artists are the ones who work on the visuals of the game. They create the user interface, the characters, the 3d environments, vehicles, VFX, etc.
3. **Programming** - Implement the game mechanics from the design team, build/customise the game engine, work on rendering technology, implement networking, etc.
4. **Production** - A team of producers/managers who plan out what needs to be done, when it needs to be completed and manages the team throughout development. AAA games tend to have a 2-3 year schedule. Their job is to manage the team to complete on schedule. Not an easy job due to the fluidity of game development.

“Level design is applied game design”

- Kremers, 2009

Simply put, level design is applied game design.

The game designers create the mechanics:

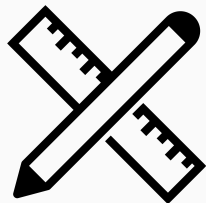
- They decide how the player character can move, e.g. how fast they move, how high/far they can jump, whether they can climb.
- The designers decide on what abilities the character has, e.g. the spells they can use, the actions they use on the world, etc.
- They'll also decide what challenges will be in the game. These could consist of environmental challenges, puzzles or enemies.

Level designers interpret these game rules, and translate them into a level that best facilitates play.



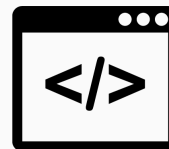
Most levels consist of a player, a goal and then a route between them filled with challenges/obstacles. It is the level designers job to plan this out and ensure it is engaging for the player.

Where does level design fit sit?



- Designing the layout/geometry
- Planning wayfinding
- Planning enemy locations
- Puzzles/challenges

Level Design



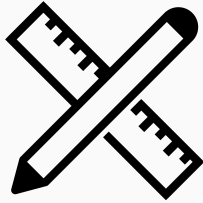
- Implementing level specific functionality.
- Programming AI
- Programming events and triggers



- How will the level look?
- Composition of objects
- Architectural rules
- Colour/lighting theory

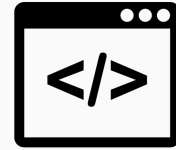
Level design generally comes under the design department. However, it requires skills from all 3 disciplines and therefore I see it as the intersection of the 3 core disciplines.

Where does level design sit?



- **Designing the layout/geometry**
- **Planning wayfinding**
- Planning enemy locations
- Puzzles/challenges

Level Design



- **Implementing level specific functionality.**
- Programming AI
- Programming events and triggers



- How will the level look?
- **Composition of objects**
- **Architectural rules**
- **Colour/lighting theory**

These are core roles of a level designer. But how does that manifest itself in the game.

What does the level designer produce that the players get to experience?

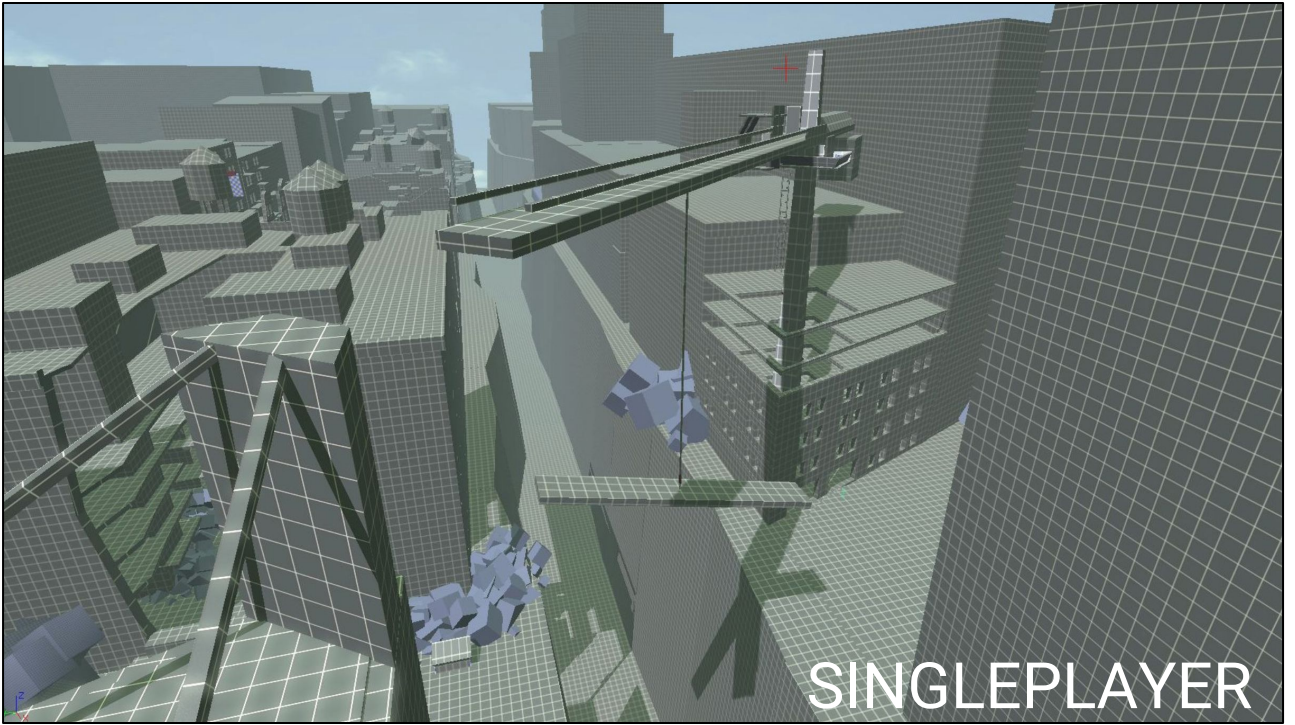


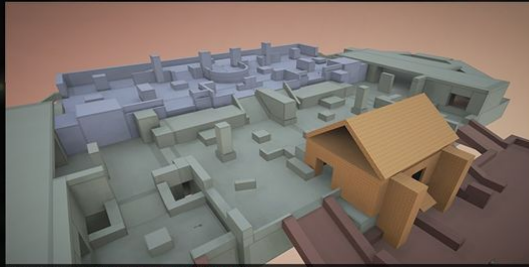
The result is a game level. A place for the players to utilise the game mechanics devised by the game designers.

The level designer must develop the geometry, the layout, the routes the player can take and then test the player on their ability to master the game mechanics.

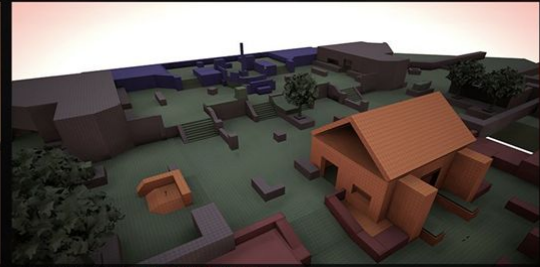
But we don't just go from an idea in our heads to this finished version here. The art in the scene takes weeks if not months to develop by the 3D artists.

Instead, we develop blockouts. The level designers spends most of their time in the blockout stage, iterating and refining the play space and gameplay.





RECLAIMED / BLOCKOUT STAGE 1



RECLAIMED / BLOCKOUT STAGE 2



RECLAIMED / BLOCKOUT STAGE 3



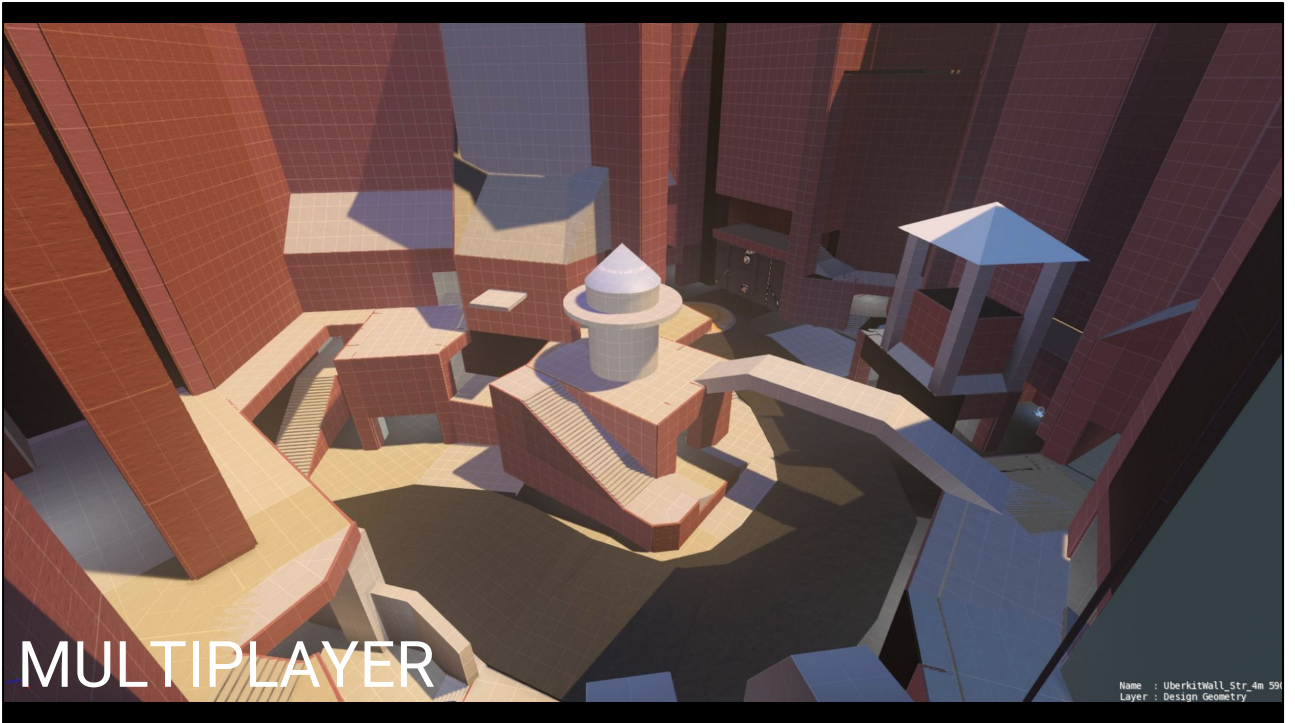
RECLAIMED / FINAL GAME

Gears of War 4 blockout iterations to the final version.



Sunset Overdrive blackout to final form.





Junkertown - Overwatch



MULTIPLAYER

Name : JMKR_Town_Facade_02_08

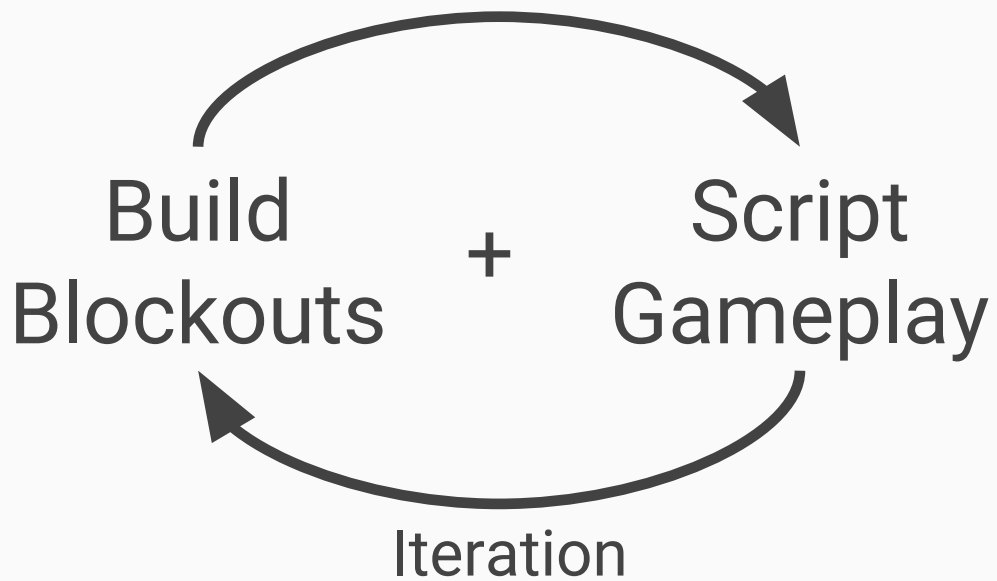
26
Left

SINGLEPLAYER

Uncharted 4 chase scene showing blackout, stage 2 block out and then final form.

<https://twitter.com/kurtmargenau/status/915653024214691841>

Basic Needs of a Level Design Workflow



The core needs for a level designer are:

- The ability to construct blockouts of gameplay spaces.
 - The ability to script functionality
- And to do this through rapid iteration.

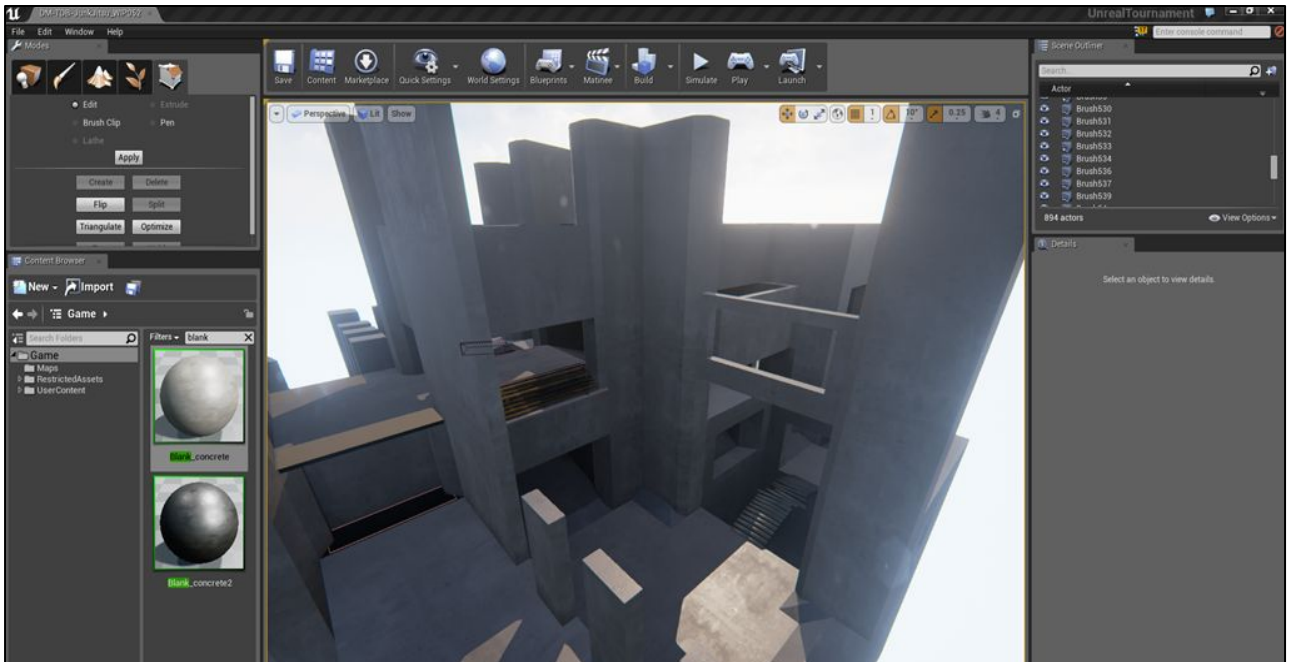
What's the problem?

What's the problem?

All the games in the previous examples are games played on 2D displays that were developed on 2D displays.

VR Games aren't played on 2D displays but in stereoscopic 3D.

You wouldn't try and create a colour film on a black and white screen... so why do the same for VR.



VR Games built with tools designed for games displayed on 2D displays

Traditionally we've built 3D games that will be displayed on 2D screens. Built with tools for use on 2D screens. This workflow works well. However, VR games are viewed in stereoscopic 3D often in first person where the player can control the camera. We're now stuck building 3D stereoscopic games on 2D screens.

Current tools have the ability to create blockouts for VR games, but the workflow isn't efficient:

1. You build the blockout on a 2D display
2. You're happy with it but now need to check the result in 3D.
3. You scramble around for your headset
4. Put it on and play the game.
5. You then have to find the right place in the level.
6. You view the blockout and realise it needs changing
7. Take headset off
8. Jump back into the editor on your 2D display
9. Make changes
10. Rinse repeat

It's not very rapid and slows down iteration.

Research Question /Vision

We envision that mixed reality workflows can improve level design workflows for VR games.

- Can it improve iteration time?
- Does it deliver additional benefits? What are they?
- Does working with our hands improve usability/accessibility?

Improving the iteration time means level designers can spend more time iterating, experimenting and refining levels.

There also other potential improvements from a mixed reality workflow. Such as:

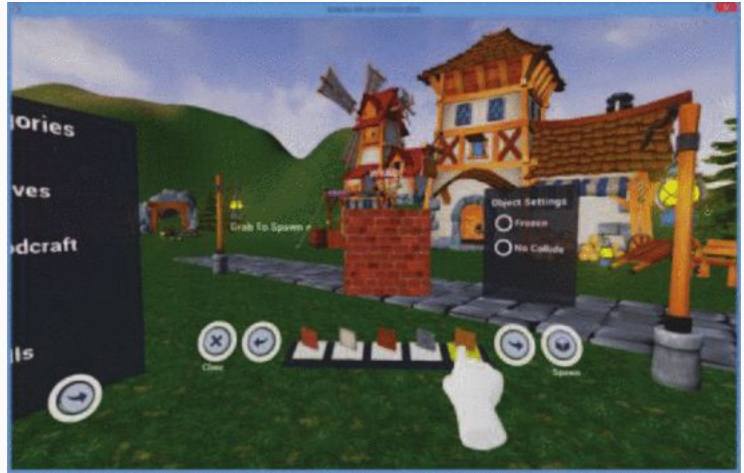
1. Offering more natural interaction with content. Tilt brush is a good example of this.
2. Stereoscopic 3D aiding composition.
3. 6 DOF tracking improving ability to quickly change pose and perspective whilst developing.
4. Building directly in VR helps with scaling the levels.

Current Research

Genesys: A Virtual Reality Scene Builder

- Focus is on building scenes with objects.
- Hand tracking using Leap Motion.
- Inspiration for my direction.

[De Leon, Tavas, Aranzano, & Atienza, 2016]

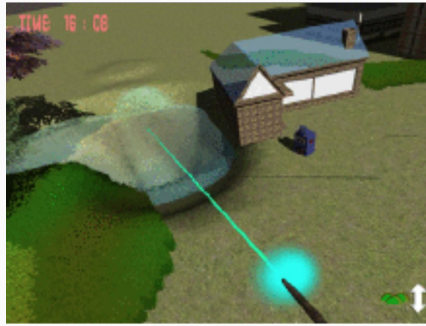


Current Research

DIY World Builder

- Focus is on building scenes with objects.
- Wand tracked and wrist mounted tablet for object selection.
- Locomotion through flying and also omni-directional treadmill.

[Wang, Leach, & Lindeman, 2013]



Current Industry

UE4 VR Mode

- Focus on replicating current editor tools.
- Object placement and manipulation.
- Some BSP/blockout tools available.

[Epic Games, 2018]

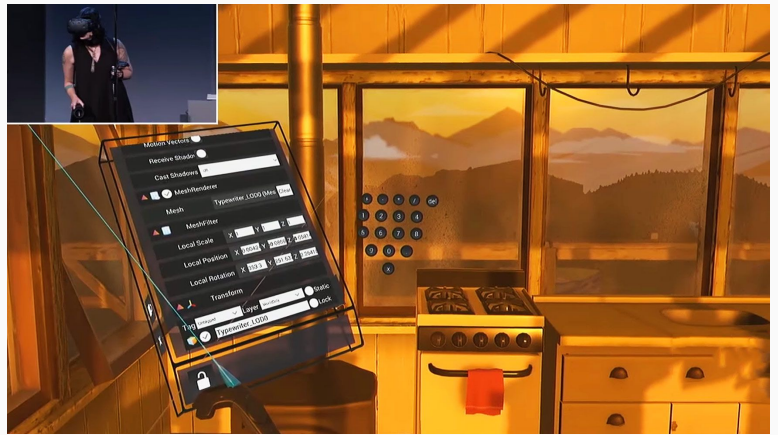


Current Industry

Unity Editor VR

- Focus on replicating current editor tools. Modal windows.
- Object placement and manipulation.
- No blackout tools for rapid iteration.

[Unity Technologies, 2018]



My Direction



Current Research Common Features:

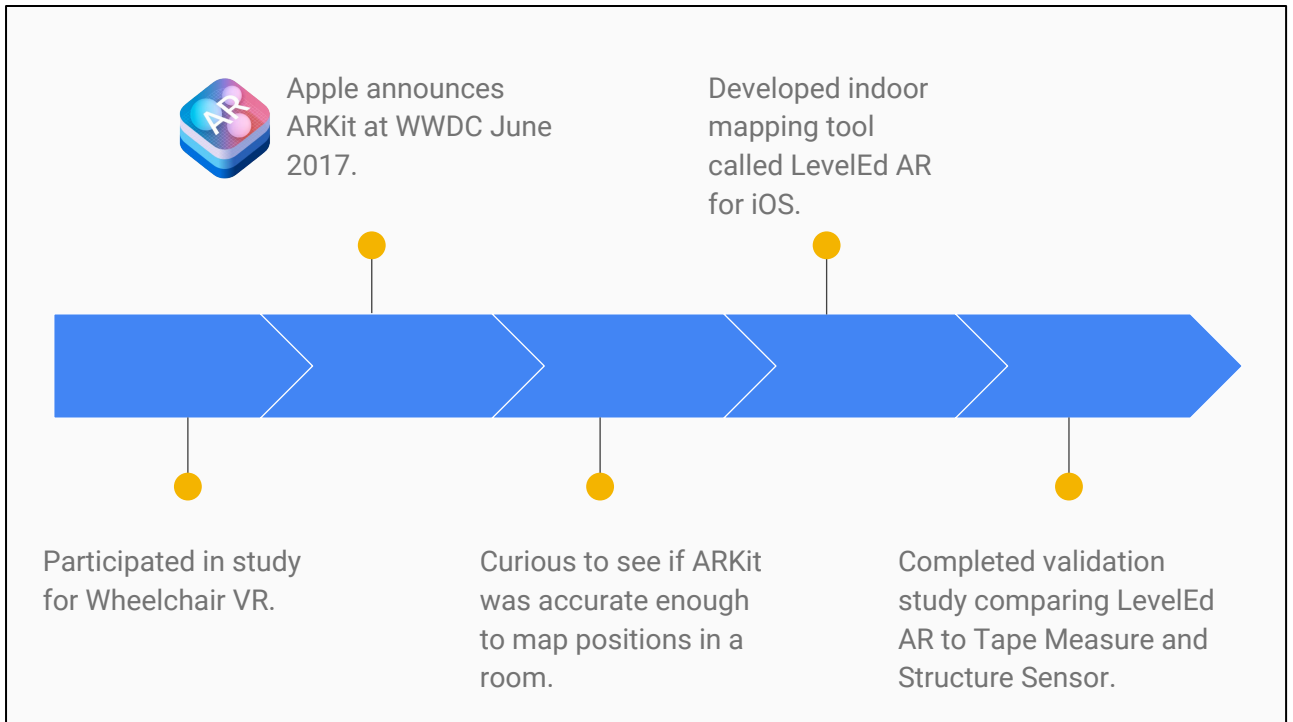
- Focus on object placement
- Recreating editor tools in VR (1 for 1)
- Jack of all trades
- No scripting functionality
- Runs in the editor and not at runtime
- Not portable/can't be shipped

My Direction:

- Approach the problem from a level designer perspective
- Rapid blockout iteration is key
- Ability to edit/play/edit as quickly as possible
- Simple scripting framework
- Build a framework for developers to implement into their games
- **Explore beyond just VR by also looking at AR**

By approaching it from a level designer perspective I mean a focus on creating effective spaces that best showcase the game mechanics for a player trying to get to a goal. Gameplay > Visuals.

LevelEd AR Study



It all started when I participated in the Wheelchair VR study by Tom. Wondered if recreating users real world spaces in VR would help them with learning to use the wheelchair and potentially show issues with their current spaces.

Then Apple announced ARKit at WWDC 2017 to ship with iOS 11. ARKit is an augmented reality framework for iOS developers. It is markerless and works by using computer vision to track changes in details between frames and the accelerometer to detect movement. Called Visual Inertial Odometry (VIO).

I started to see examples of apps that could measure things using ARKit. Wondered if it would be possible to map key points in a room to mark out positions and heights of not only walls but the objects inside it. This could be useful for serious games where having a to scale space is important. See Wheelchair VR.

Went ahead and developed an indoor mapping tool, that uses ARKit to track the position of the user, the user places markers in key points and then specifies the height or walls and objects. Took a lot of trial and error to get it to work. Also a new way of thinking about interactions. No longer just 2D images/text on a screen but objects that float in augmented reality.

Created a study to compare the accuracy and efficiency of using LevelEd AR to a tape measure and Structure Sensor.

Indoor Mapping Techniques



- Standard tape measure or laser measure, writing down measurements as you go along. Time consuming.
- Structure sensor that uses infrared to create a depth map of a space. Quick but not always accurate.
- HoloLens quick to create spatial maps but expensive and not intuitive to use.
- Laser scanning. Highly accurate but expensive. Pictured is £18k.

LevelED AR

An augmented reality spatial mapping application

Lee Beever

I developed the LevelEd AR application as a test app to see if indoor mapping was possible with ARKit.

https://youtu.be/IOd8JQjQs_U

The study structure

- 3 x mapping techniques.
 - LevelEd AR Application
 - Structure Sensor
 - Tape Measure
- 4 x mapping tasks.
- 18 participants (staff and students).
- Randomised crossover for both mapping techniques and tasks.
- Key measurements of each task captured. Enough to rebuild in Unity.
- Time to complete task recorded.
- Participant tape measure taken as ground truth

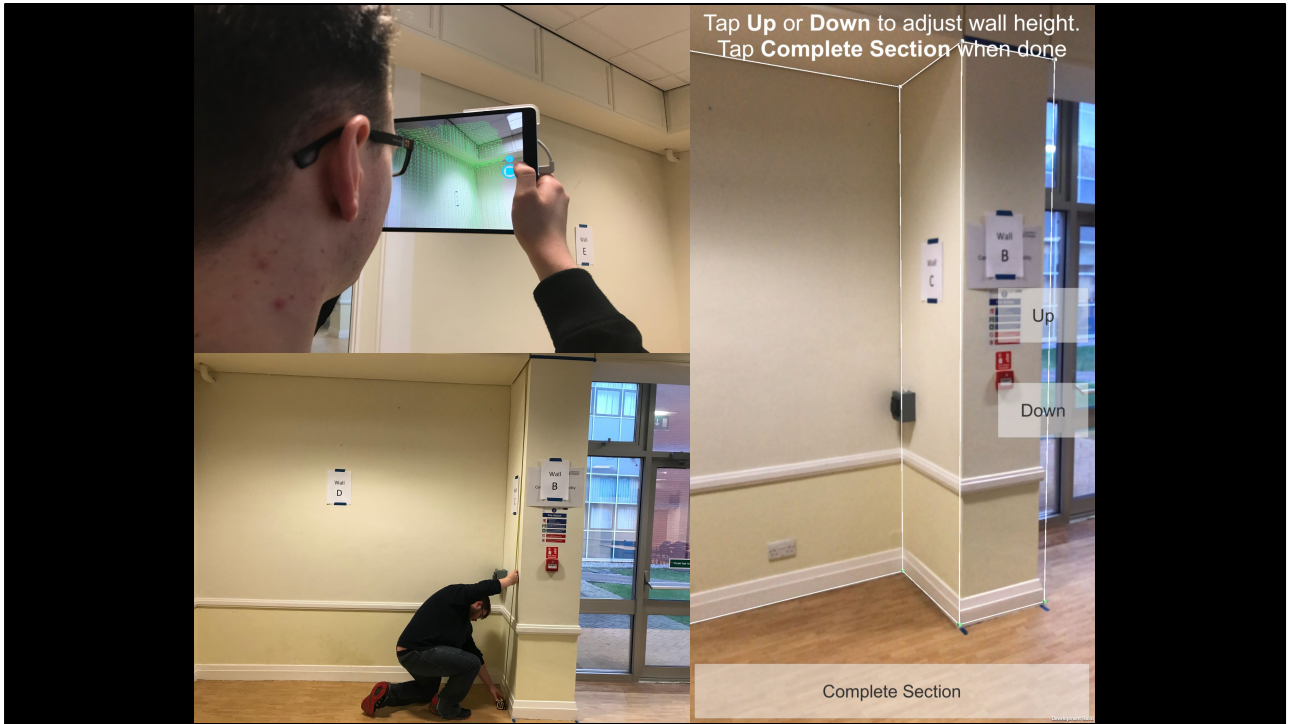


Task 1 - Area of single wall

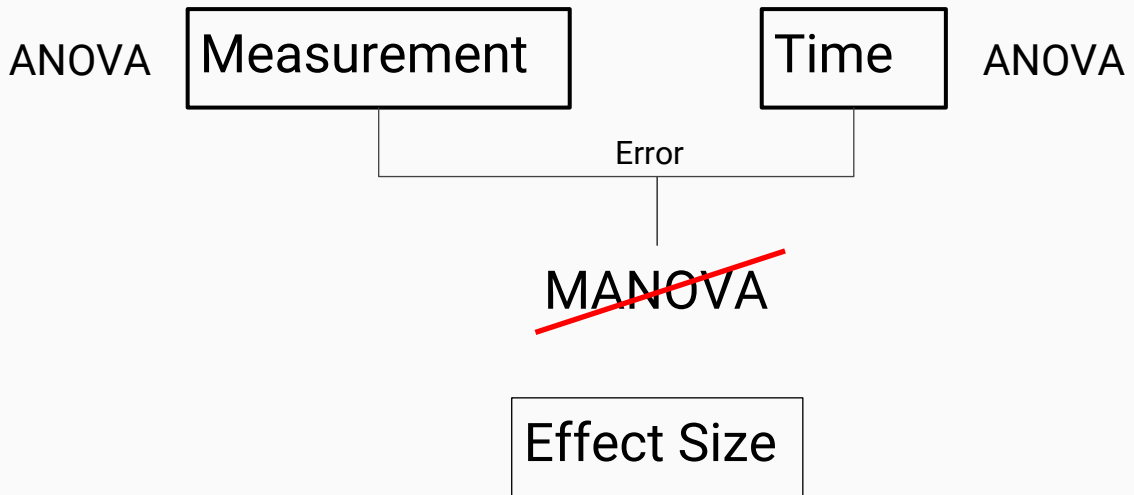
Task 2 - Volume of box with no restricted movement

Task 3 - Volume of box with restricted movement

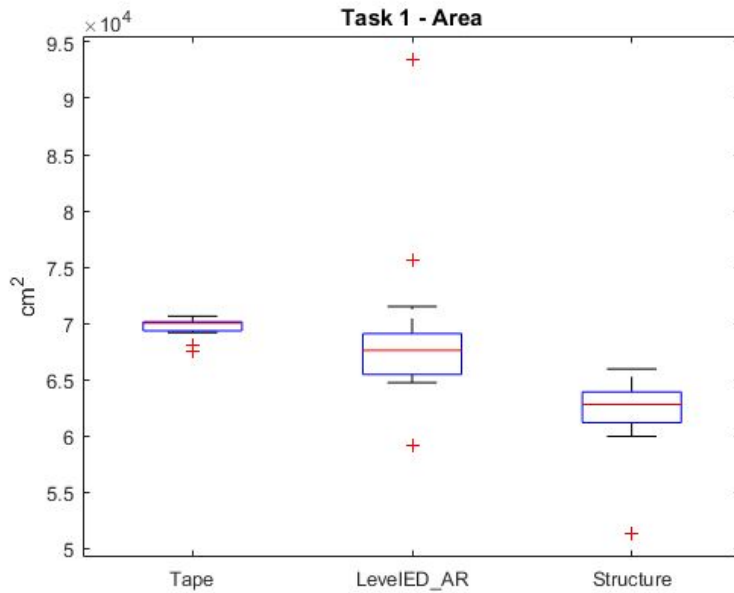
Task 4 - 4 walls and volume of box and position of box. Meant to simulate a room.



Structure Sensor
Tape Measure
LevelEd AR app



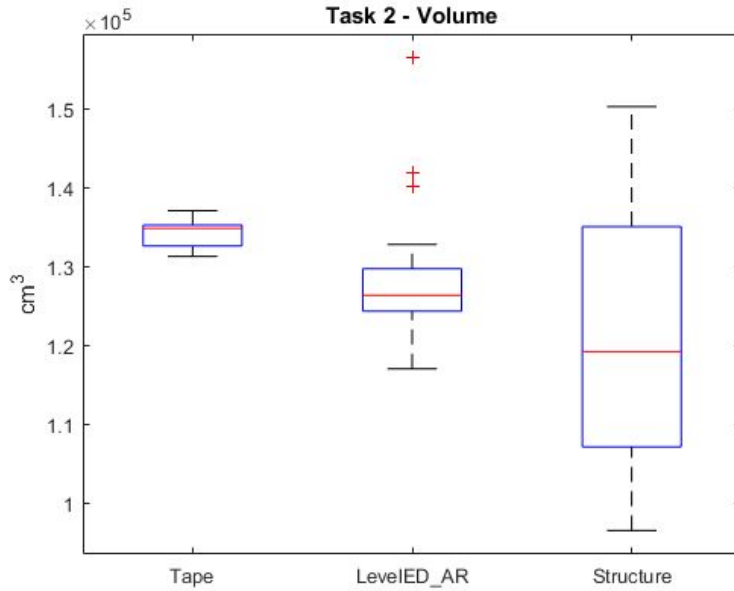
Traditionally we would need to analyse using MANOVA due to having 2 dependent variables. However, we calculated the effect size in SPSS which demonstrated that the effect was strong enough that we could separate the measurement and time variables. This allowed us to calculate ANOVA tests for each variable separately. The ANOVA tests showed that the results were statistically significant.



Task 1 - Area

Very small variability on the tape measure. Common throughout the tests. Therefore it was accurate and consistent enough to be taken as ground truth. We also created our own ground truth measurements using a tape measure and compared it to participant tape measures and found an error rate of only 99.6%.

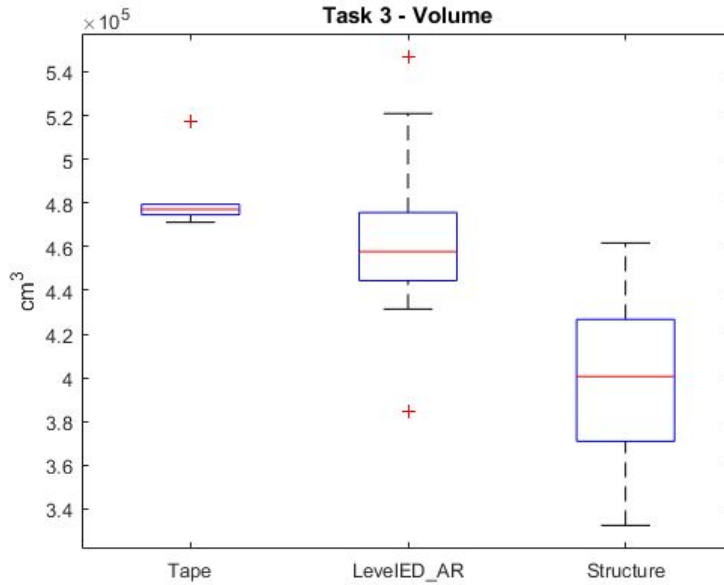
Reasonable variability on the AR application but median is much closer to the ground truth than structure sensor.



Task 2 - Volume

Again, very small variability on results for tape measure.

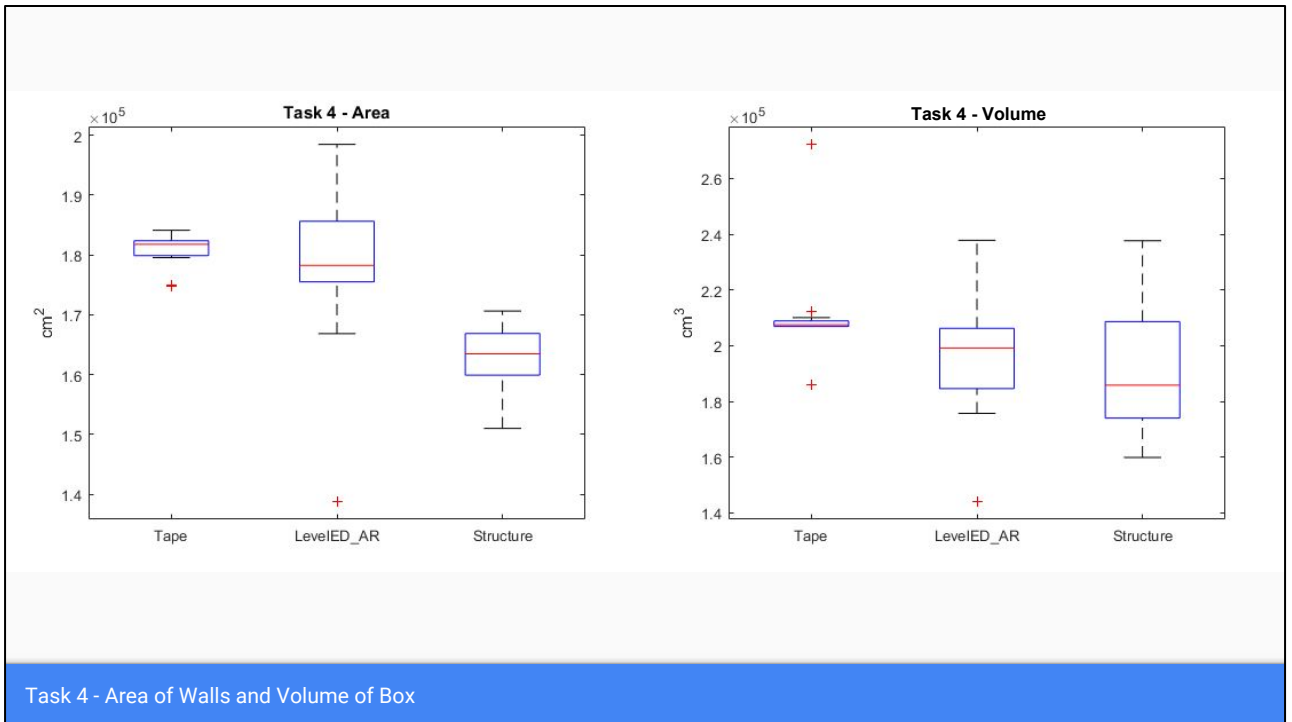
Slightly more variability on the AR application but median is much much closer than the Structure sensor.



Task 3 - Volume

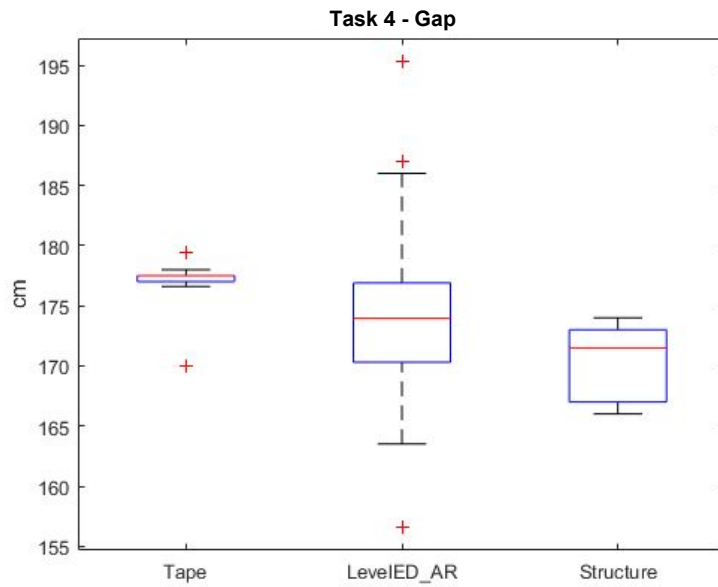
Very close results for tape.

Reasonable variability for AR app, but again the median is much closer to tape than Structure. Structure is under reporting distances quite a bit.



AR app is closer on average. However, there is a bigger spread than previous tests. This was hypothesized before the study and was the reason this task was included. The spread is likely due to drift as the map created by ARKit loses sync with the real world.

Still closer than Structure on median.



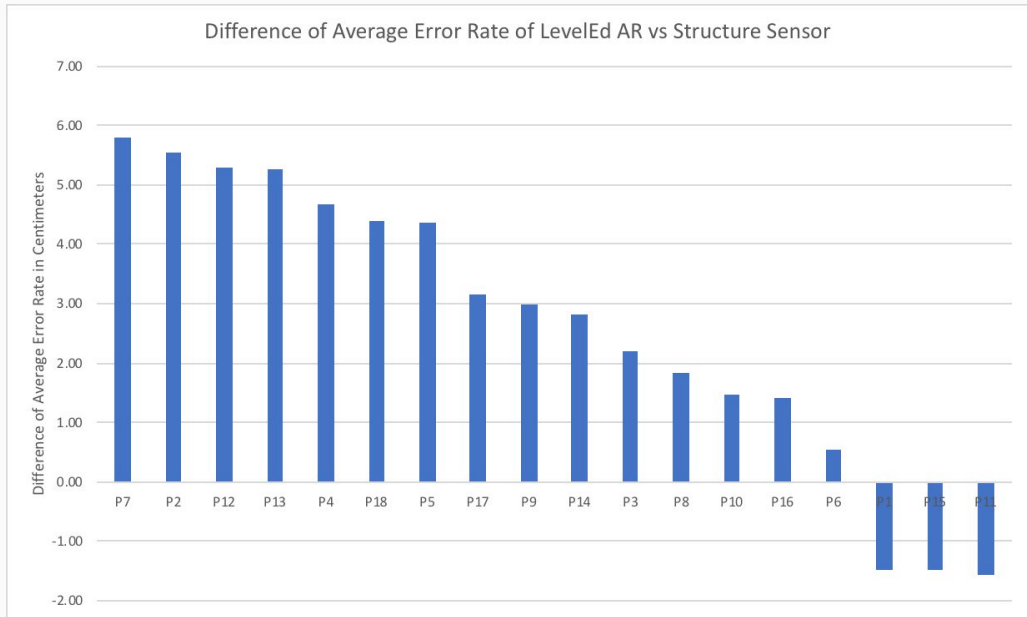
Task 4 - Gap between box and wall. Measuring the position of the box in the room.

Big spread of values. Again, this is due to the box being the last part of the task measure by the app and therefore more chance of previous points drifting over time. Error propagation.



Error Rates

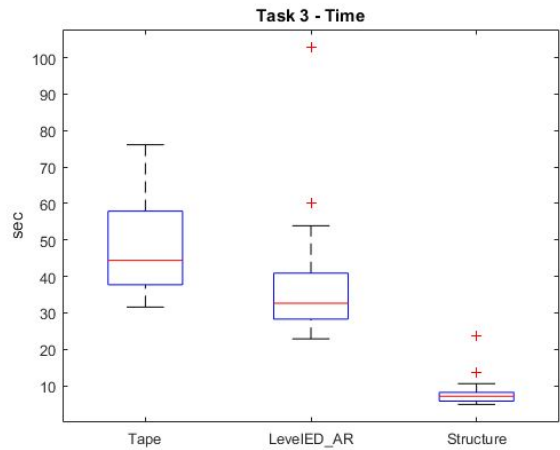
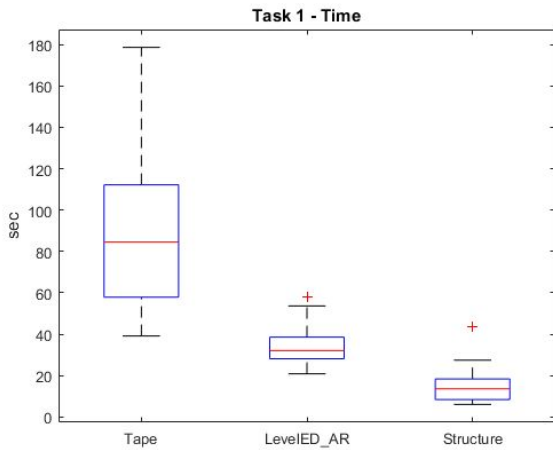
We calculated the error rates by comparing both the LevelEd AR and Structure measurements against the tape measure for each participant. This was done for every measurement taken. The error rate was captured regardless of whether it was smaller or larger. These error rates were then averaged out for each participant and shown above.



Error Rates

We then found the difference between the error rates of structure versus LevelEd AR.

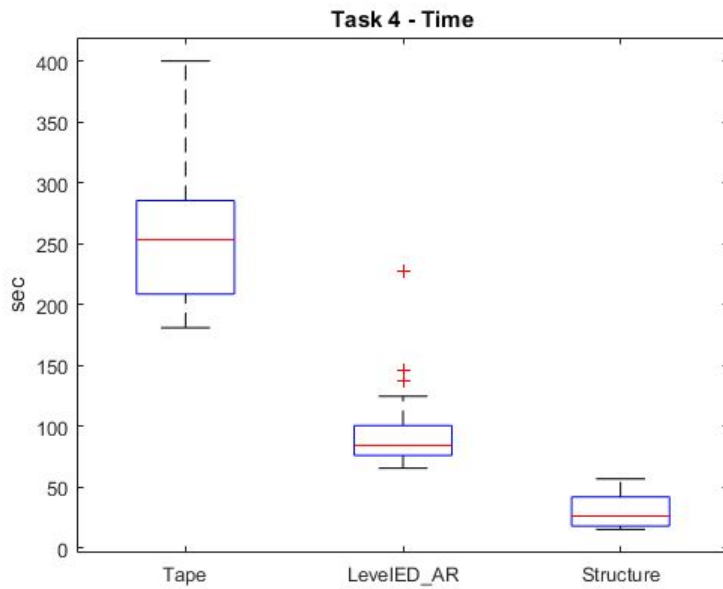
Positive values show Structure as having a higher error rate than LevelEd AR. Negative values show participants who had more accurate results from Structure sensor.



Task 1 and Task 3 - Completion time.

Task 1: AR app much much quicker than tape. But slower than Structure sensor. Tape measure had large variability due to many different techniques used by participants.

Task 3: Volumes show a closer time of completion between tape measure and AR app due to more complex interactions required by AR app. Structure sensor way ahead.

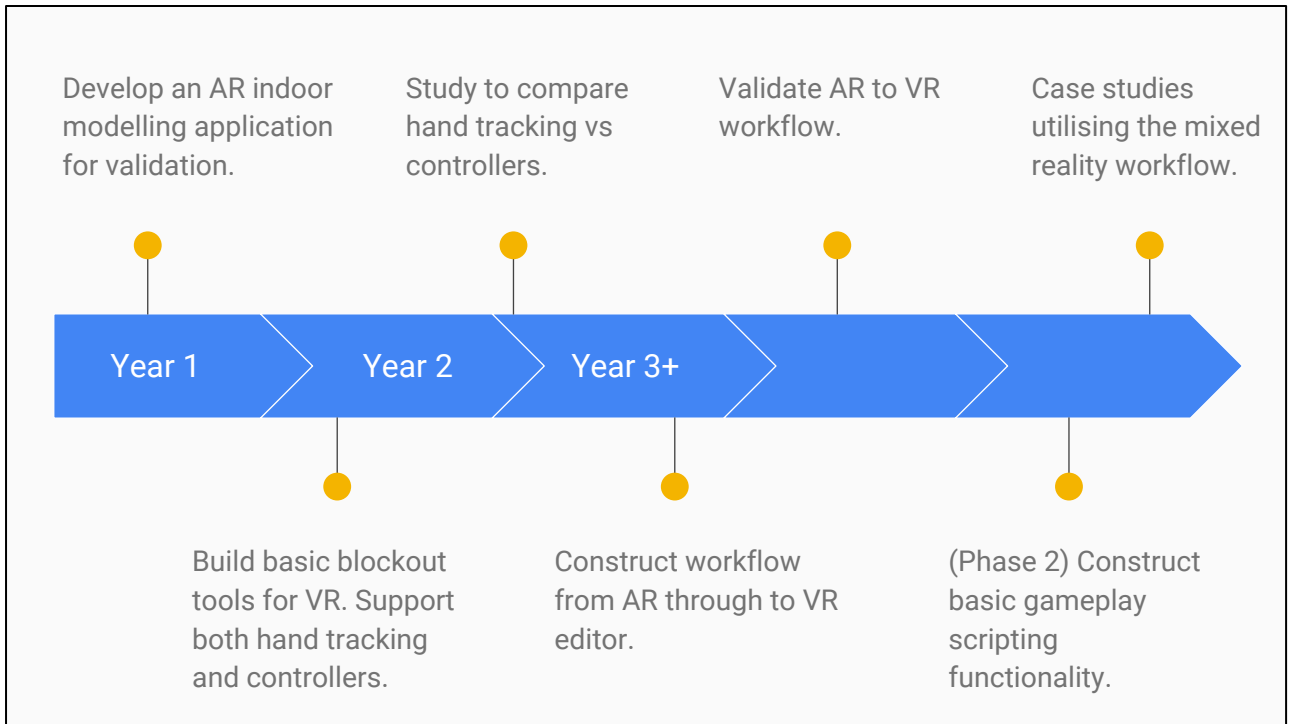


Task 4 - Completion Time

When completing task 4 which is a much more complex scene we can see that the AR app is much quicker than the tape measure and not that far behind structure. The larger and more complex the task the bigger the spread between the AR app and tape measure.

Conclusion

- The results are statistically significant.
- Results suggest LevelEd AR more accurate than Structure Sensor across the board.
- LevelEd AR potentially suffers from drift over longer tasks.
- LevelEd AR considerably quicker than tape measure but slower than Structure Sensor.
- Younger participants (18-35) found LevelEd AR easy to use, but those who were older struggled.
- Now working on writing up for ISMAR or EuroVR.



Potential timeline for PhD. Likely to change as research continues. Next step is to decide if work should continue with the AR app or move straight on to the VR component.

Areas of Research

Interaction

Workflow

Locomotion

How do we interact with tools:

- Do we use motion controllers or hand tracking? In real life we'd use our hands to manipulate stuff rather than chopsticks (controllers).
- Do we use a menu that appears on one hand or do we use a dashboard.
- Do we use voice control?

Locomotion. How do we move around the scene without making the user sick? This is still an unsolved problem in games:

- Do we use the teleporter method?
- Do we allow them to scale and grab/move the world?
- Can they fly?
- A lot of the current research is on moving around an environment that is complete. How do we move around an environment that is incomplete or doesn't exist yet?

Workflow.

- Does the AR to VR workflow help at all?
- Does it bring something to the table that can't be done in any other way?
- Can a VR editor improve the workflow for non-VR games as well?

Thanks for
listening

Any questions?

[@beever](#)

l.beever@chester.ac.uk

[#blocktober](#)

References

De Leon, J. D. O., Tavas, R. P., Aranzanso, R. A., & Atienza, R. O. (2016, November). Genesys: A Virtual Reality scene builder. In *Region 10 Conference (TENCON), 2016 IEEE* (pp. 3708-3711). IEEE.

Epic Games. (2018). Unreal Engine VR Mode. Retrieved from <https://docs.unrealengine.com/en-us/Engine/Editor/VR>

Kremers, R. (2009). *Level Design: Concept, Theory and Practice*. Wellesley, MA: A.K. Peters

Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12), 1321-1329.

Unity Technologies. (2018). Unity-Technologies/EditorXR. Retrieved from <https://github.com/Unity-Technologies/EditorXR>

Wang, J., Leach, O., & Lindeman, R. W. (2013, March). Diy world builder: An immersive level-editing system. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on* (pp. 195-196). IEEE.