CMP404 Report

AR Adventures

1. Introduction

The concept of Augmented Reality (AR) hardly new, being roughly described as far back as 1901 in the book The Master Key by L. Frank Baum (author of Wizard of Oz).

However, implementing this concept proved rather difficult, having been tackled by multiple people throughout the following years, to varying degrees of success.

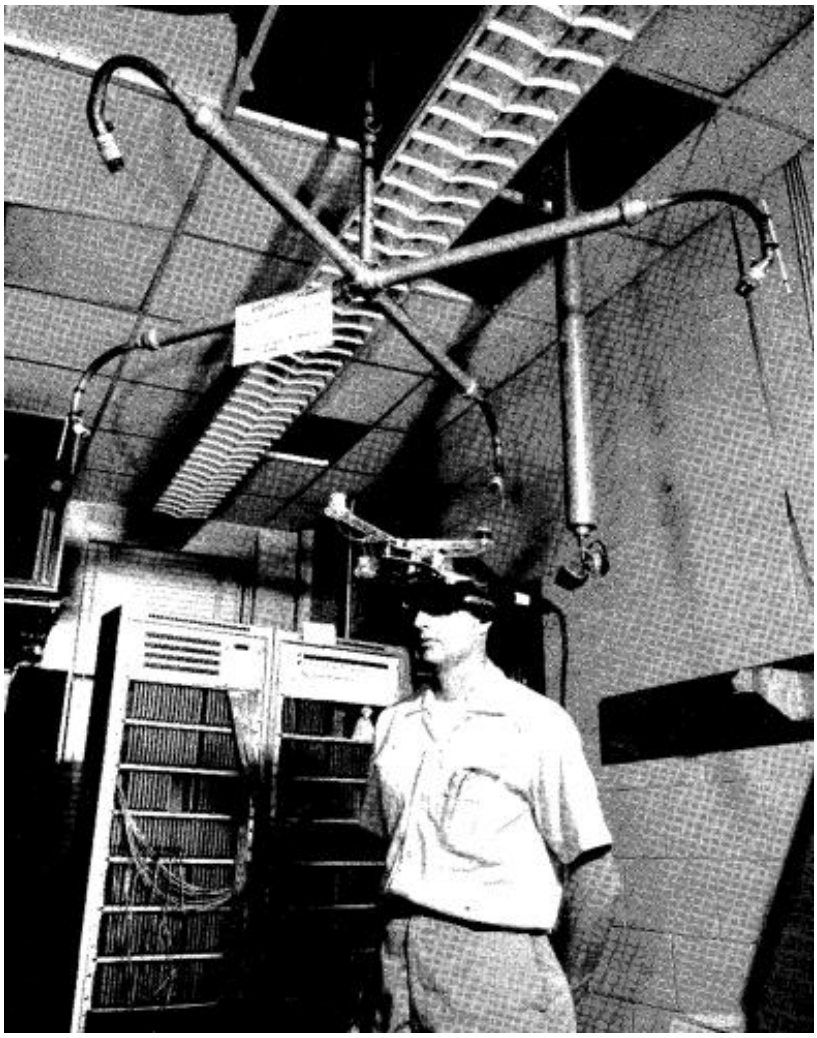

Figure 1: "A head-mounted three dimensional display" deployed by I. Sutherland in 1968 is considered to be one of the first major parts in the history of AR, despite its unwieldiness.

More recently, AR as we view it today started making its way to the general public in 2010s, mostly in terms of advertisements (interactive billboards) or video filters.

In 2012, Google attempted to bring AR to the everyday person in a form of a rather ambitious product, the *Google Glass*. Needless to say – this did not fare as well as was expected and their production stopped in just 3 years.

Only the realisation of AR's potential within phone applications managed to fully re-ignite the excitement of the public.
Namely, the development and subsequent release of *Pokémon Go* in 2016 has brought on a large wave of expectations and excitement about the use of AR in video games.

Unfortunately, upon its release*, Pokémon Go* did not utilize AR as much as the fans hoped for, replacing the wave of excitement by a feeling of disappointment for many.
This was due to the fact that at the time (and even still), logic of plane detection was only present on more expensive devices, and it had a long way to go in terms of performance and deployability.

Figure 2: Pokémon Go, arguably the first big AR videogame title received mixed feedback regarding its (lacking) use of AR.

Despite this setback, many fans still enjoyed this type of Geo-located gameplay, further inspiring many other games such as *Minecraft Earth* or *The Walking Dead: Our World.*

This all changed during the pandemic, as people were not able to leave their homes to go outside. Many companies saw this as an opportunity to bring the outside to their customers' homes instead. A perfect example of a well-developed AR feature that emerged during the pandemic is Ikea's preview furniture option. This lets the user place, move, and rotate a virtual piece of furniture in their real-world environment, making interior design much more efficient and accessible.

Figure 3: Ikea's new AR feature of previewing furniture was a smashing success, fully proving and demonstrating the relevance of AR.

The developed application which is the subject of this report intends to make much more use of AR than games like *Pokémon Go*.
However, it is also worth a mention that geolocation and gameplay surrounding it complements adventure, survival and exploration games in AR extremely well, as it makes the player view the entire world as a game environment to explore.
Therefore, implementing geolocation is also a possibility for the future of this application.
Furthermore, unlike in *Minecraft Earth*, where the user builds miniature AR blocks on a surface around them (often meaning user sits at a table and builds on it), the models in this game are meant to be to-scale with the real world, placed far enough apart so that the user must actually walk around and "physically" explore the augmented environment.

2. User guide

2.1 Input

As with most phone games, the user can interact with the application by tapping or sliding their finger on the screen.

While this application does not utilize microphone input, it is heavily dependant on the camera, as that is an integral part of AR applications.

2.2 UI and Settings

In the following paragraphs a brief description of this application's UI functionality is presented, supported by an annotated image (view Figure 4 below).

Figure 4: Annotated UI elements of the main game screen (right) and settings (left). Refer to section 2.1 for their description.

A: Switches between the settings and the in-game UI

B: Toggle between the Examine and Interact mode

C: Player's current tool level (please excuse the improperly aligned corresponding UI text)

D: Materials in the player's inventory

E: Scan the environment

F: Toggle ARPlanes visuals between default and Spotlight.

G: Resets the game, along with all AR elements

H: Adjust resource spawn density

I: Adjust the opacity of ARPlanes (also enabling full transparency)

2.3 Recommended environment

Due to the nature of this game, large open spaces are preferrable for a more fulfilling experience.
For the purpose of this submission, resource spawning density has been roughly balanced to the size of room 1521 on Abertay University campus.
For more customization, the user can also change this value as they play, allowing gameplay (albeit distorted) even in small spaces.

3. Gameplay

(For a video demonstration please refer to https://web.microsoftstream.com/video/a063406b-d47a-40d2-b307-70df4a5f7cfe)

3.1 First look and Examine state

Upon starting the application, the player will see the world around them as normal, with some simple UI overlay. Shortly after, various resources start appearing around the player.

At this moment, a seasoned survival game player will immediately know the objective.

A more casual player's attention would likely go towards the glowing yellow question mark in the corner of their screen. This button would let them enter the ''Examine'' mode (inspired by old-school MMORPG games like Runescape or WOW), allowing them to tap on anything and receive some helpful information. In addition to this, pressing the button will provide the player with one of few helpful tips, prepared by the developer.

3.2 Gathering Resources

As the main gameplay loop, the player must harvest resources that spawn on the detected planes around them. These consist out of stones scattered on the floor, as well as trees and multiple ore deposits, both comprising of multiple tiers in terms of their 'Health' and the number of materials they drop.

3.3 Environment scan

Because some objects may be occluded by real-world objects or simply hard to notice (especially faraway ore deposits), the user is also equipped with a scanning tool, which highlights the location of all resources, differentiating the colour of the marker by the type of resource (even type of ore inside the deposit).

This functionality is complemented by a visual representation of, which gradually highlights all virtual objects alongside the detected planes, further deepening the feeling of immersion.

3.4 Upgrades shop

Once the player collects enough resources (5 wood, equivalent of 2 small trees), the UI will prompt the user to build a shop. There (as hinted by examining or interacting with the structure) the player can upgrade their harvesting tool in exchange for materials.

4. Software architecture design

4.1 C++ Class structure

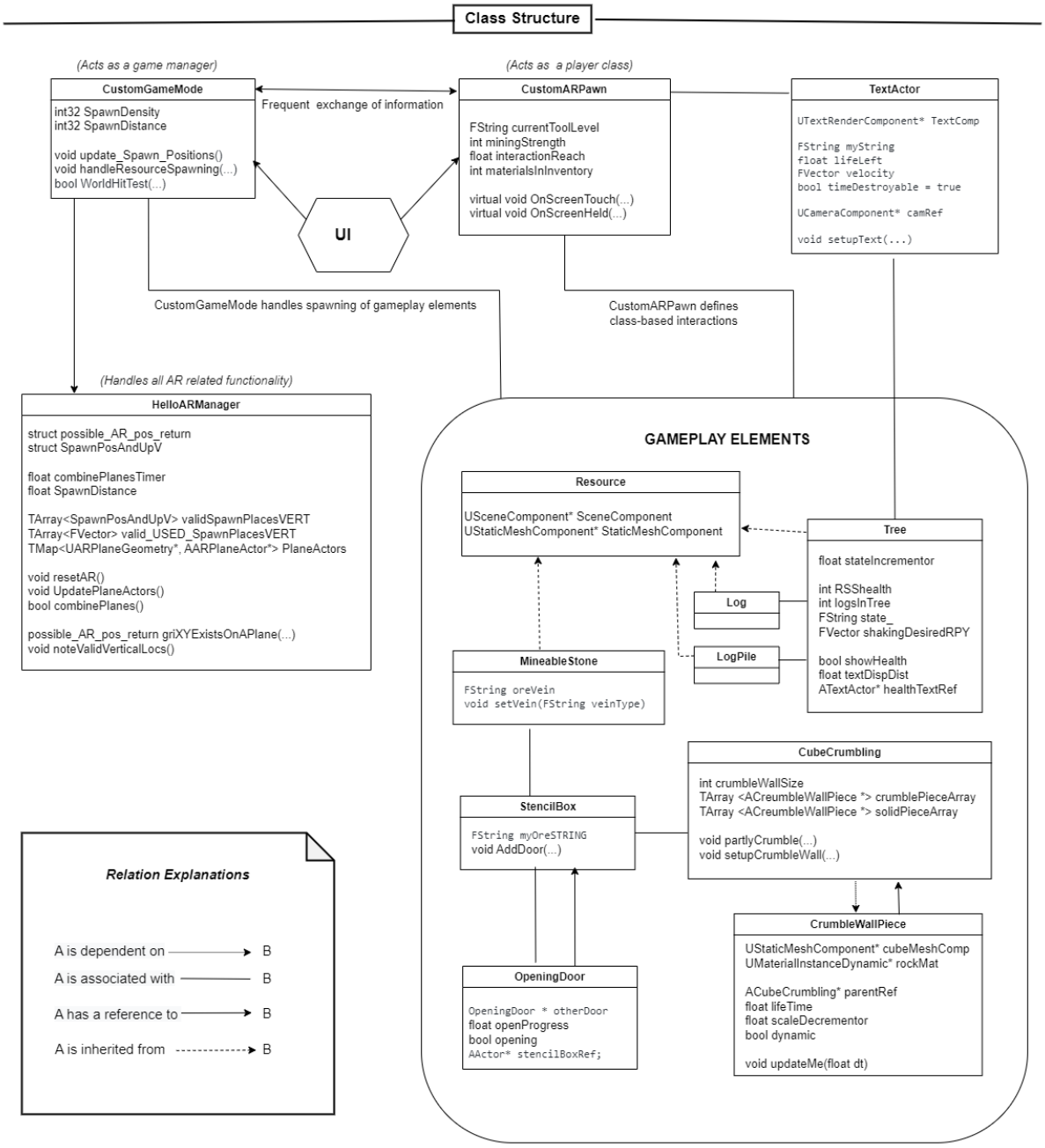

Figure 5: UML diagram of C++ classes, along with some useful annotations. This diagram only displays the most important functions and variables of a class.

Throughout the development of this project, effort was made to structure classes and functions in a clear, efficient, and extendable manner.

The largest portion of the application logic is defined within the *CustomARPawn*, *CustomGameMode* and *HelloARManager* class.

Furthermore, there are numerable additional classes, containing the logic of virtual objects the user can interact with.
Because some of these classes are simply different types of materials, their base logic is inherited from a custom, more general and extendable *Resource* class.

4.2 Multi-platform development

Application development is inherently a multi-dimensional effort, benefiting from the use of multiple tools in its process.
It is only when these efforts are tied together, that one can achieve fully immersive, satisfying, and well-developed results.

In the case of this project, despite the majority of this application's logic being developed in C++, Blueprints have also been utilized, namely for UI and for some specific global modifying of materials, which have also received a considerable amount of development time.

5. Exploitation of AR

When designing any application, the first step is to realise the type of software it will be using (in this case AR).
Developing gameplay that does not require (or benefit from) AR will inevitably lead to issues in terms of performance and ease of use, limiting the potential player base and possibly confusing the users.

5.1 Augmenting the reality

This is why this application aims to directly use the tools and possibilities of AR, attempting to literally augment the user's reality by introducing interactable, real-world scaled, virtual objects into the world around the user.

Additionally, to take the credibility of virtual objects a step further, the possibility of using a stencil buffer in a complementary way to AR was explored.
In the case of this application, *StencilBox* acts as an ore deposit (or as a special easter egg box on vertical walls).
While a square box for an ore deposit might not seem as the most visually pleasing solution, it has been developed mostly to showcase functionality of Stencil buffer in AR.
Using this approach, there are many development possibilities for this technique, such as placing holes, ponds, portals or even rooms inside solid ground or walls.

Figure 6: Stencil-buffer based material on a box spawned inside the ground. Upon moving around, it creates an illusion of a real hole.

5.2 AR to the user

To further establish the concept of plane detection and AR for the user, the default ARPlanes have been replaced by a barely visible circle displayed on the plane which is in the centre of the user's screen.
This effect has been achieved using material programming.
By treating the camera forward direction as a line and only displaying pixels whose distance from the closest point on the line is smaller than the radius of the desired circle, the program is essentially casting a 'cylinder and seeing which points on the plane are inside it.

This results in a circle being cast on the detected plane that gets smaller as the user looks further.

Another noteworthy feature, which is also a gameplay element is the Scan. From a gameplay perspective, it highlights items around the player. However, because this scan also highlights the detected (even when hidden) planes, it ties the real world (or rather its detected planes) even closer together with the virtual objects, as they receive the same visual.

Both the 'spotlight' and Scan functions have been made as material functions, therefore adding the scan visuals onto more objects is as simple as adding the Scan function into their material.

Figure 7: Scan highlights both the ground and the objects it passes through.

6. Reflection

6.1 Noteworthy encountered issues & solutions

As with any application, there were some issues encountered throughout development. First of such issues was the inaccuracy of the default Planes detecting code.
While not a game-breaking issue, the planes are sometimes detected but then overlapped by a bigger plane slightly offset on the Z axis, resulting in a fairly large number of redundant planes, unnecessarily hindering performance. For this purpose, *CombinePlanes* function was developed within *HelloARManager*, which iterates through all planes (currently every 5 seconds) and calculates which planes are fully inside another plane on the X and Y axis. Then, if the distance on the Z axis is smaller than the defined combine threshold, they get removed.

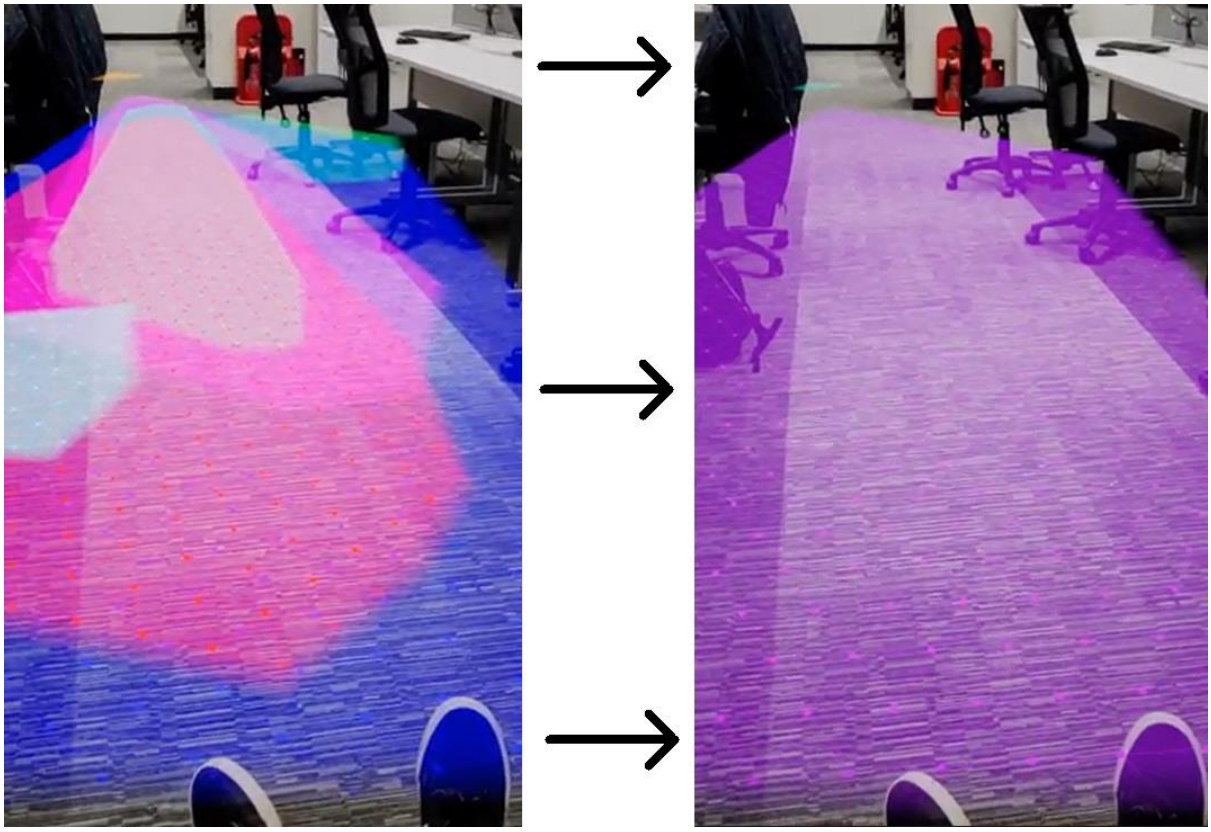

Figure 8: Consuming of planes in action (in the final, submitted, version of the application the larger plane retains its colour).

6.2 Bugs present in the final application

- Examine/tool level text in UI are improperly anchored.
- Because stones are coloured grey, the base occlusion logic often struggles to hide them.
- Last hit on trees does not update their health text to 0.

All these bugs are considered to be relatively minor, not hindering the gameplay in any way and requiring little time to fix.

6.3 Extending the AR

6.3.1 Dynamic Depth Occlusion

Dynamic occlusion in this application has only been implemented in a very basic level. Given more research and development time in this area, one may make it much smoother and work better with grey objects.

Furthermore, the Scan function could be extended using depth occlusion for a better visual effect.

Figure 9: Dynamic occlusion works well with most objects (right), however struggles with grey objects (left).

6.3.2 Multiplayer

To really take this application an extra step in terms of its immersion and enjoyment would be successfully adding multiplayer.
Presumably, there would be some sort of global server.
However, when two players are very close by, a peer-to-peer structure would be ideal for best performance.
Some possible features one can expect from adding multiplayer is that nearby people can interact with the same environment, as well as see each other doing so. These players would presumably have their username hovering above them, possibly along with some stats or visual effects.

As MMO games have proven time and time again, there's no way to add immersion to a virtual game that is as effective as adding other real people into it.

6.4 Final reflection

Overall, this application has successfully achieved its aim.
The user can experience augmented reality with multiple gameplay elements, which all fit AR extremely well.
Despite some functionalities being rather simple, everything has been developed in a highly extendable manner, allowing for further improvements in the future.

7. References

Report sources:

- Sutherland, I. (1968) 'A head-mounted three dimensional display'*, AFIPS '68 (Fall, part I): Proceedings of the December 9-11, 1968, fall joint computer conference, part I,* doi:10.1145/1476589.1476686

- Undisclosed, (2022) 'A brief history of Augmented Reality', *Numerized*, Available at: https://numerized.com/augmented-reality/history-augmented-reality/ (Accessed: 19 December 2022).

- Boland, M. (2019) 'The Age-Old Question: Is Pokemon Go AR?'*, AR Insider*, Available at: https://arinsider.co/2019/04/10/the-age-old-question-is-pokemon-go-ar/ (Accessed: 19 December 2022).

- Brenner, M. (2021) 'The Resurgence of AR and VR Content in a Post-Pandemic', *Marketing Insider Group*, Available at: https://marketinginsidergroup.com/content-marketing/the-resurgence-of-ar-and-vr-content-in-a-post-pandemic-world/ (Accessed: 19 December 2022).

Report Image sources:

- Seb, J. (2017) *Introduction Picture* [Photograph]. Available at: https://digiday.com/media/ikea-using-augmented-reality/ (Accessed: 19 December 2022).

- O'Roque, P. (2016) *Introduction Picture* [Photograph]. Available at: https://mobilesyrup.com/2016/07/12/pokemon-go-review-canada/ (Accessed: 19 December 2022).

- Sutherland, I. (1968) *FIGURE 3* [Photograph]. Available at: https://dl.acm.org/doi/pdf/10.1145/1476589.1476686 (Accessed: 19 December 2022).

Application sources:

SFX:

- Varkalandar (2019) *Cracking Sounds* [Audio]. Available at: https://opengameart.org/content/cracking-sounds (Accessed: 17 December 2022).

- Vinrax (2019) *Pickup/Plastic Sound* [Audio]. Available at: https://opengameart.org/content/pickupplastic-sound (Accessed: 17 December 2022).

- Soundsnap (2021*) Sledgehammer - using to break a block of concrete.* [Audio]. Available at: https://opengameart.org/content/sledgehammer-using-to-break-a-block-of-concrete (Accessed: 17 December 2022).

- StarNinjas (2021*) 10 Impact/Shield Blocks* [Audio]. Available at: https://opengameart.org/content/10-impactshield-blocks (Accessed: 17 December 2022).

- remaxim (2010) *Win Sound #2* [Audio]. Available at: https://opengameart.org/content/win-sound-2 (Accessed: 17 December 2022).
- NenadSimic (2012) *Menu Selection Click* [Audio]. Available at: https://opengameart.org/content/menu-selection-click (Accessed: 17 December 2022).
- NenadSimic (2012) *Picked Coin Echo 2* [Audio]. Available at: https://opengameart.org/content/picked-coin-echo-2 (Accessed: 17 December 2022).

Music:

- RE-LOGIC (2011) *Journey's Beginning* [Music]. Available at: https://terraria.fandom.com/wiki/Category:Music (Accessed: 17 December 2022).
- RE-LOGIC (2011) *Underground* [Music]. Available at: https://terraria.fandom.com/wiki/Category:Music (Accessed: 17 December 2022).

Images:

- REXARD (2019*) Craft Resources Icons* [Images]. Available at Epic Games Marketplace

3D objects and VFX:

- NOTLonely (2017*) Greenwood Fantasy Village* [3D assets]. Available at Epic Games Marketplace
- tharlevfx (2016*) RPG FX Starter Pack* [VFX]. Available at Epic Games Marketplace