Collaborative Augmented Realities Research Notes

The ability for the users to be able see each other within the gallery is of primal importance if they are not physically within the space. With this in mind, we needed to discover a method to accomplish this, especially with the current situation in play, meaning a proper in-person demonstration of the program's capabilities will no longer be possible potentially. This research document will showcase the research undertaken prior to the practical that demonstrated this exact measure and further research conducted afterwards to fit to our specific context. This will also be done so in tangent with constructing said component and testing it for its functionality with another group member to see if any errors arise in the process. The two team members that were involved in this construction and testing process were Thomas Bickley and I as we were the two members that were tasked with this component. The component as a whole can be viewed on GitHub, with updated versions being added as it gets further refined.

The research conducted beforehand consisted of determining varying methods of how to first lock the camera to a user, so they would get a first-person perspective of the constructed gallery. This however would also bring up the challenge of keeping the camera in this state whilst the user was navigating the space. The most we discovered was how the camera would be able to follow an individual from behind, this would prove to be challenging for a user if their avatar was visible to themselves, meaning their view would be blocked. We considered taking a third-person approach in order to compensate for this if need be but would prefer following the intended design. After this we began brainstorming how to actually showcase the user seeing others and their respective actions through the program, this mostly consisted of a discussion as to whether we would actually implement this to be a working factor or take a smoke and mirrors approach as this is merely a proof of concept.

This then leads into the third week and the practical in which we actually learnt of a system called Photon and its' networking capabilities, from here we then were able to use Photon to host the application server and via the use of Photon's own documentation, we were finally able to construct the camera to be a fixed first-person perspective via determining whether or not the avatar was the local player, meaning the user would see what is in front of their avatar and not someone else's, then using documentation about Unity's camera. This approach of using the camera to have a first-person perspective though wasn't explicitly demonstrated within the online documentation and meant we still needed to tweak certain aspects to fit our purpose. This was done by changing it from being fixed to track the back of the user avatar to being fixed to the avatar's face itself and showcasing what is before it. This proved to be somewhat challenging at times as with the camera's positioning would occasionally show the avatar's components, such as it's nose and eyes, meaning the user would essentially see these floating objects affixed to their screen. This involved some alterations in the program values so to essentially position the camera right at the tip of the avatar's nose to avoid this.

Once this was built, we determined that it doesn't run on the android device for some reason, failing to load the program entirely. Moving on from this with the aim of testing the functionality, the core component that was constructed was then sent to the other to test. This was a success as both users could see the other avatar turning and moving around. We then discussed the possibility of the application not working on the android device due to the lack of internet connection, this was proven to be the problem, afterwards we were both able to use the devices to run the application rather than just moving and seeing the movements within Unity itself.

The next aim was to work off this model and use gyroscope to be able to track the user's movements, removing the need for fixed UI elements on the phone screen, allowing for more organic navigation through the digital space.

This was all crafted because of Photon Unity Networking (PUN), a system that allows one to create a server that can be joined by others who have the same code. It also supports cross-platform play, allowing people to join no matter their device, this though not as necessary for the project currently would be useful for when it is expanded to also support iOS systems. This would allow most individuals to be able to access the virtual portfolio and interact with its components with no device restrictions. The fact PUN also implements a client-2-server model means that the client, in this case the users of the application, doesn't have to be concerned about how well the server is performing in order to meet the request-response communication. Another benefit to this model is that multiple users can be supported as it is designed as a centralized system, in order to prevent this from potentially being overworked, a load-balancing system, a technique that optimises the response time for each task via distributing the tasks over various resources, can be devised in order to expand the server over multiple machines. This method avoids the redundancy and resource downtime that may be witnessed in other networking methods.

PUN also grants us the ability to allow the users to select their server if need be or be sorted randomly into one, though we are unsure whether or not we will fully implement this the desired approach would be for the users be able to enter a server with a link or the name of the lobby that would be shared to them, allowing people to enter the portfolio only if they have this. The benefit to this approach is that it does provide some privacy for one's portfolio, meaning that not just anyone can enter.

This whole process is a large step to completing one of our central tasks as a group, being the user being able to connect into a server and view the changes. Once the movement aspect has been completed alongside developing example artefacts that can be viewed within our demonstration, the group can then move towards implementing the UI elements that are attached to not just the virtual space but the artefacts themselves. Enabling us to then further construct how users interact with the artefacts and being able to witness the changes that stem from these interactions. Though there were some drawbacks at the beginning of this project cycle we are now back on track to completion.