2020-01-24 Biweekly Report

Tasks completed

1. Project report website has been set up and can be accessed via http://students.cs.ucl.ac.uk/2019/group11/index.html
    a. Detailed pages including Home, Requirements, Research, HCI, Design, Testing, Evaluation and Appendix.
    b. A project video that we recorded, including the background of this project, the design of system architecture and some key features.
2. Built VR meeting environment using Unity
    a. A simple round-table environment implemented.
    b. Presenter and attendees standing/sitting around the table.
    c. Oculus device set up. A user may see the VR meeting environment in full vision via Oculus Go.
3. Back-end server for connection between attendees.
    a. Signalling.
    b. Meeting session management: Initiate, join or leave a meeting session.
    c. Use TCP and HTTP to handle user requests.
4. We resolved real-time voice chat solution by introducing Oculus party, which is the Oculus platform voice chat API, to our project.

Problems and difficulties encountered

1. The integration between Oculus Go and Unity was particularly complicated.
    a. The official documentation for Oculus Go development is very unclear, as it tends to mix between other Oculus devices such as Gear VR or Oculus Rift.
    b. All Oculus gadgets are classified under Android device which is better handled using a Windows machine. However, all team members use Mac OS machines, so we have to research into online tutorials to help with the deployment process.
2. The Oculus Go device is becoming outdated and development is very inefficient and not dynamic.
    a. Changes are constantly being made to the Unity model, such as model building, visual rendering and add in object functions.
    b. Unity editor does not recognise the Oculus Go headset and controller inputs when in the ‘game’ mode: possibly due to the fact we are using a Mac OS rather than Windows PC. However the editor is compatible with Oculus Rift and GearVR, both are more advanced devices than the Go.
    c. In order to see these changes, we have to build an Android Package (.apk) of the entire project, then deploy onto Oculus Go via a driver called Android Debug Bridge (adb), then open it up in the headset, to finally be able to see and test the new features.
    d. This cycle has to be repeated every time to see the changes, which is time consuming as the size of the project grows.
    e. It might be possible to find a workaround.
3. Having tried many approaches, the game objects dose not seem to be reacting to the controller inputs when testing the UI buttons.

Tasks planned for the next two weeks

1. We aim to integrate our Unity model with better rendering possibilities.
    a. Add colours to the models for better user experience.
2. User gestures remain to be implemented.
3. The ability for the presenter to upload presentation files.
4. UI design integration.
    a. Name tags and speaking indication.

    b. Interactive buttons.
5. Integration test and end-to-end test coverage.
6. Utilise the hand held tracking device connected to the Oculus Go.
7. Integration between Unity application and central server.
    a. PC-based platform testing.
    b. Oculus Go based testing.
8. User account management.
9. Integration with IBM Watson Personality Insight API.