

# UCL Peach Group 39: Reality

## Bi-weekly Report #2

Timur Kuzhagaliyev  
Fraser Savage  
Laura Foody

28<sup>th</sup> October 2016

### 1 Overview of the last 2 weeks

One of the main tasks we focused on during the last 2 weeks was finalising the requirements, deliverables and the toolchain for development. We've had a couple of meetings with our client where we discussed which features they expect the final Proof of Concept (PoC) implementation to contain and how we as a team could achieve that. As an outcome we were able to finalise the list of requirements and deliverables for the whole project, which you can find in the appendix.

We've also finished setting up a development toolchain for HoloLens development and familiarised ourselves with all of the solutions we're going to use (Unity HoloLens Technical Preview, Visual Studio 2015, HoloLens Emulator, C# language). This allowed us to begin working through the tutorials provided by Microsoft Holographic Academy (<https://developer.microsoft.com/en-us/windows/holographic/academy>) which in turn gave us a chance to better understand how HoloLens works, what it is capable of and how user input can be interpreted. This knowledge led us to introducing a separate web interface to import models and DICOM images, more about it in section 2.2. Additionally, we begun thinking of the ways the user can interact with our application, which is covered in the same section.

As for team communication: We decided to stay on Slack because the advantages of integrations into services like Atlassian JIRA and GitLab outweighed the inconvenience of having to keep up with both Ryver and Slack chats. Additionally, we have established a logical file structure for both the shared Google Drive folder and the GitLab repository and begun documenting both so that future developers will be able to easily continue development. You can find more information in the summary of meetings held below.

### 2 Summary of meetings held

#### 2.1 Friday October 14<sup>th</sup>, 2016

During this meeting our client had a chance to try out HoloLens for the first time, and understand its capabilities and limitations. One major limitation we agreed on was the limited field of view, which ruins the immersion of the user when using the device. As a result, we had to abandon the idea of doing 3D data visualisation on HoloLens as it would be very hard to produce a good effective visualisation with such a limited field of view. We've also discussed which gestures would be the most intuitive to use to control our application and manipulate models in 3D space, and after some research we came to conclusion that introducing our own gestures would not be feasible since even the gestures predefined by Microsoft (e.g. "Bloom", "Gaze", etc.) are not as reliable as we would want them to be.

#### 2.2 Friday October 21<sup>st</sup>, 2016

Taking into account the conclusions made during the last meeting, we begun designing a user-friendly interface. Making it intuitive has posed a major issue because we only have a limited amount of gestures we can use to

control an application with a reasonably big amount of features, so we decided to add voice control support to our application to makes some tasks easier.

Additionally, while discussing how user can import models and DICOM images into our application we decided to introduce a separate web app which will do exactly that, i.e. let the user upload models and DICOM images to a remote server where they will be stored. When our application will be launched on HoloLens, it will retrieve the list of all available models using a web API and let the user choose the files they're interested in, downloading and displaying them after the selection has been made. Considering the research done so far, this seems to be a feasible concept but the downside is HoloLens will require internet connection to retrieve new models (models that have been downloaded before can be cached).

### 2.3 Friday October 28<sup>th</sup>, 2016

During this meeting we had a chance to present the refined list of requirements and deliverables as we see them to our client, who has approved it after making a couple of corrections. This is a huge milestone in our project since now we can begin prototyping the final product and planning the experiments we need to conduct before we begin producing PoC implementation of the system. Additionally, we managed to resolve the concerns our client had about only being able to change the colour/transparency in the exploded view and only being able to view 1 DICOM image at a time by explaining the rationale behind the former and removing the limit from the latter.

Our client, Dr. Ramachandran, introduced us to Lorenz Berger, who is working on converting CT scan data and DICOM images into 3D matrices which can then be used to produce colour-coded 3D models. Lorenz gave us a lot of insight into how these matrices are produced and how we can use them in our application to generate high quality models. He also told us he'll be able convert CT scan data into 3D matrices per our request, which will make it way easier to test our PoC on real cases.

## 3 Tasks completed

- All team members now have the complete toolchain for HoloLens development and can begin conducting experiments.
- Analysed legacy source code and pinpointed the elements we're going to reuse.
- Produced some basic concepts of interaction with the user.
- Finalised the requirements and deliverables for the project, as seen in the appendix.
- Conducted the first experiment involving loading models on the HoloLens.
- Setup JIRA with a Kanban board to track progress and responsibilities.

At this point we're ready to begin conducting experiments and have already produced some concepts which require testing to determine their feasibility, so it is safe to say that the project is progressing well.

## 4 Problems encountered

- Using gestures to interact with our HoloLens application turned out to be not as reliable of a process as we expected, hence we'll have to resort to using only the gestures predefined by Microsoft.
- The issue above in turn forced us to reconsider the current mechanisms of interactions with the our application, so we're planning to optimise our controls concept to be as simple as possible while remaining powerful and intuitive.
- Now that we decided to use a web interface to import models, we have to find a reliable way to communicate with the HoloLens over internet.

- Producing high quality models from 3D matrices provided by Lorenz seems challenging, more research required

## 5 Plan for the next 2 weeks

- Complete the template for the static website, begin producing documentation for the research done so far and its conclusion which will go on said website.
- Conducts experiments using the HoloLens Emulator to test out how well our concepts for gesture control work in practice.
- Study the models, CT scan data and binary masks provided by Lorenz to see how we can produce our own meshes using the MCubes algorithm.
- Produce user stories to better understand how our client wants to interact with our system.
- Research ways we can connect HoloLens to a web API.

## 6 Individual reports

### 6.1 Timur Kuzhagaliyev

I spent the last 2 weeks researching capabilities of HoloLens and producing concepts for the user interface of our application. Major topics I researched involve loading files into HoloLens, communicating with HoloLens over internet, capabilities of Windows Holographic (operating system running on HoloLens), gestures pre-defined by Microsoft and creating custom gestures, as well as which HoloLens APIs are available in Unity3d.

This research led to me proposing an idea of introducing a separate web interface and a web API (see section 2.2) which the HoloLens will use to download models and DICOM images. In my opinion, this will not only make importing models into HoloLens easier by avoiding most of the complicated interaction with Window Holographic, but will also allow us to make the interface of the HoloLens app simpler by making the web app do some of the legwork.

Additionally, I analysed the requirements for our PoC and begun working on the concepts of the HoloLens app UI and UX, trying to keep them as simple as possible while still satisfying the requirements. I decided to stick with the 3 predefined HoloLens gestures: "Bloom", "Air Tap" (used with "Gaze") and "Manipulation" (summarised here: <https://support.microsoft.com/en-gb/help/12644/hololens-use-gestures>). I have been able to produce what seems to be a reasonable interface concept using the combination of these 3 gestures and voice control, some attempts to visualise said concept for my teammates can be seen in the appendix.

Last but not least, I have refined the existing template for our static website with documentation, established a logical structure for our GitLab repository and Google Drive folders, and begun playing with demo projects provided by Microsoft in Holographic Academy.

### 6.2 Fraser Savage

Since the previous report I have set up a JIRA project for our team, as we opted to use that over Trello to keep track of project tasks and epics. Once the JIRA project's Kanban board had been set up I integrated it with an internal Slack channel dedicated solely to update messages posted regarding JIRA, Version Control and Continuous Integration.

Additionally, I have begun to experiment with Unity and the HoloLens emulator. My initial experiment has involved loading a set of meshes from the db28.zip file on the client's Google drive into a Unity HoloLens application and creating a gaze cursor to interact with the meshes. Following on from this I will be learning how to implement interactions and manipulations of the subject matter as well as carrying out some research on interaction design for the experiment.

To aid further experiments I will be producing a Unity Project template for future experiments based on discoveries made during this first experiment, while incorporating Microsoft's HoloToolkit for Unity (<https://github.com/Microsoft/HoloToolkit-Unity>) where it will assist in rapid experimentation.

### **6.3 Laura Foody**

In these last two weeks I have been reading through the documentation of the code that was written for the project last year. I have installed the hololens emulator so that I will be able to start developments for the project. So that I don't have any difficulties later on; I have also been using Code Academy to learn git, as that's something I'm not very confident using. I will be starting some hololens tutorials on unity so that I can start with making some models. I will also be starting to write up our user stories.

## 7 Appendix

Figure 1: Requirements as finalised during the meeting on Friday 28th of October.

3D models
User must be able to import models into the system. (FBX, STL, VTK, other formats tbc.)
User must be able to project imported models in 3D space using HoloLens.
User must be able to register the projected model with an anchor in 3D space.
User must be able to move, rotate and scale projected 3D models.
<a href="#">User must be able to explode projected model</a>
User must be able to change colour and transparency of various parts of the model after it has been exploded
User must be able to save colour/transparency information so that the model looks the same the next time it is loaded.
User will only be able to view 1 model at a time.
DICOM images
User must be able to import DICOM images into the system.
User must be able to project and register (lock in some position) DICOM images in 3D space.
User must be able to move, rotate and scale DICOM images.
User must be able to scroll through projected DICOM images.
User will be able to view multiple DICOM images at the same time.
Real time patient data
User must be able to view patient data in real time. (Pulse, blood pressure, features tbc.)
User must be able to toggle between poster view and heads-up display.
User must be able to quickly reveal/hide heads-up interface.
User must be able to show/hide certain parts of the interface, e.g. only show pulse
Other
Agree and publish a VR/AR design guide which matches the 2D PEACH design guide, on the same website

Figure 2: Deliverables as finalised during the meeting on Friday 28th of October.

Deliverables
HoloLens application that supports features listed in the column on the left.
A web app that allows user to import models and DICOM images and exposes an API that HoloLens will use to fetch said models/images.
Documented source code for both applications listed above.
Exhaustive manual and videos explaining how to use both HoloLens app and web app.
A static website containing all of the information about research done, requirements gathered, experiments conducted, prototypes produced, etc.
Medium articles documenting progress

Figure 3: Some concepts for the ways user can interact with the system using gestures and voice commands.

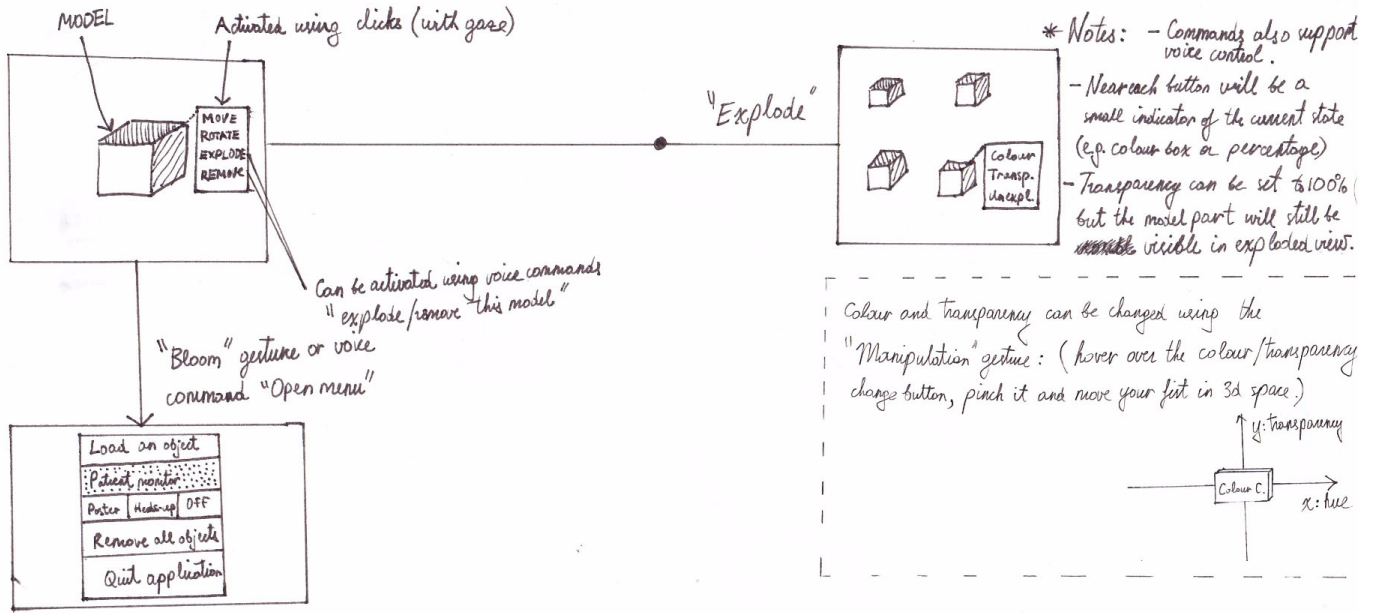


Figure 4: A concept of wrapping every object into a bounding box which will later be used to interact with the system. X and Y are the buttons the user can use to perform certain actions, like translating or scaling the model.

---

Bounding box and buttons appear when "Gaze" cursor hovers the bounding box (not model). After the cursor leaves the bounding box all elements but the model itself become fully transparent with 1 second delay.

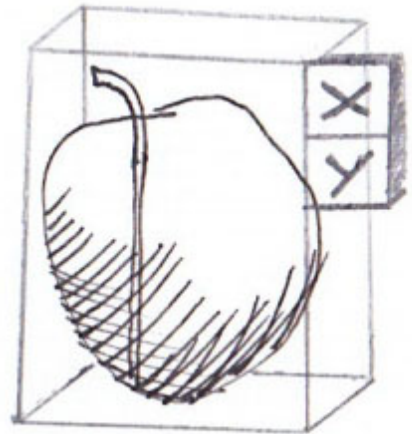


Figure 5: A render of a tumour in a kidney, inside the HoloLens emulator.

