UCL Peach Group 39: Reality

Bi-weekly Report #4

Timur Kuzhagaliyev
Fraser Savage
Laura Foody

2nd December 2016

1 Overview of the last 2 weeks

During the last 2 weeks our primary focus was implementing as many key features of our prototype as possible for the presentation on the 5th of December. We've met with our client and supervisors several times to discuss our progress and improvements that can be done to our existing solutions, and as a result we made numerous changes which are summarised below.

One significant feature that was recently added to the scope of our project is the ability of creating data markers and attaching them to different objects. The concept is briefly demonstrated on Figure 2 in the appendix. Data markers will store various data like notes, multimedia files and SNOMED CT codes, which can later be exported as a holographic case study in a brand-new file format which we are about to develop.

Finally, considering the new requirements and the upcoming presentation, some changes were made to the Gantt chart submitted with the previous report. The updated version can be seen on Figure 1 in the appendix, and by the time of writing of this report we've successfully accomplished most of the tasks described there.

2 Summary of meetings held

2.1 Monday November 21st, 2016

The primary focus during this meeting was the API and the way our web app and HoloLens itself will interact with it. We discussed strategies of continuous deployment of the backend for our web app and the API to Azure services. We decided to stick to JSON Web Tokens as the primary authorisation scheme that will be used by both web app frontend and HoloLens app. We established the implementation design for our preprocessor in such a way that we'll be able to easily add new features.

Additionally, we discussed which gestures should be used for different manipulation modes such as scrolling through a DICOM image or changing the transparency to ensure consistent user experience within our app. Finally, we agreed on producing additional sketches and storyboards to better understand the needs of our client.

2.2 Friday November 25th, 2016

On this meeting we discussed our prototype with our client, and received useful feedback and suggestions on possible improvements. They involved adding scanogram or other indicators to DICOM scrolling to show current progress, 2 different types of scrolling (rough and image-by-image), alternative way of exploding models (parallel to line of sight instead of perpendicular), using photorealistic textures for better immersion, and changing transparency of some elements before loading the model to make work flow quicker. Additionally, our client suggested merging medulla and kidney into a single object and that compressing CT scan images to save space might not be a good idea.

2.3 Tuesday November 29th, 2016

On this meeting we discussed our project with Dean Mohamedally, who gave a lot of useful feedback and suggested we introduce some key features which were later approved by our client. One of said features were so-called data markers, which can be attached to any part of a model mesh. These markers will hold written notes, multimedia files such as voice recordings or snapshots from Skype calls, or SNOMED CT codes.

Additionally, Dean gave useful insight into how to structure our presentation on the 5th of December and suggested that we should develop a brand-new file format to store holographic patient case data that can be generated using our app.

2.4 Wednesday November 30th, 2016

We used this brief meeting to discuss the new requirements, especially the new data marker concept. You can see the concept we came up with on Figure 2 in the appendix. Additionally, we discussed the progress done by each team member and adjusted our working schedule to make sure we implement at least some of the new requirements by the presentation of the 5th of December.

2.5 Friday December 2nd, 2016

During this meeting we had a chance to go over our presentation with Dean and adjust the content to ensure that it conveys exactly what our project is about and that the purpose of the project, current architecture and possible use cases are clear from our slides. Additionally, we had a chance to go over the project website with our TA who gave us useful feedback on which areas can be improved and how we can adjust the structure of our website to better represent our work.

3 Tasks completed

- Developed a REST API that serves models and DICOM images with relevant meta data to the HoloLens
- Added an authorisation system to said API using JSON Web Tokens
- Developed a file preprocessor for the API that processes and converts files
- Created a persona, storyboards, sketches and scenarios for our application to better understand user needs
- Refined project requirements, adding MoSCoW analysis and reconsidering priority of some requirements
- Finalised web app mockups, begun producing the UI for the web app using React
- Developed a new model viewer prototype, with context menus, improved main menu, support for multiple DICOM images and models
- Produced a list of improvements that need to be done to the project website

We're definitely doing good progress and I believe we'll have a prototype with a lot of key features implemented ready for the presentation on the 5th of December.

4 Problems encountered

- The built-in Unity APIs for web requests seem to be poorly designed and don't work well with our current API. We'll either have to write our own libraries for communication with the API and retrieval of files
- The version of HoloToolkit we're using seems to have a different approach to manipulation than that described in Microsoft Holographic Academy, so we'll either have to develop our own libraries or think about different design for our API.

- Developing a single API that would be used for both frontend and HoloLens meant that we had to work on our methods of retrieving data from the API for our React web app using AJAX as opposed to using of the existing framework that simplify server communication with React.

- Loading multiple models with mesh colliders on HoloLens has a significant negative impact on performance. We'll have to find ways to minimise polygon count on our models without ruining user experience.

5 Plan for the next 2 weeks

- Begin designing a file format to store holographic patient cases with models, DICOM images and data marker data such as multimedia clips.

- Combine the newly-developed module for downloading files from API to the HoloLens with the main model viewer prototype to allow users to choose which files available on the server they want to view.

- Finalise the web app interface using React with AJAX requests to the API to fetch user data and upload files.

- Improve performance on HoloLens when viewing multiple models, possibly by optimising models during loading and then applying various smoothing techniques to artificially increase the quality.

- Refine the contents of the project website to make sure it clearly conveys every single aspect of our project.

6 Individual reports

6.1 Timur Kuzhagaliyev

In the last 2 weeks my primary focus was developing a prototype for our model viewer. I was aiming at implementing enough features to demonstrate a typical work flow of analysing models and DICOM images of our app on the presentation of the 5th of December.

I've made several improvements over the previous iteration of the model viewer, including but not limited to: Improved dynamically generated bounding boxes that are now used as anchors for various manipulations, such as model scaling or DICOM image scrolling; A new context menu framework which can be used to attach context menus and program buttons to any object on the scene; A voice instruction manager that plays audio clips with short tutorials whenever a specific action is triggered; A manipulation controller that deals with parsing user gestures and applying various transformation to objects; Support for multiple models and DICOM images to be loaded at the same time; Improved shaders for transparency adjustment; Somewhat better performance due to model optimisation during loading;

A short demonstration of some of the features mentioned above can be seen on this video.

6.2 Fraser Savage

Since the completion of the hologram rotation interaction the main body of my work has been on rapid prototyping of the REST API in Java EE. The functionality which has been prioritised is all which relates to serving meta-data and objects to the HoloLens at runtime.

The API endpoints implemented as of this report are the POST request for /api/authenticate and the GET requests for /api/list, /api/model/id and /api/dicom/id. In general, what this allows a client of the API to do is receive a response containing a list of models and DICOMs stored on the server which have been marked as visible, along with their id and any linked meta-data. An example of meta-data would be the list of parts which make up a model, as well as the red, green and blue colour values for each part. From this list, a request can be made to retrieve the model or DICOM by its listed id. All GET requests are denied with a HTTP 401 Unauthorised code unless the client provides a valid authorisation token as part of the request, which is obtained by the POST authenticate request.

As mentioned in the previous report, I also produced a Unity project ”starter template” for experiments, which was in part utilised and modified by Tim since its inception. In addition to the work on the API and the production of the ”starter template” I also explored the programmatic conversion of STL files to OBJ files server side, and in doing so managed to use the Java Native Interface to work with the Assimp C++ library and create a function which imports STLs into runtime and exports them as an OBJ to the file system.

6.3 Laura Foody

In the last two weeks I have been working on designing the user interface that will allow users to upload their objects onto the hololens. I started by creating some mockups of how I wanted the user interface to look, I sketched some at first and then after running them by my team mates, created some electronic mockups on balsamiq. I used the UCLH Peach design guide website to help me to plan out how I wanted the pages to look, and I got feedback from our client too and used that to make a few final changes. I then created a persona based on our client and used this to write a couple of scenarios to show how our app would be used in real-life situations. I also drew some storyboards of the must-have requirements from our MoSCoW analysis.

I then moved on to creating the actual UI, our client wanted us to use electrode.io which is a node.js and react.js based framework; however this was very complicated to use so we decided as a team just to use react.js and then at a later date we can try to incorporate the electrode framework. So far I have successfully implemented user authentication, however it does use stormpath (for testing purposes) but I need to work on using the APIs from the back end.

Finally, in the last two weeks I have looked at evaluation methods for prototypes and drafted up a table we can use to evaluate our prototypes. I also compared our project website to the example websites that were emailed to us, and wrote a list of improvements to make to our project website.

7 Appendix

Figure 1: Updated Gantt chart for the plan from 18/11/16 till 5/12/16.

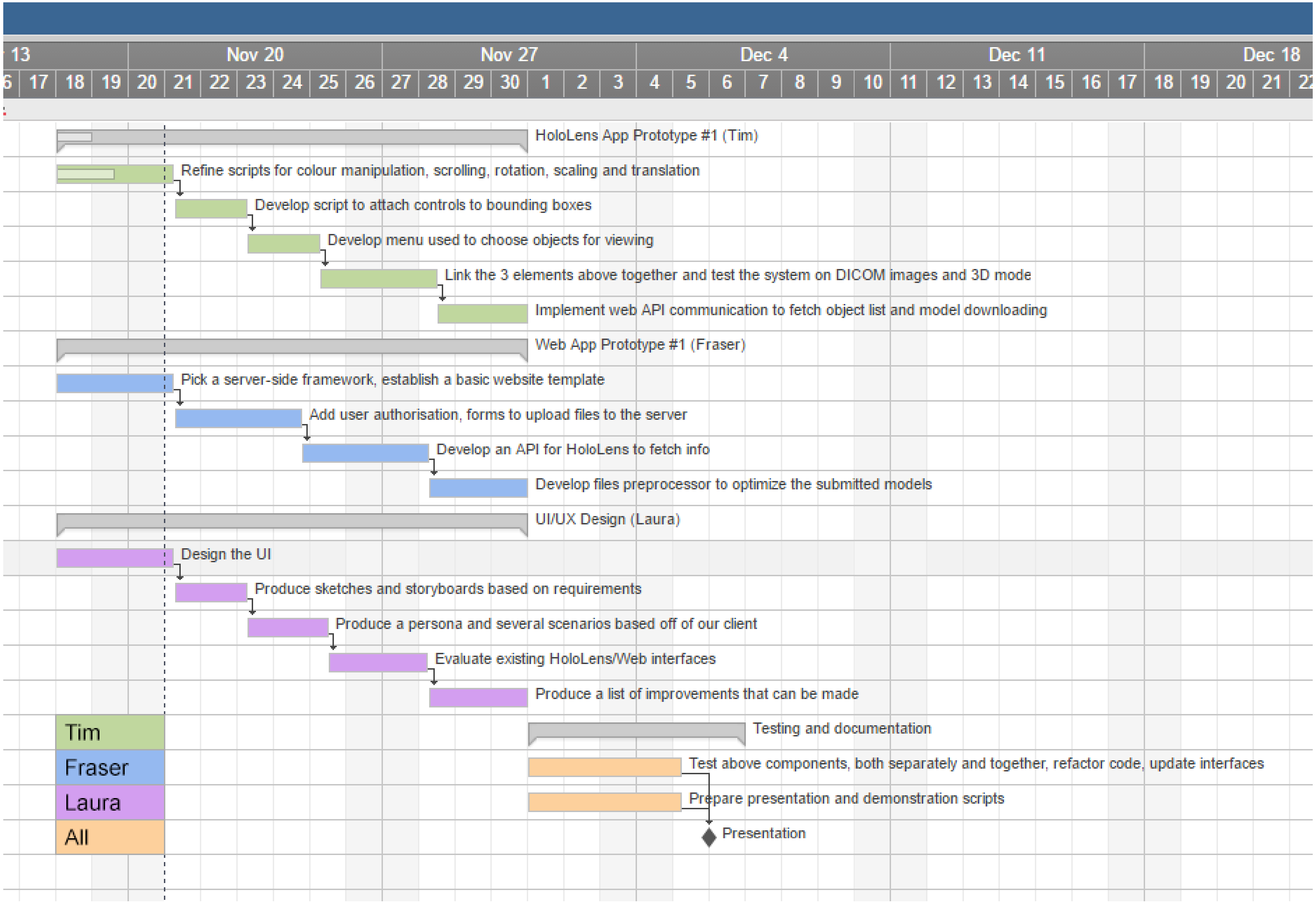
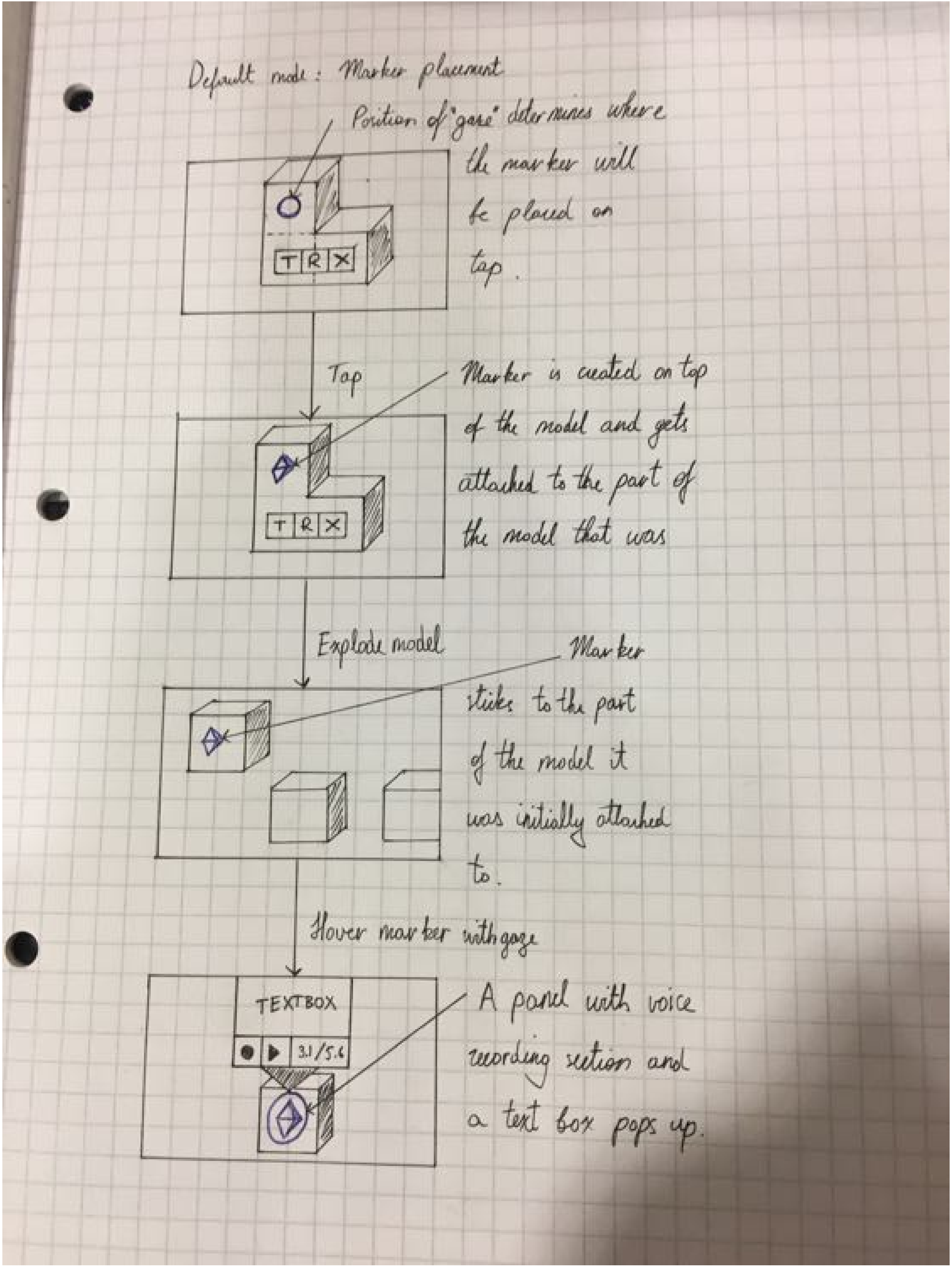


Figure 2: Data marker concept.



Default mode: Marker placement

Position of "gaze" determines where the marker will be placed on tap.

Tap

Marker is created on top of the model and gets attached to the part of the model that was

Explode model

Marker sticks to the part of the model it was initially attached to.

Hover marker with gaze

A panel with voice recording section and a text box pops up.