UCL Peach Group 39: Reality
Bi-weekly Report #10

Timur Kuzhagaliyev
Fraser Savage
Laura Foody

24th March 2017

1 Overview of the last 2 weeks

In the last 2 weeks we were focusing on the development of the API. Below you can find the details of decisions made and actions taken.

- We have extended the API to use Azure Blob Storage as a secure storage for all files stored in our application. This feature will allow us keep the files separate from our main API server, meaning we can easily redeploy a server into a new location without losing sensitive data. Additionally, the abstraction we used to implement Blob Storage allows us to easily switch to a different storing solution in the future, e.g. NHS N3.
- Using the Blob Storage feature described above, we've added support for CT scan uploading. CT scans can now be attached to holographic patient cases and manipulated using the relevant API endpoints.
- We've introduced a concept of CT scan status to our holographic patient case specification. Users will be able to request conversion of a CT scan using a special button in the webapp. Then, the webapp will poll the API to check the status of a CT scan, which will either be Idle, Currently Being Converted, 3D Model is Being Generated or Converted.
- On Friday 24th we've shot a promotional video for PEACH together with our client and Dean Mohamedally. The video was orchestrated by a filming crew from Microsoft, and involved several interviews about our project and our future plans.
- We've continued working on integrating the neural network provided by Innersight Labs into our project, but we've encountered an issue when it came to processing the output of said neural network. We have to run several algorithms to produce a 3D model in a common file format based on the raw bitmaps outputted by the neural net, which proved to be quite computationally expensive. We'll be trying to improve performance of this part of the application as much as possible since it's a crucial part of our system.
- Collaborating with the 1st year team, we've made several improvements to the web app prototype. We've implemented a page that allows users to update their settings. Additionally, the first year team, using the API endpoints we've recently added, implemented listing of holographic patient cases, file upload and real-time 3D model previews in the browser.

2 Tasks completed

- Added support for Azure Blob Storage in the backend API

- Begun producing a manual for the end users, including FAQ section, tutorials for both webapp and HoloLens application
- Extended API endpoint functionality for holographic patient cases, which now supports CT scan uploading
- Performed first implementation experiments with the neural network for CT scan conversion
- Added support user settings into the webapp, supervised real-time model viewing in the browser from the first year team
- Took part in production of a promotional video about the PEACH project, including some footage of the project in action and short interviews

3 Problems encountered

- Converting the output of the neural network from Innersight Labs (NIfTI files, `.nii.gz`) into 3D models that can later be used by our application proved to be reasonably resource-intensive and time-consuming task. We'll have to look into ways of speeding up the process to make our pipeline more convenient for our users.

4 Plans for April

- Finalise the specification for the backend API
- Release a manual for the end users of our system, including tutorials for both the web app and the HoloLens application
- Finalise integration of the neural network into the backend API
- Improve the visuals of the models in the HoloLens application
- Produce a prototype of the overall system
- Extensively test the whole pipeline in action, starting from uploading CT scans to web app and finishing with annotating models in the HoloLens application and exporting holographic patient cases

5 Individual reports

5.1 Timur Kuzhagaliyev

In the last 2 weeks I focused on integrating the the neural network provided by InnerSight Labs into our backend API. The neural network itself run as a Docker container and was pretty straightforward to setup. As it is a simplified version, it was only trained to work with several pre-selected CT scans, and it successfully processes CT scans by taking NIfTI CT scan bitmaps extracted from DICOM files using a separate script. The output produced are also NIfTI bitmaps, which have been segmented to represent different parts of the model (kidney, urinal tubes, veins, arteries, tumour, etc.). Although these bitmaps represent volumetric model data, they must be converted into a common 3D model format before we can use them in other parts of our application. This conversion proved to take a lot of resources and time to complete, which is something I've been trying to fix to improve the performance of the system. I'll try to find a more efficient algorithm for the job.

Additionally, in the last 2 weeks I kept in touch with the first year team to make sure that they're not experiencing any issues with our part of the system and also to synchronise our progress and make arrangements for putting different parts of our system together for the final prototype.

5.2 Fraser Savage

Work done since the last report has consisted of implementing the base tier of functionality for holographic patient cases and fixing bugs as they arise by working in tandem with the first year team as they make use of the API.

The functionality achieved to date regarding cases enables the creation, retrieval, deletion and listing of cases. In addition to this, I have introduced the capability to upload DICOM files to Azure blob storage and attach them to a case. This functionality performs its job although it will need tweaking to make it a bit more flexible.

The next goal of focus for me will be to add the ability to update users, update cases and to complete the DICOM functionality for patient cases. Once those goals are achieved I will do the same for models and files as I did for DICOMs.

5.3 Laura Foody

In this last sprint I finally made a successful request to the login API. I had to fix my ajax request to expect the correct content type. When then working on getting the webapp to store the token in localstorage I encountered an error and realised that my ajax request was not successfully retrieving any data from the API. I tried modifying the request but will now need to work out how to store the token in order to properly authenticate a user. I found a npm dependency called axios which abstracts away an ajax request so that it is one command to open and send an ajax request, so I will try to use that to configure the authentication.

I also finished building the "edit user settings" form and page which should allow users to edit their details after they logged in. Besides fine-tuning the ajax request to correctly retrieve and store the token I will also spend the next sprint working on building the "Patient Cases" page and working on some of the styling of the webapp so that it looks more like the peach design guide.