AR-Assisted Medical Training:

Tutorial Generation & Eye Gaze Tracking Analysis

Team members: Prateek Bhatnagar and Allan Wang
Mentors: Ehsan Azimi, Chien-Ming Huang, Peter Kazanzides, Nassir Navab, and Camilo Molina
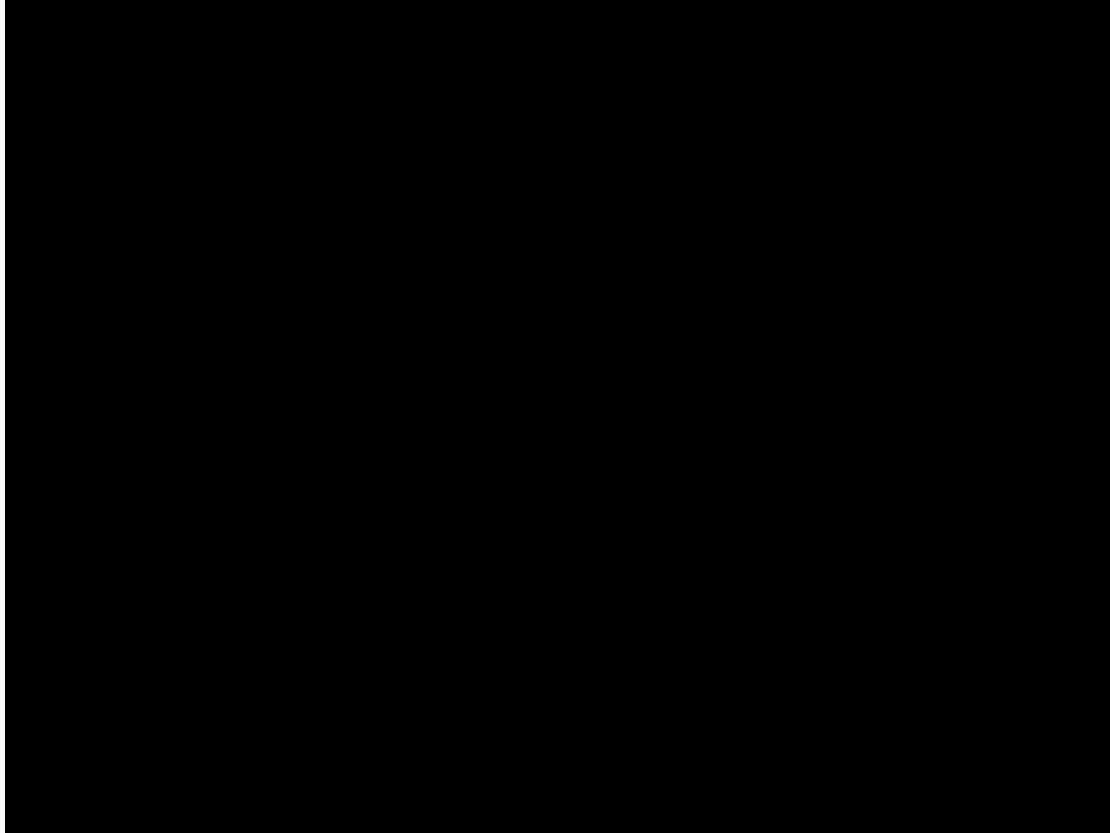
Review of Project and Goals

- Create a software tool to facilitate the **semi-automated creation** of medical training tutorials
  - Capture speech and visual data
  - Provide a user-friendly AR interface to generate JSON training files
- Use **eye gaze tracking data** to facilitate performance analysis
  - Facilitate image capture for tutorials
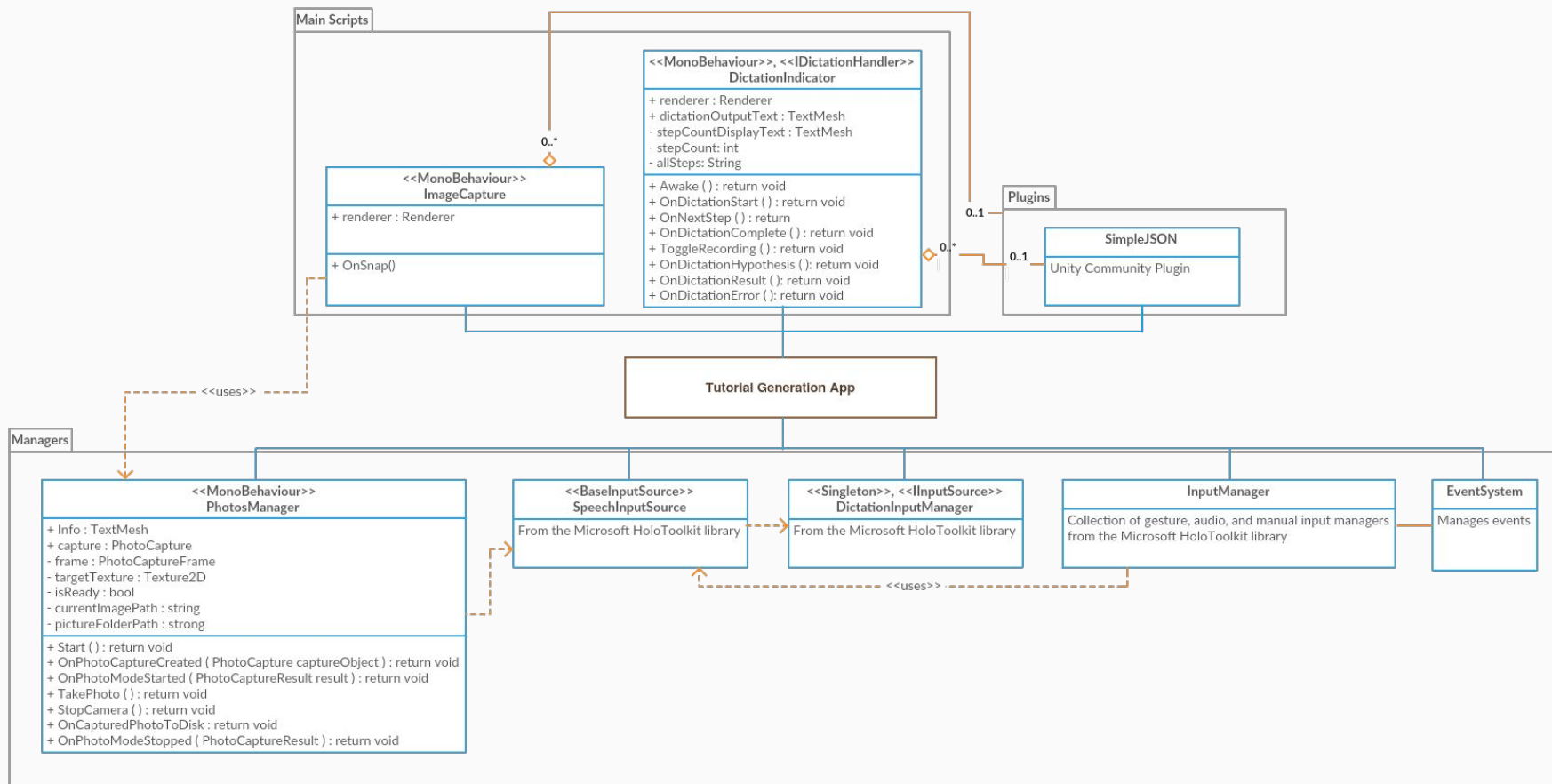 - Generate heatmaps during training
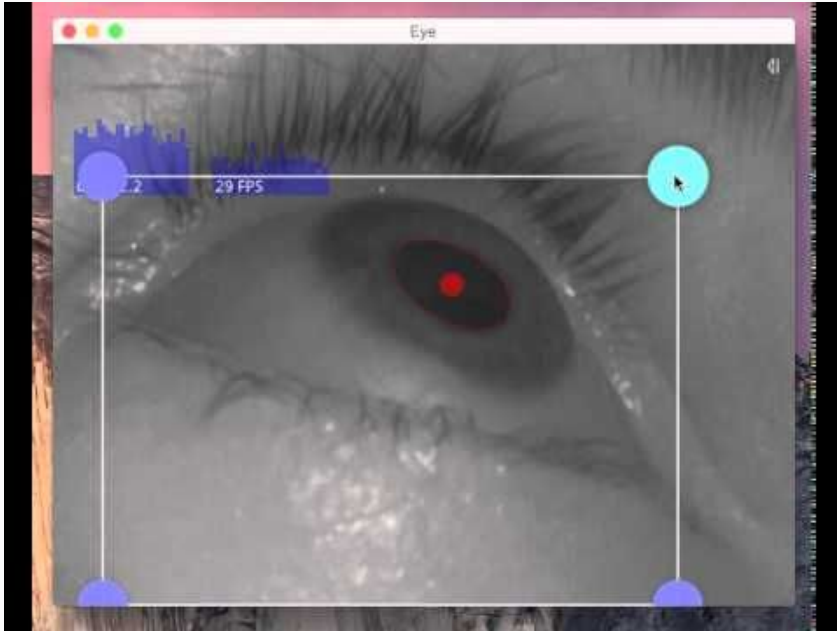
Current State: Tutorial Generation
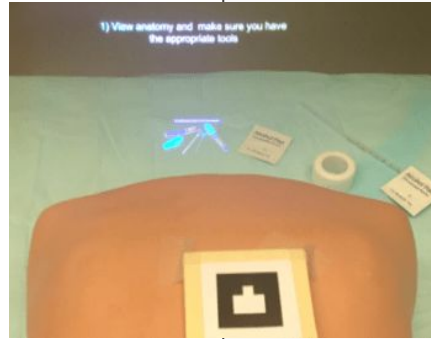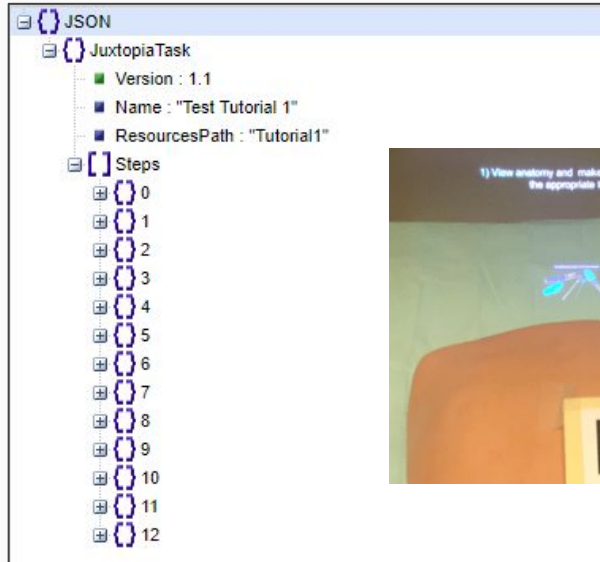
- Voice commands implemented
  - "Next," "Snap," "Start recording"
- Dictation implemented
  - Saving to JSON files
- Images can be saved

Software Structure

Main Scripts

<<MonoBehaviour>>, <<IDictationHandler>>
DictationIndicator

+ renderer : Renderer
+ dictationOutputText : TextMesh
- stepCountDisplayText : TextMesh
- stepCount: int
- allSteps: String

+ Awake ( ) : return void
+ OnDictationStart ( ) : return void
+ OnNextStep ( ) : return
+ OnDictationComplete ( ) : return void
+ ToggleRecording ( ) : return void
+ OnDictationHypothesis ( ): return void
+ OnDictationResult ( ): return void
+ OnDictationError ( ): return void

0..*

<<MonoBehaviour>>
ImageCapture

+ renderer : Renderer

+ OnSnap()

Plugins

0..1

0..*

0..1

SimpleJSON

Unity Community Plugin

Tutorial Generation App

<<uses>>

Managers

<<MonoBehaviour>>
PhotosManager

+ Info : TextMesh
+ capture : PhotoCapture
- frame : PhotoCaptureFrame
- targetTexture : Texture2D
- isReady : bool
- currentImagePath : string
- pictureFolderPath : strong

+ Start ( ) : return void
+ OnPhotoCaptureCreated (PhotoCapture captureObject) : return void
+ OnPhotoModeStarted (PhotoCaptureResult result) : return void
+ TakePhoto ( ) : return void
+ StopCamera ( ) : return void
+ OnCapturedPhotoToDisk : return void
+ OnPhotoModeStopped (PhotoCaptureResult) : return void

<<BaseInputSource>>
SpeechInputSource

From the Microsoft HoloToolkit library

<<Singleton>>, <<IInputSource>>
DictationInputManager

From the Microsoft HoloToolkit library

InputManager

Collection of gesture, audio, and manual input managers from the Microsoft HoloToolkit library
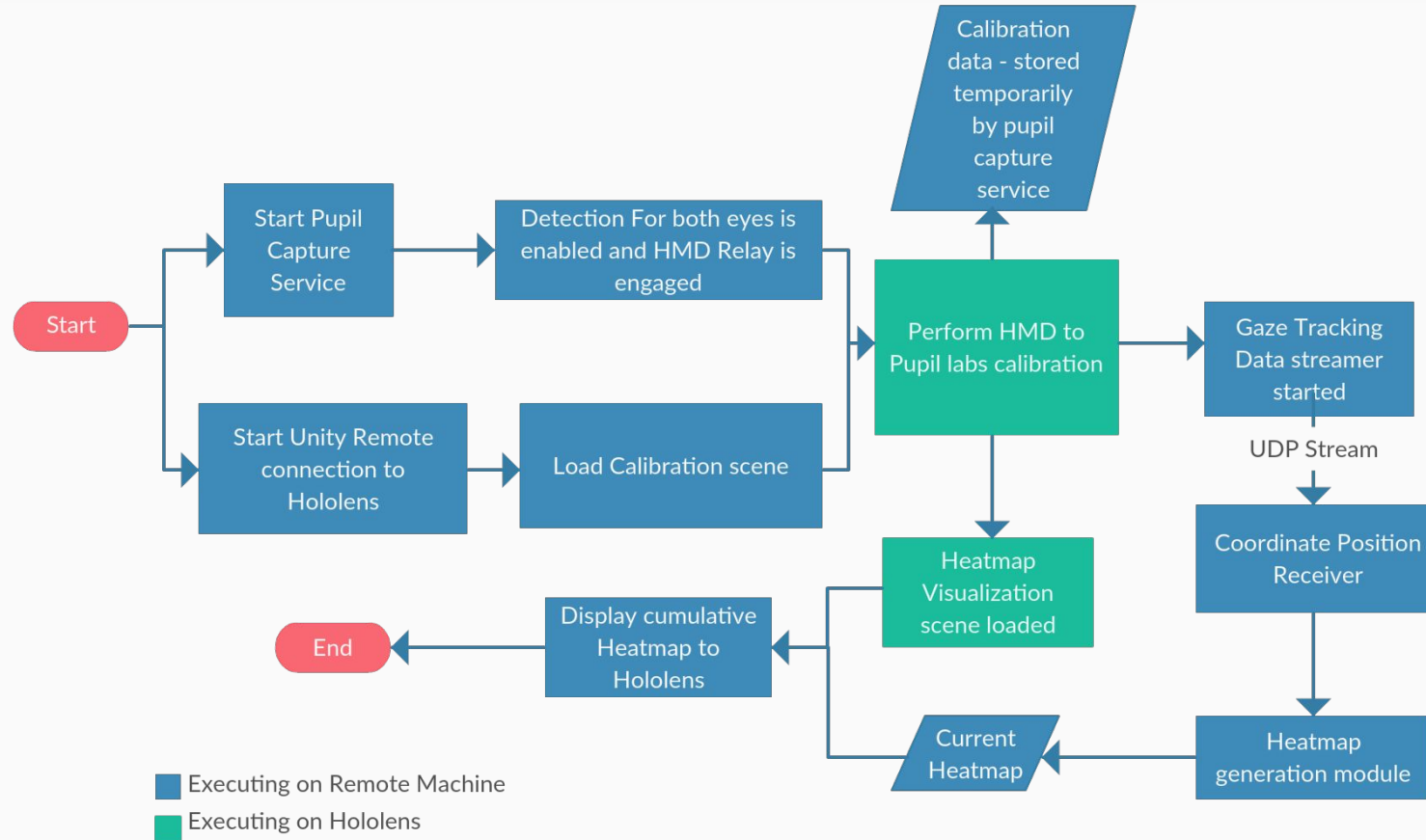
EventSystem

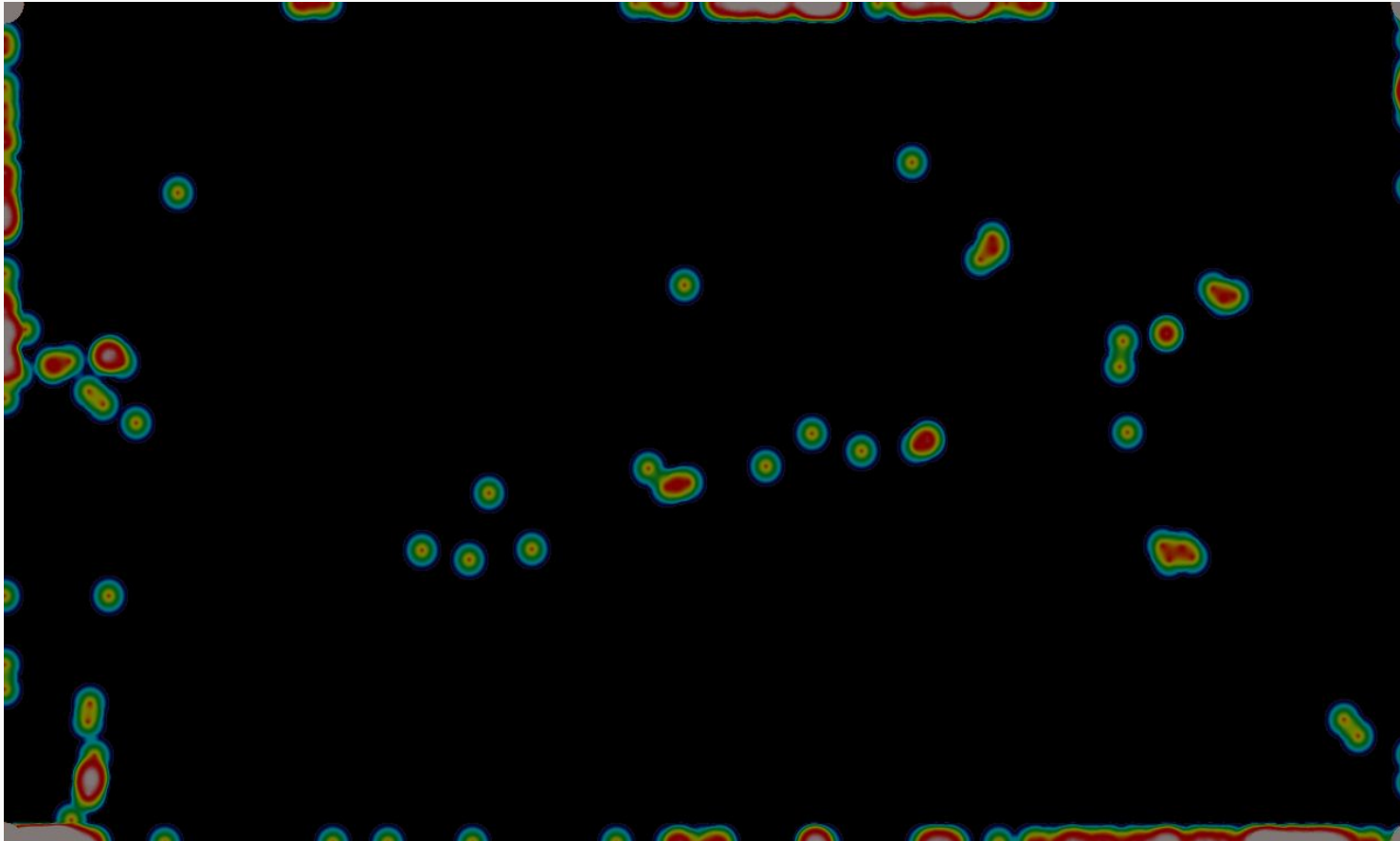Manages events

<<uses>>

Current State: Gaze Tracking

- Gaze Tracking works!
  - Gaze tracking calibration works
  - The Capture service is able to stream the tracked eye coordinates
- The Unity application is able to
  - Display a generated Heatmap
  - As well as aid with calibration
- The current workflow :
  - Pro - It works
  - Con - Its very inefficient
  - Requires a small overhaul

Control Flow for Heatmap

Start → Start Pupil Capture Service → Detection For both eyes is enabled and HMD Relay is engaged

Start → Start Unity Remote connection to Hololens → Load Calibration scene

Perform HMD to Pupil labs calibration → Calibration data - stored temporarily by pupil capture service

Perform HMD to Pupil labs calibration → Gaze Tracking Data streamer started

Gaze Tracking Data streamer started → UDP Stream → Coordinate Position Receiver → Heatmap generation module → Current Heatmap

Perform HMD to Pupil labs calibration → Heatmap Visualization scene loaded

Heatmap Visualization scene loaded → Display cumulative Heatmap to Hololens → End

Executing on Remote Machine
Executing on Hololens

Black is set to be Transparent for the Hololens

Documentation

- Currently using Github to track changes
- Code has been commented
- OneDrive Used as a secondary backup

```
/// <summary>
/// Beginning dictation on voice command
/// </summary>
public void OnDictationStart() {
    renderer.material.color = Color.red;
    dictationOutputText.color = Color.red;
    ToggleRecording(); // toggle dictation recording
}

/// <summary>
/// Begin next step. Reset text and increment counter.
/// </summary>
public void OnNextStep() {
    stepCount++; //increment step counter
    stepCountDisplayText.text = "Step " + stepCount.ToString(); // update step number
    dictationOutputText.text = "Say, \"Start recording\" to record text."; // reset instructional text
    cameraStatusText.text = "Camera ready"; // reset camera status
    renderer.material.color = Color.white;
    dictationOutputText.color = Color.white;
}
```

Progress

- **Minimum**
  - Working demo of tutorial editor ✔
    - *Speech-to-text* ✔
    - Generation of *2D heatmap* of gaze ✔
- **Expected**
  - Working demo of tutorial editor
    - Speech-to-text ✔
    - *Image capture, **In Progress***
  - Generation of 2D and 3D heatmap of of gaze
    - ***In Progress***

- **Maximum**
  - Working demo of tutorial editor
    - Speech-to-text
    - Image capture
    - *Marker creation*
    - *Expertise levels*
  - Using 3D and 2D gaze tracking heatmaps to optimize processes
  - Testing with ventriculostomy procedure under guidance of medical professional

Potential obstacles

- Tutorials
  - Displaying captured image to a texture in Unity to have a "live preview"
- Heatmaps
  - Interface between spatial heatmap and captured image
  - Streaming Heatmap related data to and from the hololens
  - Possible Inability to run python directly on the Hololens itself - there seems to be no 'EASY' way
- Recording Video off the Hololens

Upcoming Milestones

March 18	March 31	April 15	April 28	May 6
User Interface ✔	**Text-to-Speech & 2D Heatmaps ✔**	**Image Capture, Working Demo & Single User Trial**	**Marker Creation & 3D Heatmaps**	**Final Report & Demo**
• Able to accept voice commands • Synchronizing wIth video feed	• Able to generate text-based tutorials • Gaze-tracking implemented with 2D heatmaps	• Tutorials include both text and images • Have neurosurgeon create training module	• Implement ability to create virtual markers • 3D heatmaps added	• Have demo ready for live demonstration • Complete final report and presentation

Immediate Goals

- Complete photo capture: Generated tutorials will now be able to match the manually created JSON training files
- Integrate gaze tracking as a parallel process with the generator app
- Optimise Workflow for gaze streaming implementation

Reading List

1. Evaluation of Optical See-Through Head-Mounted Displays in Training for Critical Care and Trauma.
2. Kato, H., & Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In Proceedings of the 2Nd IEEE and ACM International Workshop on Augmented Reality (p. 85--). Washington, DC, USA: IEEE Computer Society. Retrieved from http://dl.acm.org/citation.cfm?id=857202.858134
3. Birt, J., Cowling, M., & Moore, E. (2015). Augmenting distance education skills development in paramedic science through mixed media visualisation.
4. Armstrong, D. G., Rankin, T. M., Giovinco, N. A., Mills, J. L., & Matsuoka, Y. (2014). A heads-up display for diabetic limb salvage surgery: a view through the google looking glass. Journal of Diabetes Science and Technology, 8(5), 951–6. https://doi.org/10.1177/1932296814535561
5. Tai, B. L., Rooney, D., Stephenson, F., Liao, P.-S., Sagher, O., Shih, A. J., & Savastano, L. E. (2015). Development of a 3D-printed external ventricular drain placement simulator: technical note. Journal of Neurosurgery, 123(4), 1070–6. https://doi.org/10.3171/2014.12.JNS141867
6. Atkins, M. S., Tien, G., Khan, R. S. A., Meneghetti, A., & Zheng, B. (2013). What do surgeons see: capturing and synchronizing eye gaze for surgery applications. Surgical Innovation, 20(3), 241–8. https://doi.org/10.1177/1553350612449075
7. Kersten-Oertel, M., Jannin, P., & Collins, D. L. (2012). DVV: a taxonomy for mixed reality visualization in image guided surgery. IEEE Transactions on Visualization and Computer Graphics, 18(2), 332–52. https://doi.org/10.1109/TVCG.2011.50
8. Eck, U., Stefan, P., Laga, H., Sandor, C., Fallavollita, P., & Navab, N. (2016). Exploring Visuo-Haptic Augmented Reality User Interfaces for Stereo-Tactic Neurosurgery Planning. In G. Zheng, H. Liao, P. Jannin, P. Cattin, & S.-L. Lee (Eds.), Medical Imaging and Augmented Reality (pp. 208–220). Cham: Springer International Publishing.