

Holographic Brain Project

Technical Documentation

*Garage Internship Winter 2017*

[Master Deck](https://microsoft.sharepoint.com/teams/givw17/_layouts/15/guestaccess.aspx?guestaccesstoken=VMFSDIi02PO3AdMumQoQNcyEyc4lZrs9K3Fd9OeJbY4%3d&docid=2_165047e74023643f99522627441d77afb&rev=1)

Interns **UBC Researchers**

Kevin Jacyna Claudia Krebs

*Developer* [ckrebs@mail.ubc.ca](mailto:ckrebs@mail.ubc.ca)

Reem Razak Parker Holman

*Developer* [parker.holman@gmail.com](mailto:parker.holman@gmail.com)

Roxanne Plante Tamara Bodnar

*Developer* [tammybodnar@gmail.com](mailto:tammybodnar@gmail.com)

Salma Aly

*Developer*

Sean Tohidi

*Developer*

Shirley Mao

*Designer*

PM Contact

Stephane Morichere-Matte

*Senior Program Manager*

*Garage Internship Vancouver*

[stmorich@microsoft.com](mailto:crellis@microsoft.com)

Sponsors  
  
**BigPark**

Craig Ellis

*Principal Group Program Manager*

[crellis@microsoft.com](mailto:crellis@microsoft.com)

Peter Doidge-Harrison

*Principal Software Engineer Manager*

[pdoidge@microsoft.com](mailto:pdoidge@microsoft.com)

Table of Co­ntents

[Problem Definition 4](#_Toc481076505)

[Solution Definition 4](#_Toc481076506)

[Target Audience 4](#_Toc481076507)

[Installation 4](#_Toc481076508)

[Architecture 6](#_Toc481076509)

[How to Start/Restart the App in the HoloLens 9](#_Toc481076510)

[How to Use the App 10](#_Toc481076511)

[Looking Forward 11](#_Toc481076512)

[Known Bug 14](#_Toc481076513)

[Tips 15](#_Toc481076514)

[Key Learnings 15](#_Toc481076515)

# Problem Definition

UBC researchers understand that most neuroanatomy students are visual learners, which makes it difficult for them to translate abstract 2D diagrams and images (like MRI scans and brain slices) into 3D structures in their minds. While plastic brain models are more useful than conceptual diagrams, students still struggle because the context within the brain is lost once the model is taken apart.

# Solution Definition

By offering various interactions with 3D models of a real brain, the goal is to provide a new way to view the brain that would benefit not only students, but even established neuroscientists. Researchers can use the app to see the brain as they have always imagined it, whereas students can improve their understanding of the spatial relationships between brain structures and solidify the concepts they learn in class.

# Target Audience

* Professors teaching neuroscience and neuroanatomy
* Neuroscience and medical students learning about the anatomy of the brain

# Installation

1. Downloads and installs (mostly following the checklist found at <https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools>)
   1. Download and install Visual Studio 2015 (<https://my.visualstudio.com/Downloads?pid=2084>)
   2. Download and install the Windows 10 SDK (<https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk>)
   3. Download and install the HoloLens emulator (<http://go.microsoft.com/fwlink/?LinkID=823018>)
   4. Download and install Unity 5.5 (<https://store.unity.com/download>)
2. System recommendations (mostly for the HoloLens emulator)
   1. 64-bit Windows 10 operating system
   2. 64-bit CPU with 4 cores
   3. 8 GB of RAM
   4. The BIOS must have the following features enabled:
   5. Hardware-assisted virtualization
   6. Second Level Address Translation (SLAT)
   7. Hardware-based data execution prevention (DEP)
   8. The GPU must have DirectX 11.0 or higher as well as WDDM 1.2 driver or later
3. Opening the project and configuring the build settings (mostly following the tutorial at <https://developer.microsoft.com/en-us/windows/mixed-reality/holograms_100>)
   1. Clone the Dev branch of the repository from VSTS (<https://microsoftgarage.visualstudio.com/GIV.W17/HoloBrain/_git/HoloBrain#version=GBDev&_a=contents> )
   2. To get the project file, launch Unity, choose "Open" and open the "Source" folder of the repository
   3. Select Edit > Project Settings > Quality; select the dropdown under the Windows Store logo and select Fastest. **If the Windows Store logo is not there, go through steps 5.a and 5.b to download the Windows Universal 10 SDK.**
   4. Select Edit > Project Settings > Player; in the Inspector panel click on the Windows Store tab; expand the Other Settings group; in the Rendering section, check the Virtual Reality Supported checkbox to add Windows Holographic as a supported virtual reality SDK. **If the Windows Store logo is not there, go through steps 5.a and 5.b to download the Windows Universal 10 SDK.**
   5. In the same tab, expand the Publishing Settings group; in the Capabilities section, check the Microphone and SpatialPerception checkboxes
4. Configuring the IP address of the sharing service
   1. Open the HoloBrain scene. Click on the Sharing game object in the hierarchy window to select it. In the inspector, change the Server Address field to the IP address of the VM hosting the sharing service. Repeat this process for the EnterSessionIDScene scene. **It is crucial to change the IP address for the Sharing game object in both of these scenes, and they must match for sharing to work correctly.**
5. Building and deploying the project
   1. Select File > Build Settings; In the Platform grouping, choose Windows Store. On the right-hand side of the window, choose Universal 10 as the SDK, HoloLens as the target device, D3D as the UWP build type, Build and Run On Local Machine, and check the Unity C# projects checkbox.
   2. When choosing the platform, if Windows Universal 10 SDK does not appear, click on the download button that will show in unity.
   3. Click "Build" and then create a new folder to hold the build of the project (e.g. VSBuild). **Sometimes errors will occur when building the app in Visual Studio if VSBuild is put in the wrong place. Make sure to create this folder at "HoloBrain/Build/VSBuild".**
   4. Open this folder in the File Explorer and open the Visual Studio solution
   5. In Visual Studio set the architecture to x86 and the device to HoloLens emulator.
   6. If debugging, set the mode to Debug and select Debug > Start Debugging to launch the app. If not debugging, set the mode to Release and select Debug > Start Without Debugging to launch the app.

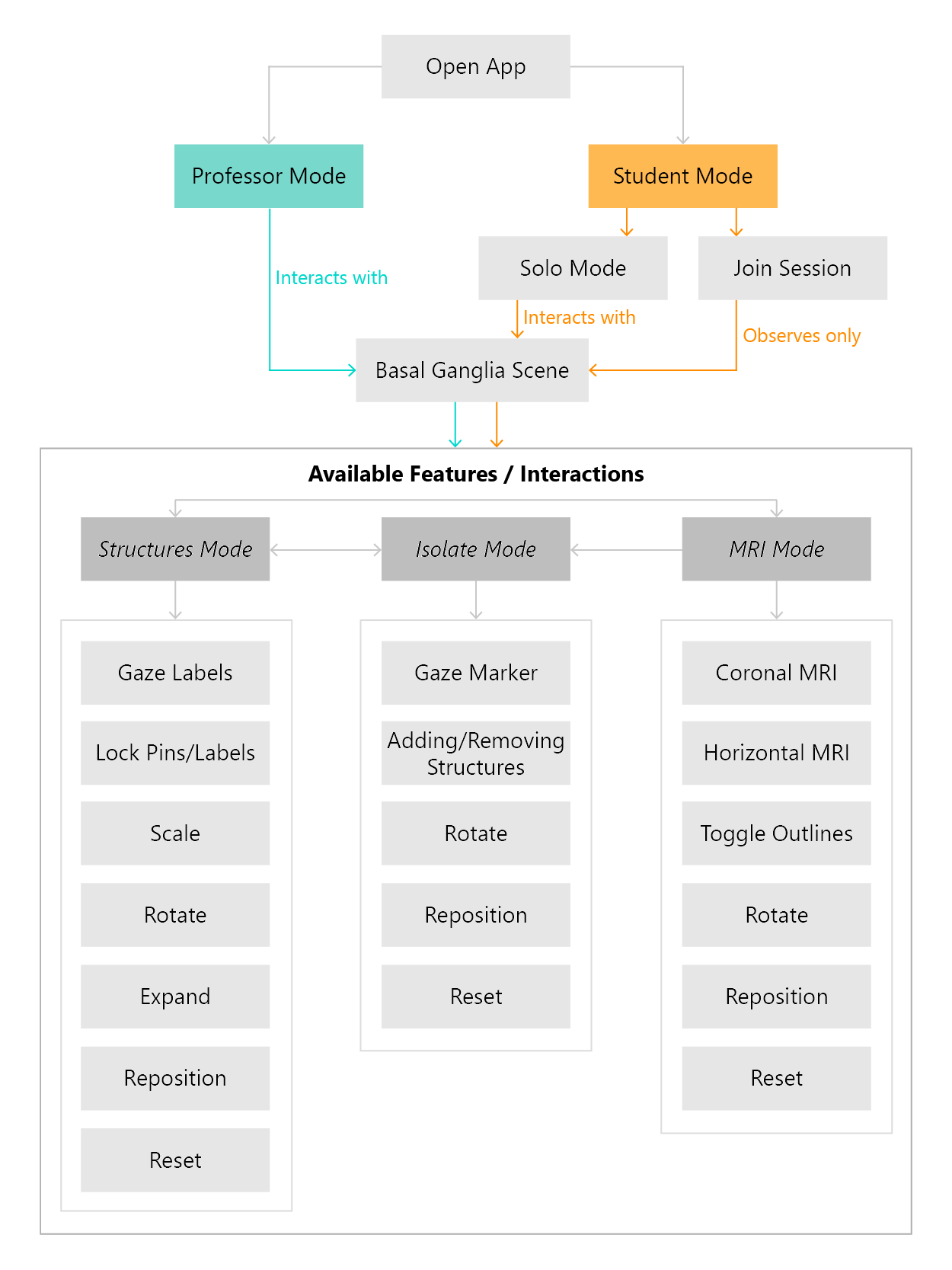
# Architecture

* Different Unity Scenes
  + StartAppScene, allows user to choose between student and professor mode
    - The UIManager game object is the parent of the student and professor buttons, and text, and this pattern is followed in both of the scenes below
    - ButtonAction - sends the button name that was tapped to the ModeSelection scripts OnTap method
    - ModeSelection - moves the app into the desired mode chosen by the user. The current app mode is held in PlayerPref.GetString("mode") and is accessible in every scene of the application
  + StudentOrSolo, scene is loaded if user selects Student in StartAppScene, user can opt to join a session as student or enter solo mode
    - ButtonAction - defined above
    - ModeSelection - handles the join session, solo, and back buttons. The WriteAdjustmentAngle method is used to ensure that when a new scene loads it is centered at the same location as the previous scene
    - The ModeSelection object is disabled in this scene, as it is not destroyed in StartAppScene, and can therefore still be accessed without creating a new instance of it
  + EnterSessionId, loads when user wishes to enter an existing session
    - DigitAction - sends the digit name that was tapped to the DigitHandler script
    - DigitHandler - is responsible for creating a session id from user input, checking to ensure the session exists, and then entering the Holobrain scene as a student
    - The sharing object is in this scene in order to check if the session exists before loading the Holobrain scene
* Hierarchy of the Unity Game Objects in the HoloBrain scene
  + The sharing service, HoloLensCamera, Lights, Cursor, LabelDisplay, LoadingScreen, VoiceControl, and ModeControl game objects are all in the top level of the hierarchy.
  + The HologramCollection game object in the top level contains the Brain game object and the ControlsUI game objects. The Brain game object contains the MinimapPositionObject, the pins used for labeling structures, the MRICollection used for MRI mode, the BrainUI, and the BrainParts game objects.
  + The BrainUI is separated from the ControlsUI because the ControlsUI can be pinned while the BrainUI must be placed relative to the brain.
  + The BrainParts game object contains all the structures of the brain. Each of the structures has the HighlightAndLabelCommands script to handle changing its colour when highlighted or locked as well as activating its pin when locked. The BrainParts game object has all of the scripts used for interactions.
* BrainParts interactions scripts
  + The isolate structures script is used to move the brain to isolated mode and back. Entering isolated mode moved the entire brain towards the position of the MinimapPositionObject while cloning the ventricles and enlarging them. When other structures are isolated, clones of them are created and then moved to the isolated position and size. Concluding isolate moves all the structures back to the original position and destroys the clones.
  + The RotateStructures script is used to turn on/off the auto-rotate feature. If in the rotation state, the scripts either rotates the isolated structures (while in isolated mode) or the BrainParts game object.
  + The ExplodingCommands script is used to turn on/off exploded mode. Exploded mode works by moving every bilateral structure away from the center of the brain, proportionally to how far they are from the center initially.
  + The ScaleToggler script is used to change the scale of the BrainParts game object.
  + The StateAccessor script is used by other scripts to check the state of the brain (isolated, in MRI mode, able to take an interaction, etc.)
  + The ColoursAccessor script is used to turn on/off all the locked colours for the brain structures
* Scripts on other game objects
  + The Brain game object has the HologramPlacement script, which handles repositioning the brain. When in a shared session, the anchor for the brain is set when the brain is repositioned.
  + The HologramCollection game object contains most of the scripts involved with sharing (ImportExportAnchorManager, AppStateManager, RemotePlayerManager, LocalPlayerManager) as well as the gaze manager, gesture manager, and gaze stabilizer scripts.
  + The LabelDisplay game object has the LabelDisplayManager script, which updates the label when the user gazes at a structure
  + The VoiceControl script on the VoiceControl game object takes in all the voice commands and handles them
  + The ModeControl script on the ModeControl game object changes the scene depending on which mode is selected (solo, student, or professor)
* States
  + There are 3 main states. The default Structures state, the isolated state, and the MRI state. The user can switch between the states using voice commands or buttons on the controls UI.
* Sharing component and VM
  + The HoloToolkit library was used to handle sharing between the professor and the students. A sharing service executable is running on a VM that the devices connect to. When someone launches the app in professor mode, a session is generated automatically with a randomly-chosen number as the name. Students can join the session using that session number.
  + When the professor interacts with the brain, a message is sent using the CustomMessages script attached to the sharing service. Messages are received and handled by each script where appropriate.
  + All the VM is used for is running an executable file that receives and passes on messages from the users. The IP address of the VM must be entered in the Unity editor for two different scenes, as specified in part 4 of the installation instructions (configuring the IP address of the sharing service)
* User Interface
  + There are three control user interfaces, one for each of the game modes. The control UI is placed below the level of the HoloLens. The current control UI is set by the current mode. The control UI follows the user around by default, but can be pinned in place. A session UI persists near the brain and indicates both the session ID as well as the number of people in the session. The session ID cannot be seen in student mode.

# How to Start/Restart the App in the HoloLens

1. How to find the application in the HoloLens?
   1. The application is called "Holographic Brain Project". If the project does not show up in the main page of the HoloLens menu, airtap on the '+' button and select it from there.
2. What to do to exit the application?
   1. To exit the application, simply bloom. This bloom gesture brings the user back to the main holographic world, with the Holographic Brain Project flat app present. Closing this flat app closes the application.
3. How to go back after blooming?
   1. After blooming, locate the Holographic Brain Project flat app and airtap on it to enter the application again.

# How to Use the App



1. How to start the app
   1. When using the application, the HoloLenses must have an internet connection.
   2. Always spatial map the room before starting the application. Instructions on how to spatial map can be found in the Spatial Mapping section.
   3. To start the application, open the HoloLens menu by blooming then find the "Holographic Brain Project" application. AirTap on it, then place the flat app by airtapping again.
2. For a solo session:
   1. When the application opens, choose the "Student" option, then airtap on "Solo". This will start a session that no one else can join. Air tap to place the brain after loading.
3. For a student/professor session:
   1. Start the application on the HoloLens with all session attendees standing next to each other. The attendees should be facing the same way throughout the login process.
      1. The person leading the session should open the app first and choose the "Professor" option. When the brain appears after loading, the user should airtap to place the brain where they want it, then take note of the session number indicated in the grey box on the left of the brain. The students will need this number to join the session.
      2. The other attendees must choose the "Student" option when opening the app, then they must choose "Join Session". To join the right session, the students must enter the session number the professor sees.

# Looking Forward

Due to the short nature of the Garage program, there were many features that we were not able to implement. Below, we have listed the features in order of highest to lowest priority.

**Brain Slices**

If we had the time, we would have liked to implement a mode for brain slices. Like the MRI scans feature, we would have allowed the user to choose from different coronal and horizontal slices by tapping on different slices of the cortex. Once selected, a photo of a brain slice would show up.

**Gaze Marker in Isolate Mode**

Because of sharing issues, we were not able to include the gaze marker in the final product. At the moment, the only way the gaze marker will share properly is if the app is deployed from exactly the same position in the real world. It works in Professor and Solo mode, but since the marker doesn’t show up in the same place in a shared session, we chose not to include it in the final build. However, the code is still in the scripts should UBC choose to work on this feature.

Please refer to the GazeMarkerManager script for instructions regarding how to add this feature.

**Pathways between different structures**

Traditionally, the pathways between the structures of the basal ganglia are taught using abstract 2D diagrams that are difficult to comprehend. We would have liked to display these pathways using 3D arrows and/or animations in conjunction with the basal ganglia models.

**Comparing healthy and diseased brains**

If we had implemented a feature that displayed the direct and indirect pathways between structures, we would have enhanced it by offering the ability to turn pathways on or off to simulate specific brain diseases, like Parkinson’s.

**Blood Vessels & Internal Capsule Models**

Given the fact that we had other higher priority features and not enough time to reduce the polycount of additional models, we were not able to incorporate models of the basal ganglia’s blood vessels and the internal capsule. Since these are structures that are also difficult to see using traditional teaching methods, displaying them in the HoloLens would further aid students’ understanding of the basal ganglia.

**Additional brain subsections**



In the future, this app should be able to import other subsections besides the basal ganglia, such as the hypothalamus and cerebellum. This would be a task for UBC should they choose to assign developers to continue this project.

**Rotation presets**





Besides auto-rotate, the ability to choose between different views of the brain would have been useful. Aerial view would orient the top of the brain towards the user in control of the application, while Bottom view would make the bottom of the brain face the user. Bottom-Up would flip the brain upside down so that the bottom would be facing the ceiling, which is how students usually examine brain specimens in a lab.

**Professor’s ability to allow students to interact with the brain**

Currently, only users who create and lead a session (like professors/instructors) can interact with the brain through voice commands and the UI. Users who join sessions as students can only gaze at structures and see their labels. Ideally, users who are leading a session should have the option to give the other users in the session the ability to use all interactions.

**High Contrast colour preset**

The colours of the UI and models have not been tested to see if they are easily discernible for colourblind people. For a more inclusive design, different colour presets for the UI and models, such as Default and High Contrast, should be available.

**Settings Menu**

To allow for customization, there should be a Settings menu with the following options:

1. Ability to auto-save sessions when using the app
2. Adjust rotation speed
3. Adjust letter spacing
4. Turn on/off voice command cues
5. Default and High Contrast colour presets
6. Muting sound effects

# Known Bug

**Fixing audio feedback when choosing a mode**

In the scene where the user chooses between the Professor or Student modes, the air-tap sound effect is cut off before it has a chance to fully play. This was a low-priority bug, so given the time we had left, we chose to fix other more important issues.

# Tips

* Links to Holo Academy:
  + <https://developer.microsoft.com/en-us/windows/mixed-reality/academy>
  + <https://developer.microsoft.com/en-us/windows/mixed-reality/holograms_240>
  + This is a good shared application resource
* Dealing with Unity
  + If you are experiencing strange errors, or a build is not deploying, often deleting your build folder and then creating a new one to build in will work
  + Making changes in playmode (entered by pressing the play button) are not saved and will be lost when playmode is stopped
* Dealing with HoloLens
  + If Cortana begins speaking during a session, waiting and saying ‘no’ very firmly should stop her
  + Make sure your Hololens has not gone to sleep when deploying
* Voice commands
  + If you’re having trouble using voice commands, enunciate syllables and consonants as clearly as possible

# Key Learnings

**Spatial Mapping**

* The HoloLens uses its surroundings to determine the position of the holograms.
* Always spatial map the room thoroughly before starting the application:
  + Spatial mapping is done by navigating to the main menu of the HoloLens, and then making a bloom gesture. Then start tapping on objects in the room, you should see red mesh lines forming around the object and spreading out across the space. Continue doing so the until meshes you see accurately represent your space
  + Make sure to have all attendees spatial map before the start of the lesson at the same time and in the same sequence.

**Performance**

* Performance refers to the stability of the holograms in the surroundings. If the holograms feel jittery, it might result in a feeling of discomfort for the user.
* For future developers for the project:
  + Always Maintain 60FPS frame rate.
  + To measure the frame rate of the application, use the HoloLens web portal: <https://docs.microsoft.com/en-us/windows/uwp/debug-test-perf/device-portal-hololens>
  + Some Factors that Affect Performance:
    - Shaders:
    - The application currently uses HoloToolKit/FastConfigurable shaders on the game objects in unity. If future developers of the application decide to use different shaders it is recommended to measure the effect on the frame rate before making the changes.
* Objects Polygon Count:
  + - When adding new 3D models to the unity scene, make sure the models are of minimal polygon count. More than 30K polygons in the field of vision of might result in a decrease in the frame rate.
* For more information about performance visit the following links:
  + <https://developer.microsoft.com/en-us/windows/mixed-reality/performance_recommendations_for_unity>
  + <https://developer.microsoft.com/en-us/windows/mixed-reality/performance_recommendations_for_hololens_apps>