Structural Analysis App
Step-by-Step Guide

1. I used the tutorials found here to get started with the Microsoft HoloLens: https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-100.

2. Open Unity version **2019.1.6f1**; select **New**; select **2019.1.6f1** under **Unity Version**; select **3D** under **Template**; select **Create project**.

3. Right click **Untitled** under the **Hierarchy** panel; select **Save Scene As**; name the scene.

4. Select the **Main Camera** under the **Hierarchy** panel; change **Position** under **Transform** in the **Inspection** panel to **0, 0, 0** for x, y, z values; change **Clear Flags** to **Solid Color**; change **Background** to **0, 0, 0** for R, G, B values; change **Clipping Planes -> Near** from 0.3 to **0.85**.

5. Go to **Edit -> Project Settings -> Quality**; change **Levels** under the blue **Universal Windows Platform** icon to **Very Low**; close window.

6. Go to **Edit -> Project Settings -> Player**; select the blue **Universal Windows Platform** icon; under **Publishing Settings**, scroll down to **Capabilities** and select **SpatialPerception**; under **XR Settings**, select **Virtual Reality Supported**; click the **+.** sign and select **Windows Mixed Reality**; for **Depth Format**, choose **16-bit depth**; close window.

7. Go to **File -> Build Settings**; select **Add Open Scenes** under **Scenes in Build**; select **Universal Windows Platform** under **Platform**; select **Switch Platform**; select **HoloLens** for **Target Device**; select **x86** for **Architecture**; select **D3D Project** for **Build Type**; select **Latest installed** for **Target SDK Version**; select **10.0.10240.0** for **Minimum Platform Version**; select **Latest installed** for **Visual Studio Version**; select **Release** for **Build configuration**; close window.

8. Steps 4-8 ensure that holograms appear in the physical environment when deployed from the HoloLens, rather than trapped in a 2D window, unable to be interacted with.

9. Download the **HoloToolKit** version **2017.4.0.0** from https://github.com/microsoft/MixedRealityToolkit-Unity/releases?after=2017.4.2.0 onto the desktop.

10. In Unity, go to **Assets -> Import Package -> Custom Package**; select **HoloToolKit-Unity-2017.4.0.0** from the desktop; click **Open**; in the window that appears, select **Import**.

11. Delete the three **Editor** folders that create errors whenever you push the play arrow at the top of Unity (Unity will indicate exactly which folders are causing the errors; all you have to do is delete those folders until the errors are resolved).

12. I used the 2D sketch I was given to create the 3D structure in Unity. I used basic 3D objects (i.e. Cubes and Cylinders) by right clicking under the **Hierarchy** panel, selecting **3D Object**, and configuring the objects to create the overall structure. For the triangle bases, I used thin Cubes to make a triangle shape. I converted inches and feet to meters, as **1 Unity unit is equivalent to 1 meter**. Please see the StructuralAnalysisApp project file and explore exactly how the structure is put together!

13. I used this tutorial on making the entire holographic structure moveable, scalable, and rotatable on the Microsoft HoloLens, using the prefabs and C# scripts available in the HoloToolKit: https://codeholo.com/2018/06/19/twohandmanipulatable-part-2-of-moving-scaling-rotating-objects/.

14. I used the **UITextSelawik** prefab from the HoloToolKit and lines I created from 3D Capsule objects to display information on the geometric features of the structure. The **ToolTip** prefab from the HoloToolkit does the same thing automatically, and the **ToolTipSpawner.cs** script even allows users to tap a section of a hologram and have textual information displayed at runtime. However, my use of tooltips was not successful, and resulted in the generation of a magenta quad visible through the HoloLens that I was unable to get rid of. The closest I came to getting rid of the quad was by looking at the **StabilizationPlaneModifier.cs** script in the **InputManager** prefab. However, commenting out or editing the function **OnDrawGizmos()** on lines 266 to 283 of the script, responsible for drawing the quad, did nothing to resolve the issue, and I was unable to find a solution online.

15. When you are ready to deploy to the HoloLens, go to **File -> Build Settings**; select **Build**; create a new folder in the popup window called **App**; choose **Select Folder**, which will save the Visual Studio project to it.

16. Open the **App** folder in the new window that appears; select the **.sln** file with your project name; in Visual Studio, make sure the following are selected in the toolbar: **Release**, **x86**, and **Remote Machine**; go to **Project -> Properties -> Debugging**; input the IP address of the HoloLens in the field **Machine Name**; select **OK** to close the window; turn on the HoloLens; in Visual Studio, go to **Debug -> Start Without Debugging** to deploy the app to the HoloLens; look at an open space in the room to see your app run; explore the generated hologram!

Structural Analysis App
Documentation

Project Goal

The app I aim to develop for my research project has two components. 1) Structural Analysis: allows users to bring up a moveable, scalable, and rotatable holographic structure for analysis, and see information in the form of text. 2) Structural Recognition: allows users to identify structures in the real environment, such as a street, sidewalk, building, or bridge, and view information on that structure in the form of text. The app will be a barebones contribution for future improvement and development, with simple functionality that can later be broadened.

Week 1

The first step of my research process was to get a general idea and background of augmented reality in civil engineering applications. I used the following resources to broaden my understanding and knowledge:

1. https://vimeo.com/showcase/5603313
2. http://onlinepubs.trb.org/onlinepubs/webinars/181205str.pdf
3. https://www.forbes.com/sites/quora/2018/02/02/the-difference-between-virtual-reality-augmented-reality-and-mixed-reality/#71ddd452d07c
4. https://rubygarage.org/blog/best-tools-for-building-augmented-reality-mobile-apps
5. https://www.researchgate.net/publication/250355839_Impact_of_Augmented_Reality_AR_in_Civil_Engineering

Week 2

The second step of my research process was to create holographic objects in Unity and see them through the Microsoft HoloLens. I used Windows Mixed Reality for development, and used the following tutorials to get started: https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-100. I was able to see the hologram of a simple cube I created following the MR Basics 100: Getting started with Unity tutorial. My next goal was to not only see a holographic object, but to interact with that object. I used the Origami tutorial from MR Basics 101: Complete project with device to create a hologram with spatial sound that I could walk around and interact with through gaze and gesture.

Week 3

The third step of my research process was to use the knowledge and skills I had learned from the previous two weeks to create my own project from scratch. To accomplish this, the goal was to learn how to use the HoloToolKit to create a moveable, resizable, and rotatable holographic

object using the following tutorial: https://codeholo.com/2018/06/19/twohandmanipulatable-part-2-of-moving-scaling-rotating-objects/. I used the above tutorial to create a holographic cube that could be moved, resized, and rotated. The next step was to add a tooltip to the holographiccube. I used the following article, https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Tooltip.html, to create a tooltip for the holographic cube.

I ran into a big problem earlier in the week deploying apps to the HoloLens in 3D. The problem was that my app deployed in a 2D window, barring me from interacting with the hologram. I used several HoloLens development help forums online and experimented until I found an answer. To get the MoveSizeRotate app to work: 1) check virtual reality supported and Windows Mixed Reality under Windows tab in XR Settings; 2) change to 16-bit depth buffer in XR Settings; 3) uncheck Windows Holographic in XR Settings; 4) check SpatialPerception under Other Settings; 5) make sure to have Release, x86, and latest version of Windows 10 in Build Settings.

Week 4 and Week 5

The fourth step of my research process was to create and test the structural analysis component of the app to be developed. First, I created a holographic structure in Unity, based on 2D sketches of an existing frame, using 3D objects. Since 1 Unity unit is equivalent to 1 meter, I made sure to convert the units of measurement from inches and feet to meters. The 3D structure is more of a conceptual representation of the 2D sketches, and is not as detailed as the real structure. However, it gives a good approximation of the scale and basic features of the actual structure. Second, I made the beam moveable, scalable, and rotatable, similar to the interactive cube demo and using the exact same tutorial. Third, I displayed information on the basic components of the structure with a text prefab from the HoloToolKit. The following measurements were used to create the structure in Unity:

“W” stands for “wide flange beam”
“pL” stands for “plate”

FRAME

W12x87 column height = 8 feet 5 inches (w/o 1 inch plate)
W12x87 column “d” = 12.53 inches
W12x87 column “tf” = 0.81 inches
W12x87 column “tw” thickness = 0.515 inches
W12x87 column “tw” width = 12.53 inches - (0.81 inches + 0.81 inches) = 10.91 inches
W12x87 column “bf” = 12.125 inches

W12x96 beam length = 96 inches - (2*6.265 inches + 2*1.375 inches) = 80.72 inches
W12x96 beam “d” = 12.71 inches
W12x96 beam “tf” = 0.9 inches

W12x96 beam “tw” thickness = 0.55 inches
W12x96 beam “tw” width = 12.71 inches - (0.9 inches + 0.9 inches) = 10.91 inchesW12x96 beam “bf” = 12.16 inches

TOP CONNECTION

1’’ plate length = 2 feet 2 1/2 inches
1’’ plate width = 12.125 inches
1’’ plate height = 1 inch

7/8” stiffener plate length = 10.91 inches
7/8” stiffener plate width = 12.125 inches / 2 = 6.0625 inches
7/8” stiffener plate height = 7/8 inches

1/2” stiffener plate length = 10.91 inches
1/2” stiffener plate width = 12.16 inches / 2 = 6.08 inches
1/2” stiffener plate height = 1/2 inches

1 3/8” plate length = 1 foot 7 3/4 inches
1 3/8” plate width = 12.125 inches
1 3/8” plate height = 1 3/8 inches

1/2” plate length = 11.023622 inches
1/2” plate width = 12.125 inches
1/2” plate height = 1/2 inches

Bolt diameter = 1 1/8 inches

BASE CONNECTION

Base plate length = 3 feet
Base plate width = 16 inches
Base plate height = 1 1/2 inches

1/2” base triangle plate length = 3 feet
1/2” base triangle plate width = 6 inches
1/2” base triangle plate height = 1/2 inches

Floor anchor diameter = 2 5/8 inches

I used the following resources to get information on wide flange beams and bolts: https://www.engineersedge.com/standard_material/Steel_ibeam_properties.htm and https://www.portlandbolt.com/products/bolts/structural/.

Week 6 and Week 7

The fifth step of my research process was to research and experiment with the Unity physics engine, to see if deformation and vibration of the structure was possible.

Week 8 and Week 9

The sixth step of my research process was to explore object detection through the Microsoft HoloLens. I took pictures of 16 doors and followed this tutorial: https://docs.microsoft.com/en-us/windows/mixed-reality/mr-azure-310. The goal is for users to take pictures with the HoloLens using a tap gesture. The software will then send the picture to a cloud-based Microsoft Azure service, which will identify objects in the picture using machine learning. Unfortunately, due to build errors within Unity, I was unable to deploy the application to the HoloLens. Nevertheless, the steps I documented for deploying apps to the HoloLens still holds.

I also investigated importing photogrammetry files into Unity from the following source: https://docs.unity3d.com/Manual/. The accepted file formats are *.obj, *.3ds, *.dae, *.fbx, and *.dxf. There are no restrictions on file size. However, model components must be imported separately (for example, colors, textures, animations, etc.).

I completed Structural Analysis and got started on Structural Recognition.