Component Design

Team: Logic Legends
Members: Thomas Klein, Rohan Chandrasekar, and Mya Chappell

Project: Spatial Archive

System Component Overview

- Unity application (HoloLens)
    - Capture mesh data using mesh filters
    - Convert mesh to STL binary format
    - Communication module to send STL data over a wireless socket
- Server-side Applications
    - Receives STL binary files over wireless socket
    - Save the STL files locally
    - Mesh cleaning to remove artifacts and imperfections
    - Attaching timestamp to each mesh
- Client-side application (Terminal Interface or GUI)
    - User inputs for timestamps and mesh calls
        - Two modes: live streaming and calling previous meshes
    - Opens OpenGL visualization of called mesh
    - Mesh interactions
        - Mesh comparison at different points in time
        - Mesh segmentation
        - Classify segmented components

Component Description

Unity Application (HoloLens):

Assigned Members: Mya

The mesh filter data will be accessed through Microsoft MRTK spatial observers during HoloLens runtime. Mesh filter data will be converted to STL binary file by writing the normals and vertices of the mesh to a binary file. The binary files will be sent through our socket connection. This component will be a client socket. The Unity socket library will be used for creating our client (e.g., “System.Net.Sockets,” “System.Net.Http”). This module will be in C#.

Unit Tests:

1. Check if the mesh filter data is converted correctly to the STL binary file by checking the size of the binary file
2. Test if the client socket is working by sending simulated data and checking if the data is being passed correctly to a simulated server
3. Check if we are getting the mesh filter data

Server-side Application:

Assigned Members: Thomas

The server listens for incoming connections from the Unity Applications and receives the binary STL file, attaches a time stamp, and saves it locally or on a database. The server will then clean the received mesh to get rid of any unwanted artifacts or imperfections and remove the room ceiling for better viewing. This could also involve various mesh processing techniques like smoothing, decimation, or hole filling. The server will handle user input for timestamps and retrieve the corresponding mesh data accordingly. Additional functionality will be included for selecting two different meshes for mesh comparison, selecting certain parts of the mesh for mesh segmentation, and classification for the segmentation mesh.

Unit Tests:

1. Connection Handling
    a. Establishing connection check
    b. Handling failed connection
2. Request and Response Handling
    a. Sending appropriate responses based on request
    b. Handling invalid requests without without crashing
3. Multi-thread handling
    a. Ensuring shared resources are accessed safely (Mesh Data)
    b. Testing ability to scale with multiple requests (Simulate multiple connections accessing meshes)

Client-side Application:

Assigned Members: Rohan

The client-side application will be a terminal interface or GUI. There will be two modes, one for streaming the live spatial mesh from the HoloLens and one for allowing the user to input timestamps for calling previous meshes. If the former mode, the client-side application will send a request to the server for the corresponding mesh data to the timestamp that was inputted by the user. The called mesh will be visualized using OpenGL.

Unit Tests:

1. Test for Requesting Mesh Data in Live Streaming Mode
2. Test for Visualizing Mesh Data Using OpenGL
3. Test for Retrieving Mesh Data for Specified Timestamp