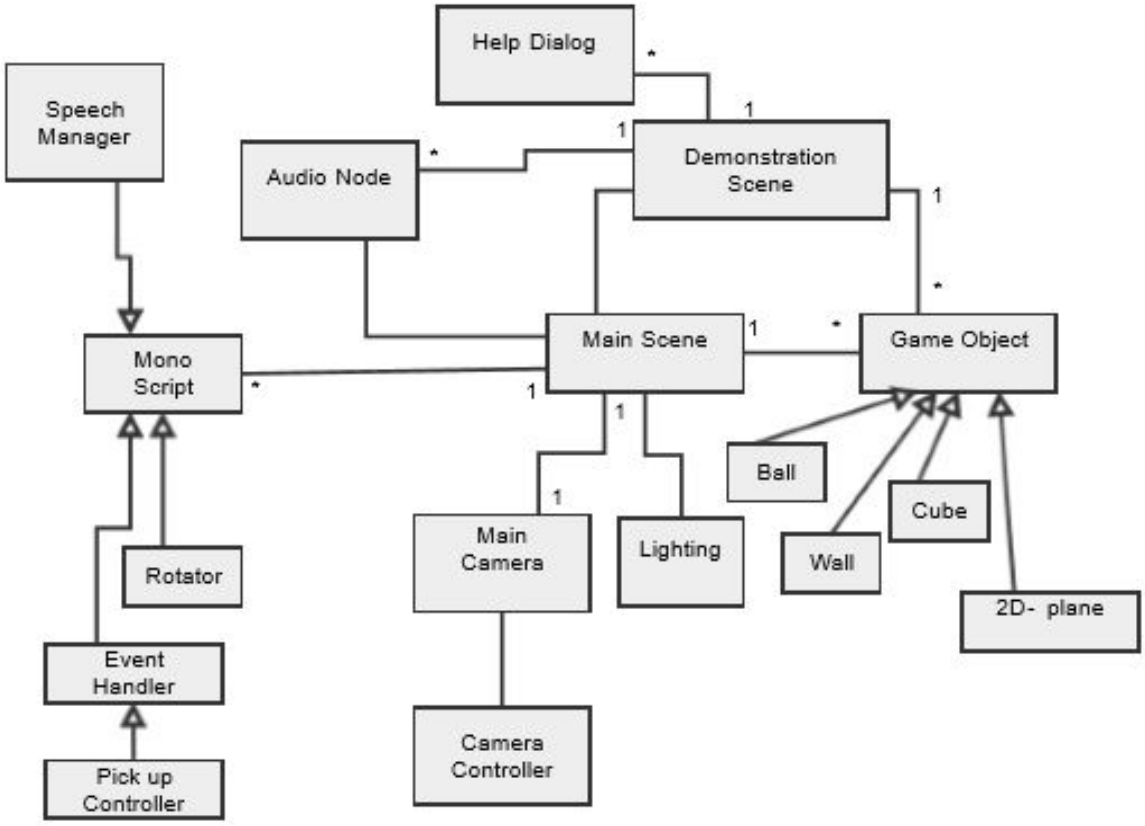
"CS 386 HoloLens Project" by Stephen White, Jack Garrard, Colton Nunley, Daniel Williamson, and James Todd

Github Link: https://github.com/Swhite9478/CS386-HoloLens-Project

Trello Link: https://trello.com/invite/b/z5QYDqJx/77df811166edfedfdc5a11206a15c38a/public-board

CS 386 Software Engineering, Spring 2017 Instructor: Marco Gerosa

Deliverable 4: Class Diagrams



-Daniel's Class Diagrams -

Speech Manager *Quick Description: Voice recognition will be used to command the hololens with your voice.*
Open: opens an application using your voice. Close: closes an application using your voice. Run: runs an application using your voice.

Demonstration area *Quick Description: demonstration areas will be used to hold the various demonstrations of this project*
Color: used to color the area Size: size of the area Shape: the shape of the area.

Unity Engine Main Scene *Quick Description: The main "container" for which all other holograms exist.*
Mass: How large of container. Dimensions: What dimension is the container. L x W x H.

-Stephen's Class Diagrams -

Ball *Quick Description: This ball will be part of an interactive demonstration of the HoloLens, which we will design.*
Color: The ball will have its own color Mass: The ball is a rigid body, therefore it has mass Physics: The ball must be able to roll when the world is tilted by the user

Cubes *Quick Description: These cubes will be part of the same scene in the Unity Engine throughout development.*
Color: Each cube will have its own color Trigger: The cubes should disappear when the ball touches them (i.e. the ball is a trigger) Rotation: These cubes will rotate in a given fashion, to give them animation

Lighting *Quick Description: Lighting on all holograms will be taken care of via the Unity Engine. This lighting will determine how dark and/or bright any object will appear to the user wearing the HoloLens*
Direction: Lighting would be pointless without any direction, therefore it must be defined Intensity: The intensity of the light will determine how dark/bright an object appears to the user Radius: The radius of the light will determine which holograms will have light on them throughout the demonstration

-James' Class Diagrams -

Wall *Quick Description: a wall that can be placed in main scene so that it can encase the user within an augmented environment.*
Color: the color of the walls are be picked within source code. Size: the size of the walls will be picked and will stay Static placement: a wall will have a static placement within the augmented world.

Audio Node *Quick Description: Audio nodes will have an audio clip that can be play at will*
Link to an audio clip: a link to an audio clip that is played when a user activates it. Shape: the shape of the Node will be static. Color: the color of the node will be static. Is plating: this is a boolean that when the node starts to play the clip, the user won't have it activate more than once.

Event Handler *Quick Description: event systems in unity watch of input and knows how that to react to that input.*
Event System manager: this is how events are manage. Touch input module: this watchs for from the user Ray caster: this looks out into a scene to see what objects should receive events.

-Colton's Class Diagrams -

Main Camera *Quick Description:* This class will take care of what and how the user can see the scene.
Projection: Deals with the angle of the main camera. 3D Position: Position in 3D space of camera. Field of View: How much of the scene is that is visible.

Help Dialog *Quick Description:* This will be a hologram that the user can access to receive text that gives instruction and or information on a certain feature.
Position/Size: Where this hologram will be placed as well as what the dimensions are. Text: What will be displayed into the dialog box.

2D Plane (Ground) *Quick Description:* Just like the title implies, this will serve as a ground like object.
Mesh Renderer: Creates the 'mesh box' around the plane that will let the collider know when to hold up an object. Mesh Collider: This is what has the script for holding the object. 3D Position: Position in 3D space of the camera.

-Jack's Class Diagrams -

Camera Controller *Quick Description: Allows for the movement of the camera using the Hololens' positional data in tilt-a-ball.*
Camera: The camera to manage. Floor: The floor of the level design.

Rotator *Quick Description: Allows for rotation of any game object over time.*
Object: the game object to rotate.

Pickup Controller *Quick Description: Allows a pickup to be collected and increments the global score for hilt-a-ball.*
Pickup: The pickup to manage Score: The player's score

Game Object *Quick Description: The class that every game object inherits from. Made for unity compatibility.*
Position: The object's position Velocity: The object's velocity

Monoscript *Quick Description: The class that every script inherits from. Made for unity compatibility.*

-Group Member Participation -

- *Stephen White:*
  - *Communicated through slack*
  - *Set up the deliverable in Google Docs*
  - *Communicated with team members in person to better understand the end goal of our project*
  - *Ensured uniformity of document*

- *James Todd:*
  - *Communicated on slack.*
  - *Completed 3 class diagrams.*
  - *Help with communicating the end goal of the project with the group.*
  - *Help with designing class diagram*
  - *Made the class diagram pdf*

- *Daniel Williamson:*
  - *Communicated through slack.*
  - *Communicated with team members in person to better understand the end goal of our project.*
  - *Completed sections that were given.*

- *Jack Garrard:*
  - *Communicated through slack*
  - *Added several classes to document*
  - *Communicated with team members to build project*

- *Colton Nunley:*
  - *Collaborated for next deliverable*
  - *Communicated through slack*
  - *Complied and gave input on organization of deliverable*
  - *Planned and communicated about what needs to be done in the next few weeks*