**EECS 495 Software for Accessibility**

**“A Day At The Dentist Is The Best Kind Of Day”**

**Design Document**

**Group members: Khalil Crawford (kkjcraw), Seungyoun Lee (seungy), Ritsuma Inaba (rinaba), Jacob Sturgis (jsturg)**

System Description

**System Purpose:**

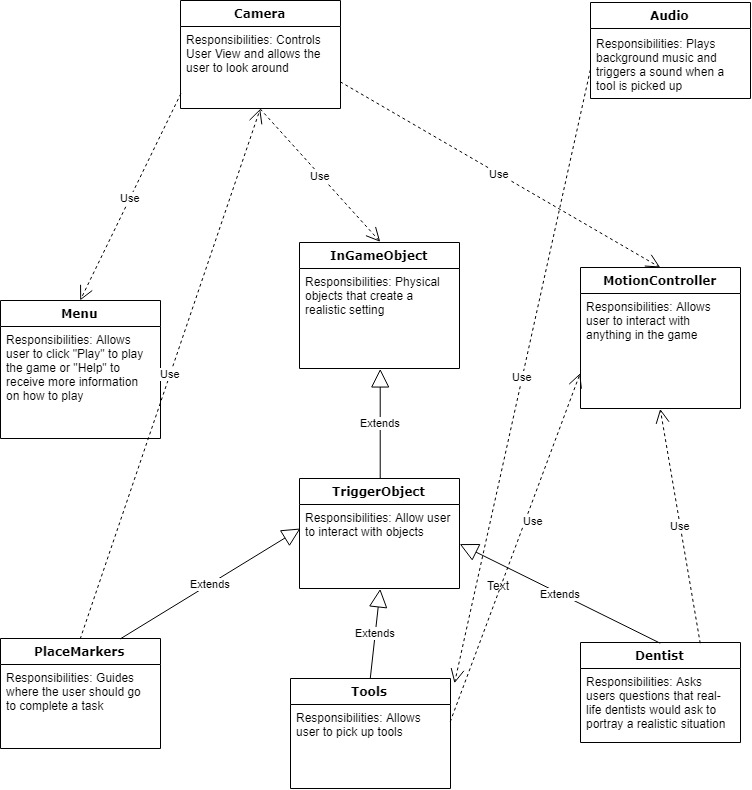
Children with autism spectrum disorder (ASD) follow routines everyday in order to fully adjust to environments or new tasks. However, many children with ASD have trouble adjusting to their dental appointments, because they only go twice a year normally and the process is not part of their everyday routine. It may take a long amount of time and possibly several appointments for children to ASD to become adjusted to the environment. The project’s focus is to desensitize children with ASD by having them explore and interact a dental office using mixed reality headsets. The goal of the project is to see if mixed reality environments, when used by children with ASD, improve their quality of life in many unfamiliar areas, such as hospitals and new schools.

**Project Scope:**

Once the game is started by the user, the user is directed to the start menu. Once the user is in the start menu, the user have two options. The user can go to the help screen, which includes the controls needed to play the game, or the user can start the game. After starting the game, the user will first go through a tutorial to get used to moving the character in the virtual reality environment.

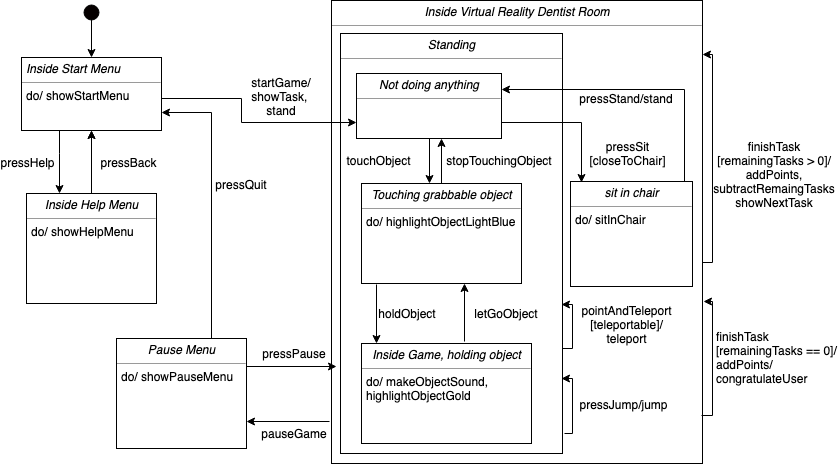
Inside the game, the user is able to interact with an item. Whether the user picks up an item or uses the item, some kind of sound will accompany the users action. They can also complete a task within the game. The user receives a reward only when the task is successfully completed.

System-level Class Diagram:



For the diagram has 9 classes. They are Camera, Audio, Menu, InGameObjects, MotionControllers, TriggerObjects, PlaceMarkers, Tools, and the Dentist class. The Camera is dependent on Menu as one of Menu’s functionalities include teleporting the player, giving the camera a different viewpoint. The Camera depends on InGameObject by restricting users from walking through the objects. The Camera depends on MotionControllers as the teleportation function will always change the Camera view point. The Audio depends on Tools because when a tool object is picked up, a sound will trigger. The InGameObject includes a child class called TriggerObject. The TriggerObject includes all objects that include a function dependent on the user-behavior: it’s child classes include PlaceMarkers, Tools, and Dentist. The PlaceMarkers depend on Camera as they will only appear if the Camera signals that it is at a certain spot. The Dentist class and the Tools class depends on MotionControllers as it is necessary for the user to use MotionControllers to interact with these objects.

System-level Statechart Diagram:



The player first begin in the start menu. In the start menu, the player can either press the help button or the start button. If the help button is pressed, the start menu disappears and help menu pops up. The player can hit the back button to return to the start menu.

After pressing the start button, the player gets transported to the dentist room. The player would be standing and see instructions on the first task to accomplish. After successfully completing the task, the total number of points will increase and the player will be assigned a new task. After completing all of the tasks, the player will be congratulated for completing the game. Even after completing all of the tasks, the player will still be able to move around the room.

While inside the dentist room, the player can pause the game at any time. If the game is paused, the pause menu will show up. The player can either resume or quit the game from the pause menu. Quitting the game will bring the player back to the start menu.

Inside the dentist room, the player can either be standing or sitting. Sitting on the chair is only possible if the player is close to the chair and not touching/holding any object. While in the standing state, the player can always jump and teleport to move around the room. When a player touches a grabbable object, the object's color will become light blue. From this state, the player can press the grab button to grab the object or move the hand away from the object. While the user is holding the object, the object's color will turn gold and make a sound.

Class/Function Level Design:

**Class:**

* Name: Placemarkers
* Description: The purpose of Placemarkers is to guide the user through the tasks by showing the user where to go and enables various other objects to help along the way. It appears as a big light blue cylinder in the environment.
* Attributes:
  + Name: enableObjects
    - Description: A list of objects to be enabled after the Transmarker is disabled Transmarker’s collided function is called
    - Types: List<GameObject>
  + Name: disableObjects
    - Description: A list of objects to be disabled after the Transmarker’s collided function is called
    - Type: List<GameObject>
* Operations:
  + Name: calcXZDistance
    - Description: Calculates the distance between the user and the Transmarkers’ physical blue cylinder.
    - Output: None
    - Input: None
  + Name: Update
    - Description: Calls calcXZDistance on every time step to check if user has entered the Transmarkers radius. If so, calls collided then disables itself.
    - Output: None
    - Input: None
  + Name: collided
    - Description: Enables and disables GameObjects according to the two attributes
    - Output: enableObjects, disableObjects
    - Input: enableObjects, disableObjects

Traceability Matrix (Requirements Cross Reference)

|  | Camera | Audio | Motion Controller | Tools | Trigger Objects | Background Objects | Dentist | Menu |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change User Position | ✓ |  | ✓ |  |  |  |  |  |
| Change User Orientation | ✓ |  |  |  |  |  |  |  |
| Grab/Let Go Tools |  |  | ✓ | ✓ |  |  |  |  |
| Tool Sounds |  | ✓ |  | ✓ |  |  |  |  |
| Vibrate Tools |  |  |  | ✓ |  |  |  |  |
| Play Game |  |  | ✓ |  |  |  |  | ✓ |
| Tasks |  |  |  |  | ✓ |  | ✓ |  |
| Score | ✓ |  |  |  | ✓ |  | ✓ |  |
| Physical Objects | ✓ |  |  |  |  | ✓ |  |  |

Design Alternatives and Rationale

One design alternative that came up at the beginning of the project cycle is which application we wanted to develop in. We thought about developing with the Unreal Engine, as one team member has previous experience developing with the Unreal Engine. Developing on the Unreal Engine minimizes the amount of work that needs to be done and has less bugs on average when developing on the Unity engine. The glaring issue with the Unreal Engine is the lack of documentation on the Unreal Engine while using Acer Mixed Reality Headset compared to the large amount of documentation on the Unity Engine while using the Acer Mixed Reality Headset.

References

FusedVR, director. *Getting Started with Windows "Mixed Reality" Toolkit ( Unity 2017 VR Tutorial )*. *YouTube*, FusedVR, 17 Feb. 2018, [www.youtube.com/watch?v=PV8ycLfXFyM&t=584s](http://www.youtube.com/watch?v=PV8ycLfXFyM&t=584s).

“Getting Started with VR Development.” *Unity Tutorials*, Unity, unity3d.com/learn/tutorials/topics/xr/getting-started-vr-development.

Turner, Alex. “Unity Development Overview - Mixed Reality.” *Microsoft Docs*, Microsoft, 20 Mar. 2018, docs.microsoft.com/en-us/windows/mixed-reality/unity-development-overview.

Wester, Matt, director. *Pick up and Move Objects | Unity 2017 Tutorial*. *YouTube*, YouTube, 5 Sept. 2017, [www.youtube.com/watch?v=JtflOvhOO1Y](http://www.youtube.com/watch?v=JtflOvhOO1Y).

D. Kline, “HoloToolkit 2017.4.2.0”, 2017, Github Repository

https://github.com/Microsoft/MixedRealityToolkit-Unity/releases/tag/2017.4.2.0