SOFTWARE REQUIREMENTS SPECIFICATION

for

VR Construction Training System

Version 1.0

Prepared by
Group Name: Beaver Games
Michael Commins , Steven Phillips , Xizhi Yang

CS 463
Spring 2020
Github Repo: git@github.com:MCommins/VR-Construction-Training-System.git

Contents

1 Introduction

1.1 Purpose

This document specifies the Software Requirements Specifications for the Virtual Reality Construction Training System. It describes the scope of the software, system interfaces, and the functional and non-functional requirements of the software.

1.2 Scope

The purpose of the Virtual Reality (VR) Construction Training System is to create immersive and engaging exercises to train construction workers on jobsite safety and operations. The system will be an application created using the Unity game engine that is compatible with VR headsets that provide both auditory and visual stimuli. The application will allow users to interact with a three-dimensional (3D) construction site environment through the VR headset and its handheld peripherals, allowing them to identify workplace hazards and complete tasks. The Construction Training System allows clients to customize the virtual environment in order to create scenarios and tasks that comply with their safety standards and jobsite surroundings.

1.3 Product overview

1.3.1 Product perspective System Interfaces:
The Construction Training System is a desktop application that will currently be available to run on the Windows and Mac operating systems.

User Interfaces:
The application will have separate GUI's for users using a VR headset and users that are customizing the 3D environment. The VR GUI will use the controllers and other peripherals that accompany the VR headset to allows users to interact with menus and virtual objects, and to navigate the virtual construction site. Users that are customizing the 3D environment will use the mouse and keyboard to interact with menus, toolbars, buttons, and grids that appear on the computer monitor.

Software Interfaces:
The application will use Unity's built-in VR support to send audio and visual data to the VR headset.

Memory:
Users will save their customized construction environments as files on their computer(s).

1.3.2 Product functions
The major functions performed by the VR Construction Training System are outlined in Section 3 "Specific Requirements".

1.3.3 User characteristics
There are two distinct groups of users expected to make use of the VR Construction Training System:
Construction workers are the primary users of the VR headset, although they are not expected to have past experience using such equipment. They will be expected to be able to be physically able enough to move their bodies in

order to look around the virtual environment and use virtual tools and equipment.

Safety Coordinators, along with other construction company employees charged with training workers, will be using the application to create tasks and environment to be experienced using the VR headset. These users are expected to know the safety guidelines of their company, the proper procedures for solving jobsite hazards, the names and operation of the tools and equipment represented in the application, and the general architecture of the structures their workers will be navigating. They are expected to have experience using a keyboard and mouse in order to navigate menus and to drag and drop items on the screen. They are not expected to have experience using VR equipment.

1.3.4 Limitations
Most VR headsets do not have methods for tracking the user outside of a 5ft-by 5ft area without purchasing additional sensors, meaning the user's motion is limited to that area. Most headsets are also limited to tracking the movement of the user's head and the handheld controllers.

The number of virtual objects and customization options made available to users have a limit due to the user's computer having a limit to the processing power and storage space available for the application.

1.4 Definitions

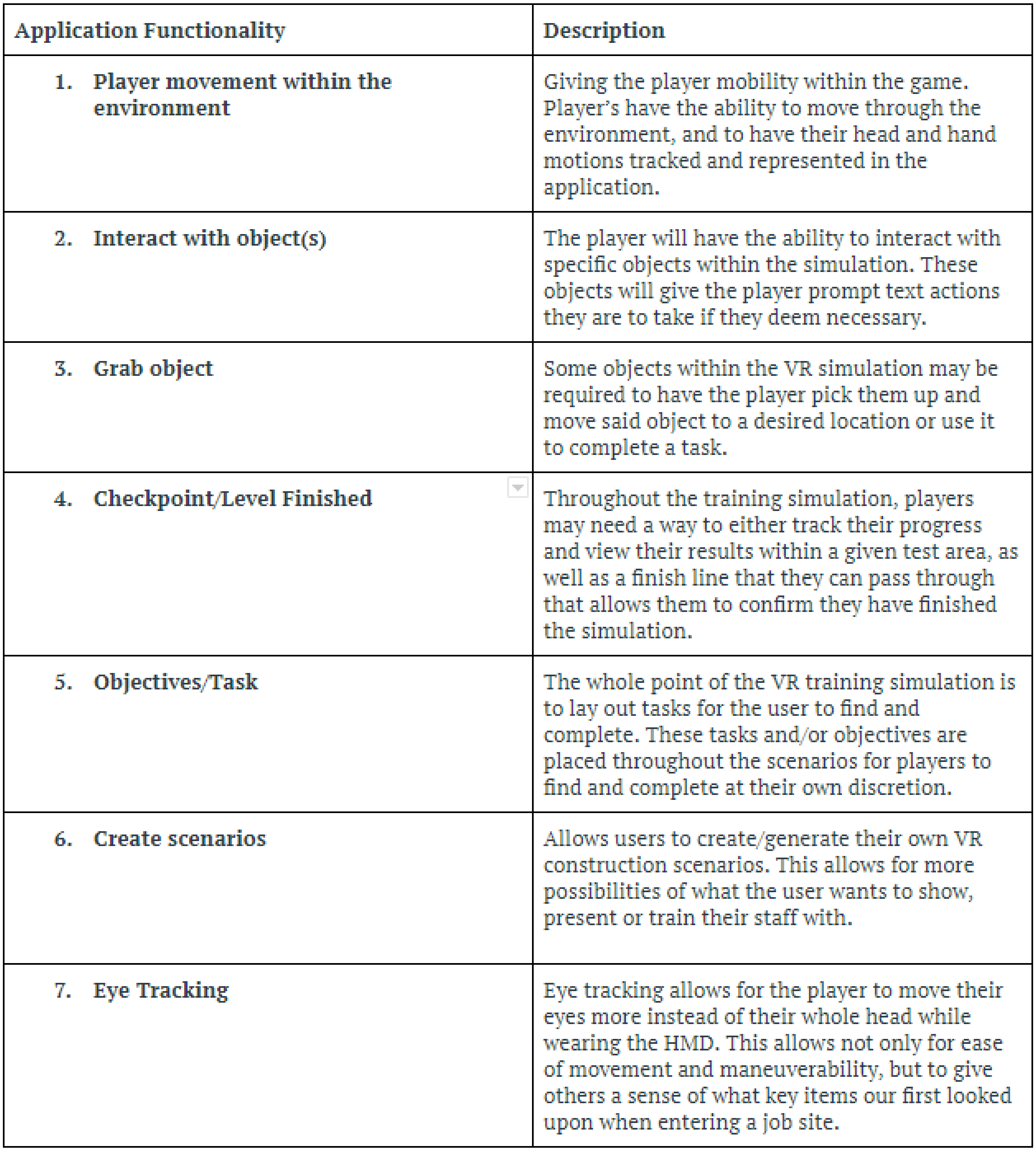
Unity - Unity is a cross-platform game engine that can be used to create three-dimensional, two-dimensional, virtual reality, and augmented reality games, as well as simulations and other experiences.

2 References

N/A

3 Specific requirements

3.1 Functions

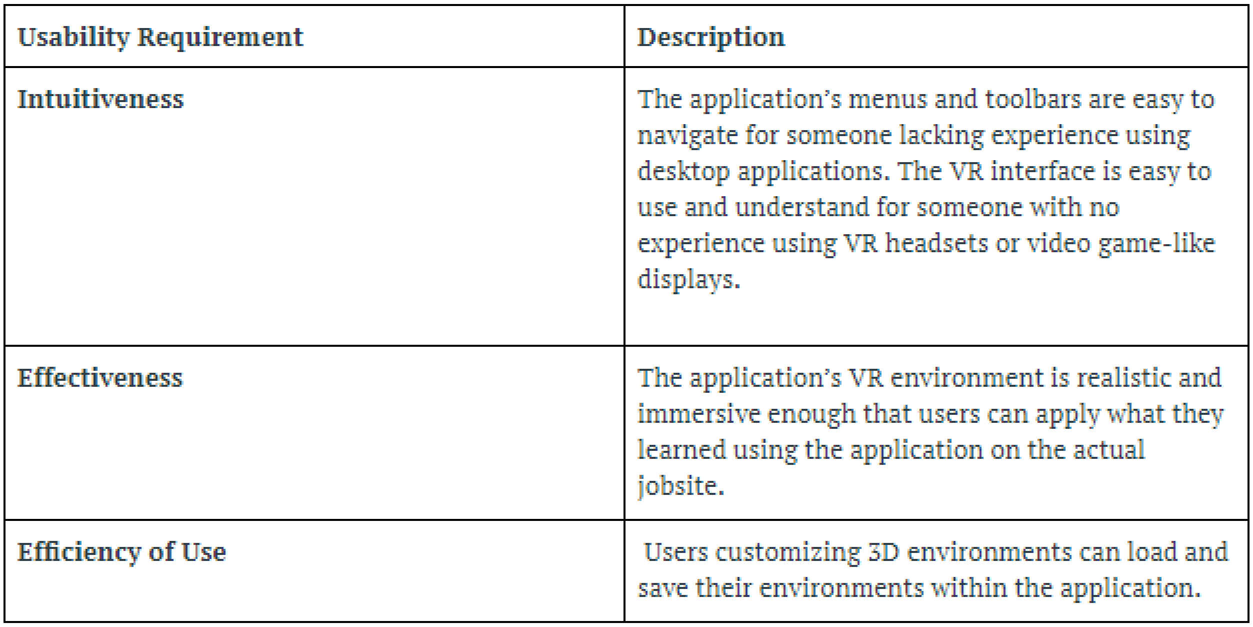


Application Functionality	Description
1. **Player movement within the environment**	Giving the player mobility within the game. Player’s have the ability to move through the environment, and to have their head and hand motions tracked and represented in the application.
2. **Interact with object(s)**	The player will have the ability to interact with specific objects within the simulation. These objects will give the player prompt text actions they are to take if they deem necessary.
3. **Grab object**	Some objects within the VR simulation may be required to have the player pick them up and move said object to a desired location or use it to complete a task.
4. **Checkpoint/Level Finished**	Throughout the training simulation, players may need a way to either track their progress and view their results within a given test area, as well as a finish line that they can pass through that allows them to confirm they have finished the simulation.
5. **Objectives/Task**	The whole point of the VR training simulation is to lay out tasks for the user to find and complete. These tasks and/or objectives are placed throughout the scenarios for players to find and complete at their own discretion.
6. **Create scenarios**	Allows users to create/generate their own VR construction scenarios. This allows for more possibilities of what the user wants to show, present or train their staff with.
7. **Eye Tracking**	Eye tracking allows for the player to move their eyes more instead of their whole head while wearing the HMD. This allows not only for ease of movement and maneuverability, but to give others a sense of what key items our first looked upon when entering a job site.

3.2 External interfaces

Using Unity to build the software. The software can be interacted with HMD and the controller.

3.3 Usability Requirements

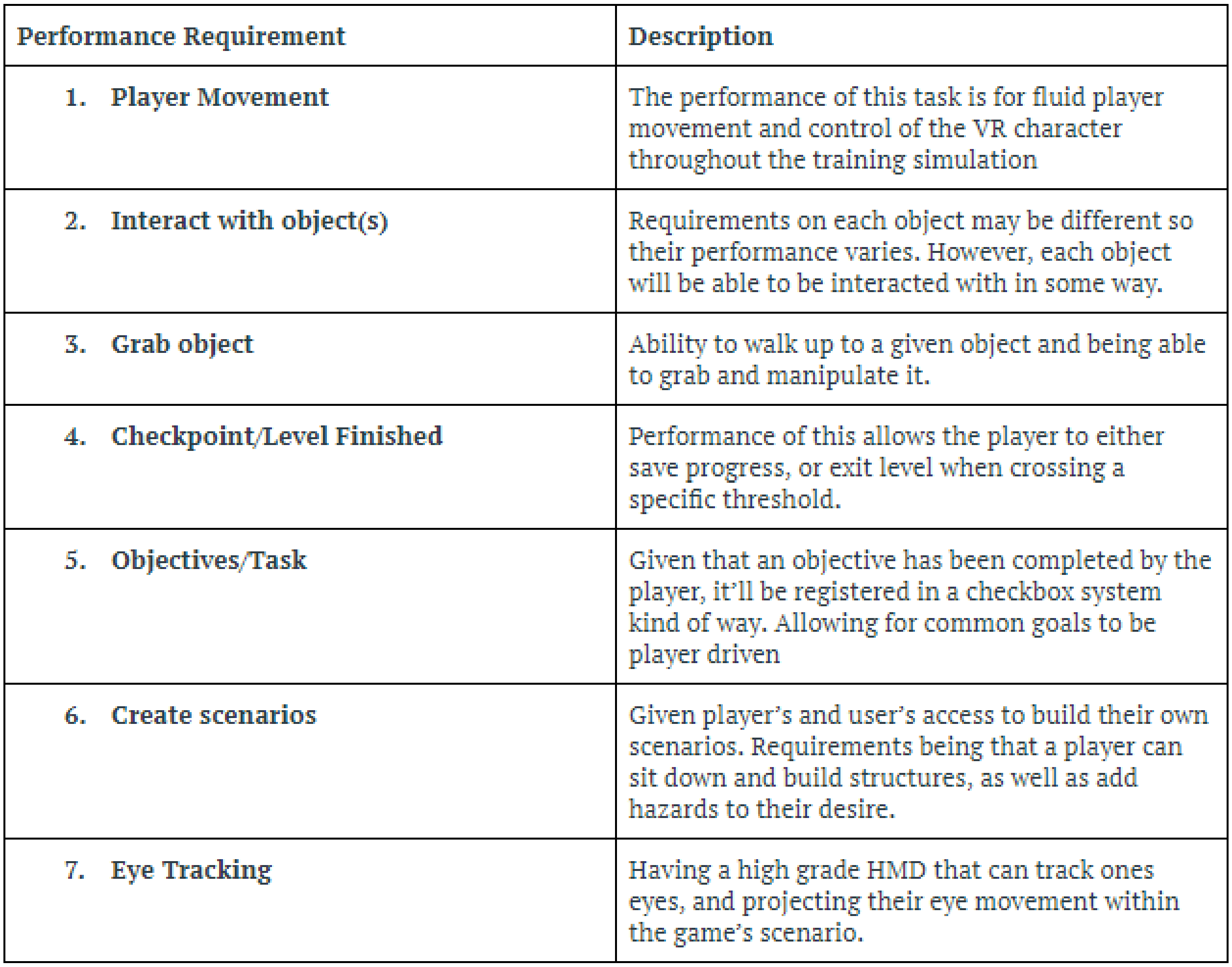


Usability Requirement	Description
Intuitiveness	The application's menus and toolbars are easy to navigate for someone lacking experience using desktop applications. The VR interface is easy to use and understand for someone with no experience using VR headsets or video game-like displays.
Effectiveness	The application's VR environment is realistic and immersive enough that users can apply what they learned using the application on the actual jobsite.
Efficiency of Use	Users customizing 3D environments can load and save their environments within the application.

3.4 Design constraints

Hardware: Some of the HMD don't have eye tracking function. People with Virtual reality sickness will not be able to use software.
Software: Depends on the environment, some large objects can't be added to the scenario.

3.5 Performance requirements



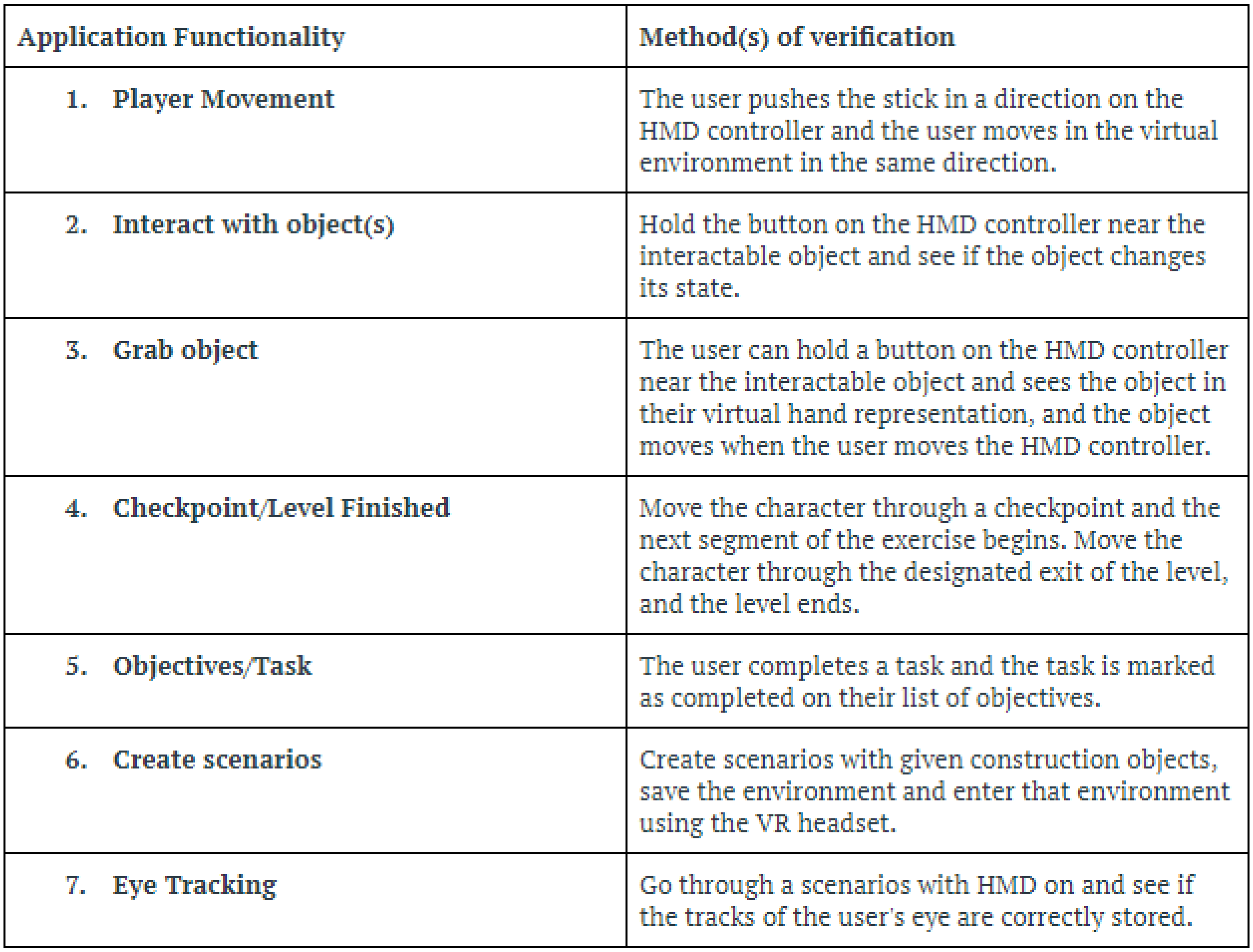
Performance Requirement	Description
1. **Player Movement**	The performance of this task is for fluid player movement and control of the VR character throughout the training simulation
2. **Interact with object(s)**	Requirements on each object may be different so their performance varies. However, each object will be able to be interacted with in some way.
3. **Grab object**	Ability to walk up to a given object and being able to grab and manipulate it.
4. **Checkpoint/Level Finished**	Performance of this allows the player to either save progress, or exit level when crossing a specific threshold.
5. **Objectives/Task**	Given that an objective has been completed by the player, it'll be registered in a checkbox system kind of way. Allowing for common goals to be player driven
6. **Create scenarios**	Given player's and user's access to build their own scenarios. Requirements being that a player can sit down and build structures, as well as add hazards to their desire.
7. **Eye Tracking**	Having a high grade HMD that can track ones eyes, and projecting their eye movement within the game's scenario.

4 Verification

4.1 External interfaces

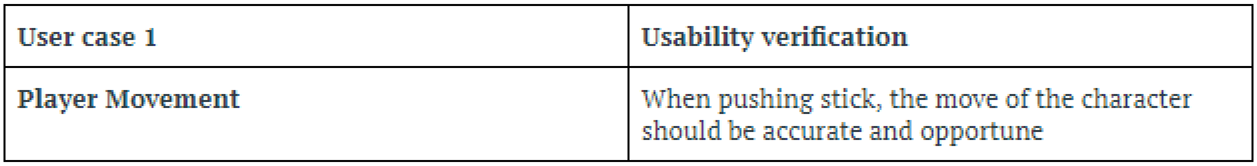
Software can be run at the HMD. User can interact with the software using controller.

4.2 Functions Verification

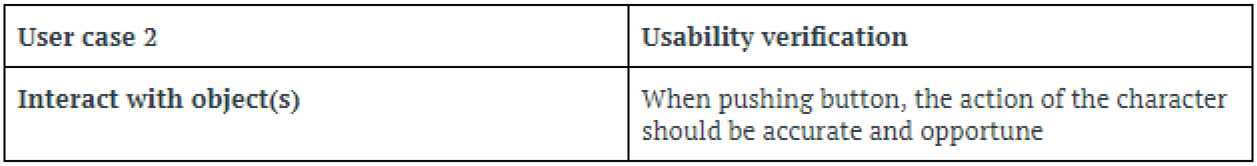


Application Functionality	Method(s) of verification
1. **Player Movement**	The user pushes the stick in a direction on the HMD controller and the user moves in the virtual environment in the same direction.
2. **Interact with object(s)**	Hold the button on the HMD controller near the interactable object and see if the object changes its state.
3. **Grab object**	The user can hold a button on the HMD controller near the interactable object and sees the object in their virtual hand representation, and the object moves when the user moves the HMD controller.
4. **Checkpoint/Level Finished**	Move the character through a checkpoint and the next segment of the exercise begins. Move the character through the designated exit of the level, and the level ends.
5. **Objectives/Task**	The user completes a task and the task is marked as completed on their list of objectives.
6. **Create scenarios**	Create scenarios with given construction objects, save the environment and enter that environment using the VR headset.
7. **Eye Tracking**	Go through a scenarios with HMD on and see if the tracks of the user's eye are correctly stored.

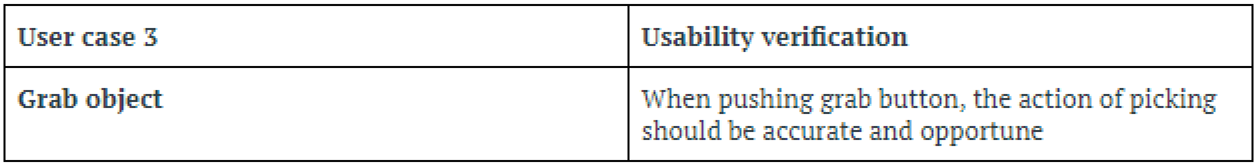
4.3 Performance requirements



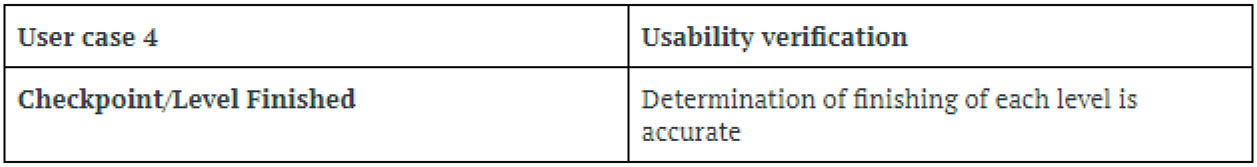
User case 1	Usability verification
Player Movement	When pushing stick, the move of the character should be accurate and opportune



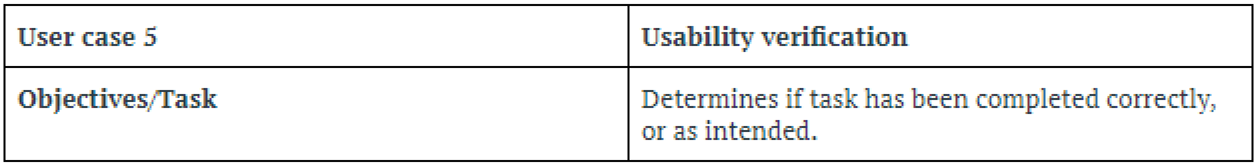
User case 2	Usability verification
Interact with object(s)	When pushing button, the action of the character should be accurate and opportune



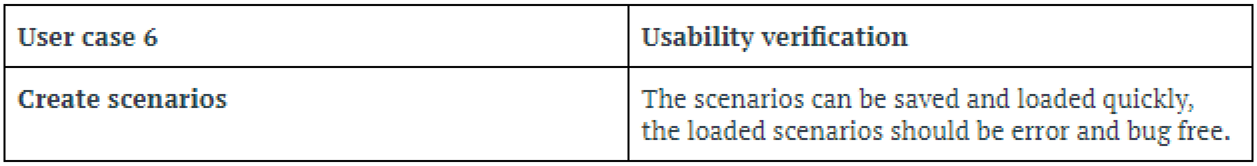
User case 3	Usability verification
Grab object	When pushing grab button, the action of picking should be accurate and opportune



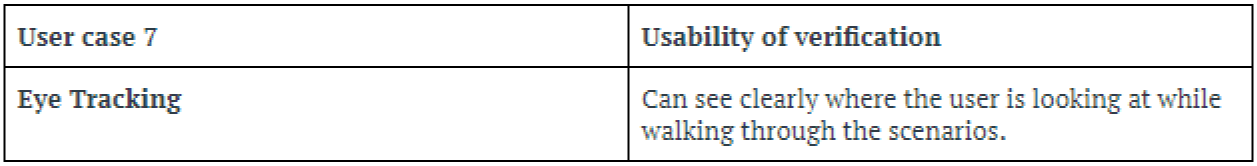
User case 4	Usability verification
Checkpoint/Level Finished	Determination of finishing of each level is accurate



User case 5	Usability verification
Objectives/Task	Determines if task has been completed correctly, or as intended.

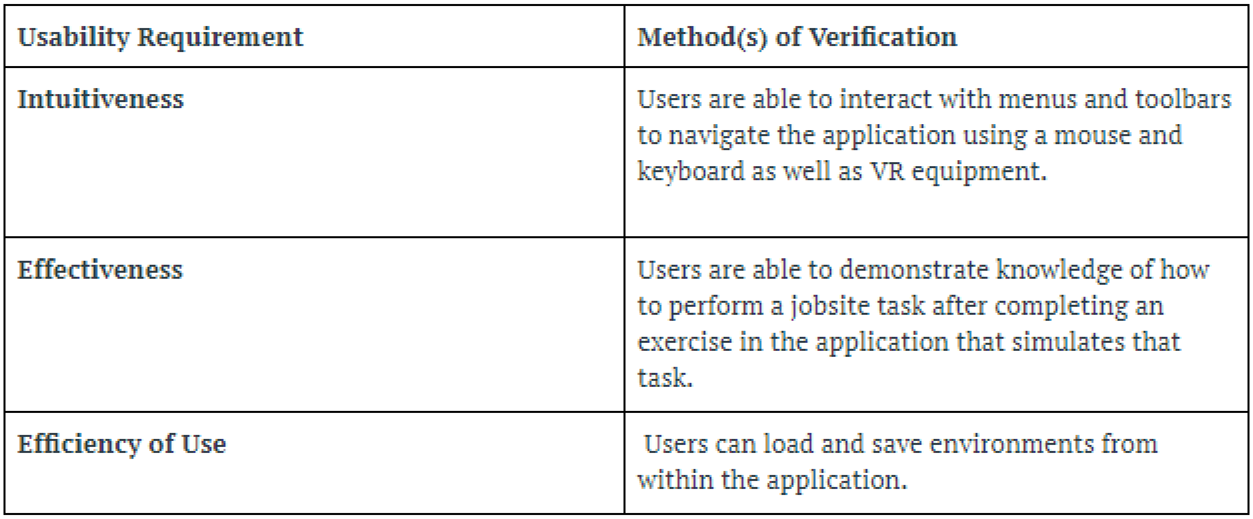


User case 6	Usability verification
Create scenarios	The scenarios can be saved and loaded quickly, the loaded scenarios should be error and bug free.



User case 7	Usability of verification
Eye Tracking	Can see clearly where the user is looking at while walking through the scenarios.

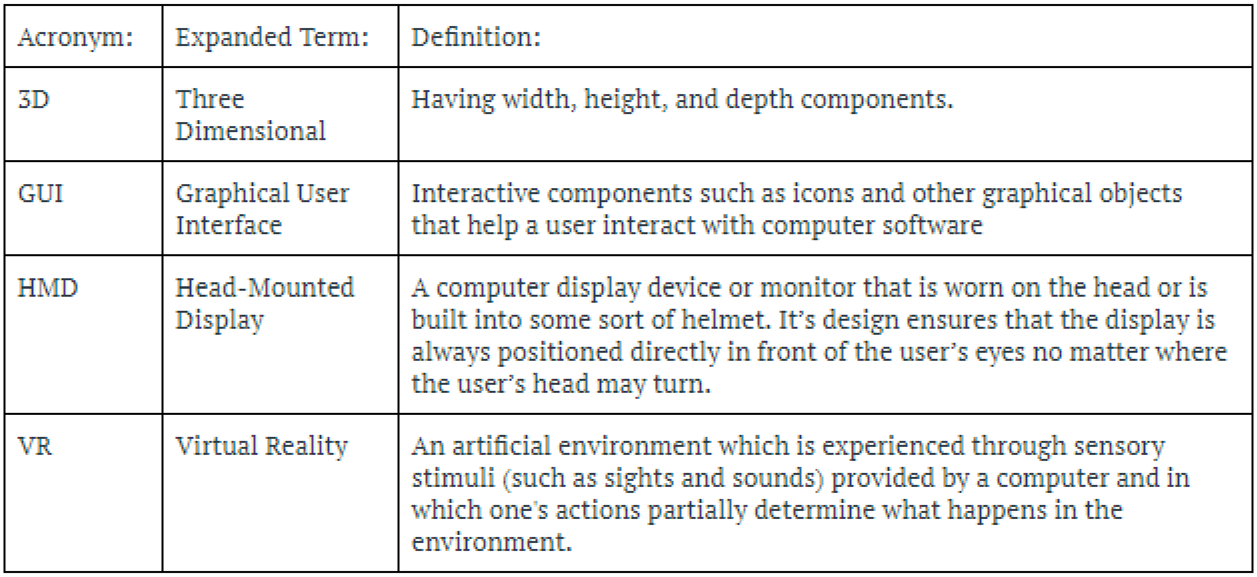
4.4 Usability Requirements Verification



Usability Requirement	Method(s) of Verification
Intuitiveness	Users are able to interact with menus and toolbars to navigate the application using a mouse and keyboard as well as VR equipment.
Effectiveness	Users are able to demonstrate knowledge of how to perform a jobsite task after completing an exercise in the application that simulates that task.
Efficiency of Use	Users can load and save environments from within the application.

5 Appendices

5.1 Acronyms and abbreviations



Acronym:	Expanded Term:	Definition:
3D	Three Dimensional	Having width, height, and depth components.
GUI	Graphical User Interface	Interactive components such as icons and other graphical objects that help a user interact with computer software
HMD	Head-Mounted Display	A computer display device or monitor that is worn on the head or is built into some sort of helmet. It's design ensures that the display is always positioned directly in front of the user's eyes no matter where the user's head may turn.
VR	Virtual Reality	An artificial environment which is experienced through sensory stimuli (such as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment.

6 Requirement Changes

As of now we have yet to make any changes in regards to our project. The only things we have yet to implement for this code review is our checkpoint/finished requirement. However, come code freeze May-8th it will be implemented.