CS 440: Testing and Inspection Report
VR Escape Room

The Document Detailing the Functions and Design of the Game

Prepared by
Aakash Kotak, Patricia Guera, Dylan Ou, and Krystian Stanuch
***for use in* CS 440**
at the
University of Illinois Chicago

April 2020

Table of Contents

I Project Description

1 Project Overview

The focus of this project is to provide a unique virtual experience for users to solve escape rooms from the comfort of their homes. The application consists of four different rooms that the player can choose to solve. The application is primarily compatible with HTC Vive.

2 Project Domain

The project domain is entertainment/gaming. It was created using Unity and a VR simulator.

3 Relationship to Other Documents

The project report for VR Escape Room from 2018 [2] consisting of the description, requirements, and design is what this project, VR Escape Room from 2020, is based on.

4 Naming Conventions and Definitions

4a Definitions of Key Terms

HTC Vive: A Virtual reality headset developed by HTC and Valve. It allows the user to move in 3D space and use motion-tracked handheld controllers to interact with the environment.
Virtual Reality (VR): A computer-generated simulation of a 3D image or environment that can be interacted in a seemingly real or physical way by a person using compatible electronic equipment.
Escape Room: Themed rooms where you have to find clues and solve puzzles to either escape the room or complete a goal within a certain amount of time.

4b UML and Other Notation Used in This Document

This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in [1]. Any exceptions are noted where used.

4c Data Dictionary for Any Included Models

Levels = {Book interaction + PlayAudio}
Completion = {exitLevel + StopMusic}

II Testing

5 Items to be Tested

We would want to test all our levels to see if all the VR mechanics are working correctly (ID#1) and the player is able to move properly in them, meaning not glitching out of the map. We would

also want to test if our game actually works with VR because we use a VR simulator for our game (ID#2). We also want to test if each level can be completed and has an ending, meaning you can get to the "You Win!" screen after completing the puzzle (ID#3).

6 Test Specifications

ID#1 - Levels

Description: We want to make sure that the levels are properly implemented with VR mechanics such as our picking up stuff, rotating stuff, and opening doors scripts, as well as make sure the player can walk inside the room without glitching outside the map.

Items covered by this test: Levels and VR mechanics scripts.

Requirements addressed by this test: NA

Environmental needs: HTC Vive

Intercase Dependencies: NA

Test Procedures: To run the test, make sure that the levels are all implemented in the game, and that there are no missing assets. Attach the scripts to the items/assets in the level that you want to pick up/rotate/open. Make sure the player is inside the level, and not glitching out of the rooms.

Input Specification: Put the scripts on the items you want to pick up/rotate/open.

Output Specifications: You should be able to pick up/rotate/open items if the scripts worked, meaning that the VR mechanics worked.

Pass/Fail Criteria: You will pass this test if all the VR mechanics were implemented correctly on the items in each level.

ID#2 - VR

Description: We want to test if the game actually runs with a VR device such as HTC Vive since we use a VR simulator.

Items covered by this test: VR mechanics and seeing if the game is compatible with VR.

Requirements addressed by this test: You need a HTC Vive, and need to make sure the game is compatible with VR. All the VR scripts are on items for each level.

Environmental needs: HTC Vive.

Intercase Dependencies: ID#1 needs to be completed to get this to work properly.

Test Procedures: Make sure the HTC Vive is connected properly with the computer you are using, and press play to see if you are able to see the game in VR.

Input Specification: NA

Output Specifications: The game should be able to run in VR.

Pass/Fail Criteria: You pass the test if the game is able to run in the HTC Vive.

ID#3 - Completion

Description: We want to test if the game has an ending, meaning you're able to beat the level and then go back to the main menu and select another level to play.

Items covered by this test: Testing the win condition for each level and VR mechanics.

Requirements addressed by this test: NA

Environmental needs: HTC Vive.

Intercase Dependencies: ID#1 needs to be completed to get this to work properly.

Test Procedures: Complete the puzzle for the level and see if you're able to get the win screen such as picking up the key to win the game in the Living Room, or opening the door in The Bunker.

Input Specification: NA

Output Specifications: You should be able to win the level if you pick up the item to win the game or open the door to exit the level.

Pass/Fail Criteria: You pass the test if you are able to win the level and get to the main menu screen.

7 Test Results

ID#1 - Levels

Date(s) of Execution: 02/23, 03/13, 04/05, 04/17, 04/19.

Staff conducting tests: We had all of us testing the levels to see if they ran properly with the game, and Dylan tested the VR mechanics to see they worked.

Expected Results: We expected the levels to run correctly, but the VR mechanics may have some problems such as you're not able to pick up the item properly.

Actual Results: Both the levels and the VR mechanics were able to function correctly.

Test Status: Pass

ID#2- VR

Date(s) of Execution: 03/16

Staff conducting tests: We had Dylan test to see if the VR headset would work properly with VR.

Expected Results: We expected that using the headset would work with our game when we would get the HTC Vive from Professor Bell.

Actual Results: Due to the pandemic, we didn't end up getting the VR device from professor Bell, so we had to improvise and work with a VR simulator for our game.

Test Status: Fail, because we didn't end up testing it with the HTC Vive, but Pass because our VR simulator works properly.

ID#3 - Completion

Date(s) of Execution: 04/17, 04/19

Staff conducting tests: All of us tested to see if the win conditions were possible.

Expected Results: We expected the rooms with doors would be able to get the win condition, but the rooms that needed an item to win the game would not work.

Actual Results: All of the rooms were able to get the win condition on 04/19.

Test Status: Pass

8 Regression Testing

ID#1, ID#2, and ID#3 are all repeated because we wanted to make sure that all of these tests worked properly since they all depend on each other.

III Inspection

9 Items to be Inspected

```
Krystian Stanuch
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;


public class exitLevel : MonoBehaviour
{
   public bool isEntered;


   public Image win;
   public Animator anim;


   AudioSource audioSource;


   private void Start()
   {
       audioSource = GetComponent<AudioSource>();
   }
```

```
if (other.tag == "MainCamera")
        {
            isEntered = true;
            if (!win.GetComponent<Image>().enabled)
            {
                win.GetComponent<Image>().enabled = true;
                StartCoroutine(ChangeLevel());
            }


        }
        else
        {
            isEntered = false;
        }
    }


    IEnumerator ChangeLevel()
    {
        anim.SetBool("Fade", true);
        audioSource.Play();
        yield return new WaitUntil(() => win.color.a == 1
&& !audioSource.isPlaying);
        SceneManager.LoadScene(1);
```

```
Patricia Guera
using System.Collections;
using System.Collections.Generic;
using UnityEngine;


public class StopMusic : MonoBehaviour
{
   // Start is called before the first frame
update
   void Start()
   {
       if (BackgroundMusic.Instance != null)
       {

BackgroundMusic.Instance.gameObject.GetComponent
<AudioSource>().Stop();
       }
   }


   // Update is called once per frame
   void Update()
   {
```

```
Aakash Kotak
using System.Collections;
using System.Collections.Generic;
using UnityEngine;


public class PlayAudio : MonoBehaviour
{
   // Start is called before the first frame update
   void Start()
   {
       if(BackgroundMusic.Instance != null &&
!BackgroundMusic.Instance.gameObject.GetComponent<Au
dioSource>().isPlaying)
       {

BackgroundMusic.Instance.gameObject.GetComponent<Aud
ioSource>().Play();
       }
   }


   // Update is called once per frame
   void Update()
```

```
Dylan Ou
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using VRTK;
using UnityEngine.UI;
using UnityEngine.SceneManagement;


public class BookInteract : MonoBehaviour
{
    public bool isGrabbed = false;
    public Image win;
    public Animator anim;
    AudioSource audioSource;


    // Start is called before the first frame update
    void Start()
    {
        if (GetComponent<VRTK_InteractableObject>() == null)
        {
            Debug.LogError("Team3_Interactable_Object_Extension is
required to be attached to an Object that has the
VRTK_InteractableObject script attached to it");
```

```
GetComponent<VRTK_InteractableObject>().Interactab
leObjectGrabbed += new
InteractableObjectEventHandler(ObjectGrabbed);
        audioSource = GetComponent<AudioSource>();
    }


    private void ObjectGrabbed(object sender,
InteractableObjectEventArgs e)
    {
        isGrabbed = true;
        if (!win.GetComponent<Image>().enabled)
        {
            win.GetComponent<Image>().enabled =
true;
            StartCoroutine(ChangeLevel());
        }
    }
```

10 Inspection Procedures

A simple checklist of code inspections was added below to showcase what the criteria was for inspecting each group members code submission.

	Krystian's Code	Patricia's Code	Aakash's Code	Dylan's Code
Was the code complete?				
Is the code easily understandable?				

Was the code written using the set standards as the other code?				
Is the same set of code duplicated more than twice?				
Is the code easily debuggable to see any errors in it?				
Is the code bug free?				
Does the code have too many responsibilities?				

11 Inspection Results

Each person inspected everyone else's code, so for example for Krystian’s code, the people who inspected it were the 3 other group members. This was done 4 times, for each person who submitted code for inspection. The inspections happened on April 25, 2020 and a checklist was added below for each submission to display any flaws or faults in the code.

	Krystian’s Code	Patricia’s Code	Aakash’s Code	Dylan’s Code
Was the code complete?	X	X	X	X
Is the code easily understandable?		X	X	
Was the code written using the set standards as the other code?	X	X	X	X
Is the same set of code				

duplicated more than twice?				
Is the code easily debuggable to see any errors in it?	X	X	X	X
Is the code bug free?	X	X	X	X
Does the code have too many responsibilities?				

IV Recommendations and Conclusions

2 of the 3 tests have passed the testing process while all of the inspection processes have passed. Some actions that should be taken next is that we want to figure out how to get the VR system to work without using a VR simulator so we can pass the test involving the VR system.

V Project Issues

12 Open Issues

Unity is occasionally updating, so whether the assets we have would be affected by these updates is uncertain.

New devices could affect the Virtual Reality Toolkit asset as it only supports mainstream virtual reality devices currently (in 2020) such as HTC Vive, Oculus Rift, Playstation VR, and some more. Unless the VRTK updates to reflect these new devices, we cannot develop a port to them.

13 Waiting Room

Proper use of virtual reality devices such as HTC Vive, Oculus Rift, and Playstation VR could not be tested due to the 2019-20 coronavirus pandemic and the fact that we did not have a virtual reality device to test with. Despite this, Professor Bell has stated that we could have used his HTC Vive and just needed him to bring it in order to test the game.

More mechanics could have been implemented such as opening shelves, interacting with objects in your hands, and other forms of movement besides teleportation for other players to use.
A proper pause menu for players to return to the main menu/level select.

There could have been more sound effects to make use of the three dimensional audio features in Unity.

More complex levels could have been created besides one room levels.

14 Ideas for Solutions

We must make sure to have a virtual reality device available to use to test this game during development.

More research on the Virtual Reality Toolkit asset would help the development of the game and add more features.

More original assets such as models and sound effects would help give this game a unified look, sound, and feel.

15 Project Retrospective

Using the Virtual Reality Toolkit asset really made things easier for virtual reality development, but we could have definitely dived more into its features for an even more immersive game experience.

Making the levels work with the Virtual Reality Toolkit was the most difficult part of the project. Escape rooms are puzzles where you have to look around for clues. Planning out the steps of the puzzles and making it work with the level design and VRTK had to be addressed often due to our knowledge of Unity being minimal. Despite this, we definitely learned a lot about game development by working on an ambitious project such as this.

VI Glossary

HTC Vive: A Virtual reality headset developed by HTC and Valve. It allows the user to move in 3D space and use motion-tracked handheld controllers to interact with the environment. The HTC Vive is used for this project.

Virtual Reality (VR): A computer-generated simulation of a 3D image or environment that can be interacted in a seemingly real or physical way by a person using compatible electronic equipment. The project is VR-based.

Escape Room: Themed rooms where you have to find clues and solve puzzles to either escape the room or complete a goal within a certain amount of time. In this project there are four escape rooms.

VII References / Bibliography

[1] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

[2] J. Arcivar, B.G.Gibis, D. Mehta, M Gandhi, CS 440: Project Report VR Escape Room, 2018.

[3] VRTK - Virtual Reality Toolkit Documentation

[4] Unity - Manual

VIII Index