Final Handoff Report

Escape Room Challenge

David McVittie

CS 463 Spring 2024

Oregon State University

Abstract:

The purpose of this document is to record the project goals, current status and technical details of the Escape Room Challenge. Firstly the aim of the project is to make an immersive and accessible escape room experience through a video game medium. The game has a first person perspective, and has the user solving various puzzles with the end goal of escaping the room. This project aims to achieve these goals by allowing the player to interact with the game world by picking up and holding onto objects and manipulating free standing objects in the world, examples being a flashlight and a wall mounted button, respectively. The game is not in a functional condition, many components need to be completed before the first Puzzle Room is ready to play. The Game was developed using Github repositories and task boards, Unreal Engine 5 and Visual Studio 2022 to create custom C++ classes for the games assets. The project has had many changes made to its development plan and list of features to be implemented. Lastly this document is also here as notes for a future development team that may be considering picking up this project from where it has been left off.

Contents:

Introduction:

After months of development time, the project has gone through many changes, both in the game's design, and in the composition of the development team. In January one team member was lost, and in late February the entire development team was disbanded, leaving just myself to carry on through the project. From January to the disbandment, I was having some kind of compatibility issues with the project when updating my version of the software to include the team's changes at the time. Because of this lack of access to those existing assets, I eventually chose to rework the software completely from scratch.

As of now, the game is still in a preliminary state, many of the game object interactions have not been implemented yet, and the ones that have been are not functioning properly. Specifically the posters in the room are supposed to change the image they display when the blacklight flashlight is shining on them and for whatever reason this is not the case. The holders that the small spherical statues are supposed to be placed in currently do not receive the

statues. The game's hint system does display text as needed but since the puzzles have not been fully implemented there are no relevant hints to display.

What does work is the inventory system. Items can be picked up by clicking on them while they are close enough to the player. The player has an inventory space that allows for up to five items to be carried at once. The player can scroll through their inventory with the scroll wheel of the mouse, and the equipped item will appear in front of the character's perspective to simulate holding the item. The blacklight item itself functions as a light source as well. When equipped a spot light effect is emitted from the flashlight model, and shines a black colored light where the player is looking. All the structure object's collision has been modified to be more complex, so that non-stationary objects and items can be set on the shelves without being ejected by the shelving's collision. The character controller I have written also works well, it is what enables the inventory system to function properly and also allows the player to move their character throughout the game world.

Lastly, the Save game feature functions properly. The game records the location of the character controller and is accessible for the game to load it again, the player also seems to retain their inventory, though more testing is required to verify this. The Load game feature has a minor issue. When loading the save, the player is taken to the save's coordinates, and they seem to have the same inventory they had, however, the player is shrunken down to about half their original size. The cause of this is currently unknown. Interestingly, the held items can still be scrolled through and are still held in front of the player at their new scale, though they are at about floor level.

Creation Systems:

The software's source code is hosted on the website Github. This is a very accessible and reliable way to manage the development process. It also likely contains various other states of the project in earlier phases of development.

To develop the software Unreal Engine 5 was primarily used. This seemed to be a very good solution, it is free to use and has a lot of documentation and tutorials online to use. It can have a pretty high learning curve but once some familiarity has been gained, it can be utilized quite effectively to design UI, item properties, level collision, character models and properties, object animations, and more.

Used with Unreal Engine, Visual Studio 2022 was also used. This was included for the purpose of enabling the usage of custom C++ blueprint classes. This is useful because it allows the developer to write their own functions that game objects can call within Unreal. Naturally, this requires considerably more technical skill and knowledge than using Unreal Engine on its own but it also allows the developer more freedom and flexibility with what they can do with the objects and actors in their games. There are other semantics that make using this software difficult to use, however. Building the project takes a considerably long time to finish, Unreal Engine must be closed in order to make the changes to it and somehow it seems to use more of the development computer's resources than Unreal Engine.

Development Plan Deviations:

The Development Plan went through numerous deviations throughout the entire development time. Shortly after the first plans were made, a team member dropped out of the

project. From there the team carried on with little changes made to the development plan. The team experienced setbacks with compatibility issues between updates of the project, as well as general computer access issues for another team member. As issues with the team's computers went on, delays were slowly being felt and eventually the team was disbanded. This is the largest deviation that occurred during the project's development. From their on the project was worked on by one person alone, and due to the compatibility issues, the project was reworked completely from the ground up. This caused many assets to be remade, and slowed down the production of the software severely. Many features were cut, such as a lot of sound design, a second Puzzle Room, an option to customize the controller mappings for players, a hub room of sorts to choose which Puzzle Room to begin, an introductory area for informing players on controls and actions. The checkpoint system was also cut, and the timer didn't get the chance to be implemented.

Challenges and Solutions:

The compatibility challenge was very odd, other team members could access and use the initial version I pushed to the repository, but when I went to pull their changes, there would be files missing and the software wouldn't allow me to access many assets that were there. This had a strange solution of disbanding the team and working on the project as a team of one. This led to the scrapping of what was there and rebuilding the project from scratch. From then on there were no more issues accessing assets and version changes.

Another challenge arose after recreating the project. The tutorials and documentation that were followed were for the previous version of Unreal Engine 5. (UE4) From the beginning the two versions looked very similar and there weren't many problems until it came to implementing the C++ classes and using some blueprints in general. The solution came from a lot of troubleshooting and searching the internet for solutions and more how-tos. Many components of Unreal Engine 5 have gotten more complicated since 4, and in many cases they seem unnecessary for the uses in this project but in more ambitious projects than this one they are likely much more useful. These changes also led to a lot of confusion of implementing self made C++ classes, because using the functions made by the development team was different than shown in the UE4 tutorials.

Project Continuance:

Features that are still missing or are not functioning properly include:

Timer - important this is the main source of conflict in the game.

Load Game - important the entire player's perspective is inconsistent after loading the game.

GUI - very important, the only way the player knows for sure what they have is by cycling through their inventory. This could cause a lot of confusion if they are holding duplicate items, especially if they are occupying adjacent spots.

Statue Pedestal Functionality - important, this is the solution for the room's first puzzle.

Poster Functionality - important, this is the solution for finding out the order in which to place the sphere statues.

Wall Button Functionality - Important this is where the solution for the room's second puzzle is implemented. It is supposed to be interacted with to relay a message in morse code.
Sound Design - important, this would be the clue to using the wall button to advance the game.
Custom Controller Mappings - not very important, accessibility feature that is not completely necessary, could make things easier for players with disabilities, however.

These missing features definitely show that the project is not in working order. To support these missing features, some troubleshooting needs to be done for all posters so that they interact with the blacklight properly. More C++ classes need to be added for functionality with the pedestals and button. These new and pre-existing C++ classes need to have their blueprints arranged to get the most out of the functions added to the custom C++ classes. Additional testing should also be done for unintended uses of existing functions, and even for detection of cheating software and how the game handles when changes like cheats are added. A considerable amount of work remains to be done from a team or individual that wishes to pick up where this project left off.

Conclusion:

Throughout this capstone experience I have definitely learned a lot about what the development process looks like and what the technical writing aspect of the process looks like. It is rare in typical classes to work in a team, and that is a valuable experience to have, considering how large development teams can get for building nearly all forms of software these days. Additionally I've learned about how much work goes into building a video game, and what it must be like for so many developers to be working in such a tight time crunch, it is a very common topic of discussion in the gaming industry. Lastly I've always been interested in video game development, and I am most grateful for this learning opportunity, I have a good idea on how to begin a project on my own and how much time it may take to get anywhere meaningful with it.