MULTIPLAYER FIRST-PERSON ADVENTURE GAME POWERED BY UNITY ENGINE

—----------

A Capstone Proposal

Presented to the Faculty of

CST 499 at

California State University, Monterey Bay

—----------

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science

in

Computer Science

—----------

by

Alonzo Machiraju

—----------

Advised by Jesse Cecil

Summer 2022

EXECUTIVE SUMMARY

Multiplayer First-Person Adventure Game Powered by Unity Engine

by Alonzo Machiraju

Bachelor of Science in Computer Science

California State University, 2022

The purpose of this project is to demonstrate Unity's networking libraries with movement, player interaction, non-player character (NPC) interaction, and environmental interaction replicated across a network. This game will initially target Windows and macOS with future deployment to Linux as well as iOS and Android. In addition, a WebGL deployment will be used to show the project on my CSUMB portfolio website. While this project will not be made for commercial purposes, it will be used as a basis for future commercial projects to be sold on digital storefronts such as Steam, Windows Store, Mac App Store, iOS App Store, Google Play, etc.

To expedite the process, the Unity game engine by Unity Technologies will be used to build the project. Unity uses C# scripts to define the behavior of in-game objects, which are placed into scenes through a built-in visual editor. These in-game objects, which are known as GameObjects, can then have multiple components added to them including C# scripts. All of a GameObject's properties, such as its position, rotation, scale, etc can be accessed through C# code. In addition, Unity provides graphical rendering subsystems, networking tools, map navigation, and other important tools to allow the end-user to focus on game state management and other behavior.

TABLE OF CONTENTS

Introduction

PROJECT NAME AND DESCRIPTION

Unity is the world's most popular game engine across multiple industries, with a particular lead in the mobile space. This is a major reason why it dominates the indie game space alongside Unreal Engine and Godot, due to the sheer complexity of creating a custom game engine for small and solo developers. It is also a major reason why Unity powers many revivals and reinterpretations of video game genres from the past that are no longer considered commercially viable by the AAA industry despite considerable dedicated fanbases.

This game project will apply the aforementioned strategies to an adventure game emphasizing storytelling alongside interaction with NPCs and puzzle solving. While combat elements will be present, they will not overwhelm the game's storytelling or puzzle-based elements. These puzzles will be based around combinations of environmental manipulation in the form of moving elements around, trading items with NPCs, and negotiation within dialog trees. Some puzzles may allow multiple solutions, such as cases where a key to a specific door can be acquired either by trading items with an NPC for it, holding up the NPC with a weapon, or killing an NPC and then raiding them for the key. Some paths will also affect the other elements of the gameplay, such as records of holdups and killings of neutral NPCs increasing hostility of other NPCs to a particular player.

END USERS

The target audience consists of two demographic groups. The first is Xennials, people born at the cusp of Generation X and millennials and thus are likely to have fond memories of classic LucasArts adventure games such as Monkey Island alongside an appreciation for classic first-person shooters such as Quake. They also are likely to remember the original Myst fondly.

The second demographic is Zennials, people born at the boundary between millennials and Generation Z. They are likely to enjoy some level of methodicalness while also having a strong affinity for crafting mechanics due to the influence of Minecraft. At the same time, they are likely to have fond memories of 3D Legend of Zelda titles on Nintendo consoles. In addition, the expected gender breakdown of players is not likely to skew significantly either way, as opposed to how first-person shooter games tend to skew predominantly male and puzzle/casual games tend to skew predominantly female.

This game will be made available on my CSUMB portfolio site, both as an in-browser experience as well as a downloadable game for macOS and Windows. In addition, source code will be available on my GitHub. Future games based on this project will be distributed commercially on Steam and other digital storefronts such as Apple's Mac and iOS App Stores and Microsoft's Windows Store.

Users will play the game using a mouse and keyboard using standard WASD controls as well as additional keys for additional behaviors. On-screen tooltips will guide players to prevent confusion during encounters with enemies, inventory management, NPC interaction, etc. They will be able to create a host game using a UI button and will also be able to automatically join an existing host to play in a cooperative manner.

Feasibility Discussion

From the environmental scan, I have deduced there is market potential for the type of game that I have outlined, especially as a multiplayer experience. Its benefits are primarily of a recreational nature, though educational applications can be conceived as a potential use case for any derivative project. The fact that there may still be potential for single-player games based on this project is good, since there is a chance that it may be delivered as a single-player experience.

Overall viability of the project is likely to be enhanced by implementing it as a single player game for the purposes of this project. This is because even though it's viable to build as a multiplayer game, some major technical challenges have developed with the way host servers and clients interact with each other. Moreover, Unity's NetCode framework is not supported in WebGL builds and thus any version embedded within my portfolio site will have to be single player anyway In the interest of meeting deadlines, the deliverable is likely to be changed to a single player experience in order to focus on other aspects such as combat mechanics, inventory management, and conversation trees. Nonetheless, lessons from this experience will be applied to developing new games based on this project in the future for commercial

Among ethical concerns, this project's largest one is disability access due to being a video game. This is especially critical when addressing vision impairments. Some vision concerns such as color blindness and low visual acuity can be accommodated with color coding on multiple axes, icons recognizable as monochrome cues, and variable font sizes. However, total blindness is almost impossible to accommodate unless a game is designed from the ground up to rely entirely on spatial audio cues. Moreover, deafblind accommodation presents and even greater challenge due to the need for specialized interface hardware based entirely on touch. As a result, accessible design possibilities are limited in video game design. This is especially true for multiplayer games, since a player without sensory disabilities will inherently have unfair advantages no matter how many alternative cues are baked into the design of the game.

Limb differences are another important accessibility consideration, though they are more likely to be addressed by specialized hardware than software design itself. Examples of such accessible hardware are adaptive controllers designed to take input from those with limited

mobility as found in cerebral palsy as well as limb differences. Future advancements could entail commercializing direct brain-computer interfaces based on reading brainwaves.

Design Requirements and Usability Testing

PLATFORM

This project is made using the Unity Game Engine. Unity was chosen over Unreal due to a higher level of familiarity despite Unreal offering superior flow in some aspects. In addition, Unity offers a larger variety of target platforms than Unreal including WebGL for deploying to web browsers. This allows for easier integration into my portfolio site. Lastly, I have more experience with Unity including deployment of products.

MAJOR FUNCTIONS

This game is designed to feature standard movement patterns alongside inventory management, item gathering, item-based puzzles, interaction with non-player characters (NPCs), and some basic combat. Multiplayer support is present in the codebase, but will not be deployed in this project due to technical scope concerns. These include bugs that came up in relation to Unity Netcode as well as Unity's WebGL player not supporting Netcode.

USABILITY TEST PLAN

The plan is still to test a game prototype on a small set of local college and graduate students. However, if logistical issues arise, I may cast a net that goes to a general age band without necessarily relying on academic institutions. Usability testing will consist of evaluating users as they play through the game, answering any questions they may have. This will help with refining any future tutorials as well as refining gameplay mechanics.

USABILITY TESTING/EVALUATION

After a few weeks of delay, I was able to test a build of this project on my main point of contact. Upon evaluation, he suggested that I rework the combat system to focus on jumping to defeat enemies as per platform games. He also noted that there was a lack of functionality with the conversation system as well as acquiring items for the inventory. After testing, parts of these issues were traced to raycasting bugs that are still being investigated despite being partially resolved.

TIMELINE/BUDGET

Due to technical issues that came up with the multiplayer system, it was abandoned in favor of a single-player approach while supporting Unity NetCode in the future. Some aspects of the project had to be pared back, but the biggest issue remains getting the inventory and conversation systems to function as expected despite being off the shelf solutions from the Unity Asset Store. The combat system has to be pared down to focus on jumping to defeat enemies as per platform games due to continued difficulties in weapon tracking upon pickup. Nonetheless, a stripped down version of the project will be delivered on the portfolio website.

Budget-wise, the project has not had any impact. No third-party assets were purchased, since all assets used are free. This also applies to cases where third-party code is used alongside custom code. As a result, many licensing issues are reduced and/or eliminated when using this project as a basis for future commercial projects.

FINAL IMPLEMENTATION

Due to aforementioned technical issues that came up with the multiplayer system, it was abandoned in favor of a single-player approach while supporting Unity NetCode in the future. Moreover, the character selection system was dropped, and the decision was made to focus on

movement, inventory management, and a conversation engine along with a combat system. Due to outstanding bugs, the combat system will be reworked to operate based on jumping to defeat enemies as per platform games. This game will solely be distributed as a WebGL demo within my CSUMB portfolio website.

DISCUSSION

Overall, this project was found to be quite complex for an 8-10 week project, though as a basis for prototyping future projects it still provided value. Commercial projects can still be made using custom artwork and additional code alongside the existing implementations of custom code and code from free Unity Asset Store packages.

Appendix

Because this was a solo project, Alonzo Machiraju handled all aspects of the project's planning including targets for each week. He also handled deciding between custom code for the conversation engine and inventory system, planning the input system, and reworking off-the-shelf assets to use the new Unity Input Manager system. He also selected third-party art and audio assets while also making sure that all third-party assets in the project remain free.