

**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

Summer 2022

Copyright 2022
by Alonzo Machiraju
All Rights Reserved

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

EXECUTIVE SUMMARY.....	3
------------------------	---

LIST OF FIGURES.....	5
----------------------	---

PART

I. BACKGROUND AND APPROACH

A. Introduction

1. Project Name and Description

B. Goals

C. Objectives

D. Environmental Scan

E. Approach and Methodology

II. SOCIAL AND LEGAL CONSIDERATIONS

A. Stakeholders

B. Ethical Considerations

C. Legal Considerations

PART I

BACKGROUND AND APPROACH

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.

ISSUE: MARKET SHIFT TO MULTIPLAYER-CENTRIC DESIGN AFFECTS INDIE GAME DESIGN PATTERNS

In today's environment, there is an emphasis on multiplayer game design alongside a dearth of cooperative experiences other than team-based combat. While this appeals to many younger and many competitive players, there remains a dearth of cooperative gameplay options when compared to a quarter century ago. While there have been some experiences designed for novel forms of cooperative play from time to time, they remain outnumbered by games centered around multiplayer team combat. Thus, a potentially important and lucrative market niche exists.

Cooperative multiplayer can encompass shared problem-solving in terms of in-game puzzles. This can encompass movement puzzles such as those in Portal, or even the combination of combat and puzzles found in Half-Life: Decay in Half-Life's PS2 port. There was even a time when first-person shooter games allowed cooperative multiplayer in their single player campaign missions. These forms of cooperative play are synchronous, as opposed to turn-based multiplayer as found in Super Mario World's two-player mode.

SOLUTION: FIRST-PERSON MULTIPLAYER EXAMPLE GAME

Items will be stored in two categories, with a general inventory for items other than weapons and specified weapon slots for specific weapon categories. This is a way to require strategic decisions when choosing a weapon of a particular category while also pushing the game's emphasis towards its non-combat elements. It also allows for the general part of the inventory system to be implemented as a variable grid without requiring an emphasis on item weight. This is because weapons tend to weigh a lot more than other types of items and thus separating them from the general inventory allows for the general inventory to function solely on a grid with a single slot per item of equal weight.

Network elements are to be handled by Unity's Netcode library even though it's officially still considered pre-release. This is because Unity's existing network architecture is deprecated

and will be removed in Unity 2022. Moreover, Netcode is expected to become finalized during the course of the capstone project, making it easier to deal with any issues that come up if the capstone project uses Netcode from the beginning.

For the sake of accessibility, the capstone project will be deployed to WebGL in order to be playable on my CSUMB portfolio website. Depending on the terms of use for any art assets used in the game, it may or may not be released as an open source project on GitHub. In addition, native executables for macOS (x86 + ARM), Windows, and Linux may be offered for download depending on the size of a built application.

Art assets will generally either be free assets or custom-made by me in Blender. However, initial builds will use default capsules and assets provided by Unity Technologies for the purposes of rapid prototyping. The most likely custom assets will be player characters in order to provide a small choice of avatars to players. There may also be a small set of custom NPCs for the sake of art style consistency, but they will be at a lower priority level than custom player characters. In any case, all custom assets will be designed such that there will be texture atlases for entire categories of models (such as a single texture atlas for all player options and another texture atlas for all NPCs) in order to minimize draw calls for possible future mobile deployment. Any humanoid assets will depend extensively on animation retargeting and reuse via Mecanim in order to save time. The reason for focusing on rapid prototyping is to focus on building a core codebase that can be used across multiple games with very few changes to the codebase itself. Moreover, any games based off of this example project for public distribution will use custom assets and/or different Asset Store assets to create a different and consistent feel for players.

EVIDENCE OF NEED

We see from aforementioned market trends that opportunities for genre revival and revision in the indie game space have created an opening for blending multiplayer co-op with aspects of adventure games. We also see that modern multiplayer trends engage in one space or the other in most cases. We also see that this project can provide a basis for many games that tell vastly different stories while sharing core mechanics, thus saving a large amount of time and effort in indie game development.

OBJECTIVES

The main goals of this project are the following:

- Leverage Unity's Netcode for GameObjects library instead of legacy networking code
- Players can enter a network session and be dropped into a game lobby between sessions
- Players may drop into a session on invitation
- Player movements are propagated across the network, as are item interactions and any environmental changes

GOALS

- Create a first-person player prefab with networked movement code
- Allow multiple players to explore an adventure game setting featuring items, inventories, combat, and character dialog
- Establish a core codebase that can be used to build multiple future projects for wider market distribution within the adventure game genre

ENVIRONMENTAL SCAN

There are many multiplayer games in the market today, ranging from first-person and third-person shooters to collaborative construction games and many more genres. There are even

games that combine elements of multiple genres, such as Fortnite's combination of construction and crafting with third-person survival shooter mechanics. Some popular examples include:

- Minecraft is an extremely-popular game centered around creating things within a blocky world through carving it and mining resources. People can collaborate on minecraft servers and build things out of different materials mined from a procedurally-generated world of cubic voxels.
- Fortnite, as mentioned above, is a third-person arena shooter where players aim to be the last one standing at the end of a round. The environment supports crafting from finding resources and allows different kinds of buildings and structures to be created. Unlike in Minecraft, buildings are built from panels defined in the game's code, and Fortnite does not use voxels for the environment.
- Xonotic is an open-source first-person arena shooter based on a highly-modified version of the Quake engine. It is notable for its extremely fast-paced gameplay.
- Portal is a first-person puzzle game that features multiplayer co-op for its visuospatial challenges based on placing portals on walls. This game stands as one of the few examples of cooperative multiplayer in the 21st century that isn't based on construction or team warfare.

The proposed multiplayer adventure game is meant to provide a collaborative environment in which players take part in a story. It is a revival of previously-mentioned approaches to multiplayer and also aims to combine storytelling with gameplay in a manner reminiscent of of adventure games from the mid-1990s while also incorporating combat elements into a first person perspective like many modern action games. This project is designed to stand out from a

marketplace where multiplayer experiences don't often emphasize narratives and where narrative-focused indie titles tend to de-emphasize or neglect multiplayer.

PART II

SOCIAL AND LEGAL CONSIDERATIONS

STAKEHOLDERS

As with any software project, there are internal and external stakeholders. Internal stakeholders are those directly involved in its creation, while external stakeholders are those who will be directly affected by the project. There may be some who qualify as both kinds of stakeholders, such as those who will be both directly involved with the project's creation and will also be directly affected by it.

This project will be used to fulfill graduation capstone requirements at California State University, Monterey Bay. However, it will also be used as a base for future independent game projects to be sold on digital storefronts such as Steam, Mac App Store, Windows Store, etc. Thus, the main stakeholders are myself as a student at CSUMB, CSUMB as an organization, and the broader gaming public at large. As a student developer, I am a 33 year-old mature student who is pursuing a Bachelor of Science Degree in Computer Science to fulfill graduate study requirements while improving professional standing alongside prior technical and professional experience.

CSUMB, as an institution, has a strong interest in student success. In the case of the online BS CS completion program, CSUMB has a demonstrated track record of innovation and also has a strong interest in creative student projects. Because CSUMB has seen some outlier capstone projects in the past including a mobile application that generated significant revenues, it

is in CSUMB's interest to see a project that leverages a student's prior experience and knowledge in a widespread industry tool such as Unity.

Extra stakeholders include end-users within the gaming community at large. In particular, 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.

ETHICAL CONSIDERATIONS

As a video game, this project's largest ethical concern is disability access. 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 an 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.

LEGAL CONSIDERATIONS

Anytime a software project uses third-party assets, such as 3d models, textures, images, and audio files, there will be considerations pertaining to usage rights. This is because some assets may require royalties when used in a project, or at least credit to the creator. Some others may even be available under open-source licenses such as the MIT license, or even copyleft licenses such as GPL. In the case of GPL and other copyleft licenses, there will be legal conflicts when combined with royalty-based licenses and other licenses that prohibit copyleft distribution. As a result, GPL-licensed assets will not be used. Moreover, licenses requiring royalty payments will not be a factor in this project because it is for free distribution and thus royalty payments are not a factor. In the case of licenses requiring credit, a credits screen will be provided to fulfill the requirement.

PART III

PROJECT SCOPE

TIMELINE AND BUDGET

The project shall be implemented according to the following timeline:

- Week 1: Basic Netcode project
- Week 2: Multiple networked characters including multiplayer test
- Week 4: Inventory system and some item-based puzzles
- Week 6: Conversation system for NPCs including some negotiation puzzles
- Week 7: Enemies with combat
- Week 8: Character selection screen

Milestones for the project include:

- Creation of a basic project that demonstrates Unity Netcode functionality
- Implementation of networked character movement using Netcode
- Networked pickup and inventory system
- Development of simple, network-enabled in-game puzzles
- Creation of a conversation tree system for NPCs
- Implementation of enemies and a combat system
- Creation of a character selection system for players

The resources required to complete the project are as follows:

- A computer with a medium to high-end graphics processing unit. This requirement will be fulfilled by a gaming PC that I have already built as well as my 13” M1 MacBook Pro.
- The Unity editor, which has already been installed on both systems.

RISKS AND DEPENDENCIES

The main risks to the project are unexpected complications with Unity's Netcode library. This is a very important risk to consider because Netcode is an extremely important dependency. The risk is noteworthy because Netcode is still considered a preview feature, to be finalized in Unity 2022. While Unity 2022 is no longer considered alpha software, it is not considered Long-Term Support (LTS) and is thus not considered fully stable. Instead, its current state, known as Unity Tech Stream, is designed for getting user feedback on new features so that they can be refined for the LTS release. This applies to Unity Netcode very strongly.

PRELIMINARY USABILITY TESTING PLAN

Usability testing will be carried out in two stages. The first phase will be internal testing where I will run one executable instance alongside the Unity Editor to test all multiplayer functionality in addition to playtesting all inventory, weapon, and character interaction subsystems on my own. The second stage will consist of a small sample of local college students playing my game and testing it for user testing. This will allow the game's usability to be tested by third parties and will also provide fresh perspectives on gameplay challenges.

FINAL DELIVERABLES

The final deliverable will be a browser-based multiplayer adventure game, deployed from Unity to WebGL. In addition, Windows and macOS native builds may also be delivered for download based on application footprint. The source code may or may not be made available on GitHub depending on the specific licensing terms of any third-party assets used. The WebGL client will be available to play on my CSUMB portfolio website, hosted on AWS.

APPROACH AND METHODOLOGY

The project will use Unity's Netcode library, a new networking library created by Unity Technologies to replace their old UNet library, to handle network replication and remote

procedural calls for players and their interactions. In-game art assets will be a combination of free Unity Asset Store and custom 3d models made using Blender and textured with Photoshop. Audio assets will be a combination of free assets and custom sounds recorded using Audacity or composed with GarageBand by Apple. Voice chat features will not be used because Vivox, the solution provided by Unity Technologies, does not support Apple Silicon (ARM) and has other problems when being built for macOS on ARM.

Exploration of Unity's NetCode architecture has already begun, and its usage will be combined with the student's prior Unity development experience in order to build an adventure game experience. Networking remains the most complicated aspect of the project, though some basic tutorials have already been followed to build a basic networked application. Unity's own examples will be studied in detail in order to get an understanding of how to handle networking alongside player inventories and statistics. The networking elements of this project will be more complex than my previous experiences with developing and deploying Unity projects, but having prior experience with Unity and deploying on iOS and macOS will be extremely helpful and beneficial to the completion of this project. This also applies to my prior experience with Blender when creating assets such as characters for use in Unity. Thus, this project shall be viable to deliver over the summer. This is especially true because the project is a starter project for creating a networked multiplayer first-person adventure game, as opposed to a full commercial project in and of itself.

TEAM MEMBERS

This project is a solo project, therefore the team consists solely of myself. However, because Unity Asset Store assets may be used for some parts of the project, the team may include the developers of third-party assets in the credits section.