Dissertation Project –

Networked Multiplayer Application using Unity and Photon Cloud

Table of Contents

[Introduction 2](#_Toc157602496)

[Inspiration for development 2](#_Toc157602497)

[Game Engine 2](#_Toc157602498)

[Choice of Networking library 3](#_Toc157602499)

[Development Stages: 3](#_Toc157602500)

[Testing Approach 4](#_Toc157602501)

[Implemented Features and Development Insights 4](#_Toc157602502)

[Limitations 5](#_Toc157602503)

[Conclusion and Future work 5](#_Toc157602504)

[Kanban 5](#_Toc157602505)

[Commits via unity cloud 5](#_Toc157602506)

[Poster 5](#_Toc157602507)

# INTRODUCTION

This document will provide a comprehensive overview of the developmental stages and the inspirational elements behind the creation of my Multiplayer Networked application, utilising Unity and the Photon networking libraries. It aims to offer insights into the intricate process of bringing this networking project to life, shedding light on the creative journey and technical milestones achieved throughout the development journey.

# INSPIRATION FOR DEVELOPMENT

The inspiration of my game stemmed from the ideas of other captivating titles that had captured my interest before and during the development process. One of these influential games was the recently launched Super Mario Wonder, along with other notable platformers like Celeste. These titles inspired me to embrace their design choices and embark on creating a platformer that blended the realms of 2D and 3D elements.

I chose to begin development of this application as I have always had a real enjoyment when it comes to platformer games and playing online with friends and felt I would enjoy contributing towards the online multiplayer gaming environment. The genre of choice being platformer and adventure type was a result of primary and secondary data sources that i had collected and curated some figures on which genre people would prefer, in hopes that I could release my project for the enjoyment of others.

A graph with purple bars

Description automatically generatedI considered various popular online multiplayer genres, including Action-Adventure, First-Person Shooter (FPS), Role-Playing Games (RPGs), Sports Simulation, Platformers, Real-Time Strategy (RTS), and Survival Horror. After conducting an online survey, it became evident that platformers resonated most with the audience, as reflected in figure 1.

Figure 1 Genre of Game Results

A graph with purple bars

Description automatically generatedThe survey not only highlighted the popularity of platformers but also underscored the significance of multiplayer features in gaming experiences. This insight further fuelled my commitment to delivering a project that aligns with the preferences and desires of potential players.

Figure 2 Multiplayer Statistics

A pie chart with numbers and a few different colored circles

Description automatically generated with medium confidenceTo create the most captivating applications possible I also ran a survey based on the type of multiplayer game that people would enjoy the most, 2D, 3D or a mixture of aspects. This resulted in the mixture genre being picked, which gave me the opportunity to expand on my skills and work on both 2 and 3 dimensional environments with the same characters, movement physics and networking scripts.

Figure 3 Game Mechanics

# RESEARCH ON OPTIMUM GAME MECHANICS

In order to find the most optimum features and mechanics to invlude within my game, I also did some secondary research from different databases supported on the Kaggle website. I then used some data science techniques to compile diagrams displaying which is the most popular within gaming and which networking library is the most popular.

# GAME ENGINE

To begin development of my project, I made the decision to utilise Unity as my game engine for development. This allowed me to use their inbuilt tools for creation of my application for ease of development and to ensure that I met deadlines on time and with the best quality work I could produce.

# CHOICE OF NETWORKING LIBRARY

To achieve the goal of developing a fully functional multiplayer title, I opted to integrate the Photon API and library. This choice enabled my game to establish connections among players across a network, utilising the Photon servers.

I selected this library based on my prior experience, its affordability, and the inclusion of a matchmaking API. Additionally, its advantages in handling real-time applications proved crucial for a platformer game, where players may need to collaborate to navigate through diverse levels.

# DEVELOPMENT STAGES:

Prior to delving into network implementation, I meticulously crafted the scene and sprites essential for both the environment and players. This initial step paved the way for constructing a prototype level, incorporating a rigidbody for seamless testing and integration of player sprites and movement. The scripting process involved creating dedicated scripts for 2D and 3D movement.

Having achieved a solid foundation, the focus shifted to the networking phase, marked by the following sequential steps:

1. Successful creation of a server.
2. Facilitation of client connections to the server with mutual visibility.
3. Integration of GamerTags for player distinction.
4. Implementation of a ping counter to gauge connection strength.
5. Introduction of a chat feature for player communication across the network.
6. Synchronous movement and animations between players.
7. Seamless transition to and from levels upon player completion.
8. Introduction of a lobby selection list for user-friendly navigation instead of manual lobby name input.

These steps laid the groundwork for the creation of a comprehensive multiplayer game, ensuring synchronised movement and progression while fostering team collaboration within individual lobbies. Contingency planning for server or client disconnection was also addressed. Upon completion of this stage, attention shifted to implementing anti-cheat mechanisms to enhance overall gameplay experience. Additionally, the introduction of private lobbies, accessible only through a password, catered to players seeking a more exclusive gaming environment.

Subsequently, the development focus expanded to level creation and game mechanics, maintaining the synchronisation of animations and progression among players. This phase involved building upon the initial application's framework and enhancing the overall gaming experience with diverse menu features, levels, and quality-of-life enhancements.

The final phase encompassed the release of the game and ongoing maintenance to address any issues, introduce updates, and ensure a consistently enjoyable experience for players.

TESTING APPROACH

For the testing phase of my application, I opted for the Unit Testing method to systematically evaluate each feature independently. This approach ensures that individual components are robust before conducting comprehensive tests involving all features simultaneously.

Networking Unit Testing

This testing method focuses specifically on establishing and validating the connection from the client to the server. The goal is to guarantee seamless functionality and connectivity between clients, ensuring synchronisation of animations, movements, and features such as chat and gamertags.

Movement Unit Testing

Testing the movement feature individually is crucial to ensure that players can successfully navigate through levels without encountering disruptions caused by incorrect or faulty movement physics. Addressing potential issues in movement is essential to maintain user motivation throughout their gameplay experience.

Rigidbody Check

Specific checks for the rigidbody physics are implemented throughout the title. This step aims to prevent scenarios where the character might 'fall through' the floor, requiring users to restart levels due to such issues. Identifying and rectifying these issues contributes to a smoother and more enjoyable gaming experience.  
The rigidbody check was required to be tailored towards each level within my application, this would be because of the dimensions changes as the user navigates the title, going to and from 3D and 2D levels.

Results

After unit testing was completed, I found some small changes that needed to be implemented in order to improve the overall quality of my application and make it more of an enjoyable experience to navigate overall.

IMPLEMENTED FEATURES AND DEVELOPMENT INSIGHTS

During my development journey, I strategically incorporated a diverse range of features into my application. Below, you'll find some of these features along with insights into their development stages and the associated challenges:

Multidirectional Movement - The implementation of multidirectional movement presented a significant milestone in the development process. Ensuring fluid and responsive movement in all directions was a key focus. Overcoming challenges related to movement physics and responsiveness demanded meticulous attention during the development stages.

Multiplayer Implementation – Multiplayer implementation was another massive milestone in the production of this project

Enemies -

Servers -

Text Chat and Gamertags -

Pause, Start Menus -

LIMITATIONS, PROBLEMS AND RESOLUTIONS

The limitations I experience when creating this title was the further amount of time that I could have used to improve some of the features.

A problem that I faced during the development of my project was the fact that in some cases the Photon PUD, this used up time and meant development during this stage was slow

CONCLUSION AND FUTURE WORK

KANBAN

| Task ID | Task Description | Status | Priority | Deadline |
| --- | --- | --- | --- | --- |
| 001 | Scene and sprite creation | Completed | High | 11/15/2023 |
| 002 | Prototype level with rigidbody | Completed | High | 11/25/2023 |
| 003 | 2D and 3D movement scripts | Completed | High | 12/05/2023 |
| 004 | Server creation | Completed | High | 12/10/2023 |
| 005 | Client connections and visibility | Completed | Medium | 12/20/2023 |
| 006 | GamerTags integration | Completed | Medium | 01/02/2024 |
| 007 | Ping counter implementation | Completed | Low | 01/10/2024 |
| 008 | Chat feature development | Completed | Medium | 01/15/2024 |
| 009 | Synchronous movement and animations | In-progress | High | 01/25/2024 |
| 010 | Transition between levels | In-progress | High | 02/05/2024 |
| 011 | Lobby selection list | In-progress | Medium | 02/15/2024 |
| 012 | Anti-cheat mechanisms | Pending | High | 02/28/2024 |
| 013 | Private lobbies with password | Pending | Medium | 03/05/2024 |
| 014 | Level creation and game mechanics | Pending | High | 03/15/2024 |
| 015 | Menu features and quality-of-life updates | Pending | Medium | 03/25/2024 |
| 016 | Release and ongoing maintenance | Pending | High | 04/05/2024 |
| 017 | Unity Cloud commits for version control | Pending | Low | 04/15/2024 |
| 018 | Matchmaking | Pending | Low | 04/15/2024 |

REFERANCES

https://hackernoon.com/unity-realtime-multiplayer-part-2-tcp-udp-websocket-protocols