# 

# Team Project

# Cosmic Decay



|  |  |
| --- | --- |
| Title of Project | Cosmic Decay |
|  |  |
| Student Name & Number: | Matthew O’Rourke Bourke – K00287407 |
| Student Name & Number: | Corey O'Brien – K00284626 |
| Student Name & Number: | Zach Ahearn – K00282339 |
| Student Name & Number: | Szymon Halada - K00284376 |
| Date | 02/10/2024 |
|  |  |

Software Development with Games Programming

Technological University of the Shannon (Tipperary)

Certificate of authorship

We hereby certify that this material, which we now submit for assessment on the programme of study leading to the award of [degree title] is entirely our own work, and has not been taken for the work of others save and to the extent that such work has been cited and acknowledged within the text.

|  |  |
| --- | --- |
| Name of Candidate: | Matthew O’Rourke Bourke |
| Signature of Candidate: |  |

|  |  |
| --- | --- |
| Name of Candidate: | Zach Ahearn |
| Signature of Candidate: |  |

|  |  |
| --- | --- |
| Name of Candidate: | Szymon Halada |
| Signature of Candidate: |  |

|  |  |
| --- | --- |
| Name of Candidate: | Corey O’Brien |
| Signature of Candidate: |  |

Dated: 21/11/2024

Contents

[Team Project 1](#_Toc183102979)

[Cosmic Decay 1](#_Toc183102980)

[Abstract 5](#_Toc183102981)

[1 Introduction 6](#_Toc183102982)

[1.1 PURPOSE 6](#_Toc183102983)

[1.2 Scope 6](#_Toc183102984)

[1.3 Project Members 6](#_Toc183102985)

[2 Problem Definition and Background 7](#_Toc183102986)

[2.1 Background 7](#_Toc183102987)

[2.2 Requirements 10](#_Toc183102988)

**3** [**Project Management 12**](#_Toc183102989)

[3.1 Schedule 12](#_Toc183102990)

[3.2 Communication & Collaboration plan 14](#_Toc183102991)

[3.3 Development Methodologies 15](#_Toc183102992)

[4. Technical Review and Research 17](#_Toc183102993)

[4.1 Game Mechanics 17](#_Toc183102994)

[4.2 Implementation 18](#_Toc183102995)

[4.3 Quality Assurance Plan 18](#_Toc183102996)

[5 System Design and Configuration 19](#_Toc183102997)

[5.1 Architecture 19](#_Toc183102998)

[6 Testing & Implementation 20](#_Toc183102999)

[7 Critical analysis 20](#_Toc183103000)

[8 Code 21](#_Toc183103001)

[9 User Instructions 21](#_Toc183103002)

[10 Assets 22](#_Toc183103003)

[11 Individual Contributions 24](#_Toc183103004)

[Student 1 Reflection 24](#_Toc183103005)

[Student 2 Reflection 25](#_Toc183103006)

[Student 3 Reflection 25](#_Toc183103007)

[Student 4 Reflection 25](#_Toc183103008)

## Abstract

In this document we will have various sections explaining the game, the process of creating the game, the technical review, the project management side of things and our individual contributions to the project.

We created this game called ‘Cosmic Decay’ as our first project as part of a team with the hopes of developing our interpersonal and professional skill levels. We also had the goals of creating a functioning game as part of a team as up to this point it was all an individual effort. We hope to convey in this document how we have achieved this while also covering all the aspects of the work, collaboration and the technical side of working on the project.

We will also be providing our own conclusions to the project. These will contain a quick overview of how we got on as working in a team and how we found making the game.

# Introduction

## PURPOSE

The proposed game being made is a 2D angled-top-down game called Cosmic Decay. This game takes place in space where four unlikely heroes must fight zombie aliens while trying to repair their ship and escape. This game utilizing some code from the game ‘Zombie Arena’. Many specialized classes will be utilized such as level manager, arena, enemy and player. The game arena will consist of three levels, and a shop system which the player will be able to access in between levels. This will serve as a place for the player to obtain upgrades. Each new level will have a unique appearance from the last.

## Scope

Included

* Angled top-down point of view
* Multiple playable characters
* Shop system
* Free to Play

Excluded

* Will not be a multiplayer game
* Will not have microtransactions

## Project Members

| **Team Member** | **Role** | **Contact Information** | **Responsibilities** |
| --- | --- | --- | --- |
| Zach Ahearn | Developer  Sound Asset Creator | K00282339@student.tus.ie | Design, Coding, Documentation.  Creating sounds assets |
| Szymon Halada | Developer  Asset Manager | K00284376@student.tus.ie | Design, Coding, Documentation  Creating game assets |
| Matthew O’Rourke  Bourke | Developer  Project Manager | K00287407@student.tus.ie | Design, Coding, Documentation.  Project management of the sprints, allocating tasks. |
| Corey O’Brien | Developer  Technical Infrastructural Engineer | K00284626@student.tus.ie | Design, Coding, Documentation.  Create the technical drawings for the game. |

# Problem Definition and Background

## Background

The background information for the game comes from ideas conjured from other games. Influences for this game come from other games such as ‘Returnal’, and ‘Astroneer’. These games are included here as they are in the same genre as the game being created, that of the rogue-like genre.

The ‘PacMan’ game is what will be used as a starting point for the code. This will provide some of the basis for the map code, in that the map can be changed using a text file. The code utilises a sprite sheet which is then easy to change to the intended sprites to be used in the game. This starting code also helps with having pre-made enemies that are scripted to follow the player.

A video game with a blue square with a black background

Description automatically generated with medium confidence

Figure 1: Pac Man Game

Research will be aimed at the user experience, AI, the user interface, game systems and narrative.

Hades is similar as it has the multiple levels concept, but it also has a shop mechanic that appears every few levels to improve the player’s weapons etc. Research will focus on how to make an angled top-down shooter similar to Hades.

A video game screen shot

Description automatically generated

Figure 2: Hades

Cult of the Lamb is a game very similar to Hades in the way of combat and proceeding through multiple levels after eliminating all enemies in each level. Research here will focus on the aspect of proceeding through rounds and looking at how to prepare a top-down angle in our game:



Figure 3: Cult of The Lamb

Inspiration for the narrative came from the game ‘Returnal’. Returnal is also about a person lost in space which is the basic plot for the proposed game. This game will be researched as to look at how to portray a space game/rogue-like properly in this context.



Figure 4: Returnal

A character selection system will also be utilized like the game ‘Castle Crashers’ in that the player has the option of choosing to play from four different characters. Research will be undergone to learn how to properly integrate a character selection system in the game.



## Requirements

The requirements regarding the user for this project may include: The user will be signed in as a guest user, there will be no log in function. The user will be greeted by a splash screen of the title, this also includes a main menu where the user can select between options of play game and quit game. The user can pick play game this will provide another screen to have options to pick different characters. When a character is selected, the user can then begin playing the game. The game will require the user to defeat enemies, collect fuses and progress through numerous levels. Each level will require the user to collect a certain number of fuses to progress. When all levels are completed, the user will go back to the main menu. The user also is required to press the quit game button at the main menu to exit the game.

A diagram of a menu

Description automatically generated

A diagram of a company

Description automatically generated

A black screen with white text

Description automatically generated

## Project Management

## Schedule

On the first week, our group came up with a schedule to hit all the targets of our team project. We tried to do this in a way that set realistic targets and expectations.

As time went on, we used this schedule less and less and instead, we worked down a list of features we wanted in our game.

|  |  |  |  |
| --- | --- | --- | --- |
| Week Number | Week Beginning | Deliverables | Deadlines |
| 1 | 9th September | Brainstorming ideas. Sprint board created |  |
| 2 | 16th September | Git repo created. Initial proposal Finished and submitted. Skeleton of must have files created. | 19th September |
| 3 | 23rd September | Technical doc finished and submitted. Finishing off ‘Must Have’ files skeletons and adding some of the basis to these files | 26th September |
| 4 | 30th September | Continuing work on ‘Must Have’ files,  Prepare presentation | 7th October |
| 5 | 7th October | Continuing work on ‘Must Have’ files |  |
| 6 | 14th October | Continuing work on ‘Must Have’ files, Nice to have files?? |  |
| 7 | 21st October | Continuing work on ‘Must Have’ files, Nice to have files?? |  |
| 8 | 28th October | Coding week. Objective: get a good prototype running | 4th November |
| 9 | 4th November | Demo for prototype | 4th November |
| 10 | 11th November – 25th November | Finish any remaining code, work on documentation | 25th November |
| 11 | 18th November |  | 25th November |
| 12 | 25th November | Deadline for code and documentation | 25th December |
| 13 | 2nd December | Demo and presentation | 2nd December |

We also used a website called Trello to follow as a rough structure as to what we need in the game. We put everything in order of how important the events were in game. So, we went in order from: Start event, to level all the way down to end game event.

See below for our Trello board:

A screenshot of a video game

Description automatically generated

A screenshot of a video game

Description automatically generated

Again, even with the Trello board, as time went on, we didn’t follow this board. We found it was better to discuss the importance of what files we needed to finalize and work on for the coming week.

## Communication & Collaboration plan

Communication was a key part of the project. We used discord to hold our sprint meetings every Tuesday. We found discord to be extremely useful for our communication. This was because of the functionality of creating a server and on this server, we had separate channels such as a call channel, a text channel for general discussions such as if there was a push made or to organize a call, a channel for problems, so if we had issues, we could share screenshots of our files or error codes and work on them collaboratively.

We also used GitHub to work on our code collaboratively. Here was where we pushed our code to the cloud, this enabled the other members of the group to pull the code from this repository to pull all our code together to make a functional game. This also enabled us to create our own branches to work on our code individually. It was a bit of a learning experience to get the branches set up correctly but the reason we wanted to do this was because when we merged our code to the main branch it was much easier than pulling all from the main as sometimes it would overwrite some of our code or even not work at all.

Finally, we used Microsoft’s OneDrive to work on our presentations and word docs like this technical document collaboratively while saving it all to the cloud simultaneously.

## 3.3 Development Methodologies

The development methodology we decided to use was the Agile design methodology. We decided on this one as we first, knew the most about it. We learned quite a bit about it in our games design module in 2nd year and in our taught content for this module with Jacqueline.

Agile enabled us to follow a good structure in terms of getting our work done fast and structured, but it also was very flexible in that we could alter the method to suit our needs and work styles.

Part of this methodology was to use sprint meetings to discuss work we needed to get done. Also, there is of course a planning stage for this in which we made our must have, need to have, and nice to have categories to properly assess what work needs to be done and to the importance of the files we had planned to put in the game.

Below is the board we made for these categories back at the very beginning of the projects concept stages:

A bulletin board with sticky notes

Description automatically generated

We also thoroughly tested our code. We did this by getting friends and family members to try out the game. We also added debugs to the game to check if for example the shop system works properly.

As we went along in the process our project supervisor Jacqueline Humphries looked at our code and ensured we were constantly working on the project.

# 4. Technical Review and Research

*Outline the literature reviewed and research completed to inform the completion of your project. This should include commentary on current and emerging technologies/systems, key data, information, source, code etc. which was sourced as part of your research. Where relevant indicate any conclusions or recommendations arising from your research*

## Game Mechanics

The following constructs of rules and/or methods will be designed for interaction with the game state and unique abilities, thus providing gameplay:

|  |  |  |
| --- | --- | --- |
| **Object** | **Mechanic** | **Result** |
| All Player characters | Shooting | Will fire a projectile |
| Enemy | Bite | Deals a small amount of damage |
| Health Pickup | Healing | When picked up, health is replenished by a specified amount. |
| Ammo Pickup | Reloading | When picked up, you receive 16 extra bullets to your ammo clip. |
| Fuse Pickup | Goal Pickup | The player must collect all fuses on the map before they can repair their shuttle and advance to the next level. |
| Shuttle | Shop/ Next Level | When the player has picked up all fuses, they can return to the shuttle and use points earned to buy upgrades and advance to the next level. |

We created these objects and their mechanics with the aid of the projects we have been working on up to this point. Mainly, our “Zombie” game and our “Thomas was Late” game from second year. With a bit of testing and research, we were able to implement different concepts from both games to make a good base for our game. For example, “Zombie” helped to create our aiming and shooting mechanics and “Thomas was Late” helped us the generate our levels with the use of text files.

## Implementation

We began our project by brainstorming the concept and just jumped straight into coding where we saw fit. This became a problem as we ended up running into problems where there was not a proper plan put in place. We then decided we would need to follow a development methodology to maximise or coding efficiency.

The development methodology we utilized was the Agile Development method.

This method was utilized because it is a familiar concept learned about in the Second year of Games Design and Development. It is felt by the group members and project supervisor that it is the most efficient method of planning, designing, and completing the game.

## Quality Assurance Plan

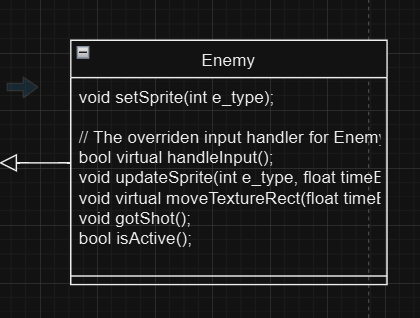
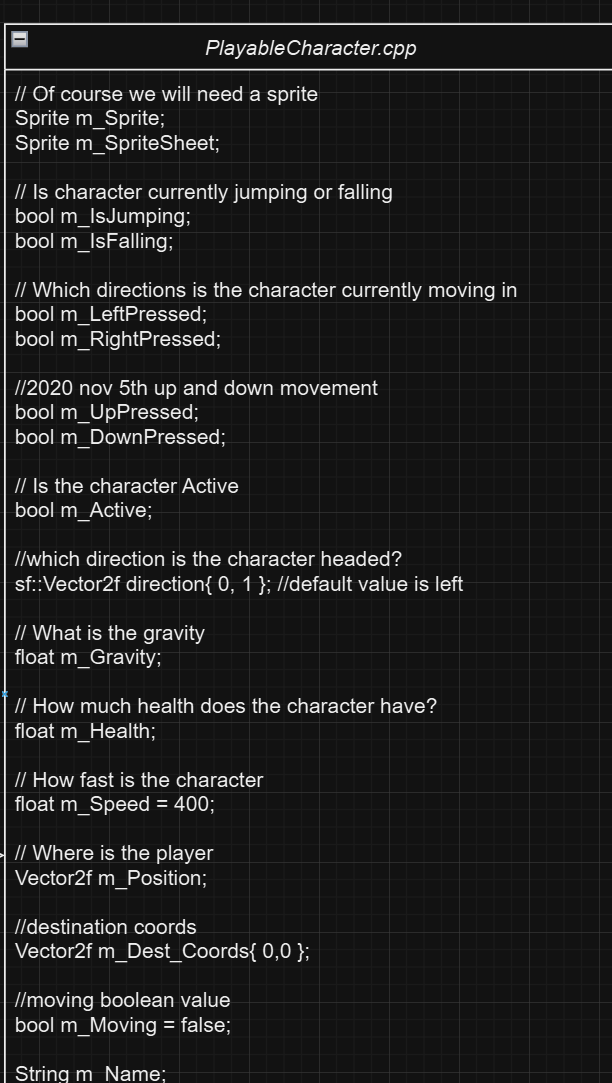
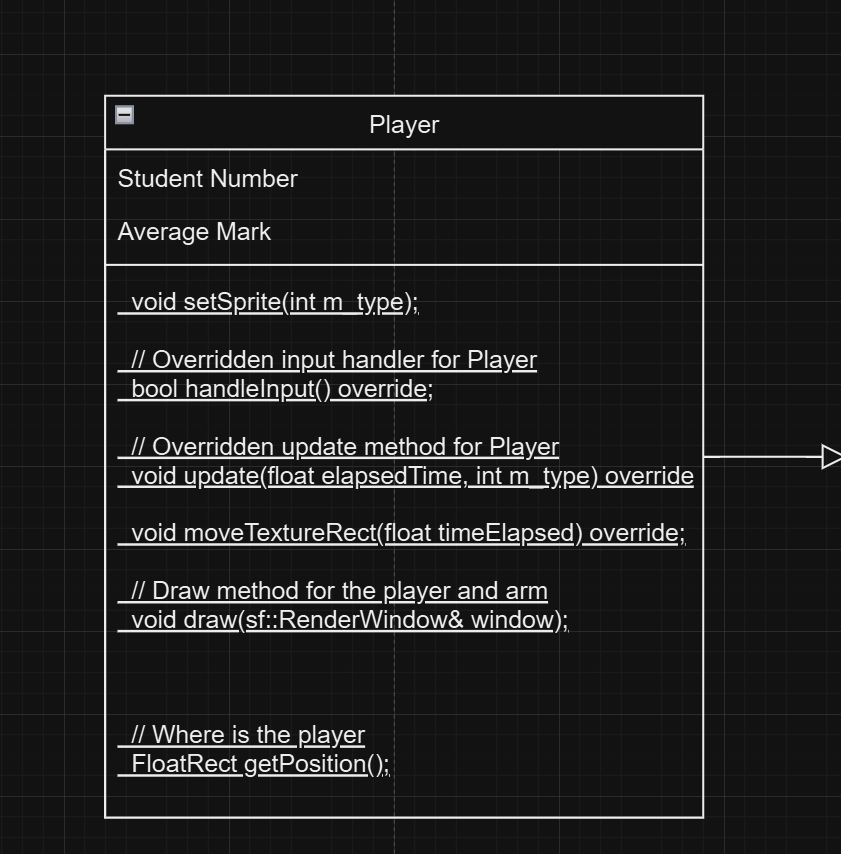
A big part of creating this game was ensuring that the quality was too a good standard. This standard had to be in place for not only the gameplay but the visuals and sounds also. To ensure this quality was to a good standard, we held regular testing of the game to find out what felt good, what didnt and what we needed to add or remove to improve the gameplay. In terms of the art assets, when Szymon, (our asset manager), finished creating a sprite to use, he would show the rest of the group, and we gave a group opinion and feedback to get assets we were all happy with. Same goes for the sound assets made by Zach, (our Sound asset manager).

We found while coding early on that if we were all working in the main branch and someone pushed their code. The others lost their progress when they pulled that push, or alternatively we had to save a copy of our changes before pulling. We learned about branches which allowed us to work on our changes simultaneously on our own copy of the main branch and merge our branches together during our sprint meeting to see if there are any conflicts and then test the full code together.

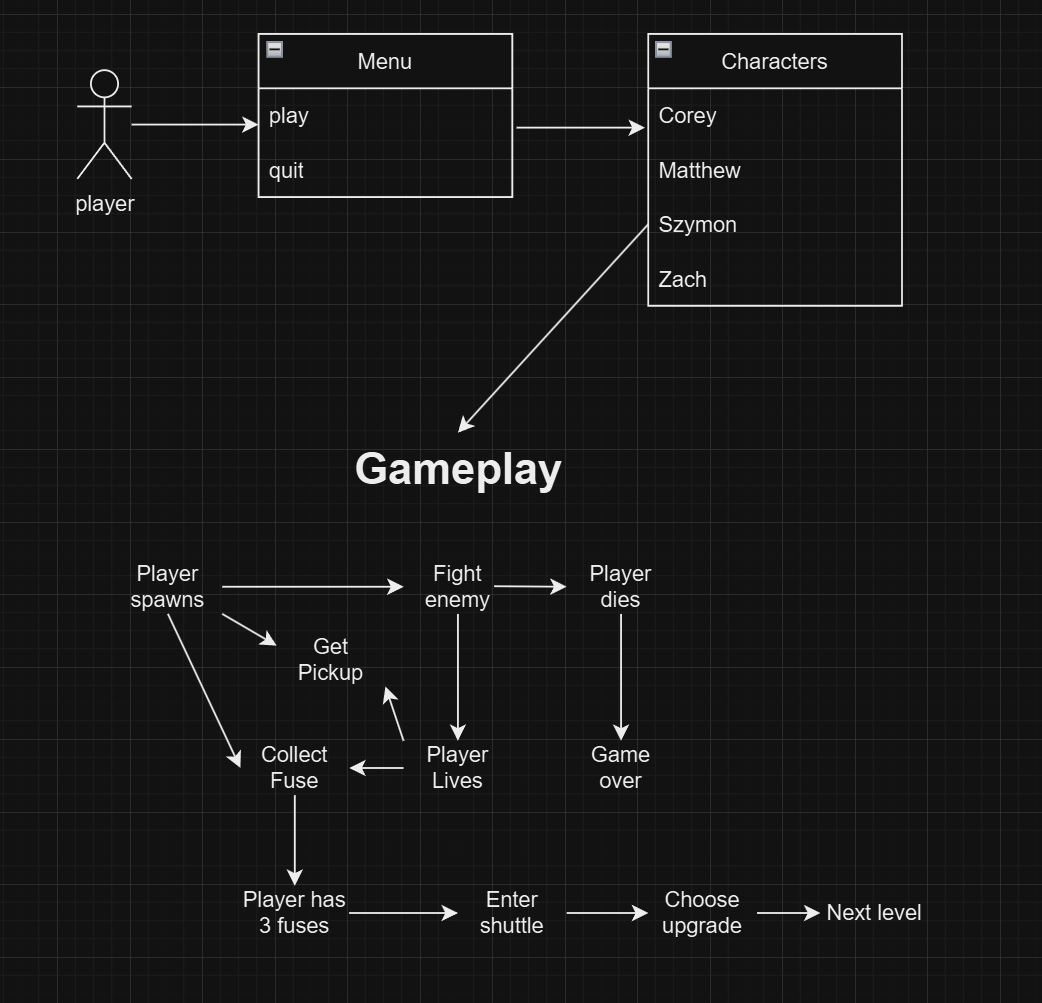
# System Design and Configuration

For our system architecture, files such as player, enemy, and playablecharacter handle character movements and interactions. Levels are handled using levelmanager and Loadlevel with collisions being handles in detectcollisions and playablecharacter where the enemies read the tiles in front of them to see if they can enter them.

The player and enemy class inherit attributes from the playableCharacter class.



To play the game, you must first go through the main menu and select a character, Then you are in the first level where you must collect 3 fuses per level to progress on to the next level while fighting enemies. Player can pick up items such as health and ammo, and upgrade stats between levels from the shop once they enter the shuttle.



## Architecture

The system requirements for this project will be: The game should be run on a windows operating system as we are unable to test on any other operating system. It will also require a functional keyboard and mouse/touchpad. The resolution should be dynamic to adapt to the user’s screen resolution. The system should also have c++ libraries and sfml libraries installed. The project will also require visual studio 2022 community edition to be installed with relevant c++ libraries. The system will also require a functional sound system, this could be built in speakers or headphones etc. to fully understand the game.

A black screen with white squares

Description automatically generated

# Testing & Implementation

After a few months of our planning, coding, and testing. We finally ended up with the final solution which we have uploaded. However, this did not come without a fair share of problems. During the beginning of the project, we had some issues understanding and implementing GitHub. We spent longer than we had hoped to get our heads around using GitHub. As mentioned before in section 4, we mistakenly worked on the main branch together at the same time whilst coding. This led to conflicts when we all wanted to push our changes and resulted in loss of progress and changes. But as mentioned before as well, once we started using the branches we started to flow a lot quicker though the coding process as we worked simultaneously without affecting the other member's progress.

We followed the following weekly plan for working on the project. Every Tuesday we would have a sprint meeting to delegate different aspect of the game to work on for the week. We would each work on these aspects in our own branches and test while we coded. Then on the following sprint, we merged our code together, resolved any conflicts and did a big group test to figure out what needed to be done for the following week. We found this to be the best method for testing and weekly planning.

As we had many ideas for our game, we had problems in implementing all of them. Most were due to time management, there were some aspects of the game we had spent too much time on which led to some ideas being left behind. Other feature we had issues in figuring out how to implement them and had to move on to focus on other features. One such feature was our arm for the player. We planned to have an arm separate to the player sprite which would rotate around the player sprite in the direction of the crosshair to show the gun aiming in that direction. However, this proved to be more challenging than we expected and due to time constraints we had to move on from it.

If we are chosen for the Games Fleadh or decide to update the game for our own enjoyment, we would love to implement some of these concepts we had to leave out. One such concept is the players abilities. Because we had a character selection, we wanted to not only differentiate the characters visually but also in terms of gameplay. Each character was planned to have a special ability unique to them, but due to time constraints and having to focus on more key features, we had to leave it behind. We had begun to code some of the abilities which you can see in the project.

# Critical analysis

Our project began with a very ambitious idea with many different features that ended up out of our reach. We had ideas to have working abilities, arm rotation, high score, random shop upgrades, more detailed levels, bosses, four playable characters, different weapons, cone of vision.

We were able to keep some of the ideas that never came to fruition in a simplified form such as the shop with a couple of choices of upgrades such as increased fire rate and even though we have the code for abilities we never got to implement them. Bosses and arm rotation ended up being scrapped as we ran into too many issues trying to implement them. High score was also scrapped. Enemies do not have a cone of vision, instead they go for the player position.

We started the project by following the timeline but ended up finding different priorities other than the timeline followed so we instead discussed new priorities every week in our calls and worked on them then which proved more beneficial for us as we needed more time discussing different aspects of the project.

The biggest take away from this project was learning how to communicate as a group to plan and achieve goals and to code together using GitHub.

## 8 Code

**[Link to GitHub – Click Here.](https://github.com/K00284626-CoreyOB/CosmicDecay.git)**

## 9 User Instructions

How do you play the game? -

Use W to move forward

Use A to move left

Use S to move backward

Use D to move right

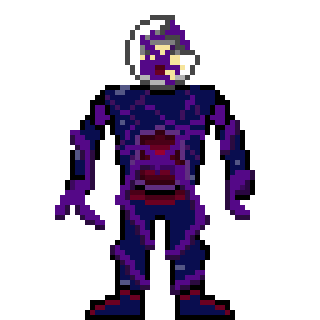
Use the mouse to aim

Use left click on the mouse to shoot

## 10 Assets



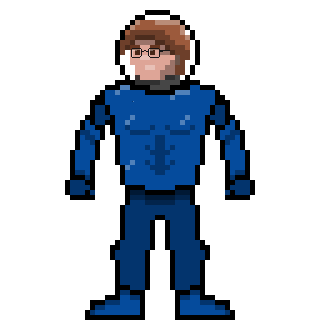
Main Menu Concept



enemy concept



playable characters



early gameplay concept

A screenshot of a computer

Description automatically generated

## 11 Individual Contributions

## Student 1 Reflection – Matthew O’Rourke Bourke

## Introduction

For my third-year team project we have recently completed, we were required to build a game using the C++ language and SFML libraries. This was to be done under the theme for Games Fleadh – ‘Small World’. I was in a group with three other people, and I was given the role of project manager. We decided on the game Cosmic Decay. Before we began work on the game, we had some classes on development methodologies like agile and learned some project management skills like having weekly sprint meetings. So instead of just getting straight into coding with no real structure we decided to adopt the Agile methodology and really knuckle down and talk though our plans, ideas and issues in the project through our weekly sprint meetings. In this individual reflection, I will be discussing how I felt I performed in a team, my lessons learned and discussing how we integrated the taught content into our approach to building this game.

## Planning and Methodology

The first challenge of this project was sitting down and making a firm plan of our goals and the schedule regarding work to be completed in this project. I learned about the agile approach in parts of second year and in our lectures for this module. Agile is a very straightforward method of managing workload but can also be adjusted as needed. As project manager, it was my job to think of ways we could integrate this methodology into our work process. In class we looked at various methods of planning. We decided on using Trello for our planning tool.

## Belbin Team Roles

When I took the Belbin team roles quiz, I identified strongly with two roles – Team Worker and Research Investigator.

Team Worker – I feel I identify with this role mainly because one of the biggest characteristics for a team worker is to avoid conflict and I feel like I try my best to keep everyone happy in the group and keep conflicts to a minimum. I also enjoy working as a team as I find it very satisfying to all get on a call or meet up and share our ideas or our progress and I do enjoy helping my teammates with problems they have.

I also identify with some of the weaknesses in that I can be quite indecisive for the big decisions in that I wouldn’t voice my opinion to loud and mostly go with what everyone else is most happy with.

Resource Investigator – I feel I also identify strongly with this role as I do enjoy looking at ideas for projects and going in depth and researching ideas and how they might be implemented. I also feel that I identify with the weaknesses in this role as sometimes I may tend to lose interest after an allotted time has passed.

## Work Log

|  |  |  |
| --- | --- | --- |
| Date | Work Completed | Work to be completed |
| 9th September | Added to initial proposal |  |
| 16th September | Added to idea board, added to technical proposal, | Technical proposal, code |
| 18th September | Reworked initial proposal, created arena.cpp and arena.h.. added skeleton code to each | Technical proposal |
| 23rd September | Completed various sections of technical proposal | Technical proposal |
| 25th September | Completed more of technical proposal.. added schedule finished section of part 2 and completed part 4 of proposal | Technical proposal references, code |
| Week of 7th October | Completed project management slides on presentation. Added load level file. Working on loading sprite for background, deleted loadBackground.cpp and .h, added new objects for engine.h |  |
| Week of 14th October | Started work on levelmanager, created objects in engine.h, continued working on loadLevel, tried to spawn in some enemies and player to test. Added a timer to the game. Created update.cpp, worked on it, added some enemy stuff to update. |  |
| Week of 21st October | Restarted project on pac man. Worked on player inputs, stats, .cpp and .h file. |  |
| Week of 28th October | Edited player spawn position created and implemented 3 pick-ups - ammo, fuse and health. Created the code for bullet collision. Updated ammo to add to stockpile |  |
| Week of 4th November | Started work on HUD, added .cpp and.h, added in some basic objects for text, set font, made some get and set methods. | Need to get the HUD to display, add some update methods |
| Week of 11th November | Got HUD to display, added updating ammo, health, new score and level hud elements. | Need to change health to be a bar rather than number and fix collision for enemies and player |
| Week of 18th November | Added a shop, requires points to upgrade characters abilities, will display between levels. Added that the enemies respawn if killed on each level. Added a block to only allow player to progress levels if they collected enough fuses. Added player death if health below 0. Nerfed enemies speed. Restart game now available upon completion. Added so the pickups clear on each level.  Worked on technical document and individual contribution | Presentation for 2nd of December to be completed. |

## Technical contribution

I would now like to discuss my own contribution to the project on a technical level. As you can see above in the work log, I outlined what I added to the project through GitHub and what I did in the documentation and presentations.

To go through in more detail; At the beginning I learned how to work collaboratively using GitHub. This was a difficult learning process for me as I did not adapt to it as quick as I had hoped, I had issues with pulling documents and code being overwritten. Merging code was also an issue I had to overcome. Eventually through trial and error I did learn how to properly use GitHub, and I feel I am now very capable of using this a tool to work and share code while working as part of a team.

Initially, we created all our code from scratch. In this first project attempt, I created a file for the arena, the level manager and the engine.h. I also added a timer to the game which would be scrapped eventually anyway. However, as we went on I had so many issues with this base code we had created which was pulling from many of our old college projects that it was decided to use our old Pac man project as a framework for our code. When this was happening I and the rest of the team made sure that we didn’t just take code that wasn’t new and upload that, almost all the code from Pac man was reworked to suit our new game and now you couldn’t even tell that they are based on the same framework.

After this debacle on restarting from scratch I began work on the loadLevel.cpp. In this file I used it to spawn in the player at a set position and the enemies of our game with their own speed variables that were based on multipliers of the player’s own speed. I also used loadLevel to spawn in our 3 pickups – the fuse, health pickup and ammo pickup. I also used Update.cpp to add functionality to these pick ups giving a reason for the player to use them. Then I began work on the bullet collision. It checks for the bullets position and the enemies to see if they intersect and will update accordingly.

I added a functioning and updating heads up display (HUD) to our game. This hud consisted of the players health, ammo count, the level number you’re on, your score and the number of fuses you have gathered. These were done in a hud.cpp and hud.h.

I also added a shop to the game which allows the player to spend their point to upgrade their abilities as they progress.

## Team performance

To speak to my own performance in the team, I feel I learned quite a bit about working as a group. This included compromises I had to make regarding the ideation process of the game.

I also learned a lot as part of my role as project manager on how to manage a group of people, these tasks include ensuring that sprint meetings take place each week, delegating files to other group members and resolving conflicts.

I feel that it helped that before the project started, we were already friends so it’s a nice step in the door to professional teamwork that might take place in the workplace. By the end of the project basing my views on the team bell curve we learned in class I would say we ended at the real team tier. This was given that we already knew each other though and I believe that if we were strangers, it might have taken more time to get to that stage.

## Critical analysis

This section I’m going to make about what I would have liked to have gone differently in the project if we had more timed or planned more efficiently.

At the beginning of the project we had so many ideas, many of which never actually made it into the game. I feel this is due to us not managing our time correctly.

To correct this, I would have liked to learn about GitHub long before we started the project as I firmly believe that GitHub was our biggest time waster on the project. None of us knew how to use GitHub properly and maybe if we had learned about it in year 2, we could have balanced time better.

## Conclusion

Overall, I feel this team project benefited me immensely. I learned so much about working as part of a team, time management and learning how to manage a group as project manager.

I do feel there were some areas including time management that we could have done better but with the time that was given to us and the fact that it was our first time working as a group I am extremely happy with how this project turned out.

## Student 2 Reflection – Zach Ahearn

# Work Log – Zach

|  |  |  |
| --- | --- | --- |
| Date | Work Completed | Work to be completed |
| 30/09/24 | Group Presentation | Fix Map |
| 08/10/24 | Creating CreateHorde.cpp | Continue fixing map |
| 21/10/24 | Rebuild project using Pacman – map fixed | Change Pacman to player |
| 29/10/24 | Centre camera | Get arm to rotate |
| 30/10/24 | Creating the player’s arm | Rotate the arm with mouse. |
| 12/11/24 | Complete level 1 layout | Begin other level layouts |
| 18/11/24 | All level layouts complete  Add sound files | Adding sound effects code |
| 21/11/24 | Sound effect have been added |  |

# Group Roll/Individual Contribution – Zach

**Introduction**

In our Team Project, the tasks I took on were those that played to my strengths, but also helped to improve myself in other areas. In this reflection, I will be talking about the specific tasks I undertook, the challenges I faced, and how I worked collaboratively to overcome them.

**Team Role**

My main roles in the project were as a Developer and a Sound Asset Creator. These roles were important in the completion of our game, and they played to my strengths due to my past projects. I have made many games during my college course and have plenty of experience with the coding aspect of the project. In terms of being a Sound Manager Asset Creator, in almost all my projects, I have either searched for the perfect sound effects online for a certain aspect of my game, or I have created a sound effect myself, using a variety of online tools. This meant these roles were very fitting for me.

**Belbin Team Role Questionnaire**

During the project, one day, my lecturer taught me and my class about the Belbin Team Role Questionnaire. She gave use each a printed-out version of the questionnaire and assigned us to complete it. After completing the questionnaire, my scores came out as follow:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CO | SH | PL | ME | IMP | TW | RI | CF | SP |
| 3 | 9 | 6 | 8 | 13 | 8 | 3 | 6 | 14 |

According to these results, my strongest role in a group is as ‘SP’ or as the ‘specialist’. I then read through the handout to see the description, strengths, and weaknesses of my role. The basic characteristics of someone who suits this role are professional, self-starting, and dedicated. I believe I can relate to these character traits because when I’m assigned a task, I like to get it done early to avoid stress later. I took this mindset into my group project and wanted to get started straight away.

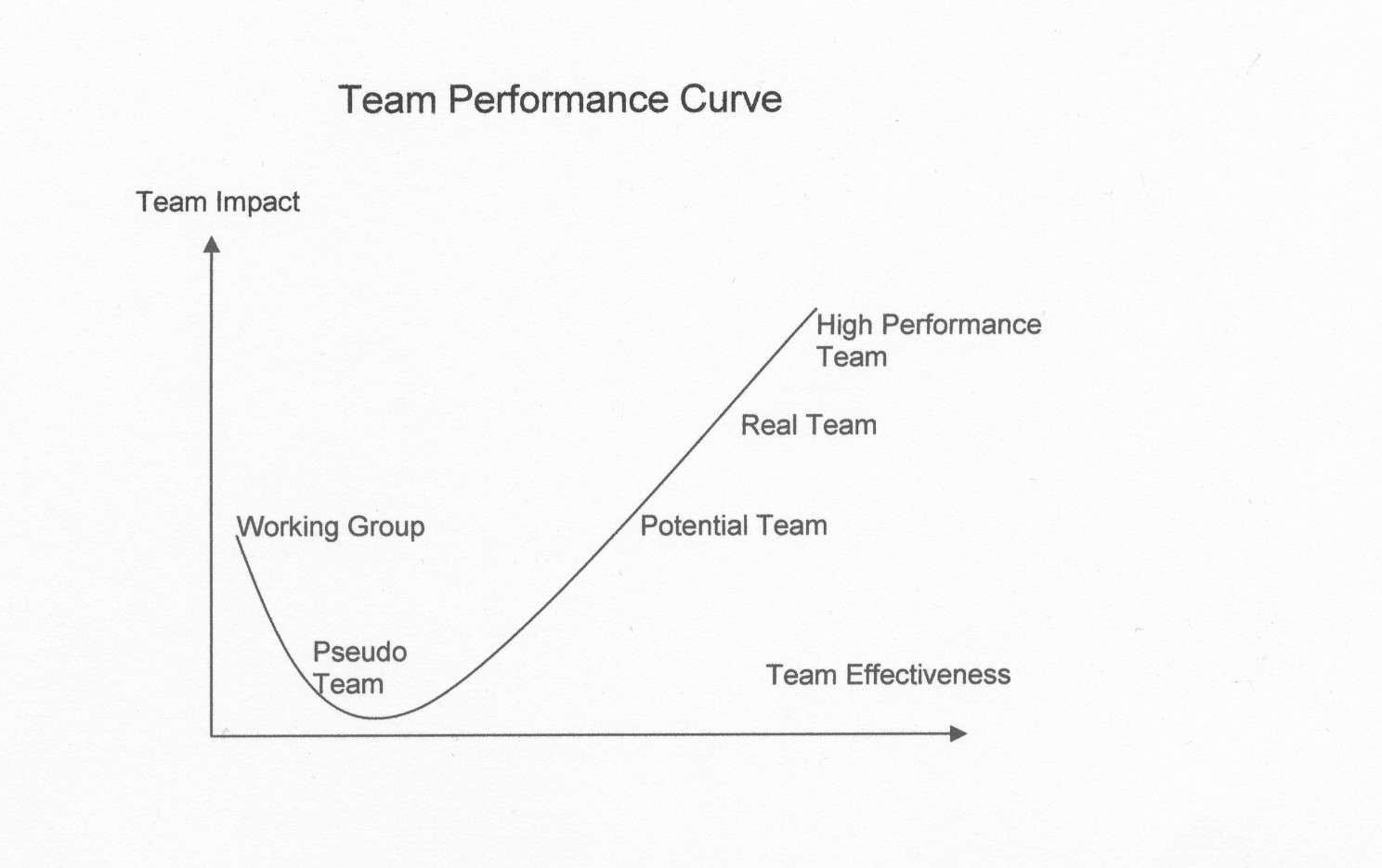
The strengths of this role are that they provide knowledge or technical skills where needed. I agree with the statement in that if a colleague is having any issues with their work and needs help, I won’t hesitate to help them if I can.

Although, the weakness of a ‘specialist’ states that they tend to only contribute on a narrow front. I can honestly say that I agree with this too. When I am completing an assignment, I tend to only do one specific part of the assignment at a time and am less inclined to multitask.

**Technical Contribution**

When my team first started on the coding for this project, we built the game from scratch while taking pieces of code from our other projects. This strategy worked for a while without errors as we all worked on creating the separate files. The problems came when we needed the game to display on the screen when ran. We could not get the game to display. One of my biggest contributions to the team was recreating the project but using the base files from our ‘PacMan’ game. This lets us start with a game that displayed to the screen, and we could change and build onto the game from there. When it came to our group working on the project at the same, I learned how GitHub worked and shared with the team so we could all utilise it. We were originally taught how to use GitHub through the control panel. Then, we learned the basics of using GitHub’s control panel in Visual Studio Code. I then figured out how to use GitHub directly in Visual Studio 2022 which allowed us to upload and share the code with one another much easier. Some of the other areas I contributed to were making the layouts of the levels by editing the text files associated with each level. I made each level about 20% smaller than the last to increase difficulty. I also created the sounds for the game which include shooting, getting hurt, hurting enemies, and many others. These sounds were acquired by a mix of me finding them online, or me making them myself. Once most of the project was completed, I implemented every sound effect into the game.

**Critical Analysis of Team Performance**,



The Team Performance Curve was shown to me by my lecturer. It was created by Jon Katzenbach and Douglas Smith to determine the effectiveness of a team by categorising them. I believe my team and I may have started as a ‘Potential Team’. This is because we were all friends starting off and wanted to work hard to achieve a high grade. However, we did need to work out our goals and a common approach. Although, I believe that throughout the project, we were slowly moving up to the ‘Real Team’ level. This was thanks to learning everyone’s strengths and weaknesses in the project as well as doing our weekly sprint meetings. Every Tuesday, I would attend sprint meetings with my team where we would discuss what we have completed and what still needs to be done by dedicating specific tasks to one another. I found these meetings to be incredibly important as they were the period where we got the most discussion and coding for our project completed.

**What Would I Do Differently**

There are some things that, if were to start our project over from the beginning, I would do differently. A big one was time management. I believe we would have benefited greatly from starting the project earlier. Not so much so that we would ignore other assignments and focus on the group project, but just that we could have dedicated more time to it. There were some features for our game that we had in mind but because we gave ourselves too little time, we could not implement them. I think we also could have done a better job at planning out the project. In the beginning, we just threw out ideas that we all liked and agreed to implement them without thinking about how complex those ideas would be to implement. When it then came to attempting the implementation of those ideas, we wasted a lot of time and effort with ideas we ended up not using in the end as we weren’t successful in adding them to our game. Another aspect of the project I would do differently would be my tracking of tasks that I completed. When I made changes to the project, I was quick to commit the changes on Git. This resulted many commits made by me in the project repository’s history. This was inconvenient to go through when researching my contribution to the project. Another side effect of this was that because I was committing changes so often, I stopped being descriptive in what I was committing. Again, this made it inconvenient to go through exactly what changes I made.

**Conclusion**

While I think there are areas that could have been improved upon for our project, I’m still quite pleased with how it turned out and enjoyed the experience of completing it. This project taught me valuable information of what its like working in a team and coding a game together. This is very important for my future as group work is likely to be important in my field of work. This project thought me a lot about object-orientated programming and how to classes can interact with one another through inheritance. I also learned about how to use GitHub and how useful of a tool it is when working on a group coding project.

## Student 3 Reflection – Szymon Halada

# Work Log – Szymon

|  |  |  |
| --- | --- | --- |
| Date | Work Completed | Work to be completed |
| Oct - 21 | Player and enemy sprites | Sprites and art Created |
| Oct - 21 | Ability Files Added | Ability Files Added |
| Oct - 31 | Fixing controls and pacman test sprite | Setting sprites to work in game |
| Oct - 31 | Tiles and new spritesheet | Enemy Sprites added |
| Nov - 12 | Enemy animation fix | New tiles added to game |
| Nov - 14 | Arm code attempt | Arm code attempt |
| Nov - 18 | Shuttle.cpp | New levels |
| Nov - 18 | Levels added with new tiles | Levels added |

# Group Roll

According to my Belbin roll assessment, my three main rolls are Team worker(14), Plant(12), and Resource Investigator(12). With the work I have done so far for the project, I would personally think that the assessment is rather accurate as I came up with the idea and theme for the game using my creative side which ties into the Plant personality. I also found ways to make my job making sprites and sprite sheets easier by exploring the web for online editors to ease that aspect of the project which ties into the Resource investigator characteristics. I also try to suggest innovative ideas to solve issues that arise such as for example the idea of the character sprite to consist of two individual parts; the body, and a separate arm which rotates in the direction of the mouse to make it look like the player is shooting in that direction. I suggested that we learn from the zombie code from second year as we used similar code for a character in that game. In terms of weaknesses I definitely struggle with communicating my points with others and I tend to come up with a general idea but struggle to provide practical details. As far as my strengths go, I can definitely find the resources I need to complete my task and use those opportunities to aid me and also, I have a good imagination for ideas related to the game.

# Team Potential

On the team Performance Curve, I would like to say we ended up somewhere between a potential team and a real team. With more time and dedication, I would say we could become a real team. We definitely increased in productivity by the last two weeks of this project with clear goals such as getting the levels to work correctly or character picking to be implemented. Each of us on the team definitely have different strengths which we used to complete the project accordingly.

# My Additions

## Game Idea

I came up with the initial idea the game including a lot of the features (even though some ended up scrapped) and the setting which allowed me to also create all of the art for the game since I made sprites before and I already had imagined what the game would look like in my mind.

## Art

To start off the project, I created art for each of the characters and then I create a prototype of what the game will look like. Followed this by creating a poster for the game, Sprite sheets for player and enemy, and all the pickups. I also made new tiles and objects but we never ended up using the objects as we placed tiles and objects in the level using numbers from 0 to 9 on a text file and we did not have enough numbers to add all the objects.

*Figure 1 Poster* *Figure 2 Player Spritesheet*

*Figure 3 Enemy Spritesheet Figure 4 Objects Figure 5 Fuse Pickup Figure 6 Health Pickup*

*Figure 7 Ammo Pickup*

## Abilities

One of the ideas for the game was for each of our four characters to have some sort of special ability depending on who you play as. I created and coded each of the ability classes for each character. Coreys character was supposed to become invulnerable, Matthew invisible, Szymon shoot faster and have infinite ammo, and Zach was supposed to have a melee weapon and more Health. I created a cpp and h class for each of us and an abilities.h class. We did not end up implementing the abilities to the game as we took too long to get the basic characters to work correctly and select between characters.

*Figure 8 Zach Ability cpp*

## Fixing errors

I spent a lot of time fixing errors (mostly caused by myself) such as the errors caused by the addition of the abilities classes which messed up a lot of other code because I was trying to brute force it to work or even compile but I eventually fixed those issues caused by the classes. Another issue occurred when we tried to add new levels but they kept crashing the game. Initially Zach worked on adding the new level files but once the crash occurred, I came around to fix it by editing the level array code to include the new tiles for the player and enemy to walk and path find on in the playable character class.

## Arms

Initially, one of the ideas I came up with for the game involved the players to have a secondary sprite attached to them to allow for the arm to rotate. With this, the players arm would rotation with the gun would follow the mouse. Multiple of us in the group tried to get the arm to work but it never did. We ended up scrapping the idea at the end.

*Figure 9 player Arm example*

## Levels

For the levels, I created new tiles for the floor to represent a different environment on a different planet. I also created a new file called shuttle.cpp which would serve to hold the sprite for the “Goal” that takes you to the next level. This shuttle would be on each level and once enough fuses are collected per level, the player can get on the shuttle to move on to the next level.

*Figure 10 Shuttle*

*Figure 11 goal code*

# Conclusion

In this project, I contributed significantly by leveraging my creativity, resourcefulness, and technical skills. From generating the initial concept of the game to crafting its visual and mechanical aspects, my work helped shape the project's direction and its key features. I created all the art, including characters, sprites, and environmental elements, and even explored innovative mechanics like a rotating arm for the player, though not all ideas came to fruition.

I also played a vital role in debugging and ensuring the game's functionality, particularly when adding new levels or addressing code issues stemming from the abilities classes. While we could not implement all the intended features, such as the abilities system or additional objects due to time constraints, my efforts provided a solid foundation and vision for the game.

Through this process, I embraced the strengths identified in my Belbin assessment showcasing my ability to generate creative solutions as a "Plant" and resourcefulness as a "Resource Investigator." However, I also recognized areas for improvement, such as better communication and providing more detailed practical plans. Overall, I am proud of my contributions and the collaborative effort that went into bringing this project to life.

## Student 4 Reflection – Corey O’Brien

## Introduction

For our group project for our 3rd year, we were tasked to create a C++, SFML game using the theme for Games Fleadh 2025; “Small World”. I was put in a group with three of my classmates and we were tasked to complete this assignment together as a team. This reflection will go over how I found working in a group, what I learned about myself, my coding skills, my communication skills, and how I found the assignment overall.

## Planning and Methodology

Our first task as a group was to plan out what game we would be making. As an individual, the hardest part of my projects is producing an idea I am happy with. This process is a lot easier and more efficient in a group as you can bounce ideas off each other until they meld together to create an idea everyone is happy and excited about. This is exactly how we produced the concept for Cosmic Decay. Everyone suggesting ideas, some serious and some not, and mixing them together to create this group concept.

After creating the concept, we then assigned roles for each other in the team. My roles for this project were Developer and Technical Infrastructural Engineer (TIE). This meant not only was I programming on the project but as the TIE I had to understand a lot about the systems and applications we would be using as well as the responsibility to create the different diagrams such as the User stories which you have seen previously in this document. I also in collaboration with the rest of the group helped to understand the use of GitHub for the project.

## Belbin Team Roles Questionnaire

According to the Belbin Assessment, I am an Implementer. An Implementer is “well organised, enjoy routine and have a practical common-sense and self-discipline.” Although I believe my self-discipline and my organisation could be better, this is an accurate description of my role in my team. I work best when I know what I must do and when I need it done by. The more specific I can get my instructions the better my work will end up being. I also work best in a routine as without it I would have the tendency to leave work until closer to the deadline.

Belbin also says that “Implementers are useful because of their reliability and capacity for application.” I do believe that I am reliable to get my work done for the team on time and to a good standard. I push myself to try find a solution to any problems that may occur and believe that I put a good effort into the project.

In terms of weaknesses, Belbin defines the Implementer weaknesses as, “Lack of flexibility, resistance to unproven ideas.” I believe I do fall under these weaknesses; I do get a little frustrated when plans get changed especially after I have a full picture of the project in my head. I am reluctant to change plans on the fly when I have begun working on an aspect of the project.

## Work Log

|  |  |  |
| --- | --- | --- |
| Date | Work Completed | Work to be completed |
| 09/09/2024 | Added to Initial Proposal |  |
| 16/09/20204 | Set up Git Repository  Created Initial project and created CosmicDecay.cpp | Create Player header file and Player.cpp |
| 17/09/2024 | Created empty Player header and Player.cpp files, with basic functions and variables. (to be developed on). | Develop on these files with teammates. |
| 18/09/2024 | Helped finish the Initial Proposal with Matthew. |  |
| 23/09/2024 | Started writing Section 3 for the Technical Proposal |  |
| 24/09/2024 | Created User story diagrams for Section 2.1 in the Technical Proposal. |  |
| 25/09/2024 | Made final changes to Technical Proposal. |  |
| 04/10/2024 | Co-Created Project Presentation |  |
| 07/10/2024 | Started coding for Bullet and Engine. |  |
| 08/10/2024 | Finished code for Bullet and more development of the Engine class. | Have the player shoot a bullet and have the bullet collide with enemies. |
| 14/10/2024 | Input class and a temporary background created | Develop the input handler more. |
| 29/10/2024 | Finished code for Bullet and smoothed out the player and enemy animations |  |
| 30/10/2024 | The player shoots a bullet and displays a crosshair for aiming |  |
| 31/10/2024 | Created cooldown for pickup collisions |  |
| 12/11/2024 | Enemies spawn with different health values and sprites |  |
| 17/11/2024 | Updated the menu |  |
| 18/11/2024 | Character selector working and showing different sprites |  |
| 22/11/2024 | CPP files commented correctly and fixed some minor bugs |  |

## Technical Contribution

As you can see from my work log above, I have contributed my fair share of work towards this project. Although I believe I could have done more coding than I did at the start of the project, I put a good amount of effort into creating this project. I always made myself available to help anyone who was having issues and helped resolve any errors they may have been getting and the same vice versa. One error I helped a lot for was the pickup collisions. We got the pickups spawning and the player could collect it, but it was counting as two collisions instead of one. I had a similar problem in a previous project and was able to resolve the issue by making a collision cooldown for the pickups.

Another one of my big contributions was the character selection screen and the menus. I made sure that the menu displayed and handled input correctly. Same applied for the character selection which I made sure that the correct sprite would display and animate depending on what character you chose in the selector. This was one of our key unique features to implement and I was incredibly pleased with how it ended up.

## Team Performance

As a team we performed good together on this project. As we were already close friends before this project, we could skip over the introduction stage on teamwork, and we could go straight into planning and programming. Being close friends also had its downsides as nobody wanted to initially take charge or step on each other's toes. We eventually came to agreement on team roles and began our programming.

Like most groups, we were not without conflict. Although not much, there was still small conflict which arose from frustration with ideas and code not working correctly. But we made sure to keep effective communication and resolve those conflicts so that we could continue not only to project but also keeping the friendships intact.

In our lectures we were show the team performance curve which is used to measure how effective and good a team works together. We went through this curve together and ended up around the Real Team end of the curve. I believe we could easily reach the High-Performance Team if we work in a group again and take what we learned from this project into the next one.

## What would I do differently?

One key aspect I would improve on is time management. I would start programming a lot earlier than I did for this project and I would devote more of my time earlier than what I did. I feel not only would I benefit in terms of handling stress and other college work but the whole group could have benefitted which could have helped avoid any frustrations and stress towards time constraints.

I would also have ensured that we all knew the plan for the project as there were a few confusions early on about certain aspects of the game. Although these confusions did not create big issues, we could have benefitted from all being on the same page from the get-go. I would create a text file of the ideas so we can all refer to it when programming and not have to remember every aspect of the project.

## Conclusion

From this group project, I have gained a lot of knowledge that not only helped to better my programming skills, but also skills that will help me in terms of future employment. Although I have experience in working in teams from previous jobs I have had outside of college, this project gave me the experience of specifically programming in a team which will benefit me if I get a position as a programmer in the future. I also gained an insight into how a group project for a professional position may be like.

In terms of my programming ability, this project pushed me to research and put a lot of my time and effort into doing my absolute best for this project and I feel I have gained a lot of dedication from this. I also feel that I have bettered my knowledge on different programming methods and practice thanks to the research and practical work I have put into this project.