University of Plymouth

School of Engineering,

Computing, and Mathematics

COMP3000

Computing Project

2024/2025

Timeline Takedown

Morgan Tomos Hodge

Insert your Registration Number

BSc (Hons) Computing and Games Development

# Acknowledgements

I would like to thank me for working relentlessly this year

I would like to thank Anthony Edwards, he has been incredibly helpful this year going out of his way to help out and genuinely caring about issues I encountered or questions I have.

And lastly thank you to me again

Abstract DONE (314 words)

This report outlines the development of Timeline Takedown, a first-person, wave-based survival shooter developed using the Unity game engine. The project draws inspiration from classic round-based survival modes, such as those found in Call of Duty Zombies. The motivation behind this project stems from my personal experience growing up with games in this genre. In recent years, I have observed a noticeable decline in the quality and innovation of similar games, which led me to explore creating my own version - tailored to deliver a fun, replayable experience for my friends, family, and broader audiences who enjoy this style of gameplay.

This report begins with an exploration of existing games within the round-based survival shooter genre, examining what made them successful and identifying the key developers and titles that influenced the field. Following this, I will outline the objectives and deliverables of the project I created, along with the method of approach taken to meet these goals. In this section of the report will also be a lit review, where I discuss blank.

This main body of this report focuses on the development and implementation of the game. I will be discussing the project management techniques I used to maintain a steady and efficient workflow throughout the project. I will also be breaking down the development sprints and discussing what was achieved at each phase, and when they took place. Additionally, user testing will be covered detailing the tools and methods used to gather feedback and the resulting changes made to the game.

The final section of the report will feature an end of project review, where I will reflect on what was achieved throughout this project, and the quality of the work achieved. This will be followed by a reflection, evaluating what went well, what could have been improved, and how this experience will affect how I work on projects in the future.

# Contents

Contents

[Acknowledgements 2](#_Toc196244248)

[Abstract 3](#_Toc196244249)

[Contents 4](#_Toc196244250)

[Word Count - 5](#_Toc196244251)

[Code Links: 5](#_Toc196244252)

[e1 - Introduction (400 words) 5](#_Toc196244253)

[1.1 Overview 6](#_Toc196244254)

[1.2 Audience 6](#_Toc196244255)

[1.3 Purpose 6](#_Toc196244256)

[2 - Background, objectives & deliverables 7](#_Toc196244257)

[2.1 Project Background - 347 7](#_Toc196244258)

[2.2 Market / Competitors 7](#_Toc196244259)

[2.3 Objectives & Deliverables - 235 9](#_Toc196244260)

[3 - Ludology 10](#_Toc196244261)

[3.1 Introduction to Ludology / Game Studies 10](#_Toc196244262)

[3.2 Genre & Gameplay Analysis – Round-Based Survival Games 10](#_Toc196244263)

[3.3 Case Study – Call of Duty Zombies 10](#_Toc196244264)

[3.4 Map Design and Environmental Ludology 10](#_Toc196244265)

[3.5 Mechanics & Player Feedback 10](#_Toc196244266)

[3.6 Ludology Reflections on Your Development 10](#_Toc196244267)

[3.7 Conclusion 10](#_Toc196244268)

[4 - Method of Approach 11](#_Toc196244269)

[4.1 Methodologies - 195 (I think is done) 11](#_Toc196244270)

[Limitations 11](#_Toc196244271)

[Alternatives 12](#_Toc196244272)

[4.2 Technologies 13](#_Toc196244273)

[4.3 Project Management Approach 14](#_Toc196244274)

[Planning and Structuring the Project 14](#_Toc196244275)

[Visal Planning Tools 15](#_Toc196244276)

[Tracking Progress & Reflecting 15](#_Toc196244277)

[5 – Legal, Social, Ethical and Professional issues 16](#_Toc196244278)

[5.1 Legal 16](#_Toc196244279)

[5.2 Social 16](#_Toc196244280)

[5.3 Ethical 16](#_Toc196244281)

[6 – Project Management 17](#_Toc196244282)

[6.1 Project Management 17](#_Toc196244283)

[6.2 Version Control 17](#_Toc196244284)

[6.3 Meetings 17](#_Toc196244285)

[7 - Implementation 18](#_Toc196244286)

[7.1 Planning & Research 18](#_Toc196244287)

[7.2 Sprint 1 (8.11.2024 – 22.11.2024) 18](#_Toc196244288)

[7.3 Sprint 2 (22.11.2024 – 6.12.2024) 19](#_Toc196244289)

[7.4 MVP 19](#_Toc196244290)

[7.5 MAP 19](#_Toc196244291)

[7.4 Uploading The Game 19](#_Toc196244292)

[8 - User Testing 20](#_Toc196244293)

[8.1 First Play test 20](#_Toc196244294)

[8.2 Changing the Testing Approach / Method 21](#_Toc196244295)

[8.3 Continuing testing 21](#_Toc196244296)

[8.4 Testing Results 21](#_Toc196244297)

# Word Count -

# Code Links:

GitHub Repository:

Itchio Page where the game is hosted:

Insert link to your code submission directory

# e1 - Introduction (400 words)

## 1.1 Overview

Timeline Takedown is a video game developed for PC, with an initial release planned for Itch.io, and potential future release on Steam. The game was created using Unity Editor version 2022.3.18f1, with all scripting written in C#. Inkscape was used to design the game’s user interface elements.

The gameplay takes place across two distinct levels, each featuring a unique theme to provide variety and maintain player engagement. In each level, the player must survive five escalating waves of enemies before facing a final boss encounter in round five. Defeating the boss rewards the player with a timepiece part, which is required to progress to the next level. This core gameplay loop is designed to offer a structured yet intense survival experience that encourages strategic play and replayability.

## 1.2 Audience

**Timeline Takedown** is designed to fall under the **PEGI 12** age rating, as it features “violence in a fantasy setting or non-realistic violence” (1). The visual style and gameplay avoid graphic content, making it suitable for younger audiences compared to more intense survival shooters. Despite its simplified tone, the game retains the core appeal of the round-based survival genre, offering an engaging and fast-paced experience that is accessible to a wide range of players.

The game fits within the round-based survival shooter market, alongside titles such as *Call of Duty Zombies*, *Killing Floor*, and similar wave-based experiences. However, unlike these examples which typically target an audience of males aged 18 to 34 (2), often due to more mature themes and realistic violence, Timeline Takedown is intended to reach a slightly broader and younger demographic. Its simpler mechanics and less intense tone are designed to be more approachable, reducing the learning curve while maintaining the genre’s core excitement.

## 1.3 Purpose what is the problem this is solving

The purpose of Timeline Takedown was to create a game that would be enjoyed by my friends , family , and players all over the world who enjoy playing games in this category. I wanted to create a game in the Round Based Survival category that hadn’t quite been done before, I wanted to make something unique that I would enjoy playing, I wasn’t bothered about if the game would be successful or make any money.

The aim of this game is to survive, explore the map design and have fun. The game includes story telling at parts such as the introduction to the game and the end game.

# 2 - Background, objectives & deliverables

## 2.1 Project Background - 347

Timeline Takedown is a round-based survival shooter, these types of games have become a widely recognised and engaging subgenre within the wider category of first-person shooters. These games typically challenge the player to survive increasingly difficult waves of enemies, the player tends to play strategically, use map knowledge to their advantage, and ration their resources in order to survive. One of the most iconic examples in this genre is the Call of Duty Zombies mode that was originally introduced as a bonus reward for completing the main games campaign in Call of Duty: World at War. This example was one of the first instances that popularised the round-based survival genre that included fast paced combat, puzzle like map design, hidden easter eggs, and wave-based progression.

This genre is known for its high replayability value, and its ability to create intense moments of gameplay. Players are placed in intricately crafted level maps, facing off waves of AI enemies that increase in both number and difficulty with each round.

This project explores the design and development of a custom round – based survival game that is created in Unity. Timeline Takedown and my thought process behind certain mechanics in this game take direct inspiration from the mechanics and pacing of Call of Duty Zombies. The goal is to develop a game that captures the intensity and satisfaction of surviving waves of enemies, while also exploring exciting locations through custom made levels and enemies. (Maybe more like not custom levels but relevant cux it has same weapon theme and enemy and map)

As part of this project, two (possibly three) fully playable maps have been created, each offering unique layout, theme, weapons, and enemies. These maps aim to test different survival strategies and enhance repalyabilty as they all have completely different layouts, For example one is inside a tight cornered hospital and another is outside in the sandy climate of Egypt. This project also investigates how wave escalation, player agency, and environmental design contribute to overall player engagement in round based survival games.

Could maybe talk about the books I read

## 2.2 Market / Competitors

Waved-based gameplay is a core mechanic used in many successful titles. The following table compares several games that utilise this system, highlighting their unique features, supported platforms, and player feedback to help inform future design considerations.

|  |  |  |  |
| --- | --- | --- | --- |
| Game Title | Features | Platform | Reviews (Customer Feedback) |
| Call Of Duty Zombies (1) | * “Easter Eggs” – Objectives hidden in maps requiring puzzle solving * Custom maps * Perk system * Progressive weapon upgrades * Large AAA team behind development | PC, Xbox, PlayStation, limited Nintendo products | Generally positive – praised for addictive gameplay, creativity, and replay value. However, more recent titles have shown a lack of player engagement due to poor choices by the developers such as replacing beloved voice actors with AI(3). |
| Killing Floor | * Co – op survival horror * Unique perk system * Boss fights * Variety of enemies and weapons | PC, Xbox, PlayStation | Positive – loved for teamwork focus, fun to play with friends. But some say it can feel repetitive. |
| Left 4 dead | * AI Director (4) dynamically changes the gameplay * Co- op story mode * Different infected types * Fast-paced wave combat | PC, Xbox | Very Positive – acclaimed for co op fun, replayability, and modding support |
| Sker Ritual | * Round based survival horror * Co-op (1-4 Players) * Supernatural enemies * Weapon upgrades and unique abilities * “Easter Eggs” – Objectives hidden in maps requiring | PC, PlayStation, Xbox | Mostly Positive – praised for atmosphere and gameplay loop, though some desire more content |
| Deep Rock Galactic | * Co-op wave survival * Objective based missions * Class system * Procedurally generated caves | PC, Xbox, PlayStation | Very Positive – praised for teamwork mechanics, humour, and replayability |

*Figure 1: Competitor Analysis*

## 2.3 Objectives & Deliverables - 235

A Game Design Document (GDD) was created for Timeline Takedown and can be found in the appendix of this report. During the development of this document, a list of key deliverables was outlined and organised using a priority-based system. This helped structure the development process by identifying which features and assets were essential for the core gameplay experience, and which could be considered lower priority or stretch goals, depending on the time available.

Must Have :

o Functional wave-based combat system with enemies and bosses

o Historical themed levels with unique enemies and environments

o Basic UI – Health display, ammo, wave counter, etc)

o Player movement and combat mechanics( movement, shooting, enemy targeting)

o Basic level progression with the time machine parts and boss fights

Should Have :

o Multiple difficulty settings( easy/normal/hard)

o Weapon variety (different weapons)

o Power Ups

o Sound Design that matches the map theme

Could Have:

o 2 player CO-OP

o Boss-Specific mechanics that vary depending on location (special attacks)

o Cosmetic customisation for player model and weapons

o Player level progression

Want to Have:

o More than 3 or levels, possibly up to 10

o Online support for friends to play over the internet

o Hidden challenges/missions (easter-eggs)

Although not being stated in the GDD directly, having an itcho page built for the game and uploading a working .exe on the itch page was also a “must have” requirement.

# 3 - Ludology

 is the study of games, the act of playing them, and the players and cultures surrounding them.

## 3.1 Introduction to Ludology / Game Studies

## 3.2 Genre & Gameplay Analysis – Round-Based Survival Games

## 3.3 Case Study – Call of Duty Zombies

## 3.4 Map Design and Environmental Ludology

## 3.5 Mechanics & Player Feedback

## 3.6 Ludology Reflections on Your Development

## 3.7 Conclusion

# 4 - Method of Approach – 2090 words

## 4.1 Methodologies

Throughout the development of Timeline Takedown, I followed a flexible and iterative approach that allowed me to respond to challenges and make improvements as the project evolved. While not following a strict software development methodology, my workflow was heavily inspired by Agile principles, particularly the focus on breaking the project down into smaller tasks and continuously building on each version of the game.

I chose Agile because it was the best fit for the nature of game development, where creativity and adaptability contribute significant importance to the development of the game. This methodology allowed for significant creative freedom, enabling me to implement changes as I worked, and structure my workflow into manageable chunks. To support this approach, I used Trello as my task management tool. Trello enabled me to visually organise the different aspects of the project, such as sprint planning, deadlines, feature tracking, and bug lists.

Maybe a bit more about why I chose it and why it was good

I used a **Kanban-style task management system(isthisarefrence)** through Trello to keep track of my backlog, active tasks, and completed work. This helped me stay organised and prioritise tasks based on what was most important or time-sensitive at each stage.

I didn’t follow traditional test-driven development at the start of the project, but I regularly played through the game during development to identify bugs, balancing issues, and areas for improvement. I also made use of **runtime parameter adjustment** **(isthisarefrence)** in Unity (e.g., for enemy speed, wave timing, etc.), which helped me iterate quickly without rebuilding the game constantly.

Alongside this, I recorded devlogs and kept notes throughout the project to reflect on progress and decisions. These proved useful for tracking what had been achieved, what needed reworking, and where time was being spent.

### Limitations

While Agile proved to be a highly effective approach for my project, it was not without its limitations. One challenge I encountered was related to long-term planning and how quickly priorities could shift mid development. For example, Level 2 was originally intended to be an underwater stage, but after struggling with the complexity of implementing swimming mechanics, which would have required a significant amount of time and technical problem-solving, I made the decision to pivot. Instead, I designed an abandoned hospital level, which allowed me to focus on gameplay rather than mechanics that were beyond the scope of a solo project.

This shift is a good example of Agile’s flexibility, but also highlights a downside: some ideas had to be cut or changed quickly, and not all time spent on the original concept translated into the final product. In terms of task prioritisation, I did well in focusing on core mechanics first before diving into level design. However, one thing I would do differently is prioritising animations earlier in development. This aspect of the game took longer than expected, and I continued to run into issues with animation syncing and quality even late into the project timeline. If I had addressed those earlier, it could have smoothed out a lot of later stages and testing.

Despite these limitations, the benefits of using an Agile-inspired workflow - particularly the ability to pivot quickly and iterate based on feasibility were crucial in managing the unpredictable nature of game development.

### Alternatives

When researching different development methodologies ,I found several alternatives that had potential to be the right fit such as Waterfall, Scrum, and Lean Development. Each approach had its own strengths that could have aligned with certain aspects of Timeline Takedown.

Waterfall was considered due to its structured, sequential nature with clearly defined stages such as planning, design, implementation, testing and deployment. This method can be beneficial for projects with fixed requirements and a predictable outcome. However, game development is rarely that linear. Creative features, gameplay mechanics, and balancing often need ongoing adjustment, and Waterfalls rigidity could have slowed down the process or required major overhauls after key stages were completed.

Scrum, a more structured subset of Agile, was also a strong contender. Its emphasis on regular sprints, stand-ups , and reviews helps push for consistent progress and team accountability. If this was a team project than Scrum might have been ideal as it provides a clear communication framework and keeps all members busy through frequent check ins. However, as a solo developer, maintaining all the formal Scrum ceremonies would have added unnecessary overhead without much added value.

While each one had their own potentials I ultimately chose to stick with a flexible Agile-inspired approach. It gave me the best of all worlds: structure without rigidity, room for creativity, and the ability to adapt quickly to changes or new ideas. My familiarity with Agile also meant I could dive into development without a steep learning curve.

On top of that, industry professionals frequently recommend Agile for game development due to its iterative structure, feedback-driven loops, and suitability for handling the unpredictable nature of game design and mechanics. For instance, Clinton Keith, a veteran game developer and author of *Agile Game Development with Scrum*, advocates for Agile methodologies, stating that they can help teams deliver games more efficiently, rapidly, and cost-effectively, while also enhancing the development experience for team members . This made Agile the most natural and effective choice for me. <https://www.oreilly.com/library/view/agile-game-development/9780136204831/>

## 4.2 Technologies

In order to bring Timeline Takedown to life, I relied on a range of technologies that supported different aspects of the games development process, from planning and design to version control. Below is a breakdown of the key software and tools used throughout the project, along with the reasoning behind each choice. These technologies were selected based on their compatibility with my workflow, accessibility, and suitability for solo development.

* **Unity Engine**

The game was developed in Unity Engine (version 2022.3.18f1), which was chosen due to its flexibility, familiarity, and strong support for 3D games development. Unity also offers a large range of built in systems to make development slightly easier, such as physics, animations, UI.

* **Programming Language: C#**

All gameplay scripting was done in C# as its Unity’s primary supported language, and the most commonly used within Unity. It was used to implement core systems such as enemy AI, wave spawner mechanics, player interactions, and UI functionality.

* **UI Design: Inkscape**

Inkscape was used to create the majority of the UI seen in this game, drafts were created inside of Inkscape and then iterated into the final version, as seen in appendix #. Inkscape was chosen as I have had a lot of prior experience with this software, and I knew what both I and the software is capable of achieving. It is also free and offers a range of useful features such as image size scaling, easy exporting to any file type, and a lot more.

* **Development Logs (Devlogs)**

Development logs – or commonly known as Devlogs, were recorded at the end of every 2 sprints. They were used as a method to track progress, record key milestones, and upload onto YouTube for viewers to watch.

* **Version Control**

Version Control was managed by using GitHub and GitHub Desktop, allowing for daily commits and backup, and allowing for easy tracking during development as I also added a description to most commits. GitHub desktop was mostly used due to its simplicity and user friendly interface.

* **Map Designs**

Initial map designs were sketched on paper to plan out the layout, player flow, and enemy spawn points before being built in Unity. These paper designs were then refined using the online tool Dungeon Scrawl (<https://app.dungeonscrawl.com>), which allowed for a more visual and structured digital representation of each level.

Each tool and technology listed above played a vital role in shaping Timeline Takedown. Choosing tools that I was familiar with such as Inkscape and GitHub Desktop – while still pushing myself to explore new techniques such as dungeon scrawl and paper based designs – allowed me to work efficiently as a solo developer. I considered including a full breakdown of hardware and software used but I have decided to instead include that in the appendix to keep the main section focused on the tools directly involved in development, if you wish to see the full breakdown of tools please refer to APENDIX 7.2. Overall, the combination of these technologies supported a smooth workflow and allowed me to bring the vision for the game to life.

## 4.3 Project Management Approach

Throughout the development of Timeline Takedown, I used a flexible and iterative project management approach inspired by the Agile methodology. While I did not follow a strict formal Agile framework such as Scrum, I applied many of its core principles such as breaking the work into smaller manageable tasks, working in sprints, and frequently reflecting on progress throughout the use of development logs and meetings. This approach suited the nature of game development and gave me the creative freedom to adapt, iterate and evolve the game design as needed.

### Planning and Structuring the Project

The project began with the creation of a comprehensive Game Design Document (GDD), which acted as the foundation for the games vision. This document included early design ideas, gameplay mechanics, art style reference, and technical requirements. I also researched into other games that I took inspiration from and discussed what I would like to implement from these games into mine, such as round based mechanics from call of duty zombies, and inspiration on how The Simpsons game did level and character themes. The full GDD can be found at section XYZ IN THE APENDIX.

Development was divided into bi-weekly sprints, each with focused goals such as implementing a specific feature, completing level layout, and fixing bugs. I used Trello to organise and manage these sprints through a Kanban style board, with columns for “To Do”, “In Progress”, “Completed”, and addiational colums for side notes such as “Useful Resources” and “Important Dates”. This helped me prioritise tasks and track development progress visually and effectively as if I was to take a different approach such as a mind map it would be bad etc.

Image of Trello

As a method of monitoring progress and staying accountable - and also a requirement of this project, I attended bi-weekly meetings with my supervisor. At these meetings I presented updates, discussed issues I was facing, and received feedback. These sessions were essential for keeping the project on track as In some situations such as when I had a issue with the animation and the models feet clipping under the ground , I was able to be given a resource that would prove to fix this problem. Without these meetings the project would have moved a lot slower as I would have got stuck on issues and the progress would have slowed down due to being this. (re read)

### Visal Planning Tools

During the early stages, I created paper based diagrams for both map layouts and UI mockups, allowing for me to experiment freely with ideas before committing to digital formats. These designs were later recreated using Dungeon Scrawl (3) – an online tool for map creation - to create clean, digital versions of the level layouts. This tool made it easier to visualise flow, structure and enemy spawn point placement before building the levels inside Unity. If you would like to view these paper based diagrams please check appendix axyz.

1 IMAGE OF MAP

### Tracking Progress & Reflecting

At the end of each sprint, I recorded and uploaded Development logs to YouTube. These served as development checkpoints and allowed me to reflect on progress, document challenges, explain key decisions and talk about what I will be working on for the upcoming sprint. Not only were they useful for personal tracking, but they also allowed for public visibility on the projects evolution, and can be referred back to through my career in games development.

A link to the Devlog playlist that includes all the videos created can be found here:

I maintained a balance between feature development, testing, and visual polish by continually adjusting the Trello board and reviewing my GDD and other design documents I created. For example when I realised the underwater level was too complex due to the technical demands of swimming mechanics and animation, I changed to a hospital themed level, which allowed me to reuse existing gameplay systems more effectively. This flexibility was one of the major strengths of my Agile workflow.

# 5 – Legal, Social, Ethical and Professional issues

## 5.1 Legal

Maybe how assets were free etc

Age rating

## 5.2 Social

## 5.3 Ethical

# 6 – Project Management

## 6.1 Project Management

## 6.2 Version Control

## 6.3 Meetings

# 7 - Implementation

## 7.1 Planning & Research

Read Books

Watcged videos

GDD

## 7.2 Sprint 1 (8.11.2024 – 22.11.2024)

For my first sprint I started by creating my GitHub Repository

Set up Trello and github

Paper Level Designs

GDD

Movement

Player

Playing around with modelling tools and finding assets

## 7.3 Sprint 2 (22.11.2024 – 6.12.2024)

Started to create level 1 Design

Started to implement mechanics ?

Create Basic UI

Create Game States (start, menu , pause)

Devlog 1?

## 7.4 MVP

## 7.5 MAP

## 7.4 Uploading The Game

8 - User Testing GO THRO AND RE RITE

User Testing was crucial to the development of this project as I needed to get an understanding of what the general public thought of the game at its current state, so I can receive feedback and apply changes that become apparent through that feedback.

When creating the user testing forms I decided to make the questinare annonamouse as I had no need to collect personal data, I just wanted to receive feedback on the game, I was asking no personal questions. With having no personal data collected , that meant I did not have to spend more time adhering to GDPR and other data regulation rules as I just cut it all out by making it annoanmous.

I wont discus every single playest as this would take up half of this report, I will cover the most impactfull ones, and leave the data below:

8.1 First Play test17/02/2025

My first Play Test was conducted on the 17/02/2025, this playtest was shortly after I had finished with my first level. I conducted this playtest in person inside of the University of Plymouths Smeaton Building. For this playtest I created a excel spreadsheet that held questions and required the testers to fill out their responses in a dedicated box next to the questions. You can find photos here link to apendenciees

Speak about what I chose to test in this ..

A screenshot of a computer screen

AI-generated content may be incorrect.

*Figure IDK, User Testing Excel sheet*

The playtest proved to be successful as I got feedback that I would proceed to use to iterate on my game such as a bug with the boss not spawning in, and players not knowing that they can pick up health packs. I took this feedback and made it so the boss spawned and so on…

## 8.2 Changing the Testing Approach / Method

However, this method of testing using an excel sheet was extremely unprofessional and so much more complex than just creating a forms doc using google forms or Microsoft forums as this method overwhelmed the tester with so much text on the screen and unclear instructions on what they need to fill out and where. Because of this I created a new user testing document using Teams Forums , that I would use in the testings in the future. I used this because …. Ez to turn into redable data etc

You can view the revised empty testing document here : link

Or View photos In the appendix

## 8.3 Continuing testing

All dates testing happened

## 8.4 Testing Results

The results were turned into data for graphs and stuff so its easy to read, this was used to improve changes to the game. You can view all the data here in apendenciees.