Designing Game Mechanics from a Given Brief

Technical Design Documentation

Designing & Implementing Game Mechanics - WRIT 1 – CIS5011



## **Met: Gone Small**

*V0.1*

*Sova Arts*

*Liam Barrett (ST20228164@outlook.cardiffmet.ac.uk)*

*15/12/23*

*Version number 1.0*

# Table of contents—

[Met: Gone Small 1](#_Toc158575799)

[Table of contents— 1](#_Toc366290743)

[Development Requirements 2](#_Toc1051851661)

[Asset Specifications 2](#_Toc162658202)

[Project Structure 2](#_Toc2017669836)

[File Naming Convention 3](#_Toc441084313)

[Level / World Details 3](#_Toc940105960)

[Development Plan 4](#_Toc1617181489)

[Playtesting 5](#_Toc1460230496)

[Appendix(ces) 6](#_Toc125494978)

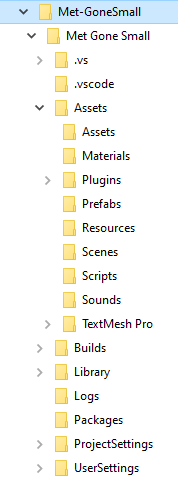
Development Requirements -

* Development – Visual Studio 2022 / Visual Studio Code 2022
* Game engine – Unity version: 2022.3.11f1 with textMeshPro extension
* 3D Software – 3Ds Max, Blender
* Project Management – GitHub
* Source control - GitHub
* Sound software – Audacity, BeepBox

Asset Specifications—

|  |  |  |  |
| --- | --- | --- | --- |
| Model | File format | Poly Count | File Size |
| BigIron  (Singleshot) | .FBX | 156 | 856kb |
| TommyGun (Autoshot) | .FBX | 330 | 892kb |
| EnemySpider | .FBX | 1041 | 1458kb |
| Monitor | .FBX | 656 | 1008KB |

Project Structure—



This is the directory structure of the Project; this shows all the asset folders as well as the Unity folders needed to develop the game and to run it.

File Naming Convention—

For our file naming convention, we will be using PascalCase. We will be using Pascal\_Snake\_Case for asset files.

Level / World Details **—** Layout and asset list including appropriate size limitations.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Function | Size Limitation | Image |
| Monitor | A surface to wall ride onto to move throughout the level | 27.7 m by 23.6 m |  |
| Enemy (Spider) | An Enemy to chase, deal damage and kill the player |  |  |
| SingleShot | A single shot gun holding 7 bullets. Used to kill enemies | N/A |  |
| AutoShot | A RapidFire gun holding 25 bullets. Used to kill enemies | N/A |  |
| Player Character | A character for the player to move, shoot and wall run through the levels |  |  |

Development Plan—

|  |  |  |  |
| --- | --- | --- | --- |
| Milestones | Date | Deliverable | Approval |
| Pre-Production End | 15/12/23 | GDD  TDD  Production Plan  Prototypes | Whole team |
| Alpha | 01/02/24 | Assets imported Game version Alpha | Team Leader |
| Beta | 16/03/24 | Game version Beta | Team Leader |
| Final | 28/5/24 | Final game version | Project lead |
| Pitch and Play | 2/7/24 | Full game build  All documentation  Game pitch | Project lead |

Playtesting— Overview of the kinds of tests, e.g., player control, combat, puzzle etc. and individuals responsible.

1. The minimum specification for the target hardware your game will run on will be used to justify optimisations in your code.

The Hardware Specifications for the game to run minimally and for recommended:

Minimum:

Graphics card: ASUS NVIDIA GeForce GT 730

CPU: Intel Core i3 10110U

Ram: 4GB

Recommended:

Graphics card: GTX 560

CPU: Intel Core i3 10100F

Ram: 4GB

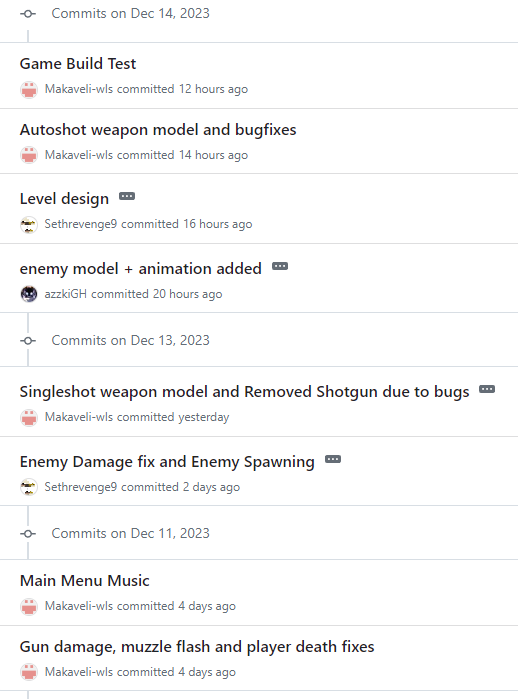
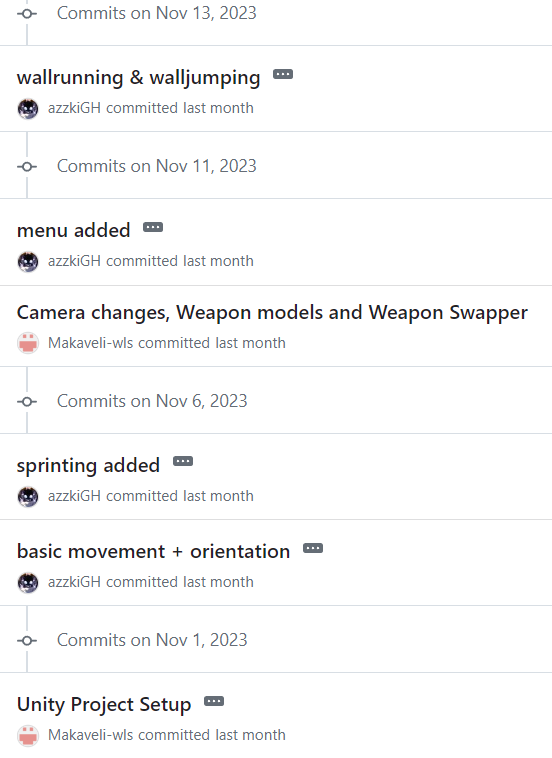
2. Discuss the mechanics of your game and distinguish non-core and core.

**Core** – We added running, jumping, double jumping, sliding, wall running & jumping, and shooting. These are integral to the game’s design and form of gameplay and so are classed as core mechanics. The basic movement mechanics (running, jumping, sliding) are for maneuvering across the level and to escape from enemies. Shooting is added as a way for the player to defend themselves against the enemies.

**Non-Core** – We added gun tilting and camera tilting for when the player is wall-running. These are non-essential to the game and are in place for aesthetic purposes, classing them as non-core mechanics. Reactive tilting is to create a sense of immersion for the player and give them an enjoyable way to move and shoot.

# Appendix(ces)

**Version control logs-**





*(Makaveli-wls – Liam, Sethrevenge9 – Rhys, azzkiGH – Carl, Cers04 – Cerys)*

Outcomes of team meetings

|  |  |  |
| --- | --- | --- |
| Team Meeting no. | How Long | Outcome |
| 1 | 2 hours | Decided structure, theme concepts. |
| 2 | 1 hour | Setting up Repository and ensuring everyone has access. |
| 3 | 30 min | Discussing who does what in the code. |
| 4 | 30 min – 45 min | Updating each other on our sides of the split code, readjust code discussion. |
| 5 | 25 min | Play Test. |
| 6 | 30 min | Check in on how the code is coming along. |
| 7 | 45 min | Discussing how each of our code parts will combine and communicate together (enemies and ammo for example). |
| 8 | 25-35 min | Play Test. |
| 9 | 30-45 min | Discuss TDD and Closing Kit and their relevant information. |
| 10 | 30 min | Ensure the project in its entirety is ready to go. |
| 11 | 20 min | Final check that everything is completed. |

Reflection & Contributions

Liam - I had a great experience working with this group as we all worked well and supported each other when we were struggling or when we had issues with any code. I chose to work on the weapon mechanics as well as the health and death for the player. I chose to work on these because I've had the most experience with working on these aspects of the game and could put the best work in on that area of the game. I think we worked well as a team overall and got almost everything working how we wanted it to.

Carl – My teammates were incredibly supportive when I needed help. I decided to start the project off by working on the player’s movement mechanics. Since I had more time, I decided to include wall-running, wall-jumping as well as include reactive camera tilting dependent on your position on the walls. In the later stages of the project, I decided to make the game look more interesting since I already completed my parts and created an enemy model and an animation for it. I am very familiar with movement code so this was a good suit for me. I am glad to have been part of a productive and strong-armed group.

Rhys – My experience with the group was nothing but good, we seemed to get along very well and complemented each other with our ideas and practical skill. I did all of the enemy's functionality, the pause menu and the level design. The task I chose was to my strength and I thank the group that I was able to choose them, the support with code and documentation is second to none. I couldn't have asked for a better group

Cerys – My experience with the group was good. They supported me and helped me when I was stuck. I worked on Sliding mechanics and Double Jump mechanics. I didn’t have a strong ability in any particular area, so I did the smaller code pieces. Our discussions about the game were always constructive and we really built on each other's ideas.

**Carl** – walking, running, jumping, wall-running, reactive camera tilting, wall jumping, enemy model, enemy walking animation, main menu screen + buttons.

**Liam –** Weapons, models for weapons, reloading, player health and death, death screen and restarting, music and sound effects and health and ammo UI display

**Rhys –** Enemies, Enemies Chasing the player, Enemies dealing damage, Enemies dying, Enemies spawning randomly, Pause menu and level design

**Cerys –** Double jumping, Sliding, Health and Ammo Pick Ups