**<TEAM NAME>**

*Assignment One:*

*3D Game (Alpha)*

**Team Members:**

11688025 – Deinyon Davies

11685740 – Daniel Nelson

11697822 – Matthew Wale

11729631 – Jesse Walker

**How To Play.**

Welcome to <game>! <game> is a fantasy-themed tower-defence styled game ***(not sure if this should be re-worded considering the FPS approach we are taking for the alpha)*** in which the player must stop enemies from reaching their home base. This is achieved by setting up a path of obstacles and defences for the enemies to route through. Players will be able to place a wide variety of walls and weapons across the map, and these will aid in stopping the waves of enemies attacking their home base. The defences will be automated, attacking any enemies in range. Players win by stopping all of the enemy waves, and maximising their score in the process.

**Technical Features.**

• ***evaluation of features based on criteria***

Our game includes a basic 3D map, with a camera and multiple 3D objects placed within. At present these are tanks on a flat plane. The player is represented by the camera, which can be moved with the keyboard, and aimed with the mouse. ***(Is mouse picking used in the alpha?)***

Objects can detect collisions with basic ***(not sure which collision detection method was used)*** , and objects will respond by ***(once again I don’t know enough)***.

Our game has simple audio, including a backing track that plays to set the tone. There are also sound effects for ***(Tanks? Player? Towers?)***.

The code is very modular, as we have added plenty of classes to use and adapt. The inclusion of parent classes such as Entity (for any in-game objects) allow for extensive reusability and flexibility to easily and quickly make changes to child classes. ***(I’m not sure this is the right thing to say)***

The game has primitive kinematics, and we have tried to make the movements of the camera and other objects as natural as possible. Basic gravity for jumping and other forces and accelerations have been implemented to achieve this. ***(Feel free to add more I am just winging this)***

Two steering behaviours have been included: pursue and arrive. The tanks will pursue the player within a certain range, actively chasing the player based on their current position and where they are predicted to be slightly ahead of time. When the tank is close enough to the player, it will begin to slow itself until it collides with the player, at which point it stops. ***(Again, I feel like this is slightly wrong)***

Filler text to avoid colour reformatting! ***(Objectives, conflict, and scoring… do we have much to say?)*** Filler text again.

Filler? I hardly know her! ***(Not sure if we have anything to say for bonus technical stuff)*** Filling the void once more.

• ***justify desired grade***

Our desired grade for this assignment is the maximum possible marks. We believe that we have adequately filled the assignment criteria, having added all of the necessary technical requirements. ***I’m not sure if spatial partitioning or what have you has been attempted but if it has hot damn could we claim some of those sweet, sweet bonus marks.***

***Also if there is anything I have missed…*** Filler.

• ***describe extra features worth mentioning***

A filling prospect. ***These guys have got me here, I don’t really know enough about the code.*** …and yet another.

**Peer Evaluation.**

*Deinyon*: <role>

**Score:** 0.0-5.0

<Justification spiel>

*Daniel*: Designer/AI programmer

**Score:** 5.0

Researched A\* path-finding and implemented a demonstration for it. Also led design talks and contributed lab code to the project.

*Matthew* : Documentation/Administration

**Score:** 3.0

Organised and compiled elements for the read-me document.

*Jesse* <role>

**Score:** 0.0-5.0

<Justification spiel>