**Team**Cool Bananas

*Assignment One:*

***3D Game (Alpha)***

**Team Members:**

**11688025** – Deinyon Davies

**11685740** – Daniel Nelson

**11697822** – Matthew Wale

**11729631** – Jesse Walker

**Introduction**

Welcome to **Cylinder Pathfinding Simulator 2014 (CPS2014)**! CPS2014 shall become a fantasy-themed tower-defence styled game in which the player must stop enemies from destroying their home base. As a prototype, the game works more as a technical demonstration, though it exhibits many of the fundamental algorithms required to continue development.

**Controls**

|  |  |
| --- | --- |
| **Operation** | **Action** |
| Mouse Movement | *In FPS and Flying modes:* Rotate the camera |
| Left Mouse | Fire a projectile **or** place a Wall Tower in the Pathfinding Sandbox |
| Right Mouse | Pathfind to the selected block in the Pathfinding Sandbox |
| Middle Mouse | *In Top-Down mode*: Press and drag to rotate the camera |
|  |  |
| W,A,S,D Keys | Translate the camera about the world |
| Spacebar | *Only in FPS Camera mode:* Have the player jump |
| Shift | Increate the translation speed of the camera or player |
|  |  |
| 1 Key | Use an FPS Camera |
| 2 Key | Use a Top-Down Camera |
| 3 Key | Use a Flying Camera |

**How to Play**

Evade the enemy tanks and their projectiles. Hide behind models in the world to distract the enemies. Use the various camera modes to place obstructive towers (left mouse button) or pathfinding waypoints (right mouse button) in the Pathfinding Sandbox. Fire projectiles at objects with the left mouse button.

**Technical Features.**

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

The game exhibits a three-dimensional world composed of multiple static models, non-static entities, NPC instances and a pathfinding-and-"tower placement" sandbox area. This abstraction of objects into four distinct classes allows for modular updating and drawing of each component of the game world.

The game demonstrates **two** collision response algorithms. Projectiles fired from the enemy tanks or the player, performed by the ProjectileManager, are tested for intersection against the world and the player - when an intersection occurs, a sound plays, and in the case that it intersects the player, damage is taken and the camera shakes using a Thread delegate function. Enemy tanks implement a ray-sphere intersection algorithm to compute an opposing steering force, and steer away from the nearest collision threat.

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. The modular code also makes debugging particularly easy as there is no need to trawl through hundreds of lines of code to find a problem as each part can be viewed separatly.***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. Real world speeds for all entitys and the player camera makes for a more real experience. ***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. (Assuming we have a working fps and are aiming for a tower defense as the next part) The objective is to defeat the waves of enemy tankes as they attempt to attack and kill the player. The conflict is with the tanks attacking and the player returning the aggression on the tanks? (enemies, or are we the tanks?) The player receives a score when destroying a tank.

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* Documentation/Designer

**Score:** 0.0-5.0

Participated and helped with group discussion, contributed lab code to the project helped with elements of the read-me document.