Jole Challinor

Fundamentals of Game Programming

Assignment 1

**Noun and Verb Analysis**

The **horizontal shooter** is a game where the player controls a **ship** that is “anchored” to the **left-hand side** of the **screen**. The player can move their **ship** **up and down**, and **shoots** **bullets** which fly to the **right-hand side** of the screen. **Enemies** appear off the **right-hand side** of the **screen** and **move** to the **left-hand side** of the screen. The **player** can **destroy** these **enemies** with their **bullets**. If an **enemy** reaches the **left-hand side** of the **screen**, the **player loses** a **life**.

Nouns and Verbs

* Horizontal Shooter
* Ship
* Left-Hand side
* Screen
* Up and Down
* Shoots
* Bullets
* Right-Hand side
* Enemies
* Move
* Destroy
* Loses
* Life

**UML Diagram of Game Classes**

Faded blue are structs and general objects, such as the score display, game controller, menu, and game object class.

Green is for the player and the player controller.

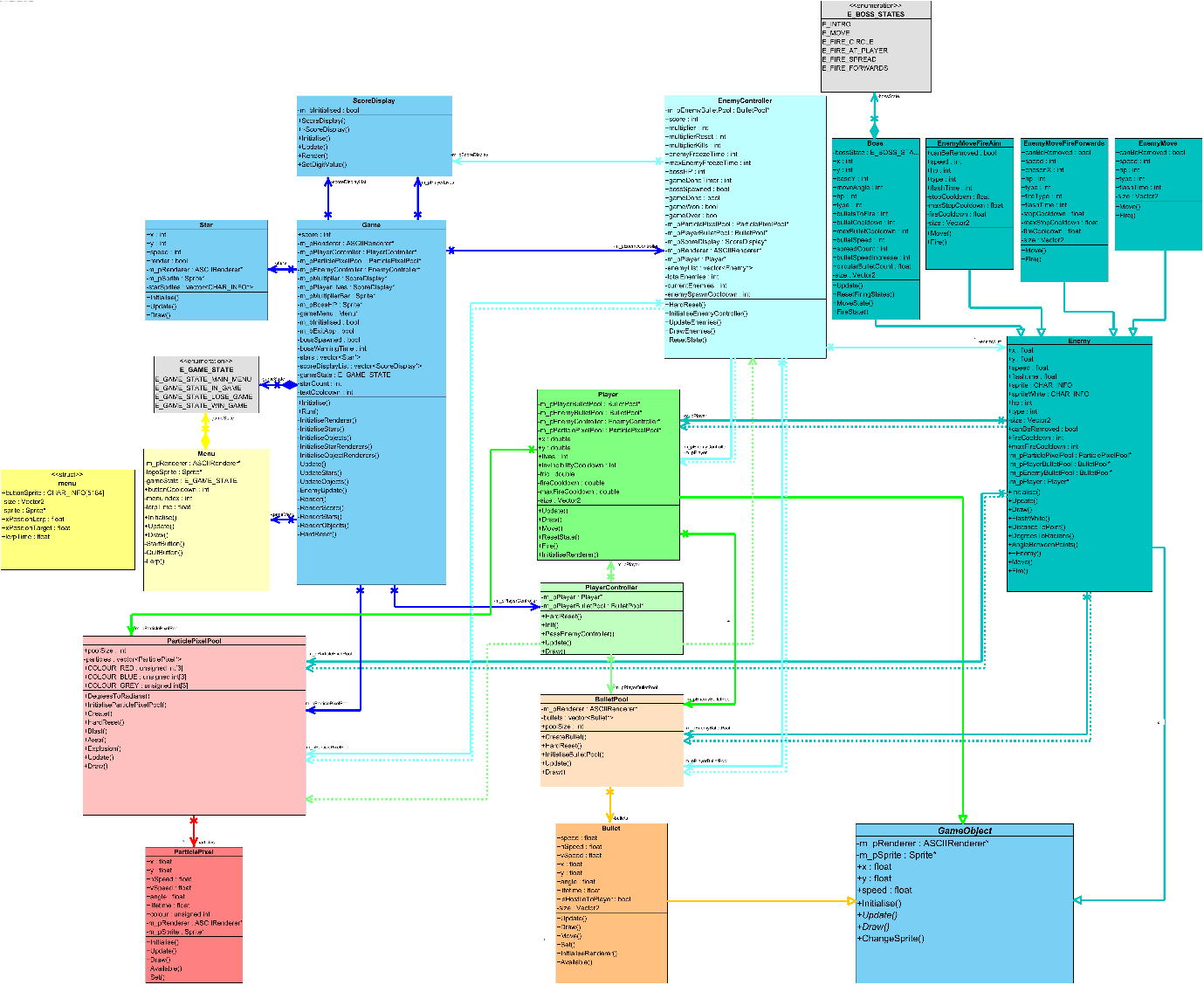
Orange is for the bullet and bullet controller.

Red is for the particle and particle controller.

Yellow is for the menu.

Cyan is for the enemy and enemy children, the light blue is for the enemy controller.

Grey is for enums.



**This image is included here so the report is more complete. However, a higher DPI version is available with the source files, and it should be accessed from there to be viewed in detail.**

**Extra Features**

Enemies

Enemy.h Enemy.cpp

* **Different enemy types –** Rather than there being a single enemy type, there are 3 that behave slightly differently. For example, enemy 1 behaves as required in the specification, but enemy 2 will occasionally stop and fire bullets at the player. Enemy 3 does not move to the left hand side but stops and moves up and down, firing a different pattern of bullets.
* **Enemies can shoot bullets –** Enemies can also shoot bullets to add more of a challenge. These bullets can be fired in different patterns and at different speeds.
* **Enemies have health –** Each enemy has 3 HP rather than dying in a single hit. Each enemy also flashes white when hit and display a smoke effect using a particle system when they reach 1 HP. Enemies lose 1 HP upon contact with a bullet.
* **Boss battle at the end –** Rather than the game ending after defeating all enemies, the player must defeat a boss. The boss has a lot more health (150), and over 5 different attacks. The boss does not move to the left hand side of the screen.
* **Enemies have different sprites –** Each enemy, and the boss, has a different sprite. This makes it much easier to differentiate the enemies.
* **Enemies have extra visual effects –** The enemies can fire particles like a “blast” when they fire bullets, and a trail behind them. They also create particles when dying for an explosion.

Player

PlayerController.h PlayerController.cpp

* **“Physics” for the player movement –** The player gradually speeds up and slows down with friction and bounces off the top and bottom of the screen. This makes movement more interesting.
* **Player has extra visual effects –** The player makes use of explosion effects with particles, as well as a trail, and a “blast” when they fire bullets.

Miscellaneous

* **Particle System –** The game has a particle system which can create effects with a single line of code. The particle system also makes use of several methods to allow for easy effects (like an explosion method rather than having to individually make particles to form an explosion). Found in ParticlePixel.cpp ParticlePixel.h
* **Star Background –** The game has a star background which makes use of multiple star objects. These objects fly to the left hand side of the screen, and are then moved to the right with randomised speed and positioning. Found in Star.cpp Star.h
* **Score Multiplier –** Killing enemies quickly increases a multiplier. This multiplier is applied to the score, and resets to 1 if the player dies or if they do not kill an enemy in time before it resets. Found in Enemy.cpp Enemy.h
* **Object Pooling –** Rather than reusing and destroying objects constantly, the game makes use of object pooling for several features (specifically the particles and the bullets). The object pools allow for objects to be reused, and removes the extra lag and slowdown from constantly destroying and creating new objects.

**Testing Phase**

N/A is evidence that cannot be effectively shown in a screenshot.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Location / Object** | **Input** | **Purpose** | **Expected Output** | **Comment/Verified** | **Evidence Location** |
| 1 | Star.cpp/Game.cpp |  | Check to see whether star background works correctly. This will run everywhere | Stars are constantly visible and moving correctly | Working | See All Screenshots |
| 2 | Star.cpp |  | Check to see whether stars are updating correctly | Stars are moving to the left hand side of the screen | Working | See All Screenshots |
| 3 | Star.cpp |  | Check to see whether stars are drawing correctly | Stars are drawing correctly | Working | See all Screenshots |
| 4 | Menu.cpp |  | Check menu image loaded correctly | Image loads | Working | See Menu Screenshots |
| 5 | Menu.cpp |  | Check start button image loaded correctly | Image loads | Working | See Menu Screenshots |
| 6 | Menu.cpp |  | Check quit button image loaded correctly | Image loads | Working | See Menu Screenshots |
| 7 | Menu.cpp | Space key | Check start button works correctly | Game state switches to E\_GAME\_STATE\_IN\_GAME | Working | N/A |
| 8 | Menu.cpp | Space key | Check quit button works correctly | Game quits | Working | N/A |
| 9 | Enemy.cpp |  | Check to see if enemy 1 (green) is rendering | Enemy 1 will render when on screen | Working | See Game Screenshots |
| 10 | Enemy.cpp |  | Check to see if enemy 2 (red) is rendering | Enemy 2 will render when on screen | Working | See Game Screenshots |
| 11 | Enemy.cpp |  | Check to see if enemy 3 (yellow) is rendering | Enemy 3 will render when on screen | Working | See Game Screenshots |
| 12 | Enemy.cpp |  | Check to see if Enemy Controller is correctly creating and removing enemies | Enemies will spawn once all enemies are dead, and enemies will be removed correctly with no junk memory left behind | Working | N/A |
| 13 | Enemy.cpp |  | Check to see if Enemy 1 moves to the left hand side of the screen, and dies upon contact with the barrier | Enemy 1 will die upon reaching the left hand side of the screen. The game will respond by killing the player, lowering life count by 1, and then causing a soft reset | Working | N/A |
| 14 | Enemy.cpp |  | Check to see if Enemy 2 moves to the left hand side of the screen, and dies upon contact with the barrier | Enemy 2 will die upon reaching the left hand side of the screen. The game will respond by killing the player, lowering life count by 1, and then causing a soft reset | Working | N/A |
| 15 | Enemy.cpp |  | Check to see if Enemy 3 moves to the correct position, and then proceeds to alternate between moving up and down. | Enemy 3 will stop at its chosen position, and then alternate between moving up and down | Working | N/A |
| 16 | Enemy.cpp |  | Check to see if Enemy 1 takes damage in response to bullets | Enemy 1 will take damage upon a bullet reaching a distance threshold | Working, Collisions slightly wonky | N/A |
| 17 | Enemy.cpp |  | Check to see if Enemy 2 takes damage in response to bullets | Enemy 2 will take damage upon a bullet reaching a distance threshold | Working, Collisions slightly wonky | N/A |
| 18 | Enemy.cpp |  | Check to see if Enemy 3 takes damage in response to bullets | Enemy 3 will take damage upon a bullet reaching a distance | Working, Collisions slightly wonky | N/A |
| 19 | Enemy.cpp |  | Check to see if Enemy 1 will flash in response to bullet damage | Enemy 1 loads the sprite correctly, and switches to it before switching back correctly | Working | See Game Screenshots |
| 20 | Enemy.cpp |  | Check to see if Enemy 2 will flash in response to bullet damage | Enemy 2 loads the sprite correctly, and switches to it before switching back correctly | Working | See Game Screenshots |
| 21 | Enemy.cpp |  | Check to see if Enemy 3 will flash in response to bullet damage | Enemy 2 loads the sprite correctly, and switches to it before switching back correctly | Working | See Game Screenshots |
| 22 | Enemy.cpp |  | Check to see if Enemy 1 will display smoke particles upon reaching 1 HP | Enemy 1 displays smoke (grey) particles correctly | Working | See Game Screenshots |
| 23 | Enemy.cpp |  | Check to see if Enemy 2 will display smoke particles upon reaching 1 HP | Enemy 2 displays smoke (grey) particles correctly | Working | See Game Screenshots |
| 24 | Enemy.cpp |  | Check to see if Enemy 3 will display smoke particles upon reaching 1 HP | Enemy 3 displays smoke (grey) particles correctly | Working | See Game Screenshots |
| 25 | Enemy.cpp |  | Check to see if Enemy 1 will die upon reaching 0 HP, and create an explosion | Enemy 1 is destroyed upon reaching 0 HP, and creates an explosion with particle effects | Working | N/A |
| 26 | Enemy.cpp |  | Check to see if Enemy 2 will die upon reaching 0 HP, and create an explosion | Enemy 2 is destroyed upon reaching 0 HP, and creates an explosion with particle effects | Working | N/A |
| 27 | Enemy.cpp |  | Check to see if Enemy 3 will die upon reaching 0 HP, and create an explosion | Enemy 3is destroyed upon reaching 0 HP, and creates an explosion with particle effects | Working | N/A |
| 28 | Enemy.cpp |  | Check to see if Enemy 2 will fire bullets at the player | Enemy 2 stops perodically, fires bullets at the player, then continues moving | Working | N/A |
| 29 | Enemy.cpp |  | Check to see if Enemy 3 will fire bullets and move correctly | Enemy 3 moves into position, and then alternates between moving up and down. It will periodically stop to fire it's choice of 3 possible bullet patterns, then repeat from moving up and down | Working | N/A |
| 30 | Game.cpp |  | Check to see if Boss warning is displayed correctly | Boss warning is displayed for several seconds upon boss creation, and then disappears (image also loads correctly) | Working | See Boss Screenshots |
| 31 | Enemy.cpp |  | Check to see if Boss displays sprite correctly | Boss sprite is loaded correctly and renders correctly | Working | See Boss Screenshots |
| 32 | Enemy.cpp |  | Check to see if Boss teleports correctly, creating appropriate teleportation particle effect | Boss teleports correctly, using the particle pool explosion method to create blue particles | Working | N/A |
| 33 | Enemy.cpp |  | Check to see if Boss correctly attacks the player | The boss will reset all attack variables, then choose between 1 of many attacks to use against the player | Working | N/A |
| 34 | Enemy.cpp |  | Check to see if Boss will flash in response to bullet damage | Boss loads the sprite correctly, and switches to it before switching back correctly | Working | See Boss Screenshots |
| 35 | Enemy.cpp |  | Check to see if Boss will take damage in response to bullets | Boss takes damage in response to bullets, lowering HP by a randomised amount | Working | N/A |
| 36 | Enemy.cpp |  | Check to see if boss displays low health effects | Upon reaching the 3 hp thresholds, the boss will add another hp effect. These are smoke, fire, and mini explosions | Working, slightly unoptimised | N/A |
| 37 | Enemy.cpp |  | Check to see if Boss dies upon reaching low HP | Boss correctly dies upon reaching low HP, creating an explosion | Working | N/A |
| 38 | Enemy.cpp/  Game.cpp | | Check to see if Game switches to win state after killing the boss, and renders all sprites correctly | Upon boss defeat, the enemy controller and the game will correctly handle switching to game state E\_GAME\_STATE\_WIN\_GAME and renders all sprites correctly. They are then removed from memory | Working | N/A |
| 39 | Enemy.cpp |  | Check to see if Boss correctly spawns upon reaching 1 enemy | The enemy controller will correctly detect when 1 enemy is remaining, and use that slot to spawn the boss | Working | See Boss Screenshots |
| 40 | Player.cpp |  | Check to see if the Player correctly moves into position upon game start or after life loss | The player correctly moves into position | Working, could be faster (not performance wise) | N/A |
| 41 | Player.cpp | Up Key | Check to see if the Player correctly moves upwards, bouncing off the top of the screen | The player correctly moves upwards | Working | N/A |
| 42 | Player.cpp | Down Key | Check to see if the Player correctly moves downwards, bouncing off the bottom of the screen | The player correctly moves downwards | Working | N/A |
| 43 | Player.cpp | Space Key | Check to see if the Player correctly fires bullets, and creates particle effects | The player correctly fires bullets and uses the particle system to create a blast effect | Working | N/A |
| 44 | Player.cpp |  | Check to see if the Player renders sprites correctly | The player correctly loads and renders the sprite | Working | See Player Screenshots |
| 45 | Player.cpp |  | Check to see if the player explosion and trail effect work | The player will create a trail of particles and create an explosion upon death | Working | See Player Screenshots |
| 46 | Player.cpp |  | Check to see if the player dies upon contact with bullets | The player will die upon contact with an enemy bullet, within a distance threshold | Working | N/A |
| 47 | Game.cpp |  | Check to see if the game renders the UI correctly | The game renders and displays score correctly in the top left, alongside multiplier, lives and bars | Working | See Game Screenshot |
| 48 | ParticlePixel.cpp |  | Check to see if the particle system responds to method calls from other objects, correctly rendering, initialising, and moving particles | The particle system correctly responds and all particles work as expected | Working, could be faster (performance wise) | See Game/Enemy/Player/Boss Screenshots |
| 49 | Bullet.cpp |  | Check to see if the bullet pool correctly moves and renders bullets in it's pool | The bullet pool will correctly move bullets assigned to it, and load and render sprites correctly | Working | See Game/Enemy/Player/Boss Screenshots |
| 50 | Game.cpp |  | Check to see if the game correctly switches to lose game state, and renders sprites correctly | The game correctly switches to game state E\_GAME\_STATE\_LOSE\_GAME, and renders all sprites correctly. They are then removed from memory |  | N/A |
| 51 | Game.cpp |  | Check to see if game correctly performs hard reset upon return to E\_GAME\_STATE\_MAIN\_MENU state, calling hard reset methods for the player, bullets, particles, and enemies | The game correctly calls these methods, and upon starting the game again (E\_GAME\_STATE\_IN\_GAME), it is as if the game has reset. The game does not reinitialise objects, to conserve memory and make programming easier | Working | N/A |

**Screenshots**

Menu Screenshots:

A screenshot of a computer

Description automatically generated with medium confidence

Game Screenshots:

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Boss Screenshots:

A screenshot of a video game

Description automatically generatedA screenshot of a video game

Description automatically generated

Graphical user interface

Description automatically generated

Player Screenshots

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

End Screenshots

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence