**RegNo: 1702097**

**Asteroids Manual:**

**The Game:**

Asteroids is a game inspired by Atari’s hit arcade game with the same name that was released in 1979. The game allows to pilot a ship across the galaxy avoiding obstacles and defeating enemies that wish to rob you of your treasures. As you progress thorough the game the levels get increasingly harder to navigate, are you up for the challenge ?

**How to play:**

**Before you begin make sure you’re running the latest version of JAVA Or ASTEROIDS.jar may not start !**

1. **In the ASTEROIDS folder go to ASTEROIDS\_jar / ASTEROIDS.jar and start playing.**
2. **If the game does not start Try updating your version of Java.**
3. **Alternatively go to ASTEROIDS/src/game1/GAME.java and run it in a Java IDE to play the game.**
4. **The .jar files can be run using the IDE also.**

**CONTROLS:**

**Up Directional key:** Thrust

**Left Directional key:** Turn Left / Rotate Anti-clockwise

**Right Direction key:** Turn Right / Rotate Clockwise

**SPACEBAR :** Fire weapon

ESC

**ESCAPE:** Quit Game

**ENTER**: Restart Game after GAME OVER

**THE SHIP:**

The ship allows you to interact with your surroundings, destroy asteroids and kill enemies :



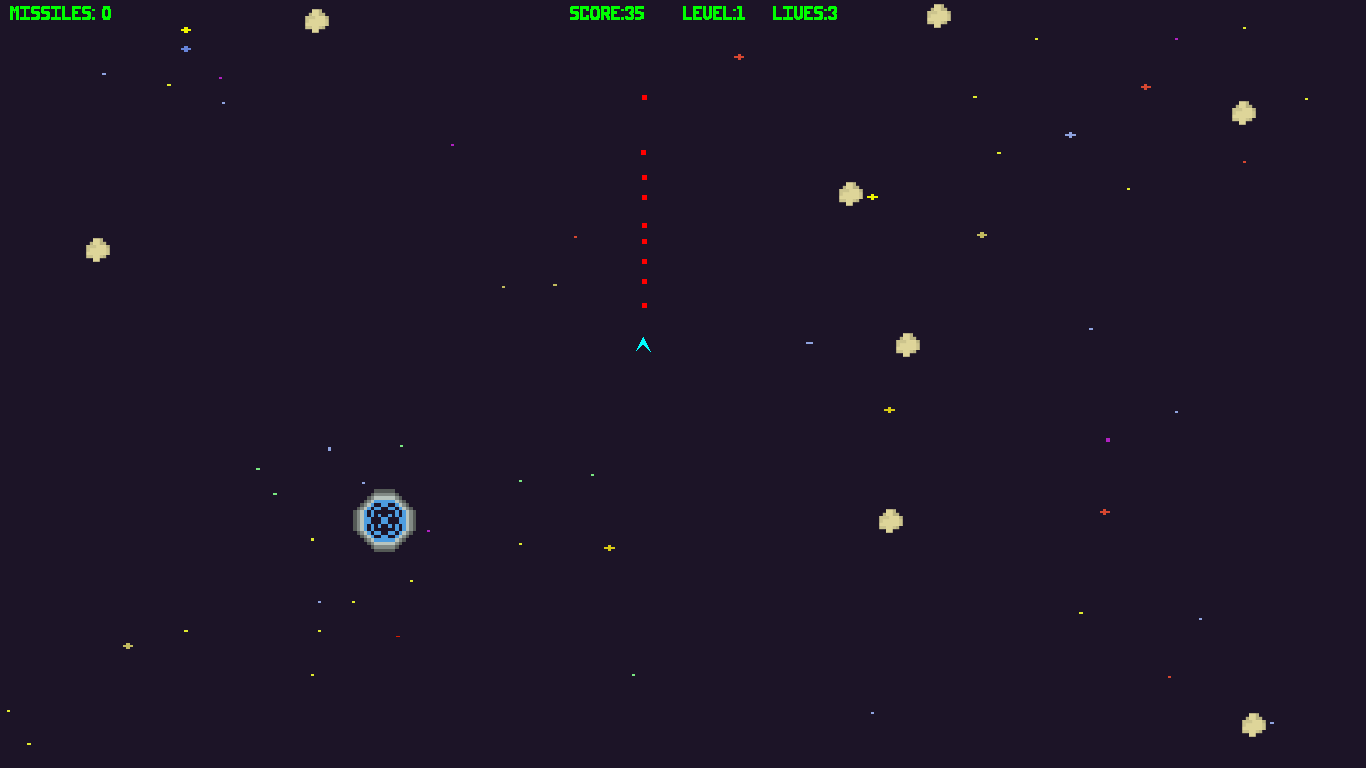
You can move the ship by using the UP directional key, which will cause it to thrust in the direction the it is pointing in and leave a trail:



When the game begins the ship will be drifting through space, surrounded by asteroids or an enemy.



The ship can fire weapons, by default the fire weapon key will let the user fire lasers from the ships turrets. However upon collecting the Missile powerup the player will be able to shoot a missile using the fire weapon key.



Shooting asteroids gains the player 5 points and shooting down an enemy ship will cause the player to gain 20 points.

Colliding with an asteroid or an enemy ship, will cause the player to loose one life and respawn, killing the other object in the process.

**Enemies:**



Enemies look like red versions of the player, they are not affected by asteroid. The aim of the enemy in the game is not only to kill the player but also reduce the number of points available to the player. The enemy bullets will destroy the asteroids as well as the player ship.

Enemies fly from left to right of the screen in a straight line and star shooting once the player is in range, destroying anything in front of them.

Each level has a 50% chance of spawning an enemy ship.

**Wormholes:**

****

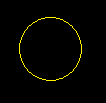
Wormholes allow game entities to travel from one part of the map to the other. They can be used by the player to escape areas that densely populated with asteroids and enemies. However they also transport other entities to various parts of the screen, making the map highly unpredictable.

Each level has one Wormhole.

**Power-ups:**

1. The missile power up allows the player ship, to use the missile attack.

Each level after level 1 has a 33% chance of spawning this powerup.



1. The health gain power up adds an additional life to the player’s current lives. Each level after level 1 has a 25% chance of spawning this powerup.



1. The 100 health gain power up is the rarest power up in the game with a spawn chance of 2% and adds 100 lives to the player.



**Other Entities:**

**Asteroids:**

Asteroids have a random moving patterns, destroying asteroids adds 5 points to the player’s score and will result in a respawn if the player collides with one.



**Missiles:**

Missiles are added to the player’s arsenal once they collect the missile power up. They have a larger impact radius than the laser and can be used to clear a densely asteroid-populated area, especially useful for higher levels.



**DEVELOPMENT REPORT:**

The software is designed and implemented with the idea of the of making the game a classical arcade shooter, much like the game that this project is inspired on. The game uses a Vector2D class that is used to implement and simulate basic physics and vectors functionalities. The graphics are built using java swing and awt libraries, no external libraries are used to implement any functionality. The graphics are sourced from itch.io (see below), they have been selected to give a more classical 8-bit feel while feeling updated.

**PARAMETERS**

The 3 main parameters of the Game objects that affect the gameplay to high degree are the directional, positioning and velocity vectors. Tuning these drastically changes the gameplay experience and physics of the game.

**Position vector** : Contains the x and y co-ordinates of a game object, this is used to position objects on the screen. Adding the object’s velocity vector weighted with the delay time of the thread, thereby moving the object.

**Velocity vector :** Containsthe X coordinate velocity and Y coordinate velocity of the game object, this when added to the position of the object is what allows the object to move from one object to the other. Subtracting a drag factor from the velocity creates the drift effect on the player ship when static.

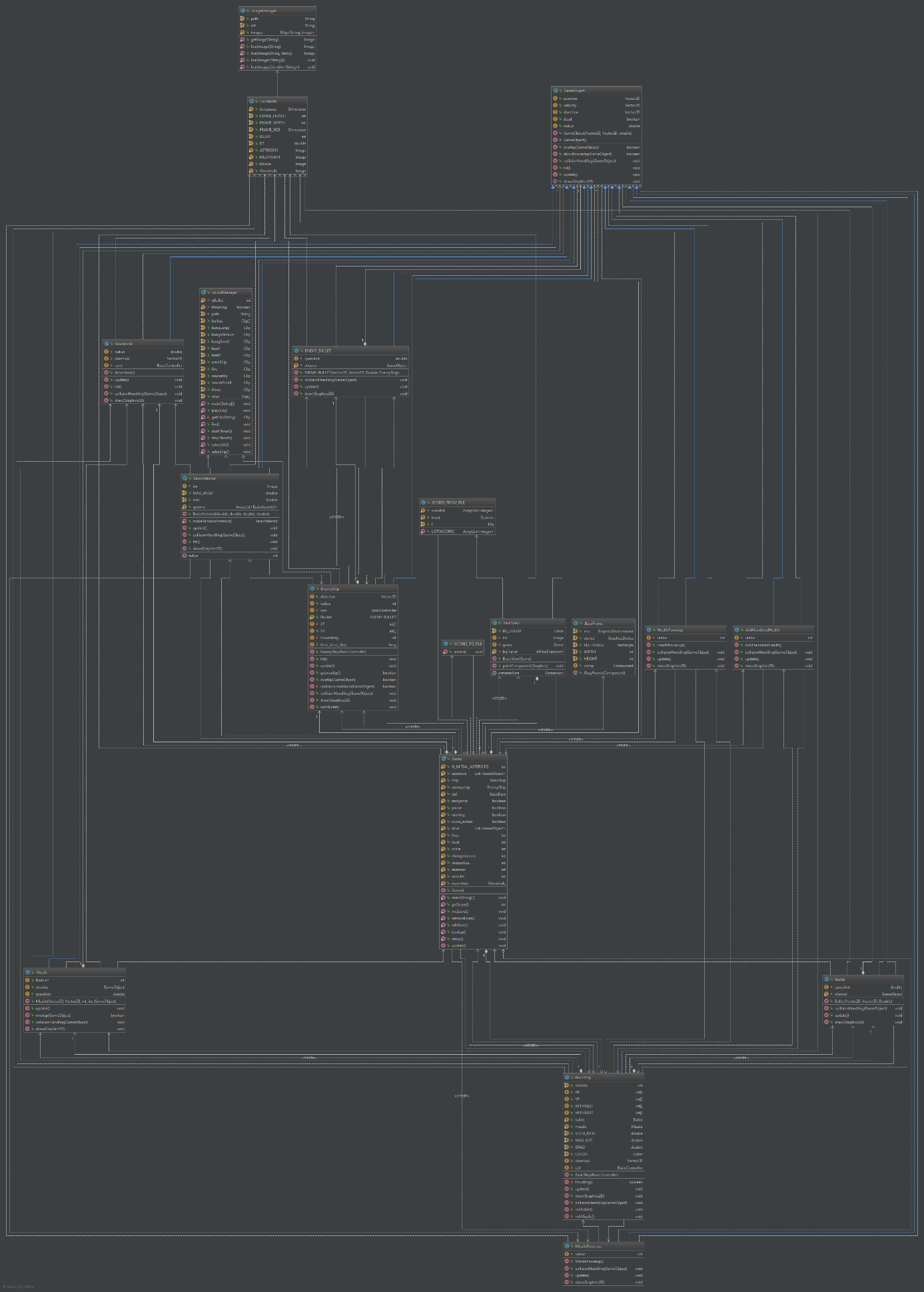
**Direction vector:** Contains the X and Y coordinates of towards the direction in which the object points to. This is essential in the shooting mechanisms of the player ship and enemy ship.

The third parameter that required tuning was the delay time of the thread which determines the update cycle of the game. This was tuned so that the game and the audio/visual effects worked at a optimal capacity.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SEE NEXT PAGE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**UML CLASS DIAGRAM:**

The diagram below describes the relationships of the classes in the game:



A clearer version of the image is uploaded at : <https://imgur.com/QfShiHc>

**Appraisal:**

The project was completed in the span of 2-3 Months of work through the CE218 labs, many of the classes and objects were modelled by the CE218 lab scripts. The project went through several iterations of testing as well as several stages of development.

While the game was based on the labs in CE218, several classes, objects and structures were designed independently by me. The missiles, graphic choices and powerups were designed by me and I am very satisfied with how they turned out. The key structure however, being the GameObject class was based on the CE218 labs and allowed the implementation of a plethora of in game objects such as bullets, ships, wormholes etc.

A features that was harder to implement than it seemed was the enemy craft A.I, this feature required a lot of time to implement as the idea for it was to not just target the player but also reduce the points available to the player by destroying the asteroids. The A.i. produced in the end is simple yet very effective targeting the player when they are in range.

There are several issues in the game yet to be resolved, such as concurrency issues which while have been minimized have not been completely removed. Hence the user may sometimes experience the loss of an extra life when damage is taken.

The sourcing of the graphics and the implementation of the Game A.I. are parts of the project that I am most proud of as they were the elements for me that severely enhanced the gameplay and allowed me to use my creativity to solve them.

**Resources used:**

**Astroids, WormHole and BackGround:** [**https://hexadecimalwtf.itch.io/space-pixels**](https://hexadecimalwtf.itch.io/space-pixels)

**Font by Jupiter Hadley:** [**https://jupiter\_hadley.itch.io/jupiters-font-bundle**](https://jupiter_hadley.itch.io/jupiters-font-bundle)