**GunFun Documentation**

Contents

1) Welcome to GunFun ----------------------------------------------------------------------------------------------------- [3](#WelcomeToGunFun)  
 1.1) What is GunFun? --------------------------------------------------------------------------------------------- [3](#WhatIsGunFun)  
 1.2) Controls --------------------------------------------------------------------------------------------------------- [3](#Controls)  
 1.3) How to install/compile -------------------------------------------------------------------------------------- [3](#HowToInstall)  
 1.4) Assets ----------------------------------------------------------------------------------------------------------- [3](#Assets)

2) Steering Algorithms ------------------------------------------------------------------------------------------------------- [4](#SteeringAlgorithms)  
 2.1) Seek -------------------------------------------------------------------------------------------------------------- [4](#Seek)  
 2.2) Separation ----------------------------------------------------------------------------------------------------- [5](#Separation)  
 2.2) Flock ------------------------------------------------------------------------------------------------------------- [6](#ComplexSteering)

3) Other Algorithms ---------------------------------------------------------------------------------------------------------- [7](#OtherAlgorithms)  
 3.1) Shoot ------------------------------------------------------------------------------------------------------------ [7](#Shooting)

4) Bugs --------------------------------------------------------------------------------------------------------------------------- [8](#Bugs)  
 4.1) Gun Rotation ------------------------------------------------------------------------------------------------- [8](#GunRotation)  
 4.2) Laser Rotation ------------------------------------------------------------------------------------------------- [8](#LaserRotation)  
 4.3) Lag ---------------------------------------------------------------------------------------------------------------- [8](#Lag)

5) References ------------------------------------------------------------------------------------------------------------------ [9](#References)

**1 WELCOME TO GUNFUN**

This section will cover what GunFun is, how to install GunFun and how to play the game. The section also provides the controls for the game along with links to the assets used.

**1.1 What is GunFun?**

GunFun is an exciting 2d space shooter game where you are in charge of a skilled space pistol, yup you heard it right, you play as the Pistol. With the mouse as your trusty weapon, you’ll engage in intense battles where quick reflexes and precise aiming are crucial for survival.

The objective of the game is to defend yourself from the relentless horde of incoming aliens. Each alien killed gives you 10 points, but if you kill the leader alien which spawns once in a while (the alien in the big blue spaceship), you’ll get 50 points! Be careful though, the other aliens will do everything they can to protect their leader alien by flocking around it.

Initially, the aliens spawn at a moderate rate, but the more you kill the more they spawn. The game ends whenever an alien comes in contact with your Pistol.

**1.2 Controls**

Here are the controls of the game

WASD – Gun Movement  
Mouse – Gun Rotation  
Left Mouse Button – Shoot

**1.3 How to install/compile**

To install/compile the game:

1. Open the project in Android Studio
2. Open Android Studio Terminal
3. run “./gradlew desktop:dist”
4. The output file desktop-1.0.jar will be found in “TMA1\GunFun\desktop\build\libs”

Note: Under one occasion, I have noticed that the jar file just stopped working. I’m not sure if that’s an issue on my end but after running the command in step 3 again, I could not replicate the issue. In any case, if the jar file stops working. Try running the command in step 3 again.

**1.4 Assets**

All assets used in the game can be found in the GunFun/assets folder. There are a couple of extra unused assets which I might include in a later version of the game.

* The Gun texture is part of an assets pack and has been borrowed from: <https://arcadeisland.itch.io/guns-asset-pack-v1>
* The Laser texture is part of a laser textures pack and has been borrowed from: <https://opengameart.org/content/assets-free-laser-bullets-pack-2020>
* The background texture has been borrowed from:

<https://images.pexels.com/photos/176851/pexels-photo-176851.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1>

* The Alien texture has been borrowed from:

<https://www.pngwing.com/en/free-png-pfqla>

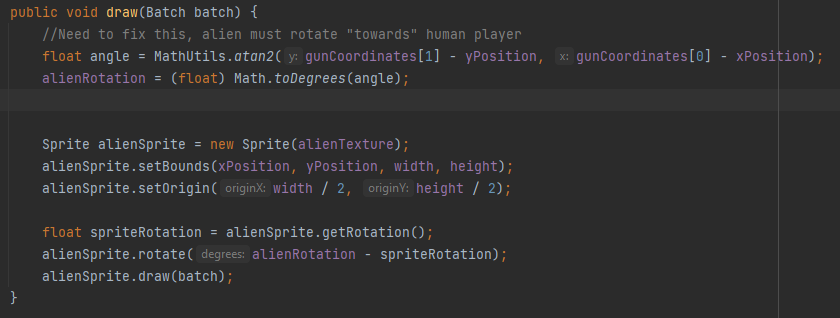
**2 STEERING ALGORITHMS**

This section talks about the steering algorithms that have been used in the game. Overall I have used 2 basic steering algorithms: seek and separation, and 1 complex steering algorithm: flock. In addition, I have also implemented shooting which I will talk about in section 3.

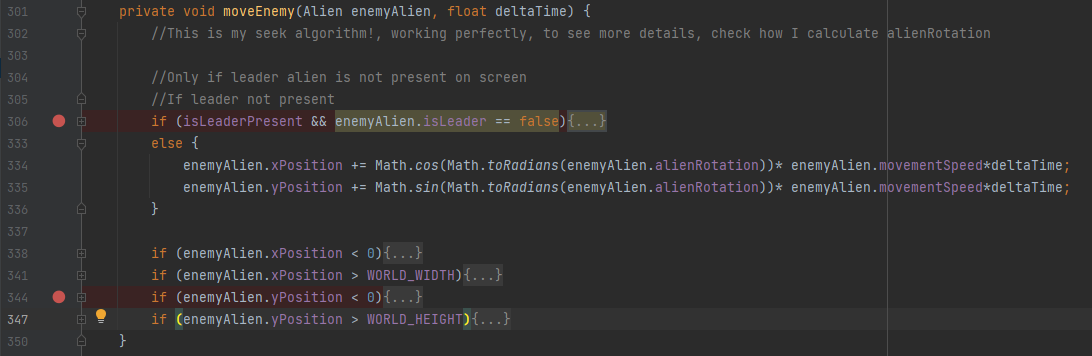
**2.1 Seek**

I have used a simple kinematic seek algorithm for the aliens. By default the aliens seek towards the Gun at Max speed at all times unless the leader is present (Details in section 2.3)

Here’s how I implemented the seek algorithm:



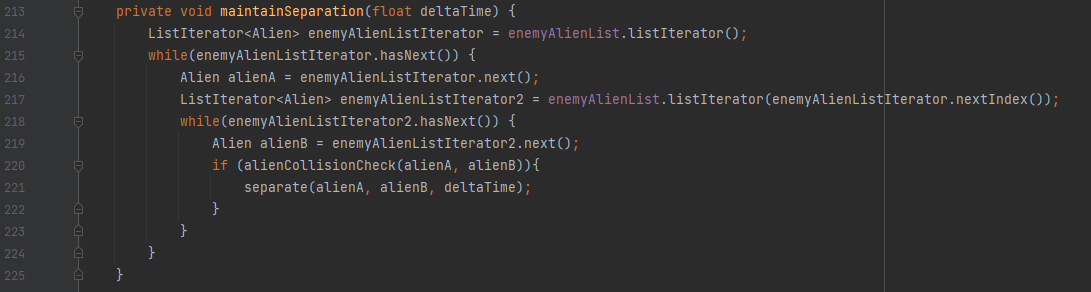
Every alien on the screen has the coordinates of the gun as one of its attributes. The angle of rotation is calculated using these coordinates and the coordinates of the alien itself. Next, the alien is rotated towards this angle.



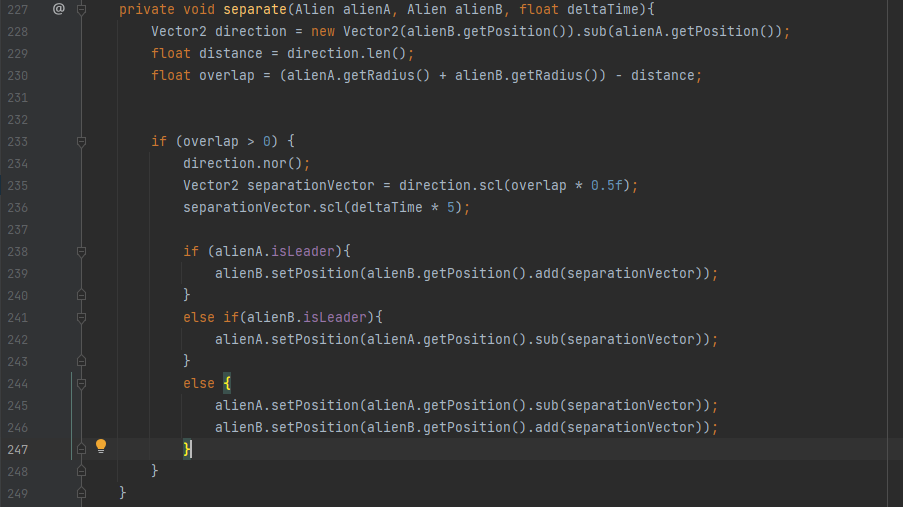
Lines 334 and 335 show how the default movement for the seek algorithm is done. The moveEnemy method is run every frame and the alien moves towards the direction of the userGun object at maximum speed.

**2.2 Separation**

The separation algorithm prevents collision between the aliens and makes the game look more realistic by preventing clustering. Here’s how I implemented the algorithm.



This maintainSeparation algorithm is called in the render method, hence it runs for every frame. Here there’s a nested loop where for every alien pair, the separate method is run, which creates a separation force for each alien.



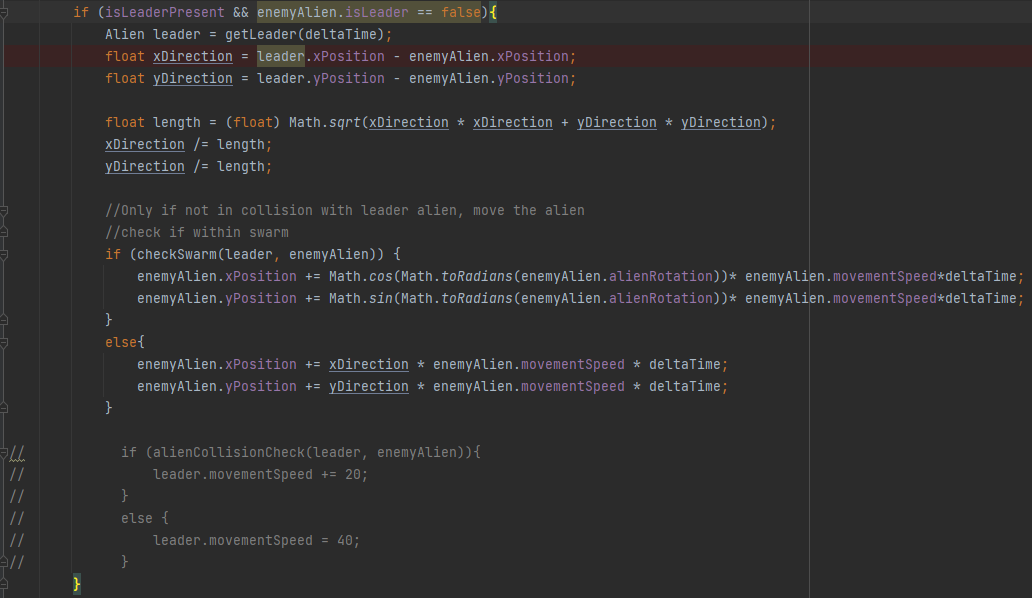
Now, inside the separate method, between lines 228 and 230, we first calculate the existing overlap between the two aliens. If there is an overlap, depending on the overlap, we create a separation vector and then scale it according to deltatime.  
  
Now the logic between lines 238 and 243 is for the separation between a leader alien and a follower alien. This prevents the leader from getting pushed away by the follower alien. This logic helps such that the leader alien’s position is not modified by the follower alien but instead, only the position of the follower alien is modified by the leader alien. This actually fixes a bug where previously the motion of the leader alien was disrupted by the separation force of the follower aliens.

The logic between lines 244 and 247 is for the separation between two ordinary aliens. Depending on the overlap, each alien affects the position of another in such a way so that the overlap decreases or becomes 0.

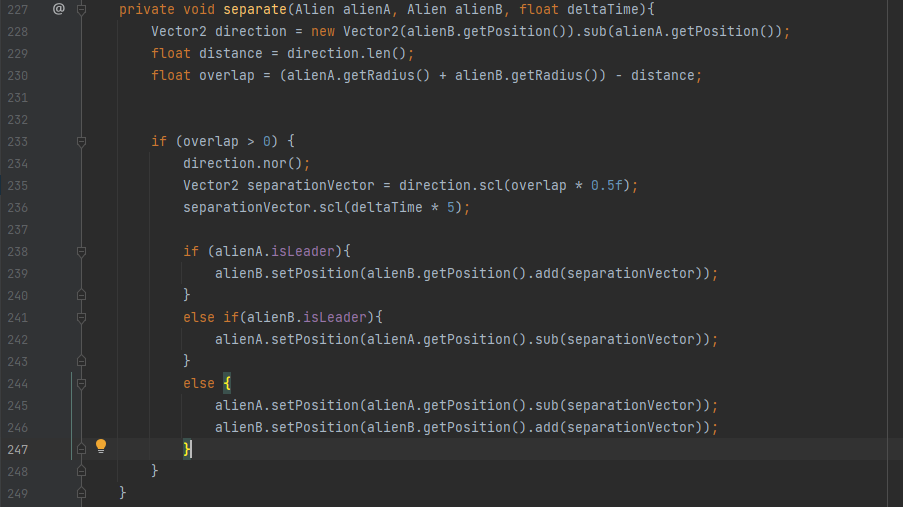
**2.3 Combined Steering Behavior – Flocking/Formation**

Now, combining this seek, separation and another cohesion algorithm that I wrote, I created a flocking algorithm for the aliens. Every now and then, a leader alien appears on the screen which you can kill for bonus points. But not so fast! – The moment the leader alien appears on the screen, all other aliens flock to it, and surround it to protect it. The leader alien now seeks the pistol surrounded by his flock of aliens. This makes the leader alien particularly dangerous. Unless you want to be faced by a giant horde of aliens coming from one direction, take out the leader alien fast!  
Here’s how I implement this behavior.

Whenever the leader alien is present on the screen, all aliens seek towards the leader alien. This logic can be seen here in the moveEnemy method of the GameScreen class:



If the aliens are within a specific radius of the leader, the movement of the leader and the follower aliens is automatically coordinated. Otherwise, the aliens all seek towards the leader. As discussed in the previous section, the separation logic for the leader is a bit different from the other aliens and is handled here:



Refer to the previous subsection to understand how the separation logic works for the leader. These combined steering algorithms cause the leader to be automatically surrounded by the follower aliens, and then in turn, the entire group follows the leader alien towards the pistol.

**3 OTHER ALGORITHMS**

This section will mainly talk about the shooting algorithm.

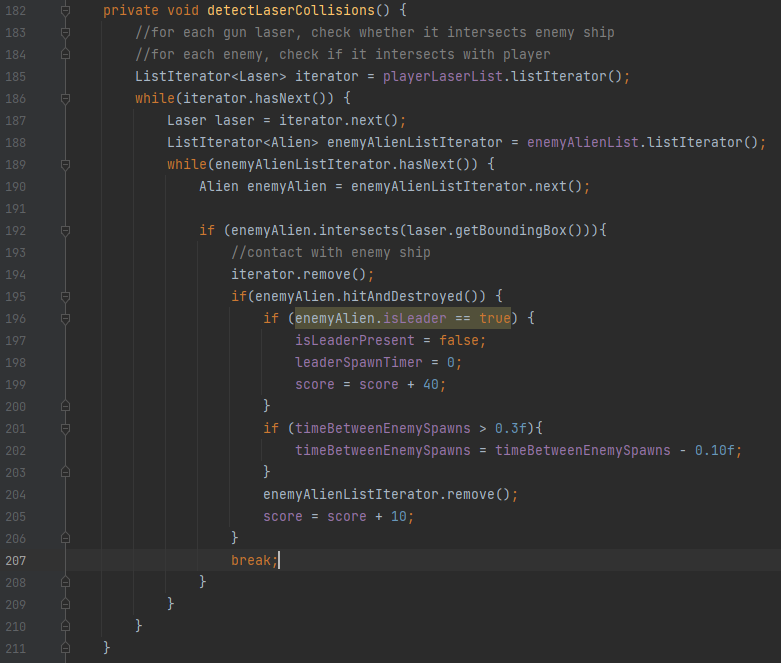
**3.1 Shooting**

The game has a simple linear shooting algorithm. When the user presses the left mouse button, the Gun fires a laser with constant speed. The reason that I did not use any forces of acceleration/gravity on the laser is because since the game takes place in space, there are no forces present. The shooting is mainly implemented through a Laser class and this fireLasers() method.



Here, every time the user presses the left mouse button, a Laser object is added to a list iterator objects. Inside the render function in the GameScreen class, all Laser objects in the list iterator are rendered every frame.

To detect collisions between the Laser objects and aliens, I use the detectLaserCollisions() method in the GameScreen class. I will explain the main features of the method below:



Through a nested loop, we check for every laser if it collides with any alien. If it does, we remove the alien from the list iterator of Aliens. We also handle increasing the score upon every alien hit in this function. On top of that, the spawn rate of the aliens increases with every alien kill until the spawn rate reaches 0.3s. The max fire rate of the Gun is 0.2s, this means a perfect player can theoretically play the game endlessly (although it becomes very difficult)

**4 BUGS**

This section discusses the bugs that I have encountered in the game.

**4.1 Gun Rotation**

One problem with the game is that the Gun rotation is not 100% accurate. The gun points in the general direction towards which the mouse pointer is present but the direction of rotation is not 100% accurate. Because of this players might have to adjust their aim accordingly.

**4.2 Laser Rotation**

Since laser rotation is tied to gun rotation in my code, the buggy gun rotation also affects laser rotation. Moreover, the starting position of the laser should be adjusted to the nozzle of the gun for each rotation angle.

**4.3 Lag**

Whenever memory intensive applications such as Chrome and Discord runs in the background, the game seems to lag whenever the user moves the mouse very fast. For optimal experience, I recommend closing all background applications before playing GunFun.

**5 REFERENCES**

I learned libgdx from this youtube playlist:   
<https://www.youtube.com/watch?v=DK1sGc4rOT4&list=PLfd-5Q3Fwq0WKrkEKw12nqpfER3MG5_Wi&index=3>

There might be similarity in the code in this video, and my code as I’m new to libgdx and took help from this tutorial to learn how to set up my objects. However the code for the steering algorithms and the shooting mechanism is my own.