Assignment 2
Boids

Trym Varland

UiT - Norges Arktiske Universitetet

February 20, 2023

1 Introduction

In the second assignment in INF-1400, we are going create the classic flock simulator boid. The implementation will be written in python, with the goal of using principals of object-oriented programming as classes, inheritance and methods. The simulator should simulate a moving flock of boids moving in life like manner, and some set of rules. The simulation should also consist of hoiks and obstacles. The rules the boid need to follow:

1. Boids steer towards the average heading of local flockmates.
2. Boids steer towards the average position of local flockmates.
3. Boids attempt to avoid crashing into other boids.

The simulator should also contain:

- Obstacles the boids need to avoid.
- Predators (hoiks) that will try to eat the boids.

2 Technical Background

2.1 Inheritance

Inheritance creates the opportunity for a class to inherits methods and properties from other classes. The parent class, is the class it's inherited from. Child class inherits from the parent class (or another class). This gives us the opportunities, to give real-world relationship. [1]

2.2 Boids

The flock simulator was written by Craig Reynolds in 1986, and had the goal to mimic coordinated animal motion such as birds and fish's. This is done by, having three steering behaviors (see Fig. 1): [2]

1. Separation: Steer to avoid crowding local flockmates (avoid crashing).
2. Alignment: Steer to the average heading of local flockmates.
3. Cohesion: Steer to move the average positions of local flockmates.

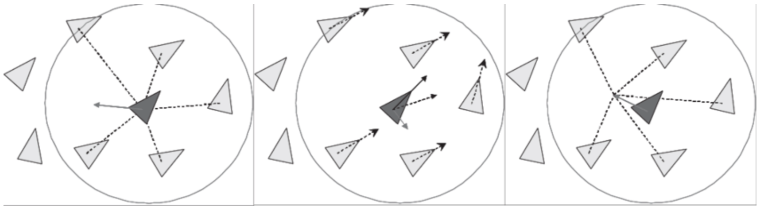

Figure 1: Rigth image: illustrates the separation, middle image: shows the alignment and the last image: illustrates the cohesion. [3]

3 Design & Implementation

3.1 Moving

Moving is the parent class in the program, which means that it contains methods used in the children classes. As illustrated in figure 2, it have several methods. The rules in sub section 2.2, is not followed in numeric order. Reason, it made more sense by do it this way.

3.1.1 Move & update

The move method have the function to make boid and hoiks appear on the other side of the screen if they go out of the screen. Update method, updates the position with the velocity.

3.1.2 Alignment

The alignment method, is the second rule. Which makes the boid match the velocity of surrounding boids (local boids). The method works as follow; If a boid have a smaller distance than the perceived distance, it appends to the local boids list. If the number of local boids is bigger than one, all local boids corresponds/updates to the velocity vector.

3.1.3 Cohesion

Cohesion method, is the third rule. An gives the boids, ability to steer towards the center of the local area. Method works as follow; Sets up the local boids, then sets all the local boids steer into the center of all local boids,

3.2 Separation

Separation method, is the first rule. Makes the boid to steer away from other boids in local boids, so that they don't collide. The method works as follow; If boid is equal to local boid and distance is smaller than the perceived distance, set as local boid and calculate the difference between the boid and local boid, than change the velocity vector accordingly.

3.2.1 Hunt to eat & avoid obstacles

The hunt to eat method, gives the hoiks ability to hunt and eat the boids. If a boid is in the hunt distance(perceived distance to hoiks), the hoik change it's velocity vector to catch up with boid. And when hoik collides with boid, the boid is removed from the boid list.
Avoids obstacles method, gives boids and hoiks ability to not collide with the a obstacle.

3.3 Boids

The boids class, consists of three methods and inherits from Moving class. The first method rotate, just rotates the corresponding png file in the direction of the velocity vector. Avoid hoiks method, makes the boids avoid the hoiks. by adjust the velocity vector, opposite of hoik. draw and behavior method, draws the boids and it's behavior.

3.4 Hoiks & Obstacles

Class hoiks, have 2 methods. Where one is to rotate the hoik png file according to the velocity vector, and the other to draw the hoik and it's behavior.
Class obstacles, only consists of one method. That's the draw method, an it's only used to draw the obstacle png file.

3.5 UML diagram

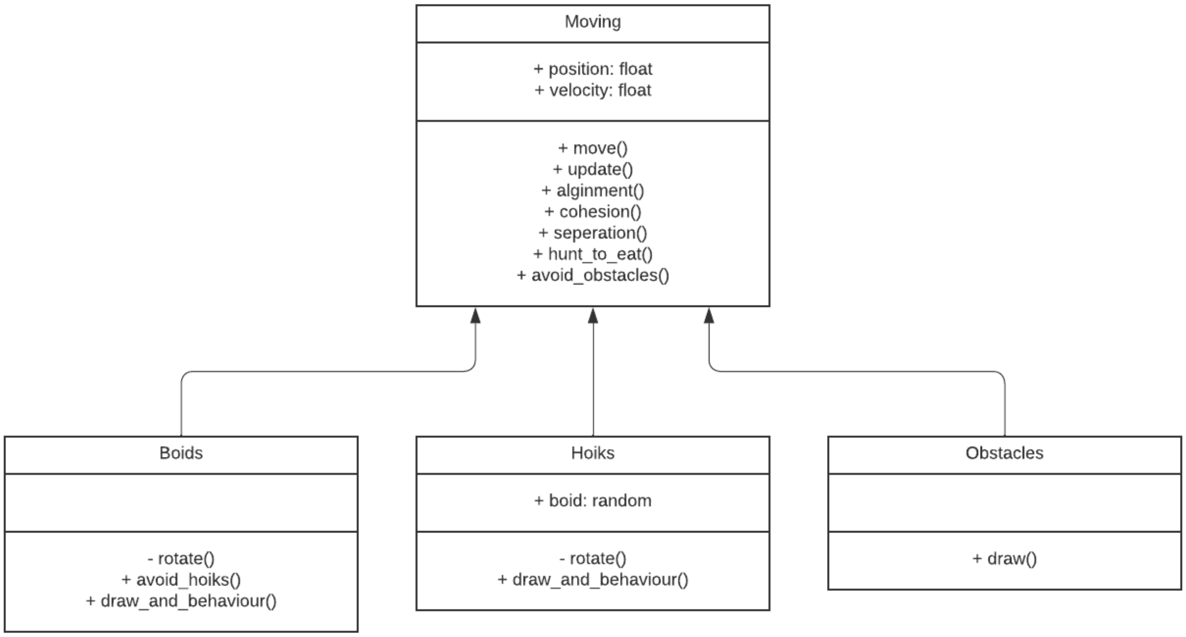

Figure 2: UML diagram to file: boids_simulation.py

4 Conclusion

The goal of the assignment was to recreate the classic flock simulator boid, by using principals of object-oriented programming. Which is accomplished, by having a parent class called moving and children classes boids, hoiks and obstacles. We also accomplished to recreate the flock simulator, which is seen by running the programming and see that the boids flock together and it is hunted by the hoiks.

5 Appendix

References

[1] Python Inheritance, February 2023. [Online; accessed 20. Feb. 2023].

[2] Boids (Flocks, Herds, and Schools: a Distributed Behavioral Model), July 2007. [Online; accessed 20. Feb. 2023].

[3] Figure 1: The three different types of steering behaviors. The first..., February 2023. [Online; accessed 20. Feb. 2023].