1. **Personal information:**

Flock simulator

Pyry Satama

896188

Computer Science

2020

27.4.2022

2. **General description:**

This program is a flock simulation where some amount of boids (read from a file) are moving according to three rules (alignment, cohesion, separation). The rules have different weights and user can change the weight in real time when the simulation is running.

3. **User interface**

After you run the program a black canvas with sliders and start button on the left-hand-side appears. Simulation starts when user clicks the start button. User can change the weight of the rules from the sliders on the left.

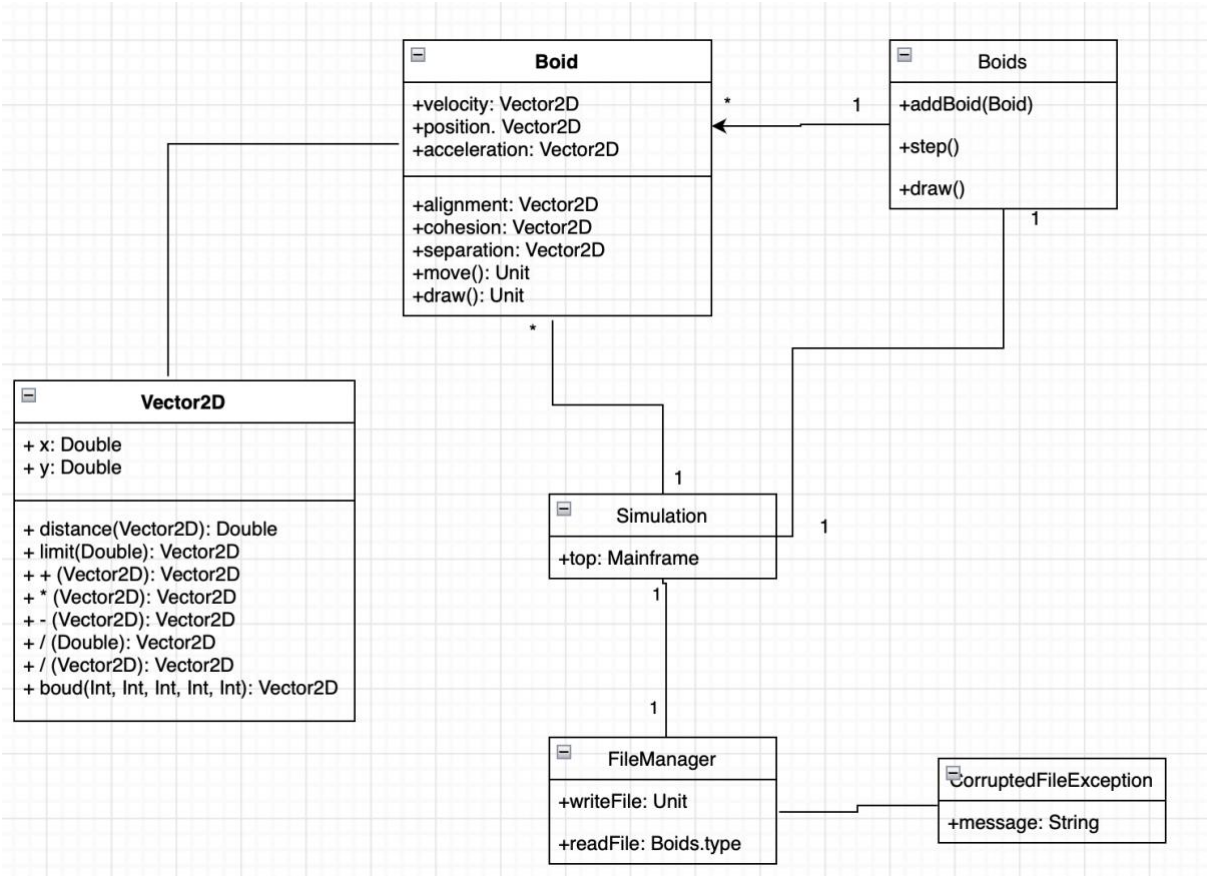
4. **Program structure**

I have 7 classes in my program: Boid, Boids, Vector2D, Simulation, FileManager, CorruptedFileException and a test class SimTests. Vector2D class includes 2D vector calculations that help in calculating boids vectors. Vector2D methods are being used only in Boid class. Test class "SimTests" test methods of Vector2D.

Simulation object takes care of the user interface. There is top function that creates the Mainframe and puts boids and sliders to the frame.

FileManager takes care of writing and reading the file where the initial state of the simulation is written. CorruptedFileException sends a message if something goes wrong in reading or writing the file. The file is written and read in Simulation object.

In object Boid there is a buffer of all the boids in the simulation. It also has methods addBoid that adds new boid to the buffer, step() that makes every boid in the simulation move, draw() that draws all the boids to the frame in Simulation object.



5. Algorithms

In this program, I had to make different vector calculations to get the boids move as they should (flocking). As I mentioned already, the methods in Vector2D class takes care of different calculations needed for the rules of the boids movement (alignment, cohesion and separation). The helpr methods for vector calculations are used in Boid class.

There is for example distance method that calculates distance between two points by calculating the hypotenuse of two tangents. The tangents are calculated by difference between the two points x and y values.

There is also limit method that limits a vectors length to specific length given to the method. This method check that is the length of the vector bigger than given double. If it is, method returns the same vector with a length of the given double. If the vectors length is smaller than given double, method just returns the same vector.

There are also methods for basic vector calculation such as vector summation, product of vector and some number, etc.

Lastly there's a bound method that makes the boid appear on the other side of the frame if it goes out of bounds. This method is a little bit more complicated. It checks boids position and if boid is over the bounds in either x or y direction then this method returns a Vector2D that is a position on the opposite side of the frame.

6. Data structures

I used mutable Buffer to store all the boids in the simulation. I think that Buffer was a good choice and it worked very good for my project. I could have also used for example arrays or lists but I think that everything went smoother using Buffers.

7. Files

My program deals with text files. It has writeFile and readFile methods in FileManager. File consists of all the boids that go into the simulation. File starts with 'BOIDS:' and end with 'ENDOFFILE'. Every boid in the file has Velocity, Location and Acceleration. There should be a file named "start" where the program writes initial state before running the simulation.

Example:

```
BOIDS:
Boid (Velocity, Location, Accerelation):
2.3649999380193156,1.0936930917512866
1087.6568512193091,654.4474040124993
0.0,0.0
-
Boid (Velocity, Location, Accerelation):
2.7400834016664595,0.38633929235080045
1385.6975125156596,560.4102832607815
0.0,0.0
ENDOFFILE
```

8. Testing

My programs testing process was done as planned. I tested Vector2D classes methods with unit tests in SimTests class. Other aspect such as movement of the boids was tested through user interface. Program passes all the tests done.

9. Known bugs and missing features

I haven't found any bugs in my program. Only thing that I noticed is that the simulation tends to be a bit 'laggy' when inserting bigger amount of boids into the simulation (about 400 or more) but this is probably because the amount of calculation happening simultaneously.

10. Best sides and weaknesses

I think that the movement of the boids came out really nicely, also I think that it's a good feature that user can modify the weights of the movement rules in real time. This makes simulation fun to play with.

Weaknesses for my program could probably be that the graphical user interface could be a little bit more interesting. For example the boids could have some other shape or more color could be added. I however decided to not make another class that could have made boid for example triangular and

also decided to make simulation just black and white because I like that it looks simple and not too flashy or complicated.

Movement could also make better with some rule that boid moves randomly when it's not near other boids. Now in my program the boid moves just forward when it's not close to others. This could be fixed just by doing another method to Boid class that gives a boid random direction when there is no other boids nearby similar to alignment, separation and cohesion.

11. Deviations from the plan, realized process and schedule

I started this project by first implementing the Simulation object so that I got a black frame. Then I continued by doing classes Boid and Boids. First I just implemented one boid and tried to draw it to the screen. After that I started implementing the movement, Vector2D class, etc. After the Vector2D class was ready I made test class for Vector2D methods. Lastly when program was otherwise ready, I made FileManager class and modified other classes so that the initial situation is read from a text file.
I think that the order that I implemented different aspects, went according to the plan. I just had to add more classes than I had planned. Time schedule in my plan was pretty accurate, which was 30 hours for the whole project.
I learned a lot about doing a program with multiple classes and also learned a little bit about using swing.

12. Final evaluation

Doing the project was very fun and I learned a lot about implementing a little bit bigger project. I think my program is pretty good and it has all the requirements and little bit more (changing weights of the movement rules in real time). I don't think I necessarily have any shortcomings but my program is not perfect. Some improvements for the program could be for example improving graphical aspect of the program (this is somewhat matter of opinion), there could be random movement for lonely boid. You also could limit boids vision only to front (now it has 360 degrees vision). I read that this would make the boids move in a triangular formation like birds in a flock. Changing this program into 3D simulation could also be cool.

Eventually I made all the thing I wanted to make. I think that the class structure is pretty good and it is easy to make improvement or add new things to the program.

13. References

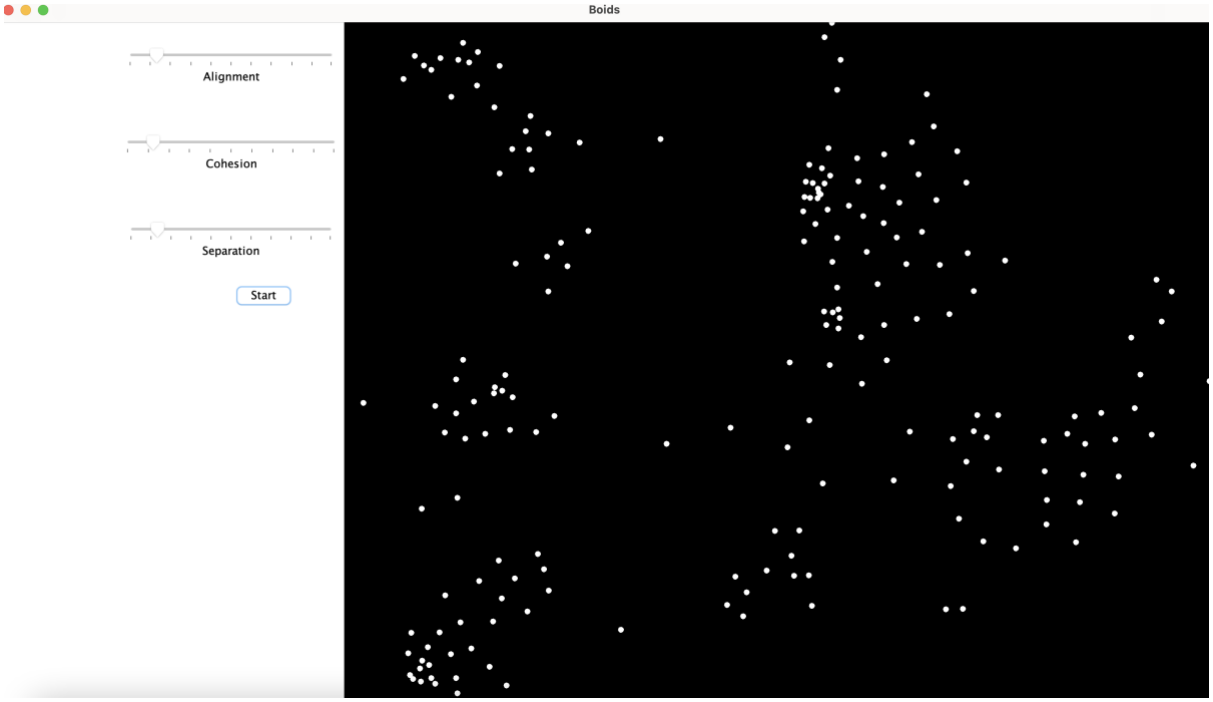
https://www.youtube.com/watch?v=LltugBg4dtk&list=LL&index=2&t=316s

https://www.youtube.com/watch?v=mhjuuHl6qHM&list=LL&index=5

http://www.red3d.com/cwr/steer/gdc99/

https://www.scala-lang.org/api/2.12.9/scala/collection/mutable/Buffer.html

Screenshot of the program before starting the simulation



After pressing “start”