Project Plan

Casper Tillander, 907116, Bioinformation technology, 2nd year, 23.2.2023

General description and difficulty level

The project is a simulation in which individual characters move in an abstract space. Each individual in the simulation is either susceptible to the disease, spreading it, recovered from it, or has been removed from the simulation (deceased).

Susceptible individuals catch the disease with a certain probability when they have been nearby a person with the disease. After catching the disease, a susceptible individual turns into a spreader, and after a certain time, the spreader either recovers or is deceased. Recovered and deceased individuals can no longer catch the disease.

The simulation is set up with a certain number of characters, a certain number of spreaders, and other parameters (e.g., the likelihood of catching the disease, the period of time it takes to turn into a spreader, the time spent being a spreader, and the mortality rate of the disease). Depending on the difficulty, the simulation's characters move around according to a particular logic. The simulation ends when everyone in the simulation has either caught the disease or when there are no spreaders left.

Robotworld from the exercises will be used as the base for this project. The goal is to have a more complex simulation where there could be several different character types. The disease spreads only from close contact. The user of the simulation will be able to affect the disease in the simulation by manipulating the parameters of the simulation.

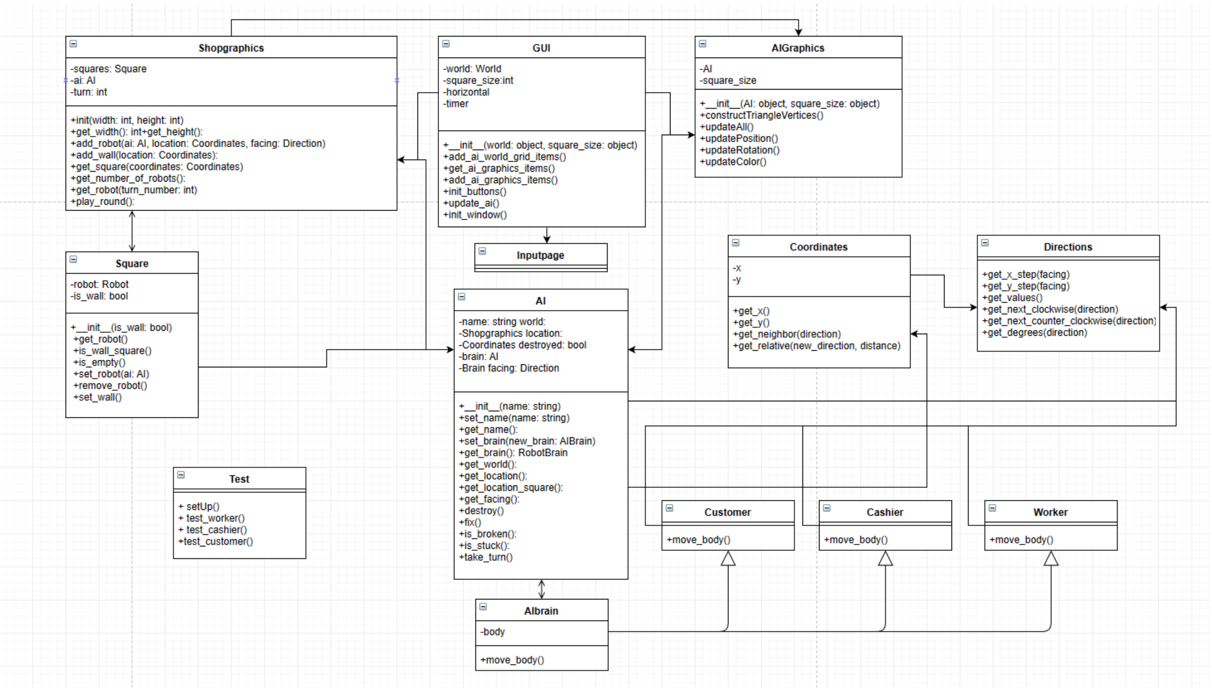
The goal is to make the project according to the hard difficulty standards. I would like to have the user first choose the standards, then run the simulation. The unique feature I would like to implement in my program is that it works as a grocery store. The goal is to make the simulation as realistic as possible and have the AI behave as much as humans as possible. The humans in the simulation can be customers, cashiers, or workers (working amongst the customers in the shop).

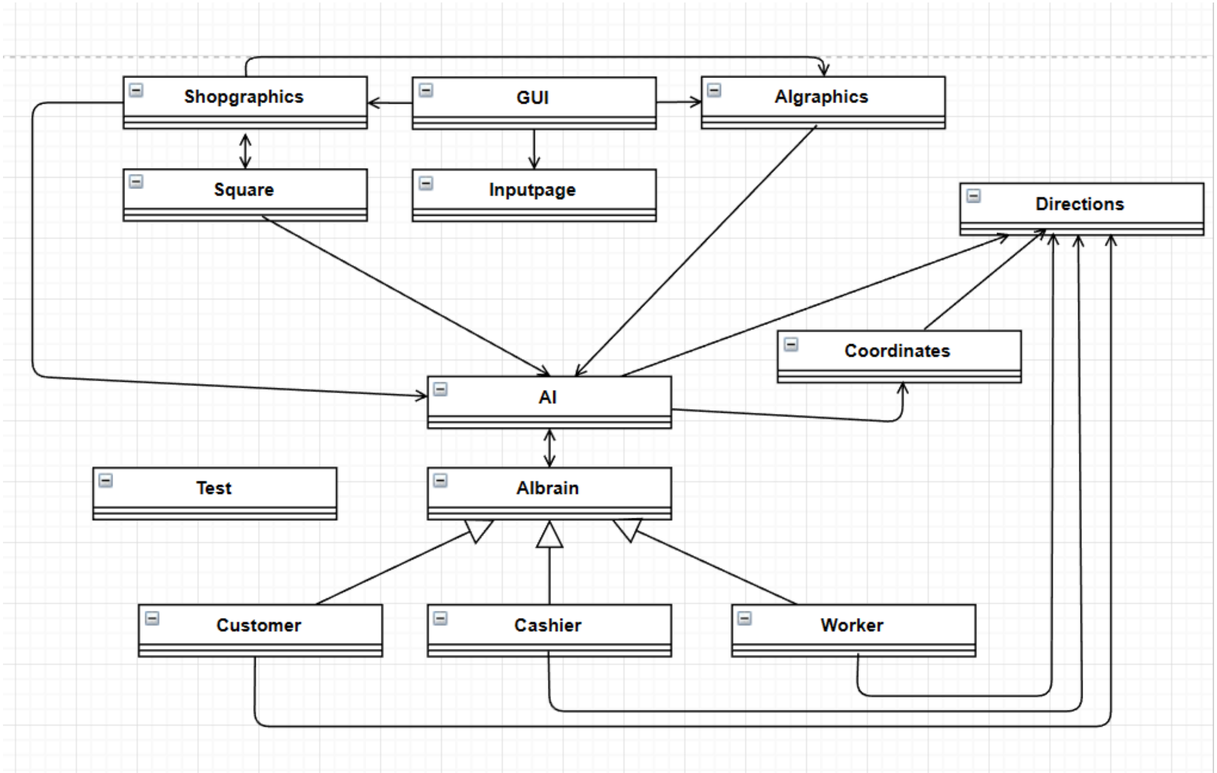
Use case description and draft of the user interface

The goal is to have a graphic user interface where the user can input the different parameters. This will be done on the starting page (called input page in the UML diagram). The user will then be able to run the simulation with the specified parameters. This will be done in a different window.

Program's structure plan

The program's class structure is shown in the UML diagram below. The graphical user interface is used for building the graphics in the program. The building of the more specific graphics will be done in separate classes: Shopgraphics, AIgraphics, and Inputpage. The shop will be a square grid where the different AIs can move, and a different class will be used for the squares. The AI class will be used for the general description of the AIs, and the AIbrain will be used for the movement of the AIs. The AIs are then divided into Customers, Cashiers, and Workers. These all behave differently and will therefore be described by different classes. The class coordinates and directions will be used to describe the different AIs movements and keep track of their position. The base for this structure plan is the program RobotWorld.





Data structures

Lists and strings will be used in the project. Arrays, dictionaries, and tuples might be used.

Files and file formats

This program doesn't need to handle any files.

Algorithms

At this point, I don't know that I will need any algorithms. I don't have any experience using them but

if I come across a problem that requires them I will find out how to implement them. For loops will be used in many places in the code to loop through for example the squares. Algorithms could be used to make the loops more efficient.

Testing plan

The program will need to handle different parameters that the user inputs. This means that the program should run flawlessly within the parameter ranges. The program must raise errors if the user inputs parameters that are not within the allowed range, or if the user inputs another data type than the one asked for (for example a string instead of an integer).

There needs to be tests that check these different scenarios and make sure the program does not crash in case of faulty input. This will be done by examining the places in the code where the user's input can make the code crash.

Libraries and other tools

PyQt6 will be used for the graphical interface.

Schedule

Week 8

Project plan deadline 24.2.2023.

Week 9

Guidance meeting 27.2.2023 11.00-11.15

Finnish at least the PyQt6-introduction part of the exercises. (Estimated time: 10h)

Learn about the features needed for the project's graphical part (Estimated time: 5h)

Week 10

Finnish all the exercises in this course (Estimated time: 10h)

Build the basic simulation with RobotWorld as the base, improve the simulation and play around with it, change the AI, and so on.

(Estimated time: 10h)

Week 11

Build the input page where the user can change the parameters and launch the simulation. (Estimated time: 10h)

Week 12

Deadline 1, at least some code in Git. 24.3.2023.

My goal is to have the simulation pretty much finished by this time since I have a lot of other courses that increase in difficulty and time consumption. (Estimated time: 10h)

Week 13

Start working on the tests and try figuring out how to get 100% coverage. The simulation should work if the parameters are within their allowed intervals and it should not crash, ever. (Estimated time: 10h)

Week 14

Deadline 2, at least some tests in Git.

The goal is to have all the code done by this week, including the tests. (Estimated time: 5h)

Week 15

Start working on the project's documentation. (Estimated time: 5h)

Week 16

Finalize the project, do a last check of everything, and finish the project's documentation. Make changes if necessary. (Estimated time: 5h)

Week 19

Final deadline for the project. 12.5.2023 14.00

Literature references and links

No sources have been used in this project so far, except for the course A+-page. These are some sources (links) that I plan on using for getting a better understanding of the graphical part of the project.

1. https://www.khanacademy.org/computing/ap-computer-science-principles/x2d2f703b37b450a3:simulations/x2d2f703b37b450a3:creating-simulations/a/programming-simple-simulations
2. https://www.pythonguis.com/pyqt6-tutorial/
3. https://wiki.python.org/moin/PyQt/Tutorials