26.4.2020

Ohjelmointistudio 2

Tower defense project

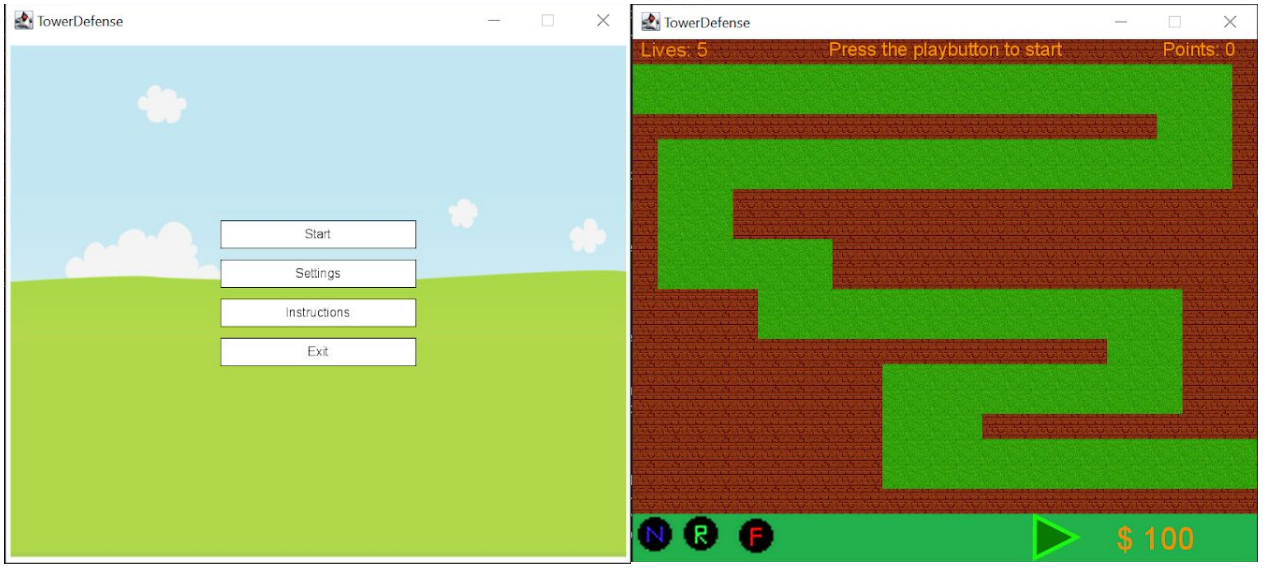
Anton Sulkava
653897
Automaatio- ja informaatioteknologia 2017

General description

The project is a real time tower defense game. In the game enemies try to reach a target and your goal is to stop them. The route is usually predetermined and you defend the target by using towers next to the path to shoot the enemies. The enemies come in rounds and the rounds get harder eventually. you get money from killing enemies and you will lose the game if you run out of lives.

User interface

The program is started by running the towerdefense.scala file as it has the main object for the game. Once the file has been ran, you will be greeted with a main menu where you can see some instructions, go to the settings or start the game.



On the left is the main menu and on the right is the actual game once you play it.

In the setting you can change the difficulty or change the map. Once you are in the actual game you can buy towers by left clicking the icons on the bottom left and you can reset your mouse and UI by right clicking. The play button to the bottom right starts the next round. You can also see your current lives, points and money.

Program structure

The program is divided into 5 different packages. Enemies, towers, game states, interface and the main package. The GameStates package includes all the different states that control the program. They are used to draw, update, start and end the different states when needed. The enemies and towers packages host the different classes for all of the towers and enemies. The interface package is used to visualize different button in the game , the information boxes and the loading of all the images. Tha main package includes mouse movement, the Pos class for all entities, the player class, enemy spawner class and the loading of the maps. It also includes the settings page and the main runner file for the program.

Algorithms

There really aren't any complex algorithms in the game. The enemy pathing is taken care by a x-y point system in the map files. The different path points are written in the end of every map file and the map loader gathers the path from the file. The hit registration is decided just by the distance of a projectile to the enemy.

Data structures

Almost everything is stored in different buffers and mutable maps. I decided to go this route as it offer easy access to the data and easy configurability.

Files and Internet access

The images for the different enemies and towers are stored in the imageFiles folder and all of them are png files and very small in memory size. The map files are .txt files stored in the mapFiles folder.

Testing

I did all of the testing in the actual program as it was easier to visualize mistakes in it.

Known bugs and missing features

I think I could have added some more features such as abilities and more tower and enemy types. However some bugs I know of are the round complete text not showing and some hit registration issues with the projectiles. I didn't have time to create sprites for the skeleton and zombie enemies, so they are just different colored squares.

3 best sides and 3 weaknesses

+ :

The game seemed to run really smoothly on my computer and it didn't crash.

The game has many possibilities for new expansions, such as maps, towers, enemies and the difficulty could affect other variables as well.

I think the game was pretty easy to use and work around, everything was really clear and the player gets information about everything.

- :

The pathfinding of the enemies is a bit clunky using the point system, but unfortunately I didn't have time to implement a full pathfinder algorithm.

I think the visuals could have been better as I didn't really pay that much attention to the different sprites and such.

The creation of maps is a bit manual and could have been done somehow differently, as now you have to manually add a text file to the resources and write all the path points by yourself.

Deviations from the plan, realized process and schedule

Mostly my project structure followed the technical plan, as I had the different packages already thought out, but I had to make a few additional classes for quality of life. My time management didn't go according to plan at all. I did most of the work in the last 2 weeks as I had other courses pressuring me with deadlines and exams earlier. The order of my progress was really random as i had to learn how to use processing core instead of swing, as I couldn't really get swing to function properly. However now I learnt how processing core should be used for the interface.

Final evaluation

I am mostly satisfied with the final product, as I learned a new library with using processing and the game runs well, which is the main point in my opinion. However I would have wished I had more time to work on this project to add more content and make the visuals better. I was not pleased with my time management and I believe the program can be easily optimized and developed further. So all in all I am satisfied with the result.

References

https://processing.org/
https://plus.cs.aalto.fi/studio_2/k2020/

Appendixes