Project Diary

Program Design and Data Structures (1DL201)

William Berger

March 2021

UPPSALA
UNIVERSITET

Dates- and brief summary

2021-02-15

Time: 3 hours

As mentioned in our shared project diary we had a meeting with our supervisor and he gave us the green light on developing a messaging application. Our first issue was setting up a server to be able to communicate through two different devices. I spent a few hours researching this online. Looked at different websites and online books through the university's library.

Found something called Network.Simple.TCP that seemed suitable for our project: `s://hackage.haskell.org/package/network-simple-0.4.5/docs/Network-Simple-TCP.html` . This TCP module should let us both set up the server and let a client connect.

2021-02-16

Time: 2 hours

By now we had decided to change our project to battle ship instead. To represent the board in the game a 2d-array seemed suitable. Found the module Data.Array:
`https://hackage.haskell.org/package/array-0.5.4.0/docs/Data-Array.html`

Spent some time to research the functions inside of this module, how to create a 1d-array, and how to read and manipulate the data.

2021-02-17

Time: 4 hours

By now we had collectively created the 2d-array to represent our play board, and a few functions. I spent a few hours to further implement a few game logic functions that do the following:

- Check if given coordinates are inside of the play board

- See if cell at given coordinates is checked or not
- Place a ship of given size and direction, on given coordinate
- Check if ship placement is valid
- Check that the ship the user wants to place does not collide with any ship already placed
- Get the size of the play board
- Check if all ships on play board are checked, aka win
- Convert the 2d-array of cells to a list of cells
- Convert the 2d-array of cells to a list the cell's states

2021-02-19

Time: 2 hours

I've had problems trying to get the modules Graphics.Gloss, and System.Random to work but finally managed to get both working today. The random module is necessary for our AI, that will utilize this module to initially decide where to shoot.

Did some research on the link below on the Gloss module to get a better understanding on how it can be implemented in our project:
`https://mmhaskell.com/blog/2019/3/25/making-a-glossy-game-part-1`
Got an understanding of how to create a window and get input by a user, through either the keyboard or mouse.
`https://hackage.haskell.org/package/gloss-1.8.1.2/docs/Graphics-Gloss-Interface-html`

2021-02-20

Time: 7 hours

Me and Jonas worked the majority of the time together. We looked into how to render our 2d-arrays in Gloss, and can now place crosses on the grid with mouse clicks. We had a lot of trouble trying to get the crosses in the

correct cell. It was not translated correctly. I later spent some time alone and used the Debug.Trace module to see what was going on. I found that the cell coordinates created by the mouse click were offset by -5 in row, and -10 in column. There is most likely something wrong when we translate the coordinates we get from the mouse, but I used dodging and temporarily solved the issued to be able to continue working on other functions, by just adding 5 and 10 manually. We'll look into this matter tomorrow.

2021-02-21

Time: 5 hours

Me and Jonas looked into the issue we got with the mouse coordinates, and were successful in solving it. The problem is that the origin of the mouse coordinates are the center of the window, and this caused all indexes left of the center to become negative. We solved this by changing the formula in the function that converts mouse coordinates to cell coordinates.

We added different ways of rendering a hit, miss and ship on the window. Hit is showed by a red cross, miss by a white cross, and ship by a grey rectangle. Then I spent some time implementing, the function that places a ship, into Gloss. We can now place ships on our board with the mouse.

2021-02-22

Time: 5 hours

We changed our ship placing, now the user places a ship with the keyboard. Added the following functions:

- Function that moves a ship, on the board, with the arrow keys
- Function that rotates the ship by clicking 'r'
- Function that confirms the ship placement and translates this ship into the 2d-array

Combined this with a few functions I made before, so now user can only confirm ship placement as long as it does not collide with a ship already placed, and rotation is only allowed as long as the ship is still within the board after rotating.

2021-02-23

Time: 5 hours

Added a function that lets the AI place ships on its board. Added at the first valid placement it could find. Added new game rules and function that checks it: ship can't be placed right next to a ship.

2021-02-24

Time: 4 hours

Me and Jonas used the new AI place function we created yesterday to generate 1000 random boards and add them to our game data type. This allows for the game to restart after someone wins (only 1000 restarts)

2021-02-25

Time: 4 hours

Added text to our gloss window. User can now see current game stage, stats, current round and instructions. Added radar animation below both player board.

2021-02-26

Time: 5 hours

Modified the already existing explosion animation. Before it was triggered by every left mouse button click, but wrote a function that checks if mouse click is inside of a specific board. So now it is only triggered if I click somewhere on the enemy's board, and only shows animation if that cell was not already checked. Blue animation for miss, red for hit.

2021-03-01

Time: 8 hours

Added function specifications and added some examples to some already existing specifications. Wrote about the placement rules in the project report, explained how they worked and added a picture to further explain it. Starting writing about the logic module in the project report. Wrote some about the game data type.

2021-03-02

Time: 10 hours

Continued working on the logic module in the project report. Described the eventHandler, and all the functions that the different event triggers. I wrote a section about how placing a ship works, describing the functions: *moveShip*, *rotateShip*, *confirmShip* and all the functions that they use.

2021-03-03

Time: 6 hours

Finished up the logic module in the report, added picture of the code and explanation to the *playerShoot* function. Wrote about the main module, explained a bit how it works and what it does. Explained further how the game data type works and what the fields inside of it represents. Explained how someone runs our program.minimum

Explained initGame. Read through the whole report to make sure the language and structure is good.

Total time

Time spent working by myself on the project: 70 hours
Time spent working together on the project: 32,5 hours
Total time I spent on the project: 102,5 hours