Project Diary

Program Design and Data Structures (1DL201)

Jonas Rosengren

March 2021

UPPSALA
UNIVERSITET

Contents

1 Introduction

This diary shows what I've been individually working on during the project.

Total individual time: 63,5 hours
Total time with group: 32,5 hours
Total time spent on project: 96 hours

2 Diary

Similar to the common project diary, this document states all occasions I have worked on the project - outside of a group meeting with all of us.

2.1 2021-02-03 - initialized git repo

Duration: 1 hour

- Initialize git repo
- First draft of group rules, grouprules.txt
- Task manager, tasks.txt

2.2 2021-02-12 - socket research

Duration: 1 hour

- Research Network.Socket
- email response

2.3 2021-02-12 - socket research

Duration: 1 hour

- Research Network.Socket

2.4 2021-02-12 - installing gloss

Duration: 2,5 hours

- Research about Gloss.
- getting gloss started on my machine. Took some time to get it working, as with other libraries in haskell..

2.5 2021-02-18 - gloss research

Duration: 2,5 hours

- Played with gloss for better understanding.
- These resources helped a lot to understand the different gloss arguments and functions:
  - https://mmhaskell.com/blog/2019/3/25/making-a-glossy-game-part-1
  - https://hackage.haskell.org/package/gloss-examples-1.13.0.3/src/
  - https://hackage.haskell.org/package/gloss-game-0.3.3.0/docs/Graphics-Gloss-Game.html

2.6 2021-02-19 - initializing src-folder

Duration: 3,5 hours

- Made a new src-folder for a top-down approach of the project. Dividing different aspects of the gloss play-function into different modules.
- started to write on the different modules to be able to run main

- added a few datatypes. Working on how to make two separate boards showing. The goal is to make two boards: one showing where the user is shooting, another one for showing where the AI has shot. It's necessary to have two different game states: one for placing ships, the other for shooting.

2.7 2021-02-20 - two grids on gloss-window

Duration: 6,5 hours

- Worked on a solution of making grids dependent on boardPos. Where boardPos is the ((x1,y1),(x2,y2)) of the board, making a rectangular shape. This will make it possible to place multiple grids on the board. Only worked on the graphics, thus an implementation of the actual grid in the logics module will still be needed.
- Worked on a solution with Berger to make the game arrays rendered in gloss. Accomplished the possibility to use the mouse to click on cells on one board to get it pop up, although it popped out in wrong cells along the y-axis. Probably has something to do with the different translations of the board, and how mouseposition is turned into a cellcoord.
- We named new types to differentiate between cellCoord and screenCoord. Also reversing so that type CellCoord = (col, row) instead of (row, col), to be more conform with x and y.

2.8 2021-02-21 - shooting ships

Duration: 2,5 hours

- Me and Berger made the boards dynamic with a border in between. Added functionality to see more than just one of the cell-types (Ship, hits, miss) shown in Figure 1.
- Tried to work on a ship-placing functionality. Where it is possible with key inputs to control the placings of battleships on the board.

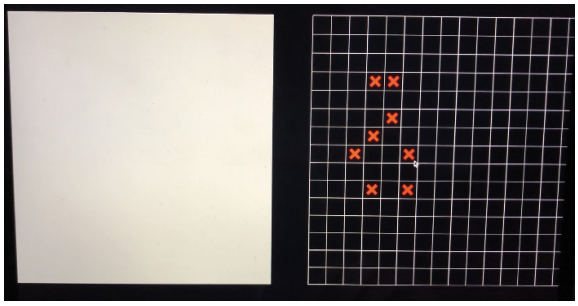

Figure 1: 21feb

2.9 2021-02-22 - Placing User

Duration: 4 hours

- worked on a convenient way to place ship for the user
- Placing Ships scheme:
  - If Stage == Placing User
  - ArrowKeys, R -> moveShip
  - Enter –> place current ship in user board array.

2.10 2021-02-23 - solved Placing User-bugs and worked on PlacingAI

Duration: 7 hours
When I woke up Berger had finished a way to use keys to move a shipPicture and place it with enter shown in Figure 2.

- worked on solving bugs in Berger's solution.
  - Fixed error when hitting another key than the valid inputs.
  - making sure the Key inputs only are valid in PlacingStage.
  - Working on solving a "head on empty list"-bug when the last ship has been placed. Also to change gameStage when the last ship has been placed. The issue lied in the rendering where the base-case when all ships where placed was missing. I added a function making the picture blank if the shipsUser-list was empty.

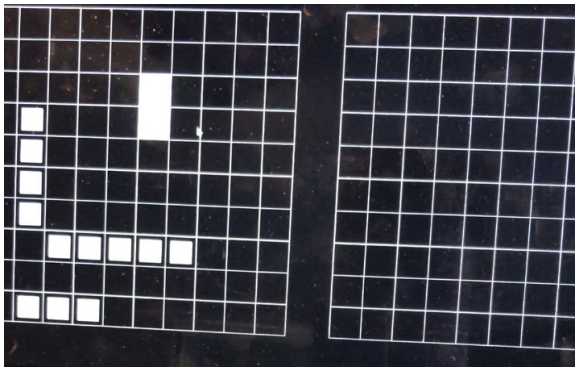

Figure 2: 23feb

- since its now possible for the User to place and shoot, its necessary to make it possible to do the same for the AI. Thus working on a way to place ship for the AI. Worked on this algorithm together with Berger:
  - ships = list of ships
  - Step 1: Find valid placements for head(ships) on board, pick random of valid placements. Put ship on board, remove from ships.
  - Step 2: Repeat step 1 until ships is empty.
  - –> randomly generated board.
- Me and Berger implemented the algorithm, but without the randomness. Still need to implement the randomly picked positions.
- Researched randomness in Haskell.
  - inspiration for random generation, https://github.com/benl23x5/gloss/blob/master/gloss-examples/picture/Gravity/Main.hs
  - problem with installing System.Random, used school computer remotely. Couldn't use Gloss because of the remote.
  - I solved the issue by sending a source of randomness in to the pure functions, and keeping track of the newly generated Gen with a tuple. like so:
    - Func :: gen –> input –> (output, newGen)

2.11 2021-02-24 - Random ShootingAI, replayable, animation research

Duration: 7 hours

- Implemented shuffle function to make randomized shooting AI. Passed the seed down from main and added it to aiShoot
- Added specifications to placingAI
- Made an ugly version to keep the game boing by initializing a ton of boards in the beginning and sending them down to the game-type with initGame. Made it possible for multiple random ai shootings, instead of only one each time running the program.
- Worked on a better shooting algorithm for ai – didn't result in new code
- Made sure to update global gen for every replayed game
- researched animation capabilities in gloss. Created an exlosion around mouseclick, and a constant pulsation. Although outside of project files.

2.12 2021-02-25 - animation

Duration: 8 hours

- Implemented the explosion animation when shooting in the src-files. Had to add to logic, game and rendering modules.
- implemented the pulsation in the background. Illustration of the application in Figure 3. The radiuses of the circles are increasing with time, until they hit the width of the screen. Then they initialize again.
- Berger changed the pulsation to a spinning radar. I implemented a way to have a gradient on the arc –> fadedArc
- Worked on a better AI. The idea is that the AI-should scan through the board and notice sunken ships. "miss, hit, hit, miss" means there's

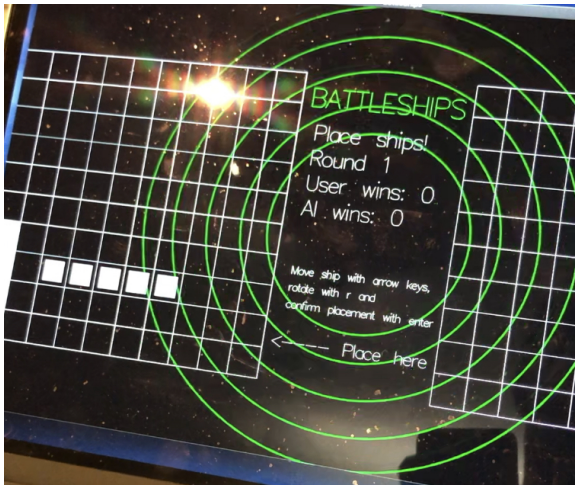

Figure 3: 25feb

a sunken 2-sized ship. If you scan through the board after valid placements of all the ships, left to sink. With these placements (coord, direction), its possible to know which coords can contain a ship, and also strategically grade the probability of success depending on how many ways its possible to place a ship on a specific coordinate. Didn't result in a new code.

2.13 2021-02-26 - dynamic replay, descriptions to logic module

Duration: 4 hours

- Fixed so its possible to replay infinite times. Instead of generating 1000 boards in the beginning I moved the generating script to Logics, where it can be used to generate a new board when the user replays.
- Added descriptions to most of the user logic functions, although not examples.

2.14 2021-02-27 - logic examples and descriptions

Duration: 1 hour

- Added a few examples and descriptions to logic functions

2.15 2021-02-28 - testcases to mainfuncs in logic

Duration: 2 hour

- Wrote 8 testcases and helping function for tests

2.16 2021-03-01 - animation and shuffle descriptions

Duration: 2 hours

- Worked on animation descriptions
- Shuffle descriptions

2.17 2021-03-02 - graphics section in report, rendering descriptions

Duration: 4,5 hours

- Game screenshots Writing graphics section of paper
- Rendering descriptions, all but text to user

2.18 2021-03-03 - graphics section in report, rendering descriptions

Duration: 3 hours

- Finishing rendering descriptions
- sketching flowchart, game as an individual object that gets updated to user
- Refining graphics section of paper with pictures etc

2.19 2021-03-04 - diary refinement

Duration: 2,5 hours

- worked on writing this diary, refining it to a final product.