

Design Report

Team Jertep The members of team Jertep are:

- Eiríkur Hákon Friðriksson
- Elías Ingi Elíasson
- Jón Sigurðsson
- Pétur Stefánsson
- Rákel Rós Geisdóttir
- Tumi Guðmundsson

Introduction This is a design report for an assignment in a Software Engineering course at The University of Reykjavík. The assignment is to implement a TickTackToe game using Test driven development. We used Github as our source control to store, share and update every part of the assignment since there were six of us working together. The implementation of the game was coded in Eclipse and in the command prompt (windows) / terminal (mac) using the java programming language. We used Gradle to as our build tool to compile the code and to run unit tests. Lastly we used Travis for automated continuous integration server.

Design The initial plan is to create a functional console game but with the option to implement a web app if we had the time to do so. We ended up creating a gui for the game which can be found in the src folder.

Everyone in the team knows how the game works so it wasn't hard to figure out the basic logic and decide how to design it. At first there would only be one class but in the finished product each functionality will be stored in separate class eg. the game, the player and the board. The board is a 3x3 square stored in two-dimensional array and is initialized as followed:

```
1 2 3
4 5 6
7 8 9
```

The players are two , X and O. The player that starts is X and if he inputs 5 then the updated board will be:

```
1 2 3
4 X 6
7 8 9
```

if there is X or O in the array where the player decides to input his symbol, the game returns a error message, aswell as having any illegal inputs outside of the game board.

Gui App We decided that the console app was a bit fiddly so we created a Gui application that is click activated but uses the functions from the original console app, instead of typing 1 - 9 you just have to click the buttons in the window that pops up.

Implementation and testing We will split the functions between ourselves and each team member is responsible to write a testing code to go along with the game code. After each implementation, the member will add, commit and push his or her code to the repository. Travis is connected to our repository and will be built there. If the build on Travis is without errors we continue, else we fix the code. We will create a Heroku web app when the console game is ready.

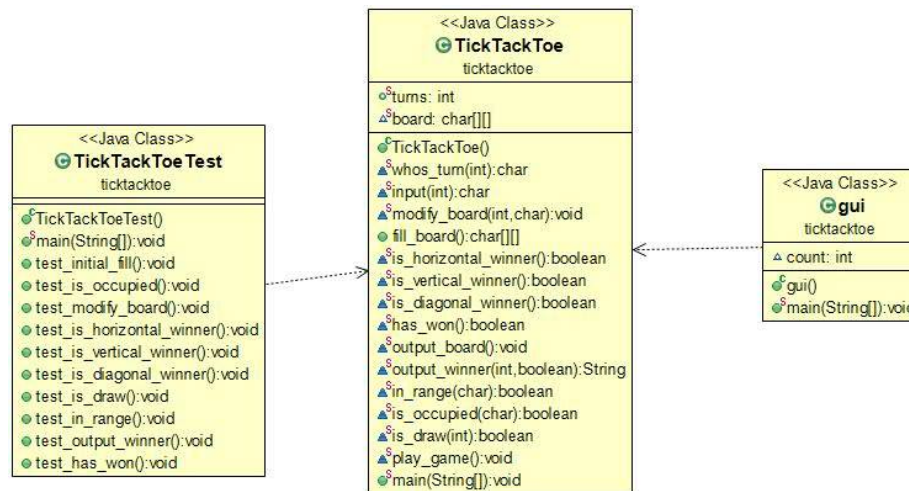


Figure 1: alt tag

Class diagram

Code coverage & inspection We will use a plugin called ReSharper to go over our code and to verify that the coding rules we had set are being followed.

The delivery The plan is to deliver the project in a web based app in Heroku, so that the user can play the game in a browser. It will be accesible here on github and can be played through a link we provide in our repository.