HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

T-303-HUGB, HUGBÚNAÐARFRÆÐI

# Tic-Tac-Toe
# Temp Name, Please Ignore

*Student:*
Baldur Már Pétursson
Daníel Freyr Sigurðsson
Ívar Markússon
Jón Helgi Kjartansson
John Russell Hutton
Lúðvík Kemp Guðmundsson
Njáll Hilmar Hilmarsson
Örlygur Ólafsson

*Teacher:*
Hannes Pétursson

30. október 2016

# Introduction

For our final assignment of this course we where tasked to develop a game of Tic-Tac-Toe using the programming language Java as well as implementing several online services that aid in software development.
The group came together early on and setup a repository or repo on the website GitHub.com, allowing all members to readily access and distribute the code for all group members. Not only that but GitHub also has built-in features, allowing other sites to directly access and test our code both before and after submission of new bits. A service called Travis CI compiled and ran all test that we had written before submitting them to a final version on GitHub. This pushed the group to only distribute code that had not errors or failed any tests, making the process of developing our game that much easier by supplying a up to date overview of what had been done and by who.

# Design Description

For the design of our program the group agreed to keep it simple yet effective, using the three-tier layer setup of backend, service and interface layers. This allowed the group to work on different layers at the same time while still ensuring that every part communicated with the others in their desired way.
When the initial programming started the group had a short meeting of how everything would be setup. There would be three classes, player, board and game classes each keeping track of a certain part of the game while the game class decides if the game is won or over and which player has the next move. While these classes are not very complex, each of them have their own test-classes. This is in accordance to the Test Driven Development process of writing a test for a specific function then implementing the function itself.

# Evaluation

The group came to the conclusion that the program would be implemented using the TicTacToe class as the main engine. The ConsoleUI class would be the user interface for the console version of the game and the Game class for the web based version of the game. The player class is an entity class that all layers can access.

## Overview

The game itself is pretty simple and our version is no different. We decided to have players enter their names before playing so that records could be kept for wins and losses. A player can have their own unique name that the database keeps track of their wins and losses.
Our version can both be played in console and on the web at http://hugb.deadbyte.is/ where the environment is a lot more user friendly. Instead of entering numbers to a console window and seeing the board as an array of characters the web version is a simple mouse click on a GUI interface board which is easier to understand as well as giving the player more direct visual feedback.

## Testing and Results

The tests for this program where written at the same time as the program itself, when we where for example writing the play class the first step would be to write a test case for the player class to ensure that development is derived from development. Having to write test cases first then implement them not only ensures that the program has sufficient amount of test before release but also ensures that the program in its entirety works as it is designed to.
At first all test results where failed, however after they where written the function that failed was then added or the program modified to preform a task. After merging all branches together gave us tests for each class to ensure all is as it was designed to do.

# Final Words

This assignment proved to be an excellent way of trying a new development method as well as implementing a wide range of services for software development. After the initial trouble that was putting all these pieces together and getting them to work with each other, this became a very easy way for our group to improve and share the code, allowing all members to constantly be on the same page.

Being that our group did consist of eight team members, getting everyone on the same page took a little time. However after that it proved to be very easy to implement and test new code and share it, while making sure that no one shared any code that was not runnable. This meant that every time a member would start coding again, they could be sure that the code they had was not only the newest version but also ran without any problems.

Now we felt that more than half of this assignment was to implement a wide range of services that companies use in the real world, giving us a chance to hone our skills for software development and do mistakes on a small scale with plenty of assistance readily available. While this wasn't a very large or hard program to develop, it did give us experience in working in a group to achieve a goal.