CS 5035 (Fall 2016)

### Project 7. Making our own types and typeclasses.

Based on chapter [8 of LYH](http://learnyouahaskell.com/making-our-own-types-and-typeclasses). [Videos](https://sites.google.com/a/lclark.edu/drake/courses/pls/lesson-7-haskell-making-our-own-types-and-typeclasses). Two weeks are allocated for this project.

Here is a program that plays Tic-Tac-Toe. It consists of two modules: [Tic-Tac-Toe](https://drive.google.com/open?id=0B-I58s-_d3o5N1FFUEYxLUY5TWM). plays the game. [History](https://drive.google.com/file/d/0B-I58s-_d3o5TjJjV1VUM1BnTjQ/view?usp=sharing) keeps track of the game sequence. Download it, load it into GHCi, play a few games, and follow along in the code.

Play a game by using the following commands.

> start -- Start a game by placing an X at an optimal location.

> next it -- Place an X or an O (depending on whose turn it is) optimally.

> play r c it -- Make the next move at row r, col c.

> startAt r c -- Start a game by placing an X at row r, col c.

After start, use either next or play to play the game.

The game takes advantage of the fact that the Haskell REPL uses it to refer to the most recent value. So the it in the commands above refer to the result of the previous play. They must follow immediately after a previous move in the game.

Play a game using the history mechanism use: hStart, hNext, hPlay, and hStartAt.

You can’t mix these two sets of commands. When using the first it will be a Game. When using the h set, it will be a History object.

Your responsibility is to read the code and explain it to me.

#### **The first explanation—even if incomplete—is due by Oct 24**.

I recommend that you concentrate on the overall structure of the program and explain how that works. Focus initially on Games using [Tic-Tac-Toe](https://drive.google.com/open?id=0B-I58s-_d3o5N1FFUEYxLUY5TWM) rather than on [History](https://drive.google.com/file/d/0B-I58s-_d3o5TjJjV1VUM1BnTjQ/view?usp=sharing). Understand how start, next, and play work.

#### **The second explanation is due by Oct 31.**

The second explanation should extend your explanation of the structure of the program and explain the details as well. You should also explain how the History component works.