

**Yafit Beyene**

Student ID: 2213990

## **CS141 Coursework 2: Report**

### **Contents**

<b>1</b>	<b>Report</b>	<b>2</b>
1.1	Functions . . . . .	2
1.2	How to run the program . . . . .	2
1.3	To Bean or not to Bean . . . . .	2
1.4	Notable Errors I Encountered . . . . .	2
<b>2</b>	<b>References and Resources used</b>	<b>3</b>

# 1 Report

## 1.1 Functions

Regarding the functions I used and the techniques used are covered in detail in the justify section of my code

## 1.2 How to run the program

To run the program all you have to do to run the program is "stack run filepath". For convenience sake I put one bgn file in the same directory as main.hs so that I could just do "stack run 1.bgn" every time rather than having to type out the longer command stack run bgn-files/valid/2.bgn.

## 1.3 To Bean or not to Bean

I initially had quite a lot of difficulty with this coursework. I was struggling to use the Parsing libraries and overwhelmed myself with trying to learn megaparsec or parsec and all these different libraries so I decided to just keep it simple and use base to parse my function.

I found that this was more simple for me and I didn't have to tamper or worry about the dependencies so much either. I only found this out after I had already spent an inordinate amount of time trying megaparsec and was feeling quite defeated when things didn't work. Removing my code and starting from scratch did a number on me and really set me back mentally. However, I don't regret my choice because I have a much better solution than I had before.

## 1.4 Notable Errors I Encountered

An error that I had for a while was that my parseMoves function was only considering as valid moves where something wasn't captured (so all moves without an x). This was difficult for me to notice so I ended up using some debugging statements (some of which I left in my code and reimplemented as a feature (I've mentioned which this is in my comments)). I ended up fixing this by using Data.List.Split, specifically splitOn, which allowed me to split it on the x and then produce valid coordinates. For a while I thought it was playMoves that caused the error and I would have kept doing so if it wasn't for me adding the putStrLn to keep track of what moves were being parsed and what was being considered.

Another issue I had encountered was that my old implementation was constantly saying no valid moves and returning an empty list. It was considering every move invalid because my parseMove function didn't have the mapping it had now. My old implementation had a parseRow and parseColumn. I did this to try and split it into the player types Blue and red since in the bgn file the first

column would be blue and the second would be red and the row would be the turn I'm on. i found that splitting it into 3 functions: `parseRow`, `parseColumn`, and `parseMove` was difficult to manage so I resorted to just using a mapping and the `mapMaybe` function instead

## 2 References and Resources used

I used this to help understand how to use `Data.List.Split`, specifically `splitOn` which was particularly useful in the first error I mentioned in 1.4 - <https://hackage.haskell.org/package/split-0.2.3.5/docs/Data-List-Split.html>

I also used Phillip Hagelocher's Haskell for Imperative Programming youtube videos to remind myself of how to use basic functions and to make sure I was doing them correctly. I found his lambda function video helpful because lambda functions is something I didn't use so often but they are nice and readable so I wanted to use them more.

Finally I also used this <https://markkarpov.com/tutorial/megaparsec.html> for when I initially tried to use `megaparsec` but ultimately switched to `base`. It wasn't particularly useful for my final solution because I had not used `megaparsec` at all but I thought I'd let you know anyways.