University of Sheffield, Department of Computer Science

COM2108: Functional Programming
2018 Assignment 3: Playing 8-Off Solitaire

This exercise counts for 50% of the assessment for the COM2108 module

1. Introduction

This assignment will give you experience in design and experimentation in Haskell.

You will build on assignment 2 and write a program to decide on the best move to make from a given position in 8-Off Solitaire. That program will then be used to play complete games and assess performance over a number of games.

Code is provided which will ouput 8-Off boards and 8-Off games to the terminal in a neat format.

You can build on your own code from assignment 2 or use PDG's code, which will be available after the assignment 2 cut off.

Skilled players win about 30% of 8-Off Solitaire games.

2. What You Must Do

1. Design, implement and test Haskell code to find **all the possible moves from a given position in 8-Off Solitaire, and choose which of these moves to make**. After each move you should also move any cards which can now be added to the Foundations by calling the **toFoundations** function from assignment 2.

   The function which finds the possible moves will be

   findMoves :: EOBOard -> [EOBoard]

   The top level function will be

   chooseMove :: EOBoard -> Maybe EOBoard

   Which normally returns **Just EOBoard** but returns **Nothing** if there is no legal move from the given position (i.e. **findMoves** returns the empty list & the game is lost). For a win, the final move will return a board with empty columns and reserve.

2. **Experiment with your move choice algorithm.** Guidance on the choice is given below. To do this you need to write

a. **eOGame** which will use **chooseMove** to play a game to completion. **eOGame** should return a score, which is the number of cards which have been moved to the foundations. For a win, the score will be 52
b. **eOExpt** which will play 100 games given an initial random seed. **eOExpt** will return the number of wins and the average score.

3. What makes a good move?

- It's good to move Aces to the Foundations if you can.
- It's good to move a King to a vacant column if you can.
- Try not to use up the Reserve. It's best to keep a minimum of 3 spare cells.
- Therefore, don't make a move from Column to Reserve unless you have to.
- If you do need to make a Column-to-Reserve move, then look for one which will allow you to make a move from Reserve to Column next time, thus recovering the reserve space.

4. Code supplied

Code is provided in a module **EOIO.hs** as follows

- **displayEOB :: EOBoard -> IO String** displays an 8-off board i.e. outputs a neat display of foundations, columns & reserve on the terminal:

```
EOBoard
Foundations  [(Ace,Clubs),(Two,Spades)]
Columns
        [(Nine,Diamonds),(Three,Clubs),(Five,Diamonds),(Eight,Diamonds),(Two,Clubs),(Eight,Hearts)]
        [(King,Diamonds),(Seven,Hearts),(Seven,Diamonds),(Three,Hearts),(Six,Spades),(Jack,Clubs)]
        [(Queen,Diamonds),(King,Hearts),(Ten,Clubs),(Ace,Hearts),(Nine,Clubs),(Four,Hearts)]
        [(Jack,Diamonds),(Four,Spades),(Queen,Spades),(Nine,Spades),(Ten,Diamonds),(Four,Diamonds)]
        [(Two,Hearts),(Five,Clubs),(Jack,Hearts),(Ten,Hearts),(Three,Spades),(King,Clubs)]
        [(Five,Hearts),(Two,Diamonds),(Five,Spades),(Seven,Clubs),(Eight,Clubs),(Six,Diamonds)]
        [(Six,Clubs),(Seven,Spades),(Queen,Clubs),(Six,Hearts),(Three,Diamonds)]
        [(Four,Clubs),(Eight,Spades),(King,Spades),(Queen,Hearts),(Ace,Diamonds)]

Reserve    [(Ten,Spades),(Nine,Hearts),(Jack,Spades)]
```

.

- **displayEOBList :: [EOBoard]-> IO String** displays a list of 8-off boards in the above format.
- **displayEOBGame :: EOBoard-> IO String** plays a game & displays it, given the starting board & repeatedly calling chooseMove.

To use **EOIO** you must use the same data structures for an **EOBoard** etc. Details are given in the code. Alternatively you can modify the EOIO functions for your own data structures.

Remember that you can't call IO functions from within a 'pure' function (see §22 of the notes).

5. Mark Scheme

Criterion	**% credit**
Is the design and associated notes a good explanation of what has been submitted?	10
Is the method (or methods) for choosing a move good?	15
Have the functions in the specification (findMove, chooseMove, eoGame, eoExpt) been completed?	20
Is the work well-coded?	20
Is the work well-commented?	10
Is the work well-tested?	10
Have experimental results been submitted and are they well-reported?	15

6. What to hand in

Hand in 2 documents:

1. Your commented code as a .hs file, ready to run
2. A report consisting of
   a. Your Design, in a diagram similar to that in the model answer for assignment 2
   b. Your test results. You should show that each function works correctly in each logically different case.
   c. A summary of your experimental results.

6. How to hand in

Hand in by MOLE

DEADLINE: Midnight Thursday 13th December (week 12)

Note that 10% of the credit is for your design.

No help with debugging without your design

When you're reporting results, or entering the competition, please give your performance as the mean and standard deviation of the number of wins and scores over 10 runs of 100 games. MeanStats.hs will compute these stats for you.

Xmas competition:

- Send me your average number of wins over 10 runs, each with a different random choice of 100 games.
- Prize for the best reported performance by January 2nd.