COMP 3258 Functional Programming Final Project

By – Siddharth Dev Tiwari

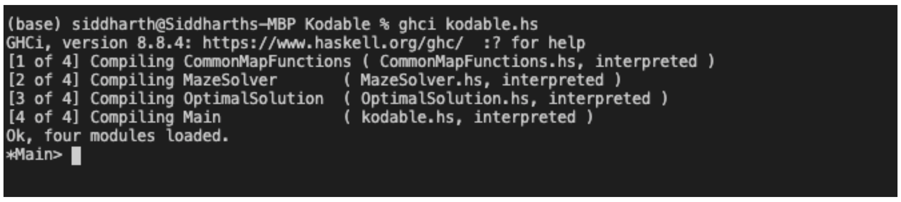
UID – 3035436791

Table of Contents

Building Kodable

To build the project follow the following instrucitons –

1. Download the Kodable folder
2. Open terminal and navigate to the downloaded Kodable folder which contains the Haskell files
3. Type `ghci kodable.hs` to build all the Haskell files.



```
(base) siddharth@Siddharths-MBP Kodable % ghci kodable.hs
GHCi, version 8.8.4: https://www.haskell.org/ghc/  :? for help
[1 of 4] Compiling CommonMapFunctions ( CommonMapFunctions.hs, interpreted )
[2 of 4] Compiling MazeSolver       ( MazeSolver.hs, interpreted )
[3 of 4] Compiling OptimalSolution  ( OptimalSolution.hs, interpreted )
[4 of 4] Compiling Main             ( kodable.hs, interpreted )
Ok, four modules loaded.
*Main>
```

Game Interface

After loading the map (explained on pg 5) you want to play Kodable on, the game interface is as below with 4 commands –

- ➔ Check – This command incorporates the logic in mazeSolver.hs and checks if the map is solvable, that if there exists a path from '@' to 't'.

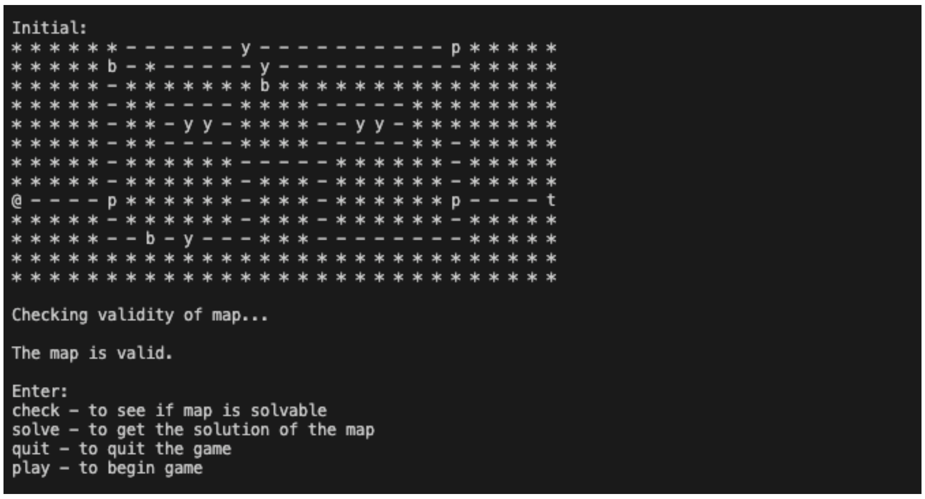
- ➔ Solve – This command incorporates the logic in optimalSolution.hs to return a solution of the given map along with the maximum possible compressed version of the solution. For eg, output of solve for the map below would be

```
The solution is:
Right Cond{p}{Up} Right Up Right Cond{p}{Down} Left
Cond{y}{Down} Up Left Down Left Down Right Up Right Down Right
Up Cond{p}{Right}

The most compressed solution is:
Right Cond{p}{Up} Function Cond{p}{Down} Left Cond{y}{Down} Up
Loop{2}{Left,Down} Function Down Right Up Cond{p}{Right}
```

- ➔ Quit – This quits the game

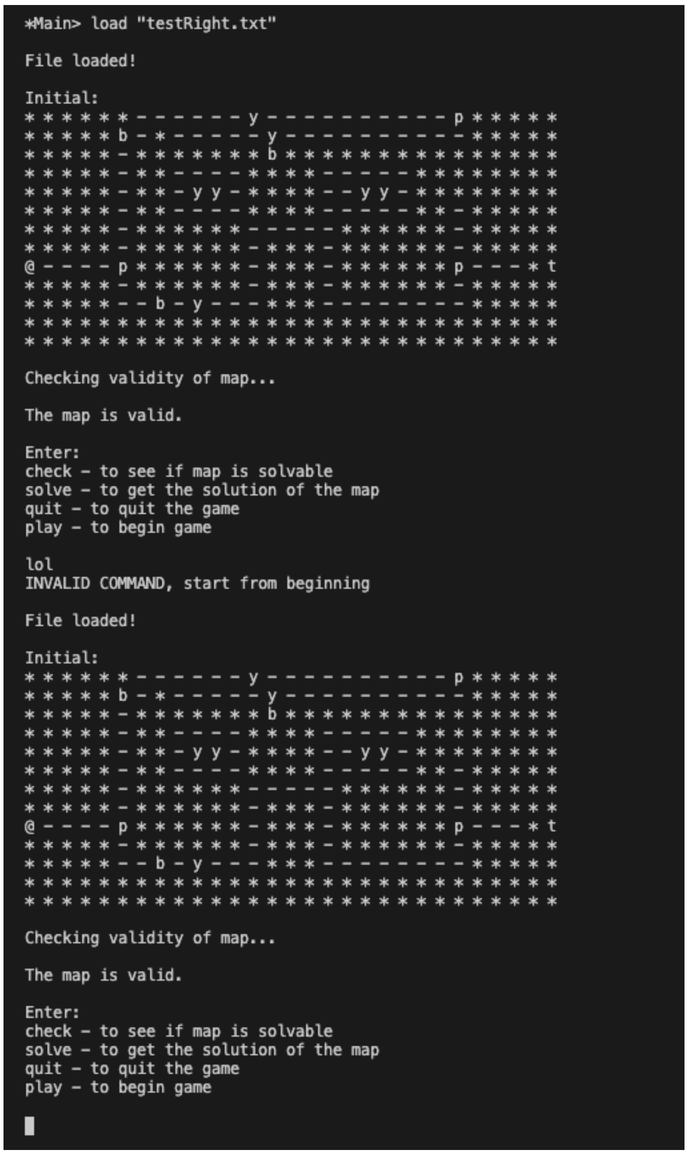
- ➔ Play – This command incorporates the logic of kodable.hs and is the main command to play the game.

  More information on these commands will be given in the following section



```
Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game
```

If you enter an invalid command the terminal prints that an INVALID command is printed and you the map is reloaded.



```
*Main> load "testRight.txt"

File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - * t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

lol
INVALID COMMAND, start from beginning

File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - * t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game
```

Basic Requirements/Functionalities

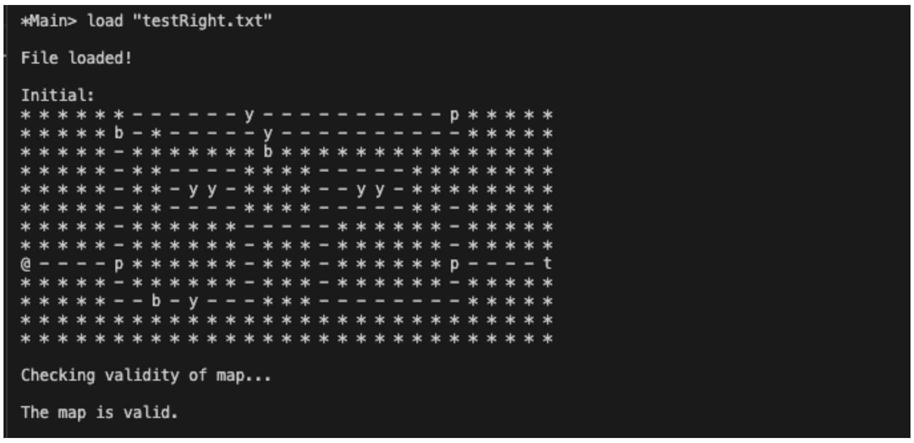
1. Load

The loading functionality loads a map stored in a .txt file. It's done via the `load :: String -> IO()` function in kodable.hs

Brief explanation of the function:
The function takes a String which is the fileName and uses the built in `readFile` function of Haskell to return and `IO String` which is given to the `lines` function to convert the String to `[String]` which is the data structure of the map. The function then checks the validity of the map and proceeds to show the game interface.

Steps to load game –

1. After building the game, type `load "fileName.txt"` in the terminal to load the map.

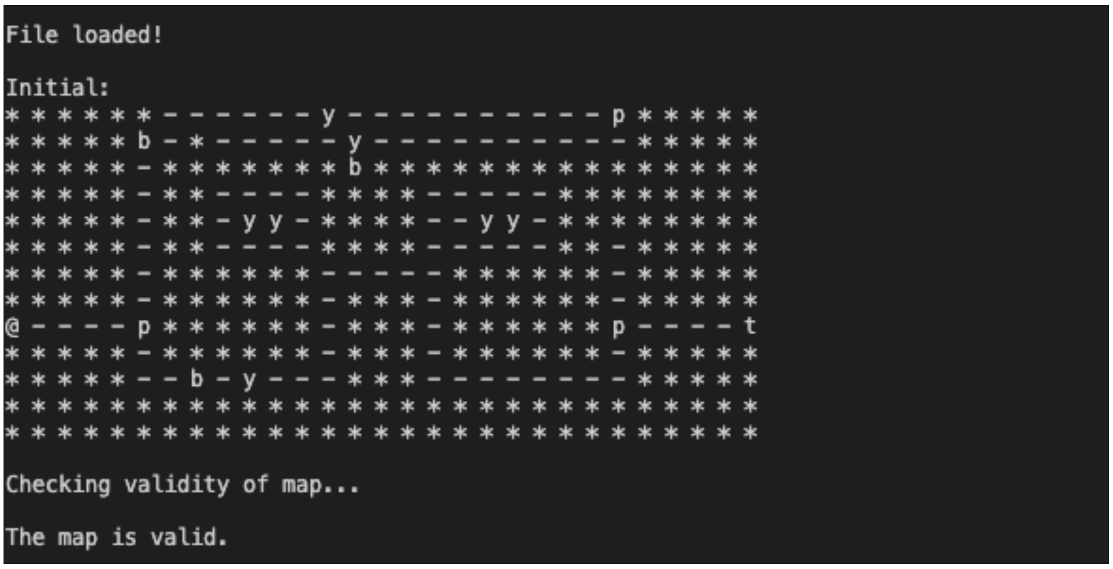


```
*Main> load "testRight.txt"

File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.
```

The load function also checks for the validity of the map and prints if the map is valid or not

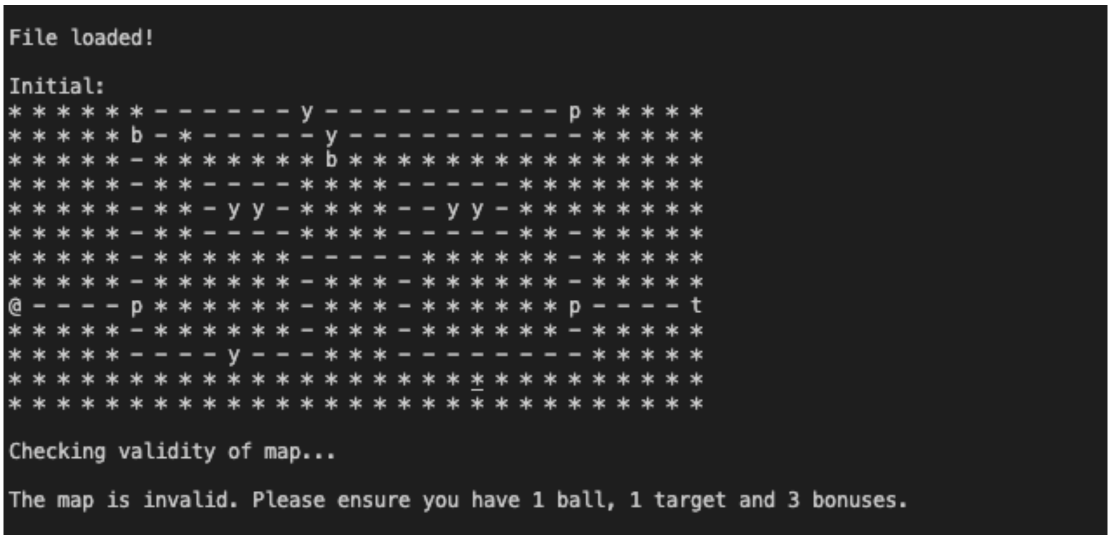
A valid map is the one which has **exactly 1 ball, 1 target, 3 bonuses.**

Example of a valid map



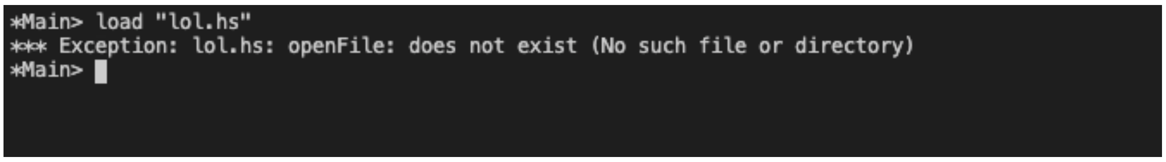
```
File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.
```

Example of an invalid map



```
File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is invalid. Please ensure you have 1 ball, 1 target and 3 bonuses.
```

If you open a file that doesn’t exits or a non .txt file it throws and exception



```
*Main> load "lol.hs"
*** Exception: lol.hs: openFile: does not exist (No such file or directory)
*Main> 
```

2. Check

The check functionality uses the logic implemented in mazeSolver.hs to return whether or not a map is solvable or not.

Brief Explanation of the solveMaze function in mazeSolver.hs

The `solveMaze :: [String] -> (Int,Int) -> [(Int, Int)] -> String -> Bool` is the main function that takes a map, (xCoordinateOfBall, yCoordinateOfBall), a list of visited nodes, the current direction where the ball is going.

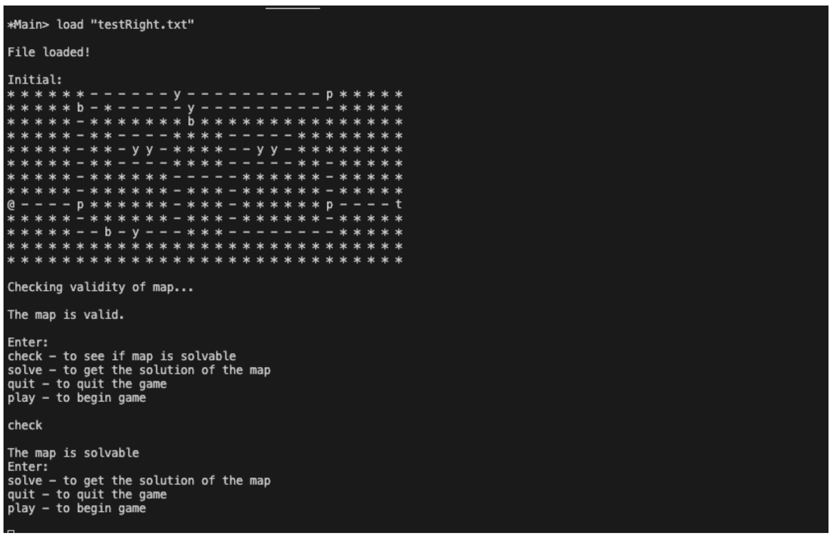
solveFunction is a recursive function which call itself until and unless the ball has reached target ‘t’ or the ball has not reached ‘t’ but has no possible moves. In the case of the former, it returns true and false if the later condition is satisfied.
If none of the above condition is satisfied, it then explores various paths the ball can travel to using the possible moves at a given node where the ball stops to change direction. This is done through the recursive call of solveMaze with different directions in the currentDirection parameter and different positions of the ball. A visited list is kept to prevent the ball from retracing already traversed path. If any of the path returns True, solveMaze returns true

For more detailed information, refer to mazeSolver.hs which contains various helper functions (with comments) for implementing the Check functionality.

Steps to run the check command

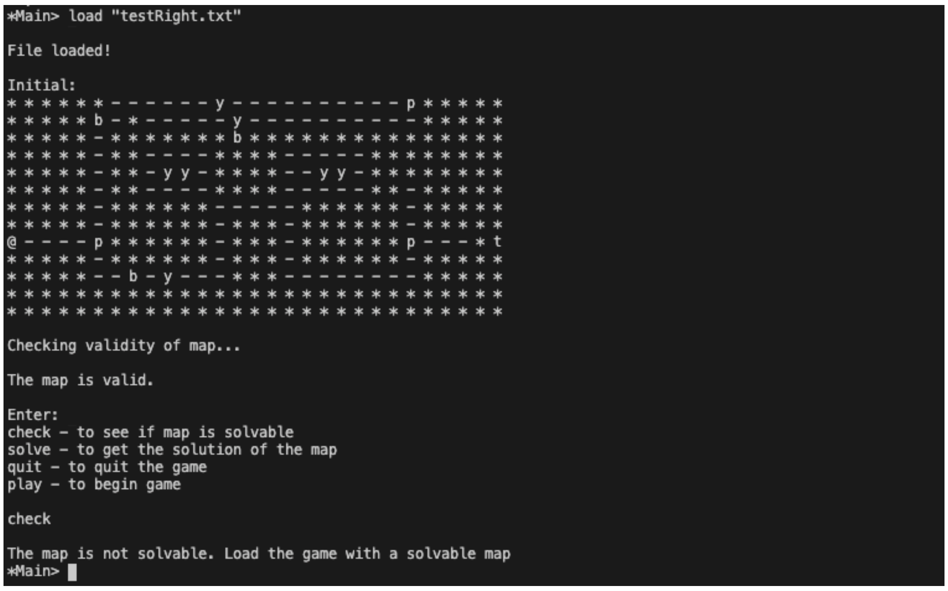
1. After loading the map type `check` in terminal to see if a map is solvable or not.

   Example of a map which is solvable



```
*Main> load "testRight.txt"

File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

check

The map is solvable
Enter:
solve - to get the solution of the map
quit - to quit the game
play - to begin game
```

Example of map which is not solvable



```
*Main> load "testRight.txt"

File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - * t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

check

The map is not solvable. Load the game with a solvable map
*Main>
```

3. Solve

The solve functionality uses the logic implemented in optimalSolution.hs to return whether the solution to a map
An optimal Solution is defined as the path followed by the ball which collects all collectable bonuses and reaches the target in least number of change in directions.

Brief Explanation of optimalSolution.hs

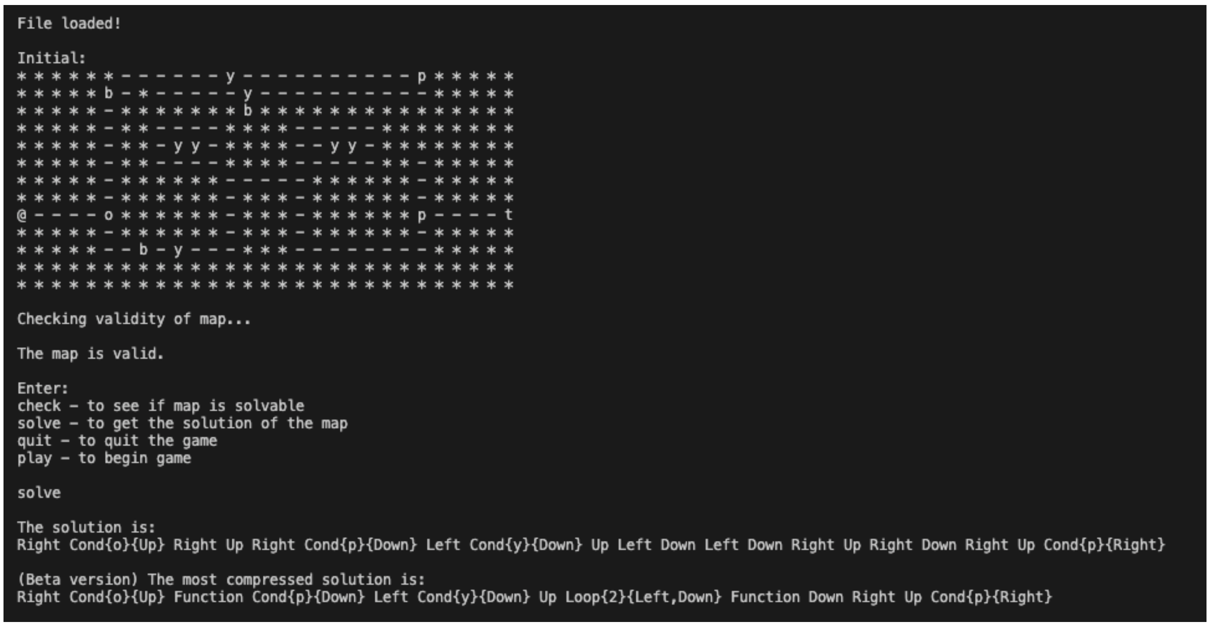
The `allSolution :: [String] -> (Int, Int) -> [String] -> Int -> [(Int, Int, Int)] -> Int -> [[String]]`
is the main function which takes a map, ball position, and empty list [] in which all solutions are added, the current bonus count, the visited nodes and the final target bonus that decides for this function to terminate. The function terminates if the ball has reached 't' and the bonus count has been achieved. If this is fulfilled, the current solution is added to a list. If the current ball position is t but number of bonus eaten is not equal to the target count, then we say that the path is redundant, not optimal and [] list appended.
If neither of the above 2 conditions are fulfilled we send the ball to all 4 directions and keep appending the various solutions received. A visited is maintained which also **contains the bonus count which acts as the unique identifier** while choosing nodes to go to. This is done so that the ball can retrace its path back after eating a bonus. For eg, if the ball stops at (8,0) with the bonus count being 0 we add (8,0,0) to visited. It then proceeds further and eats a bonus at (8,10). The bonus count is now updated to 1 and while coming back the ball checks if it has visited (8,0,**1**) and not (8,0,0). This enables it to retrace paths back to 't' after eating the various bonus scattered on the board.

The `compressOptSolUtil :: (Eq a, Num a, Show a) => [String] -> Int -> [String] -> Bool -> [String] -> a -> [String] -> [String]`

The above function is used to compress the optimal solution by using functions and loops. For eg if the solution contains (Right, Up, Right, Up) this function converts it to Loop{2}{Right,Up}. To this, this function takes a solution and splits into three branches. The first branch is for the direction to become a loop, second branch is for the direction to not do anything and third branch is for the direction to become part of a function. In every loop branch if a loop already exits and the similar directions come to the branch then the loop counts is incremented by 1. This function calls itself recursively to parse every direction and add it to a loop/function or doesn't change it. In the end various compressed solutions are returned out of which the shortest one is taken.

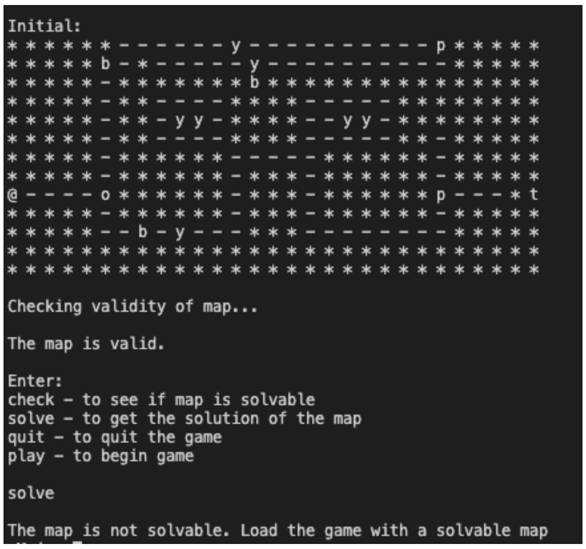
Steps to run the solve command

1. After loading the map type `solve` in terminal to see if a 2 kinds of list.
   The first list shows the most optimal solution of the path and the second list shows the most compressed form of that solution using loops and functions



```
File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

solve

The solution is:
Right Cond{o}{Up} Right Up Right Cond{p}{Down} Left Cond{y}{Down} Up Left Down Left Down Right Up Right Down Right Up Cond{p}{Right}

(Beta version) The most compressed solution is:
Right Cond{o}{Up} Function Cond{p}{Down} Left Cond{y}{Down} Up Loop{2}{Left,Down} Function Down Right Up Cond{p}{Right}
```

   If a map is not solvable an error is shown



```
Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - o * * * * * * - * * * - * * * * * * p - - - * t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

solve

The map is not solvable. Load the game with a solvable map
```

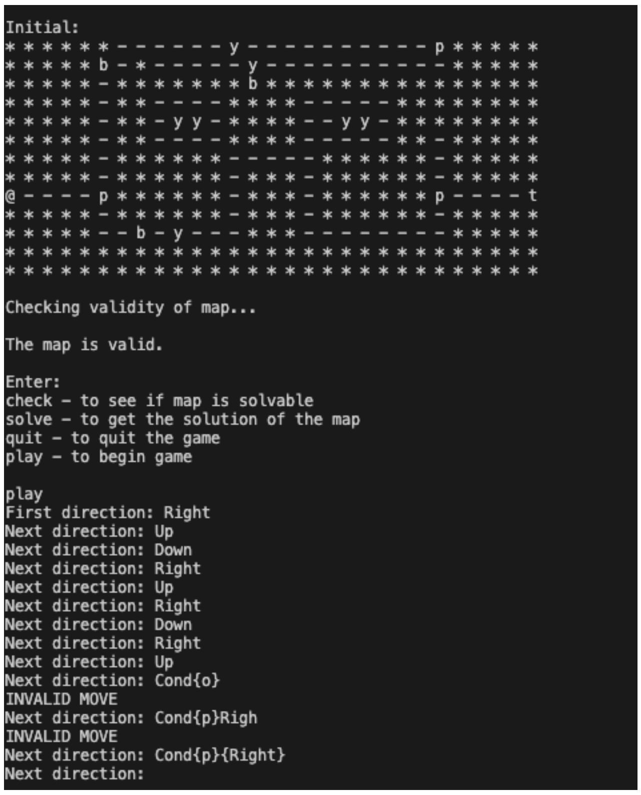
4. Play

The play functionality is the main function through which the game can be played. Its logic is applied in kodable.hs. It uses various helper functions to move the ball around the board and terminates if no more directions are left to execute. It handles various errors that will be explained below.

Steps to run the play command

1. After loading the map type `play` in terminal to play the game.

   The interface of play is as follows



```
Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

play
First direction: Right
Next direction: Up
Next direction: Down
Next direction: Right
Next direction: Up
Next direction: Right
Next direction: Down
Next direction: Right
Next direction: Up
Next direction: Cond{o}
INVALID MOVE
Next direction: Cond{p}Righ
INVALID MOVE
Next direction: Cond{p}{Right}
Next direction:
```

You can type directions directly from the terminal**. Directions can be of 4 types**

i) **Basic directions** - There are four basic directions: Left, Right, Up and Down

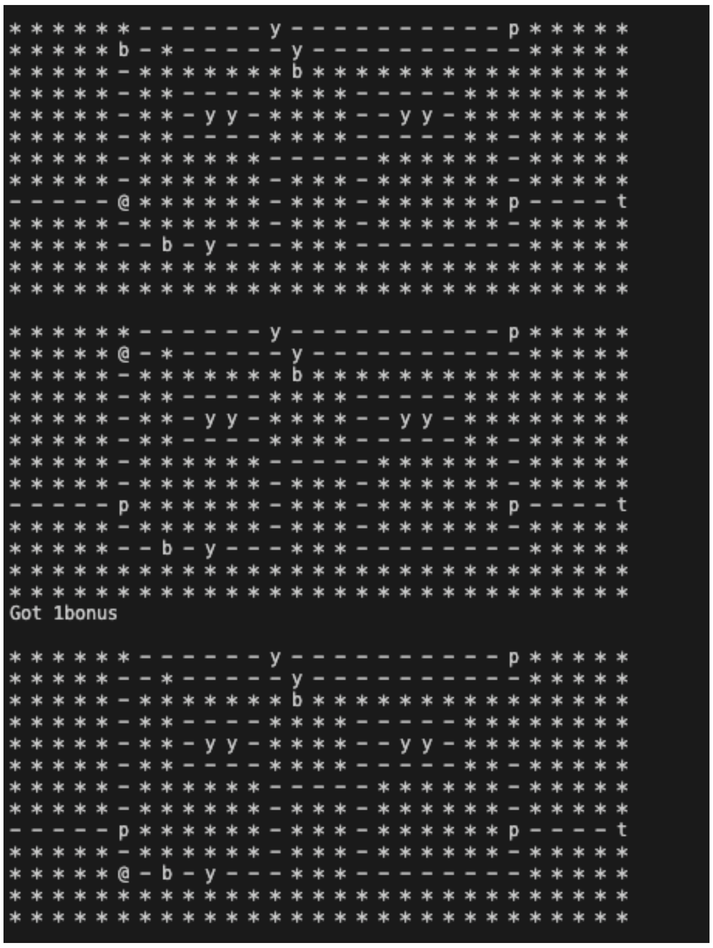
ii) **Conditionals** - Conditionals can be expressed with a syntax of the form: Cond{color}{Direction}. For example Cond{p}{Right} expresses that the ball should turn to the right when it encounters a pink block.

iii) **Loops** can be expressed with a syntax of the form: Loop{n}{Direction,Direction}. For example Loop{3}{Right,Up} expresses that the ball should move right, then up 3 times. The iteration of the loop can range from 0 to 5. Thus, loops with larger numbers are errors.

iv) **Functions -** I have assumed that only one function can be defined (like in Kodable). The syntax of functions involves two constructs. The function call is done by writing the keyword Function. The definition of the function is done by appending with Direction Direction Direction to the set of instructions.

After getting all the directions, play executes the command sequentially and prints the board after every move

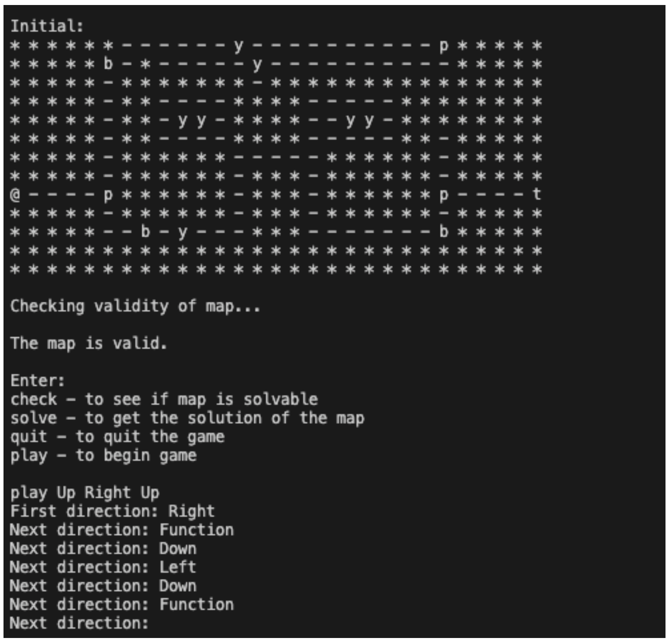
For eg if the first three commands (Right, Up, Down) are executed we get



```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - @ * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 1bonus

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * @ - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

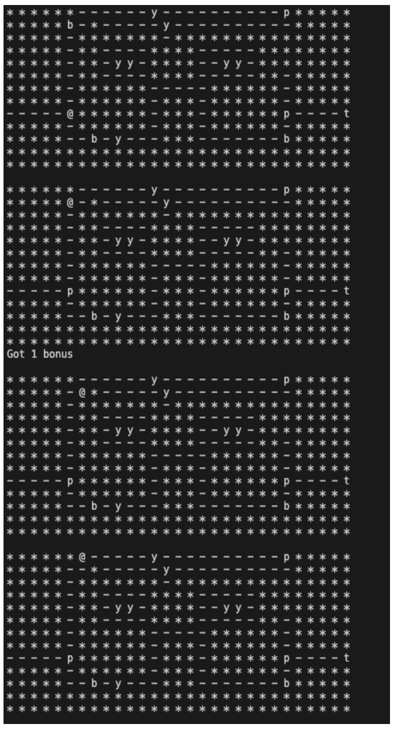
As you can notice, the number of bonus that the user has eaten is also printed

Play can also be played with a different by inputting a function after the `play` command in the form of `play dir1 dir2 dir3`. A function is then called whenever "Function" is input in the Next Direction: space.
An example is shown below

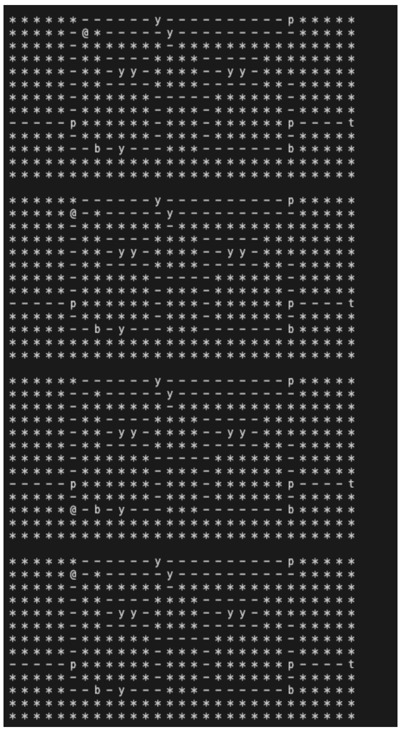


```
Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

play Up Right Up
First direction: Right
Next direction: Function
Next direction: Down
Next direction: Left
Next direction: Down
Next direction: Function
Next direction:
```

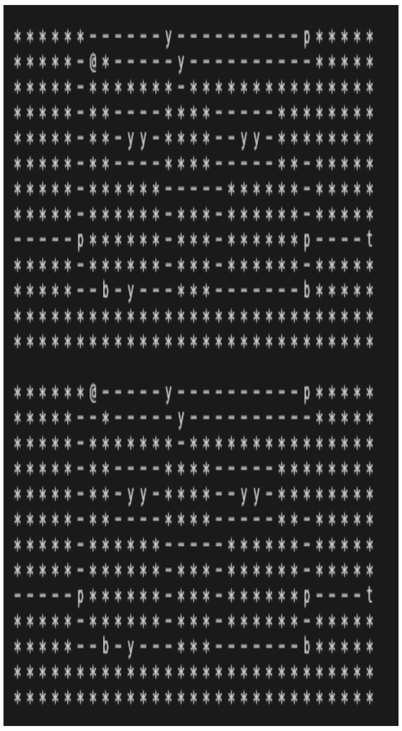
The maps that are printed following these commands are shown below



```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - @ * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 1 bonus

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - @ * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * @ - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```



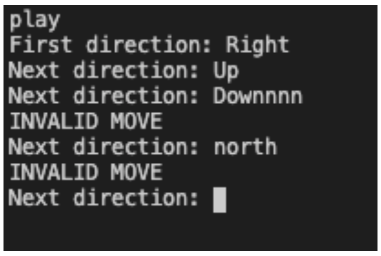
```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - @ * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * @ - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```



```
******------y----------p*****
*****-@*-----y----------*****
*****-*******-***************
*****-**----****-----********
*****-**-yy-****--yy-********
*****-**----****-----**-*****
*****-******-----******-*****
*****-******-***-******-*****
-----p******-***-******p----t
*****-******-***-******-*****
*****--b-y---***-------b*****
*****************************
*****************************

******@-----y----------p*****
*****--*-----y----------*****
*****-*******-***************
*****-**----****-----********
*****-**-yy-****--yy-********
*****-**----****-----**-*****
*****-******-----******-*****
*****-******-***-******-*****
-----p******-***-******p----t
*****-******-***-******-*****
*****--b-y---***-------b*****
*****************************
*****************************
```

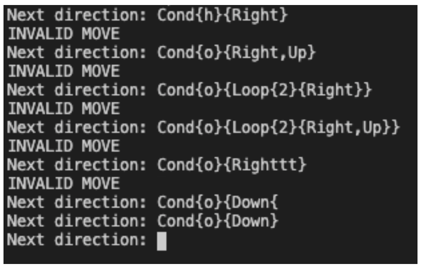
Error checking is also done to ensure that the directions entered follow a certain syntax

a. It is ensured that other than the allowed directions, no other word can be inputted.



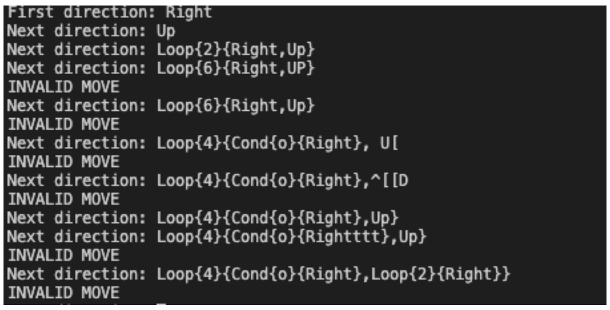
```
play
First direction: Right
Next direction: Up
Next direction: Downnnn
INVALID MOVE
Next direction: north
INVALID MOVE
Next direction: █
```

b. For a condition, the colour can only be 1 out of. ‘o’, ‘y’, ‘p’. We can also not have more than 1 direction neither can be have loops in condition. The directions that are input should belong to either of “Right”, “Left”, “Up” or “Down”



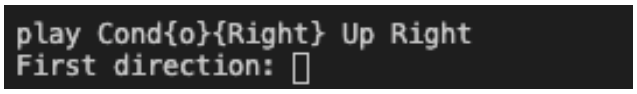
```
Next direction: Cond{h}{Right}
INVALID MOVE
Next direction: Cond{o}{Right,Up}
INVALID MOVE
Next direction: Cond{o}{Loop{2}{Right}}
INVALID MOVE
Next direction: Cond{o}{Loop{2}{Right,Up}}
INVALID MOVE
Next direction: Cond{o}{Righttt}
INVALID MOVE
Next direction: Cond{o}{Down{
Next direction: Cond{o}{Down}
Next direction: █
```

c. For loop, the number cannot be more than 5, only 2 directions are allowed. Condition directions are allowed but loop inside a loop is not allowed.

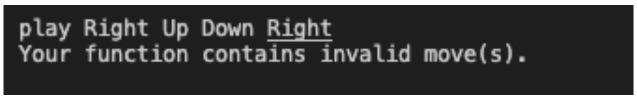


```
First direction: Right
Next direction: Up
Next direction: Loop{2}{Right,Up}
Next direction: Loop{6}{Right,UP}
INVALID MOVE
Next direction: Loop{6}{Right,Up}
INVALID MOVE
Next direction: Loop{4}{Cond{o}{Right}, U[
INVALID MOVE
Next direction: Loop{4}{Cond{o}{Right},^[[D
INVALID MOVE
Next direction: Loop{4}{Cond{o}{Right},Up}
Next direction: Loop{4}{Cond{o}{Rightttt},Up}
INVALID MOVE
Next direction: Loop{4}{Cond{o}{Right},Loop{2}{Right}}
INVALID MOVE
```

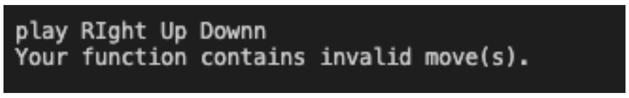
d. For a function, loops are not allowed. Only 2 directions are allowed and conditional directions are allowed.



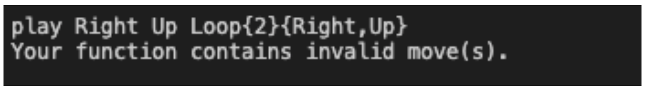
```
play Cond{o}{Right} Up Right
First direction: ▯
```



```
play Right Up Down Right
Your function contains invalid move(s).
```



```
play RIght Up Downn
Your function contains invalid move(s).
```

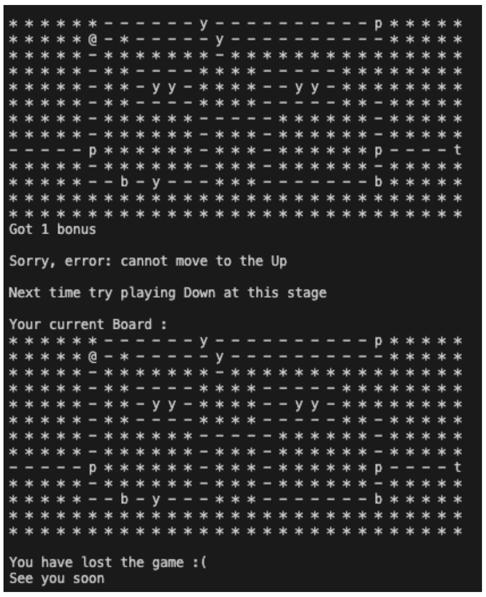


```
play Right Up Loop{2}{Right,Up}
Your function contains invalid move(s).
```

If an error message appears the board is printed again and you are prompted to type play with the right commands.

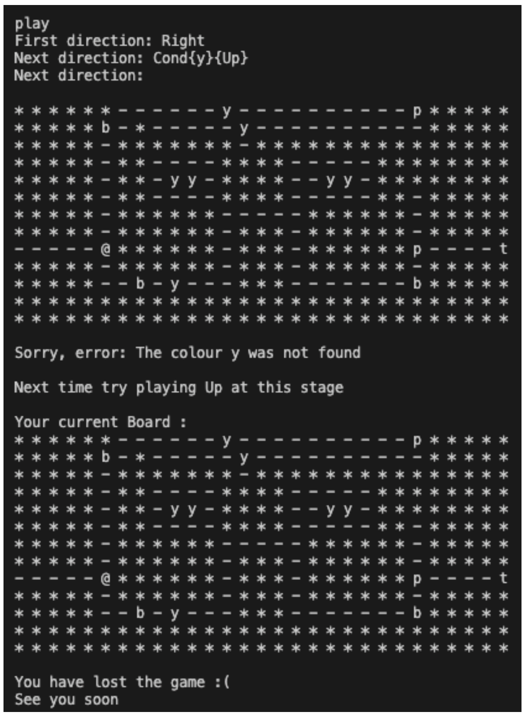
Play terminated in three cases –

1. If the user enters a direction that the ball cannot go to, play gets terminated with a message saying
   Sorry, error: Cannot move to the Direction



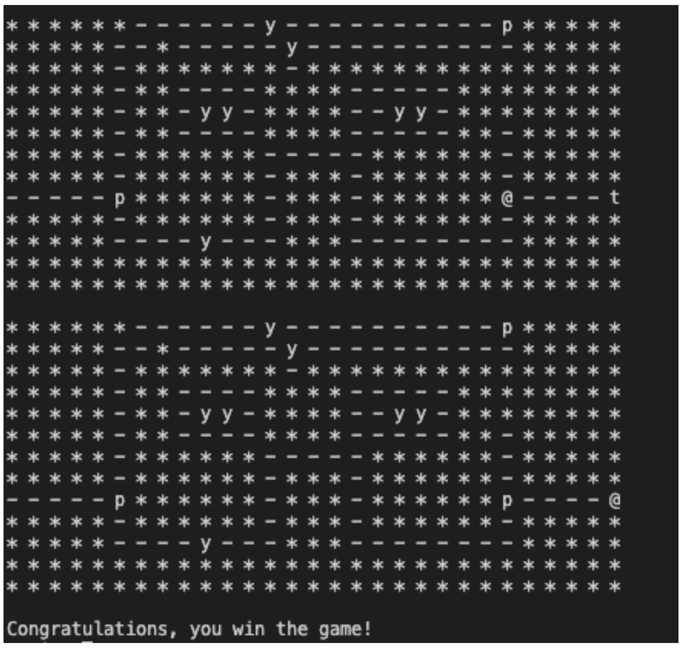
```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 1 bonus

Sorry, error: cannot move to the Up

Next time try playing Down at this stage

Your current Board :
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

You have lost the game :(
See you soon
```

2. If the user enters a Conditional direction and the colour doesn't exist, play terminates with an error message.
   Sorry, error: The colour 'colour' was not found.



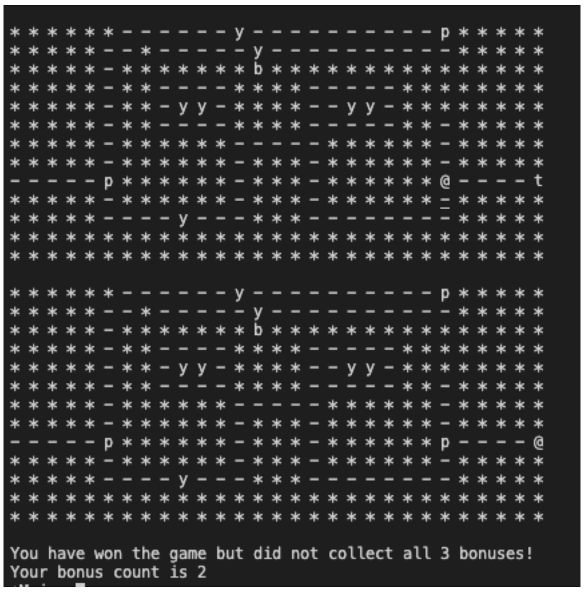
```
play
First direction: Right
Next direction: Cond{y}{Up}
Next direction:

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - @ * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Sorry, error: The colour y was not found

Next time try playing Up at this stage

Your current Board :
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - @ * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

You have lost the game :(
See you soon
```

3. Finally if the user wins play gets terminated with 2 kinds of end messages. If the user wins with all bonus collected, the end message is
   Congratulations, you win the game!



```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * @ - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - @
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Congratulations, you win the game!
```

If the users wins but not all bonus are collected, the game prints –
   You have won the game but did not collect all 3 bonuses!
   Your bonus count is *bonusCount*



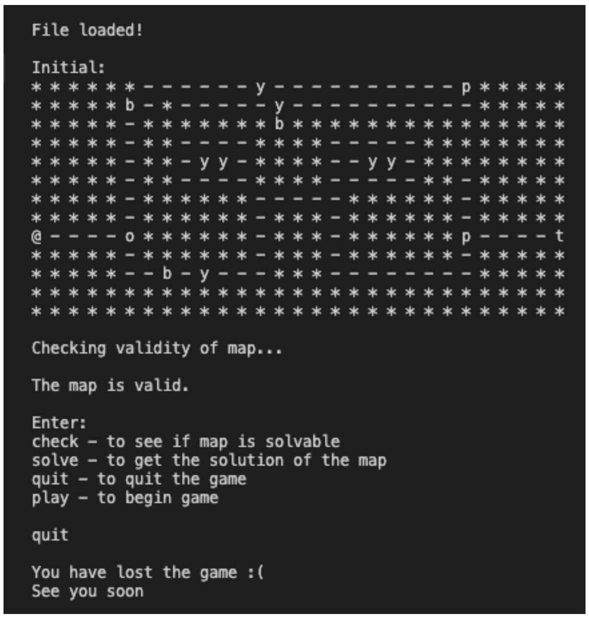
```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * @ - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - p * * * * * * - * * * - * * * * * * p - - - - @
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

You have won the game but did not collect all 3 bonuses!
Your bonus count is 2
```

5. Quit

The Quit command is used to leave the game by the user.

Steps to run the quit command

1. After loading the map the game interface is shown. User can type `quit` to quit the game before user has even started

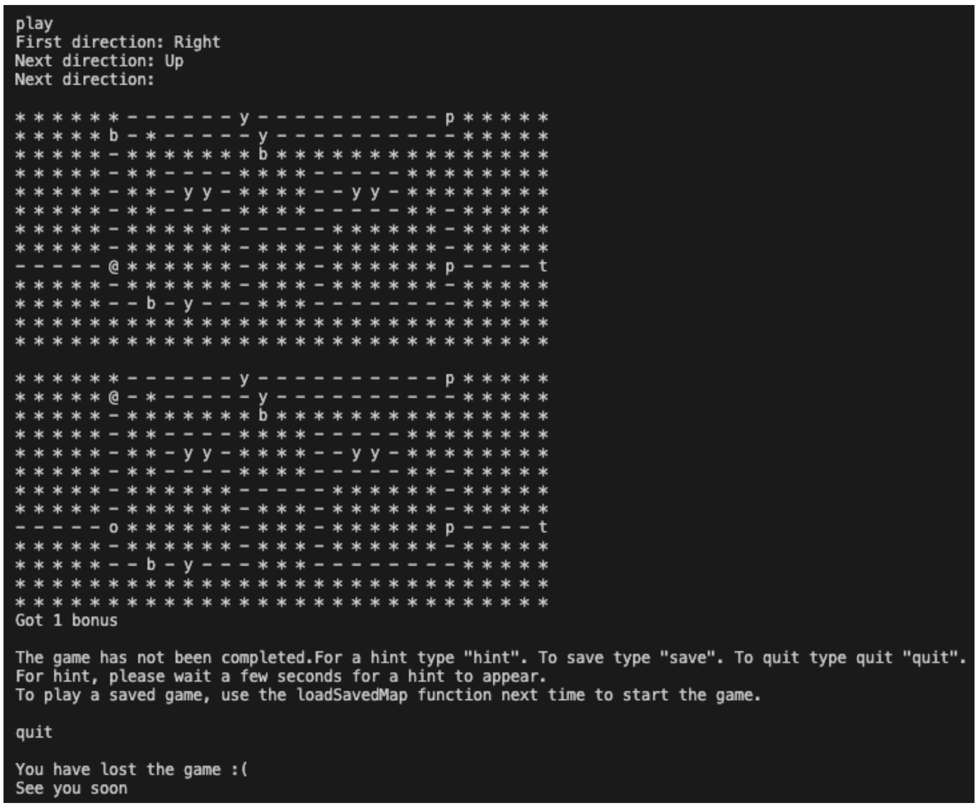


```
File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.

Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game

quit

You have lost the game :(
See you soon
```

2. If the user enters certain directions and cannot think of any move, they can press enter to see the output of their inputted directions.
   After that the game prompts
   The game has not been completed. For a hint type "hint". To save type "save". To quit type quit "quit".
   For hint, please wait a few seconds for a hint to appear.

To play a saved game, use the loadSavedMap function next time to start the game.

To get the hint, user can simply input `quit.` After the user inputs quit, the game is quitted.



```
play
First direction: Right
Next direction: Up
Next direction:

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - @ * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 1 bonus

The game has not been completed.For a hint type "hint". To save type "save". To quit type quit "quit".
For hint, please wait a few seconds for a hint to appear.
To play a saved game, use the loadSavedMap function next time to start the game.

quit

You have lost the game :(
See you soon
```

Ending of Game

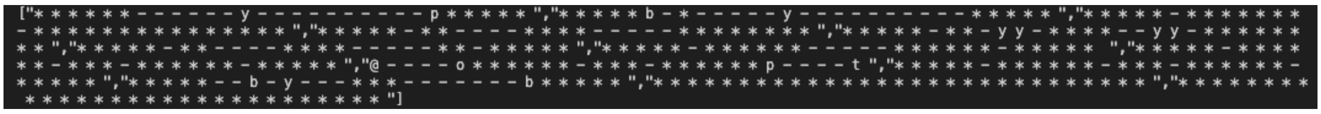
The ending of game is described on pg number 15 along with screenshots of the messages displayed.

Error Handling

I have handled various edge cases for the multiple requirements for this project and the error messages with the corresponding handled case can be seen in the Basic requirements section of the report and the additional feature section.

Choice of Data Structure

I have stored the maps as a list of Strings – [String], which means that every row is stored as one elements of the list as string. This is done for easy indexing so that I can easily modify a row while moving the ball around. Treating it as a 2-dimensional array I was easily able to take out the positions of the various characters in the map and use them for my functions. Moreover, Haskell has great support for String type and List type variables because of which I was able to use several inbuilt functions. While reading the .txt files I could use the built in readFile and lines function to convert the map to a [String] type variable. An example of the map as a [String] is shown below –



```
["* * * * * * - - - - - - y - - - - - - - - - - p * * * * * ","* * * * * b - * - - - - - y - - - - - - - - - - * * * * * ","* * * * * - * * * * * * *
- * * * * * * * * * * * * * * * ","* * * * * - * * - - - - * * * * - - - - - * * * * * * * * ","* * * * * - * * - y y - * * * * - - y y - * * * * * *
* * ","* * * * * - * * - - - - * * * * - - - - - * * - * * * * * ","* * * * * - * * * * * * - - - - - * * * * * * - * * * * *  ","* * * * * - * * * *
* * - * * * - * * * * * * - * * * * * ","@ - - - - o * * * * * * - * * * - * * * * * * p - - - - t ","* * * * * - * * * * * * - * * * - * * * * * * -
* * * * * ","* * * * * - - b - y - - - * * * - - - - - - - b * * * * * ","* * * * * * * * * * * * * * * * * * * * * * * * * * * * * ","* * * * * * * *
 * * * * * * * * * * * * * * * * * * * * * "]
```

Bonus Features

I have implemented the following bonus features –

1. **Hint**
2. **Saving the game**
3. **Undo the last direction**
4. **Suggestions to try next time if the game is lost**
5. **Map validity**

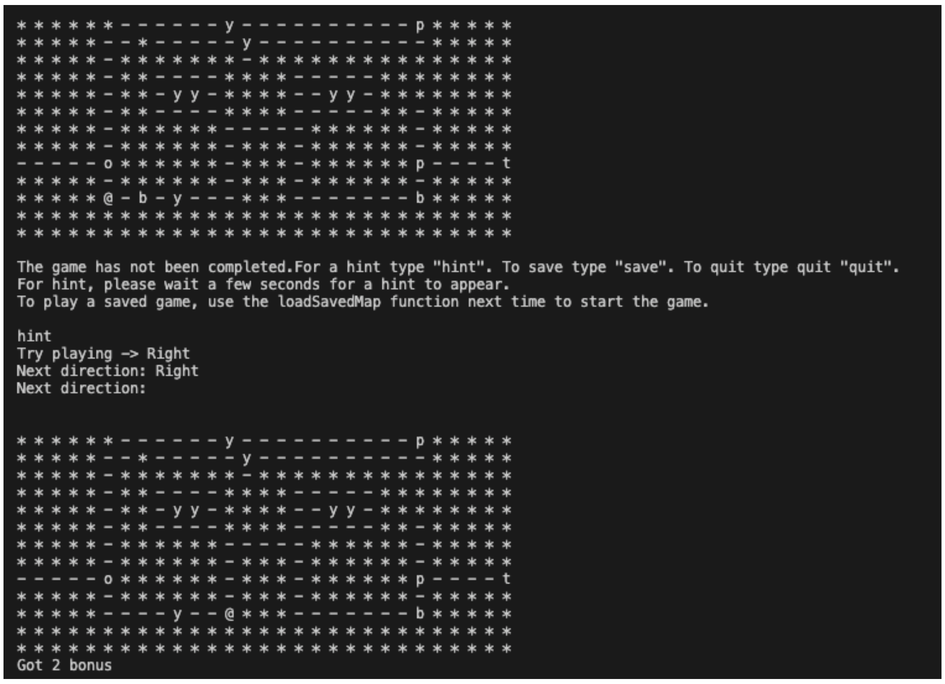
1. Hint

If the user enters certain directions and cannot think of any move, they can press enter to see the output of their inputted directions.

After that the game prompts

The game has not been completed. For a hint type "hint". To save type "save". To quit type quit "quit".
For hint, please wait a few seconds for a hint to appear.
To play a saved game, use the loadSavedMap function next time to start the game.

To get the hint, user can simply input `hint.` After the user inputs hint, they can start playing the game from the point they stopped.



```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * @ - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

The game has not been completed.For a hint type "hint". To save type "save". To quit type quit "quit".
For hint, please wait a few seconds for a hint to appear.
To play a saved game, use the loadSavedMap function next time to start the game.

hint
Try playing -> Right
Next direction: Right
Next direction:


* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - @ * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 2 bonus
```

Hint is displayed as Try playing -> Right. When the user did go right they ended up collecting a bonus and move closer to the target.

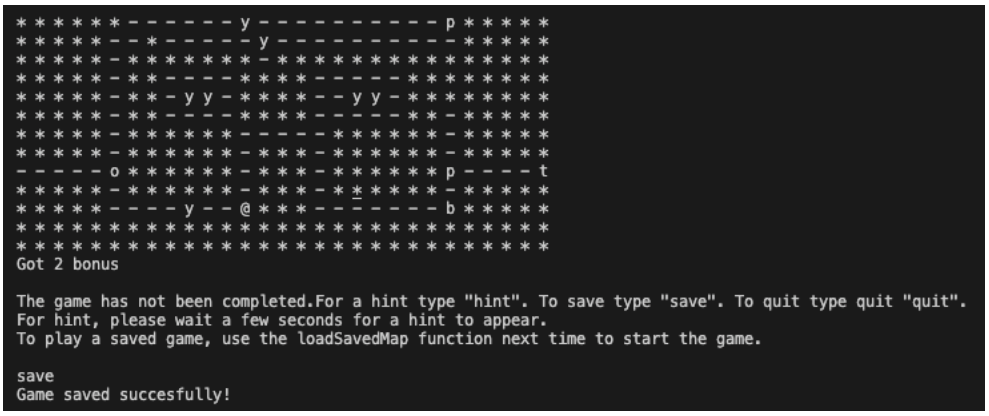
2. Saving a game

If the user enters certain directions and cannot think of any move, they can press enter to see the output of their inputted directions.

After that the game prompts

The game has not been completed. For a hint type "hint". To save type "save". To quit type quit "quit".
For hint, please wait a few seconds for a hint to appear.
To play a saved game, use the loadSavedMap function next time to start the game.

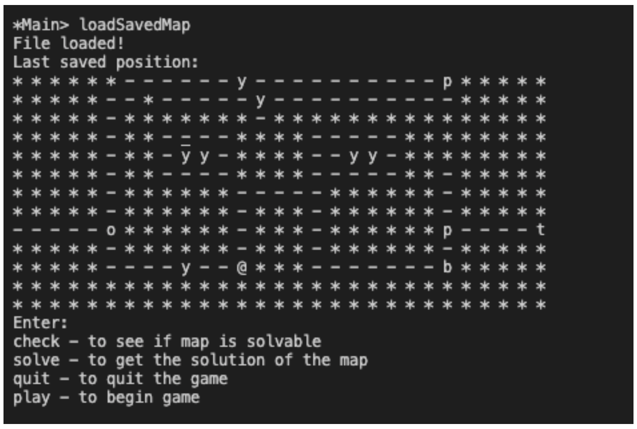
To save a game, user can simply input `save`. After this a message is displayed showing successful save!



```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - - * * * * - - - - - * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - - * * * * - - - - - * * - * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - @ * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 2 bonus

The game has not been completed.For a hint type "hint". To save type "save". To quit type quit "quit".
For hint, please wait a few seconds for a hint to appear.
To play a saved game, use the loadSavedMap function next time to start the game.

save
Game saved succesfully!
```

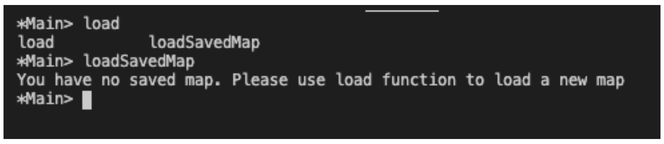
The save functionality saves the current map state as a string in a new file which it creates called savedMap.txt

To load this file the user has to input `loadSavedMap` to load the map from the last saved state. User can play the game again from this state.



```
*Main> loadSavedMap
File loaded!
Last saved position:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - @ * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Enter:
check - to see if map is solvable
solve - to get the solution of the map
quit - to quit the game
play - to begin game
```

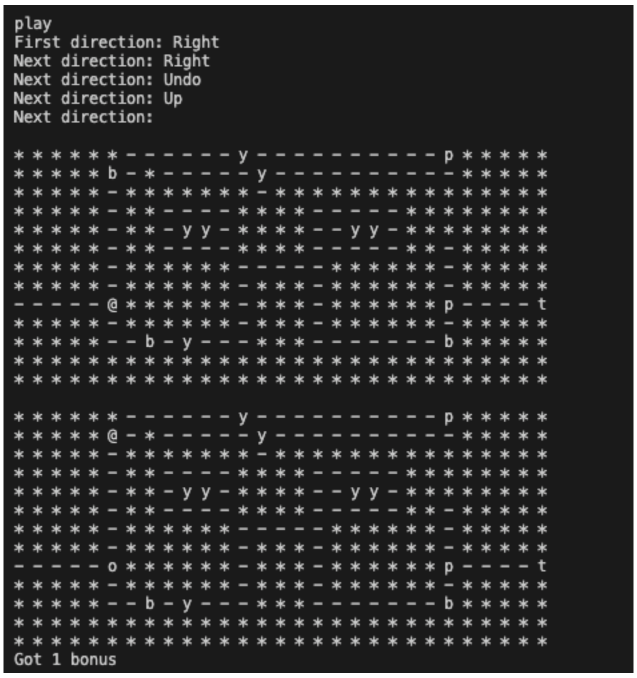
In case the user has never saved a file and still calls loadSavedMap function it displays an error.



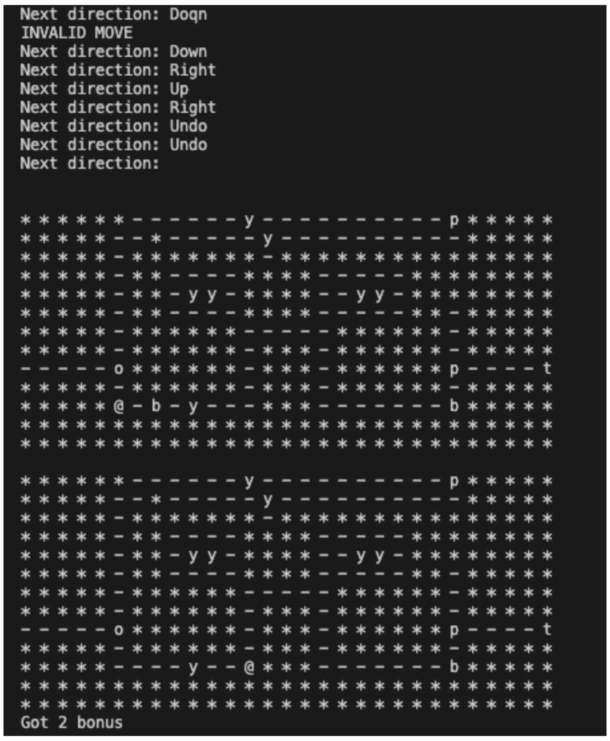
```
*Main> load
load            loadSavedMap
*Main> loadSavedMap
You have no saved map. Please use load function to load a new map
*Main> █
```

3. Undo

A user can use undo to remove a direction they might have inputted by mistake. This can simply be done by typing Undo in the Next Direction space.



```
play
First direction: Right
Next direction: Right
Next direction: Undo
Next direction: Up
Next direction:

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - @ * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 1 bonus
```

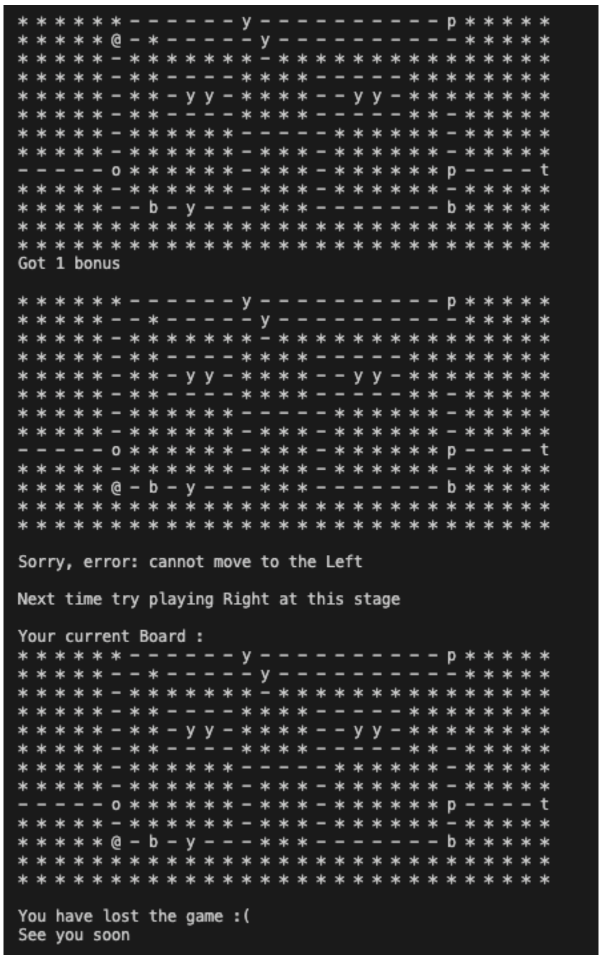


```
Next direction: Doqn
INVALID MOVE
Next direction: Down
Next direction: Right
Next direction: Up
Next direction: Right
Next direction: Undo
Next direction: Undo
Next direction:


* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * @ - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - @ * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 2 bonus
```

You can type undo as many times as you want.

4. Suggestions

Suggestions are used in case a user loses the game to tell them what they could have played and what they can play the next time they play the game and come to a similar situation.



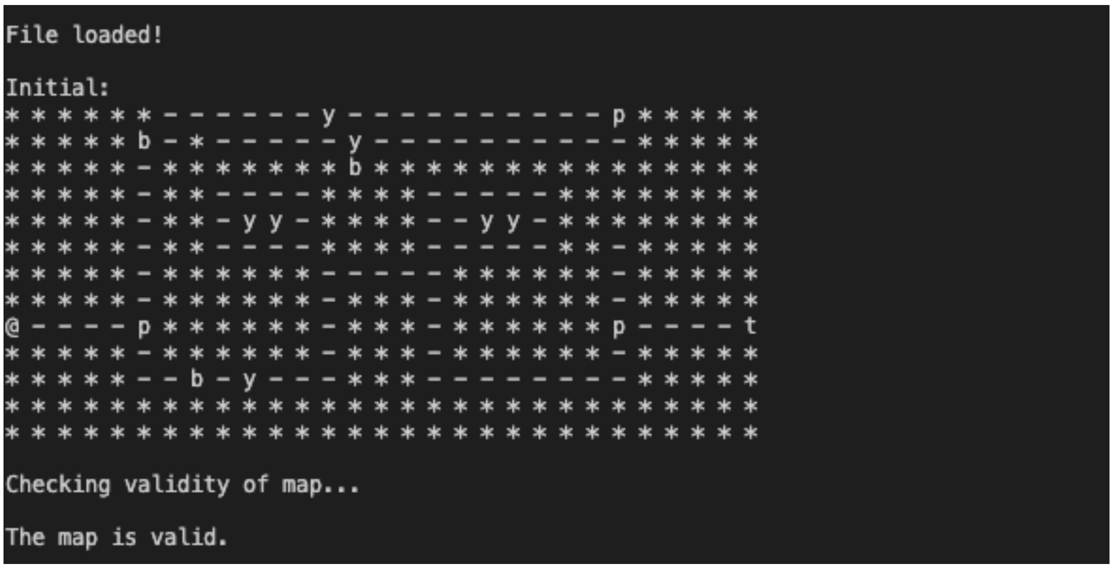
```
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * @ - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Got 1 bonus

* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * @ - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Sorry, error: cannot move to the Left

Next time try playing Right at this stage

Your current Board :
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * - - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * - * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
- - - - - o * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * @ - b - y - - - * * * - - - - - - - b * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

You have lost the game :(
See you soon
```

As you can see, after the error message the game shows which move the user could have played instead of going Left.

5. Validity

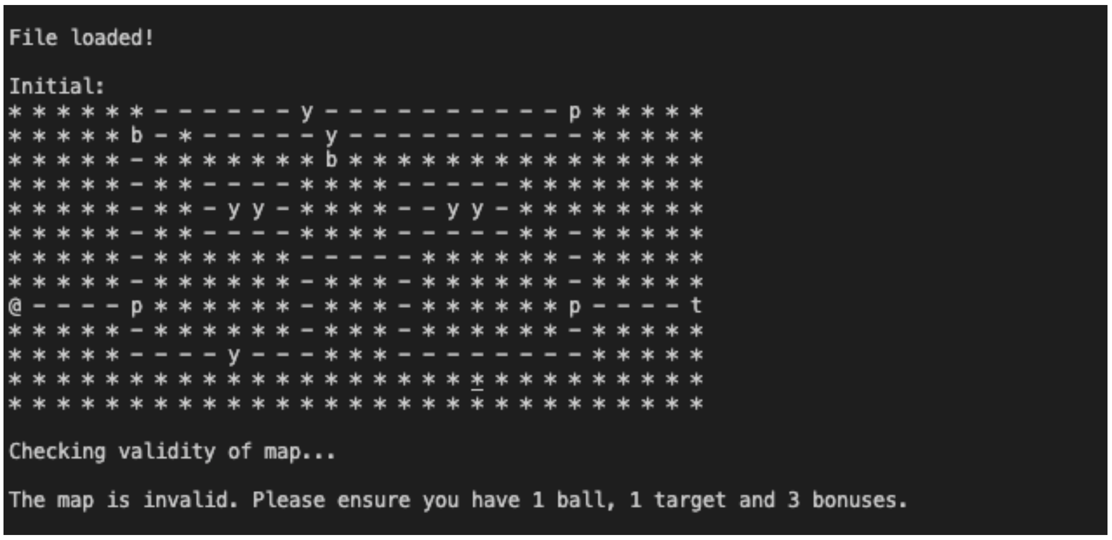
As described in the beginning of the report, every time a map is loaded it is checked for its validity which is described as a map having **exactly 1 ball, 1 target, 3 bonuses**.

Example of a valid map



```
File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - b - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is valid.
```

Example of an invalid map



```
File loaded!

Initial:
* * * * * * - - - - - - y - - - - - - - - - - p * * * * *
* * * * * b - * - - - - - y - - - - - - - - - - * * * * *
* * * * * - * * * * * * * b * * * * * * * * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * * * * * * *
* * * * * - * * - y y - * * * * - - y y - * * * * * * * *
* * * * * - * * - - - - * * * * - - - - - * * - * * * * *
* * * * * - * * * * * * - - - - - * * * * * * - * * * * *
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
@ - - - - p * * * * * * - * * * - * * * * * * p - - - - t
* * * * * - * * * * * * - * * * - * * * * * * - * * * * *
* * * * * - - - - y - - - * * * - - - - - - - - * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Checking validity of map...

The map is invalid. Please ensure you have 1 ball, 1 target and 3 bonuses.
```