COMP3258 Final Project – Report

Name: Ng Wing Ki

University Number: 3035572648

**NOTE:

- Only maps with numbers are acceptable.
- The data types of directions and colors are similar but not exactly the same as those described in project file:
- Program files: Main.hs and Smeeborg.hs
- Testcases: all text files of ".txt"

```
data Color = Pink | Orange | Yellow deriving (Eq, Show)
data Dirction = DUp | DDown | DLeft | DRight  deriving (Eq, Show)
data Action = Base Dirction | Cond Color Dirction | Loop Int Action Action |
Function Action Action Action deriving (Eq, Show)
```

1. Describing how to build the project (example run is in Point 5)

1.1 The player needs to ensure that all imported modules used in the program are installed

If Data.List.Split is not installed yet, type command line "`cabal install –lib split`"

1.2 Put the map files into the same directory with the program files

1.3 Type command line "`ghci Main.hs`"

2. Describing the functionality

2.1 Load the map

If the file name of map is called map.txt, then to load the map, type 1 and

```
map.txt
```

The program only supports the maps represented in text

If the map is in rectangle shape and the content are numerical characters, then the map is valid, it will output the message "Reading map successfully!".

After the "Initial: ", the input map will be printed.

If the map is not valid, the program will request the player to input another file again.

2.2 Check the map is solvable

The program can check whether the map is solvable or not, by typing 2.

If solvable, "The currently loaded map is solvable!" will be printed

If not solvable (e.g. no connection between start and end points), “Input wrong, select the options again!” will be printed

(one unsolvable example is the self-cannotexit.txt)

2.3 Compute an ideal optimal solution

To get the ideal optimal solution, type number 3:

The program will produce a minimal solution that requires less instructions and directions

Way to model the path for computed solution:

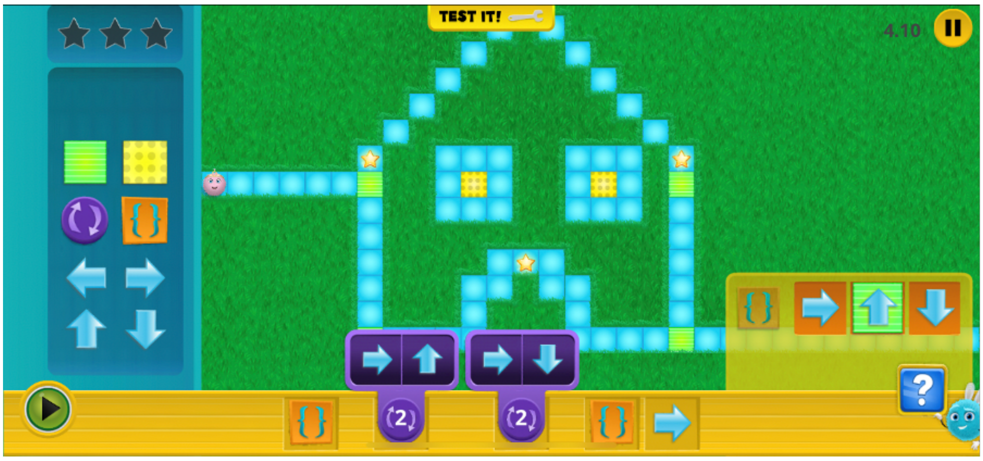
(the following is referenced from the project description file)

- *Basic directions.* There are four basic directions: `DLeft, DRight, DUp` `and DDown`
- *Conditionals.* Conditionals can be expressed with a syntax of the form: `Cond color Direction` . For example `Cond Pink Right` expresses that the ball should turn to the right when it encounters a pink block.
- *Loops.* Loops can be expressed with a syntax of the form: `Loop n Action Action`. For example `Loop 3 DRight DUp` 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.
- *Functions.* It is assumed that only one function can be defined. The syntax of functions involves two constructs. The function call is done by writing the keyword `Function Action Action Action`.

The following example using basic directions, conditionals, loops and functions (referenced from the project description file):

```
[Function DRight Cond Yellow DUp DDown, Loop 2 DRight Dup,
Loop 2 DRight,DDown, Function DRight Cond Yellow DUp DDown,
DRight]
```

is representing the following path.



2.4 Play the game

The player can propose his/her own solution to solve the game.

To start the game, after loading the map, the player can type play.

```
play
```

After that, player need to give instructions one-by-one.

Basic directions, conditionals and loops can be inputted

(functions need to be implemented at first, see point 2.5)

When the player does not type any instructions and simply presses enter, the ball will go through as the instructions, and the change of the map will be shown step-by-step (after the “test: ”statement).

- Using basic directions

  Example (referenced from the project description file)

```
> play
First direction: Right
Next direction: Down
Next direction: Right
Next direction: Up
Next direction: Right
Next direction: Down
....
Next direction: <Enter> ("here the player simply presses
enter")
test:
```

  (maps are shown one-by-one to show each move)

***** If there are color blocks, it is necessary to use `Cond Color Direction` instead of basic directions. See example program run of map3.txt in Point 6), and a simpler example below:***

- Using conditions

  Example

  ```
  > play
  First direction: Right
  Next direction: Cond Pink Down
  Next direction: Cond Yellow Right
  ```

- Using loops:

  Example

  ```
  > play
  First direction: Loop 3 Right Cond Pink Down
  Next direction: Cond Yellow Right
  ```

  ***** Please avoid using loop for testing `play`, as there may be bugs in some situations. I sincerely apologize for any inconvenience***

  ***** Loops work normally in computing optimal solution***

2.5 Play the game using function

If the player wants to use a function when proposing his/her own solution, the directions to be used in the function should be set before the game start. If the function to be used is right, up and down, to start the game, type:

```
play Right Up Down
```

then, when the player wants to use function as instruction, type `Function`:

```
Next direction: Function
```

*****Functions only support basic directions. Loops and conditionals are not acceptable***

***** See example program run of map2.txt in Point 6***

2.6 Record the number of bonus get

If the ball goes through the tile that contains bonus point, a message will be printed to announce it, like “`got the first bonus!`”

2.7 Error Checking

If the solution has errors or does not solve the game, some useful error messages should be reported, that give feedback on the problems for the current solution. (More details are described in point 4)

2.8 Quit the game

If the player does not want to play the game anymore, he/she can quit the game immediately by typing number 4.

A message "bye bye~" will be shown for success quit

Next time when run the game again, the previous record will NOT be saved and restored. So player needs to load the map again and restart the game.

3 Choice of data structures for representing maps, programs, and other parts

`board` is used to represent the map. It is the type of list of list of integer.

The first list in `board` represents the first row in the map, and the second list represents the second row, and so on.

For example, for the map1.txt provided, the board[0] is

`[1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1]`

The reason to use this type is that it is easy to find out what tile it is for a specific location, by just typing board[row number][col number]

`action` is used to represent the type of the next movement. For the basic movement, we will need only one direction. For conditionals, the first argument is color and second one is direction . For the loops, the first argument is the number of loops, and other two arguments are directions. For functions, the three directions needed to be inputted

The reason to use this data type is that it is similar to the approach in the project description file, so it is easier to check whether I implement them correct.

`solver` is using the A* searching algorithm to find shortest path. By using it, it is possible to calculate the distance between the ball and the target and to find which direction should be moved next to get them closer. The reason to use it is that it is popular to use in games for pathfinding and graph traversals.

4 Dealing with error cases and ending

- Error cases

  When a player proposes a solution, the program will check whether the solution is valid and whether it solves the game successfully.

If the solution can solve the game, then a winning message will be shown to the player, which is “Congratulations! You win the game!”

If the solution has errors or does not solve the game, then an error message will be shown to the player and give the reason why the solution fails. Also, the program will ask the player to try again to give a correct solution.
For example, if the next direction is left but there is obstacle on the left side, then a message “Sorry, error: cannot move to the Left” will be printed on the screen. Then the current board that does not make the left move will be shown to the player for referencing. After that, “Next direction: ”will be printed, the player can reattempt again by typing the correct solution next to the that statement.

- Ending
  If the player reaches the end, the game will be terminated if the player wins. It is not necessary that the player needs to get all the bonuses. That means, if the player can give a solution that allow the ball to roll from the start point to the end point, but he/she does not any bonus points, the program will still print the winning message and the game will be terminated.

5 Example of running the program

(shown in next page)

<Testcase 1, p.7: showing functionality 1-5, play using conditional>

<Testcase 2, p.15: showing functionality 5, play using function>

<TESTCASE1>

```
*Main> main
Welcome to play the Smeeborg game:

 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set of
instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 1
Please input the name of map file: map3.txt
******-------------------****
@----p--------------------***
*****-*******************-***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******y------b----t
*********-******-************
*********-******-************
 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set of
instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 2
The currently loaded map is solvable!
```

```
 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set of
instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 3
Solution: [Loop 3 (Base DRight) (Cond Pink DDown),Cond Yellow
DRight]
******-------------------****
-----@--------------------***
*****-*******************-***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******y------b----t
*********-******-************
*********-******-************

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

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

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

******-------------------****
-----p--------------------***
*****-*******************-***
*****-*******************-***
*****-bp***************---***
*******---------@-------*****
*********-******b************
*********-******-************
*********-******-************
```

```
*********-******-************
*********-******y------b----t
*********-******-************
*********-******-************

******-------------------****
-----p--------------------***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******@------b----t
*********-******-************
*********-******-************

******-------------------****
-----p--------------------***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******y------b----@
*********-******-************
*********-******-************
```

```
 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set of
instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 4
Byebye~
*Main>

*Main> main
Welcome to play the Smeeborg game:

 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set of
instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 1
Please input the name of map file: map3.txt
******-------------------****
@----p--------------------***
*****-*******************-***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******y------b----t
*********-******-************
*********-******-************
 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set of
instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 5
> play
First direction: Right
Next  direction: Cond Pink Down
Next  direction: Right
Next  direction: Cond Pink Down
Next  direction: Right
Next  direction: Cond Pink Down
Next  direction: Cond Yellow Right
Next  direction:
test:
******-------------------****
-----@--------------------***
*****-*******************-***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******y------b----t
*********-******-************
*********-******-************

******-------------------****
-----p--------------------***
*****-*******************-***
*****-*******************-***
*****@bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
```

```
*********-******y------b----t
*********-******-************
*********-******-************

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

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

******-------------------****
-----p--------------------***
*****-*******************-***
*****-bp***************---***
*******---------@-------*****
```

```
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******y------b----t
*********-******-************
*********-******-************

******-------------------****
-----p--------------------***
*****-*******************-***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******@------b----t
*********-******-************
*********-******-************

******-------------------****
-----p--------------------***
*****-*******************-***
*****-*******************-***
*****-bp***************---***
*******---------p-------*****
*********-******b************
*********-******-************
*********-******-************
*********-******-************
*********-******y------b----@
*********-******-************
*********-******-************
```

<TESTCASE2>

```
 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set
of instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 1
Please input the name of map file: map2.txt
*****-------------------*****
*****------------------b*****
*****-*****************-*****
********----*****----********
*****-**----*****----**-*****
*****-**----*****----**-*****
*****-******--b--******-*****
*****-******-***-******-*****
@-----******-***-******y----t
*****-******-***-******-*****
*****---b----***--------*****
*****************************
*****************************
 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set
of instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 5
> play Function
First direction:
 1) Load file: Loads a map.
 2) Check: Checks if the currently loaded map is solvable.
 3) Solve: Solves the map and presents the solution as a set
```

```
of instructions.
 4) Quit: Quits the game.
 5) Play: Allows the player to propose a solution as a set of
instructions.
 Select 1 ~ 5:
> 5
> play Right Down Right
First direction: Function
Next  direction: Up
Next  direction: Right
Next  direction: Down
Next  direction: Right
Next  direction: Up
Next  direction: Cond Yellow Right
Next  direction:
test:
*****-------------------*****
*****------------------b*****
*****-*****************-*****
********----*****----********
*****-**----*****----**-*****
*****-**----*****----**-*****
*****-******--b--******-*****
*****-******-***-******-*****
-----@******-***-******y----t
*****-******-***-******-*****
*****---b----***--------*****
*****************************
*****************************

*****-------------------*****
*****------------------b*****
*****-*****************-*****
********----*****----********
*****-**----*****----**-*****
*****-**----*****----**-*****
*****-******--b--******-*****
*****-******-***-******-*****
```

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

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

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

*****-------------------*****
*****------------------b*****
*****-*****************-*****
********----*****----********
```

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

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

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

```
*****-------------------*****
*****------------------b*****
*****-*****************-*****
********----*****----********
*****-**----*****----**-*****
*****-**----*****----**-*****
*****-******--b--******-*****
*****-******-***-******-*****
------******-***-******@----t
*****-******-***-******-*****
*****---b----***--------*****
*****************************
*****************************

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

Congratulations! You win the game!
```