Final Report

Name: Lin Jianan

UID: 3035577595

Contents

[1. Build the project: 3](#_Toc58916686)

[2. Functionality of my program 3](#_Toc58916687)

[2.1 Load file 3](#_Toc58916688)

[2.2 Directions 4](#_Toc58916689)

[2.3 Condition 4](#_Toc58916690)

[2.4 Loops 5](#_Toc58916691)

[2.5 Function 5](#_Toc58916692)

[2.5 Hint 6](#_Toc58916693)

[2.6 Check 6](#_Toc58916694)

[2.7 Solve 7](#_Toc58916695)

[2.8 Show bonus got 8](#_Toc58916696)

[2.9 Command grammar validation 8](#_Toc58916697)

[2.10 Quit the game 9](#_Toc58916698)

[2.11 Generate map 9](#_Toc58916699)

[3. Data Structures 10](#_Toc58916700)

[4. Deals with error cases 10](#_Toc58916701)

[5. Deals with ending. 10](#_Toc58916702)

[6. Additional Features 11](#_Toc58916703)

[6.1 Feature 1: Generate a solvable map 11](#_Toc58916704)

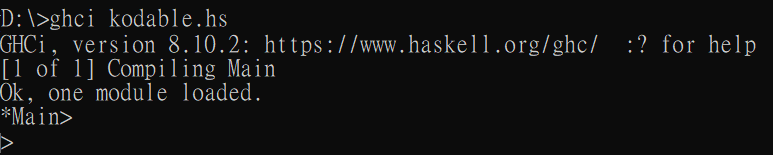
[6.2 Feature 2: Hint 11](#_Toc58916705)

# Build the project:

Method 1: type ghc kodable.hs

Method 2: type “ghci kodable.hs” then type “main” to start the program.





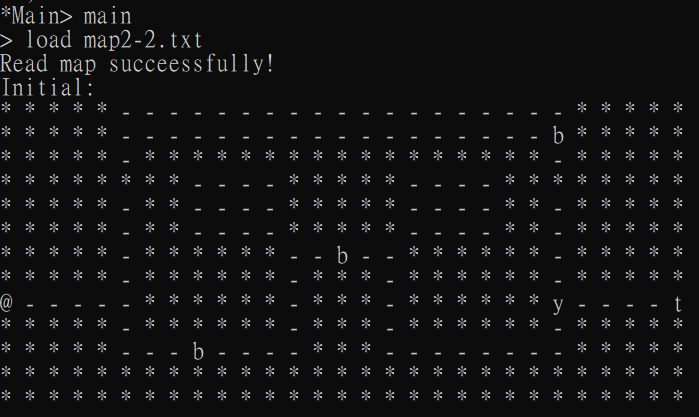
# Functionality of my program

All the input below is case-sensitive.

## 2.1 Load file

\*Main> main

> load map2-2.txt



## Directions

Attention: input is case-sensitive!

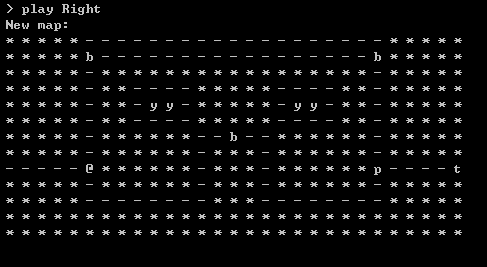
> play Up

> play Down

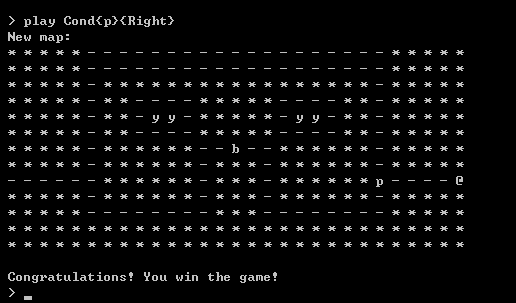
> play Right

> play Left

> play Cond{p}{Up}



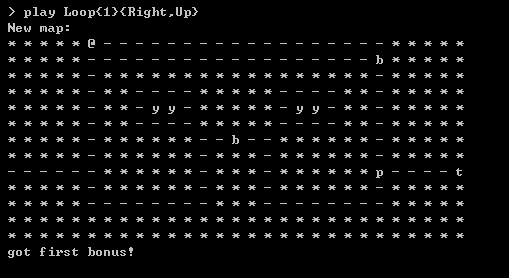
## Condition



## Loops

Loop action takes exactly 2 directions:

> play Loop{2}{Right,Down}

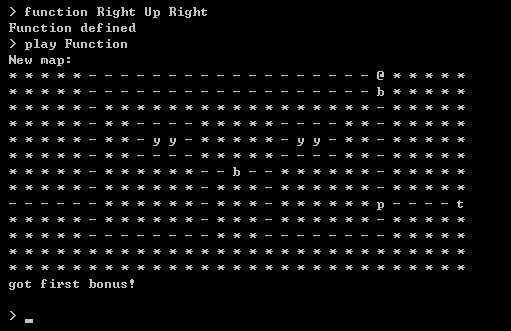


## Function

Actions of function is not limited.

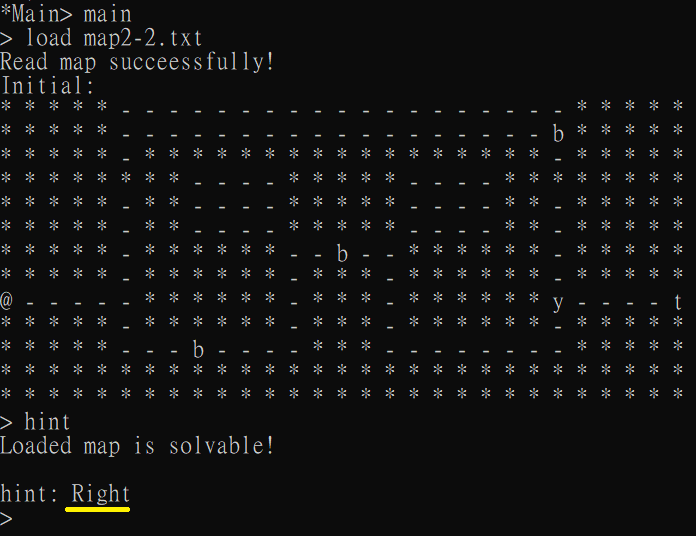
> function Up Right Loop{3}{Up,Down}

> play Function



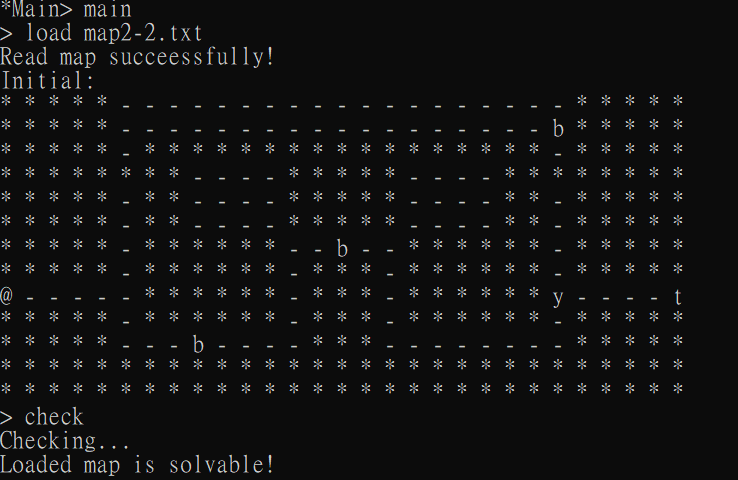
## 2.5 Hint

> hint



## 2.6 Check

> check

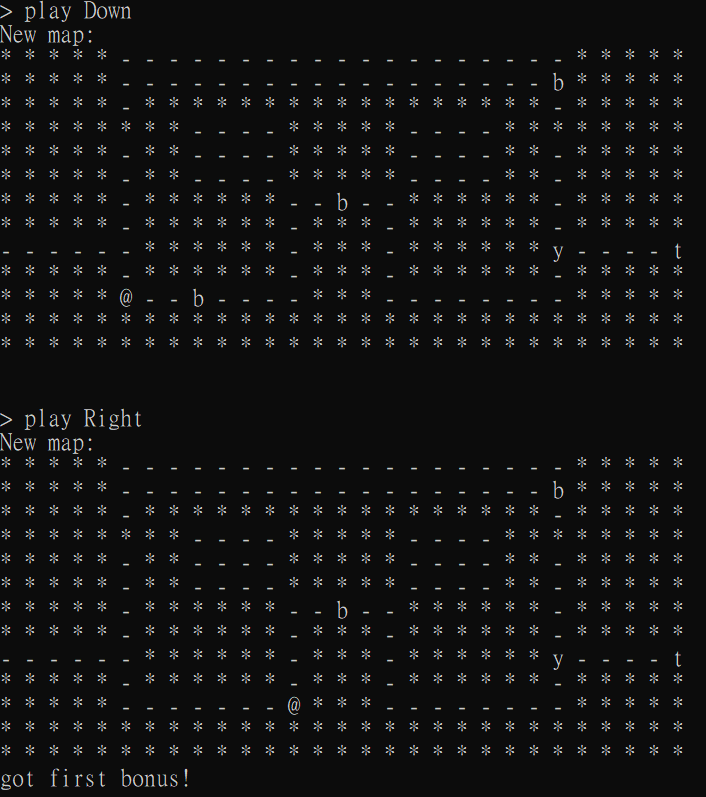


## 2.7 Solve

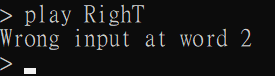
> solve



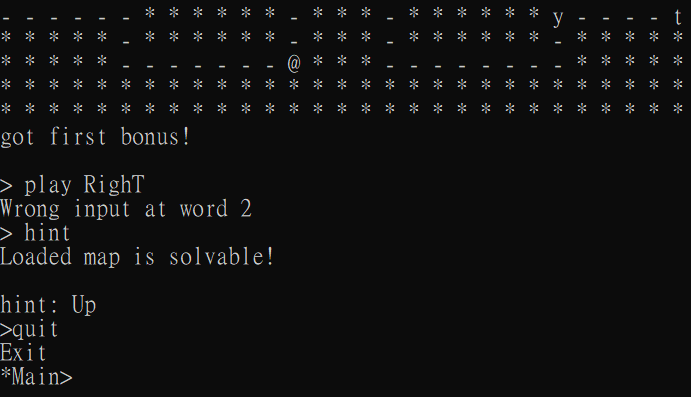
## Show bonus got



## Command grammar validation



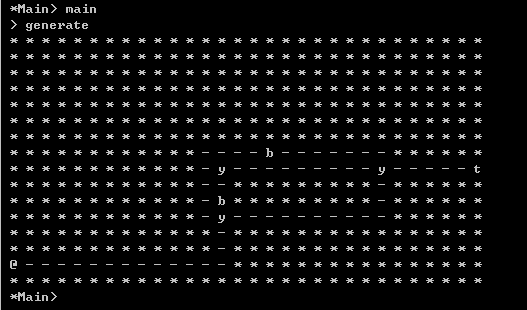
## 2.10 Quit the game



## 2.11 Generate map

Will exit after generation and player cannot continue to play

> generate



## Data Structures

Data declaration:

data Mapattr =

Mapattr { bonus :: Int, color :: Char, win :: Bool, dir :: [String], fun :: [String] } deriving (Show,Eq)

Color: colour blocks.

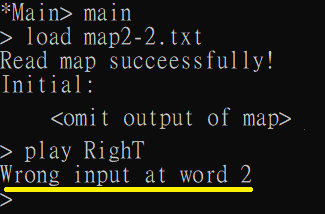
Bonus: Record number of bonus the player gets.

dir: Record the path walked by the player on the map.

Win: Store status of player: he wins or not.

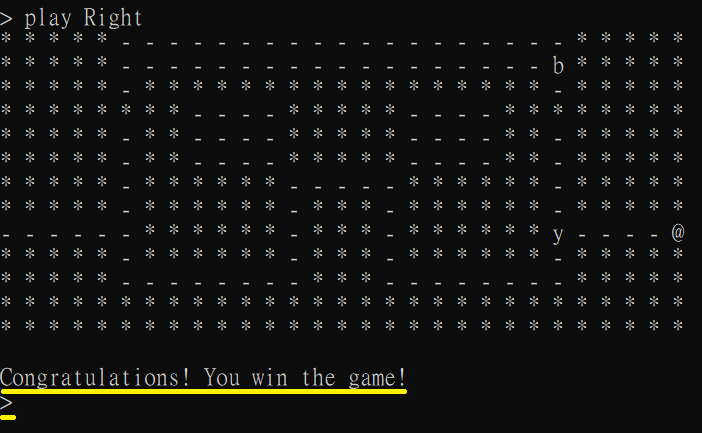
Fun: function, store Function actions defined by player.

## Deals with error cases



## Deals with ending.

Message: “Congratulations! You win the game!” will be displayed if player reaches the target.



Player can type “load map.txt” or “generate” to start new game immediately.

## Additional Features

### 6.1 Feature 1: Generate a solvable map

Step 1: Draw a zig-zag path.

Step 2: Draw a square-shaped path

Step 3: Add symbol ‘@’ and ‘t’ at starting point and ending point respectively.

Step 4: Add colour blocks at road with multiple branches.

Step 5: Add bonus on road.

### 6.2 Feature 2: Hint

Step 1: Generate solution of map based on current position of player.

Step 2: Display next step to go to target.

