

Functional Programming Assignment Element 2 Resit

07/20

Stefan M Ahmed 21359035

Intro

For a text-adventure gaming system I can think of rooms like a piano room, dance hall, laundry, dry-cleaning room, library and a drama room. The player goes in directions such as north, south, northwest to these rooms to find a box of vinyl lps. The goal of the game is to find the box of lps. The player can start playing the game by typing 'Start', type in commands such as 'Check' to see what is in the place they are in or 'Leave Game' to leave the game.

```

1  -- =====
2  -- Text adventure game
3  -- =====
4
5
6  data Entrance = Entrance String String -- two strings for two pieces of info displayed separately
7
8
9  data Place = Place String String [Entrance]
10
11 data playBuilding = playBuilding [Place]
12
13 -- study - toilet - sitting room - acting room-dance hall
14 -- | / | / \ / | /
15 -- library - dining area - gaming room - piano room
16

```

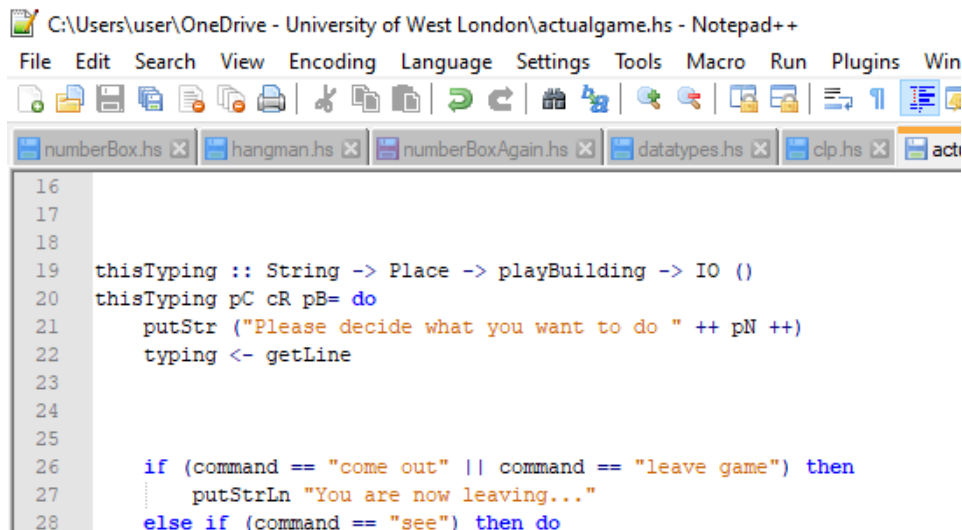
The coding I wrote to make the game run begins declaring variables. The variables Entrance and Place which are the room and entrance to each to it, need two strings. One to show the way to the place and another the name of the place.

```

93
94 start :: IO ()
95 start = do
96   -- make the building game is played in--
97
98   let study = Place "study" "You are in the study. It has chairs, desks, computers and books." [(Entrance "south" "library" ),(Entrance "east" "toilet")]
99   let library = Place "library" "This is the library. There are shelves, full of books, computers, telephone and tables to sit at. Maybe the box of vinyl lps are
100   let toilet = Place "toilet" "The box of lps can't be in here." [(Entrance "west" "study"),(Entrance "south-west" "library"),(Entrance "south" "dining area"),(Entrance "north" "toilet")]
101   let dining area = Place "dining area" "There is a candle holder on the dining table, a cabinet, pictures and a sideboard." [(Entrance "west" "library"),(Entrance "east" "toilet"),(Entrance "south-west" "library"),(Entrance "north" "toilet")]
102   let sitting room = Place "sitting room" "For sitting, relaxing, deciding or waiting." [(Entrance "south-west" "dining area"),(Entrance "west" "toilet"),(Entrance "north" "toilet")]
103   let gaming room = Place "gaming room" "Play all sorts of games here. You have found the lps." [(Entrance "west" "dining area"),(Entrance "north-west" "sitting room"),(Entrance "south-west" "library"),(Entrance "north" "toilet")]
104   let acting room = Place "acting room" "Acting lessons take place here before the live performance." [(Entrance "west" "sitting room"),(Entrance "south-west" "library"),(Entrance "north" "toilet")]
105   let piano room = Place "piano room" "You can chill out here to music being played on the piano. Is the box of lps in here?" [(Entrance "west" "gaming room"),(Entrance "north" "toilet")]
106   let dance hall = Place "dance hall" "The dancing takes place here if it is a wedding, dance class, competition or party." [(Entrance "west" "acting room"),(Entrance "north" "toilet")]
107
108   let building = playBuilding [study, library, toilet, dining area, sitting room, gaming room, acting room, piano room, dance hall]
109
110   -- start game
111   putStrLn "Please type in the name you would like to be called: "
112   playerName <- getLine
113   putStrLn ("Hi" ++ playerName ++ ". Welcome to this building. You have today to find the box of LPs to be sold in the record shop you work in. Are you ready?")
114
115   let startingArea = study
116

```

Here I've defined "start" with IO with parentheses to make the input work. This makes the user able to type "start" to start playing the game. This is by completing its definition with the common line of code "= do" to run the game when the player types "start".



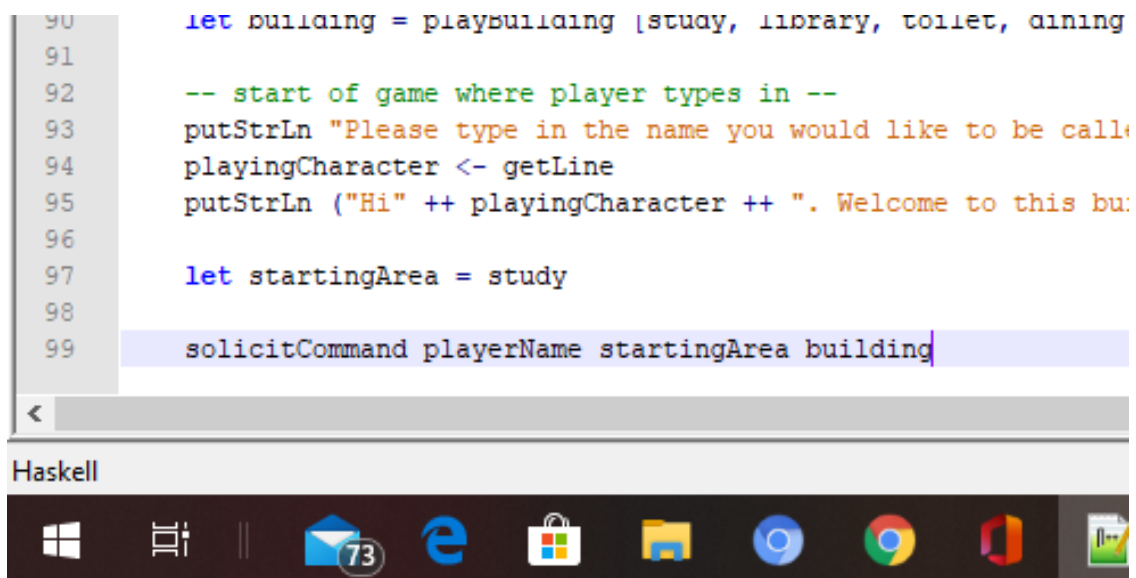
C:\Users\user\OneDrive - University of West London\actualgame.hs - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Win

numberBox.hs hangman.hs numberBoxAgain.hs datatypes.hs clp.hs acti

```
16
17
18
19 thisTyping :: String -> Place -> playBuilding -> IO ()
20 thisTyping pC cR pB= do
21     putStr ("Please decide what you want to do " ++ pN ++)
22     typing <- getLine
23
24
25
26     if (command == "come out" || command == "leave game") then
27         putStrLn "You are now leaving..."
28     else if (command == "see") then do
```

The code makes the commands being typed in by the player work. The command being typed known as “typing” has getLine with an arrow to show the telling the user to decide what to do.



```
90     let building = playsbuilding [study, library, toilet, dining
91
92     -- start of game where player types in --
93     putStrLn "Please type in the name you would like to be call
94     playingCharacter <- getLine
95     putStrLn ("Hi" ++ playingCharacter ++ ". Welcome to this bu
96
97     let startingArea = study
98
99     solicitCommand playerName startingArea building
```

Haskell

Windows taskbar icons: Start, Task View, Mail (73), Edge, Store, File Explorer, Docker, Chrome, VS Code, and a game icon.

