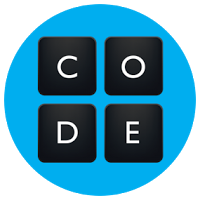
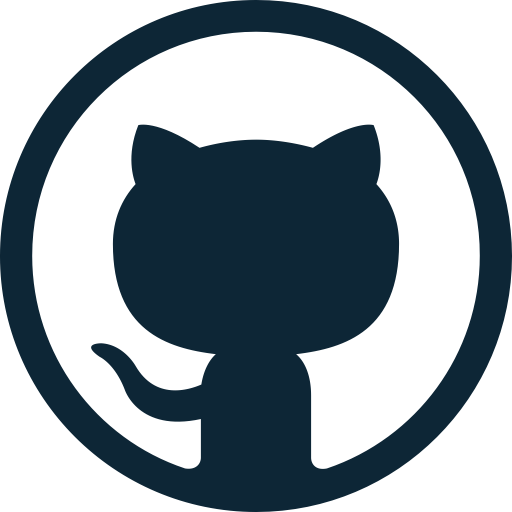
Software Engineering



Assignment 3



Patrick Lӧwe & Mark Hartnett

Contents

1. Specification 3
2. User Manual 4
3. Structured Programming Chart 10
4. Data Dictionary 11
5. Test Case Dataset 12
6. Development Log 13
7. Code 14
8. References 30
9. Declaration of Authorship 30

## GitHub: https://github.com/patricklowe/Assignment\_3

# Specification

## Introduction

Using C we created a program that would accept between 2 & 6 players, allowing each user to create a Name, Type and then be assigned to randomly generated slots. Once assigned each player can then choose to:

* 1) Move to an adjacent slot
* 2) Attack another player
* 3) Quit the game

Moving

If the current player decides to move then can only move to an adjacent slot, either up, down, left or right, if the slot exists. They are permitted to occupy slots with other players too.

Attacking

The current player can select to perform a nearby attack, a distant attack, or a magic attack

Quitting

If the player quits the game they are no longer able to play in rounds. If all the players have quit or died, then the end game results are displayed and the program closes.

User Manual

Step 1 – Player Creation

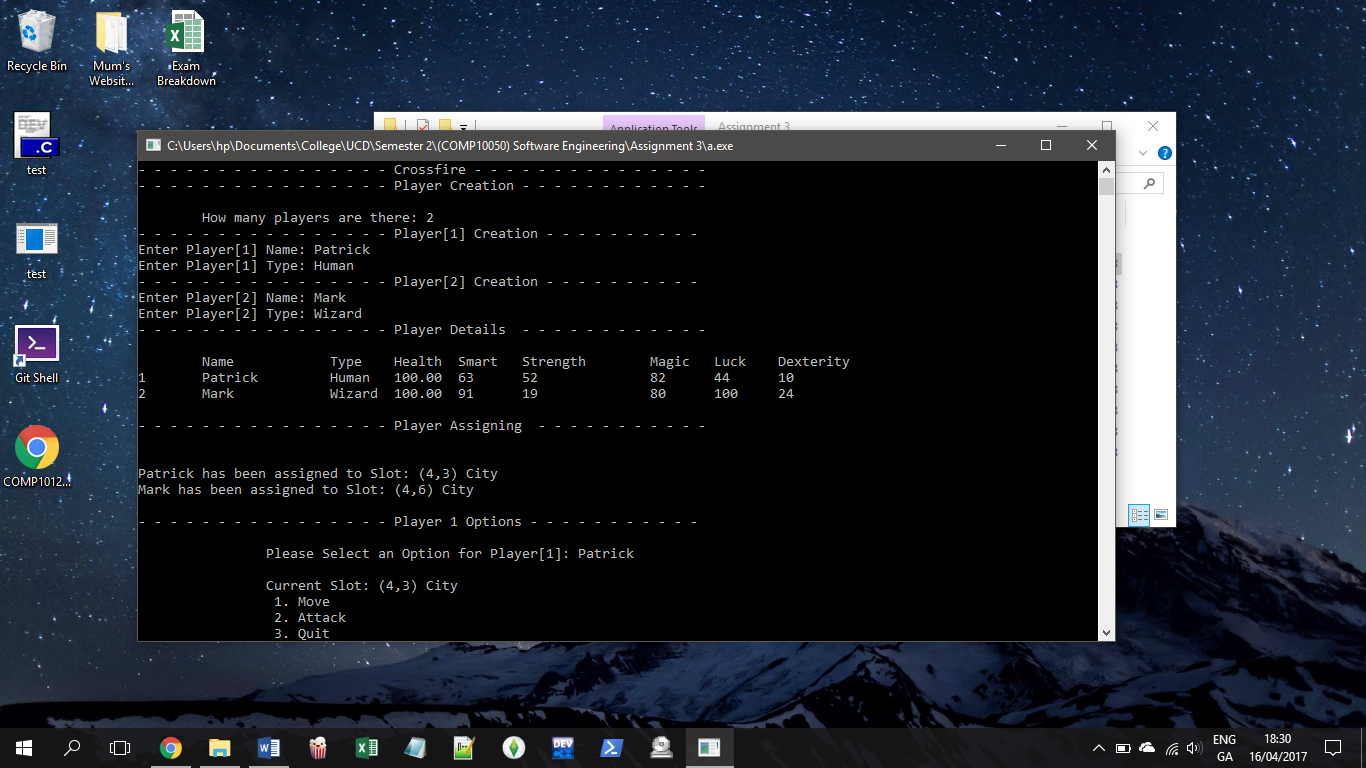


Fig.1 Player Limit & Creation

The program asks the user to input player numbers. If less than 2 it with automatically begin the game with 2 players. If more than 6 players are entered, it will automatically start with 6 players. (Fig.1)

Step 2 – Player Details

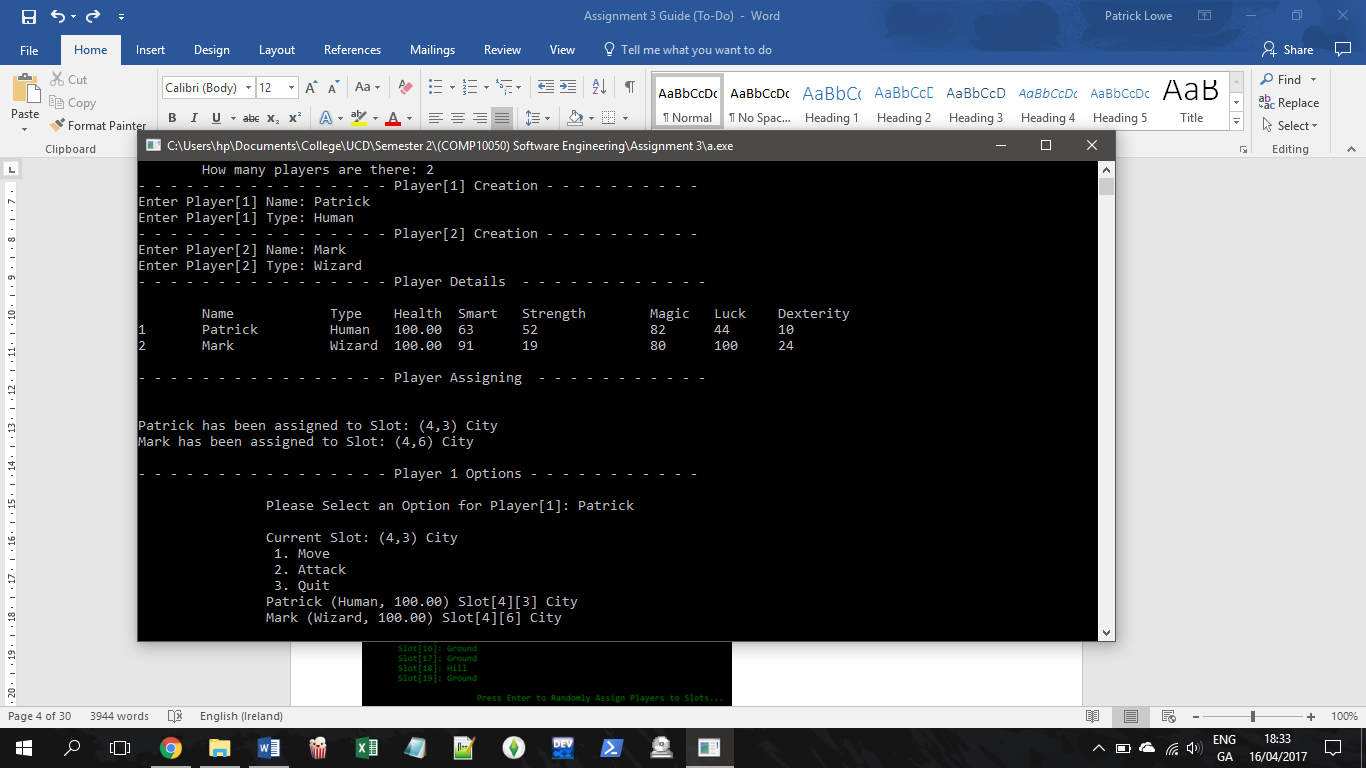


Fig.2 List of Player Details

After the players names and types have been accepted the program will autogenerate random values for their attributes, depending on character Type (Fig.2)

Step 3 – Player Assigning

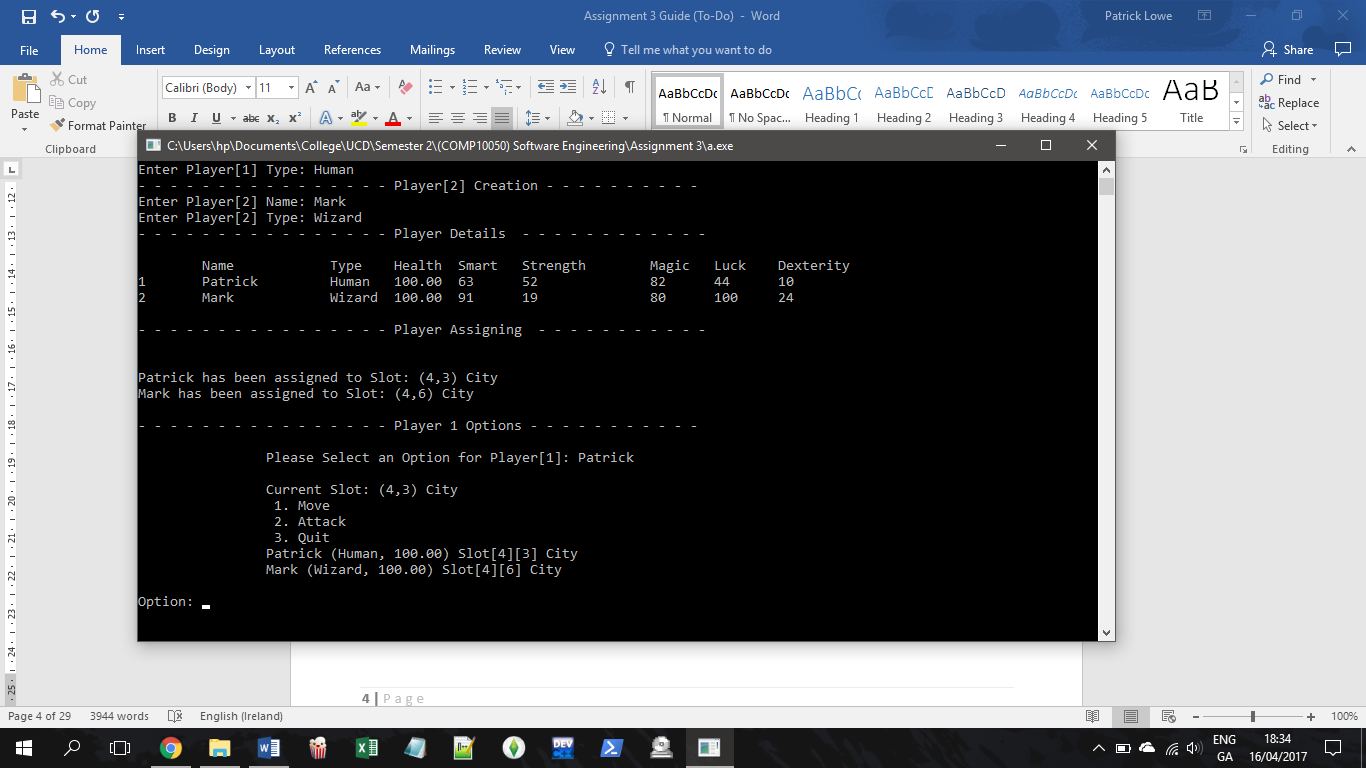


Fig.3 Board is created & players are randomly assigned slots

The program creates a 7 by 7 board and assigns level types to each (ranging from 0,0 to 6,6, and level types are Ground, City, or Hill). Their attributes are then adjusted depending on what slot they are in.

Step 4 – Player Options

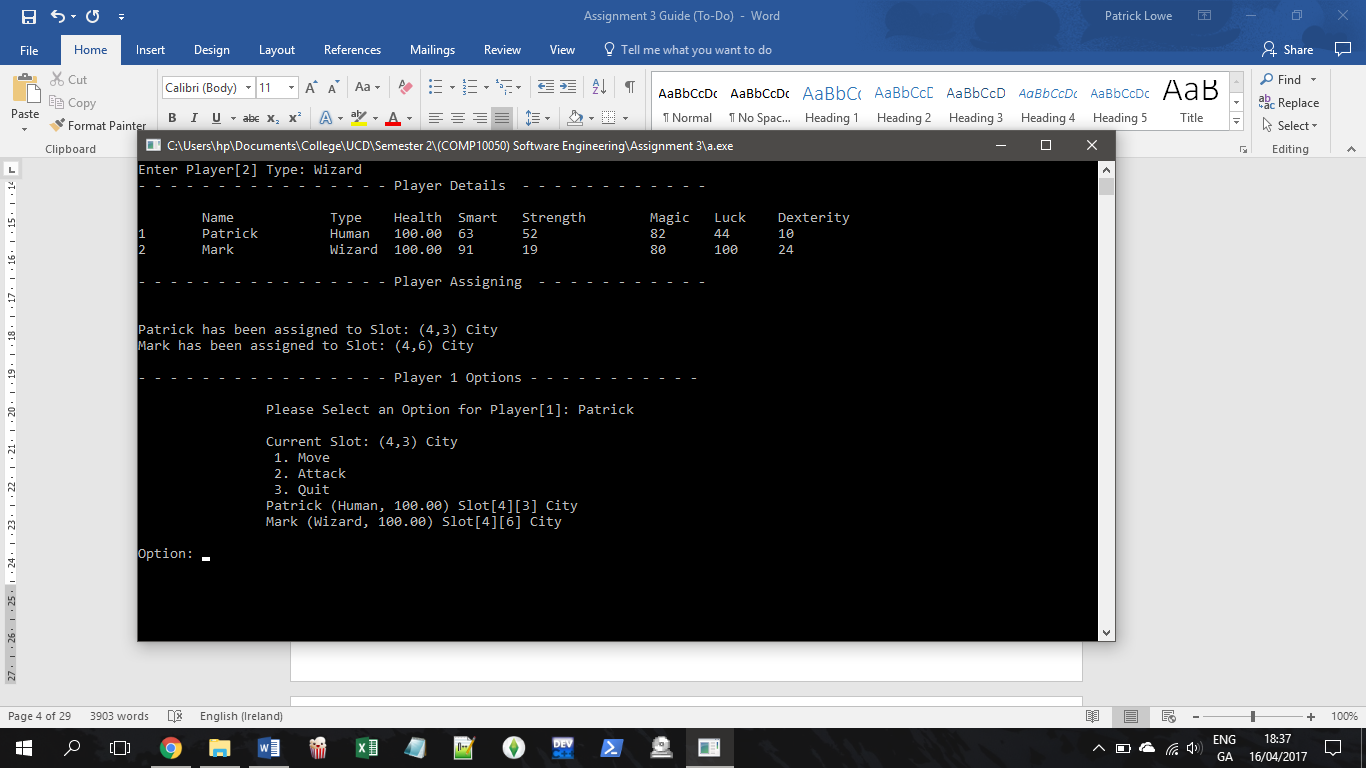


Fig.4 Player 1 Options

Users will be shown their name and options based on their current position (Fig.4). The details of all players will be displayed before an option is selected. The current player has the option to Move, Attack, or Quit the game.

Step 7 – Moving

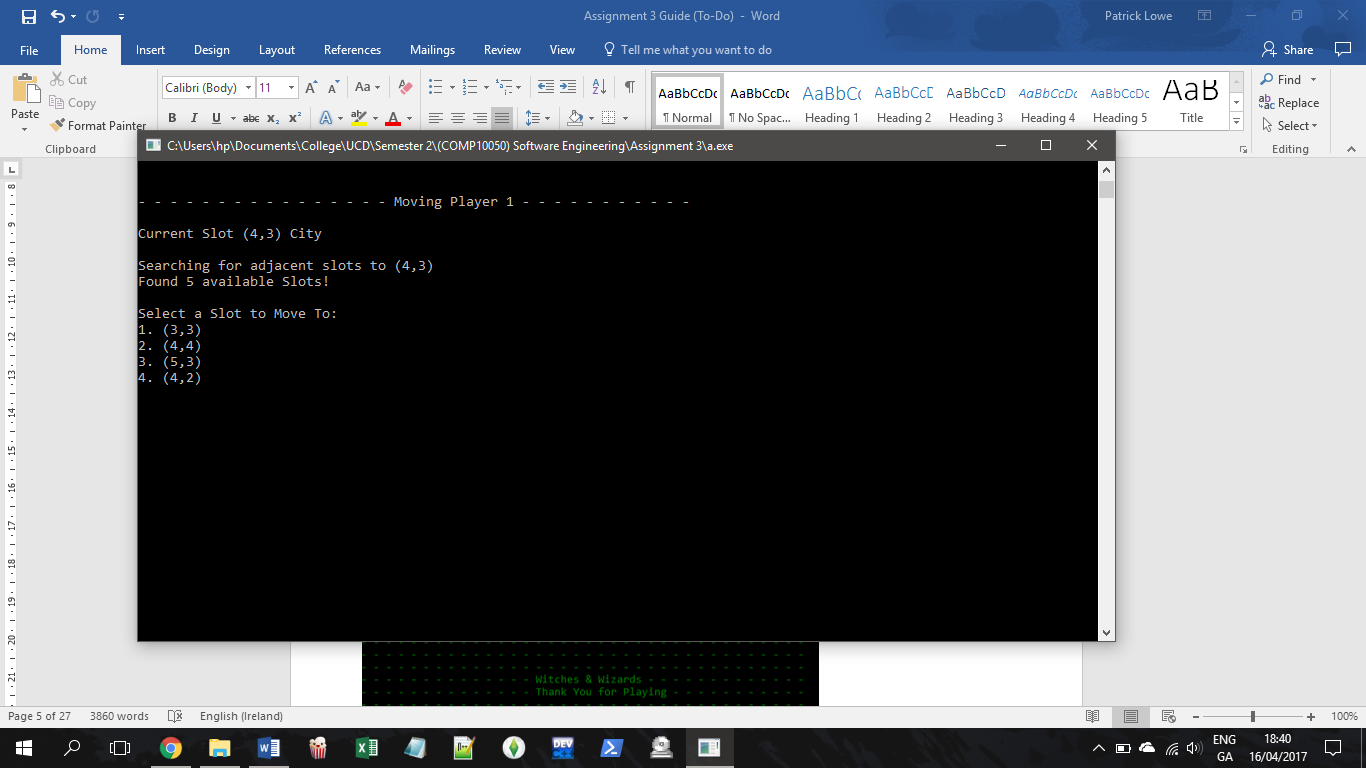


Fig.5 Choosing a slot

If the current player selects to move slot the program will search for adjacent slots. Based on their current position it will start the search in either the upper left, upper right, lower left, or lower right quarter; whichever is closer to the players position. It will then display available slots to move to. Selecing one will move the player, and then adjust their attributes depending on the slot type. (Fig.5 & 6)

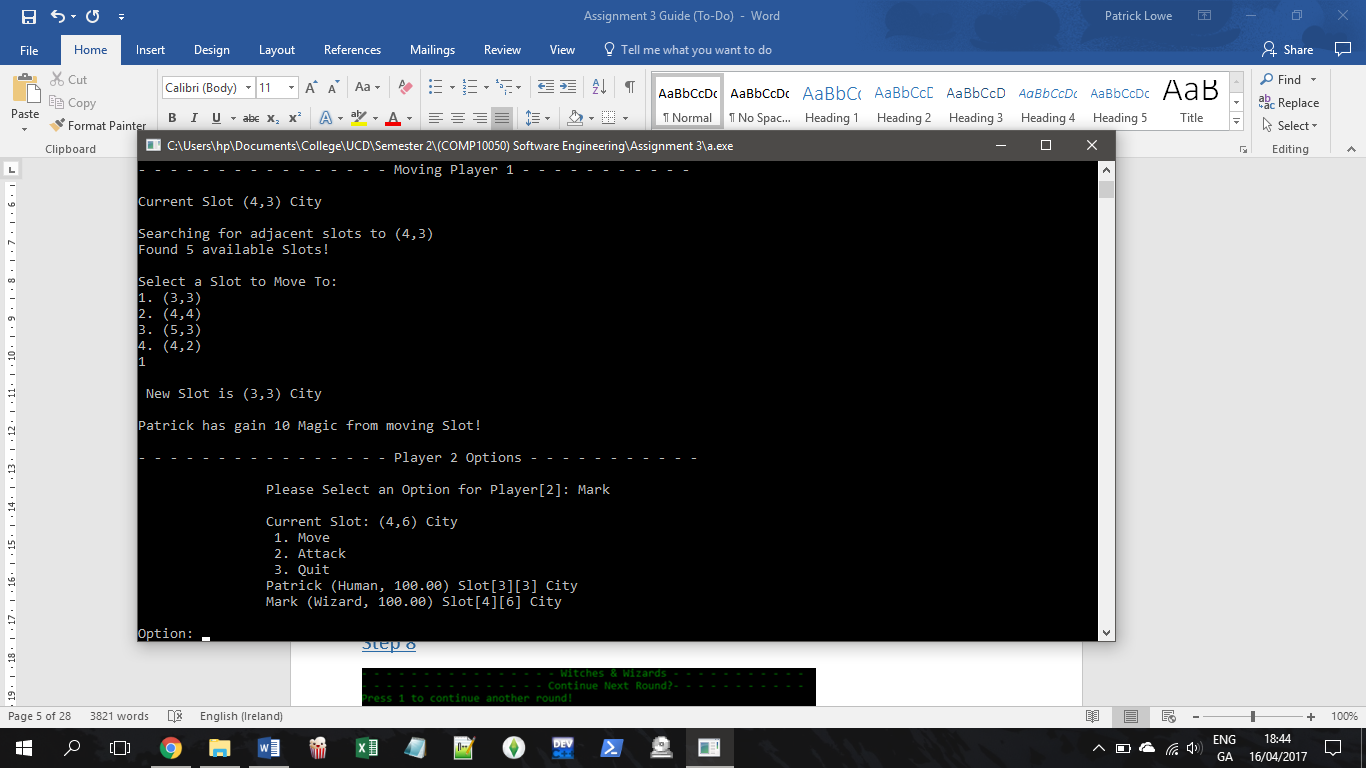


Fig.6 Slot Chosen

Step 8 – Attacking

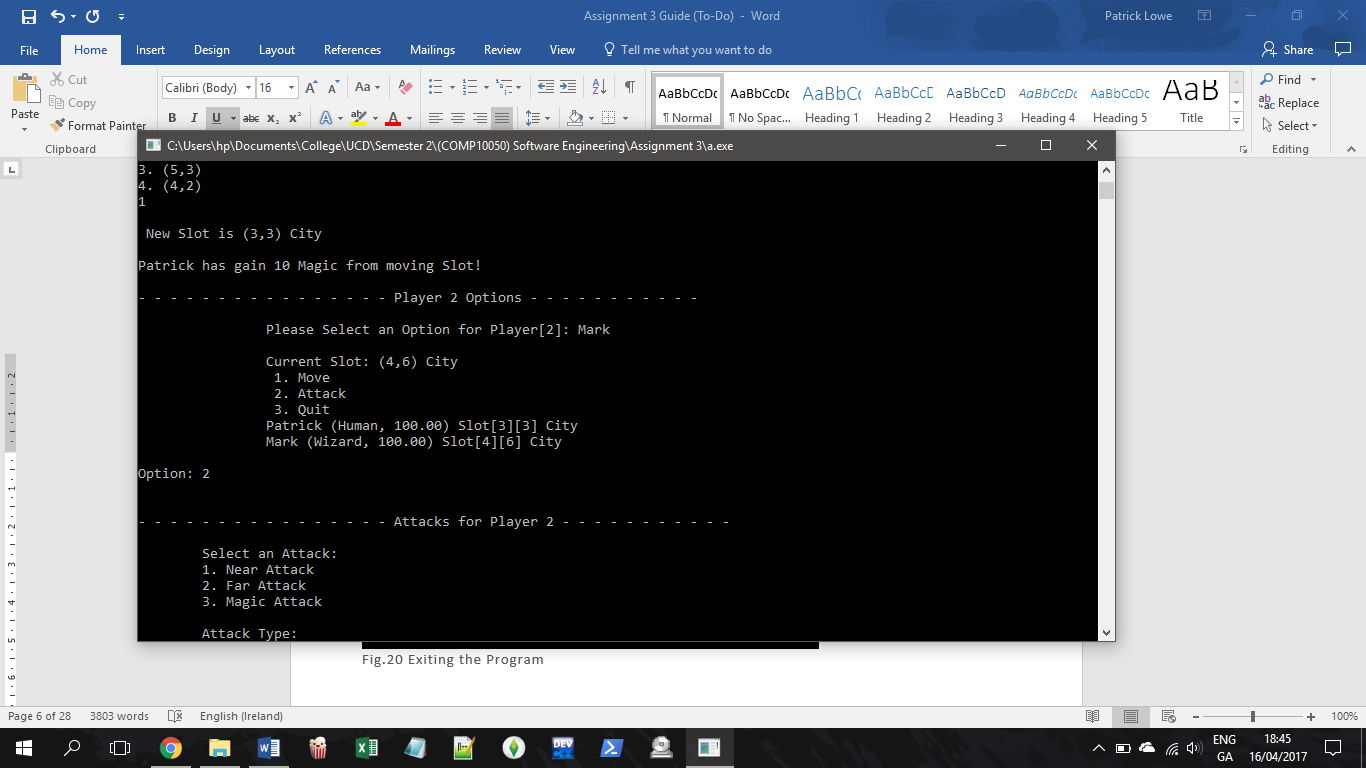


Fig.7 Attack Menu

When the attack menu is loaded, there are 3 possible choices: Near Attack, Far Attack, or Magic Attack. (Fig.7)

#### Near Attack

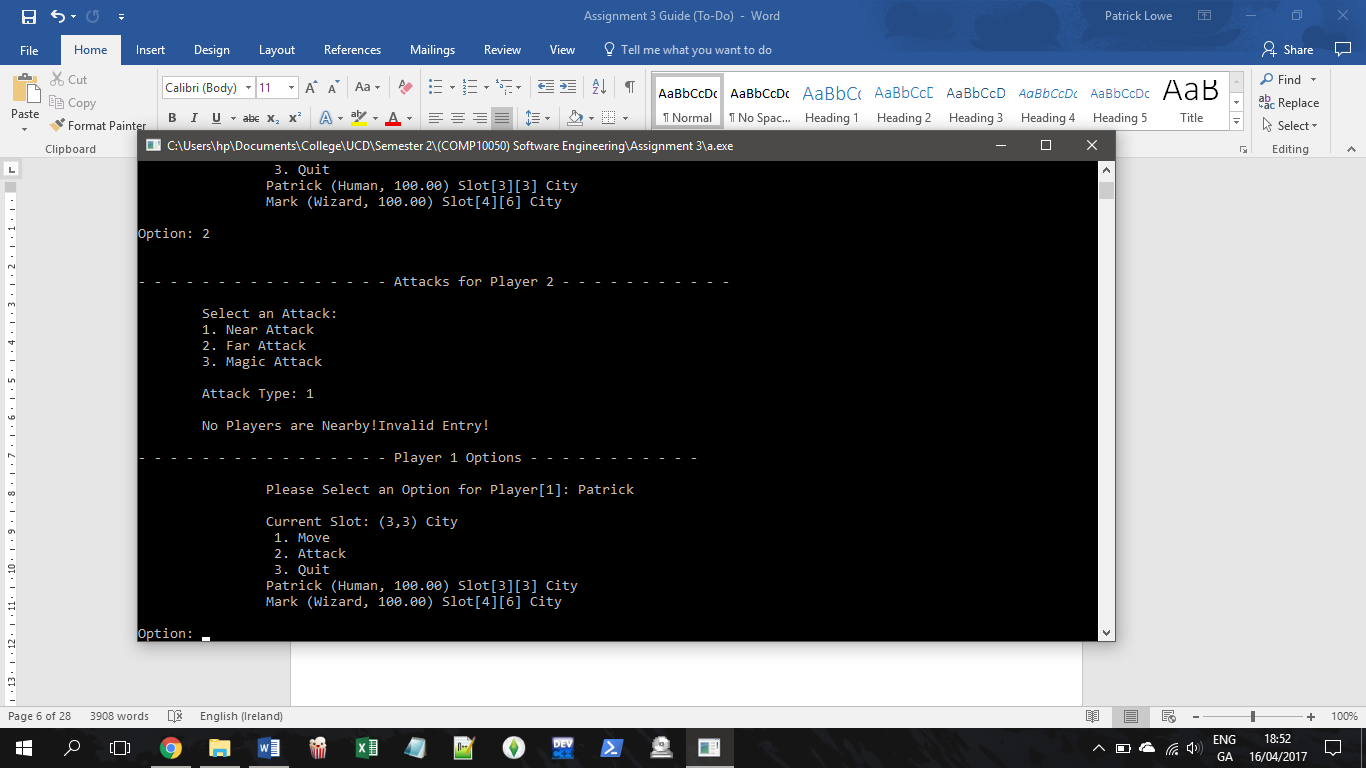


Fig.8 Nearby Attack with no nearby players

The program will search adjacent slots to the current players position. If another player is nearby they will be able to be attacked (Fig.9). If not, an error will be displayed and that round is essentially wasted (Fig.8)

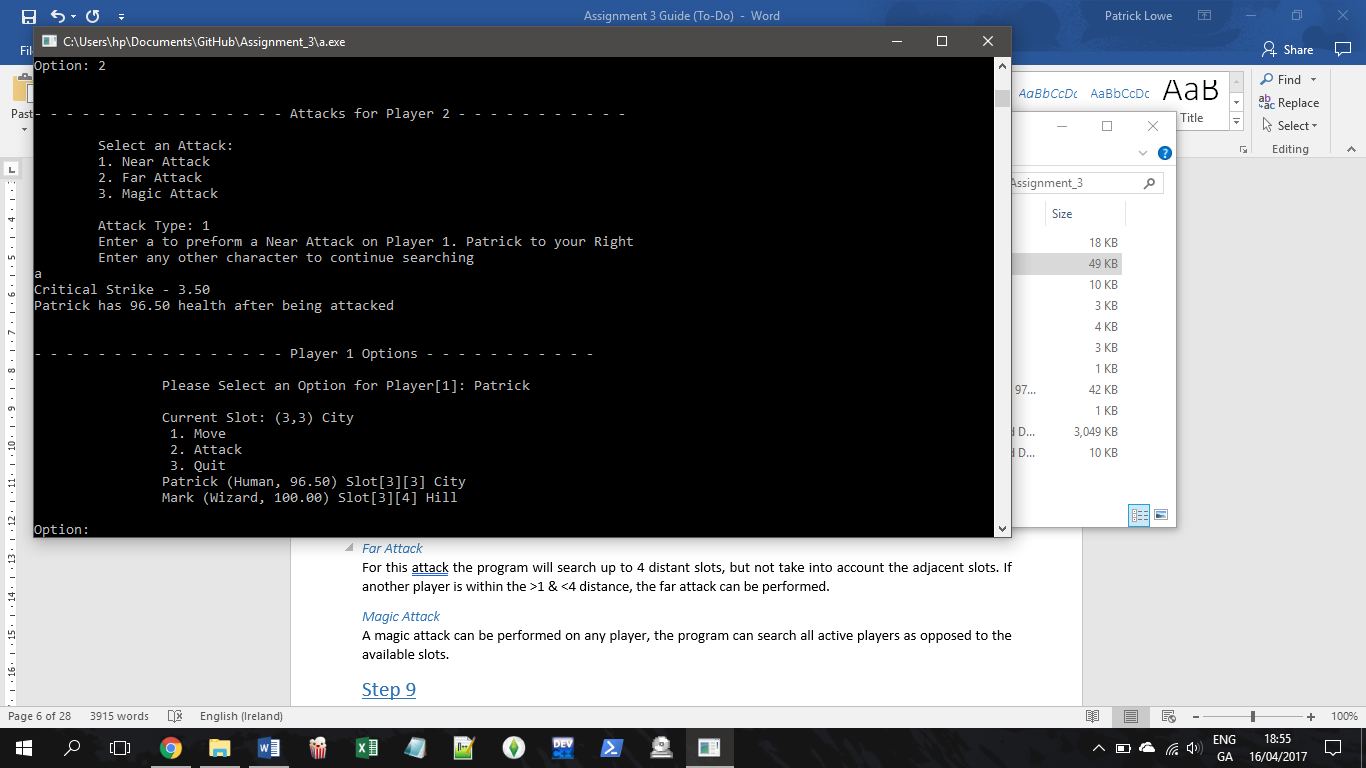


Fig.8 Nearby Attack with nearby player.

#### Far Attack

For this attack the program will search up to 4 distant slots, but not take into account the adjacent slots. If another player is within the >1 & <4 distance, the far attack can be performed.

#### Magic Attack

A magic attack can be performed on any player, the program can search all active players as opposed to the available slots.

Step 9 – Quitting

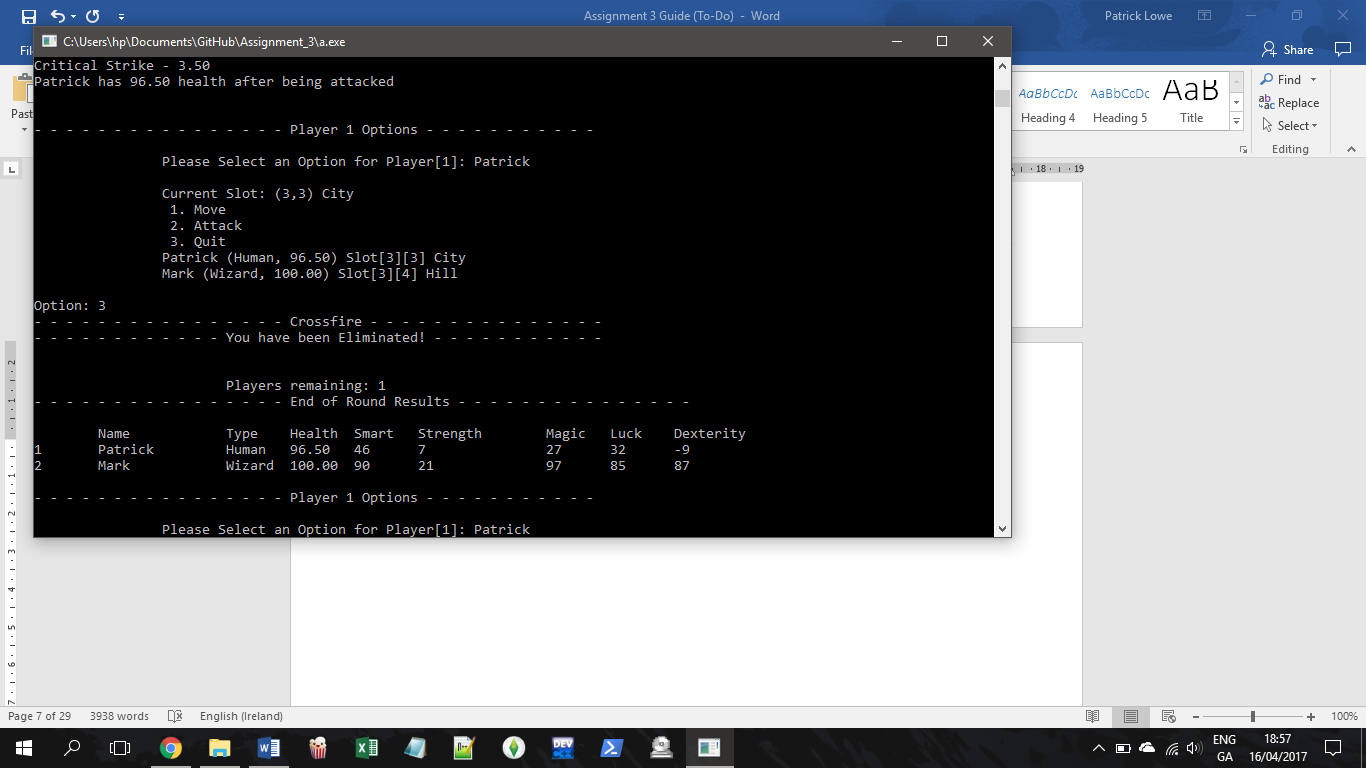


Fig.?? Player has quit the game

JSP Structured Programming Chart

A.

Player Limit

B.

Name

J.

Board Creation

M.

Slot Type

L.

Slot Num

D.

Health

C.

Type

O.

Options

F.

Smart

N.

Assign to Player

E.

Strength

H.

Dexterity

G.

Magic

\

S.

Quit Game

Q.

Changes attributes

P.

Move Slot

R.

Attack Player

I.

Luck

U.

Far Attack

T.

Nearby Attack

V.

Magic Attack

Data Dictionary

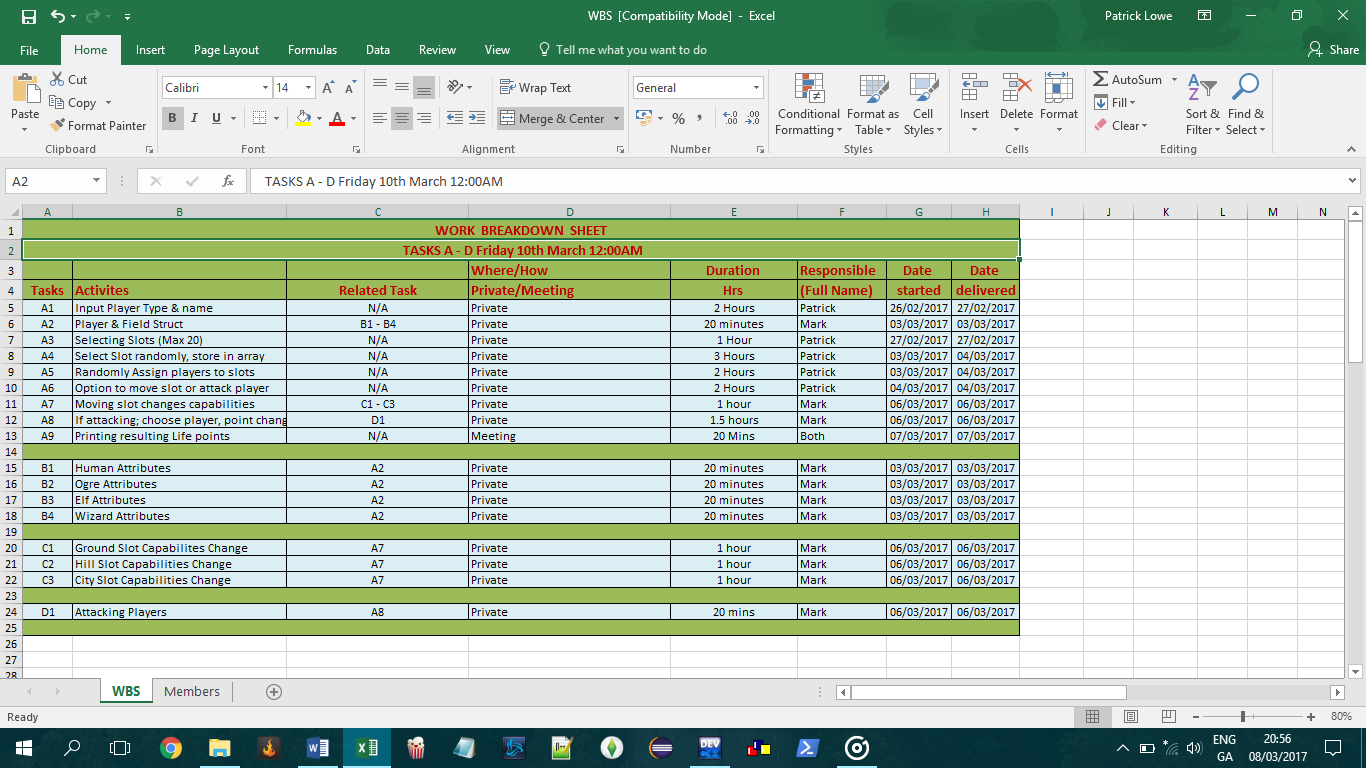
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***main.c*** | | | | |
| **Physical Name** | **Logical Name** | **Format** | **Purpose** | **Test Case Dataset Ref** |
| pName | Character Name | String | Store character Name | Page 12 |
| pType | Character Type | String | Store Characters Type | Page 12 |
| Lifepoints | Health | Float | Displaying Health to 2 decimal places | Page 12 |
| Smart | Attribute | integer | Store Attribute | Page 12 |
| Strength | Attribute | Integer | Store Attribute | Page 12 |
| Magic | Attribute | Integer | Store Attribute | Page 12 |
| Luck | Attribute | Integer | Store Attribute | Page 12 |
| Dexterity | Attribute | Integer | Store Attribute | Page 12 |
| pSlotNum | Character Slot Number | Integer | Store Player Position | Page 12 |
| pSlotType | Character Slot Type | String | Store Position Type | Page 12 |
|  |  |  |  | Page 12 |
|  |  |  |  | Page 12 |
|  |  |  |  | Page 12 |
| playerLimit | Number of Players | Integer | Correction of players (Min/Max) | Page 12 |
|  |  |  |  | Page 12 |

Test Case Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum Physical name** | **Data Dictionary Reference** | **Input** | **Result** | **Reason** |
| pName | Page 11 | Patrick | Valid | Correct Format |
|  | Page 11 | 8 | Invalid | Integer |
| pType | Page 11 | Human | Valid | Correct Type |
|  | Page 11 | Warlock | Invalid | Incorrect Type |
| Password | Page 11 | ‘login’ | Valid | Correct Password |
|  | Page 11 | Anything else | Invalid | Incorrect Password |
| playerLimit | Page 11 | 2-6 | Valid | Within bounds |
|  | Page 11 | <2 or >6 | Invalid | Out of Bounds |
| numSlots | Page 11 | 2-20 | Valid | Within Limits |
|  | Page 11 | <2 or >20 | Invalid | Out of Bounds |

Development Log

Note: Work Breakdown Sheet attached with Submission



UML Diagram

Begin

Quit Game

Next Player

New Round Or Exit

Attack Player

Move Slot

Slot Assigned

Options Menu

Failed

Enter Limit

Player Details

Auto set within Limit

Code

References

#### Cboard.cprogramming.com

#### CProgramming.language-tutorial.com

#### StackOverflow.com

#### Tutorialspoint.com

Declaration of Authorship

14 – April – 2017

Patrick Löwe Mark Hartnett