

Assignment One REPORT

Due: Monday, 20th September 2021, 11:59 PM (Extension Approved)

Weight: 15% of the unit

Name: Sohailharoon Bakhshi

ID: 20605126

3.3 Metadata and Map creation

In this section we were meant to generate the 2d char map array. Within generateMap.c I've created two functions. One that grabs the metadata and one that prints the maze.

Function One:

```
char** createMap(int*row, int*col, int objective[], int user[]) /*function to retrieve data from map.c*/
```

Function one grabs the metadata. Within the function I have listed variables and declared two double pointers one for storing the table from getMetadata() and the other for storing the maze. Within the function the getMetadata function is called which extract the table from map.c and store it in my variables. These being my arguments.

```
getMetadata(&data, &amount, &Row, &Col);
```

The address of data, amount, row and col are passed to the function. Pointers are used to be sent to the main function. Malloc has been used to allocate more memory and for loops are used to initialise the maze map by setting up borders, walls, objectives and the initial screen at which the game starts.

Function two:

```
/*function to print maze onto the screen*/  
void displayMaze(char** maze, int Row, int Col, int user[]){
```

This function uses a char double pointer and 3 integer parameters which are called in the main function and basically prints the maze by using a for loop.

3.4 Main Interface

By using `system("clear");` every time the user inputs a command the last printed screen will be cleared.

3.5 User Input

Within commands.c the function for the character controls was called getInput.

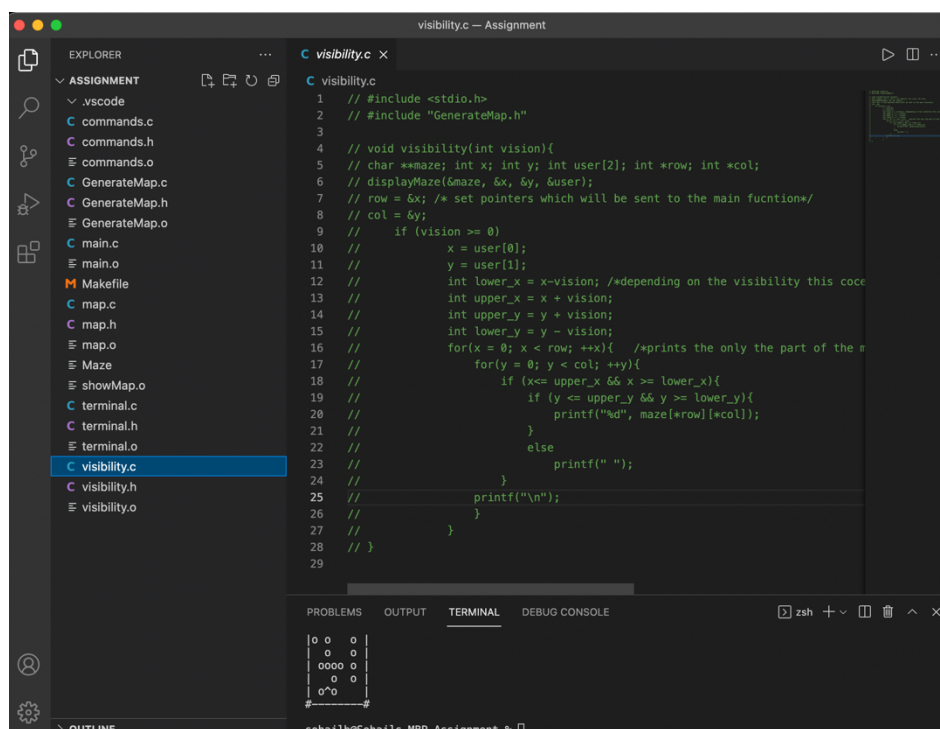
```
char** getInput(char** maze, char input, int *x, int *y)
```

A function to change the maze according to the commands by the user of w,a,s,d.

Int move[] uses an array to store the coordinates of the movements. The function uses if/else statements to create character controls. If the user hit w they go up, a to go left, s to go down and d to go right. If any other key is hit the program will throw an error.

3.6 Visibility Distance (with Conditional Compilation)

Unfortunately, I couldn't complete this section as I couldn't get it working. I tried by creating a function called visibility() and tried to pass it through to the main function but couldn't get it to work.



```
1 // #include <stdio.h>
2 // #include "GenerateMap.h"
3
4 // void visibility(int vision){
5 // char **maze; int x; int y; int user[2]; int *row; int *col;
6 // displayMaze(6maze, 6x, 6y, 6user);
7 // row = 6x; /* set pointers which will be sent to the main function*/
8 // col = 6y;
9 // if (vision >= 0)
10 //     x = user[0];
11 //     y = user[1];
12 //     int lower_x = x-vision; /*depending on the visibility this code
13 //     int upper_x = x + vision;
14 //     int upper_y = y + vision;
15 //     int lower_y = y - vision;
16 //     for(x = 0; x < row; ++x){ /*prints the only the part of the m
17 //         for(y = 0; y < col; ++y){
18 //             if (x<= upper_x && x >= lower_x){
19 //                 if (y <= upper_y && y >= lower_y){
20 //                     printf("%d", maze[*row][*col]);
21 //                 }
22 //                 else
23 //                     printf(" ");
24 //             }
25 //             printf("\n");
26 //         }
27 //     }
28 // }
```

I have deleted the file for submission but this what I had.

3.7 Winning Condition

When the user reaches the x the terminal screen prints "YOU WIN!!"