Spike: 1

Title: Grid World

Author: Krishna Adhikari, 4953193

Goals / deliverables:

The goal of this task was to create console based adventure game named “Grid World”. The player guides the character towards the goal through the grid-world of block positions. The player can choose to move in four direction. The game consists of simple game loop, which is based on getting input, updating the data and rendering.

Besides the report, these things were created:

- Game design on paper
- Grid-World game

Technologies, Tools, and Resources used:

List of information needed by someone trying to reproduce this work

- Visual Studio 2015

Tasks undertaken:

List key tasks likely to help another developer
This section should resemble a tutorial – the goal is to allow another coder to reproduce your work following these steps.

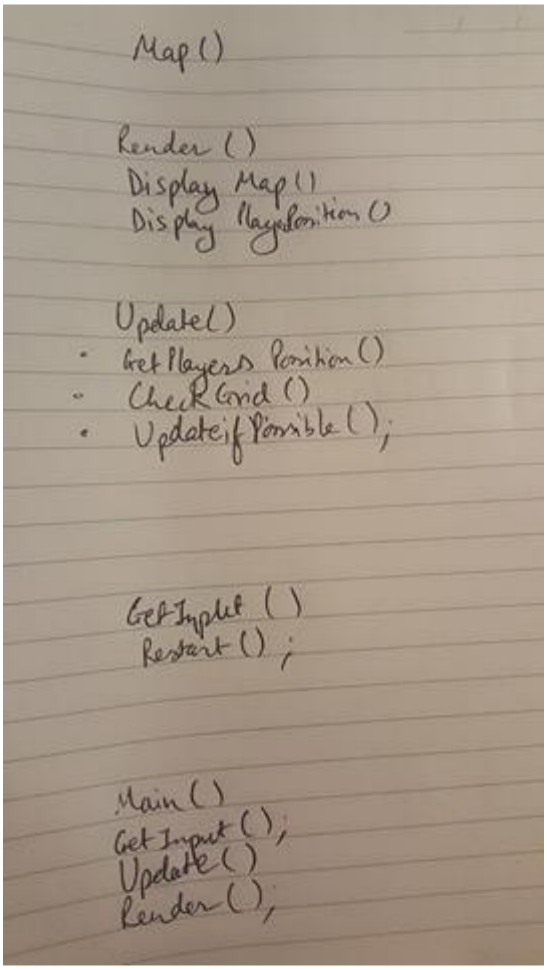
Eg: (Good)

- Download and install Visual Studio
- Design the basic game logic on paper
- Create a C++ project
- Start by implementing the main game loop which would include having a conditional loop, getting input, Updating the game and Rendering.
- Implement each functions
- Compile the code and run to see if it works.
- Debug if required

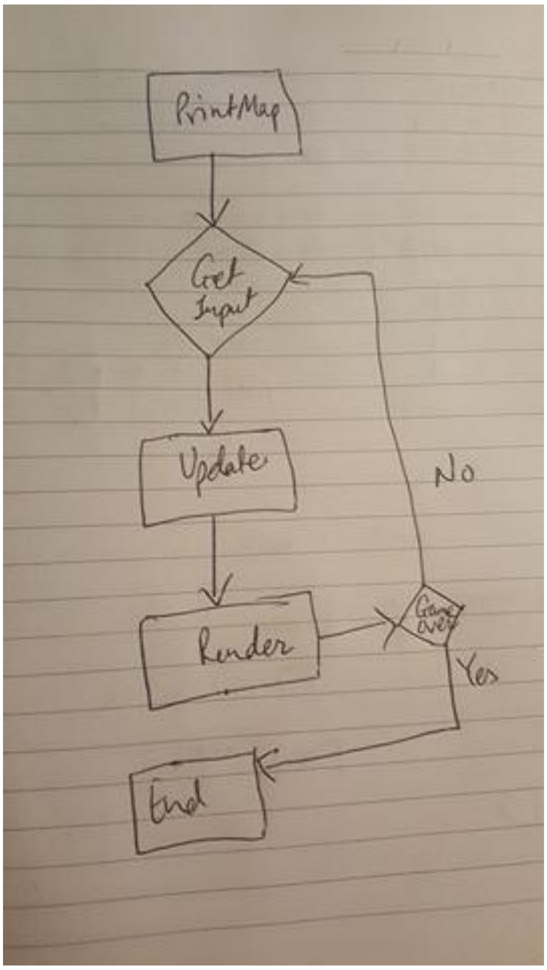
What we found out:

Describe the outcomes, and how they relate to the spike topic + graphs/screenshots/outputs as needed

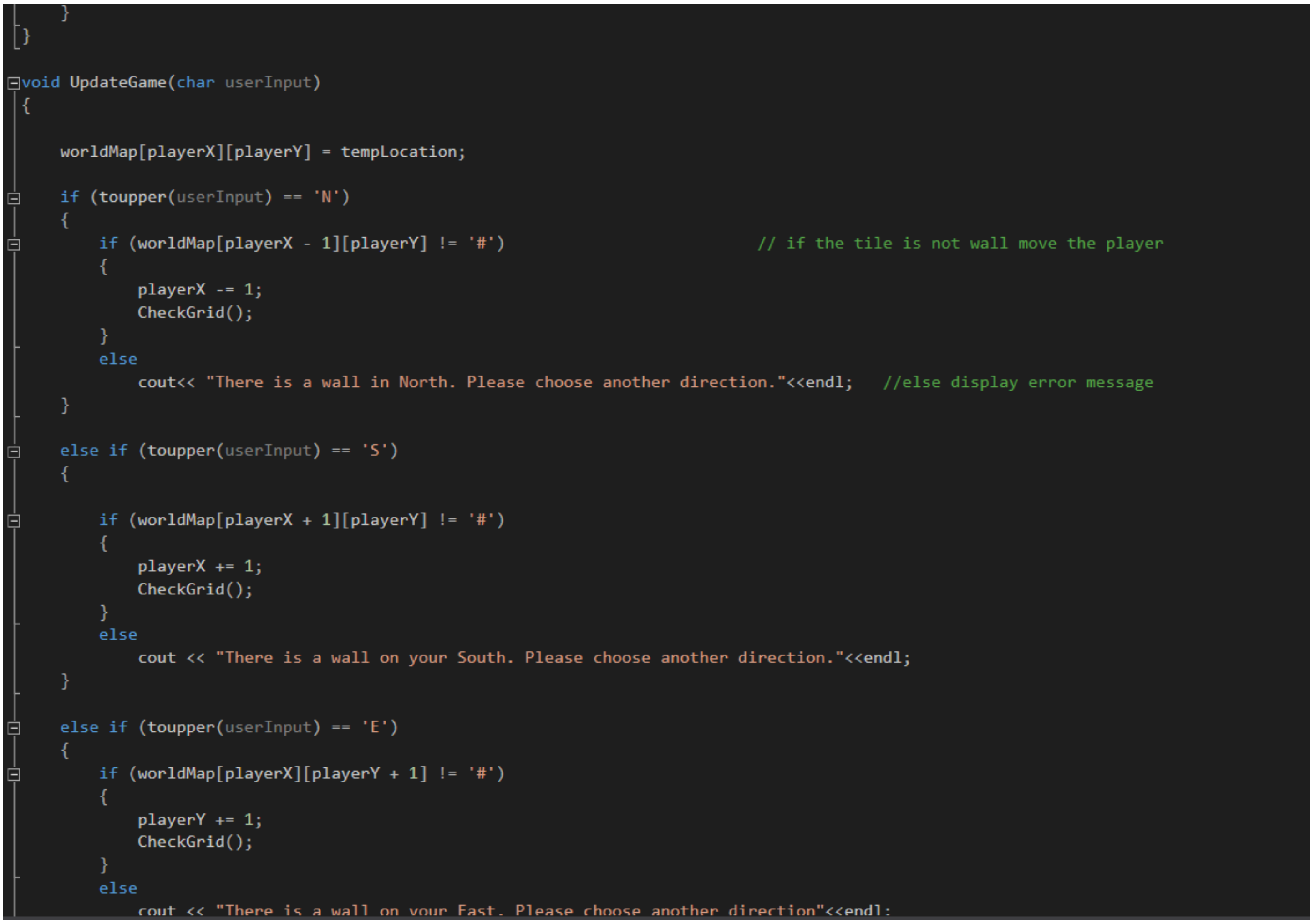
In this spike, I worked on procedural programming. The most effective practice I followed was de-modularising the codes. I create different functions to handle user input. Likewise, Updating and rendering functionality were separated so that it is easier to manage the code. Any codes that were repeated a lot were put in a separate function. I used static 2D array to store the map as it is easily manageable.



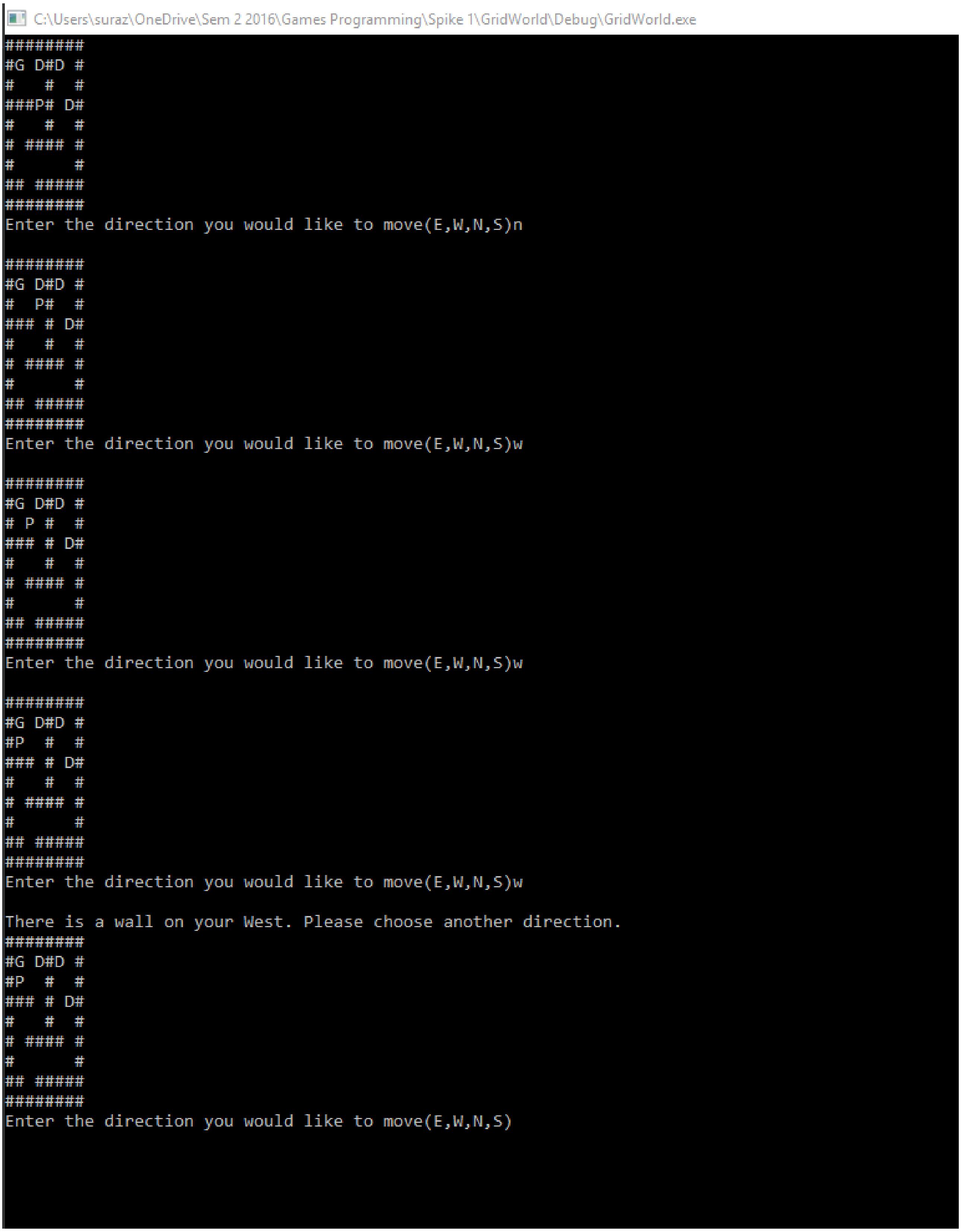
```
Map()

Render ()
 Display Map()
 Display PlayerPosition ()

Update()
• GetPlayers Position()
• CheckGrid ()
• UpdateifPossible();

GetInput ()
 Restart ();

Main ()
Get Input ();
Update ()
Render ();
```



The screenshots above show the basic workflow of the program. The main function will have a loop which runs until the game is over either by reaching the goal or due to death in process. After every move, the user is asked for the input. The input is then used to update the game and appropriate outcome is rendered on the screen. Below are the screenshots of the codes and game running.



```
    }
}

void UpdateGame(char userInput)
{

    worldMap[playerX][playerY] = tempLocation;

    if (toupper(userInput) == 'N')
    {
        if (worldMap[playerX - 1][playerY] != '#')                          // if the tile is not wall move the player
        {
            playerX -= 1;
            CheckGrid();
        }
        else
            cout<< "There is a wall in North. Please choose another direction."<<endl;   //else display error message
    }

    else if (toupper(userInput) == 'S')
    {

        if (worldMap[playerX + 1][playerY] != '#')
        {
            playerX += 1;
            CheckGrid();
        }
        else
            cout << "There is a wall on your South. Please choose another direction."<<endl;
    }

    else if (toupper(userInput) == 'E')
    {
        if (worldMap[playerX][playerY + 1] != '#')
        {
            playerY += 1;
            CheckGrid();
        }
        else
            cout << "There is a wall on your East. Please choose another direction"<<endl;
```



C:\Users\suraz\OneDrive\Sem 2 2016\Games Programming\Spike 1\GridWorld\Debug\GridWorld.exe

```
########
#G D#D #
#   #  #
###P# D#
#   #  #
# #### #
#      #
## #####
########
Enter the direction you would like to move(E,W,N,S)n

########
#G D#D #
#  P#  #
### # D#
#   #  #
# #### #
#      #
## #####
########
Enter the direction you would like to move(E,W,N,S)w

########
#G D#D #
# P #  #
### # D#
#   #  #
# #### #
#      #
## #####
########
Enter the direction you would like to move(E,W,N,S)w

########
#G D#D #
#P  #  #
### # D#
#   #  #
# #### #
#      #
## #####
########
Enter the direction you would like to move(E,W,N,S)w

There is a wall on your West. Please choose another direction.
########
#G D#D #
#P  #  #
### # D#
#   #  #
# #### #
#      #
## #####
########
Enter the direction you would like to move(E,W,N,S)
```