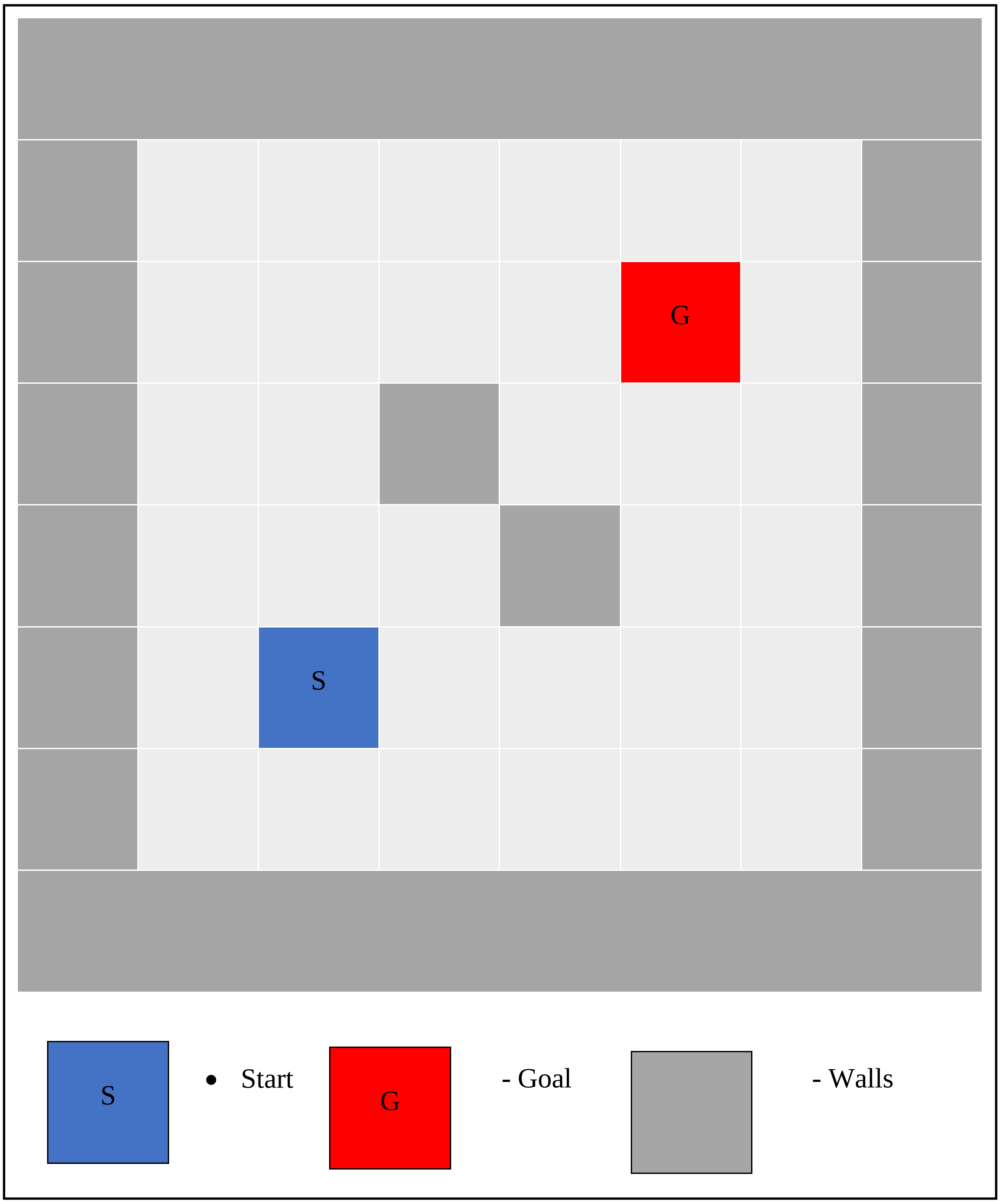
Grid World



Grid World Plane

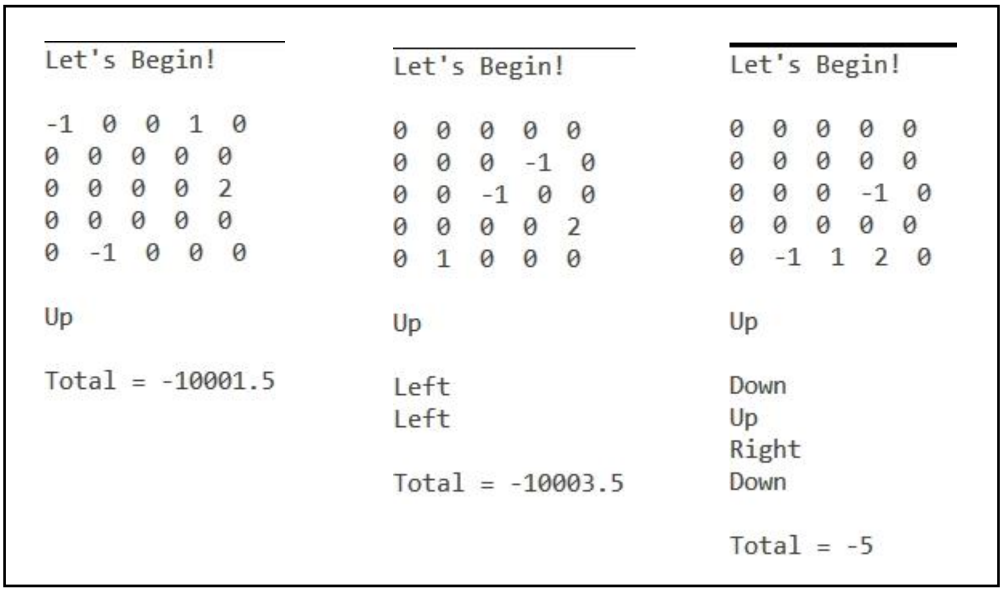
I was to create a C++ program to find a path from S (Start) to G (Goal), moving up, down, left or right. I was supposed to score for each move the player takes and eliminate them if they hit a wall. I used Code::Blocks 17.12 on Windows 10 to write C++ codes.

I decided to use the following notations in the code.

- 0 = Empty space
- 1 = Start position
- 2 = Goal
- -1 = Walls

Score started from 0 and for every move the player took, -1 was added. If the score reached -100, the player was eliminated since they made too many moves. If the player hit a Wall, -10000.5 was added to the score and because of the large negative floating-point value, it was easy to recognize if the player hit a wall. Outside Wall was equivalent to going out of the grid.

First, the program generated a random grid world plane. Then, it considered the player's array of moves. If the player had not already found the Goal or hit a Wall, then the program would generate random moves until it found the Goal or hit a Wall. Finally, it gave the score.



```
Let's Begin!

-1  0  0  1  0
0  0  0  0  0
0  0  0  0  2
0  0  0  0  0
0  -1  0  0  0

Up

Total = -10001.5
```

```
Let's Begin!

0  0  0  0  0
0  0  0  -1  0
0  0  -1  0  0
0  0  0  0  2
0  1  0  0  0

Up

Left
Left

Total = -10003.5
```

```
Let's Begin!

0  0  0  0  0
0  0  0  0  0
0  0  0  -1  0
0  0  0  0  0
0  -1  1  2  0

Up

Down
Up
Right
Down

Total = -5
```

Here are some outputs of the program. In all 3 occasions, the player onlymade one move, "Up". In 1st scenario, the player hit the outside Wall and was eliminated. In 2nd situation, the player did not find the Goal or hit a Wall. Therefore, the program made random moves, hit a Wall and then eliminated. However, in 3rd scenario, the program found the Goal.