**Spike:** 05

**Title:** GridWorld

**Author:** Ben Holmes, 103024841

**Goals / deliverables:**

* Plan for code
* The code for GridWorld with separation of:
  + Processing of input
  + Updating the game model
  + Display of the game to user

**Technologies, Tools, and Resources used:**

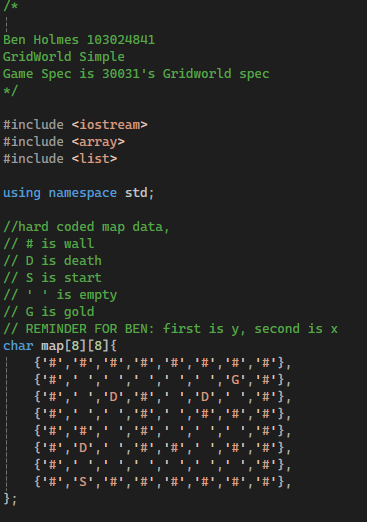
* Draw.io for diagram
* Visual Studio 2022
* Word
* C++

**Tasks undertaken:**

* Created initial file including map
* Made plan
* Created visual studio project and basic code structure
* Created display
* Created input
* Created update
* Created exit

**What we found out:**

All of the code is in the GridWorld.cpp file which was initially created with a hardcoded map.



A screenshot of a computer screen

Description automatically generated

I then created the plan for the code structure which is in the Spike05planDiagram.png and is also included below, this was the basic layout/plan, some minor differences between this and the final result including the addition of the quit global variable. This is the first of the deliverables, which is done before/early in the development in order to have a clearer idea of what needs to be done.

A diagram of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

From

The basic structure of the code was created. This included a minor mistake of having the input taking an input instead of the update taking it. I also had forgotten to create a visual studio project/solution before now (haven’t worked in visual studio in a while) so did that as well

A black screen with white text

Description automatically generated

A screen shot of a computer program

Description automatically generated

Next was the display function, which is the renderer of the deliverables. I added the functionality of showing the full map including player location as well as displaying a key and split lines.

A computer screen shot of a program code

Description automatically generated

A black background with white text

Description automatically generated

A render being separate is important for a game so that it is easy to update it on its own without having to crawl through other parts of the games code and possibly cause some errors or bugs to appear.

A screen shot of a computer

Description automatically generated

Next is the input function, which was started in the previous commit, which displays the possible inputs, including the input for quit on a separate line, and then takes an input (char option).

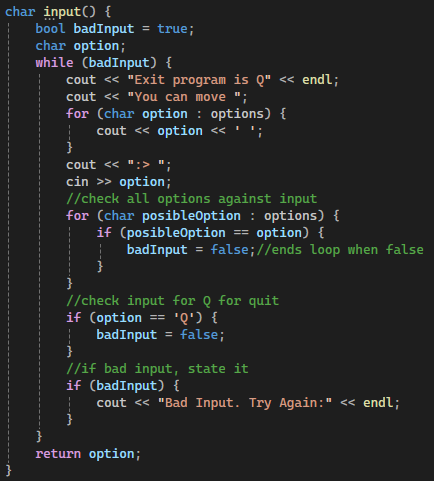
It then compares the input to the potential options (the global options variable) and if correct, exits the while loop, otherwise, states that is a bad or incorrect input and loops.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated



This is a separate section of the game loop for the same reason as the display.

A screen shot of a computer

Description automatically generated

Penultimately is the update function, which takes in the input (char option).

First it checks the option in a switch case and updates the playerX or playerY depending on the direction input, it also handles the quit (Q) input here.

The default option being blank will be explained below in the last section.

Then it checks the new tile for the death (D) or gold (G) designators changing playing if they are on that tile, also changing success if it is specifically G.

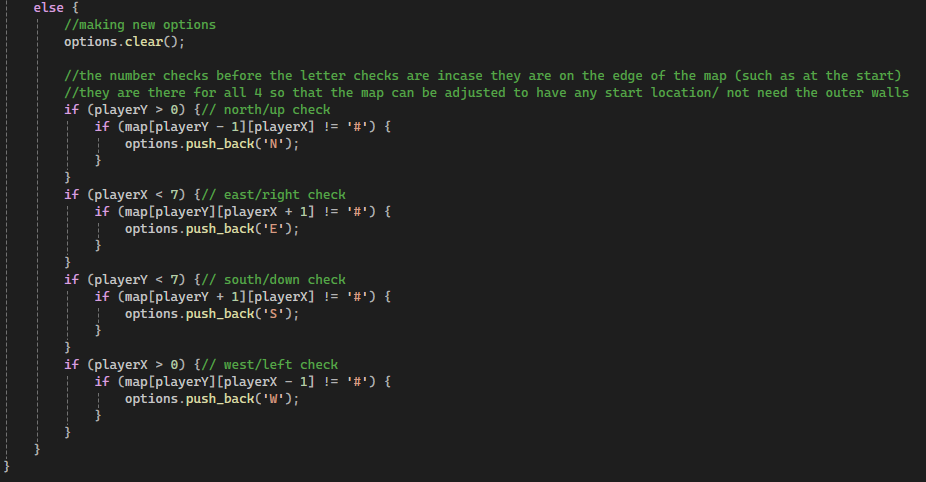
A computer screen shot of a program

Description automatically generated

If an end (D or G) is not on that tile then it checks the possible options.

First it clears the list, then it checks each of the possible directions:

* either being out of range (done first to prevent issues)
* then if the next tile is a wall (#)



A computer screen with numbers and letters

Description automatically generated

The last thing implemented was the exit comments in the main function. This was done via checking the quit bool and success bool and doing a different output depending on the combination, with quit being highest order, then success.

A computer screen shot of a program code

Description automatically generated

The default option in update is for that first update to work so that I would not have to hard code in the initial options if I wanted to change the staring location to a point that had multiple options. I would still have to hardcode the staring location though (although a function to create that manually would be simple).