**Spike:** 01 (Task 03)
**Title:** Gridworld

**Author:** Hayden Whiteford,104001272

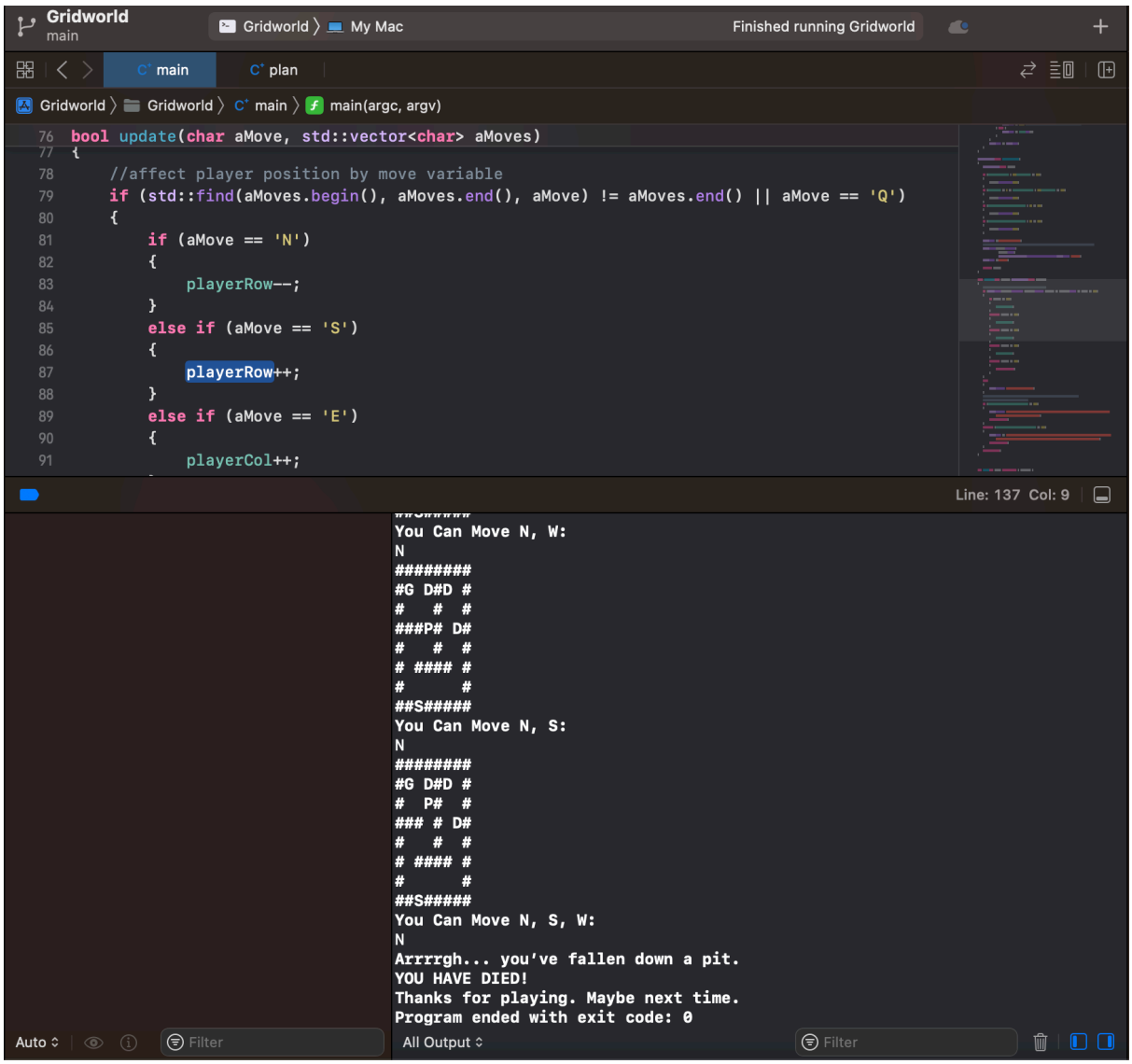
**Goals / deliverables:**
- **Create the Gridworld game as per the specification document**
- **Demonstrate separation of input, update and render code**
- **Use of game data for player and Gridworld map**

**Technologies, Tools, and Resources used:**
- Xcode for Mac
- W3 schools to brush up on c++

**Tasks undertaken:**
**How to run:**
- Download and install Xcode or visual studio
- Create a new c++ console application project
- Replace main file with my code
- Compile and run

In terms of Code:
- Created a vector containing more char vectors to create the Gridworld “map”
- Created a few functions:
- update() - updates player position based on input - decides output with intractable items D and G
- checkMoves() - Checks the valid moves for the player based on surrounding chars in the vector grid
- render() - uses a for loop to output the world, can override the grid world with a marker for the players position - “P”

**What we found out:**
Went fairly smoothly and was able to bang this out mostly in a single night. Did need to brush up on C++ to check out arrays vs vectors. Vectors worked better here as they were dynamic in terms of size, and for the checkmoves() function I could simply use the push_back function to append to the vectors the valid moves for the turn. Splitting up functions went smoothly as it seemed the most straightforward way of structuring the code - having an update function for portions and interactables, something to check the moves, and something to render the board, which I decided I should implement from a debug perspective.




**Open issues/risks:**

This wasn't written OO style. I felt I didn't need it and would have made it needlessly complicated, but could quite easily implement a game world and player object if that's needed.