COS30031-2023-103071494 / 03 - Spike - Gridworld / main.cpp

unknown Gridworld complete! (hopefully) 41 minutes ago

156 lines (132 loc) · 3.08 KB

Code Blame Raw

```
1   #include<iostream>
2   #include<cctype>
3
4   using namespace std;
5
6   class World {
7   private:
8           int mapSize = 64;
9           char map[64] = { '#', '#', '#', '#', '#', '#', '#', '#',
10                                  '#', '#', ' ', '^', '#', ' ', ' ', '#',
11                                  '#', ' ', ' ', ' ', '#', ' ', ' ', '#',
12                                  '#', '^', ' ', '#', '#', ' ', ' ', '#',
13                                  '#', ' ', ' ', ' ', ' ', ' ', '^', '#',
14                                  '#', ' ', ' ', '#', 'G', ' ', '#', '#',
15                                  '#', 'S', ' ', '#', ' ', '^', '#', '#',
16                                  '#', '#', '#', '#', '#', '#', '#', '#',};
17
18          int playerPosition = 49; //Value corresponding element S
19          bool gameRunning = true;
20
21          char* findValidMoves() {
22                  //* Dynamic arrays are pretty cewl, I'm assuming it's auto-deleted when
23                  //* out of scope.
24                  char* validMoves = new char[4];
25
26                  //'T' is a default value loaded for later presentation functionss
27                  for(int i = 0; i < 4; i++){
28                          validMoves[i] = 'T';
29                  }
30
31                  if(map[playerPosition - 8] != '#'){
32                          validMoves[0] = 'N';
33                  }
```

```
34                      if(map[playerPosition + 1] != '#'){
35                              validMoves[1] = 'E';
36                      }
37                      if(map[playerPosition + 8] != '#'){
38                              validMoves[2] = 'S';
39                      }
40                      if(map[playerPosition - 1] != '#'){
41                              validMoves[3] = 'W';
42                      }
43                      return validMoves;
44              }
45
46              void handleEvents(){
47                      if(map[playerPosition] == '^'){
48                              cout << "AH! Man's hit a spike!" << endl;
49                              gameRunning = false;
50                      }
51                      if(map[playerPosition] == 'G'){
52                              cout << "Good job, you got the goop!" << endl;
53                              gameRunning = false;
54                      }
55              }
56
57              //Only presents movement directions, ommits default array values
58              void presentValidMoves(){
59                      char* validMoves = findValidMoves();
60
61                      cout << "You can move: ";
62
63                      for(int i = 0; i < 4; i++){
64                              if (validMoves[i] != 'T'){
65                                      cout << validMoves[i] << ", ";
66                              }
67                      }
68                      cout << endl;
69              }
70
71              void handleInput(char input){
72                      switch (input){
73                              case 'N':
74                                      playerPosition -= 8;
75                                      break;
76                              case 'E':
77                                      playerPosition += 1;
78                                      break;
79                              case 'S':
80                                      playerPosition += 8;
81                                      break;
82                              case 'W':
83                                      playerPosition -= 1;
```

```
84                                      break;
85                              case 'Q':
86                                      playerPosition += 0;
87                                      gameRunning = false;
88                              default:
89                                      playerPosition += 0;
90                                      break;
91                      }
92              }
93
94              void printMap(){
95                      for(int count = 0; count < 64; count++){
96                                      if(count % 8 == 7){
97                                              cout << map[count] << endl;
98                                      }
99                                      else{
100                                             cout << map[count];
101                                     }
102                             }
103             }
104
105     public:
106             World(){
107                     render();
108             }
109
110             bool getGameRunning(){
111                     return gameRunning;
112             }
113
114             char getInput(){
115                     char input;
116                     cin >> input;
117                     char upper_case_char = toupper(input);
118
119                     return upper_case_char;
120             }
121
122             void update() {
123                     presentValidMoves();
124
125                     char input = getInput();
126                     handleInput(input);
127                     handleEvents();
128             }
129
130             /*Though personally I don't see the need for a render function within
131               a CLI game it was specified in the deliverables. This function will
132               simply reprint the starting map each turn.*/
133             void render(){
```

```
134                     printMap();
135             }
136
137     };
138
139     void introSec() {
140             cout << "Welcome to Gridworld: Quantised Excitement!" << endl;
141             cout << "Fate is waiting for you! (Coder: Thomas Horsley - 103071494)" << endl;
142             cout << "Valid commands: N, E, S & W for Direction. Q to quit" << endl;
143             cout << endl;
144     }
145
146     int main() {
147             introSec();
148             World world;
149
150             while(world.getGameRunning() == true){
151                     world.update();
152                     world.render();
153             }
154             return 0;
155     }
```