**Shortest Path Finder Steps**

1. First import the curses module, from the curses module then import wrapper, then import queue, then import time
2. The make the maze values inside nested lists

maze = [

["#", "O", "#", "#", "#", "#", "#", "#", "#"],

["#", " ", " ", " ", " ", " ", " ", " ", "#"],

["#", " ", "#", "#", " ", "#", "#", " ", "#"],

["#", " ", "#", " ", " ", " ", "#", " ", "#"],

["#", " ", "#", " ", "#", " ", "#", " ", "#"],

["#", " ", "#", " ", "#", " ", "#", " ", "#"],

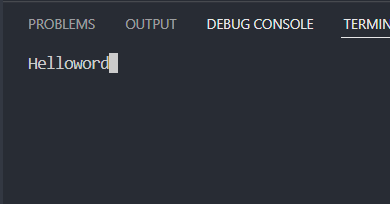
["#", " ", "#", " ", "#", " ", "#", "#", "#"],

["#", " ", " ", " ", " ", " ", " ", " ", "#"],

["#", "#", "#", "#", "#", "#", "#", "X", "#"]

]

1. Create a main function
   1. give parameters called stdscr which means standard screen
   2. clear the screen
   3. addstr to the screen and then parameters (row, col, value, color)
   4. refresh the screen
   5. getch the screen which means it will stay until we give an input for it



1. Add colors to an Output
   1. Inside of the main function add the top
   2. Make a curses with an init\_pair and then pass parameters such as (id number, curses.COLOR\_FOREGROUNDCOLOR, COLOR\_BACKGROUNDCOLOR)
   3. then create a variable for the color with the variable name and then the color\_pair (id from the init\_pair)
   4. Make 2 colors with GREEN and RED which can later be used and keep then made with color\_pair and give then different ids as well
2. To run the code use the wrapper import and then use the main function without calling it as parameter and then run the code.
3. Then we print the maze for this we need an new function
   1. for the function pass in the parameters called **maze**(meaning the maze which we made), **stdscr**(the screen which we will display the maze on), **path=[]** (the path which we want the coords to be in)
   2. First of all we need to create the colors which we want use BLUE, GREEN colors and when calling then use the method called color\_pair(id)
   3. create an for loop inside of the function which loops the maze and then **gets both the position and the index**
   4. Create another for loop inside of the for loop where it gets the index of the values inside of the nested lists and this should **get both index and the value** as well from the above made for **loop which gets the value or the list which was nested**
   5. Then we print the values to the stdscr(screen which we imported)
   6. Simply we use the addstr method and then we pass the (row \* 2, column \* 2, value) we do the \* 2 because we don’t want to squash all the values
4. Finals steps to printing the maze is that we have to remove the earilier values which we printed using the addstr inside of the main function.
   1. We **call the print\_maze function** which we made with the for loops **instead of the addstr method** and then we pass the **maze parameter** because that is what we want the for loops to go through and then **we pass then screen** as well which we gave a parameter for the main function.
   2. Then run the system you will see an maze output