The A-Mazing DS4 Race

LAB # 7

SECTION #

Riley Lawson

SUBMISSION DATE:

11/12/2019

DATE:

11/12/2019

Problem

The purpose of this lab was to develop a game where you can move left or right through a maze using the gyroscope of your ps4 controller and this was split up into a two-part lab. This consisted of using draw_character, avatar, maze, and functions relating back to main.

Analysis

For the first part of the lab, it states that we practice top-down learning, develop skills in looping, learn about (and how to use) lncurses, and understanding how to process input data for moving averages. In the second part of the lab, on top of the first lab, we add learning to use 2D arrays and practice with even more loop. It drew the maze with a level of difficulty, scan in data from the controller and time would begin. Once the user completed(or did not complete) the maze it would state "YOU WIN!" or "YOU LOSE" and the program would end.

Design

Within this program, I managed to get the solid groundwork for my process and this includes

1. Creating the moving average function (and move it over when doing part B)
2. Get something showing on the screen
3. Tweak the program within main to make the avatar move downwards (then eventually left and right)

Part 2 of the lab includes

1. Setting up walls and double-check if the avatar still moves right or left
2. Making the main function more complex to handle more functions
3. Setting up the generate_maze and draw_maze functions
4. implement those functions within the main

There were no obvious equations used besides rand and srand and variables dealing with time.

Testing

This program consisted of a massive amount of testing through trial and error trying to figure a plethora of things including (but not limited to) why the triangle button wouldn't work, why the maze "*" wasn't showing up but the maze and the avatar were, why were there stars on the side and a segmentation fault, and why was there just a black screen with a cursor on it. Most of these were just simple fixes especially within the moving_average and the draw_maze function. Something small going from "*" to WALL or placing if(down >= 80). The hardest part had to include compiling because there weren't many errors and unless you knew where to look to fix something then you had to go back through your code a few more times to identify the problem. For preventing erratic movement I added

200 to time and kept it within the parameters of 100 Rows and 80 Columns. For checking the avatar I would keep it within a certain range of gyroscope values and time would continuously move the character down. To check if the player won or lost there are conditions that were put in place. Including if the player couldn't move left or right due to the "walls" put in place or if the player reached the end of the maze meant that they either won or lost.

Comments

In this lab, I learned how to create a functioning maze that I might be able to play later. I also learned how to write loops more efficiently and what lncurses were (and how to use it).

ScreenShot(s)