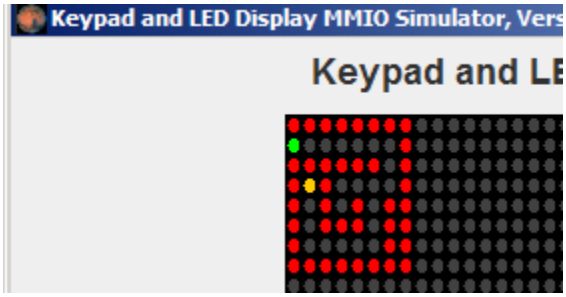
CS 0447 – Lab 4: LEDs and Keys

Description

For this project, we will use the modified MARS with the LED and Keypad simulator that you will need for your project. You can download it from the class website.

You will be implementing a simple maze program. You will be a green dot whose job is to navigate a maze of red walls to the gold treasure at the end. You will draw the maze (provided in the skeleton) and then handle keypresses to move the hero through the maze. The program ends when the goal is reached.



Requirements

- Draw an 8x8 maze from an array in the .data segment (maze skeleton provided on the class website)
- The elements of the array are the colors to display and the position is the coordinates.
- Array[x][y] is the same as Array + (x * sizeof(row) + y * sizeof (cell)). [Square 2-D arrays don't really exist, they are just larger 1-D arrays – however, jagged arrays are implemented differently.]
- After the maze is drawn, use a polling loop to get keypresses. A polling loop is a loop which checks the value of something (e.g., the keypress) every iteration. Your loop can sleep 200 milliseconds on each iteration to reduce the CPU requirements while waiting for the player to hit a key. Syscall 32 in MARS allows you to sleep for $a0 milliseconds.
- Do not allow the player to move into walls or out of the entrance.
- Each valid keypress moves the player in a direction. Erase the old position by setting the LED color to 0 (black).
- End the program when the yellow/orange pixel is reached.

Use the same submission procedure that you used last time.