Lacey Kisselburgh

Lab 022

Custom Lab Project Report

I created a game where the player controls a character who must navigate through a house with several “trap doors” that are filled with ghosts who they can destroy for points. The player cannot phase through these ghosts, but once they have been defeated, they can then move into the spots that the enemies were previously. The player is placed into a random room every time they enter through a door, except they will not be placed into the same room they just left from. Additionally, the player will not be sent into the “exit” room until they’ve visited at least three other rooms. If the player escapes, they are brought to an “end game” screen that displays their score. They received 5 points for every ghost they defeated and receive a 35 point bonus just for escaping. The player can also navigate a start menu where they can start the game, toggle music, or view controls. They can also navigate a pause menu mid-game.

User Controls

The player can use the “d-pad” to the right of the LCD screen that is simulated through 4 buttons. They can move up, down, left, or right. There is also a button to the left of the screen that the user can use to confirm selections on the screen or, while the game is running, can be used to attack ghosts that the player is adjacent to. There is another button in the top right of the breadboard that the player can use while mid-game to pause. From there, they can use the confirmation button to resume the game, edit the settings, or exit to the main menu. The user must navigate the avatar through the doors in order to leave the room.

Complexities

* Nokia 5110 LCD Screen (see nokia\_5110 files)
* Special characters for the 5110 LCD screen by using the set\_pixel function in my own functions (see bottom of nokia\_5110.h)
* Game Logic
* Saving audio settings to EEPROM

Difficulties

The biggest difficulty I faced with this project was both being too ambitious and also spending too much time on problems whose solutions I was unaware of. Firstly, I originally planned to use an analog joystick as a complexity. But when it arrived, I realized that I didn’t have the necessary interface to connect the joystick to the board. Instead of taking the risk of ordering another and having to wait for it to come in case it arrived late. I decided to order an NES controller from amazon since I could get it in two days, but then the model was a USB version and I couldn’t find any resources on how to interface it. By the time I decided to give up, I decided to get an older model of the controller since I had found many tutorials on it.

I got it the Tuesday before the final demo, and I spent many hours following several tutorials to no avail. I decided to cut my losses and just use the buttons to navigate. Additionally, I spent the first week trying to solve the issue with my speaker where two of the same frequencies would play as one long note. After reviewing a lab manual, I saw the easy fix for this issue, so I lost a week working on that. I should have reprioritized it, but I was waiting to receive the LCD screen as well.

What I Learned

Despite not being able to get the controller to function in time, I learned a lot about how it should work. I learned about pulsing the latch in order to get the input, and accessing the shift register. I also gained a better understanding of clocks, latches, and data and how they are used in interfacing hardware.

I also learned how to navigate certain limitations that C poses, such as being unable to declare local variables within a case in a state machine and not being able to alter variables that are passed into a function. By designing a game, I had to tackle several problems that required very complex logic to implement. Such as not allowing the player to phase through enemies in a room, but not being restricted to those areas after the enemies were expelled. Additionally, being able to record which enemies are dead and which are alive, and remembering them as the person passed through several rooms, required a lot of logic that was hard to keep track of, especially when it had to be written for 6 different rooms.

Video

<https://youtu.be/rjbcvl7AaJ0>

Header Links

[io.c](https://drive.google.com/open?id=1JjJA8ToGE4XVyElJgczkODNdsi0vzuZu) - implements several functions used in other files

[io.h](https://drive.google.com/open?id=16wosFtOaJJH0J77uVYSAvzs9XI99WSnN) - header for io.c

[game.h](https://drive.google.com/open?id=17DMJFDqEXmmj65I8oQFkeiy7AFuRB2D3) - contains the game logic that dictates all functions of the game

[start\_menu.h](https://drive.google.com/open?id=1ACt-fOydZ1ju1lZhjTPku4ibluAPpgcl) - contains the menu logic I created for the start menu

[pwm.h](https://drive.google.com/open?id=1ShIcpo8r_Ok7Fholrxr8qYHh4O3XNxuD) - includes logic for song playback, library taken from

[pwm source code](https://docs.google.com/document/d/1vHjsFThDFh1-m9I3hdX_kx16ess92cDW_oBkOPKDC2w/edit) - lab manual 9

[timer.h](https://drive.google.com/open?id=1DGWT03lMlHiUxYWjz1DO3mRh94bC4xxx) - used for the pwm file, taken from

[timer source code](https://docs.google.com/document/d/1xDo-B6HC0Jfr-llc-rEft0562xngp2zYMuQrh2E2IN0/edit) - lab manual 6

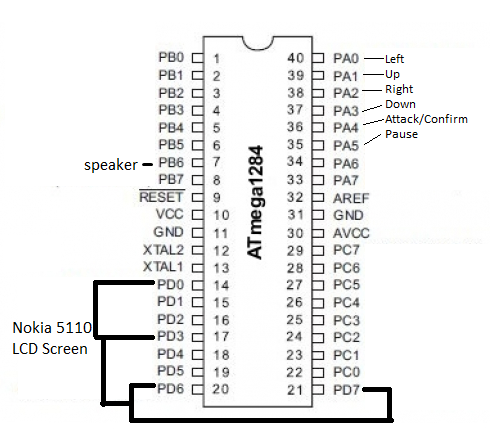
[nokia5110.h](https://drive.google.com/open?id=1nzzpPsRqqJF0MXB-9212FJWg9yUeMsFg) - includes custom functions that I wrote for the display

[nokia5110.c](https://drive.google.com/open?id=1aWgAekDXWIdhcJf7N526f80rBStmYVnE) - implemented source functions listed in the header file

[nokia5110\_chars.h](https://drive.google.com/open?id=1BD9bF09h_vTwTgl3FO0zXJdjN3YUgJJj) - a header file included with the library

[nokia5110 library source code](http://skew.tk) - also linked at the top of the nokia files

Pinout



SM Diagrams

