**SWE-350 Design Report Template**

|  |  |
| --- | --- |
| **Topic:** | Topic 8: Programming A FPGA Device |
| **Date:** | 12/15/2022 |
| **Revision:** | 1.2 |
| **Milestone Summary:** | |  |  |  | | --- | --- | --- | | **User Story / Task** | **Hours Worked** | **Hours Remaining** | | *As a player I want to see the game on an LCD so that I can play properly* | *4* | *0* | | *As a player I want to press buttons so that I can control my snake’s direction* | *5* | *0* | | *Display the score on the LED Displays* | *3* | *0* | | *FPGA 7-Segment Display Decoder* | *4* | *0* | | *Building logic design* | *5* | *0* | | *Building system design* | *1* | *0* | | *Coding logic* | *8* | *0* | | *As a player I want to be able to play again if I die* | *3* | *0* | | *As a player I want to see the high score, and if I beat it, save it* | *0* | *5* | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
| **GIT URL:** | https://github.com/hydrenoid/SWE-350Milestone2.git |

**Design Documentation**

**General Technical Approach:**

The application will be a revision of the classic game of Snake, where you control a snake going around the screen avoiding the walls and trying not to hit yourself. If you manage to eat a piece of food then your score increases and you will get a little bit longer increasing the difficulty until you lose. The hardware utilized will be two radial buttons for turning the snake and to navigate the UI, four LED 7-segment displays to display the score, and a LCD screen to display the game. The LED decoder will be built into the FPGA, and all other functions will be built in C code running on the ARM Processor.

*Diagram, box and whisker chart

Description automatically generated***System Design:**

**Application Design:**

Diagram, schematic

Description automatically generated

**VLSI Design:**

*Diagram

Description automatically generated*

**Risks and Issues:**

As of now the risks are at a minimum. Having solved this issue with how to display to the LCD by mapping out the screen pixels to a larger size to be displayed. As of now the issues are getting more complex. I want to have a high score and this would be easiest with file integration, so writing to a file on the OS what the high score is and each time you load the program it reads that file to get the high score, and if you beat it then overwriting it. I have successfully integrated the FPGA to display to the LED 7-segment display with input from the user space. I got the replay ability working correctly and selection of whether you want to play again or end, while properly resetting all memory and variables for a new game. Sometimes there is a segmentation fault while closing all of the hardware but only happens sometimes, I have yet to figure out why that is happening from which section.

**Other Documentation:**

N/A So far there is no other documentation, more will surely be added in the next revision.