Technical Report

Team Members: Omar Elmejjati, Ayman Salah

Functionality:

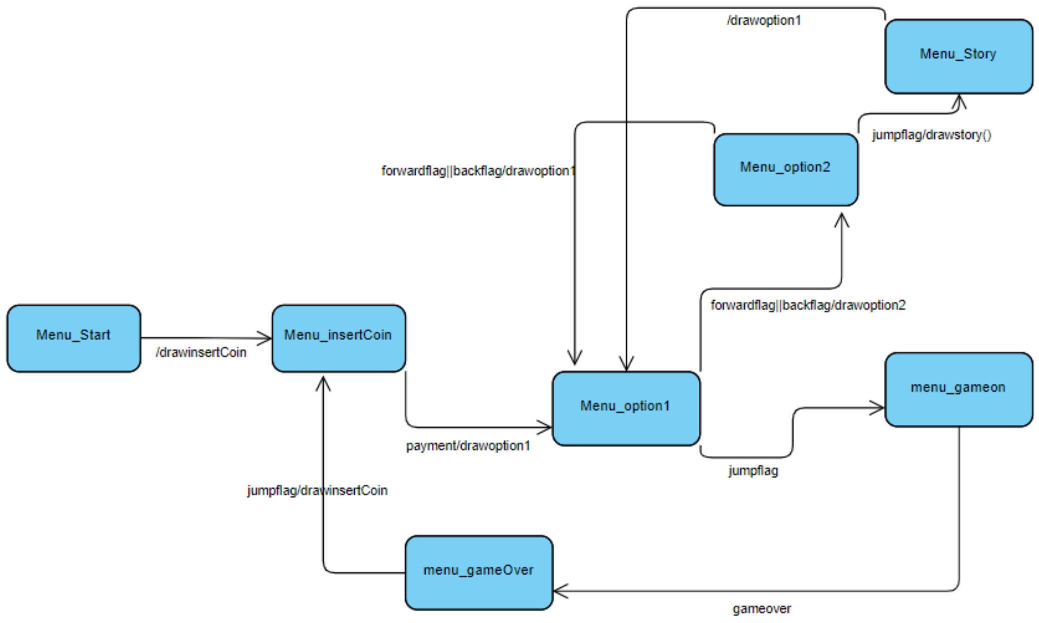
Several new additions were implemented for the game, in particular several different obstacles. New obstacles included were a laser that can spawn randomly on the lcd screen, a bee boss that moves in random ways and has a chance of moving backwards, and lastly a crusher that once it lands spreads to the cells causing the player to have a game over if they're caught in the area of effect.
The project also includes a functioning menu implemented through State machines.This implementation allows the user to have a smooth experience with our project.

Course Concepts:

Multiple threads were required for the sake of keeping code organized and uniform; it also serves to improve the overall quality of the game. This is done through separating our game aspect into 3 parts. The first being the player actions, second being the enemy action and lastly the LCD screen refresh. The player action and LCD screen refresh has a common clock which is a lot faster than the enemy action clock. This in combination with multithreading improved the response rate of the player hero compared to the enemy which made the game smoother to play.

A new state machine was added to our system to help support our new menu which functioned as a user interface. This made our project a lot simpler to use and function more like a real mini Arcade. The old state machine was slightly modified by changing the jump wait timer to account for the new clock speed however there was no change in the logic.

Lastly our project focused on introducing Significant programming complexity. This was mainly done through the new enemies which for our simple game represent the core of the gameplay. Utilizing the new improved clocking and multi threading systems, we were able to make enemies that are more challenging while still maintaining a high clear rate for these enemies. The project gallery link at the end of the report shows pictures of each enemy with a small introduction to how they work.



Base Game Functionality:

Our project base game needed to utilize a number of course concepts to begin with, the state machines were a necessity. Without the state machine the game wouldn't be able to run nearly as smoothly as it does and player characters could suffer from issues regarding lag or not being able to perform certain actions. This made the player movements independent of our obstacle movement. For the most part our original state machine remains unchanged except that the jump wait timer was changed to account for the new clock speed.

In addition, the interruption handling was used for the jump feature to allow for a quick response from our software. However interrupts weren't used for readings from joystick movement due to the fact that value remained for a longer time than (through the player holding the joystick) which simplified the issue. All user inputs go through Analog-to-Digital Conversion for example the joystick movement gets converted into digital values for our board to read.

Timing or the clock was important for our mini arcade game

Misc.

[Project Gallery]