RTOS Project Week 4

Project started at 5:23pm on 11/8/2022.

I estimate the following times to complete tasks this week:

Task	**Estimated Time**	**Actual Time**
Laser System	2 hour	45 min
Shield System	2 hour	1 hour
LEDs	1 hour	1 hour 15 min
LCD System	5 hour	6 hour
Total	10 hour	9 hour

This week, I finished the rest of the project. First was the laser system. This wasn't too bad, but I wanted to add support for the feature of multiple HMs in the future, so I added the ability to find and shoot the lowest HM. The shield system was a little tricky because it involved multiple timers to keep track of the cooldown of the active shield and the short window to reflect an HM at higher speed. For the LED system, I wanted to control the LED using PWM for brightness, but after looking at forum posts about it, it seemed like more trouble than it was worth. Instead, I opted to blink the LED at a higher frequency based on the speed of the platform. The highest frequency the timers support is hardly PWM, but it conveys the same information. The LCD system was terrifying to me. I knew this was going to be the hardest part of the project. The GLIB library has some very helpful functions, so I wasn't manually coloring every pixel, but it still took longer than expected to write. I wanted to have the auto cannon visually track the HMs, but my attempts at this didn't look like I wanted. After talking to a friend, he helped me write the code for a nice tracking function. After I finished all the code, I still had quite a bit of debugging before the project would work well. I had to track down a nasty null pointer dereference, which took a decent amount time. I had no call stack, and the debugger just went to the default_handler() function, so I had to manually step through until I found the issue. Additionally, I had some issues with sharing mutexes, semaphores, timers, and variables across multiple files. I solved this using the extern keyword, which allowed me to separate the declaration and definition of variables. This is a great bit of C knowledge to learn. I was able to add a couple extra out-of-scope features to my game. This included a menu to change several options including platform bounce, maximum HMs, and if the auto cannon is enabled. I added a score tracker and a high score tracker.

State of where project stands:

This week, I wrote the laser system, shield system, LED system, LCD system, and implemented a couple out-of-scope features. I debugged the entire project until I could successfully play the game.

Summary effort and estimate numbers:

I have completed **117%** of my currently scoped, estimated work (24.5hr actually spent / 21hr total estimate) in **121%** of the initially-estimated time. (25.5hr estimated for the items I have completed, of 21hr total estimate). For the work that has been completed, I took **1.21x** (15.5/15.5) as much time as I estimated.

List of in-scope work items:

Completed this week:

- Laser System (estimate: 2 hours)
- Shield System (estimate: 2 hours)
- LEDs (estimate: 1 hour)
- LCD System and Design (estimate: 5 hours)
- Menu