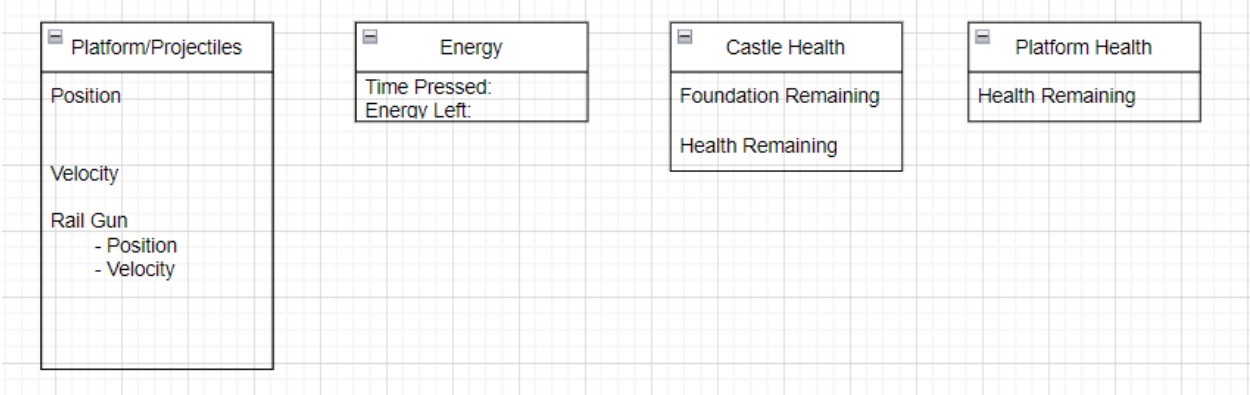
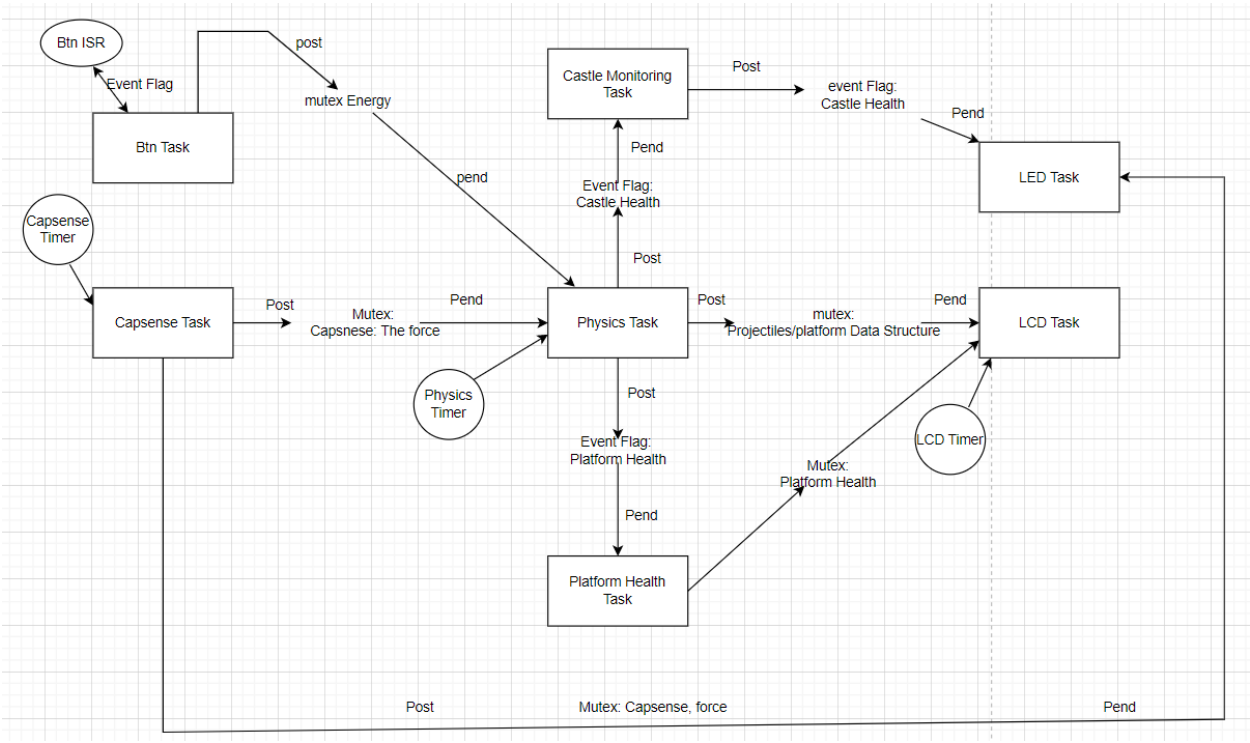
Wolfenstein Defense on EFM32PG12

[Project Repository]





Unit Tests:

Good cutting points for my design are between my physics task and all the other tasks. With these cutting points I will be able to Unit test my button Task, my Capsense Task, my Physics task, my castle Monitoring and Led Task in conjunction, and my LCD Task.

Unit Test 1: Physics Task. I plan to isolate the physics task from all the other tasks so that I am able to test that everything within the task is working correctly. I will test out the force to velocity conversion that I will have to do when the touch slider is touched. I will do this by using a random number generator to generate random forces, within the range that they can be made

from touching the slider and, and making sure that platform is in the correct position and has the correct velocity. Another functional test I will perform is to make sure the parabolic position of the projectiles is accurate based on how long the buttons are pressed. In order to accomplish this I will manually calculate where the projectiles should land based on pushing the button in increments of 1 second, and then use the incremental times in my physics task and make sure that the correct trajectories are indeed calculated.

Unit Test 2: Btn Task and ISR. I will utilize another cutting point between my button task and the Physics Task. I will create two tests, one to test that the task can detect when the button is pressed and released and another to establish that the task can determine how long the button was pressed before it is released.

Unit Test 3: LCD Task. I will utilize one of my cutting points at the LCD Task in order to isolate the LCD functionality and test that I am able to display all the things I need to have a functional game. I will test three things: the slider can be displayed within its full range of movement, The projectiles can be displayed in their full range of motion (practically the whole screen), and that the castle can be displayed properly.

Functional Tests for Demo:

- Use the slider to bounce the platform back and forth 4 times on the walls with as much force as possible (hold down the slider as accurately as you can) and see the end game screen.
- Use the shield when a Satchel is within striking range and watch it disappear.
- Hit the castle once and watch the led blink.
- Hit the foundation twice and watch the led blink.
- Get hit by a Satchel and see the end game screen.

Functional Tests:

Physics Task:

- Force applied to platform: I will test if the physics task can accurately move the position of the platform based on varying forces applied to it. I will test three different cases as well as edge cases. I will apply no force (edge case), maximum force (10 seconds on either far edge of the slider), 2 seconds on the far edge of the slider, and 4 seconds on one of the middle sections of the slider.

  Test has been run

  Pass/Fail: Pass
- Railgun while platform is still: I will test if my trajectories are correct for the projectiles when my platform is at rest. I will use varying values for the amount of energy for the projectiles and make sure that the trajectories and final destinations calculated will be correct

  Test has been run

  Pass/Fail: Pass

- RailGun while platform is moving: I will test if the trajectories and final locations for the projectiles are correct with varying values for the amount of energy the projectiles have as well as varying values for the initial velocity the projectiles have based on the platform's velocity.

  Test has been run

  Pass/Fail: Pass
- Satchel's being thrown: Test that I can accurately predict where the satchel's will land with a random number generator making the trajectories random.

  Test has been run

  Pass/Fail: Pass
- Collision with the wall: I will test that my physics task can accurately tell when the platform hits the wall with critical speed. I will accomplish this by inputting varying velocities of the platform and placing the platform at the far edge of the playable area by the wall and see if the platform health data structure is changed when the platform is at critical velocity upon collision.

  Test has been run

  Pass/Fail: This test Passed
- Shield Test: I will test of the platform is shielded from projectiles correctly when the shield is activated

  Test Not run yet

  Pass/Fail: Pass
- Health of the platform: I will test if the platform will be "destroyed" after it is hit with a satchel charge. I will test varying trajectories of the satchel charges with the platform moving at a fixed velocity back and forth and make sure that when the trajectory of the satchel lines up with the platform it is destroyed.

  Test not run yet

  Pass/Fail Pass
- Castle Health: I will test if the castle is hit by a projectile, and if it is hit if it can accurately monitor if it is hit again before everyone is evacuated.

  Test not run yet

  Pass/Fail: Pass
- Castle Health Test 2: I will test if the game will end if the castle is hit twice before the allotted time has run out for the people to evacuate.

  Test has not been run yet

  Pass/Fail:Pass
- Platform no Velocity test: I will test if the physics task can tell when equal force is being applied to both sides of the platform(both sides of the slider are being touched).

  Test has not been run yet

  Pass/Fail: Pass

LCD Task:

- Castle: I will test that I can display the correct image for the castle

  Not Run Yet

  Pass/Fail:Pass

- Satchel: I will test that I can display an image of a satchel or something to represent the satchels
  Not Run Yet
  Pass/Fail: Pass
- Area of the Satchels: I will test that I can display the satchels anywhere that it is possible to have a satchel be.
  Pass/Fail: Pass
- Projectiles: I will test that I can display the projectiles or something to represent the projectiles.
  Pass/Fail: Pass
- Area of the Projectiles: I will test that i can display the projectiles in all the areas that it is possible to shoot a projectile
  Pass/Fail: This test will fail because I havent started the LCD task
- Platform Image: I will test that I can display an image that accurately represents where the platform is.
  Pass/Fail: Pass
- Platform Movement: I will test that I can display the platform in all the possible locations that the platform can be.
  Pass/Fail: Pass
- Rail Gun Display: I will test that I can display an accurate image of the rail gun that depicts where the projectiles are shot from.
  Pass/Fail: Pass
- End Game: I will test that I can display the message when you either win or lose.
  Pass/Fail:Pass
- (OUT OF SCOPE) Test that I can display the health of the platform and the castle in some form. This is out of scope, but I think it would add a lot to the game if I have time to implement it.
  Pass/Fail: Pass

Functionality and Usability:

My project has all the in-scope functionality that I outlined at beginning working correctly. My platform gets accelerated based on where the capacitive touch sensor senses my finger. There is at least one satchel being thrown at once with a random velocity and if it hits the platform you lose the game. If you hit button 1 a shield will be activated only if a satchel is near the platform and will decrease the capacitive energy storage. If you hit button 0 a projectile will be launched with an initial velocity corresponding to the duration the button is held down. If you hit the castle, led 1 will pulse for 5 seconds and then you win if you do not hit the castle again while the prisoners escape.

Summary of Effort:

The total time I have outlined for the in scope work for this project is 39.5 hours. The work for this week took me a total of 10 hours. This accounts for 25.3% (hours of estimated work completed out of 39.5 hours of total work time) of the estimated total inscope work in 75% of the budgeted time (47 hours of time spent out of 39.5 hours of total work time). For total in scope work I have worked

In Scope Work Items:

- Task Diagramming and planning: 2.5 hours (**Completed: 2.25 hours)**
- Create Plan for Unit Tests: 3 hours (**Complete: 2.5 hours**)
- Write the Unit Tests: 6 hours (**Complete: 1.75 Hours**)
- Write the code for the tasks: 10 hours (**Complete - 40.5 Hours so far**)
- Unit test and debug the tasks based on cutting points: 5 hours (**Complete: 1 Hour**)
- Implement the intertask communication methods: 5 hours (**Complete: 1 hour**)
- Test and debug the fully complete code: 8 hours (**Complete: 2 hours**)

  Total outlined: 39.5 hours

  Total taken: 50

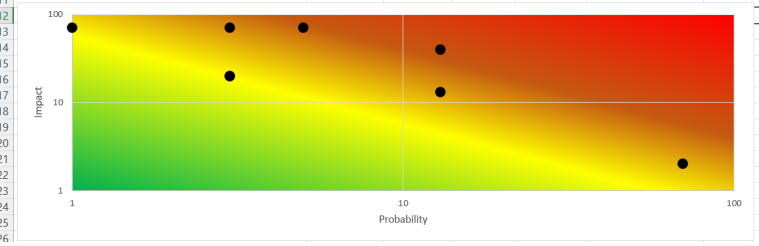
E17

	A	B	C	D
1	**In Scope Work Items:**	**Estimated Time**	**Time took**	% of estimated time that it took
2	Task Diagramming and Planning	2.5	2.25	90
3	Create Plan for Unit Tests	3	1.5	50
4	Write Unit Tests	6	1.75	29.16666667
5	Write the code for tasks	10	40.5	405
6	Unit test and debug the tasks with unit tests	5	1	20
7	Implement the Intertask communication	5	1	20
8	Test and debug fully complete project	8	2	25
9				
10				
11		**Total Estimated time:**		**Total time took:**
12		39.5		50
13				
14				
15				**Percent of total time taken:**
16				126.5822785
17				
18				
19				
20				

- ## Finishing up the code and testing:

  This week I finished up all my code and tested out some of the edge cases of the game. I have my game almost completely functional at this point and just need to add some bells and whistles to make it more playable, but I believe I have completed the base requirements of the project so far.

Item	P	I	Risk (P*	Recognize	Mitigated/ Resolve	ROAI	How
Other classes assigning a high volume of work	3	20	60	24-Mar-23	Mitigated	M	I have looked at my class schedule and have planned my lab work based on that schedule
I get sick	3	20	60	24-Mar-23	Owned	O	I will just have to work while I am sick
Having trouble with LCD display	13	40	520	24-Mar-23	Mitigated	M	I have read through some more documentation for the LCD andk know more about it, so possibility de
Having trouble inplementing the physics model	13	13	169	23-Mar-23	Mitigated	M	I have started my physics model and have an outline for it, so I feel more confident about it
My Pearl Gecko is lost or destroyed	1	70	70	7-Apr-23	Owned	O	I Own the responsibility to keep track of my Pearl Gecko
My Computer breaks	5	70	350	7-Apr-23	Owned	O	I Own the responsibility to keep my computer working
I don't have enough time to code all the required elements c	3	70	210	13-Apr-23	Owned	O	I have planned out my next two weeks so that I think that I will have enough time to complete the pro
I can't figure out how to implement that out of scope work	70	2	140	13-Apr-23	Mitigated	M	I have decreased the amount of out of scope work I orignially wanted to do
			0				



Analysis of Solution:

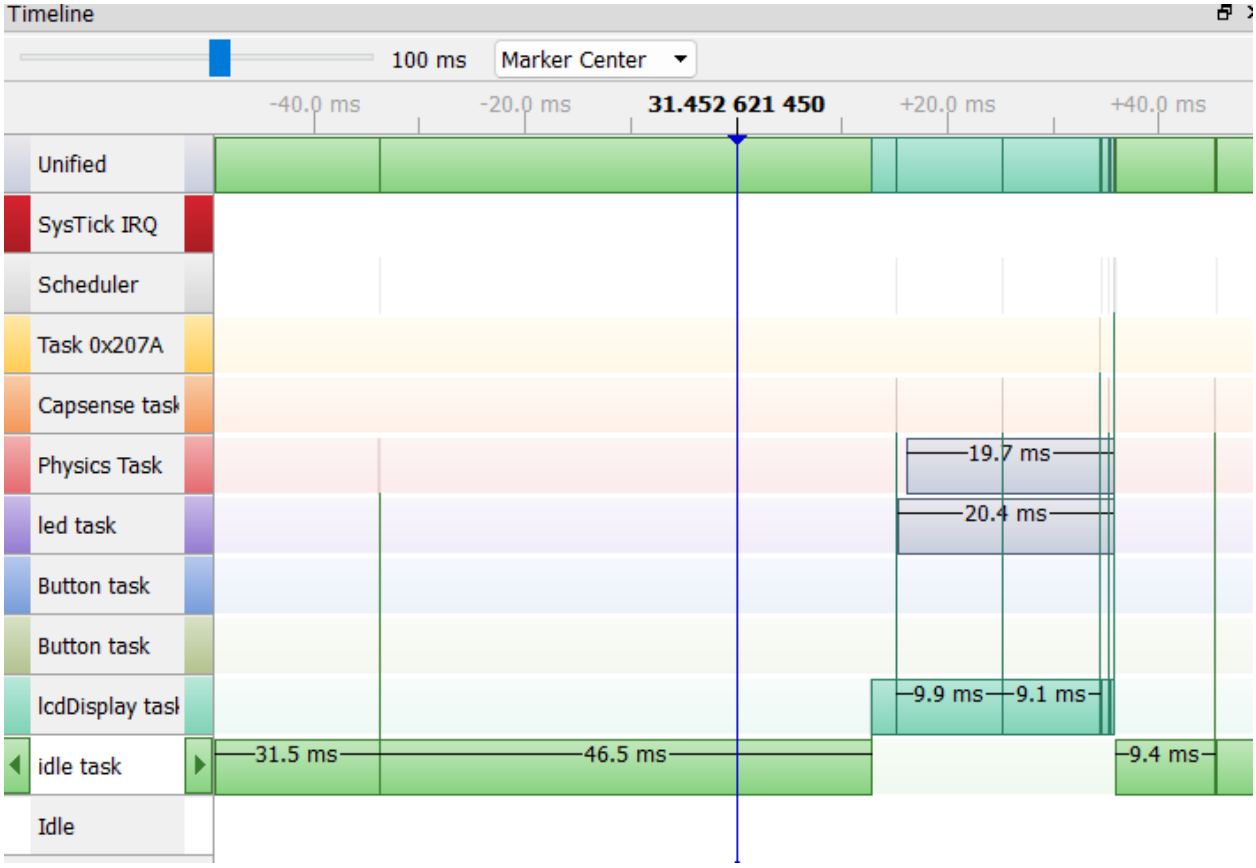
All the priorities of my tasks are the same, except for the idle task which has a lower priority. My execution times from SEGGER are what I expected and can be seen in the screenshots below. I expected the idle task to be the most frequent task. I don't see any error in the game, or the SEGGER view of the tasks execution times that would be causing issues in my program.

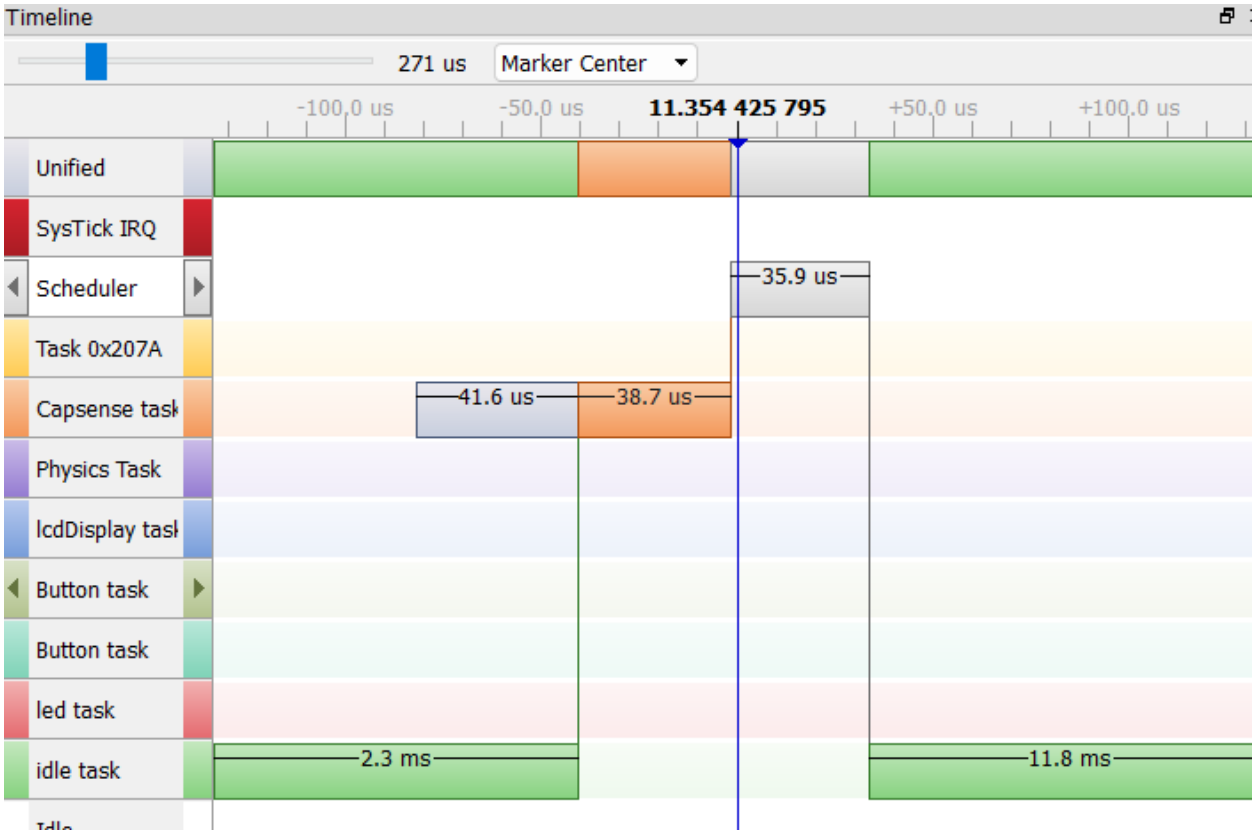
I have 800 lines of code in my app.c, which includes all the functionality of my tasks.

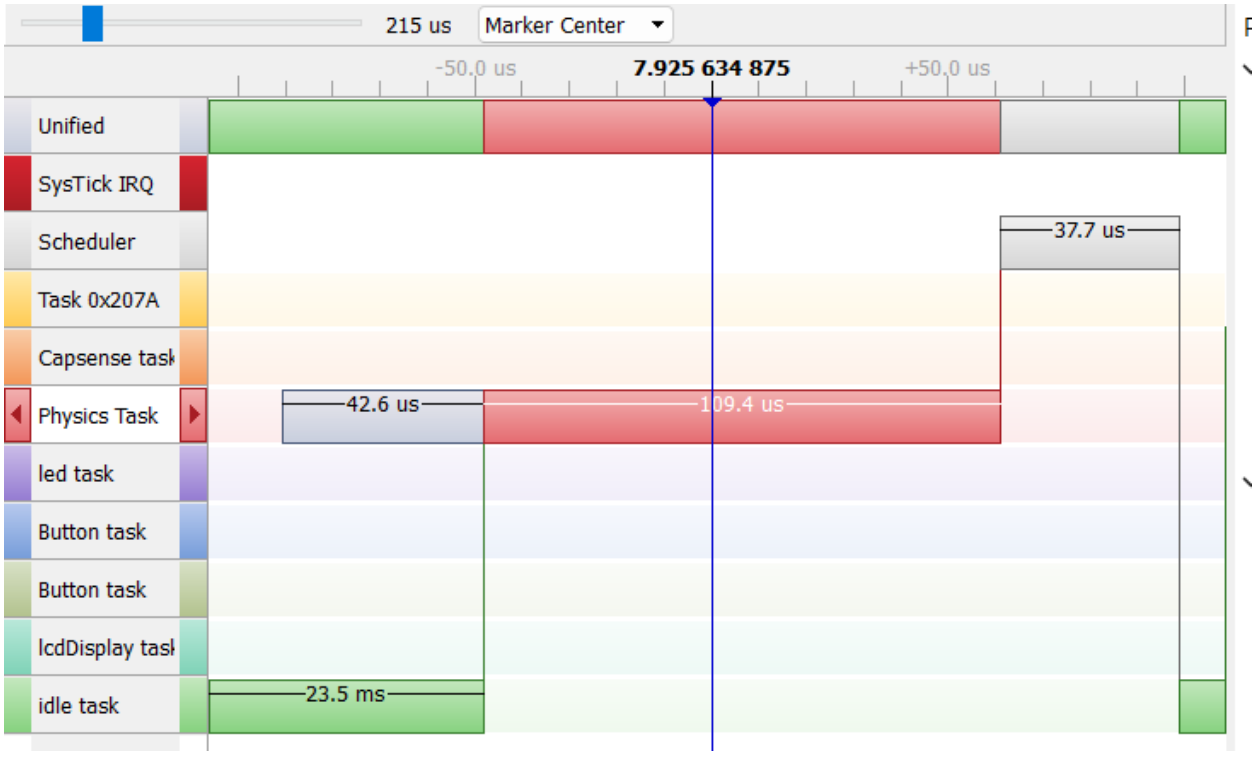
I approached the physics task as the brain of my game. I passed all the inputs for the game via a button task and a capsense task. My physics task then did all the computations for the projectiles, the satchels, the platform, and checked for the end game cases as well as edge cases. My Physics task then passed all the information needed to display everything on the lcd screen and to properly light the leds. I felt this was the best way to create my game because I could test every edge case and end game case in one task where I could mutex all the required data structures at the same time. I didn't expect to have so many issues with actually writing the physics equations so that they would work within the scale of my game. With my original numbers I would be launching a projectile into space and the platform was going close to the speed of light after a couple seconds.

I thought it would be more helpful to have some more strict and more playable configuration data given to us. I found that the original configuration data that was given to us was completely unplayable, so I ended up spending a good amount of time just decided what data configuration would actually work.

If I had another two weeks I would just spend time perfecting all the little bugs and improving my graphics I drew on the screen. I would also change some of my configuration data so that the lcd screen reflected more closely how the game worked, such as making it be when a projectile hits the edge of the castle drawn it gets destroyed instead of just checking once the projectile reached the end of the screen.


Timeline
100 ms
Marker Center
-40.0 ms
-20.0 ms
31.452 621 450
+20.0 ms
+40.0 ms
Unified
SysTick IRQ
Scheduler
Task 0x207A
Capsense task
Physics Task
led task
Button task
Button task
lcdDisplay task
idle task
Idle
19.7 ms
20.4 ms
9.9 ms
9.1 ms
31.5 ms
46.5 ms
9.4 ms


Timeline
271 us
Marker Center
-100.0 us
-50.0 us
11.354 425 795
+50.0 us
+100.0 us
Unified
SysTick IRQ
Scheduler
Task 0x207A
Capsense task
Physics Task
lcdDisplay task
Button task
Button task
led task
idle task
35.9 us
41.6 us
38.7 us
2.3 ms
11.8 ms


215 us
Marker Center
-50.0 us
7.925 634 875
+50.0 us
Unified
SysTick IRQ
Scheduler
Task 0x207A
Capsense task
Physics Task
led task
Button task
Button task
lcdDisplay task
idle task
37.7 us
42.6 us
109.4 us
23.5 ms