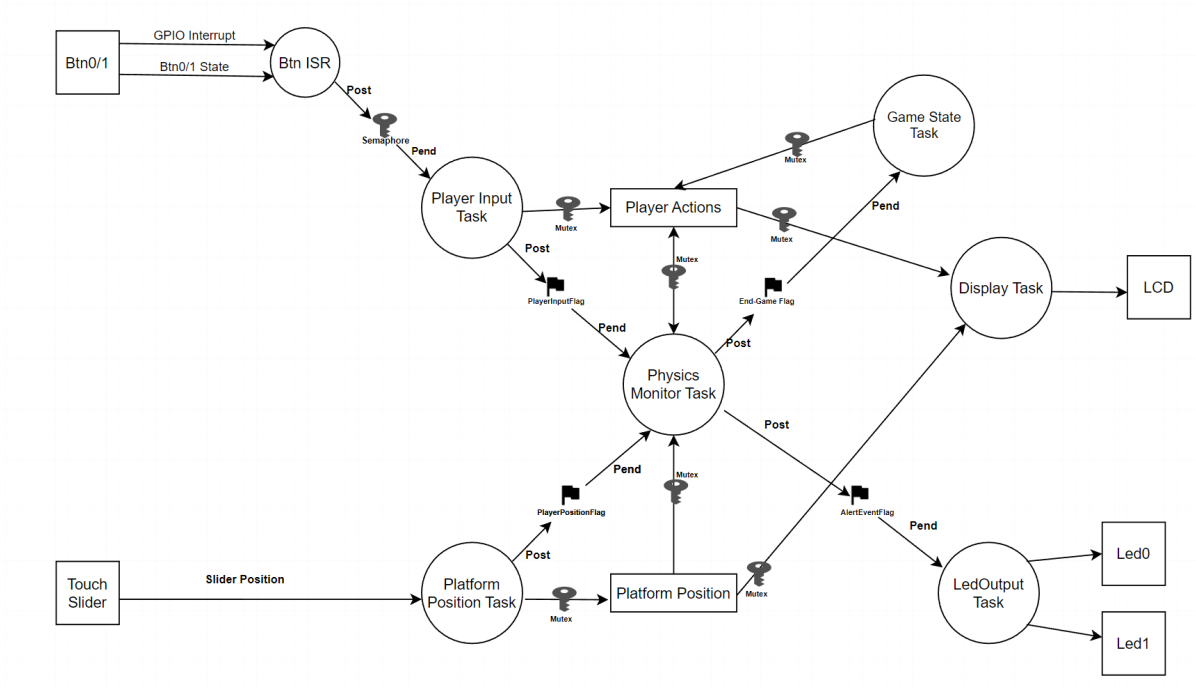
5 points:

Week 2: Task diagram, showing data flow and appropriate ITC/MutEx. (From this diagram, you should later be able to show that your design will fulfill functional requirements, and it should provide clarity about data structures between execution entities (Tasks/ISRs).)



Began implementation of task diagram into code. So far, task flow is going well and no immediate issues stood out.

5 points: Test Plan and results (3 sections: Unit Tests, Functional Tests, and Summary of tests' conditions)

Week 3: Review your described unit tests, and speculate whether each would Pass or Fail based on your current implementation. Add a section to your test plan for "functional tests", with 10 tests briefly described, and speculate whether they would pass or not as well. (Pass/Fail: likely 80+% will be "Fail" at this point.

- Unit Test 1: I will be testing only the buttons and their functionality. By cutting off my Physics Monitor Task and Platform Position Task I can isolate my Player Input task. This will allow me to see if the button interrupts are correctly being applied and correctly setting the player actions data struct.
    - **PASS** Button Test 1: Press either button and check if the player action data struct is being correctly updated in the registers.
    - **PASS** Button Test 2: Press and hold either button and check if the player action data struct is being set correctly.

- Unit Test 2: I will be testing only the touch slider and its functionality. By cutting off my Physics Monitor Task and Player input task, I will be isolating my platform position task and touch slider. Doing this will allow me to see if my platform position data structure gets updated correctly when touching the capsense slider.
    - **PASS** Touch Slider Test 3: Touch the capsense slider in each of the four locations and check to see if the platform position data structure is being updated correctly.
- Unit Test 3: Once I have tested that both my buttons and touch slider are working correctly I will test the LCD screen and LCD display Task. Since I have confirmed that my buttons and touch slider work, I only need to deal with my code for the LCD screen. This makes things much easier to debug if testing goes wrong since I only have one location it can go wrong.
    - **FAIL** LCD Button Update Test 4: Press the Buttons and see if the LCD screen is correctly showing the required status bars based on the player action data structure.
    - **FAIL** LCD Touch Slider Update Test 5: Touch the capsense slider and see if the platform is moving and updating.
    - **FAIL** LCD Castle Test 6: Confirm if Castle and cliff are correctly being initialized and updated.
    - **FAIL** LCD Rail Gun Test 7: When pressing button 0 confirm if LCD shows RailGun launching projectile
- Unit Test 4: Once I have tested and confirmed that my buttons, touch slider, and LCD screen are working correctly I can begin testing my Physics Monitor Task with the LCD display and LEDs. Doing this limits the places my code could be going wrong to the Physics Monitor Task and LED task.
    - **FAIL** Physics Monitor Task Test 8: See if physics are being correctly set and updated in their corresponding data structures.
    - **FAIL** Satchel Charge Test 9: See if the satchel is correctly being used and showing on the LCD screen.
    - **FAIL** End conditions Test 10: Test the game end conditions and see if they are correctly working and ending the game.

5 points: Statement of where your project stands:

(3 points) Accurate summary statement of your functionality deliverables and usability so far.

This week I began working more on coding and implementing my design. First I started working on the buttons and testing if they updated the player action data structure correctly. When my buttons were working as I expected them

to, I began working on the capsense touch sensor and seeing if it correctly updates my platform position data structure. Then, I tested the touch slider and saw that they were working correctly as well. Finally, I started working on my LCD display, but did not get to testing it.

(2 points) Summary effort & estimate numbers.

I have completed 40% of my currently-scoped, estimated work (40 estimated for work completed thus far /100hr total estimate) in 30% of the budgeted total-project time. (30 time spent, of 100hr total estimate). For the work that has been completed, I took 0.75x (30/40) as much time as I estimated.

5 points: List of in-scope work items (NOT just _this_ week's), indicating complete or not-yet-complete, along with your estimates of how long you think they will take in total for each

WEEK 1			
Task diagram	COMPLETE	30	90
Test Plan/Cutting Points	COMPLETE	30	30
Summary statement	COMPLETE	15	30
Summary effort & estimate numbers	COMPLETE	15	30
List of in-scope work items	COMPLETE	15	30
Update risk register	COMPLETE	15	30
WEEK 2			
Task diagram Updates	COMPLETE	0	30
Unit Testing Plan	COMPLETE	60	30
Summary statement	COMPLETE	25	15
Summary effort & estimate numbers	COMPLETE	15	15
List of in-scope work items	COMPLETE	15	15
Update risk register	COMPLETE	5	15
WEEK 3			
Augment unit tests	COMPLETE	60	60
Implement Task Diagram to Code	COMPLETE	360	180
Implement Buttons	COMPLETE	450	480
Implement Touch Slider	COMPLETE	450	480
Summary Statement	COMPLETE	15	30
Summary effort & estimate numbers	COMPLETE	15	15
List of in-scope work items	COMPLETE	15	15
Update risk register	COMPLETE	15	15
WEEK 4			
Work on LCD implementation	NOT-COMPLETE	60	
Work on Physics Task Implenentation	NOT-COMPLETE		
Work on LEDs Task	NOT-COMPLETE		

5 points: Update your risk register

Item	P	I	Risk (P*I)	Recognized	Mitigated/ Resolved	ROAM	How
Health Issues	1	100	100	23-Mar-23		M	Contact Instructor for extension if possible
Gecko Technical Issues	3	100	300	23-Mar-23		O	If issues arise, Contact TA/Instructor
Motivation	40	100	4000	6-Apr-23		M	Don't procrastinate, start early and do as much as I can each day.
			0				
			0				
			0				
			0				
			0				
			0