

# **Week 4 Work Report**

## **Goals for the Week**

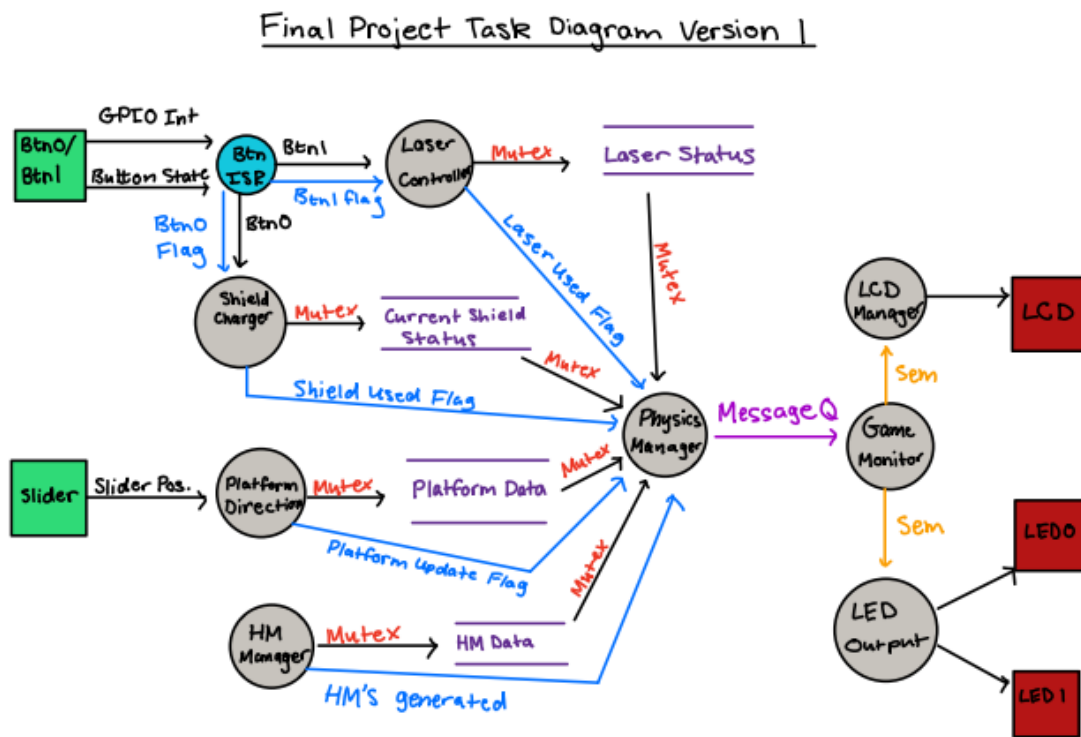
1. Intermediate Tasks
  - a. Complete a minimum of 20% or more of the items in your scoped work list each week
2. Test Plan and Results
  - a. Augment your test plan to also have a section on functional tests, and populate it with at least 10 functional tests. Summarize results. ("NotRun" will make sense for many functional tests that are not possible until you get further).
3. Current Project Status
  - a. Accurate summary statement of your functionality deliverables and usability so far.
  - b. Summary effort and estimate numbers.
4. List of in-scope work items
  - a. Indicate complete or not-yet-complete, along with your estimates of how long you think they will take in total for each.
5. Update Risk Register

## **Week 4 Work Time Estimation**

1. Intermediate Tasks: 7 - 9 Hours
  - a. This is the standard amount of work for each week. I will be breaking down my intermediate tasks into smaller pieces as I will not be able to complete entire tasks without determining how the others will work first. That will mean that parts of each task will be listed as work items now instead of the entire task as one.
2. Test Plan and Results: 2 - 5 Hours
  - a. This week's testing goal is to create a functional testing list of the tests to verify the game's functionality once completed. I will need to test each aspect of the game from inputs to physics to graphics.
3. Current Project Status: 15 - 30 Min
  - a. Simple couple sentence to paragraph update on my progress.
4. List of in-scope work items: 2.5 Hours

- a. This will take some more time this week as I need to reevaluate my time estimates for each item. I plan on recreating the entire list by breaking each item down into more manageable pieces in order to track progress more granularly.
5. Updating the risk register: 30 - 45 Min
  - a. Thinking of risks should take most of the time here along with mitigating the existing ones. Inputting them should take 5 - 10 minutes.

## Task Diagram



## Test Plan and Results (Cutting Points)

This week I needed to complete a testing plan for going forward utilizing at least 3 “cutting points”. I need to have a list of at least 10 tests. This will include Unit Tests and Functional Tests. I will create a list of each type below along with a short description of how they work. This is the initial plan going forward and is subject to change drastically upon new information or design changes.

## Unit Tests

Unit tests will cover the functionality of the code within tasks. They need to be isolated in order to unit test them, so shared data structures with other tasks will pose a challenge. The main idea behind these tests is to make sure, assuming that the ITC is working correctly, that the jobs of each task are working as expected. The tests will likely be conducted on a task by task basis with some assumptions on data structures.

1. Input Unit Test
  - a. Code to verify that inputs are being read and sent to the correct tasks
  - b. Test will consist of monitoring the inputs and verifying the correct task activating in reaction.
2. ISR Unit Test
  - a. Verification of ISR's being called upon button pushes and calling the appropriate functions with the expected output
  - b. Test will consist of verification of ISR entry and function calls via tracking code flow and marking areas of code that are activated in response to the ISR's
3. Shield Charger Task Unit Test
  - a. Verification of the inner-task behavior based on the shield button input and activation of the task via button ISR
  - b. This test will utilize one cutting point
    - i. Cutting off the Shield Charger from the physics manager
4. Laser Controller Unit Test
  - a. Verification of the inner-task behavior based on the laser button input
  - b. This test will utilize one cutting point
    - i. Cutting off the Laser Controller from the physics manager
5. Platform Direction Unit Test
  - a. Verification of the inner-task behavior based on the capsense input
  - b. This test will utilize one cutting point
    - i. Cutting off the Platform Direction task from the Physics Manager
6. HM Manager Unit Test
  - a. Verification of the creation of our HM's
  - b. This test will utilize one cutting point
    - i. Cutting off the HM Manager from the Physics Manager
7. Given Dummy Data - Physics Manager Unit Test
  - a. Verification of Physics Manager manipulating our dummy data.

## Functional Tests Plan

Functional tests will cover the behavior of the outputs based on a specific set of inputs. These tests will involve testing the behavior of the system when taking inputs. These tests will be periodically performed as the project progresses. This way we can make sure that future changes are not affecting already existing code.

1. Platform Movement
  - a. We want to observe that utilizing the slider we can see accurate and responsive movement of the platform
2. Shield Behavior
  - a. We want to observe the shield utilization via the button press along with the charging behavior
3. HM Behavior
  - a. We want to observe the creation and launching of HM's according to the parameters created by the HM. This will have less inputs than the other functional tests.
4. Final Game Behavior
  - a. Here we want to test a combination of all our previous tests. We need to rigorously test our final project with every piece working together.

## Functional Tests

1. Platform Movement Tests - No Run
  - a. Observe upon reset that the platform is in the middle of the screen. Place your finger on the far left side of the slider. Observe that the platform moves in that direction. When it hits the wall it should just stay there.
  - b. Observe upon reset that the platform is in the middle of the screen. Place your finger on the middle left side of the slider. Observe that the platform moves in that direction but slower than the far left test. When it hits the wall it should just stay there.
  - c. Observe upon reset that the platform is in the middle of the screen. Place your finger on the far right side of the slider. Observe that the platform moves in that direction. When it hits the wall it should just stay there.
  - d. Observe upon reset that the platform is in the middle of the screen. Place your finger on the middle right side of the slider. Observe that the platform moves in that direction but slower than the far right test. When it hits the wall it should just stay there.

## 2. Shield Engagement Tests - No Run

- a. Press the left button. Observe that the shield engages around the platform. Then release the button and observe that the shield disengages.
- b. Press the left button and hold it for 3 seconds. Observe that the shield disengages after 3 seconds despite that the button is still being held.
- c. Press the left button and observe the shield engage. Then release the left button and observe the shield disengagement. Then press the button shortly after releasing it and observe that it does not re-engage due to needing to charge.

## 3. Laser Engagement Tests - No Run

- a. Observe at reset a single HM is deployed from the top of the screen. Press the right button and release it. Observe that the HM is removed from the field of play and a new one replaces it if it is not the last one.
- b. From reset, engage the shield by pressing the left button. While holding the left button, press the right button and observe that the game ends due to the laser being engaged while the shield is active.

## 4. HM Tests - No Run

- a. Observe at reset a single HM is released from the top of the screen and begins falling. Allow the HM to hit the ground without positioning the platform to block it. Observe that the game ends due to missing the HM.
- b. Observe at reset a single HM is released from the top of the screen and begins falling. Position the shield to block the HM from hitting the ground and engage the shield by hitting the left button a second or two before it makes contact with the platform. Observe that if the HM is moving fast enough that the shield causes the HM to bounce away from the platform and towards the top of the screen. Observe that if the HM leaves the screen that a new HM is generated and begins falling.
- c. Observe at reset a single HM is released from the top of the screen and begins falling. Position the shield to block the HM from hitting the ground and let the HM hit the platform. Observe that if the HM is moving fast enough that the platform breaks and the game ends due to the platform breaking.

## 5. Bounce Physics Testing - No Run

- a. Observe at reset a single HM is released and when it hits an object like the walls of the canyon or the platform that the HM bounces off and loses speed unless boosted by the shield.
  - b. Observe that when the shield is engaged and an HM hits it at sufficient speed that it is boosted bounced away from the shield upwards.
- 6. Slow Shield Condition - No Run
  - a. Slow an HM down and observe that if the HM is moving too slowly when hitting the shield that it falls through it and the game ends due to the HM hitting the platform.
- 7. Auto Laser Defense Testing - No Run
  - a. Observe at reset a single HM is released and begins falling to the ground. Position the platform using the slider so that the HM will not hit the platform and will hit the ground. Observe that as the HM is about to hit the ground that the laser defense system is automatically engaged and the HM is destroyed.
- 8. Scoring Testing - No Run
  - a. Observe that at reset the score is displayed on the screen and is set to a value of zero.
  - b. Observe that when an HM is destroyed or bounced out of the canyon that the score increases.
  - c. Observe that the final score is displayed on the victory screen after the user wins
  - d. Observe that the final score is displayed on the loss screen after the user loses.
- 9. Graphical Testing - No Run
  - a. Observe at reset that the field of play is set up correctly: Platform in the middle bottom of the screen, walls on both sides of the screen, score at the top, and HM's being generated and dropped from the top.
  - b. Observe that HM's disappear when launched through the top of the screen and a new one is generated when it leaves. Also observe the score increase.
- 10. Win Conditions Testing - No Run

- a. Observe that the game ends in a win condition when all of the HM's have been bounced out of the canyon or destroyed by the laser. The screen should update with a message communicating your score and congratulating the user.

### Testing Plan Summary

As of this week, I am implementing the general plan of what needs to be tested as I develop the project. I will include more detailed explanations of each test step-by-step once we get to the implementation stage. For now, none of the tests are implemented. Tests may be added or removed as the circumstances and specifications of my design change. As of the end of Week 3, I have a solid understanding of what modules need to be tested and a general idea of how.

### Test Implementation

I did not get to implement most of my testing plan this week. I am currently struggling to keep up with my set deadlines and need to reevaluate my approach to this project. I have decided to implement compile-time switches to enable individual unit tests. There will be basically two sections to each testable module: the actual code for that section and a testing section that will try to replicate the behavior of the actual code and run tests on it.

## **Current Project Status Week 4**

Week 4 was a large clean-up week for me. I began implementing several other tasks in order to set up for tackling the physics task next week. I am coding smaller parts of each task at a time now in order to get the game to the simplest functioning state by the end of next week. I still have some more catching up to do, but I have made serious progress this week getting ready to get a barebones version of the game running.

I have completed 20.6% (estimated completed / estimated total) of my currently-scoped, estimated work (26.51 actually spent / 128.7hr total estimate) in % of the initially-estimated time. (16.5 estimated for the items I have completed, of 44hr total estimate). For the work that has been completed, I took 1.34x (25.25/18.75) as much time as I estimated.

Given the scope changes I've made and agreed to with the instructor, my latest scope is 90% of my original scope (36 hrs, vs. 40 hrs). After this week's report, I will use my new scope in my summary statements.

## **List of In-Scope Work Items**

Here is my organized list of all of the things that need to be done in order to complete this project along with documentation. Some are vague temporarily so when they appear in the upcoming weeks I can update them with specifics. I am going to be giving each item a generous amount of time since humans are terrible at predicting how much time a task will take.

### **Total List**

#### **Programming**

1. Create Project and File Structure - Incomplete
  - a. Create actual project (15 - 30 Min) - Complete
  - b. Header files for each task (15 - 45 Min) - Complete
  - c. Supporting C files (15 - 30 Min) -Complete
  - d. Includes (15 - 30 Min) -Complete
  - e. GPIO (15 - 30 Min) - Complete
  - f. OS setup (15 Min) - Complete
  - g. Additional features () - Incomplete
2. Setup Button Inputs and ISR's - Incomplete
  - a. Enable Interrupts (5 Min) - Complete
  - b. Create interrupt handlers (15 Min) - Complete
  - c. Create button function to wake up tasks (30 Min - 1.5 Hour) - Complete
  - d. Additional features () - Incomplete
3. Initialize Data Structures for the game - Incomplete
  - a. Platform basics (15 Min) - Complete
  - b. Shield basics (15 Min) - Complete
  - c. HM basics (15 Min) - Complete
  - d. Laser basics (15 Min) - Complete
  - e. Game basics (15 Min) - Complete



- f. Velocity basics (10 Min) - Complete
  - g. Force basics (10 Min) - Incomplete
  - h. Additional features (3 Hours) - Incomplete
  - i. Configurable game data defines (3 Hours) - Incomplete
  - j. Additional features () - Incomplete
4. Initialize ITC's and Mutex's - Incomplete
- a. Laser ITC (15 - 30 Min) - Complete
  - b. Laser Mutex (15 - 30 Min) - Complete
  - c. Platform ITC (15 - 30 Min) - Complete
  - d. Platform Mutex (15 - 30 Min) - Complete
  - e. Shield ITC (15 - 30 Min) - Complete
  - f. Shield Mutex (15 - 30 Min) - Complete
  - g. Laser ITC (15 - 30 Min) - Complete
  - h. Laser Mutex (15 - 30 Min) - Complete
  - i. Additional features () - Incomplete
5. Complete Shield Task - Incomplete
- a. Shield Data Structure (15 Min) - Complete
  - b. Shield task activation (15 - 30 Min) - Complete
  - c. Shield data protection (15 - 30 Min) - Complete
  - d. Shield duration timing (30 Min - 1.5 Hours) - Complete
  - e. Shield recharge timing (30 Min - 1.5 Hours) - Incomplete
  - f. Additional features () - Incomplete
6. Complete Laser Task - Incomplete
- a. Laser Data Structure (15 Min) - Complete
  - b. Laser task activation (15 - 30 Min) - Complete
  - c. Laser data protection (15 - 30 Min) - Complete
  - d. Laser activation (30 Min - 1 Hour) - Complete
  - e. Laser count (15 - 30 Min) - Complete
  - f. Laser auto engagement (1 - 3 Hours) - Incomplete
  - g. Additional features () - Incomplete
7. Complete Platform Task - Incomplete
- a. Platform Data Structure (15 Min) - Complete

- b. Task activation code (15 - 30 Min) - Complete
  - c. Platform Inputs (15 - 30 Min) - Complete
  - d. Platform data protection (15 - 30 Min) - Complete
  - e. Platform Force calculation (15 - 30 Min) - Complete
  - f. Platform Bounce (1 - 3 Hours) - Incomplete
  - g. Additional features () - Incomplete
8. Complete HM Code - Incomplete
- a. HM Data structure (15 Min) - Incomplete
  - b. HM ITC (15 - 30 Min) - Incomplete
  - c. HM size/shape (conceptual) (30 Min - 1 Hour) - Incomplete
  - d. HM size/shape (code) (1 - 3 Hours) - Incomplete
  - e. HM set for game (30 Min - 2 Hours) - Incomplete
  - f. Addition HM sets (1 - 2 Hours) - Incomplete
9. Complete Physics Task - Incomplete
- a. Object forces (1 - 2.5 Hours) - Incomplete
  - b. Object positions (1 Hour - 2 Hours) - Incomplete
  - c. Object velocities (1 Hour - 2 Hours) - Incomplete
  - d. Object Collisions (1 - 4 Hours) - Incomplete
  - e. Object removal (1 - 2 Hours) - Incomplete
10. Complete Game Monitor Task - Incomplete
- a. Scoring (1 -2 Hours) - Incomplete
  - b. Win Conditions (1 - 2 Hours) - Incomplete
  - c. Lose Conditions (1 - 2 Hours) - Incomplete
  - d. Menus (1 - 4 Hours) - Incomplete
  - e. Game Configuration (1 - 4 Hours) - Incomplete
11. Complete LCD Task - Incomplete
- a. Wall Sprites (30 Min - 2 Hours) - Incomplete
  - b. HM Sprites (30 Min - 2 Hours) - Incomplete
  - c. Platform Sprite (30 Min - 2 Hours) - Incomplete
  - d. Laser Sprite (30 Min - 2 Hours) - Incomplete
  - e. Laser Animation (1 - 3 Hours) - Incomplete
  - f. Shield Animation (1 - 3 Hours) - Incomplete
  - g. HM Animation (1 - 3 Hours) - Incomplete

h. Platform Animation (1 - 3 Hours) - Incomplete

12. Complete LED Task - Incomplete

- a. PWM for shield force LED (Left LED) (1 - 3 Hours) - Incomplete
- b. PWM for force to apply to the platform to intersect the HM (Right LED) (1 - 3 Hours) - Incomplete
- c. Blinking Loss condition (Right LED) (1 - 3 Hours) - Incomplete

Total Hours:

Reports

- 1. Task Diagram (1.5 - 2.5 Hours) - Complete
- 2. Week 1 Report (1.5 Hours) - Complete
- 3. Week 2 Report (1.5 Hours) - Complete
- 4. Week 3 Report (1.5 Hours) - Complete
- 5. Week 4 Report (1.5 Hours) - Complete
- 6. Week 5 Report (1.5 Hours) - Incomplete
- 7. Week 6 Report (1.5 Hours) - Incomplete

Risk Registers

- 1. Week 1 Risk Register Update (10 - 20 Min) - Complete
- 2. Week 2 Risk Register Update (10 - 20 Min) - Complete
- 3. Week 3 Risk Register Update (10 - 20 Min) - Incomplete
- 4. Week 4 Risk Register Update (10 - 20 Min) - Incomplete
- 5. Week 5 Risk Register Update (10 - 20 Min) - Incomplete
- 6. Week 6 Risk Register Update (10 - 20 Min) - Incomplete

Status Updates

- 1. Week 1 Status Update (10 - 20 Min) - Complete
- 2. Week 2 Status Update (10 - 20 Min) - Complete
- 3. Week 3 Status Update (10 - 20 Min) - Complete
- 4. Week 4 Status Update (10 - 20 Min) - Complete
- 5. Week 5 Status Update (10 - 20 Min) - Incomplete

6. Week 6 Status Update (10 - 20 Min) - Incomplete

### Testing

1. Week 1 Testing (10 - 20 Min) - Complete
2. Week 2 Testing (3 - 4 Hours) - Complete
3. Week 3 Testing (3 - 4 Hours) - Complete
4. Week 4 Testing (3 - 4 Hours) - Complete
5. Week 5 Testing (3 - 4 Hours) - Incomplete
6. Week 6 Testing (3 - 4 Hours) - Incomplete

### **Week 4 List**

1. Laser ITC (15 - 30 Min) - Complete
2. Laser Mutex (15 - 30 Min) - Complete
3. Platform ITC (15 - 30 Min) - Complete
4. Platform Mutex (15 - 30 Min) - Complete
5. Shield ITC (15 - 30 Min) - Complete
6. Shield Mutex (15 - 30 Min) - Complete
7. Laser ITC (15 - 30 Min) - Complete
8. Laser Mutex (15 - 30 Min) - Complete
9. Shield Data Structure (15 Min) - Complete
10. Shield task activation (15 - 30 Min) - Complete
11. Shield data protection (15 - 30 Min) - Complete
12. Shield duration timing (30 Min - 1.5 Hours) - Complete
13. Laser Data Structure (15 Min) - Complete
14. Laser task activation (15 - 30 Min) - Complete
15. Laser data protection (15 - 30 Min) - Complete
16. Laser activation (30 Min - 1 Hour) - Complete
17. Laser count (15 - 30 Min) - Complete
18. Platform Data Structure (15 Min) - Complete
19. Task activation code (15 - 30 Min) - Complete
20. Platform Inputs (15 - 30 Min) - Complete
21. Platform data protection (15 - 30 Min) - Complete
22. Platform Force calculation (15 - 30 Min) - Complete
23. Week 4 Report (1.5 Hours) - Complete

- 24. Week 4 Risk Register Update (10 - 20 Min) - Incomplete
- 25. Week 4 Status Update (10 - 20 Min) - Complete
- 26. Week 4 Testing (3 - 4 Hours) - Complete

Total Week 4 Estimated hours: 17.91 Hours

Total Week 4 Hours Used: 16.5 Hours

### **Updating Risk Register**

This week for the risk register, I mitigated my planning risk by overhauling my work list and numbers. This might affect my current progress numbers in a negative way, but it will be a much better way of tracking my work and progressing through the project. I only have two more weeks to work on the project so there are not many other risks to add. I only need to mitigate or resolve the ones that I have.