**Week1:**

I finished the task diagram. And I have created the risk register and have identified 2 cutting points.

I have completed 7.5% of my currently scoped. (3hr actually spent /40hr total estimate).

The estimated time is still about the same as initially estimated time. I will keep using this scope in my summary statements.

**Week2:**

This week, I have implemented the button 0 part, which includes the measurement of arming time and measurement of recharge time. Moreover, I have played around with the LCD. At this point, I can draw the canyon wall, masses and platform on the LCD. And I can let the mass fall from the top of the screen at some constant x-speed and y-speed.

The scope of this project is updated to 47 hr in total in this week

I have completed 13% of my currently scoped. (6hr actually spent /47hr total estimate).

I will keep using this scope in my summary statements.

**Week3:**

This week, I have implemented the shield and slider part, which includes the movement of shield and slider operation. Shield can rebounce from the Canyon Wall with same speed. I haven’t applied the real-world physics on it yet. Moreover, I have built some unit tests for the physics task, which includes x-axis position change and y-axis position change, kinetic energy calculation and x-axis velocity change and y-axis velocity change.

The scope of this project is updated to 42 hr in total in this week

I have completed 31% of my currently scoped. (13hr actually spent /42hr total estimate).

I will keep using this scope in my summary statements.

Week4:

This week, I have implemented the physics task, which includes the movement of shield and ball. Shield and Ball can rebounce from the Canyon Wall with same speed. I have applied the real-world physics on it by using the velocity formula. Moreover, I have re-built some unit tests for the physics task because different velocity had been used in the code, which includes x-axis position change and y-axis position change, kinetic energy calculation and x-axis velocity change and y-axis velocity change.

Moreover, my platform can catch the ball few times without boost function. I plan to implement the rest of the project in the next few weeks.

The scope of this project is updated to 28 hrs in total in this week. I was overestimated the difficulty of this project before

I have completed 57% of my currently scoped. (16hr actually spent /28hr total estimate).

I will keep using this scope in my summary statements.

Week5:

This week, I have improved the physics task and slider task. Now, they are more playable. And I have also added another two balls in the game with different diameter, initial velocity, acceleration etc. Thus, the only way to win the game is you successfully catch all the ball. Once you miss one of them, you fail the game. Beyond that, I have added some pattern for “Success” and “Fail”

The scope of this project is updated to 28 hrs in total in this week.

I have completed 82% of my currently scoped. (23hr actually spent /28hr total estimate).

I will keep using this scope in my summary statements.

Week6:

This week, I almost done everything. The only thing left is that using given data structure values in my project.

I think one more thing I need to do is change the frequency of my OS timer tick

The scope of this project is updated to 28 hrs in total in this week.

I have completed 96.4% of my currently scoped. (27hr actually spent /28hr total estimate).

I will keep using this scope in my summary statements.

Week7:

This week, I have completed my project except the led that used to predict the impact. Additionally, I have added some test on LCD to display the speed of mass and the speed of platform, which are helpful to visualize all the data.

Laser function and Boost function are well-implemented. The player can use boost with 0.1s cooling period and can use laser 1 time in each game.

Rebounce function for the HM and platform are well-implemented. Whenever the HM or platform have impact with the canyon wall, they will change direction with some slower x-axis speed.

The game has three data set, which means the player need to fight the HM back for three times in order to win the game. Once the player miss any HMs without using laser, the player lose the game.

Beyond that, my project passes all my function tests listed below.

1. Direction of slider by controlling the Capsense. Success when the player can control the direction of the platform in either right or left.
2. HM will pass through the platform when its velocity is less than some threshold.
3. Laser function. Success when the current HM on LCD is gone.
4. Boost function. Success when the current HM rebounce with faster speed.
5. Wall rebounce. Success when the platform will rebounce from the canyon wall.
6. KE reduction. Success when the HM’s speed is getting slower after impact with the platform.
7. LCD for Game over and won the game.

I have completed 100% of my currently-scoped,estimated work(28 hrs) in 60% of the initially-estimated time(28 of 47). (Note: It’s 100 percent because I have implemented the calculation function for prediction LED, but the result of the calculation doesn’t work in the way as I expected before.)