 RT tasks: What are your priorities, execution times vs. deadlines as seen with Segger SystemView? (Include screenshots)  Conflicts seen that kept it from operating as you planned?  (3)

Graphical user interface

Description automatically generated

The priorities:

Capsense 5

Physics 20

LCD 20

There are no conflicts that would cause gameplay problems. This is seen by how the CPU is mostly idle, the tasks are blocked for miniscule amounts of time and the fact that all tasks were able to execute properly.

The physics task and LCD tasks needed about 10ms to execute at maximum. The other tasks are under one ms.

Graphical user interface, application, table, Word

Description automatically generated

The LCD task may usually finish within 5 ms whereas interestingly the physics task usually runs under one ms but in extreme cases goes to about 10ms.

The LCD is the most intensive load.

I’m confused what code space means. I’ll talk to the professor after submission but I’m hoping that all of the data is in this file

Physics:

Standard kinematics equations were used

X = X\_0 + ut + 0.5at^2

Where a = g for mass and the applied acceleration for shield calculated form F= ma

V = u + at

The challenges were:

1. I could not use functions such as sqrt during actual execution out of risk that everything would be slowed down.
2. Overshoot needed to be accounted for. If the check occurs too late the mass may have overshot, and the render will happen with the mass on screen being outside of bounds but there still being a collision detected
3. I had not initially prepped the code for configuration, so I had to find ways to use the configuration without breaking the engine

Ranges:

All this is in relation to 100000cm canyon size. Please note that it is mathematically possible to calculate the absolute limits of playability but that would not help the user, so I have calculated these on a more practical level.

1. A gravity of 9800 is playable albeit very difficult. Over 10,000 it is too fast to be playable.
2. The force needs to neither too low for playability nor too high. A small triple digit mass platform should have a force in five-digit ranges otherwise the shield moves too fast to play. At four digits it is too slow to play. Logically we can conclude that for each possible mass, the force should have at least three more digits for optimal playability.
3. The KE reduction should be over 50% is we reasonably want to play the game otherwise the mass keeps dropping too fast to manage.
4. The KE gain should be at least 20% so that it is possible to push out the mass in a reasonable amount of time
5. The initial velocity of 4000 in the configuration was again too difficult to play with. The reasonable velocity range is between 0 to 3000 (difficult but possible). Ideally around 2000.
6. The length should be over 10,000 or the platform is too small to play. A platform size above 50,000 can lead to a condition that the player may be unengaged but still be able to deflect masses
7. The display diameter should at least be 10,000 to suitably render and play. Ideally should be below 20,000.

Plans if I had 2 more weeks:

Animations for better user feedback to play.

Cleanup the code to make it more understandable and easier to edit.

Adding randomization to allow arcade gameplay. This would also help find edge cases that I may have missed.

Allowing replay without using the reset button because it would allow debugging after game over and make the game enjoyable