**Self Reference Links**

**CS_RiskRegister.xlsx**

**CS_IN_SCOPE_ITEMS**

10 points:

**Demonstrate your working project, in our last week of lectures and lab (4 points). Show your personalizations, and share your difficulties and proud achievements.**
*Should be considered passing*

**Include at least 3 functional test demonstrations (1 point each, which the instructional staff may select from your previous week's submission)**
*Should be considered passing*

**Demonstrate full-acceleration towards the initially-farthest canyon wall (should take less than 4s even if starting on an edge: 1 point; should survive impact: 1 point; further constructive acceleration should destroy platform: 1 point)**
*Should be considered passing*

**4 points: Statement of where your project stands:**

I did not end up adding additional optional features to enhance the game, and concluded the project as it exists now, where I believe it is in an overall successful and finished state, despite this, there could certainly be many further improvements

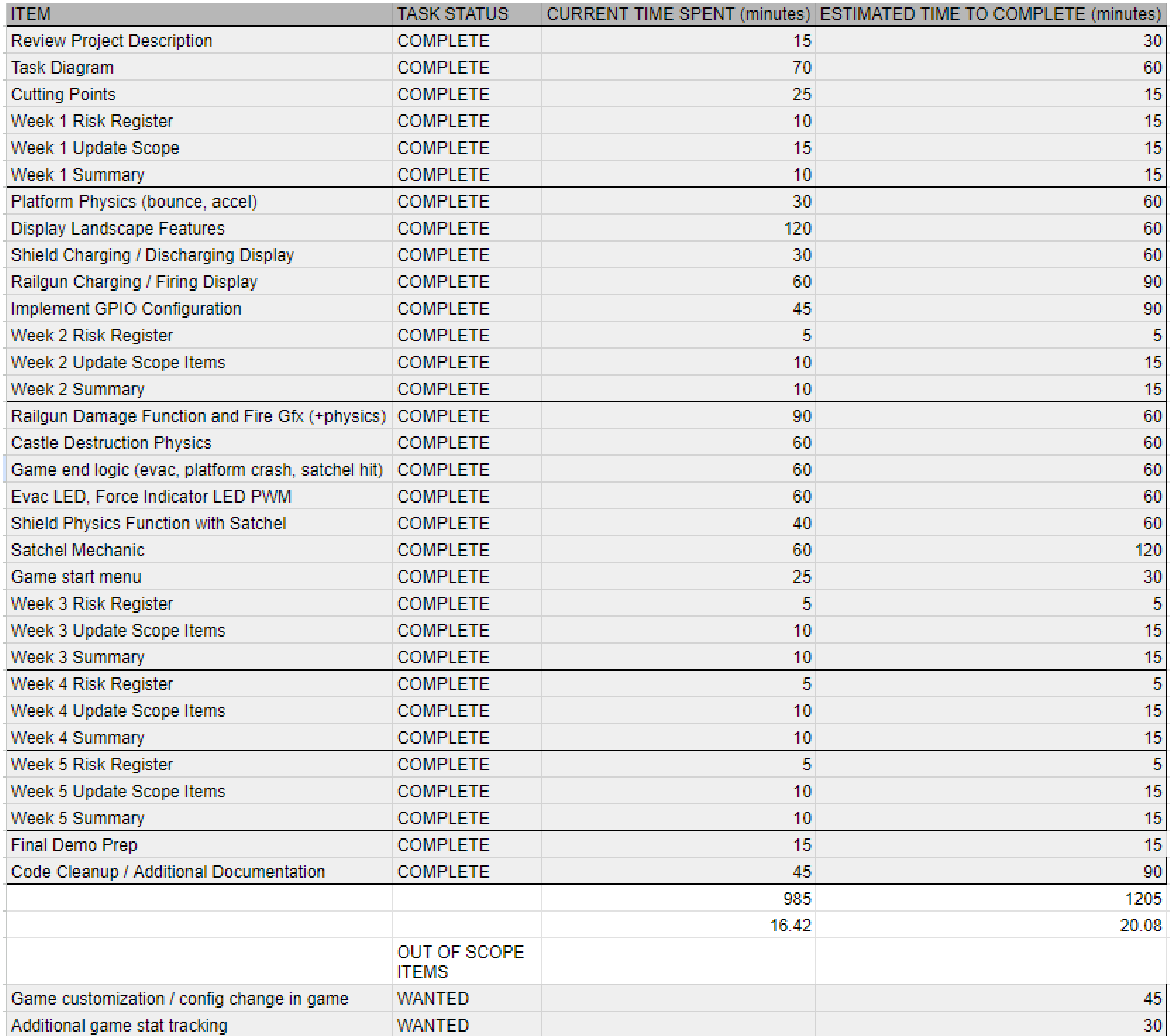
**(2 points) Accurate summary statement of your functionality deliverables and usability at the end of the class. Examples are available in prior weeks.**

**(2 points) Summary effort & estimate numbers. You get this credit simply for updating the numbers--I will not be grading you on the accuracy of your numbers! (It is also possible that you've not finished all of the in-scope work, and your points here are about being honest about that, not about reporting 100%.) This is for your learning about yourself. Note that the numerators, percentages, and multiplier in the paragraph below would update every week.**

.

I have made final updates to my in-scope work items and also updated the Risk register in accordance with the end of the project overall. I have finished about 97% of the project ( 18.43hr estimated / 19hr total estimate) in 78% of the budgeted total-project time. 16.15hr taken / 19hr total estimate. For the work that was completed during the course of the project, I took .85x (16.15/19) as much time as I initially had estimated.

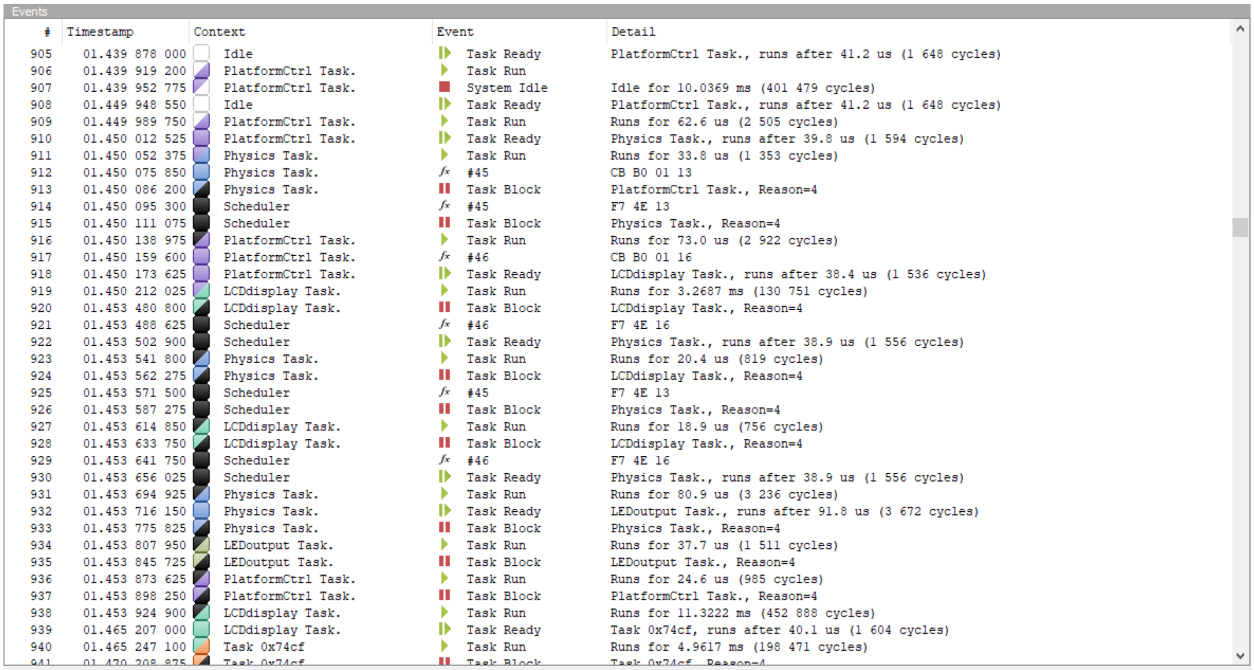
**1 point: List of in-scope work items, indicating complete or not-yet-complete, along with your estimates of how long you thought they will take in total for each (i.e. no partial-item updates). It's fine if you also list things out-of-scope that you WANTED to do, but do not include them in denominators for progress tracking, nor in numerators unless all REQUIRED work is complete--at which point you can append an extra statement to your summary effort & estimate section that includes the out-of-scope items you did as well.) Identify which work items in the list were completed since your last week's report by highlighting the "complete" status in newly-completed lines) Provide a summary statement or description for each of these items (older summary statements are removed from this week's report), and for your own learning-- if the actual time spent on one is far different than your estimate, consider WHY. Examples available in prior weeks.**



| ITEM | TASK STATUS | CURRENT TIME SPENT (minutes) | ESTIMATED TIME TO COMPLETE (minutes) |
|---|---|---|---|
| Review Project Description | COMPLETE | 15 | 30 |
| Task Diagram | COMPLETE | 70 | 60 |
| Cutting Points | COMPLETE | 25 | 15 |
| Week 1 Risk Register | COMPLETE | 10 | 15 |
| Week 1 Update Scope | COMPLETE | 15 | 15 |
| Week 1 Summary | COMPLETE | 10 | 15 |
| Platform Physics (bounce, accel) | COMPLETE | 30 | 60 |
| Display Landscape Features | COMPLETE | 120 | 60 |
| Shield Charging / Discharging Display | COMPLETE | 30 | 60 |
| Railgun Charging / Firing Display | COMPLETE | 60 | 90 |
| Implement GPIO Configuration | COMPLETE | 45 | 90 |
| Week 2 Risk Register | COMPLETE | 5 | 5 |
| Week 2 Update Scope Items | COMPLETE | 10 | 15 |
| Week 2 Summary | COMPLETE | 10 | 15 |
| Railgun Damage Function and Fire Gfx (+physics) | COMPLETE | 90 | 60 |
| Castle Destruction Physics | COMPLETE | 60 | 60 |
| Game end logic (evac, platform crash, satchel hit) | COMPLETE | 60 | 60 |
| Evac LED, Force Indicator LED PWM | COMPLETE | 60 | 60 |
| Shield Physics Function with Satchel | COMPLETE | 40 | 60 |
| Satchel Mechanic | COMPLETE | 60 | 120 |
| Game start menu | COMPLETE | 25 | 30 |
| Week 3 Risk Register | COMPLETE | 5 | 5 |
| Week 3 Update Scope Items | COMPLETE | 10 | 15 |
| Week 3 Summary | COMPLETE | 10 | 15 |
| Week 4 Risk Register | COMPLETE | 5 | 5 |
| Week 4 Update Scope Items | COMPLETE | 10 | 15 |
| Week 4 Summary | COMPLETE | 10 | 15 |
| Week 5 Risk Register | COMPLETE | 5 | 5 |
| Week 5 Update Scope Items | COMPLETE | 10 | 15 |
| Week 5 Summary | COMPLETE | 10 | 15 |
| Final Demo Prep | COMPLETE | 15 | 15 |
| Code Cleanup / Additional Documentation | COMPLETE | 45 | 90 |
|  |  | 985 | 1205 |
|  |  | 16.42 | 20.08 |
|  | OUT OF SCOPE ITEMS |  |  |
| Game customization / config change in game | WANTED |  | 45 |
| Additional game stat tracking | WANTED |  | 30 |

### Analysis of your solution: (10 points)

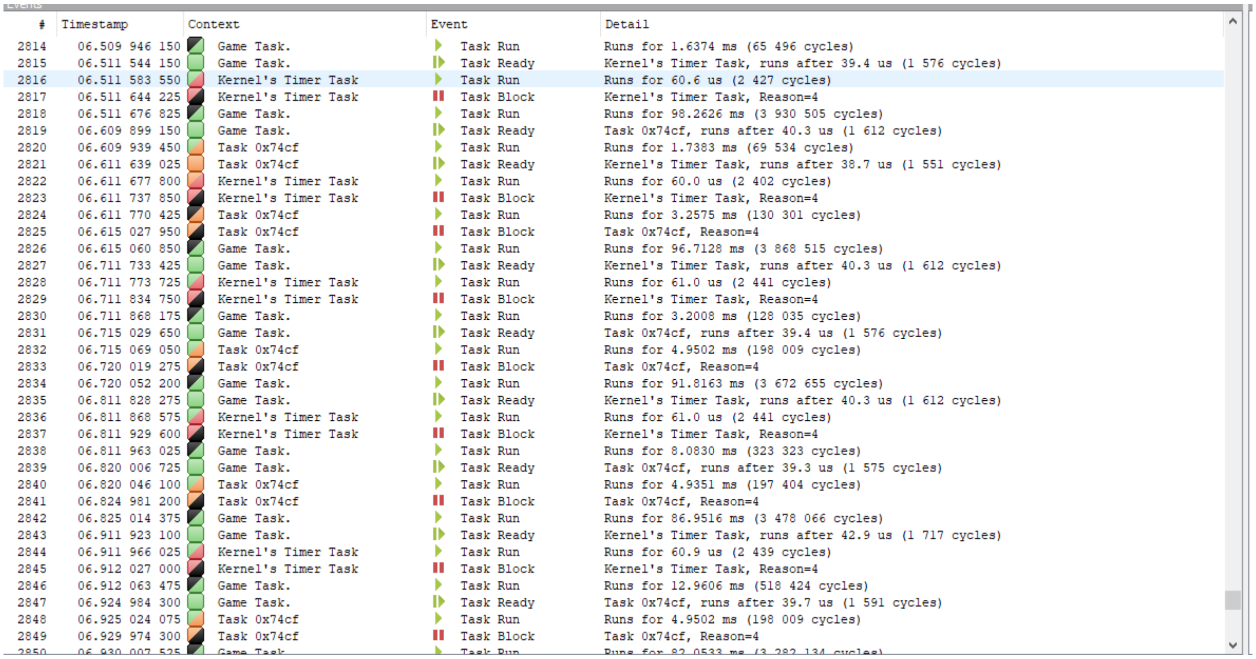
### RT tasks: What are your priorities, execution times vs. deadlines as seen with Segger SystemView? (Include screenshots)  Conflicts seen that kept it from operating as you planned?  (2)



Events

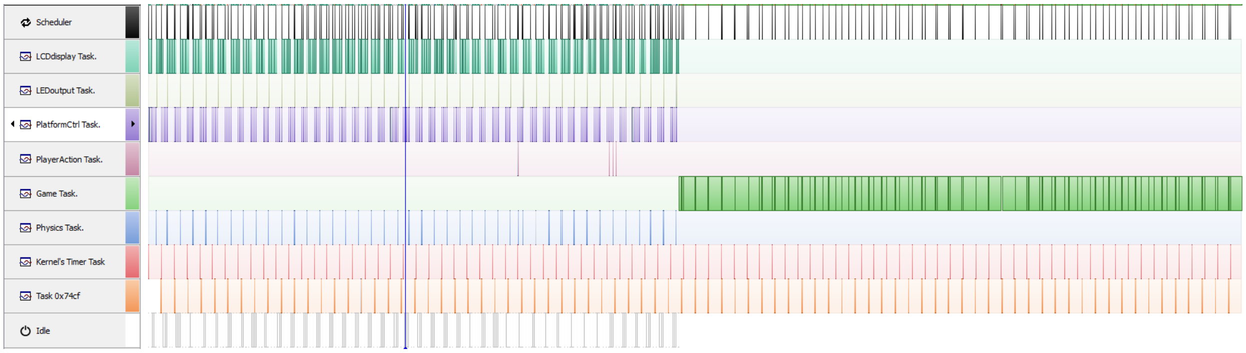
| # | Timestamp | Context | Event | Detail |
|---|---|---|---|---|
| 905 | 01.439 878 000 | Idle | Task Ready | PlatformCtrl Task., runs after 41.2 us (1 648 cycles) |
| 906 | 01.439 919 200 | PlatformCtrl Task. | Task Run | |
| 907 | 01.439 952 775 | PlatformCtrl Task. | System Idle | Idle for 10.0369 ms (401 479 cycles) |
| 908 | 01.449 948 550 | Idle | Task Ready | PlatformCtrl Task., runs after 41.2 us (1 648 cycles) |
| 909 | 01.449 989 750 | PlatformCtrl Task. | Task Run | Runs for 62.6 us (2 505 cycles) |
| 910 | 01.450 012 525 | PlatformCtrl Task. | Task Ready | Physics Task., runs after 39.8 us (1 594 cycles) |
| 911 | 01.450 052 375 | Physics Task. | Task Run | Runs for 33.8 us (1 353 cycles) |
| 912 | 01.450 075 850 | Physics Task. | #45 | CB B0 01 13 |
| 913 | 01.450 086 200 | Physics Task. | Task Block | PlatformCtrl Task., Reason=4 |
| 914 | 01.450 095 300 | Scheduler | #45 | F7 4E 13 |
| 915 | 01.450 111 075 | Scheduler | Task Block | Physics Task., Reason=4 |
| 916 | 01.450 138 975 | PlatformCtrl Task. | Task Run | Runs for 73.0 us (2 922 cycles) |
| 917 | 01.450 159 600 | PlatformCtrl Task. | #46 | CB B0 01 16 |
| 918 | 01.450 173 625 | PlatformCtrl Task. | Task Ready | LCDdisplay Task., runs after 38.4 us (1 536 cycles) |
| 919 | 01.450 212 025 | LCDdisplay Task. | Task Run | Runs for 3.2687 ms (130 751 cycles) |
| 920 | 01.453 480 800 | LCDdisplay Task. | Task Block | LCDdisplay Task., Reason=4 |
| 921 | 01.453 488 625 | Scheduler | #46 | F7 4E 16 |
| 922 | 01.453 502 900 | Scheduler | Task Ready | Physics Task., runs after 38.9 us (1 556 cycles) |
| 923 | 01.453 541 800 | Physics Task. | Task Run | Runs for 20.4 us (819 cycles) |
| 924 | 01.453 562 275 | Physics Task. | Task Block | LCDdisplay Task., Reason=4 |
| 925 | 01.453 571 500 | Scheduler | #45 | F7 4E 13 |
| 926 | 01.453 587 275 | Scheduler | Task Block | Physics Task., Reason=4 |
| 927 | 01.453 614 850 | LCDdisplay Task. | Task Run | Runs for 18.9 us (756 cycles) |
| 928 | 01.453 633 750 | LCDdisplay Task. | Task Block | LCDdisplay Task., Reason=4 |
| 929 | 01.453 641 750 | Scheduler | #46 | F7 4E 16 |
| 930 | 01.453 656 025 | Scheduler | Task Ready | Physics Task., runs after 38.9 us (1 556 cycles) |
| 931 | 01.453 694 925 | Physics Task. | Task Run | Runs for 80.9 us (3 236 cycles) |
| 932 | 01.453 716 150 | Physics Task. | Task Ready | LEDoutput Task., runs after 91.8 us (3 672 cycles) |
| 933 | 01.453 775 825 | Physics Task. | Task Block | Physics Task., Reason=4 |
| 934 | 01.453 807 950 | LEDoutput Task. | Task Run | Runs for 37.7 us (1 511 cycles) |
| 935 | 01.453 845 725 | LEDoutput Task. | Task Block | LEDoutput Task., Reason=4 |
| 936 | 01.453 873 625 | PlatformCtrl Task. | Task Run | Runs for 24.6 us (985 cycles) |
| 937 | 01.453 898 250 | PlatformCtrl Task. | Task Block | PlatformCtrl Task., Reason=4 |
| 938 | 01.453 924 900 | LCDdisplay Task. | Task Run | Runs for 11.3222 ms (452 888 cycles) |
| 939 | 01.465 207 000 | LCDdisplay Task. | Task Ready | Task 0x74cf, runs after 40.1 us (1 604 cycles) |
| 940 | 01.465 247 100 | Task 0x74cf | Task Run | Runs for 4.9617 ms (198 471 cycles) |
| 941 | 01.470 208 875 | Task 0x74cf | Task Block | Task 0x74cf, Reason=4 |

(Screenshot of regular gameplay task executions.)



```
   #   Timestamp          Context               Event         Detail
2814   06.509 946 150     Game Task.            Task Run      Runs for 1.6374 ms (65 496 cycles)
2815   06.511 544 150     Game Task.            Task Ready    Kernel's Timer Task, runs after 39.4 us (1 576 cycles)
2816   06.511 583 550     Kernel's Timer Task   Task Run      Runs for 60.6 us (2 427 cycles)
2817   06.511 644 225     Kernel's Timer Task   Task Block    Kernel's Timer Task, Reason=4
2818   06.511 676 825     Game Task.            Task Run      Runs for 98.2626 ms (3 930 505 cycles)
2819   06.609 899 150     Game Task.            Task Ready    Task 0x74cf, runs after 40.3 us (1 612 cycles)
2820   06.609 939 450     Task 0x74cf           Task Run      Runs for 1.7383 ms (69 534 cycles)
2821   06.611 639 025     Task 0x74cf           Task Ready    Kernel's Timer Task, runs after 38.7 us (1 551 cycles)
2822   06.611 677 800     Kernel's Timer Task   Task Run      Runs for 60.0 us (2 402 cycles)
2823   06.611 737 850     Kernel's Timer Task   Task Block    Kernel's Timer Task, Reason=4
2824   06.611 770 425     Task 0x74cf           Task Run      Runs for 3.2575 ms (130 301 cycles)
2825   06.615 027 950     Task 0x74cf           Task Block    Task 0x74cf, Reason=4
2826   06.615 060 850     Game Task.            Task Run      Runs for 96.7128 ms (3 868 515 cycles)
2827   06.711 733 425     Game Task.            Task Ready    Kernel's Timer Task, runs after 40.3 us (1 612 cycles)
2828   06.711 773 725     Kernel's Timer Task   Task Run      Runs for 61.0 us (2 441 cycles)
2829   06.711 834 750     Kernel's Timer Task   Task Block    Kernel's Timer Task, Reason=4
2830   06.711 868 175     Game Task.            Task Run      Runs for 3.2008 ms (128 035 cycles)
2831   06.715 029 650     Game Task.            Task Ready    Task 0x74cf, runs after 39.4 us (1 576 cycles)
2832   06.715 069 050     Task 0x74cf           Task Run      Runs for 4.9502 ms (198 009 cycles)
2833   06.720 019 275     Task 0x74cf           Task Block    Task 0x74cf, Reason=4
2834   06.720 052 200     Game Task.            Task Run      Runs for 91.8163 ms (3 672 655 cycles)
2835   06.811 828 275     Game Task.            Task Ready    Kernel's Timer Task, runs after 40.3 us (1 612 cycles)
2836   06.811 868 575     Kernel's Timer Task   Task Run      Runs for 61.0 us (2 441 cycles)
2837   06.811 929 600     Kernel's Timer Task   Task Block    Kernel's Timer Task, Reason=4
2838   06.811 963 025     Game Task.            Task Run      Runs for 8.0830 ms (323 323 cycles)
2839   06.820 006 725     Game Task.            Task Ready    Task 0x74cf, runs after 39.3 us (1 575 cycles)
2840   06.820 046 100     Task 0x74cf           Task Run      Runs for 4.9351 ms (197 404 cycles)
2841   06.824 981 200     Task 0x74cf           Task Block    Task 0x74cf, Reason=4
2842   06.825 014 375     Game Task.            Task Run      Runs for 86.9516 ms (3 478 066 cycles)
2843   06.911 923 100     Game Task.            Task Ready    Kernel's Timer Task, runs after 42.9 us (1 717 cycles)
2844   06.911 966 025     Kernel's Timer Task   Task Run      Runs for 60.9 us (2 439 cycles)
2845   06.912 027 000     Kernel's Timer Task   Task Block    Kernel's Timer Task, Reason=4
2846   06.912 063 475     Game Task.            Task Run      Runs for 12.9606 ms (518 424 cycles)
2847   06.924 984 300     Game Task.            Task Ready    Task 0x74cf, runs after 39.7 us (1 591 cycles)
2848   06.925 024 075     Task 0x74cf           Task Run      Runs for 4.9502 ms (198 009 cycles)
2849   06.929 974 300     Task 0x74cf           Task Block    Task 0x74cf, Reason=4
2850   06.930 007 525     Game Task.            Task Run      Runs for 82.0533 ms (3 282 134 cycles)
```

(Screenshot of task executions in endgame screen before restarting game with BTN press)



(Screenshot of one overall game session. Note the green task executing at the end is the game menu task once the game has ended and not been restarted.)

Game end Menu: 20

Default (all else): 22

Physics: 19

I found that I had to adjust the tau display and tau physics until it worked optimally, as well as adjust physics task priority so that the game would perform smoothly consistently.

**Code Space: how much, and evaluative comments (2)**

I used mostly good practices and had multiple code revisions over the course of the project, I separated global variables in correct areas and documented functions, function inputs, and overall followed good coding practices I believe

**Evaluation of your approach(es) to the physics update requirement. Explain your rationale for grouping the physics updates and edge case handling the way you did. What limitations did you have to deal with that were not obvious at first? (3)**

I think that with the way I considered the physics task and updating, almost all game positioning updates occur essentially simultaneously so that everything should continue as normal, rather than an object being destroyed 3 seconds after it is hit, for example, if the way that was updated was not synchronized. I realized though as I added more and more things that would be considered in the physics task, the game seemed to become laggier and overall slower at processing, which I suppose should be expected in retrospect.

**Scaling of variable spaces (think back to the lecture including "corner testing"): what ranges did you find playable? (1)**
I found that, based on all the specificities in my own project, whenever I added new configuration data, I would have to go back fairly often and make slight changes until it felt best to play, while still making sense overall. I would say I made adjustments based on that mainly, rather than more quantitative ways of considering all the physics, which is something I probably should have employed more.

**Probing questions: Which configuration data did you find unnecessary? What do you wish was more constrained instead of being undefined?**
I thought the overall project constraints were fair, but I also just decided to take some liberty with how I approached it, in making a game that worked fairly well in a reasonable amount of time. I thought some of the constraints were unnecessary or perhaps a bit unclear - but I understood the overall idea and I think my project fulfills the requirements and I'd say I ended up with a pretty working game and playable one with all the main mechanics implemented.

**What would be your next steps if you had just another 2 weeks to work on your project, and why? (2)**
If I had another 2 weeks I would work on adding the final optional in-game var adjustments and satchel drop modes, as well as debug and get more analysis for any strange issues that my game may run into, but overall it functions quite well for where it's at right now.