**Mutant Mayhem Development Log**

**2024-05-04**

I’m starting this log on Saturday May 4th, and it seems the fourth is with me! (May the fourth be with you!) I struggled for the past day and a half trying to figure out why my dictionary was unable to find keys that were created in it. I couldn’t find anything online or through AI (I started using Muse today) but managed to see online that someone had their dictionary being cleared at the end of the update(). I then made the dictionary static and voila! All is well so far.

From this “bug” I learned it is important for me to learn how to debug better. I should use more Debug.Log() to see what’s happening and should definitely look into the debug tools available in Unity.

I also learned that with a struct, you cannot go in and edit a single value, so you must create a copy of the entire struct with the desired modification and replace the entire struct. So, for my TileStats, it is now a Class, not a Struct.

I’m excited that I will be able to efficiently manage my structure tiles! I look forward to adding to the BuildingSystem and making an interactive base!

**Let get up to speed:**

Up to today I’ve been learning and tinkering in Unity for about a month. I’m using AI probably too much and can easily get lost in adjusting fine details such as my particle systems and effects!! I feel I could improve my learning technique and would possibly benefit from pushing harder into “new territory”. That said, I also feel I need to understand certain fundamentals more! As well as C#, and definitely OOP principles. I haven’t created any Class hierarchies yet on my own, and that could probably help with many situations.

I have come a long way though, I started with very little familiarity with C# and Unity just a month ago. I’ve taken lots of great classes on Udemy to learn Unity. Before that I took the free CS50 course from Harvard and LOVED it. This has given me the fundamental understanding of C and higher-level programming languages that I have been missing since I was a child. I made a few simple games in GameMaker and it’s GML language when I was in high school nearly 20 years ago, but always ran into roadblocks which I struggled to learn about. Now, learning is easier than ever!

I have gained at least an awareness of the following systems:

Particle system, physics2D system, Rigidbodies and colliders, lighting, wind, tilemaps, ScriptableObjects, Events systems, Post-processing, Layers, Layer collision matrix, Materials, Shaders (what the heck are they?), Prefabs, Input systems, UI elements and systems, Coroutines, Singletons, static and public variables, animations, audio, and a lot about referencing. Plus, I also learned about the things which, in this moment, I’ve forgotten that I learned about. lol. It’s been a lot of new information.

**2024-05-07**

My good friend Paul and I came around with an idea for the game setting:

A planet in the far reaches of the Universe was recently discovered to contain the remains of a massive biological experiment. As ancient as it is massive, it is believed that the entire planet’s ecosystem was genetically engineered by an intelligent species and orphaned long ago. Surprisingly, it is assumed to be these life-forms which created the anomalous energies which triggered our sensors, leading to the discovery in the first place. Preliminary scans have revealed some surface ruins dating 5 million years of age. We must gain a foothold in the hostile environment and learn about these powerful creatures.

**2024-05-15 \*2:46AM**

I think I should start taking more frequent micro breaks. I should also stay more organized with my goals, as I’ve noticed myself wasting a good amount of time in the morning trying to get oriented. And then in the evening I am more productive.   
  
Either way I should get more exercise and breaks. I haven’t taken time to play a game or relax for days and I work barefoot in my basement. Lol. The reason I’m having trouble taking breaks is that I absolutely love this! There has been much satisfaction in finally getting the basic buildable tiles implemented (and even working at all) after a couple - Few-ish days of work.

Over the last week I managed to get a few things done. I got the tilemap to store health and other data about each tile in a dictionary. Then I used another dictionary which is referenced by both a list and a matching enum to implement the building system, where dictionary stores whether the player has the tile unlocked, and potentially other data. This structure makes the code easily readable and adjustable.

I also made a basic skeleton of a GUI for the building panel and used the dictionary to update the appearance and ability to interact with structures on the building list. It should be fairly easy to add new structures to this system, with some tweaks and new scripts for unique tiles and such. Cool stuff!

I also experimented with a bullet-hole effect using the particle system. It worked quite well, especially since I’ve already forced the raycast collision point further into the objects it collides with. One issue is the holes appear to not be on the surface when close to the corners. Therefore, I would need to use a sprite mask, and also handle the elimination of the particles when the structure they have struck is destroyed.

Leaving it off tonight with an only slightly ugly idle walking animation, to indicate the player is in build mode and not aiming.

**2024-05-22 - 2:01AM**

I had a few late nights this week, but it’s all been worth it. I’ve gotten lots more done, and it appears I haven’t even mentioned the journey of learning rigged animation in Unity. I ended up doing a course on Udemy which took about 3 days to really grasp and complete. It helped immensely and inspired me to create a rigged character for the game. It’s quickly gone from a basic little test project to something that will likely be fairly playable and possibly even enjoyable.

It was a struggle getting the animated character to replace my original simple one. After ironing that out I added more animations, which caused me to have to do some of the same work twice due to my lack of forethought and experience. Perfect. Learning more every day. The character can now smoothly (and so far bug free) walk, run, aim, melee attack, throw grenades, reload, switch guns, and die via animation. I ended up breaking the whole animation controller I was “using” (more like abusing) and scrapped most of the code for a re-write and used the animator states for more of the state machine logic. It worked out great.

I learned about some UI components while making a scrollable window for the build menu. I ended up getting the result I wanted with code to lock the selected item to a position in the scroll window. I also changed up the EventSystem so that the build menu can be controlled with the mouse wheel, as well as the mouse.

I added a feature that highlights every cell in a structure’s shape and adjusts the highlight for if the cell is clear for building. So that is working quite nicely as well.

I mentioned I created the death animation. Well, the next step was to build the UI for death and the main menu of the starting screen. I got those done and working, and even added a simple but neat animation to the UI elements of the main menu. I added a script I wrote which takes up to 2 canvasGroups and a list of CanvasGroups, and fades them in in sequence, with adjustable parameters of course. It’s creating a nice and satisfying effect that can reflect different levels of intensity. It would be neat to add some scaling bouncing effects to the loop.

More on the missions to come, tomorrow I will likely add the UI fade-in script to the build system and create the Pause menu. Who knows what else I might get up to through the process, but I can say I’m quite excited to get back at it with a fresh brain.

**2024-05-23**

I’ve gotten the UI to animate and react nicely without any bugs. I’ve been touching up the building system since I expanded it to include different sized structures on the tilemap. It is now working bug free. The next goal is to implement pathfinding and the wave system for enemies. Along with that I will be adding the Quantum Cube (base core) and ability to upgrade and buy things. This is very exciting as the game is nearing a decently playable state! Off to work.

**2024-05-25 – 1:57AM**

I had to do some deep digging into design patterns for AI and have a good hard think about what exactly I would want my enemies to do. I decided upon a state machine pattern over a seemingly more complex behavior tree that I might not really need. I figured out I can use grouping patterns for pathfinding and other AI checks, so I will certainly want to learn more about this in a later chapter. But beyond that, the state machine appears to be the right choice.

I made good progress testing state machines for the first time. I discovered a nifty trick for using scriptable objects to customize the behavior of a state machine during the endeavor. I will be adding more logic to it tomorrow. I’ve managed to figure out how to optimize AI detection using layer masking to reduce the number of collisions and distance calculations needed. With my hand-made first try at basic AI logic, I was able to get up to around 300 enemies before serious lag started to set in. This is without any object pooling yet. But now, with State Machines, it is up to 700 before noticeable lag happens, and that’s with hundreds colliding at once (all this in the editor’s play mode). Things can certainly be optimized further, which is exciting.

There are some definite changes to workflow with this scriptable object – state machine. It’s also really neat to be working in a class hierarchy as this is something I’ve been wanting to learn more about and practice. I guess it’s a win-win! (or win+win?) lol. I should get some rest. Once I sleep on this it will flow way better tomorrow!

**2024-05-27 – 2:18AM**

I really should get to bed sooner and sleep more. 7 hours is not enough for me!

I’ve been working with the new state machine and AI the last 2 days and transferred most of the behavior from old enemies into new ones and now have things back to how they were before the switch, but new and improved! I also added actual Credits and made structures apply their cost.

I also added a StatCounter and got most of the counts done on different actions in the game. I just have to get it organized to the right format for going into my UI, which I will figure out tomorrow.

I ended up spending a lot of time debugging issues with layers and triggers. I’m quite certain there must be better methods for debugging that I will be very happy to learn of one day! I may do some digging on that front to help productivity. But, as I said, everything is working well! I get to keep moving forward with new features, as things are also starting to speed up as I get used to the new workflow with the state machine structure of the enemies and AI.

After I get stats put into the death canvas, I can see the shop and upgrade panel getting built soon! Or maybe turrets first? Hmm.

**2024-05-28 – 2:53AM**

Just trying to log things quick before bed. Had a few good struggles today but still got lots done. Progress!!

-Did work on the stats panel, got some bugs with transitions and menu control override fixed

- Fixed input action subscribe errors on restart, rebuilt input methods for animation

- Improved enemy tile melee attack detection. made the quantum cube killable, and cause game to end

- Make camera move to player or cube depending on which died first

- Made some funny text for death that gets randomly mixed together for player or cube death

- Improved the FadeCanvasGroups further with more options and functionality to fade in or out in a wave among other options

- Fixed some minor lighting issues, more to go here.

- Lots more work to go, not sure what I’ll work on tomorrow yet exactly, but time shall tell.

**2024/05/28 – 11:41PM!! Yes!**

Early night with my roommates. I still haven’t watched the new Mario movie. So, Lets’a Gooo!  
  
Hahaa, I jest. I had amazing progress today. I came up with my own design for a wave system, inspired by everything I’ve learned so far. It works awesomely!! I literally wrote code for like 2 hours and only had to fix one bug before it was working. I don’t know about you, the reader, but for me that is progress!

- Added WaveSOBase with list of SubWaveSO which has List of enemies, spawn speed and position selectively spread along circumference of circle.

- Selectable batch size, individual spawn times, times between batches.

- Great customizability with the nested SOs. Made a new Wave and SubWave showing my roommate in less than a minute while half focused. Everything worked.

- More features to this system to come tomorrow.

**2024-05-30**

- Continual progress!

- Restructured my new design for the wave system slightly.

- It could still be improved but I've got to start using it at some point.

- Added the growing multipliers by wave to enemy randomization.

- Made some new wave designs and styles.

- Put UI in for next wave and current wave.

- Build the timer for waves.

- Can now add multiple waveBases, which repeat a selected number of times before switching to the next.

- Corpses now linger then fade out.

Tomorrow:

- Day/Night

- Crabs with negative knockback?

- Tighten up players gun collider.

- And hopefully more.

**2024-06-01**

Adding the upgrade system.  
  
Figured out a structure for the upgrades.

- It will work for player stats, gun stats, tile stats, and basically anything else.

- I have more learning to do but I should be able to use it for the purchasing system for ammo, grenades and consumables.

- Today I got the structure implemented and the game still runs as expected.

- I also got some artwork done for the playerStats images. more to come for guns and tiles.

Tomorrow:

- Make UI for upgrade system. I have more dreaming to do for it. I'll probably have separate tabs for the player, tiles, and each gun. a horizontal scroll window will be suiting. There will be a lot of work involved for this system, as it is kind of half the game. I took time in exploring the best option to speed things up as much as possible. So far so good!

- Once this is in place I can start work for the Cube interactions, day/night.

**2024-06-03 – 2:59AM**

Okay, this is far too late, but for some reason I am not very tired. Time to pack it in anyway, lots to do tomorrow!  
  
More work to the upgrade system and UI.

-Upgrades are now purchasable and applicable.

-Fixed some bugs in UI transitions bewtween canvases.

-Made the escape key exit out of menus as well as open pause menu.

-Tightened up the FadeCanvasGroupsWave script up a bit. it needs an adjustment to reverse the fade order when fading out. This should help with some strange but minor effects of starting to fade out while fading in and vice versa.

- Added all the playerStats upgrades, which are now being referred to as exosuit upgrades. A few upgrades need logic still.

- Added tooltips for all the new upgrades, fixed bug with tooltip disappearing when menu closed with tooltip open

- Had some issues with UI interactions but found out hidden canvases were blocking buttons. Had to code in some fixes and transitions.

Tomorrow:

- Finish a few upgrades logics

- Add multiple panels to the upgrade menu, so player can buy ammo, grenades, and other upgrades.

- Work on more upgrades.

**2024-06-04**

Getting to bed a bit earlier, 1:27 currently.

- Finished logic for exosuit upgrades. Had to fix some issues with centralizing data from other scripts into playerStats class. I managed to get it all working great, with a little extra work needed to add ReloadSpeed increase to the animation controller. I made melee speed increase too but decided not to use it in the game as it looked too unnatural.

- Made shadows more efficient with a single composite shadow collider that matches the tilemap and updates when a tile is placed. Theres a few strange behaviors I found in the shadows but overall better than before.

- Improved tooltip system to not have to wait for a full delay unless fully faded out. It’s nice and snappy now!

- Added the functionality to switch between multiple panels. I ran into an issue with my fadeCanvasGroups script, which has gotten rather large, where the group does not fade out completely. I’m still not actually sure it is even that script causing the problem. I need a fresh brain tomorrow to decipher that one. It could be from the new Pane Switcher script, but I’ll find out tomorrow.

- Added a very basic popup message area to alert the player of things. It’s simple but allows any other script to give it a message. It could be expanded to run through a queue of messages if needed.

- After adding the accuracy upgrade, I noticed that the bullets were not shooting directly at the mouse. It took longer than expected to get this one fixed, but I seem to remember having some brain fog. Need to incorporate a bit more physical activity into my days.

- Added a max level limit to the upgrade system.

Tomorrow:

- Touch up the bug with fading panels out fully.

- Add the purchases screen for consumables. Start implementing them.

**2024-06-04**

- Ended up doing the gun upgrades first, since ammo is tied in with everything.

- Made scripts for gun upgrades

- Rebuilt the gunSOs and included upgrade stats amounts to each gun for each stat, did some quick balancing.

- Created UI Panels and UIUpgrade prefabs

- Ran into issues with UI canvas blocking button interactions, was too tired to figure it out

- Also found bug with hitting build key in quick succession would cause a bug where the player would still be holding a block for building and their gun. Still, too tired to fix, though I tried.

**2024-06-06**

- Fixed the last 2 bugs of yesterday in the first 15 mins on the computer. Therefore, I’m going to bed much earlier tonight!

- Fixed build key and panel bug right away and was stoked.

- Later found the panel bug wasn’t actually fixed when I moved the gameplay HUD UI to camera rendering mode, instead of overlay. I struggled to fix this for quite a while. I didn’t find much for solutions online but still found people with the problem. Turns out I had to put canvas group components on each canvas and set “Blocks Raycasts” to false (and control each being ‘Raycastable’ via a few additions of code). I was happy to be able to share my fix with the community and move on.

- Added ‘prev’ and ‘next’ buttons to the upgrades panel

- Added particle effects over the UI for Upgrades Applied effect and effects on player for upgrades

- Fixed some build panel scrolling misalignments from the UI render mode changes

- Fixed build panel and upgrade panel being open simultaneously causing bad behavior

Tomorrow:

- Catch up on a few bugs

- Maybe add a panel to display player and gun stats

- Started creating a multi-wave function for my wave system which needs finishing. It allows 1, 2, 3, etc., waves to be spawned at once as the game progresses. Something to play around with.

**2024-06-07**

I skipped the shutdown ritual last night but read a great article about keeping productive and continuing to learn and grow vs just getting tasks done. I’ll strive to be mindful of my daily routines and personal health, as well as resting and planning accordingly to produce quality work and make the most of my time.

Ex. The multi-wave function for the wave system turned out to be a bad idea. I was too tired to visualize the entire system and thought it was a good idea when it was not.

- Tasks completed:

- Fixed bug with AI triggers not working from bullets

- Scrapped the multi-wave idea

- Made the QCube Shop panel with for ammo, health, consumables

- Created upgrade scripts for each new upgrade

- Added QCube status panel with effects for when taking damage, as well as a glow from the cube

- Created day/night transition

**2024-06-08**

Another decent day but should have shut it down an hour or two ago. Took some good breaks today and got a decent amount done.

- Fixed sunrise/sunset bug when skipping the day cycle during transition and locked it to camera

- Ended up refactoring the upgrades and upgrade system to allow for easier additions. This was a decent surgery with quick recovery time, lol. I broke down the one UpgradeType enum into 4 separate ones, and created an enum from UpgradeFamily, which then contains specific types. This took a bit of time but went very smooth.

- Some minor adjustments to upgrades UI

- Addes message when building with not enough Credits

- Wave system was not working correctly, found an issue with current wave initialization. Fixed.

- Changed base wave multiplier to 10 to allow for smoother increase in numbers over time

- Fixed up the wave system a bit and tried tuning a handful of waves into a smooth experience. Turns out my design is still off a bit, so I spent some time experimenting and thinking up a better way. I’ll be changing the constant waves to actually be constant until the sub waves end. I may try a randomized list approach to wave generation. Creating them by hand does not allow randomization the way I have it now. So, by creating simple templates as wave bases, I can then have a random selection of these waveBases and also for constant waves, which could start a bit later. Over time the list could grow. Might use ‘boss’ waves to have added difficulty with predictability, and can add new enemies after bosses.

- Lots to try out and figure out tomorrow. Id like to have more control over the flow of the waves.

* + 1. **- 9:22PM**

- Fixed an annoying bug where the ShadowCaster2D was not updating its shape when the player or enemy would destroy a tile. Learned to use Lazy Initialization to prevent its reference from temporarily becoming null.

- Played around with the wave system a bit. Decided to make a variation of the current system but with a spawner that creates an ever growing list of randomly selected subWaves. This will hopefully help keep a smooth flow. I’m hoping to try it out tomorrow.

- Had a little more me time today and took care of some housework. Going to play a game and relax tonight.

**2024-06-09 – 11:10PM**

- Got the new randomized list generator version of the wave controller up and running

- Spent some time tuning it up, the game now has a nice progressing difficulty

- Tried a few things with the lighted and shadows hoping to allow tiles to be lit by the flashlight, but also have the shadowcaster cast shadows on themselves(so solve some shadow/light artifacts on edges of tiles in complex areas). I wasn’t able to find a work around for this but found a new-to-me technique called reflection to change read-only values of a class.

- Added a new buildable structure, the Flood Light! It lights up a large area without any shadows.

- For the new random wave system, lots of adjustments to multipliers and logic for handling the wave. It works great now, it’s easy to control the length of waves. The start and progression of difficulty is also much easier to control. This will make having multiple difficulty setting far easier.

Tomorrow:

- I’ve kind of been going with the flow but I do plan to make turrets and doors very soon, as they will completely change everything for gameplay possibilities. Windows and/or raised shooting platforms are also on the docket.

- I’m extremely happy to have the wave system working and had a few enjoyable play-throughs today while testing. I’ve started thinking more about getting active online with the game through communities and social media.

**2024-06-10**

- Added another buildable tile! A simple door. There were many different approaches that could have been taken, and I went with a simple box collider trigger, on a new layer that only collides with the player.

- The approach allows for the door to remain as part of the tilemap’s composite collider, which is dynamically generated. That same collider is also what the tilemap uses for a unified shadow effect.

- Fixed a bug where when you enclose yourself in walls the shadow generator would enclose the whole area in shadow. Did this by allowing a small crack in the door where the light might naturally shine through anyway, and the shadow geometry was successfully separated.

- Started working on adding new functionality to rotate structures in the building system, and things got messy.

- In retrospect, I went way too fast in implementing all the changes to two connected systems, the Tilemap System and the Building System. I was faced with an overly confusing web of bugs which I confronted for hours before realizing I still didn’t really know where the problems started.

- Decided to scrap all the work on rotating and revert to yesterday’s commit. I was able to keep the work on the doors and import those back in.

- Lost about a half-day’s work, with the added bonus of learning a bunch through the process. It was tough to handle, and I did feel a sense of defeat for really the first time through this project. That didn’t last long. This only makes me want to try again and do better, as I’m sure I’ll get it done.

Tomorrow:

Approach adding rotating structures in the building system from a logical start point and make changes along that chain of logic SLOWLY. Lots of logs and testing along the way so I can keep track of any issues that arise and deal with them appropriately. In theory, this will help with implementing the changes through the systems. I definitely got ahead of myself today. Lots of excitement!

**2024-06-11 – 7:57PM**

This is almost banker hours!

- Take two of the rotation system worked WAY better. It actually went fast too due to the lack of problems. Instead of changing all the existing methods through all the systems (extra ugly) like I did in take one, I just copied the StructureInHand and applied rotations to the copy. With a little refactoring it was working like a charm.

- Then came rotating the attached game objects relative to their potentially oddly shaped tiles where the target location of the object after rotation is dependent on the structures size and current rotation. Also, it took some time to find that Unity’s Tilemap system already has a method for another trick I was trying to do by hand, which helped immensely, referencing the tile’s gameObject.

- A great learned experience putting this system in, despite a call to take the lazy way. I’m very happy I continued on.

Tomorrow:

- Smooth out a few small bugs from rotation system side effects.

- Start preparing to get a decent beta version ready. It’s Father’s Day soon and I think my Dad might appreciate the satisfaction of seeing I’ve made such progress on something that I’ve been passionate about my entire life.

**2024-06-13 – 11:26PM, not bad**

A good long day. Slowed down a bit by the end of the night but had a surprisingly clear mind after getting out in nature and taking most of the day off yesterday.

- Turns out the building rotation system needed some fancy work to get the destroy function to work. I thought it was all good, but it was not!

- Had to inverse the rotation of the transform matrix of the tile structures for the destroy functions. Got it all smoothed out and working perfectly so far.

- Fixed some old bugs from building highlights not being drawn in the right order

- Added an upgrade stats function aptly named UpgStatsGetter which digs in all over the place to get stat values and gun stats, and their upgrade amounts.

- Added color coding to stat values and upgrade amounts. Upgrade costs are red if too expensive

- Spent a good hour making a UI panel and getting it integrated into the existing panel-switcher, smoothed out some layout bugs.

- Later realized that the extra panel is probably not needed, but now that I have access to the extra info, I can make the upgrade UI much more intuitive.

- Fixed up the idle and idle run animations a bit as the arms were too skinny. Turned out to be worse due to a problem in the blend tree and speed values.

- Added an Unlock button that will be used for different upgrades and upgrade families.

- Can now unlock the SMG for 3000 Credits.

- Fixed bugs in QCube interaction, UI alignments, upgrade balancing and bugs, and others.

Tomorrow:

- Make clickable area for upgrade buttons bigger

- Get rid of extra upgrade panel, will be so much nicer

- Player – DoorOpener trigger collider bug when lowering weapon

- Grenades don’t throw sometimes, need to tighten up animation trigger

Super stoked to be making good progress!

**2024-06-15 – 1:26AM**

Did some cramming today with the help of a Red Bull and some exercise. Got a bunch done!

- Player and door collider bug fixed. Doors stay open now when you lower your gun.

- Moved upgrade stats to be with the upgrade buttons. Got rid of upgrade stat panel.

- Fixed a handful of bugs with the display of upgrade stats

- Added SettingsManager, added difficulty setting for Easy, Normal, and Hard.

- Added options panel for menus.

- Added standard WASD movement. Added option for movement types.

- Balanced: Wave progression between difficulties, enemy stats, blood particles

- Wave bug came up and got fixed. Was a typo somewhere.

- Prevented wave spawner from selecting 2 of the same subwave in a row.

- Fixed bug with UI selection being cancelled on starting new wave. Removed UI Submit actions.

- Too much Red Bull bad

- More enemy balances for credits and melee attacks

- Did some exercise

- Added the repair gun! It’s technically a gun too, you select with ‘C’ or number 0. It’s quite satisfying to repair things with!

- Added repair effects into TileManager for repair gun.

- Added credits cost for repair.

- Fixed all known bugs with repair gun.

- Added a GunSights script, applied to repair gun.

- Fixed bug popped up with bullets going through walls again, was trying to tighten up the collision visuals but requires a bit more time coding to implement.

Tomorrow:

- Head look limit

- New wall shapes

- UI touch-ups

- Crab2 chase-cube logic, extra enemy?

- Prepare for an alpha build for Dad for Father’s Day, as well as family, friends, and testing.

**2024-06-15**

Okay it’s technically the 16th for 10 mins now.

I’m going to add a zip file with the game .exe after this commit. First built testing run in a while ran extremely fast, which was a relief as the editor gets slow easily.

- Added new buildable structures: 1x1 Corner, 1x4 Wall, 2x2 Corner, 2x8 Wall, and Blast Door.

- Added constraint to head rotation. No more creepy zombie head

- Added new Enemy Chase logic for only walking to cube, added to Crab2

- Fixed more Upgrade stat formatting. Did them all this time, should be good now.

- Created graphics with some help from Dall-E for the Matter Materializer (repair gun) UI image

- Manually adjusted laser pistol to create gameplay repair gun

- Fixed bug with doors “closing” when hit

- Fixed some UI scaling/anchoring and world object size bugs when using ultra-wide monitor. Found more to fix tomorrow.

- Fixed UI stats not updating on refresh

- Bullet casings are now smaller

- Found and fixed bug with Enemy Count not being reset when restarting the level, more testing needed

Tomorrow:

- Finish the current bug list, do more testing

- Refund some credits when removing a tile

- Difficulty doesn’t persist on restart

**2024-06-14**

Spent some time looking into places and ways to share my game to get some user input. Ended up having one friend who’s whole family only has Mac, so I started implementing new controls.

- Struggled with adding a second control to the scroll effect I created. Managed to find a fairly solid solution with a coroutine to slow the UI selection changes from mouse movement.

- Ended up learning some things about the input system, and also Mac computers. It turns out I know dangerously little about them, so I think I’ll buy an older one right away to play with and test on.

- Added scripts to switch some controls text for Mac users

- Balanced the Damage and Knockback Upgrades as they were OP

- Slowed the baseline for subwave spawn time

- Created basic tutorial popups for a welcome screen, build system, and upgrade system.

- Changed UIUpgrades to update their text in coroutine every 1/5 second.

- Due to the new Standard WASD movement type, adjusted movement calculation to smooth movement speed and also add a buffer to the angle toward the mouse which full speed is achieved.

**2024-06-18**

- Added a new structure! The Shooting Platform, or Raised Platform

- Allowed for shooting over walls while on platform

- Changed flashlight to no shadows when on platform, bullet casings stay on platform

- Fixed bug where the clicking OK on the build menu tutorial would close the build menu

- Wrestled with a rotation bug where the game object was not in the proper location again on rotation. This was similar to the when I added the door, and found out all the effort to create logic to handle object position for rotation was pointless after properly configuring the pivot points and collider offsets.

- Fixed bug with UI Upgrades not initializing properly on gun unlock

- Added door damage graphics and functionality to handle door states and images

- Fixed bug where highlight and preview image were not being drawn in the correct order

- Balanced crab2 a bit more, scale down melee blood splash, adjust build menu scaling

**2024-06-19**

- Made better bullet trails. Old ones were ugly and hard to see

- Added hit effects for laser and bullet

- Increased build range

- Can no longer reload when clip is full

- Optimized bullet effect handling

- Did some fine tuning to effects (it’s too early for this, but I can’t help myself!)

- I think I would benefit from learning more about shaders and materials.

- Started searching for free sounds online. Might pop into Ableton at one point soon and make some sound.

- Tested some laser sounds in game

Tomorrow:

- Work on sound

**2024-06-20**

Less production today, getting ready for a camping trip for the weekend. 😊

- Searched around online for a few sounds

- Ran into a bug where some sounds were clicking at the end

- Ended up making an AudioUtility that is used by sound-carrying objects to zero out the first sample and add a short tail of empty space. This prevents the clicking for all future files.

- The setup is for the audio manager to manage a pool of AudioSources. I created a Sound class that holds the AudioSource settings and other info. Objects that play sounds will send their sound to the AudioManager to be played. I’ve currently got 2 play types for stationary and follow target sounds.

- Played around and smoothed out the spatial sound, got rid of doppler effect for bullets.

- Reduced player projectile damage to tiles

- Fixed index range bug with wave styles breaking game at wave 16

- Fixed Crab1 not able to reach past player gun collider.

- Increased health of all enemies slightly.

**2024-06-25**

Had a solid weekend off for the first time in a long while. Went camping and rock climbing and am excited to get back to work.

- Fixed bug with enemy melee animation not playing

- Fixed blast door physics geometry

- Lowered stamina regen

- Building structures now clears ground particles and debris

- Added a recoil effect using inverse kinematics. This felt like it took me far longer than necessary as I struggled with figuring out how to convert the local positions and rotations of the hand targets to move relative to the gun and player’s position as they moved around in world space. It was good practice, but it made clear to me that I could learn more here.

- Fixed bug with removing doors while standing in them

- Gathered a couple hundred more sounds to play with

- Created a categorized list for the needed sound effects

- Added explosion SFX

- Sound class now has a list of AudioClips. I may need to adjust this design as more sounds are added to allow for a list of sounds instead of a sound with a list of clips. This way I will be able to adjust the setting of each clip easily within Unity. But, I will wait to experiment with it.

Tomorrow:

- Add more sounds

- Might make grenades do tile damage

**2024-06-26**

Still before midnight, and thinking of going for a late evening walk to digest the amazing dinner I made tonight! Decent progress today but slow going still.

- Fixed bug where grenades would throw until -1 ammo when holding button

- Added multiple sound types with their own AudioMixer channels and AudioSource pools.

- Added spawning collision checks using the tilemanager’s CheckGridIsClear I already made, so enemies don’t spawn inside structures anymore (This was quite simple as I recently changed the script to accept layermasks for the checks :)

- Added some more logic to the chase states so that enemies react differently to bullet AI triggers than to the players AI trigger.

- Enemies now stop chasing after a set time if they leave player’s AI trigger

- Enemies only chase for a set time after being shot, unless in the player’s AI trigger

- Added grass walking sounds via animation events on the players legs.

- End of night: Found a bug where the legs jump around a bit due to an attempt to sync them to the body’s walking/running frames. It was causing the footstep sounds to play erratically. I believe it’s due to the various animation states starting at different positions along their animation timelines when they are triggered. I’ll have to see how it looks to eliminate the sync and go from there.

- Side Note: I learned how to use Audacity a little bit and am now able to edit, slice, and export multiple audio files with relative ease. Still more to learn I’m sure, but it is absolutely doing the trick so far, and I may not need to redesign the Sounds holding multiple audio clips setup since editing is so quick. Plus, it sounds way better to spend the time on each sound, as I’ve been discovering.

Tomorrow:

- More sound and whatever else I find

**2024-06-27**

A bit of home/life to take care of today, I was distracted but got right into the flow by the early evening. Signing off just before midnight as usual these days.

- Went a ways down the rabbit hole today looking at ways to sync up the animation states in the way the are set up now. I ended up using a cooldown timer to prevent ultra-rapid footsteps from sync timing issues.

- Noticed bug where legs were moving way too fast or slow when walking/running while using melee, reload, or throw. Separated the speed of the legs to be independent, but still synced.

- Fixed bug where repair gun bullets flew past their max range.

- Fixed issue where clips where disappearing abruptly and casings faded too slow.

- Fixed bug with stamina regen upgrades not working

- Sliced up and balanced some reload sounds, added them to animation events for SMG.

- Ended up refactoring the AudioManager and Sound classes after an attempt at having a SoundManager with dictionaries for sound origins such as Player, Enemies, Tiles, etc, which held the Sound objects as values and string keys which I would have to manually type out in every sound-making object, and add to these lists to populate the dictionaries… Terrible idea. That’s what scriptable objects are for!

- The previous point was all about optimization, not only for my time, but for memory and CPU usage. This is my first attempt at a proper audio system.

- Turned Sound class into an SO, got rid of SoundManager.

Tomorrow:

-I could tell I didn’t have the right idea for the audio system design. I feel more confident in the new design and much more motivated to go ahead and start adding a bunch of sound. It’s been a daunting task from which the fog has lifted, and I can now see some semblance of a goal.

- I’m sure I’ll find some side tasks along the way to freshen things up 😊

**2024-06-28**

Late night tonight, at 1:30AM now. Learned a bunch and got lots done with sound. Even managed to call and chat with most of the family and make a nice dinner.

- Mixed some sounds to get a pretty awesome Laser Pistol selected sound

- Found ‘selected’ sounds for SMG and Repair gun.

- Started searching for laser sword sounds. Realized it was a much better idea to make my own to have full control over it.

- Dusted off the Ableton shortcut, hopped in, and remembered how much I forgot. It started coming back fairly quick though.

- Used Operator with effects to get a decent light-saber-like sound. Bug: Sounds exported from Ableton were all lower pitch than the original sound.

- Found out my audio interface and Windows audio sample rates were out of sync.

- Learned a bit more about bit rates and decided to try to stick with 16-bit for SFX and 24 for music.

- Finalized the sword sound and made some variations.

- Added player pain sounds

- Made the waves a bit longer at start.

Tomorrow:

- More sound and might catch up on a few features I’ve jotted down over the past couple of days for QoL / user experience.

A great day! Strange issues from Unity(?) at the end of it, but all is well

- Tuned up the build range circle to be working nicely

- Added functionality for health and repair consumables where the upgrade amount and cost text are updated real time to show the amount and cost when it is lower than the max for the consumable. Ex. Only see and pay for 7 health instead of 100 when near max.

- Added slight random speed to Randomize for enemyBase

- Had a good friend contact me on Steam today and I got him to try the game out! I was able to stay on voice chat with him through the whole thing and gained some great insights!

- Added a bit more direction to the welcome window. Controls pop up by default only when tutorial is not disabled

- Sound balancing insights from playtest, might dull the sound down a bit on death. Still need music anyway.

- Lowered melee stamina use and increased regen slightly

- Added sprint to help controls.

- Grenade spacebar may be a problem

- Added basic direction arrow to show direction back to base when it’s off screen.

Tomorrow:

Shall be another good day

**2024-06-30**

Some random stuff and mostly making enemy sounds today

- Fixed arrow not pointing directly at QCube off screen

- Noticed the laserSword sounds I made in Ableton were massive in size, so I ended up learning to use macros in Audacity to quickly convert and export the list of WAV files to OGG. This brought the ‘longer’ ones (still barely a second long) from 800kb down to 12kb or so.

- Created a bunch of variations for Crab1, Crab2 attack and pain sounds Audacity. Found a decent way to create good variety from a single sample. With more time invested and mixing this with macros, it could be quite powerful.

- Created and used a quick macro to amplify all 35 Crab1 slash sounds in Audacity

- Balanced Crab sounds in Unity

- Added cooldown for melee attack hit sounds

- Crab2 seems to be getting stuck in the open once in a while… But will walk again when it enters the players AI trigger collider and becomes aggroed. Needs more investigation.

- Stopped pain sounds being played when player is dead

- Created dog attack and pain sounds

- Tested and balanced some more.

- Stated trying to override an Input Binding in Unity to toggle or set the spacebar and other controls. Too late and tired to continue. Quick log and sleep. Fresh brains are good.

Tomorrow:

Canada Day!!

**2024-08-03**

It’s been a hectic month since I last posted, travelling to AB and staying for a month to be at my Dad’s wedding and then Mom’s summer party. I spent a bunch of time with much of the family and got to see my good friends and their new baby boy, who was born only a handful of days ago now. I got to spend a bunch of time with my dying Grandpa and today am off to a celebration of life for an old friend, full of music and dancing. I somehow have still managed to get some work done here and there, so I’ll try to summarize the past month as best I can:

- Added 2 automatic turrets to the game: Laser turret, and Gun Turret. They have a random scanning script they follow, to randomly rotate around. I plan to improve it later so they only random scan in directions facing away from the cube.

- Turrets track in a very efficient way with an expanding circle collider trigger which expands at 0.5 units per ‘time’ and ‘time’ is upgradable. The turret takes the first enemy it finds and removes the collider, so very few collisions are ever checked, even with a hundred enemies around.

- Turrets ended up being massively OP, and now have a max number you can build, which is currently zero at start, and player uses upgrades to unlock more turret “slots”

- Changed the way upgrades work: instead of upgrading the laser pistol, SMG, and each turret/gun separately, upgrades now work on all weapon of that GunType. Ex. Laser upgrades affect all laser weapons and laser turrets. Bullet upgrades affect all bullet type weapons.

- Turrets now use their own GunSOs. Turrets shoot slower and do less damage that player guns, but their upgrades are generally the same value

- Added upgrades for turret rotation speed and sensors (detect range + detect speed)

- Improved AI targeting logic for enemies, designed in an efficient way where collisions are only check OnTriggerEnter and OnTriggerExit, rather than constant checks from all enemies. They now respond to bullet hit by moving to the origin on the bullet and should stick to their target once found.

- Crab2 no longer freezes

- I visited one of my best friends for a few day on the way to AB, he tested my game and actually asked to play again the next day! He probably played for over 2 hours total. The need for a better tutorial is arising.

- Guns no longer have kickback or recoil which pushes the player back, only a visual IK motion in the hands to simulate recoil.

- Did a bunch of balancing on weapons, enemies, Credits, and upgrade costs and amounts

- Left clicking in build mode now builds a structure, instead of exiting and shooting.

- Tab will now exit build mode, and ‘B’ will exit upgrade mode

- Attempted to fix issue with player state machine (not a proper design, I made myself early on. It stands as a testament to ‘how not to build a state machine’) getting broken and stuck in isReloading state when reload speed is upgraded to be very fast, and framerates drop. Animation event should hopefully trigger always now.

- Reworked the Shooter script to be a parent class of PlayerShooter and TurretShooter. Now will work for any object.

- Fixed my ScrollRect script I created for the build menu to be in real time seconds instead of game seconds.

- Got to watch a playthrough with my rather inebriated father.

- Came up with some tutorial ideas

- Plans for tutorial: Currently, I plan to make a playable tutorial level which will introduce the player to the various mechanics. You will start in a field, making your way to a destroyed outpost to take over and defend.

- Plans for a basic melee weapon at start, and unlockable/upgradable laser sword to buy.

- Came up with idea to have character classes to choose from. Currently will aim to implement 3, a Neutral class, Fighter class, and Builder class. Each will have their own strengths and weaknesses, with Neutral being the standard experience. Could have permanent points used for class specific buffs at start of game, or keep it a public pool of buffs to choose from in order to customize even more.

- Started reworking the tutorial system to be more modular. Added TutorialManager Class.

- Added upgrade to increase health of all structures

- Added ‘close’ button to upgrade panels

- QCube stats is now called StructureStats and will contain structure related upgrades.

- Planning to add a tabbed menu to the upgrades panel.

- There is now a TurretManager class which keeps track of the turrets for building and upgrading.

- Added the functionality to existing gun upgrades to affect turrets. Ei. Laser upgrades affect all laser guns/turrets, same with bullet upgrades.

- Added option to disable the spacebar from throwing grenades

- Fixed a bug where player could build more turrets than the limit. Discovered a cool trick that I did not know was possible. I think AI told me it wasn’t once, and I believed it. Anyway, I found I can dynamically copy fields and properties of a class using System.Reflection. Cool stuff! This will prevent me from having to forget to add fields to scripts which copy properties, after I add new fields.

- Changed the enemy Chase AI to now take an ‘origin target’ instead of being hardwired to chase the player. They can now chase anything.

- Adjusted the audio listener position and all sound fall-offs to allow for more volume fall-off at shorter distances. Makes the sound a fair bit more dynamic. Created new sound profiles for turret guns vs player guns.

- Added permanent trigger area around turrets, to pull enemies in

- Made it so that going into build mode resets the current rotation

- Made UpgradeManager (previously UpgradeSystem) a singleton to allow for easier access to upgrade data. Singletons are now re-initialized at scene restart by the player script

- Had another old friend do a bit of a playthrough, didn’t get far as he was still tending to his new-born. Still, very valuable insights gained. The tutorial level is going to change everything, making it a much better experience.

- Changed tutorial panels to allow for build menu (or others) to pop up with tutorial. The previously selected UI object is stored, so when exiting the tutorial panel, the build menu’s first item is re-selected and scrolling is functional

Tomorrow (probably a few days from now, once home):

- Need to create some new art for upgrade buttons

- Need to decide if turrets will have a base like a wall, or be more fragile

- Need to fix/re-implement turret reload and add recharge for laser turret.

- Found bug where holding shift still drains stamina when not moving

- Need to double check turrets stats being initialized on build

**2024-09-07 (2am, technically the date is the 8th)**

I’m back home after an extended stay in Alberta for my Grandfather’s funeral, and attended another wedding in BC. I’m about to start a new schedule of 2 part-time jobs and squeezing some dev time in wherever possible. It feels great to be making regular progress again!

- Fixed movement and other preference settings not being initialized properly.

- Fixed repair cost bug

- Added new camera control functionality to allow for cutscene and tutorial camera zoom/focus.

- Camera now locks on player when building. This avoids accidental placement when the camera is moving.

- Made corpses slightly faded, to make live enemies stand out more.

- Cut the cost of all structures in half, reduced SMG unlock cost.

- Fixed annoying bug where the UI would deselect after accepting the build tutorial and stop the ability to click on buttons in the build panel.

- Amplified the player pain sound files and adjusted so that its always full volume

- Made sword 10% longer, fixed bug with sword image not showing

- Fixed bug where grenades throw target was not stored on click, but instead when grenade left the hand, sometimes causing grenades to fly in unexpected directions.

- Added flood lights to platforms. Made flood lights look nicer.

- SMG unlock now affects toolbar image

- Added flashlights to turrets

- Fixed issue with some turret upgrades not working.

- Other small fixes and adjustments

Tomorrow:

- Work through list of fixes and improvements from playthroughs

- UI and artwork for upgrades, re-working upgrade panels to have navigation tabs

**2024-09-08 (It’s actually the 9th, late night again)**

- Added turret reload and recharge methods, fixed associated bugs, moved some code up the Shooter hierarchy.

- Found/fixed bug with turrets not accepting target at max sensor range.

- Fixed turret rotation upgrade not working, and some other minor turret upgrade bugs.

- Added indicators for when turrets are reloading/recharging.

- Added a flood light to shooting platforms as well as a green arrow to indicate stair entry in build UI.

- Added preview tiles/images for turrets and platform. BuildingSystem now accepts an alternate animated tile for the build preview.

- Added aiming cursor, build cursor, and repair cursor.

- Changed player’s LookAtMouse method to use a faster Lerp curve. Much nicer!

- Made some artwork for the new UI upgrades, also did some changes to existing layering that was repetitive by adding a ‘Wrench’ image. Broke everything at one point by accidentally deleting all references when re-importing the same sprite sheet with minor adjustments.

- Adjusted tutorial panels to better reflect recent changes. Added new upgrade panels. More to come.

- Fixed bug with tutorial panels causing player controls to freeze or unlock unexpectedly.

- Finally found what was causing the ‘Prev’ and ‘Next’ button in the upgrade window to be blocked. It was painfully (un?)obvious that an overly wide text box was blocking it.

- Added functionality to switch back to the previously held weapon when using the repair gun quickly with ‘C’ (same key to select/deselect)

- Other small bug fixes and changes

Tomorrow:

- Upgrade UI panel work

- Maybe ready to add the rougelike mechanic for EXP point on death? I plan to add an AI spaceship which you control before the fight. It will allow you to select your upgrades and customize your clone in an engaging and organized way.

**2024-09-09**

A decent day, despite less time than hoped for to work on the project.

-Fixed some UI bugs and graphics

- Added category tabs to the upgrade panel. The tabs light up to indicate which panel you are on. The tabs are also buttons which can be clicked.

- Upgrade panels can now be switched between at full speed

- Got most of the work done to make structures unlockable with panels. Still need to implement a final method to apply to the build menu elements.

- Had to rework a complicated part of the building system to allow for reorganization of the main structures list. Enums are only handy sometimes, but I’m learning new tricks!

- Fixed bug with build highlight vector matrix not rotating properly after the recent build system changes

**2024-09-10**

Got a decent version of the upgrade panels and associated unlocks set up. It may change in the future to something cooler, but this will certainly work for now! I may add upgrade buttons to unlock special structures, like a Mining Drill or Drone Hub.

- Added unlockable structures to upgrade panels when they are unlocked.

- Most upgrade panels need to be unlocked now. They come with associated unlocks like guns and structures.

- Added ability to unlock structures in the build menu

- Structure upgrade panel unlock now gives 1 turret slot with buildings unlocked

- Laser Tech unlock now unlocks laser turret, and same with Bullet Tech and gun turret

- Added ability to quickly switch to the destroy tool and back to the previous selected structure while in the build menu.

- Added colors to build menu text, including cost text showing affordability with yellow or red

- Adjusted upgrade panels/tabs so that navigation wraps around at start and end of list

- Easy mode now gives the player $600 at start

- Fixed bug with cursor not changing to the proper image after leaving build menu with repair gun selected

Tomorrow:

- I have a ‘trial run’ at my new part time job in the morning, and some things to prepare for my first day at the other job Thursday. This marks the end of one era of development and the beginning of the next. I hope to continue with daily progress despite the lifestyle changes and look forward to the impending public demo release!

- Might start adding a rudimentary version of the death points and upgrades which will include a profile system on the main menu. The profile system will allow players to start and store new profiles for future use.

**2024-09-11**

Got that part time job today! Also got an extra day a week at the second part-time gig and looks like schedules will fit together nicely. I will still have a decent amount of time to continue with the game and won’t need to be worried about being too frugal. I managed to get some things figured out today for player profiles!

- Made the controls panel a little nicer and added a basic keyboard button image for the Controls hotkey

- Adjusted some of the help text in the build menu and removed the ‘F’ key from the InputSystem for placing buildings

- Created a system for handling player profiles. The profiles are set up to save the profile name, Research Points, and max wave reached values as a copy of the profile itself in a JSON file. More fields will surely be added later.

- Created ProfileManager class to handle selecting, adding, and removing profiles.

- Created ProfileSelectionUI class to handle the Profiles Panel and all its elements. It is a basic version with a dropdown list for the profiles, and some text on top to show the stats of the profile. Later I plan to improve it to show off a scrollable list of profiles with all their data on their own panels.

Tomorrow:

- Start with creating a new class for handling death events, centralizing it like I should have off the bat.

- Implement changes to the maxWaveReached value of the current profile on death.

- Move on to adding Research Points

**2024-09-13**

Still managing to find some time each day to get some work done, looking forward to getting a better rhythm going with the new jobs.

- Added the DeathManager class to centralize the death events and actions.

- Profile now stores MaxWaveReached on death.

- Did up the Profiles UI panels with synced dropdown menu, I think they turned out good considering my energy levels these last couple days.

- Caused a stack overflow when I was adding the synced dropdown code, oops. Then I kept getting these memory errors of which I eventually learned were just remaining artifacts from the stack overflow. I was worried and frustrated for a while, so I hope I don’t forget about this side effect any time soon.

- Realized it will soon (like last month) be a good idea to have a centralized input manager class. This is something that will make life much easier moving forward.

- Found a bug with turrets suddenly not having a collider. The enemies wouldn’t attack them and could walk right through.

- Fixed the bug, it ended up being that the 2x2 wall image the turrets use as a base needed to have the ‘generate physics shape’ checkbox checked like everything else.

- I ended up losing a bunch of text I had written for the death messages at one point. I then opened another Unity editor to pull up an old version of the project. I decided to copy and paste the string lists over and continued on my way working on the project. Hours later (or the next day?) I tried to make a build of the game and ran into a sea of errors. Uh oh.

- Fixed: Unity crossed the referenced between the projects at one point, since two editors were open. I rebuilt the entire project asset library, and this fixed most of it. I then had to fix a handful of other small changes that were induced from the episode. All good now.

- Fixed issue with cursor images not loading in built version of the game, needed to force RGBA32 texture format and read/write access.

- Did some work to consolidate button style across the project. May add the style to upgrades menu tabs.

Tomorrow:

- Now that the game builds, and profiles are ready, I’ll aim to start implementing the augmentations and mods interface, which will be an entire scene of it’s own. The plan is for an AI ship that grows clones and assists with planetary research efforts. The ship will have sections in it for the different types of augmentations and mods, like Genomics Lab, Laser Physics Lab, Ballistics Lab, Structural Engineering, etc.

**2024-09-16**

Had a busy few days at work, but got lots done today! Augmentations are now a thing!

- Added tooltips to upgrades info images

- Drew artwork for a not-so-great looking mothership after failing to get AI to create anything remotely similar to what I was looking for. I’ll probably make it look a little nicer at some point.

- Designed a simple Augmentations UI to suffice for now. I plan to add the Labs later, to help organize Augs.

- Added color coding to cost texts for affordability. Added info panel.

- Implemented selected Augs count and max Augs with ability to add selected Augs to a list which will apply all Augs one gameplay scene load.

- Added Aug types to prevent selecting doubles of conflicting types

- Augs are scriptable children of an AugmentationBaseSO class, and it’s quite easy to create new ones.

- Made about 20 Augs with descriptions and all before I realized I should have structured them differently, and allowed for levels of each Aug. Could have saved an hour or two there. Live to learn!

- There are Augs which give boosts and cost Research Points, and ones that reduce stats but give RP.

- Started basic setup for Aug levels. Need UI buttons and more code.

- Current Aug types: BulletDamageStart, CreditsAdd, CreditsIncomeMult, GrenadeDeal, LaserDamageStart, MaxAugs, with Boost and Reduce Augs for each.

Tomorrow:

A full first day at the secondary part time job tomorrow, but I will have most of the evening to work at home.

- Aug levels implementation

- Aug UI additions and fixes

**2024-09-20**

It’s been difficult to find the energy at the end of each night to write the Dev Log up, and I’ve been starting to miss the clarity that daily logging brings. I also found out today that I work 6 days next week, so development will be slowed slightly. It’s okay though, I’m still getting things done every day. Here’s what’s been done since the 16th:

- Finished adding Aug levels and their UI elements, smoothed out all known bugs. The buttons, cost text, level text, description, and total cost/gain text all update their strings and colors to reflect positive/negative levels, as well as affordability.

- I likely made a poor design choice with the Aug levels. I kept all the positive or negative effects/levels on the same Aug scriptable object, so the coding was rather difficult with a tired brain juggling multiple logic paths. A simpler approach would have been to have separate objects for positives and negatives, although in their current state it is easy to create new ones.

- Created 6 different Augs: AddCredits, CreditsMult, BulletDamage, LaserDamage, GrenadeDeal, and MaxAugs. Fixed known bugs for Augs applying their effects.

- Changed design again by not having Max Augs, but instead Max Levels. This makes it harder to stack things up like crazy. Also gives more room for progression levels.

- Adjusted profile menu to have the dropdown menu as the main “Current Profile: “ name, saving space and confusion.

- Made grenades do structure damage, with a simple method that simply hit every tile and entity in range.

- Realized that was not good enough and instead used raycasts to the center of each tile in range, where structures block the explosion damage. It took a few tries to get it right, but this design works nearly perfectly so far. I had to make it so the enemies exact point location has to be within the structures grid cell’s bounds, so there is a slight discrepancy between the radius which grenades damage structures vs entities.

Tomorrow:

- Could make it so laser sword doesn’t hit through walls, unless player is elevated.

- Need to have the Upgrade Panel’s upgrade values change for the selected weapon or figure out some other way to show what the upgrade will do for that weapon…. Or each weapon…

- Could have the icons at top of upgrade panel selectable for the weapons, so each upgrade’s values can reflect that weapon’s changes.