Mutant Mayhem 2.0 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 the 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!

Recap:

Up to today I’ve been learning and tinkering lots 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. 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.

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 ton about referencing. Plus, I also learned about the things I forgot I learned about at this moment, lol.

2024-05-07

My good friend Paul and I came around with an idea. 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 being that I absolutely love it! There was 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 basic skeleton of a GUI for the building panel, 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 ive forced the raycast collision point furthing 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 half 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. 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 out 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 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 in at death I can see shop and upgrade panel getting built! 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!!

Work on the stats panel, 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 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 the 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 upgrade logics

Add multiple panels to 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 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