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. 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 system’s basics 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 and 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!