

KYLE

DOMENICO

Week 1: Didn't have major help with it except for ideas, but needed to do something for week 2 and week 3

I added a sort mechanism for the Shop to allow the shop to be sorted with S

The sort function in the shop organizes items by price, either from lowest to highest (ascending) or highest to lowest (descending). Each time you click the sort button (or press S), it toggles between these two orders and updates the display accordingly.

For the list this is how it works!

shopItems: list of all shop items

Sorted by GetPrice() (price of each item)

sortAscending: controls order (low to high or high to low)

After sorting, items are reordered in the UI

Week 2: I've modified some of the code into a linked list instead. I edited all the damage-related .cs files that added damage into a linked list. This means ALL the artifacts that added damage only, and not the ones that are chance-based. This also meant that I made an Infinite damage command by pressing a button. (This could also include the other artifacts because of the use of ArtifactManager.cs, but for now, we use one artifact as a starter)

So this implementation enables:

Ease of removal or damage to artifacts

The ability to add infinite damage, or later on, add the ability to add those artifact modifiers to the player.

Efficiency of adding or removing modifiers (Pure damage artifacts)

Possibility of stacking the artifact

So the infinite damage updates the ui for the stats that the player has on the left side and the top left I've managed to add an artifact at the press of a button, which shows the artifact but doesn't show the ToolTip, which serves as the info card of the artifact if you hover over it, and it

doesn't update the damage when I add the specific artifact from the actual pool of damage artifacts. (The artifact I was trying to add by a press of a button was the A_SharpenedSword.cs)

Steps:

Had to start off by editing the PlayerDamage.cs, then had to edit some of the artifacts to add them to the linked list. Then it was easy to implement the infinite damage but hard to implement the artifact add by pressing 2.

Linked List added for bats! Bats now spawn an additional time.

The bat linked list controls the order of enemy spawns.

Each bat is a node, and when one bat is defeated, it tells the next node in the list to spawn the next bat.

Basically making it respawn but in reality it is 2 individual bats. This also scales with the levels. If a bat is a lvl 1 bat it will spawn a lvl1 bat if it's a lvl 2 bat it will spawn lvl2 and so on

Week 3: I've made a behavior tree for the enemy's turn AI. So this would make them pick what to do based on the nodes given, and since they were turned into a behavior tree, I could implement a command that would disable their Turn AI and make them unable to attack the player. It's as if their turn was skipped. Dom and I worked and added some more nodes to this to make the game more interesting. I added some nodes so the player can pick at random to either attack twice/once, or defend and or heal.

So, how it works is that AI needs different nodes to do different things. For example, it needs a node to attack, one to defend, etc.

Then, it picks from those 3 depending on what it gets picked. Since it's randomized, it picks from the 3 different options I put on the list of nodes it can pick from.

Then it picks those for each turn.

(explanations of the other nodes here)

HealNode.cs: This is separate from the other 3; this will be a priority instead of being random. If the player damages the entity, it will prioritize

(combined randomized pick)

These 3 are picked through a randomizer

DefendNode.cs: What this does if this gets picked is that the player's damage gets halved

AttackNode.cs: Simple node, it just attacks the player based on the damage number given on

the entity card

AttackTwice.cs: This will enable the AI to attack twice.

For visuals, whenever the damage number pops up, it changes to the 50% defense, so if you were to put the max int damage, it would still divide it even if it one-shots the entity/entities

I had to fix some bugs on the defense and heal nodes

Steps: First, what I had to do was make some node files because it was easier to do when I was thinking about it. Then I had to edit the main files that were the TurnManager and the TurnStateEnemy.cs (mostly this). The node files I made were the AttackNode.cs, the behavior node, and the CheatManager.cs, which houses the cheat for disabling the AI. IsEnemyDisabled.cs. The SelectorNode.cs, the SequenceNode.cs. These were needed so it could select the nodes it wanted to select. Then I wanted it to select by randomizing the actions except for the healing.

I originally wanted the tree to be a skill tree and tried implementing special abilities but decided to wipe that portion and we decided the best idea was to work together on this portion due to its size.

I worked on the Heal Node and Defense Node. And then kyle helped me fix it up on his own machine .

Kyle worked on the initial tree data structure, and we have different commits due to fixing and working on the nodes together

Difficulties/Reflection

Kyle: I had difficulty figuring out something for the tree, but researched a bit and saw that I could make a behavior tree instead, which is good for AIs in general. We wanted to do something else, but couldn't make it work in the end. Debugging, for the most part, was long but rewarding, and of course, time. But I also had difficulty trying to add the Artifact by pressing a button, and it was still really hard because it clones the object instead of it already being there. I was only really able to make the Icon appear. I had difficulty trying to fix the HealNode.cs and the DefendNode.cs because it wouldn't work with debug.logs that I added. Some of them might have been deleted for less clutter. Then, after some hours of trying to fix it, it kinda works some of the time. Because in the debug log it would do it even if the enemy was hit by anything, even tho I hit a different enemy.

Domenico: I had difficulty figuring out an idea for the Tree or Graph, We wanted to do a skill tree initially but found that we could implement a behaviour tree instead for the mobs, this would allow us to bypass the UI interface needed for the skill tree, and allow us to know if the mobs are actually doing what they are supposed to with the console.

I also had difficulty understanding the way that the game is set up but with time it became easier. Also I had difficulty with the unity setup at the beginning and then Kyle aswell and it just felt like everything was going downhill. One day I was trying to find a way to do the tree and I was stuck and alone. But working with Kyle allowed me to push myself further and have someone to lean on when im struggling. Thank you Kyle

Peer or Teacher comments:

I added the Bat respawn system due to someone saying we should implement something to do with the Enemies, and we also thought of that with the tree.

The Skill tree, implementing a skill tree into this game made it feel unnatural. The Game is fast paced and the player shouldnt have to spend skill knowing he will probably die in a couple of turns.

What we couldve also done was build a prestige system. The system wouldve allowed the player to progress easier. But the game does have an end so we wouldve had to make the enemies harder ever so slightly.

It was really nice to have critical feedback during the weeks. This allowed us to not feel stuck or hopeless when doing commits. Also not being forced a big project and allowing to work on smaller projects(The commits) Made our lives a lot easier and felt like we were progressing.

Conclusions:

Kyle:

For the most part, editing someone's game to add more stuff was fun. If we had more time on this, we would have added more ideas to certain parts of the game. The idea of editing someone's game is a really good way to practice and to know what ideas you can add to an already-made game from someone else's save file.

Domenico:

I realised that editing someone else's work is hard, and commenting and organization are super important. It's like stepping into someone's brain and trying to pick it apart. Finding a way to implement things into the game was challenging but fun. There is no easy way to make games or edit games. Learning and making errors is normal.

I enjoyed this activity, and it makes me want to become a better game developer