Dragon Punch Battle Prototype Postmortem

The Game

As the developer on this prototype, my feelings are incredibly mixed. The sad truth is the strongest feeling I have from playing this game is one of yearning. I wish this had all the features I was imagining. From the job system to implementing combos and special moves, there’s so much to this game that is beyond my current abilities. Its frustrating to say the least. When I can divorce those feelings from the experience, I feel a little glad to see the Slime react and fight in different ways depending on the nuance of the Adjective System.

I was lucky enough to have people test out this prototype and they managed to have a lot of fun despite the above! They really liked the different props the Slime had depending on its adjective. The only big issue happened when a player accidentally got on the other side of the Slime, since it’s programmed to only move forward. Other than that it was fun watching people punch the little guy to bits.

When I get the chance to iterate on this game, I’m not sure how much of it will carry over. The basic concepts of stats, the Adjective System, and maybe the Slime sprite will be kept, but everything else needs to be retooled or overhauled to match the actual vision for the project. The stage needs to be bigger, the animations need to match the attacks more fluidly, and entire systems need to be built from scratch. Most importantly the actual combat system will have more depth. All these changes will make the game a better experience and closer to the product outlined in the GDD.

The Process

Writing this game design document was an easy endeavor for the most part since I have previous experience. How else could you explain how organized it was? I wasn’t as knowledgeable on the systems descriptions or the way those diagrams are laid out, so this was still a useful learning experience. Of course the GDD will never be done as long as the game is in development, so in truth I can never write the entire experience of working on this living document.

Creating a game is always easier with a good GDD. Working with Unity has made it even easier, once I learned the magic words that appeased the beast and let me put a health bar on the screen. That part is what makes Unity hard to work with: knowing what part of the code is saying what. For example, I wrote some code that checked the collider of another GameObject to see if the two overlapped and then go from there. Except that wasn’t the right incantation; Unity wanted me to check the collider of the collider. Once I did that the magic worked and the game responded as expected. One word was the difference between a well-oiled machine or just spinning my wheels in the mud. Still without Unity I’d have to program a whole lot more, like how to make a collider in the first place. It’s easier to learn how to play nice with Unity than trying to figure out the multitude of variables and situations that could arise from making an engine myself, let alone an entire game on top of that.

Playtest Notes

Hearts not resolution independent

Player loved the little props for the slime

Felt like the Adjective System made for more replayability

Liked the props