Lucas Rios - Development Process for Blue Gravity Shop

After reading the document, I began writing on paper some of the mechanics I would need to create. Since the document focuses on both programming and design, I started to shape the system around these two factors, aiming to make it simple, expansive, and visually appealing.

My first approach was to develop the player sprites' behavior, as this is a significant part of the game. I chose to go with the most debuggable method (not necessarily the most designer-friendly) and handle all the animations through code. By using scriptable objects with "Sprite Blocks" containing all the sprites related to a state (idle, walking, running) and direction (north, south, east, west), I gained better control over the player sprites and improved code organization. This approach also facilitates the easier creation of additional animations in the future.

Expansion was my main objective. I love working with events and scriptable objects because they fit well in a system that needs to be loosely coupled, like this one. Of course, they have their drawbacks, like any other approach.

Every system I created, whether it was the character window, inventory window, shop window, or player movement, uses events that any component can listen to. Setting this up in my debug scene took a lot of time initially, but it paid off when I integrated everything into the main game scene later.

After creating the system for displaying sprites on the player, I started developing the trade system, which closely relates to the inventory system that I created afterward. With these two systems in place, I felt the need for a window to show items equipped on the player. As a former player of Ragnarok Online, I tried to replicate its style.

With these three systems in place, I just needed to integrate them into a real scene. So, I created the scene, developed a basic movement system for the player, added interactions with NPCs, and took my time to create an environment that was nice to look at.

From there, I focused on updating the design of the windows. My objective was to ensure the player character was never obscured, even with all the windows open. This design should work perfectly with the most common resolutions found in gaming today.

After that, the clock was ticking: some finishing touches, final tests, and minor fixes, and it was ready.

I feel my performance on the programming side was on par with what I normally achieve. Of course, time waits for no one, and towards the end, things got a bit rushed. However, overall, I believe I created a robust system that is loosely coupled and pleasant to work with. Considering the 48-hour timeframe and without reusing any code from the internet or other projects, I'm very happy with the outcome.

On the design side, I think I did okay. It could have been better with more time, but I chose to allocate more time to programming. However, I'm pleased with how the environment turned out, especially considering it was my first attempt at using tile maps. I also find the UI to be very cozy and inviting.

P.S.: I didn't create any of the game art myself.