

Hello to everyone on the Blue Gravity development team. I'm Matheus, and I spearheaded the development of the OutfitMaker prototype, aiming to capture the gaming essence of titles like Stardew Valley, The Sims, and Pokémon. In this virtual world, players can buy, sell, and deck themselves out in various clothes and items.

As a devoted fan of Stardew Valley, Palia, and Pokémon, I found creating this prototype both exhilarating and challenging. Despite the difficulties, the end result brought me great satisfaction, especially considering it was my first foray into such a project—a feature I hold dear from my favorite games.

The process of creating this prototype was straightforward. Upon reviewing the instructions, I quickly grabbed a notepad and jotted down ideas for implementation—like a dialogue system, an interface reminiscent of Stardew Valley, NPCs, and collectible items. Essentially, anything tied to the mechanics was fair game. This approach allowed me to break down these features into smaller, manageable parts, progressively assembling them into a cohesive whole.

In this brief text, I'll delve into the more intricate mechanics. Firstly, the dialogue system: I envisioned a player-NPC interaction where players could decide whether to follow the characters' requests. I created a versatile Dialogue class, customizable for each NPC or object requiring dialogue. Additionally, I implemented a DialogManager referencing the Dialogue, handling tasks such as typing out dialogues and orchestrating the flow of each dialogue line.

The equipment swapping system, coupled with real-time animations, presented a genuine challenge. While a bone-based solution was a simpler alternative, I opted for pixel art, influenced by my love for Stardew Valley. I utilized a parent object with an animator featuring different layers for each body part—represented by hair and clothing. The children of these objects held the sprite renderers, ensuring a harmonious blend of simultaneous animations of multiple body parts within one animator. From there, I incorporated Scriptable Objects to create items (hair, clothing) and devised mechanisms for changing these animations at runtime, along with the ability to purchase and equip these items.

Best Regards, Matheus Santana.