

Blue Gravity Studios Task

Diana Ramos Garcia

System Explanation

Explanation of the item store prototype system and character customization.
For this specific case, I used the pixel art character of the examples you provide.

Pre-Written Code: The scripts that were not created within this test are located in the "External Assets" folder. They mostly consist of classes from my own library.

Animation and Character Movement System

The character contains 4 movement animations and 4 Idle state animations.
The animations are connected to each other using Blend Trees and Animations Layers .
When the Player decides to move in the 4 axes, a Vector2 is sent to the blend tree, which detects if the player's direction is positive or negative.

The animator controller has 3 layers, each with an additive blending type to combine the animations within the animator.

Outfit Change System

Having used a pixel art style, it was decided to combine complete outfits through animations to facilitate the exchange of sprites and to maintain the movement functionality connected with the animations.

I divided the character's body into 3 parts, the Body, Torso, and Hair, each with a sprite renderer responsible for displaying the character's body. The animator controller references the sprite renderer to play each animation for each part of the body.

The Character Parts Manager class contains a ScriptableObject that stores each part of the character's body. In the BodyPart ScriptableObject, it stores the Idle and Walk animations for each outfit and hair. The manager class is responsible for overriding the animation clips, which it has to find in the Resources folder. This class listens to events to update the animations when the character's parts are changed.

Class Character Part Selector receives the updates that the player selects from the ClothStoreUI. The selector finds the index of the CharacterPartData and sets the new value to perform the override in the manager.

Clothe Store UI

The UI contains the scriptableObject of the store costs and the icons of each item. Upon initialization, the UI automatically assigns the values and an index to the buttons.

Hair Color Change

The UI buttons update the player's sprite renderer and assign it the selected color.

Thought process during the interview

During the test, I encountered two main issues. The first one was finding the most optimized way to store and set the character's body part animations in order to avoid having multiple Animator Controllers. However, I managed to find a solution with the override of the animator controller, which allowed me to contain all the player's animations in just one.

The second problem I faced was the complex task of creating 8 different animations for each layer of the character. Since it was pixel art, each animation consisted of 6 frames, and for each clothing element, I had to create 4 animations for Idle and 4 for Walk, one for each direction. This became a process that took up 50% of the test time.

Performance Personal Assessment

I received the test at 5 am (GMT-6) on May 12, but I started the project at 11 am. From there, I started counting the hours.

End time: 12 am on May 14.

The total number of active hours was: 26 hours.

I consider that during the given time, the project was successfully completed with all the necessary specifications. Additionally, I managed to add some extra gameplay details. The only limitation I encountered for my project was the limited access to sprite assets available for my player. That's why the player only has the option to switch between 2 outfits and 2 hairstyles.