

Task 4 - Model Answer

Before designing a character creator, we need to first consider ways in which the player should be able to interact with it. To create a truly customizable experience, the player should be able to choose from a selection of individual traits and apply them to their character. Trait options should include physical characteristics such as body type, facial features, skin color, and hairstyle, as well as additional characteristics like clothing and accessories.

When adjusting the body type of the character, the player will ideally have access to a spectrum of features, rather than a set of predetermined choices. Typically, this approach allows the player to adjust different components of the character's body across a scale of values. Allowing the player this degree of freedom requires some technical considerations, since we'll want to ensure that animations and character movements function correctly regardless of the character build selected.

The most common way to handle this is to break the skeletal mesh for the character into separate components, rather than treating it as a single mesh. For example, the skeleton might be broken into separate components for the torso, legs, arms, and head. Each of these components can then be adjusted individually before being assembled and animated.

Many game engines provide their own workflow for doing this, but we'll focus on the Unity Engine as a common open-source engine for game development in C++. Unity Engine provides a Mesh Merge functionality in Blueprint that allows these separate meshes to be easily combined into a single object. Additionally, Unity Engine provides the ability to dedicate one component of the skeleton as the Master Pose Component, which then serves as the ground truth for character animations. For example, if the torso component was labeled as the master component, then when an animation is triggered in the torso, the paired animation will be triggered in the subcomponents. This functionality is incredibly helpful for ensuring that animations remain synced between skeletal components.

Once we have the body type working in the character creator, we can then begin to consider how we will map physical features and clothing to the character skeleton. Typically, this is performed through a process called "skinning", where a 3D mesh is attached to an existing 3D form. The 3D mesh must have vertices that match to the joints in the model. This generally requires some manual work to ensure that each of the meshes fits correctly to the model. This technique can be applied to physical features as well as clothing and accessories. By configuring this system, the player will be able to select between different characteristics like facial features, skin color, hairstyle, and clothing components, and have each characteristic be mapped independently to the skeleton.

With these techniques, we can build a modular character creator that gives the player a great deal of control over character design.