Standard Characters & Cameras

For the website version, see: https://philsa.github.io/rival-doc/standard-characters.html

These packages contain the scripts & prefabs required for basic plug-n-play first person and third person character controllers, along with their cameras & rudimentary input handling. You can extract the packages into your project, and start customizing the characters & cameras from that starting point.

Main Camera

Since Cameras can't be converted yet in DOTS, we have to do a little workaround to let us implement camera controls in components & systems:

- Have a regular GameObject camera in your scene (not in a subscene)
- Add a `MainGameObjectCamera` to that GameObject. This will tell the `MainCameraSystem` that this is the camera we want to use for rendering
- Add a `MainEntityCamera` to a Subscene GameObject that is meant to represent the controllable camera entity.

Every frame, the `MainCameraSystem` will then copy the position/rotation of the `MainEntityCamera` singleton to the `MainGameObjectCamera`

First Person Character

Setup

Here is how to set up a First Person Character in your Subscene:

- Add the `FirstPersonCharacter` prefab to your Subscene
- Add the `FirstPersonPlayer` prefab to your Subscene
- Assign the `FirstPersonCharacter` to the `ControlledCharacter` field of `FirstPersonPlayer`
- The `View` object under the `FirstPersonCharacter` is what should represent the camera. You can add a `MainEntityCameraAuthoring` component to it in the inspector, or add the `MainEntityCamera` component to it via script

For more information, see the First Person Character section on the documentation website

Third Person Character

Setup

Here is how to set up a Third Person Character in your Subscene:

- Add the `ThirdPersonCharacter` prefab to your Subscene
- Add the `ThirdPersonPlayer` prefab to your Subscene
- Add the `OrbitCamera` prefab to your Subscene
- Assign the `ThirdPersonCharacter` to the `ControlledCharacter` field of `ThirdPersonPlayer`
- Assign the `OrbitCamera` to the `ControlledCamera` field of `ThirdPersonPlayer`
- Assign the `ThirdPersonCharacter` to the `FollowedCharacter` field of `OrbitCameraAuthoring`
- Optional: on the `ThirdPersonCharacter` object, you can add a `CameraTarget` component to specify a child transform of the character that will act as the real target of the camera.
- You can add a `MainEntityCameraAuthoring` component to the `OrbitCamera` in the inspector, or add the `MainEntityCamera` component to it via script

For more information, see the Third Person Character section on the documentation website