



MultiFPS – Project workflow

This document describes structure of MultiFPS and naming conventions for assets.

Contents

AnimatorControllers and AnimatorOverrideControllers	2
1. Item_BaseController	2
2. Soldier_EmptyHanded	2
Contact:.....	4

AnimatorControllers and AnimatorOverrideControllers

There are two base animator controllers that MultiFPS uses.

1. Item_BaseController – as name suggests this controller is a base for items animator controllers, they override this controller with their own animations. For example, in one animator for knife in place where we have stabbing animation we can have in another animator melee attack with gun like pistol, sniper shotgun etc. So, we can launch those different animations in different weapons using the same code and we don't have to create every time each animator controller from scratch for each weapon. But we can do this if we want, when weapon is much different than the rest and does not match our base animator controller. Shotgun has its own controller since it is reloaded differently than the rest of the guns, so it needed different approach to animating.

First person perspective arms model also uses the same animator controller as a base, since arms animations in FPP directly correspond to item animations, weapon is being reloaded, so hands need to play performing reload animations.

Naming convention: Item_**ItemName** , for example "Item_**pistol**", "Item_**rocketLauncher**"

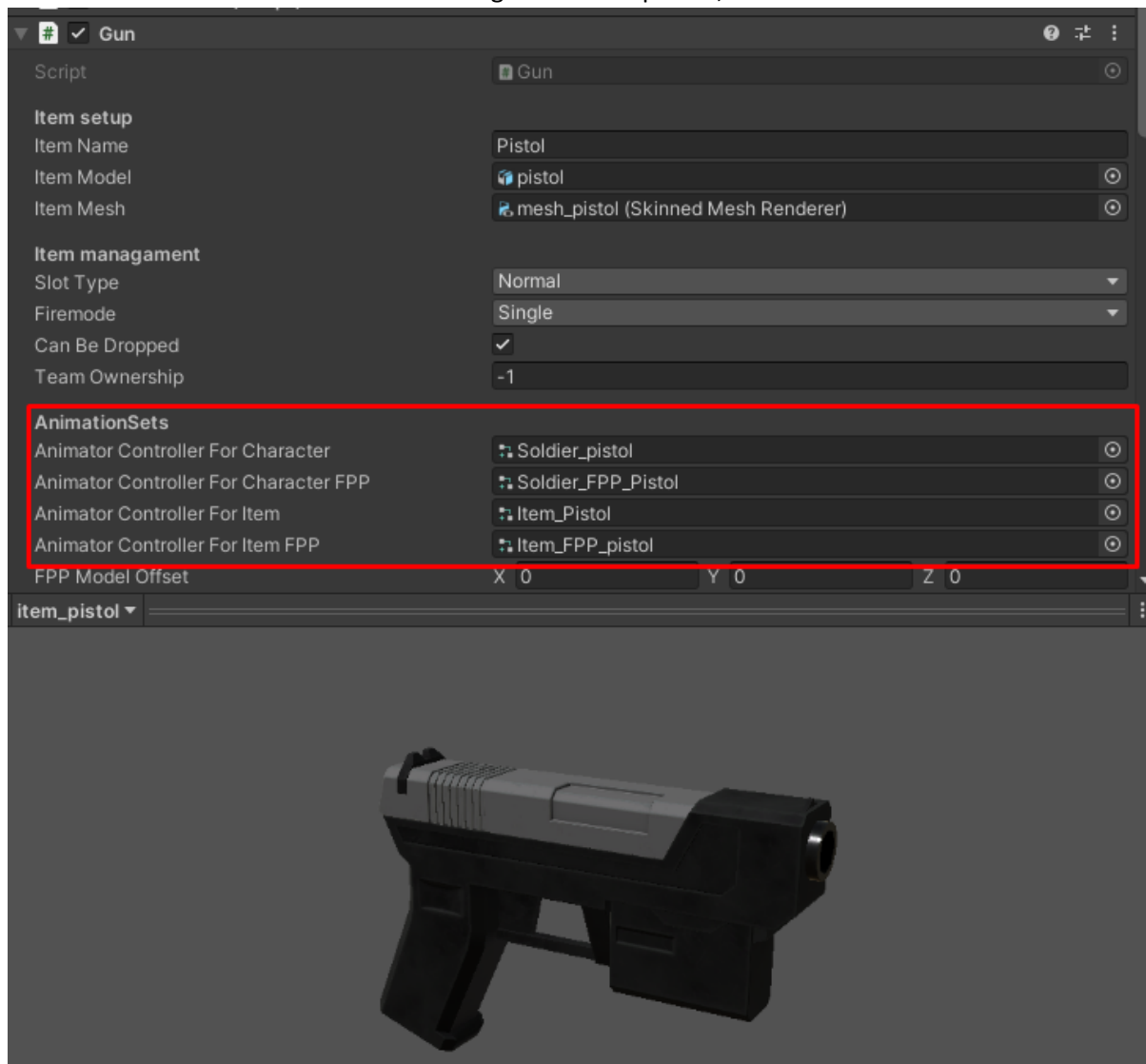
Naming convention for items viewed from first person perspective: **Item_FPP_ItemName** , for example: "**Item_FPP_pistol**", "Item_FPP_**rocketLauncher**"

2. Soldier_EmptyHanded – it is base controller for humanoid character model. It contains slots for animating walking, crouching, running, looking up/down (to achieve that, animation skeleton is separated in to two parts using AvatarMask). It also has slot for receiving damage, punching and reloading item animations. Items can place in character animator AnimatorOverrideMask that overrides Soldier_EmptyHanded controller with its own unique animations. Thanks to that each item has different melee attack animation, reload animations, looking up/down animations, and it is also possible to change movement animations dependent on item for example heavy item could implement heavier walking animations, but this is not used anywhere for now.

Naming convention: Soldier_**ItemName** , for example "Soldier_**pistol**", "Soldier_**rocketLauncher**"

In this example "Soldier_Pistol" is controller that override "Soldier_EmptyHanded" controller with animations specific to pistol.

Each of those animator controllers are assigned to item prefab, like in this screenshot:



Here for example we have prefab for pistol, and in section “AnimationSets” it stores animator controllers for:

Animator Controller For Character – Animator that humanoid character model will use to be animated appropriately to pistol

Animator Controller For Character FPP – Animator that FPP arms model will use to be animated for using pistol

Animator Controller For Item – Animator controller for item being hold by full humanoid model, so here we just animate moving parts when player shoots or reloads

Animator Controller For Item FPP - Animator controller for item being hold in first person perspective by arms model.

Thank You for using MultiFPS

Contact:

Email: desnetware@gmail.com

[Website](#)

[Youtube channel](#)

[AssetStore publisher page](#)

2023 desNetware