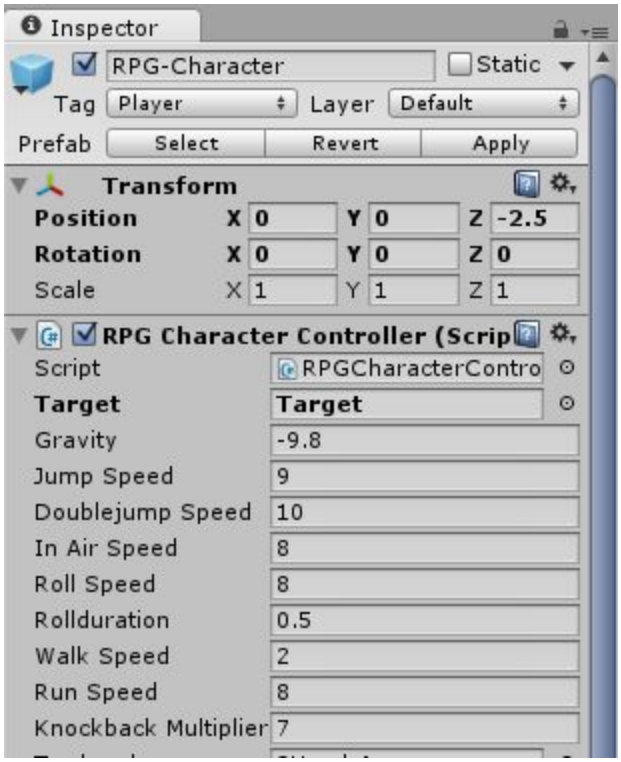
RPG Character Mecanim Animation Pack ReadMe

Last Updated **2016-3-29**

Hey, first off thanks for purchasing and using this pack! It is highly recommended to watch [Unity's Animation Tutorial Videos] before using this asset.

Installation

Before attempting to use the pack, you must first ensure that the tags and inputs are correctly defined. There is an included **InputManager.asset** included in the **InputManager.zip** file that contains all the settings. Copy these to your ProjectSettings folder.



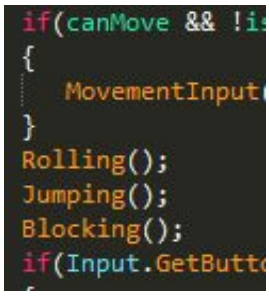
The basic movement attributes of the character can be modified via values in the Inspector, as well as jumping and rolling attributes.

Knockback Multiplier is a force that is directional force that is applied to the character when the GetHit animations are played, and correspond to front, left/right, and back directions.

Control inputs are all handled in **Update()** or referenced methods. **FixedUpdate()** contains all the physics dependant functions. **LateUpdate()** handles all the calls to the Animator.

UpdateMovement() is where all the movement processing of the character is handled, and it uses **GetRelativeCameraMovement()** to make all directions based off the camera. Character faces the movement direction via **RotateTowardsMovementDir(),** which becomes disabled when characters are strafing.

All the jumping methods use **CheckForGrounded()** to determine if the character is on the ground to set whether it can jump, etc.



If you wish to remove Jumping, Rolling, or Blocking from your character's actions, you can remove these code lines. AirControl can also be removed by deleting **AirControl().**

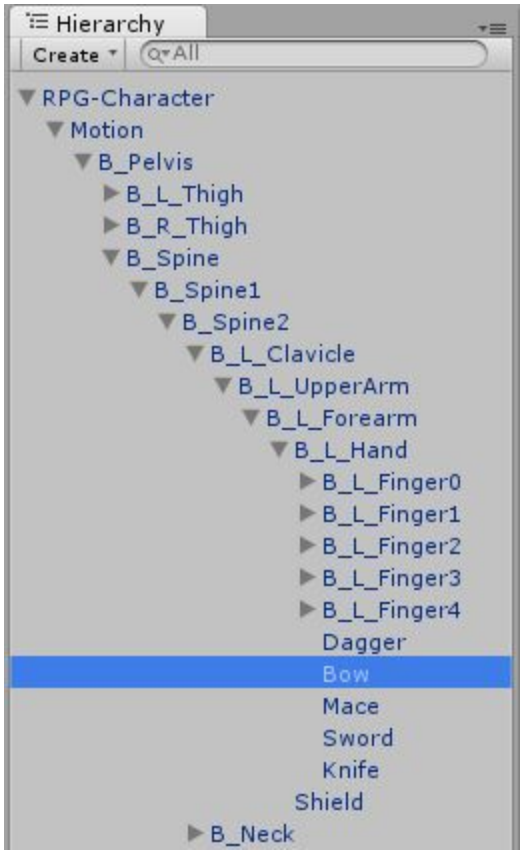
Rolling() allows freedom of movement when rolling, and then picks the closest corresponding rolling animation to play. The GUI commands for rolling are based off the direction the character is facing, and you can use these if you don't wish to allow rolling in any direction.

Coroutine **_LockMovementAndAttack()** is used by attacks and other action animations, which locks input and movement, while also enabling Root Motion on the Animator.

Coroutine **_SwitchWeapons()** handles all the weapon switching logic and is based off a weapon integer value, and also off leftWeapon and rightWeapon variables.

0 = Unarmed
1 = 2 Handed Sword
2 = 2 Handed Spear
3 = 2 Handed Axe
4 = 2 Handed Bow
5 = 2 Handed Crossbow
6 = Staff
7 = Armed / LeftHand Shield

8 = LeftHand Sword	9 = RightHand Sword
10 = LeftHand Mace	11 = RightHand Mace
12 = LeftHand Dagger	13 = RightHand Dagger
14 = LeftHand Item	15 = RightHand Item



To replace your character's weapons with your own models, place them in the hierarchy under the B_R/L_Hand nodes, and then update the appropriate weapon slot in the Inspector.

Any questions about the Pack, please Email.

Thank you!