



## RPG Character Mecanim Animation Pack ReadMe

Last Updated: **August 30, 2018**

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 if you're not familiar with Mecanim.

### Controller Overview

The RPG-Character Mecanim Animation Pack controller is composed of 5 required scripts:

**RPG Character Controller:** Main controller.

**Super Character Controller:** Handles ground detection and surface interaction.

**RPG Character Movement Controller:** Drives the character in various movement states.

**RPG Character Input Controller:** Used to set the control inputs for directing the character.

**RPG Character Weapon Controller:** Contains all the weapon information and handles weapon switching.

The 3 following scripts are optional:

**PerfectLookAt:** Makes the character look at selected target.

**IKHands:** Automatically attaches the left hand to 2 handed weapons using IK.

**GUI Controls:** This script contains all the needed code for the UI for demo purposes.

All scripts in the RPG Character Mecanim Animation Pack use the namespace:  
RPGCharacterAnims

The **Rigidbody** on the RPG Character is only needed if you want the character to interact with other physics objects, the **Nav Mesh Agent** only needed if you plan to use pathfinding on your character, and the **Capsule Collider** only required if you need player collisions.

## Setup

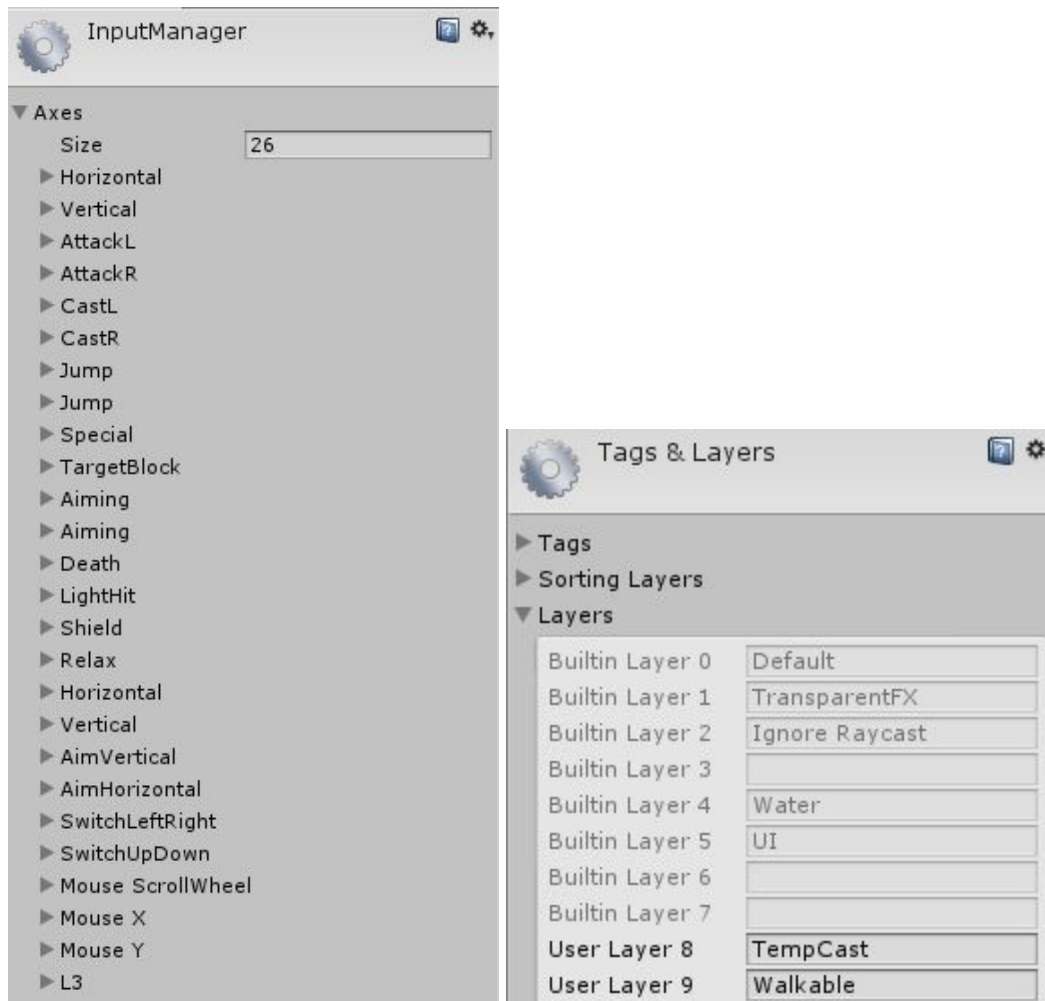
### Video Tutorial

Here's a video showing how to setup your own custom character using the RPG Character animations: <https://www.youtube.com/watch?v=l8V6SL7Oy5M>

### Pre-Installation

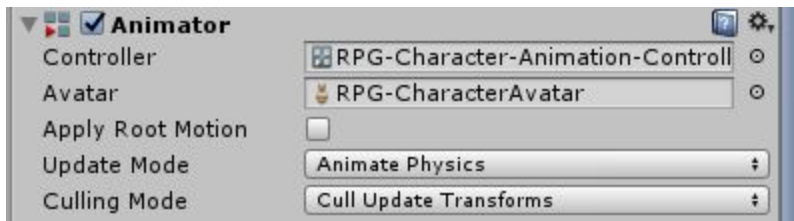
Before attempting to use the pack, you must first ensure that the tags and inputs are correctly defined. There is an included **InputManager.asset** and **TagManager.asset** included in the **ProjectSettings.zip** file that contains all the settings. Copy these to your ProjectSettings folder.

The required inputs and tags are as follows:



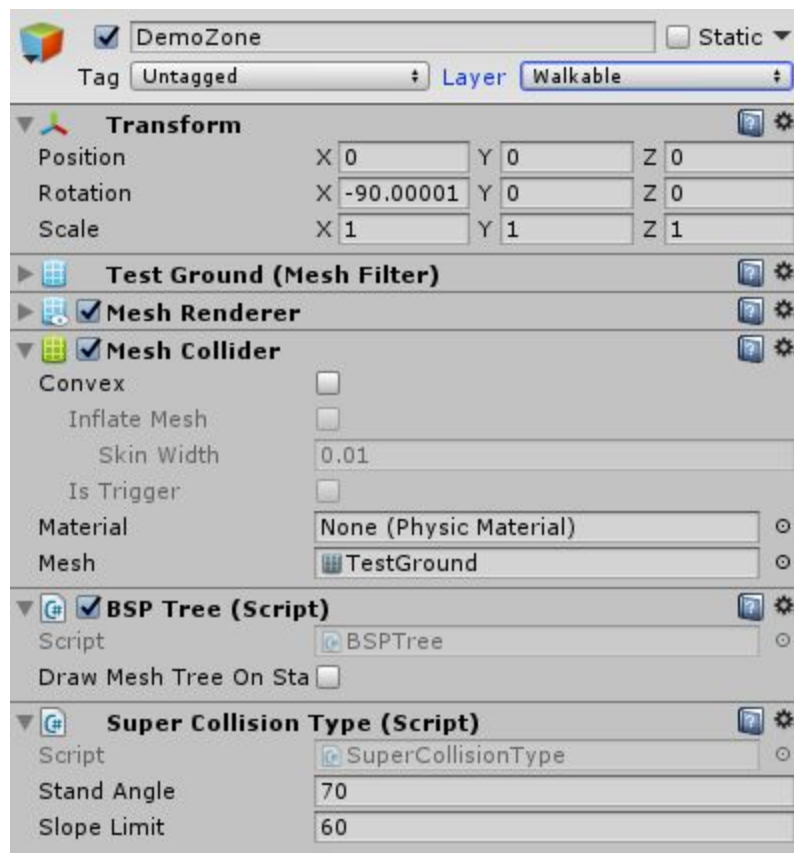
## Set the Animator Controller to RPG Character Animation Controller

The animator should be set with 'Apply Root Motion' disabled, and Update Mode to 'Animate Physics'.



## Setup World Colliders

For any objects that you want the character to walk over or collide with, set them as Layer: Walkable, and make sure this is set the same in the SuperCharacterController script. Objects with primitive colliders on them such as sphere, box, or capsule colliders won't need any additional settings, but any object with a *Mesh Collider* needs the BSPTree script attached to it. If you want to control the allowable slope height for the object, you can attach the SuperCollisionType script to it as well.



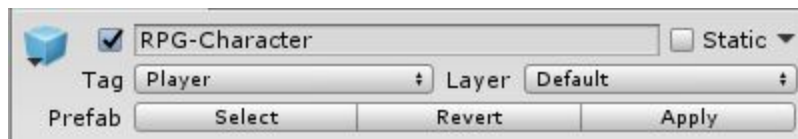
## Add and Set Target

The RPGCharacterController script needs a target object for purposes of strafing, and if you're using the PerfectLookAt controller, this also needs a target.



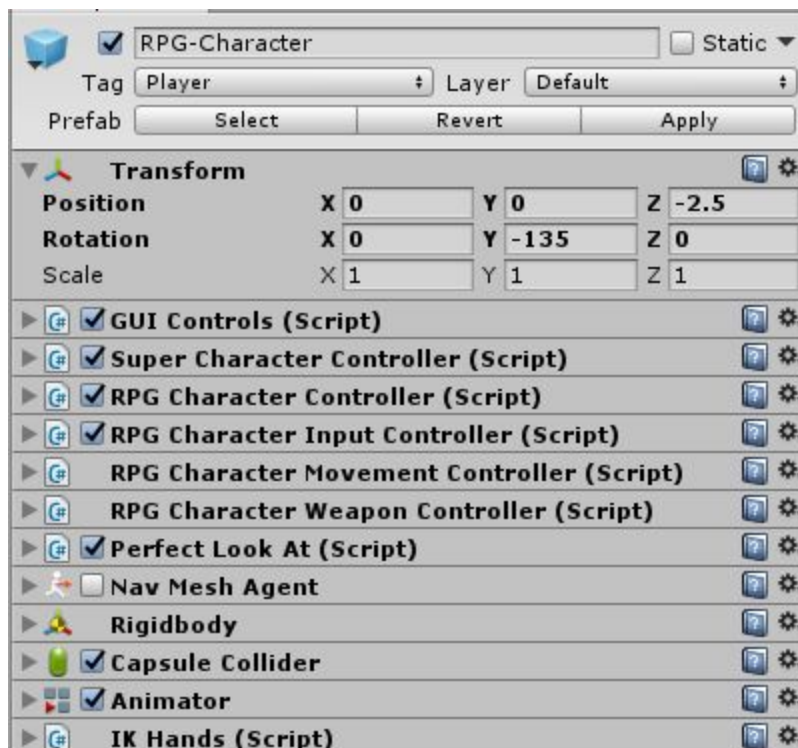
## Tag Character as Player

If you're using the included camera script, you'll need to set the character's tag to: Player.



## Copy Components

Copy all the components over from the RPG-Character prefab to your new character. If you don't want the GUI, PerfectLookAt, IKHands, and/or NavMeshAgent scripts these are all optional.



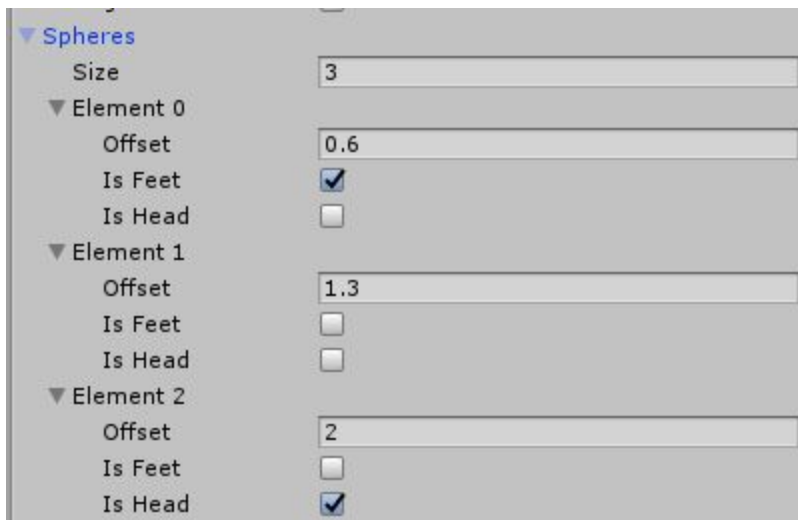
## Set IK hand

If you're using IKHands script then you need to set your character's left hand joint as the scripts Left Hand Obj.

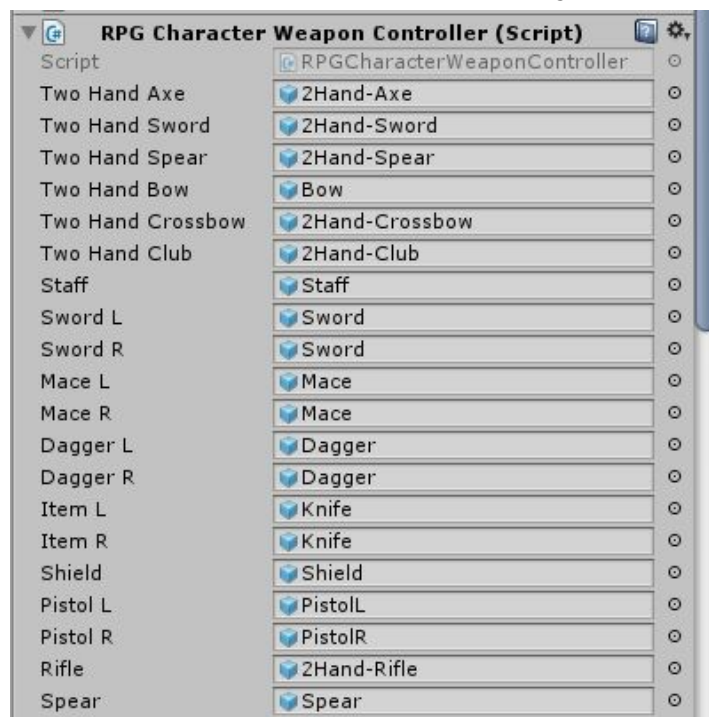


## Adjust Collider and Super Character Controller Spheres

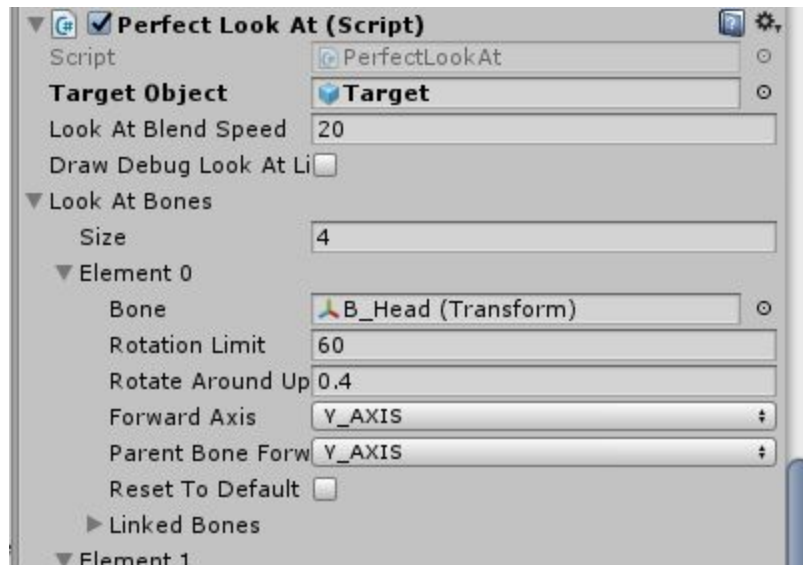
Adjust the Capsule Collider for your character if needed, and also adjust the SCC Sphere's objects to proper size and position.



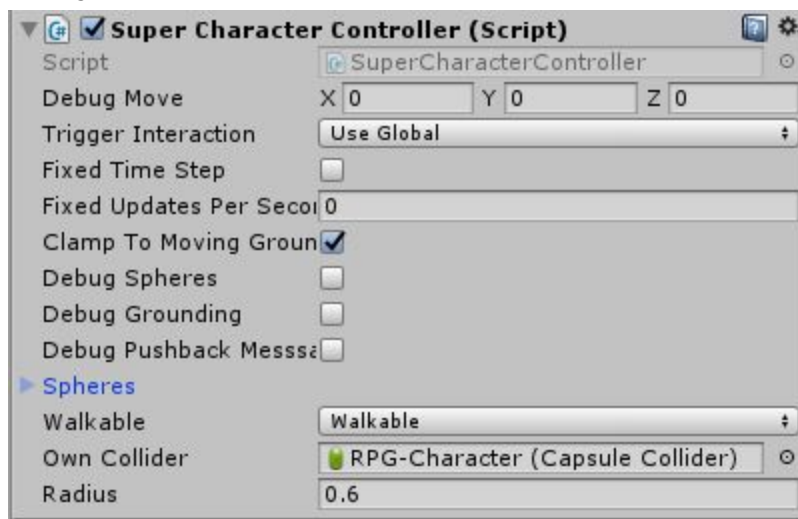
## Move and/or Replace Weapons on New Character and Copy Weapon Parameters



### Change LookAt bones and Target.



### Set SCC Walkable Layer and Own Collider.



### Position/Scale Weapons in Hand.

This is recommended to finalize during runtime so you can see how the weapon is positioned in the characters deformed hand, and then copy the transform and re-paste it back when the game is not running.





### Position/Rotate Attachment Points.

The same process is followed for placing the weapons in the characters hands. Position the Attach point, which will be the first child of the weapon game object, and then copy the transform when the game is not running.



## Weapon Switching

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

//0 = No side

//1 = Left

```
//2 = Right
//3 = Dual
//weaponNumber 0 = Unarmed
//weaponNumber 1 = 2H Sword
//weaponNumber 2 = 2H Spear
//weaponNumber 3 = 2H Axe
//weaponNumber 4 = 2H Bow
//weaponNumber 5 = 2H Crowwbow
//weaponNumber 6 = 2H Staff
//weaponNumber 7 = Shield
//weaponNumber 8 = L Sword
//weaponNumber 9 = R Sword
//weaponNumber 10 = L Mace
//weaponNumber 11 = R Mace
//weaponNumber 12 = L Dagger
//weaponNumber 13 = R Dagger
//weaponNumber 14 = L Item
//weaponNumber 15 = R Item
//weaponNumber 16 = L Pistol
//weaponNumber 17 = R Pistol
//weaponNumber 18 = Rifle
//weaponNumber 19 == Right Spear
//weaponNumber 20 == 2H Club
```

## Animation Events

Note that there are animation events for all animations, and you'll need methods in a script attached to the same component as the Animator otherwise you'll get warnings.

```
//Placeholder functions for Animation events
public void Hit(){
}
public void Shoot(){
}
public void FootR(){
}
public void FootL(){
}
public void Land(){
}
public void WeaponSwitch(){
}
```



## Animator Parameters

**Moving:** Set in RPGCWeaponCtrl if there's movement input and character motion.

**Strafing:** Set in RPCCharCtrl if pressing strafe.

**Aiming:** Set in RPCCharCtrl if aiming.

**Stunned:** If true, GetHit transitions to Stunned animation.

**Shield:** Set in RPGCWeaponCtrl if shield is equipped.

**Swimming:** Set in RPGCMoveCtrl if character collides with a water volume.

**Blocking:** Set in RPCCharCtrl if pressing block.

**Injured:** Plays injured idles and walking animation.

**Crouch:** Plays crouch animation on idle.

**Animation Speed:** Global adjustment.

**Weapon:** Set in RPGCWeaponCtrl, same as WeaponSwitch function.

**WeaponSwitch:** Used for determining which state weapon switch happens from.

**LeftRight:** Set in RPGCWeaponCtrl to determine which side for 1 handed weapons.

**LeftWeapon:** Set in RPGCWeaponCtrl, same as WeaponSwitch function.

**RightWeapon:** Set in RPGCWeaponCtrl, same as WeaponSwitch function.

**AttackSide:** Set in RPCCharCtrl and determines left/right animation.

**Jumping:** Set in RPGCMoveCtrl, 0 grounded, 1 jump, 2 falling, 3 double jump.

**SheathLocation:** Set in RPGCWeaponCtrl, determines weapon location.

**Talking:** Set in RPCCharCtrl to determine which talking animation to use.

**Velocity X:** Set in RPGCWeaponCtrl, character's sideways speed.

**Velocity Z:** Set in RPGCWeaponCtrl, character's forward/backward speed.

**AimHorizontal:** Used for Rifle, 2H Bow, and Crossbow for horizontal aim blend.

**AimVertical:** Used for Rifle, 2H Bow, and Crossbow for vertical aim blend.

**BowPull:** Blends pull back animation for 2 handed bow.

**Charge:** Blend charge animation for running with Shield.

**Action:** This is used by the various animation triggers to determine which animation to play.

Any questions about the Pack, please [Email](#).

Thank you!

