THIRD PERSON SYSTEM

User's guide

Content

1. INTRODUCTION

This document will guide you through the essentials of Third Person System. Third Person System is a package to help you create Third Person games, with fluid and smooth gameplay. It was developed to be easy to setup and work with.

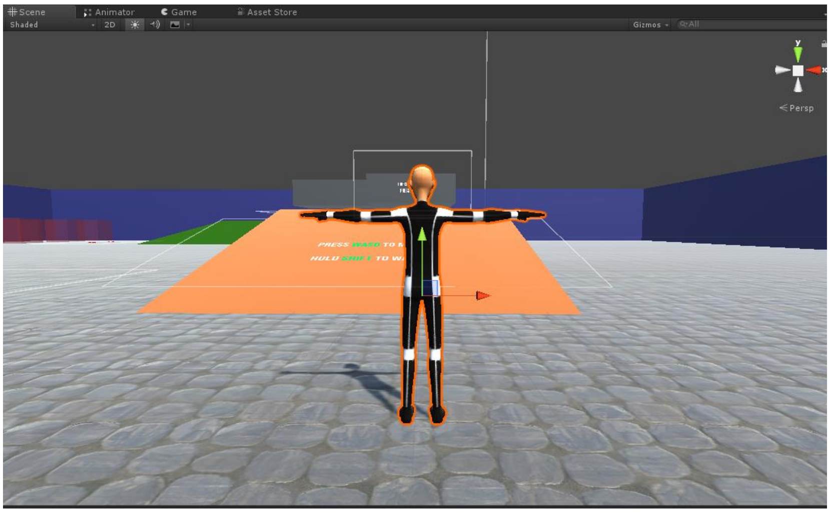
2. QUICK START

Third Person system was designed to be fast to setup a new scene. Remember that your character model must be setup as Humanoid.

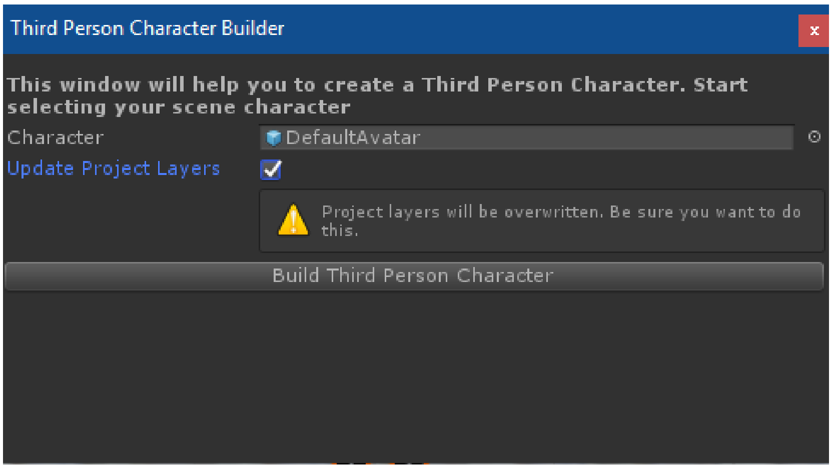
Watch this video showing how to quick setup your scene:
https://www.youtube.com/watch?v=oFXqdEIYyyI

Follow the steps:

1. Drag your character model to your scene.



2. Go to: Tools -> Third Person System -> Build Third Person Character
3. Drag your character to the field and press Build Third Person Character. If you are using this window for the first time, it's recommended to check *Update Project Layers* box. It will set layers that system needs to work.

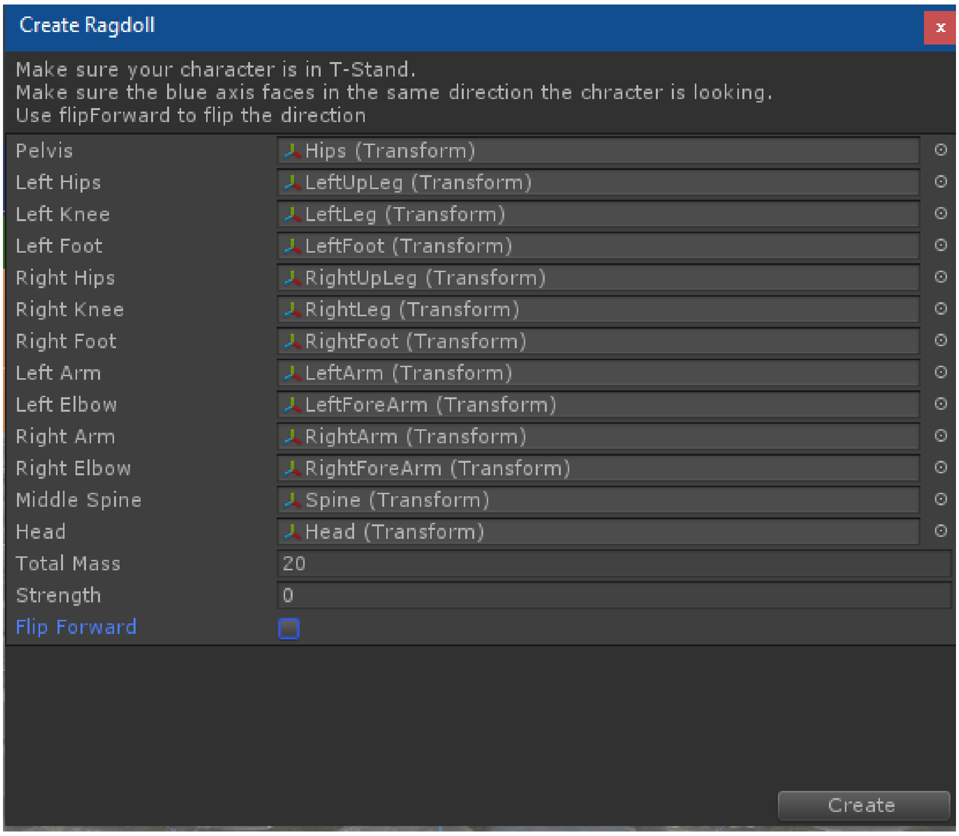


This window will help you to create a Third Person Character. Start selecting your scene character

Character	DefaultAvatar
Update Project Layers	☑

Project layers will be overwritten. Be sure you want to do this.

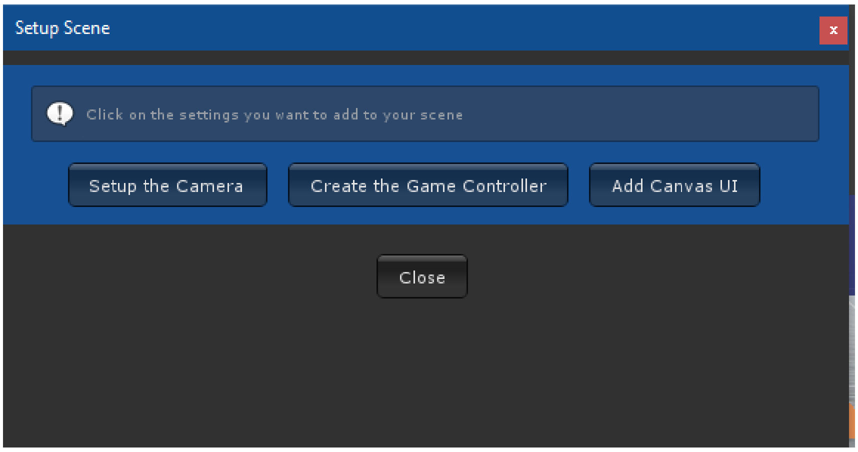
4. A Ragdoll window will open. Change settings that you want and press Create. (Create button will be enabled only after change something. You can check and uncheck Flip forward to enable the button)



Make sure your character is in T-Stand.
Make sure the blue axis faces in the same direction the chracter is looking.
Use flipForward to flip the direction

Pelvis	Hips (Transform)
Left Hips	LeftUpLeg (Transform)
Left Knee	LeftLeg (Transform)
Left Foot	LeftFoot (Transform)
Right Hips	RightUpLeg (Transform)
Right Knee	RightLeg (Transform)
Right Foot	RightFoot (Transform)
Left Arm	LeftArm (Transform)
Left Elbow	LeftForeArm (Transform)
Right Arm	RightArm (Transform)
Right Elbow	RightForeArm (Transform)
Middle Spine	Spine (Transform)
Head	Head (Transform)
Total Mass	20
Strength	0
Flip Forward	□

5. Now you need to setup your scene. Be sure that there is no Cameras attached to your scene.
6. Go to: Tools -> Third Person System -> Setup scene



Click on the settings you want to add to your scene

7. Press the three buttons to setup your scene. Close the window.
8. Press play and enjoy!

3. WORKING WITH CAMERAS

Third Person System works with Camera Data. Camera Data is a Scriptable Object that stores Camera Offset and Field of View. You can use a specific camera data for each ability and situation.

Watch this video showing how to work with camera data:

https://www.youtube.com/watch?v=e_XipD2YkJg&index=4&t=0s&list=PLiRDro5YzN41liCQ9cAy0EGCBIX-LwC04

4. ABILITY SYSTEM

Third Person System allows you to create new abilities easily. To understand how the system works, you can watch this quick video tutorial: https://www.youtube.com/watch?v=25rsKIb4sKg

Any ability has common parameters to be used. These parameters are divided in sections: **Ability settings** (for specific parameters for that ability), **Animation**, **Camera Settings**, **Abilities to ignore** and **Events**. All common parameters are listed below:

Ability Physic Material: this parameter overrides Capsule Collider material when the ability is running. You can find situations that you want your character use different friction properties.

Enter State: the name of the animator state that must be used when this ability enters the system.

Transition Duration: time that system uses to make transition between the current animation and the desired animation. Higher values results in smoother transitions.

Finish On Animation End: when this parameter is checked, this ability will exits the system when the animation finishes.

Use Root Motion: tells the system if that ability should move with animation root motion. If you checked this parameter, some options will be enabled:

Use Vertical Root Motion: check this box if you want to update Y position with animation root motion (not recommended for the most of abilities).

Use Rotation Root Motion: check this box to allow system update character rotation with animation root motion.

Root motion multiplier: used to multiply the built in root motion from animation.

Allow Camera Zoom: check this box to allow player zoom camera when this ability is running.

Custom Camera: checking this box you can choose a different camera data for that ability.

Abilities to ignore: use this section to add abilities that must be ignored to allow this ability to run. To understand how it works, watch this video: https://www.youtube.com/watch?v=25rsKIb4sKg

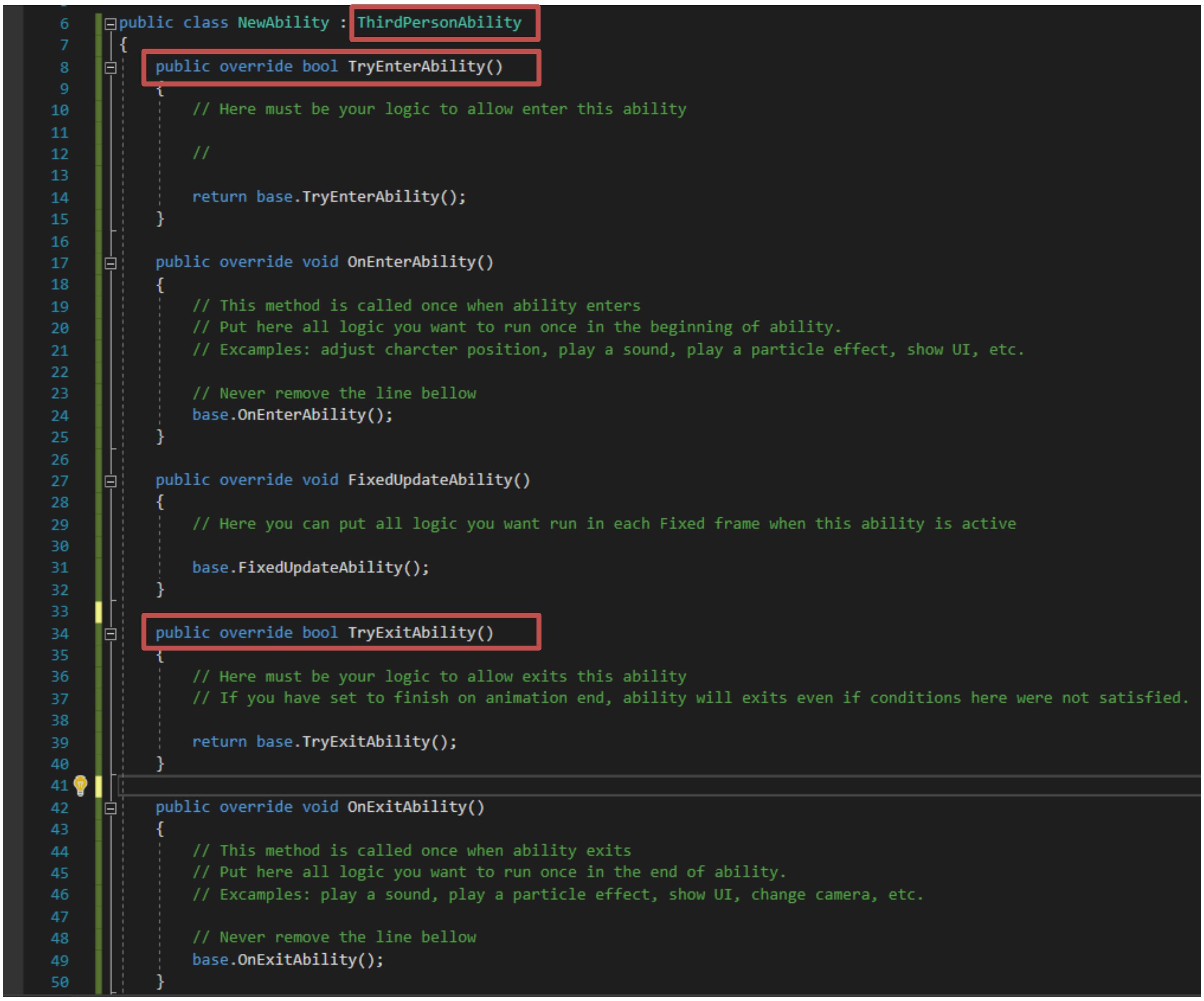
Events: any ability has two events, when ability enters the system and when ability exits. Here you can set events like: play a sound, play a particle effect, send event and anything you want.

5. CREATING NEW ABILITIES

This video explains how to create new abilities using the system:
https://www.youtube.com/watch?v=E_0Xe4JFECQ

All ability must be inherited from `DiasGames.ThirdPersonSystem.ThirdPersonAbility.`

All new abilities must start with this:



```
public class NewAbility : ThirdPersonAbility
{
    public override bool TryEnterAbility()
    {
        // Here must be your logic to allow enter this ability

        //

        return base.TryEnterAbility();
    }

    public override void OnEnterAbility()
    {
        // This method is called once when ability enters
        // Put here all logic you want to run once in the beginning of ability.
        // Excamples: adjust charcter position, play a sound, play a particle effect, show UI, etc.

        // Never remove the line bellow
        base.OnEnterAbility();
    }

    public override void FixedUpdateAbility()
    {
        // Here you can put all logic you want run in each Fixed frame when this ability is active

        base.FixedUpdateAbility();
    }

    public override bool TryExitAbility()
    {
        // Here must be your logic to allow exits this ability
        // If you have set to finish on animation end, ability will exits even if conditions here were not satisfied.

        return base.TryExitAbility();
    }

    public override void OnExitAbility()
    {
        // This method is called once when ability exits
        // Put here all logic you want to run once in the end of ability.
        // Excamples: play a sound, play a particle effect, show UI, change camera, etc.

        // Never remove the line bellow
        base.OnExitAbility();
    }
```

- **TryEnterAbility()**

This method must return true if you want that ability enters. Everytime this method returns true, a request is sent to main controller to check if this ability can enter.

All ability has a list called IgnoreAbilities. If you want this ability to stop some ability running, as Jump, Fall, or Roll for example, you have to add these abilities to the list.

After send a request to enter, controller will check what ability is running, if this ability is in the IgnoreAbilities list of the new ability, controller will stop the ability running and starts this new ability. If the ability running is not in the list, controller will ignore the new ability request. Keep it in mind when you create a new ability.

- **TryExitAbility()**

Works with the same concept of TryEnterAbility. Return true if you want stop this ability. It will not check any ability running or requesting to enter. It will simply stop the current ability. If you check Finish On Animation End, ability will be stopped at the end of the animation.

All abilities has two UnityEvent: OnEnterAbilityEvent and OnExitAbilityEvent. These two events can be useful when you want to execute simple functions. You can just add this function in the Inspector of the ability. They are called once.

6. EVENTS AND OBJECT POOLING

To learn how to use Events and Object Pooling in Third Person System, watch the videos below:

Object Pooling: https://youtu.be/5N91zdP4ras

Global Events: https://www.youtube.com/watch?v=W_loTWWVWlc&t

7. SUPPORT

To receive some support or ask for help, send an e-mail to:
rodrigoaadias@hotmail.com

8. InControl INTEGRATION

Send your **invoice number** to support e-mail to get InControl integration script:
rodrigoaadias@hotmail.com