Documentation

Overview

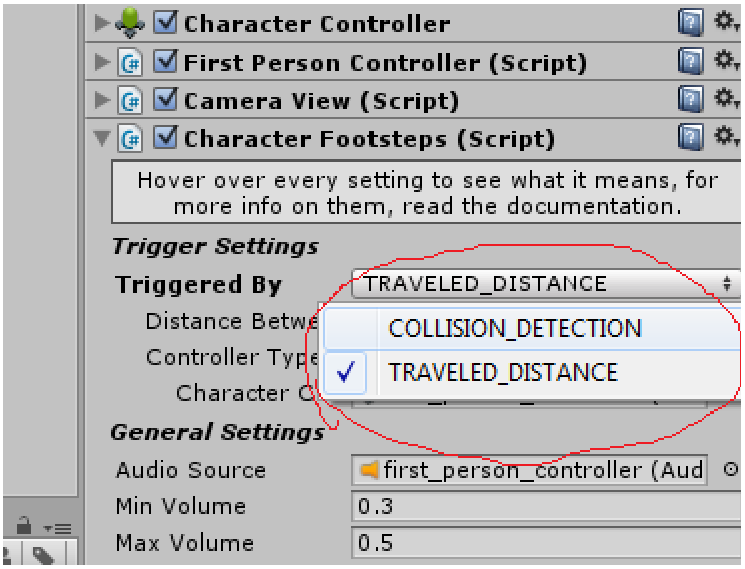
The "FREE Footsteps System" asset is a system developed by me for a game that is going to be shipped by February 2016 called "Wild Wrestle", it features two different modes of playing footsteps and a solid way to manage surfaces in the game.

* The system uses **texture** information to decide what sound to play.

* Works by detecting the **collider** below the character, then **passes** the collider and the hit point to the **'SurfaceManager'** script, which finds what texture is at that specific point, works with both Terrain and normal colliders, the only thing you need to worry about is a few simple steps to follow, in order to set up the system.

* There are two available play modes:

> - "**COLLISION_DETECTION**" - you need to have a trigger for each foot, and as the animations play, you will have the triggers detecting the moment the feet touching the ground and as such playing a footstep sound.
>
> -"**TRAVELED_DISTANCE**" - for this method to work, you need to provide either a **Rigidbody** component or a **CharacterController** component, so the system can record their velocity and calculates how much the character moves, when the character moves for example : 1.8 units / meters, the system will play a footstep based on the surface below.



☑ Character Controller
☑ First Person Controller (Script)
☑ Camera View (Script)
☑ Character Footsteps (Script)

Hover over every setting to see what it means, for more info on them, read the documentation.

Trigger Settings

Triggered By: TRAVELED_DISTANCE

COLLISION_DETECTION
✓ TRAVELED_DISTANCE

Distance Betw
Controller Typ
Character C

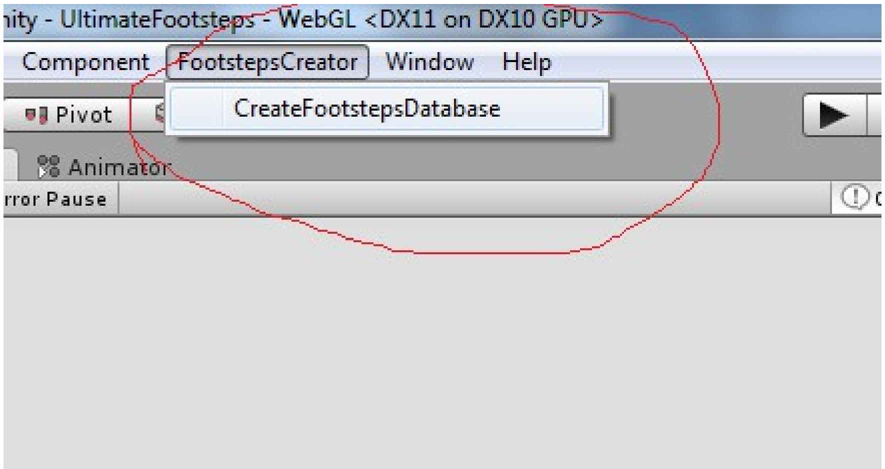
General Settings

Audio Source: first_person_controller (Aud
Min Volume: 0.3
Max Volume: 0.5

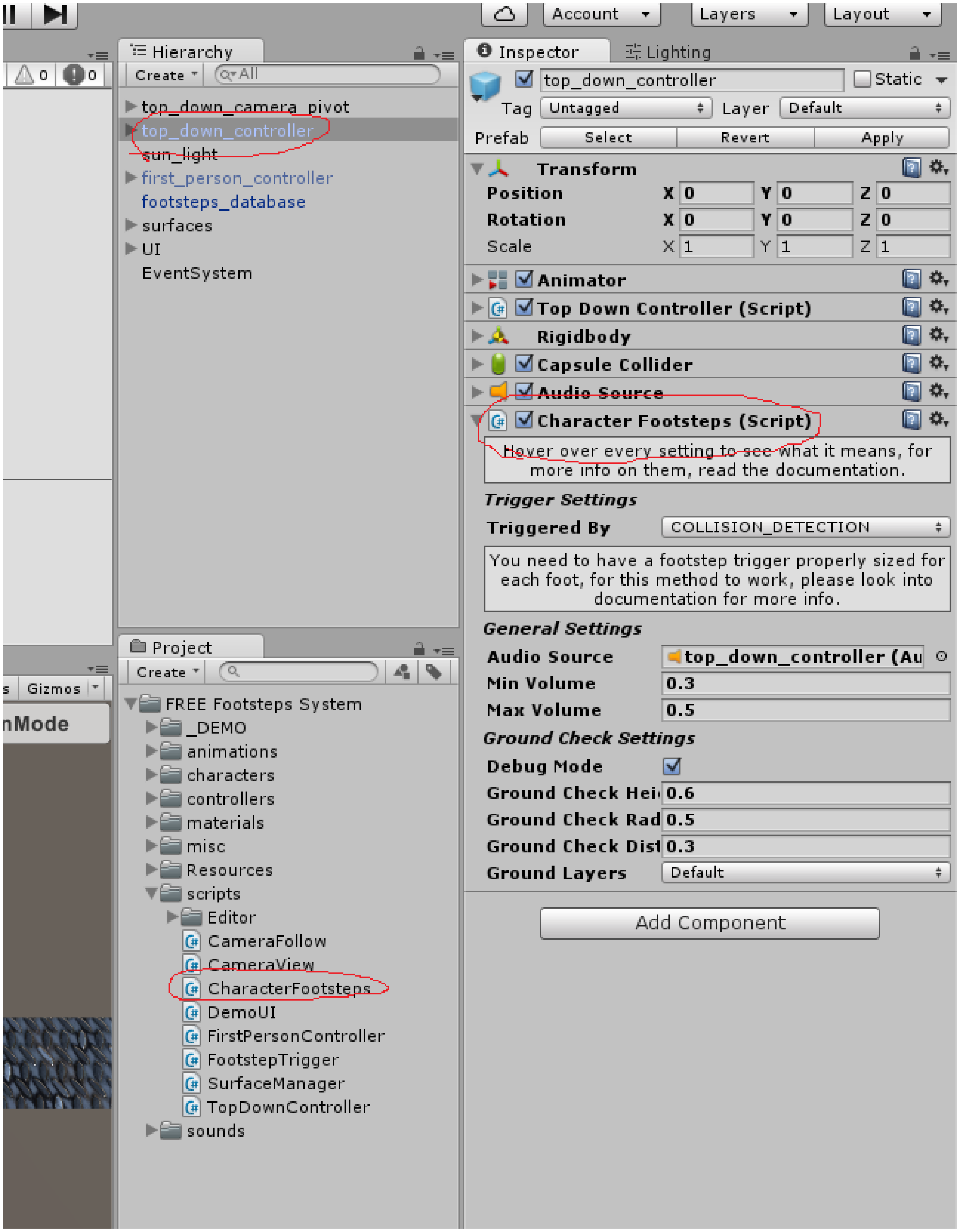
How to setup the system

* Open the scene you want to setup the system in.

* You need to create the game object that will hold all the links between textures and footstep sounds, in order to do that, click "FootstepsCreator/CreateFootstepsDatabase", this will create a new game object called "footsteps_database" in your scene, with some predefined data.



* Now supossing you have some character in the scene, add a script called "CharacterFootsteps" on it.



☑ top_down_controller □ Static

Tag: Untagged   Layer: Default

Prefab: Select | Revert | Apply

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ Animator

☑ Top Down Controller (Script)

Rigidbody

☑ Capsule Collider

☑ Audio Source

☑ Character Footsteps (Script)

Hover over every setting to see what it means, for more info on them, read the documentation.

Trigger Settings

Triggered By: COLLISION_DETECTION

You need to have a footstep trigger properly sized for each foot, for this method to work, please look into documentation for more info.

General Settings

Audio Source	top_down_controller (Au
Min Volume	0.3
Max Volume	0.5

Ground Check Settings

Debug Mode	☑
Ground Check Hei	0.6
Ground Check Rad	0.5
Ground Check Dist	0.3
Ground Layers	Default

Add Component

* Explaining the "**General Settings**" and the "**Ground Check Settings**":

- "Audio Source" - The system will use this audio source to play the footstep sounds, so create one and drag it in this slot, I placed it on the main root for my TopDown character, but I recommend creating an empty GameObject with an AudioSource attached and place it at the feet position, so it's more realistic.

- "Min Volume" and "Max Volume"- When the footstep clip it's played, it's played at a random volume between these values.

- "Debug Mode" - when using this mode, with your character selected, you can see a white sphere and a red line in the scene, which represents the radius of the ground checking and how far it will check - THE SYSTEM SEARCHES FOR THE CHARACTER BEING GROUNDED, so its not playing sounds when the character is not grounded, this feature its also used to play the landing sound.

- "Ground Check Height" - How far up (based on the character's pivot), the ground check will start.

- "Ground Check Radius" - Try to keep this value the same as the radius of your character's collider

- "Ground Check Distance" - Basically if the character is in free fall, by the time its at this distance relative to the ground, the system considers the character grounded.(0.15 - 0.3 works the best)

- "Ground Layers" - By default this is set to "Nothing", which means the system will not detect anything, in here you should choose the layers your ground objects are in, if you dont have any specific layers for the ground, just choose "Default" (supposing your objects are in this layer).

* IMPORTANT NOTE ABOUT THE GROUND CHECKING : If you are unsure about how the ground checking works, basically, while in DebugMode, **the white sphere** should be a little bit above the player's feet, and **the red line** should cross just a little bit the ground, this will make sure the ground detection works just fine.

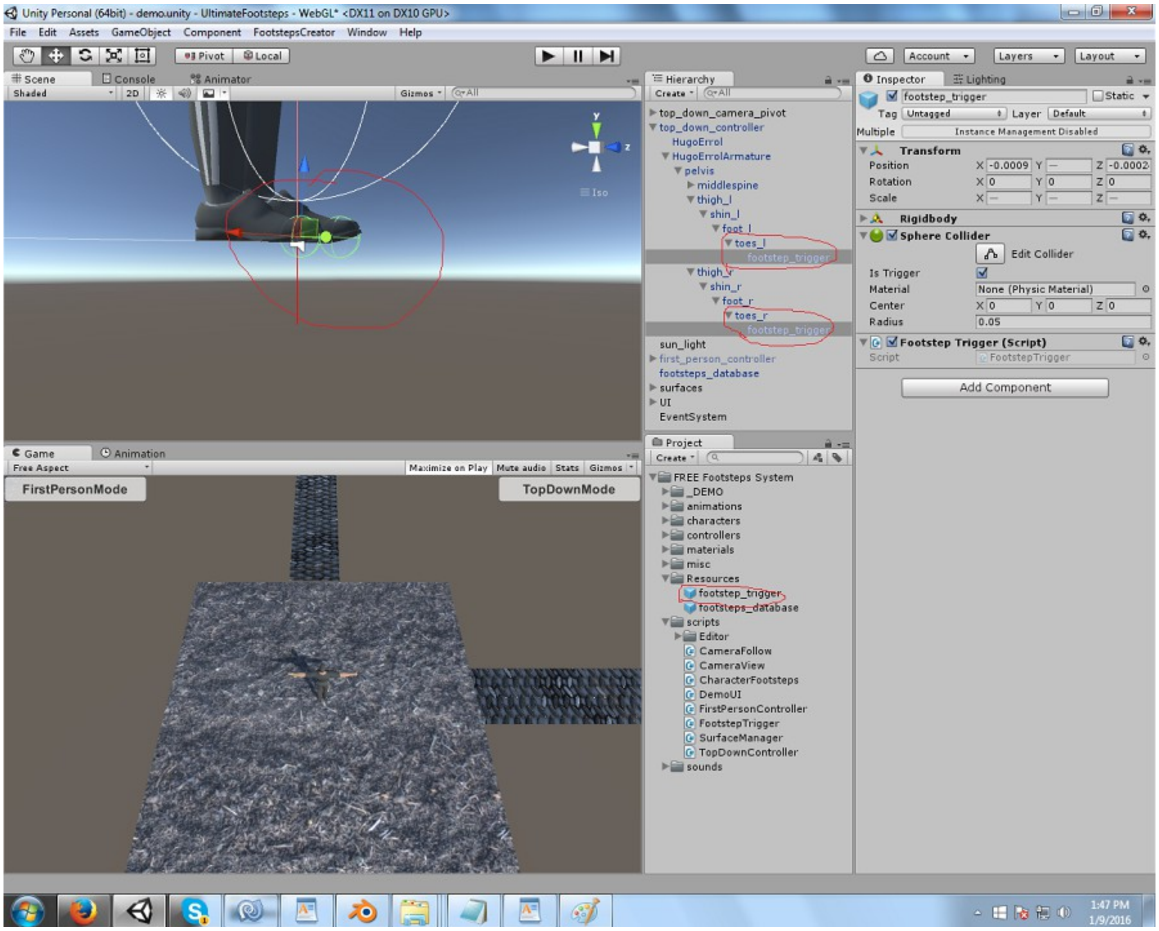
- This is because I use sphere casting to detect the ground, sphere casting its like a thick raycast(its like duplicating the sphere and checking if something is in it's radius)

* **How to setup your character with the "COLLISION_DETECTION" method: (suitable for animated characters):**

- You need to have two triggers attached to the feet, I have already setup a helper object for making this process easier, you can find it at :

"FREE Footsteps System/Resources/footstep_trigger"

- Now find the feet bones, drag the "footstep_trigger" object on them, and tweak their positions and the collider settings to look like in the image below:



footstep_trigger □ Static
Tag Untagged Layer Default
Multiple Instance Management Disabled
Transform
Position X -0.0009 Y — Z -0.0002
Rotation X 0 Y 0 Z 0
Scale X — Y — Z —
Rigidbody
Sphere Collider
Is Trigger ☑
Material None (Physic Material)
Center X 0 Y 0 Z 0
Radius 0.05
Footstep Trigger (Script)
Script FootstepTrigger

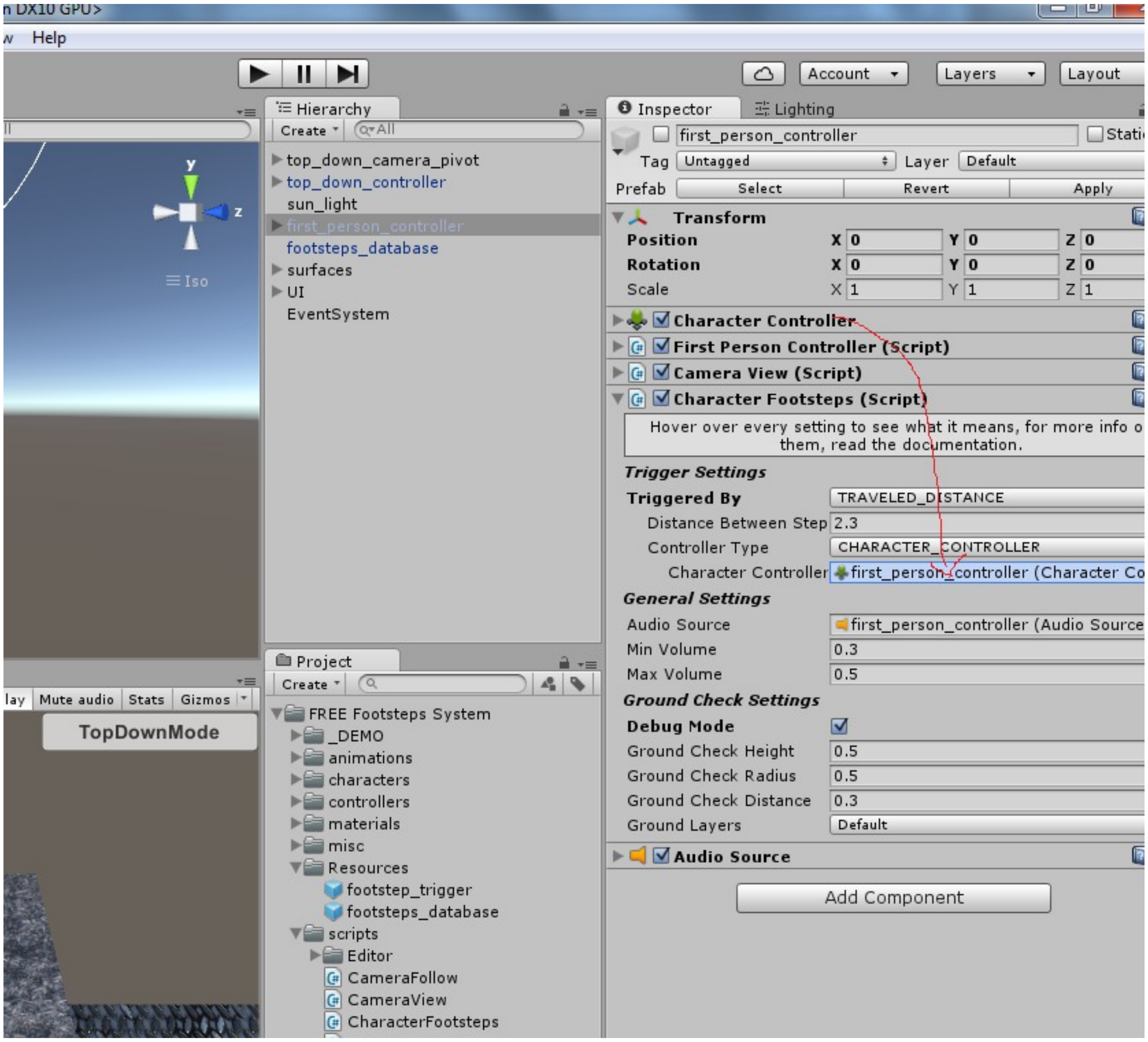
* hit PLAY, and tweak the triggers position until the footsteps sounds right

* **How to setup your character with the "TRAVELED_DISTANCE" method: (suitable for FIRST PERSON CHARACTERS:**

- The first new option that appears when you select this method is "**Distance Between Steps**", which tells the system at what distance interval to play another footstep clip, so after the character moves for example 1.5 units, the system will play the sound.

- The second option is the "**Controller Type**", you can select between Rigidbody and CharacterController, based on what method you use to move your character.

- So *in the image below* I have a First Person Character that I quickly built and it uses a "**CharacterController**" to move, so I just drag the CharacterController component in this slot:



□ first_person_controller

Tag Untagged Layer Default

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ Character Controller

☑ First Person Controller (Script)

☑ Camera View (Script)

☑ Character Footsteps (Script)

Hover over every setting to see what it means, for more info o them, read the documentation.

Trigger Settings

Triggered By	TRAVELED_DISTANCE
Distance Between Step	2.3
Controller Type	CHARACTER_CONTROLLER
Character Controller	first_person_controller (Character Co

General Settings

Audio Source	first_person_controller (Audio Source
Min Volume	0.3
Max Volume	0.5

Ground Check Settings

Debug Mode	☑
Ground Check Height	0.5
Ground Check Radius	0.5
Ground Check Distance	0.3
Ground Layers	Default

☑ Audio Source

Learn more

* After you have imported the system, you can learn a lot by analyzing the "demo" scene, located at "FREE Footsteps System/_DEMO/demo.scene", in there you can find two examples in which I use the two methods described above.

* You can always send me an e-mail with your question at : cristianpavel40@yahoo.com

* Add me on Skype : cristian.pavel40

* Add me on YahooMessenger : cristianpavel40

My name is Pavel Cristian, I am the author of this system and I hope that it becomes a useful tool for your projects,

- You can check my other products on Unity Asset Store, I plan on making a few more useful FREE assets in the future.

- Or visit my website : http://brushinteractive.tk/forum/search.php