Footsteps Scene Designer Pro: Users Guide

(version 2.0)

Table of Contents

Footsteps Scene Designer Pro: Users Guide

(version 2.0)

Description

Footsteps Scene Designer Pro allows you to quickly and easily assign distinct footstep sounds to textures and objects in your Unity scene that are played when players and mobs walk over them.

It comes with 11 predefined material sound types and 55 free sounds to get you up and running right away.

Fully documented source code is included allowing you to add additional sound types to suit your needs.

***Footsteps Scene Designer Pro* uses the character controller included in the Standard Assets package available freely from Unity.**

Footsteps Scene Designer Pro consists of the following three scripts:

1. Footsteps.cs
2. TerrainSound.cs
3. DetectWaterCollision.cs

There are also two editor scripts that help keep the user interface neatly organized. You should not have to touch these unless you plan to make code changes to the corresponding script above:

1. FootstepsScriptEditor
2. TerrainSoundScriptEditor

Footsteps Scene Designer Pro: Users Guide

(version 2.0)

Setup

Footsteps.cs

Plays appropriate sound effect when the First Person Controller or Third Person Controller walks over appropriately tagged ground or wades in water. Works in both indoor scenes and outdoor terrains.

How To Use:

1. Add to the FPSController (or ThirdPersonController) game object of your scene where footstep sounds are desired
2. Create the following tags in your scene:
    - BrokenGlass
    - Concrete
    - Dirt
    - Grass
    - Gravel
    - Leaves
    - Puddle
    - Sand
    - Snow
    - Water
    - Wood
3. Add a collider to gameObjects that you would like to produce a sound.
4. Tag gameObjects that you would like to produce a sound with one of the tags created above
5. For outside terrain footstep evaluation:
    - Add the "TerrainSound" script to the terrain object used in the scene
    - With the terrain selected assign a sound, via the inspector, to each texture using the "TerrainSound" visual interface.

6. For water collision detection make sure:
    - The following script has been added to the water plane you want to test for collisions on:
        - detectWaterCollisions.cs
    - The water plane Layer is set to "Water"

NOTE: Some built-in sample sounds have been provided. If you would like to use custom sounds drag the appropriate sound files into the corresponding array (i.e. walkingOnBrokenGlass, walkingInWater, walkingOnWood, etc.) and set "Only Use Preloaded" to "false" in the inspector.

ALSO NOTE: If You are using the standard First Person Controller provided by Unity make sure to comment out its footstep code so it doesn't override the behavior this script provides.
To do this:

1. Go to the file entitled:

   /Assets/Standard Assets/Characters/FirstPersonCharacter/Scripts/FirstPersonController.cs

2. Search for:

```
PlayLandingSound();
PlayJumpSound();
PlayFootStepAudio();
```

   And change it to:

```
// PlayLandingSound();
// PlayJumpSound();
// PlayFootStepAudio();
```

DetectWaterCollision.cs

Allows footsteps.cs script to detect entering and exiting water.

How To Use:

Add to any water prefab in your scene

TerrainSound.cs

Allows you to visually assign Footstep Sound Types to Terrain Textures.

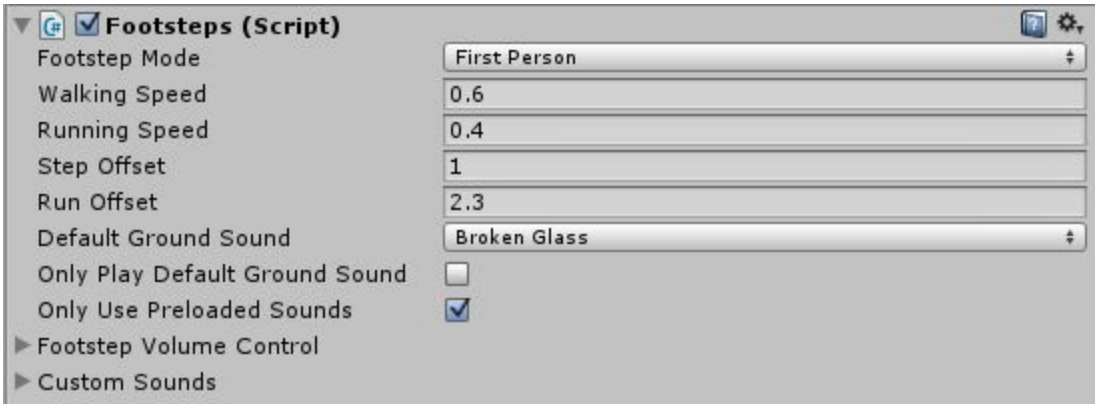
How To Use:

Add to the terrain object in your scene.

Configuration

FootstepsScript.cs

Once attached to your game controller the footsteps script can be configured in several ways through the inspector.



Note: Not all properties are available in every *Footstep Mode*.

Footstep Mode

First Person

Select this option if you are using the FPSController.

Third Person Simple

Select this option (or Third Person Advanced) if you are using Unity's ThirdPersonController. *Third Person Simple* mode is a quick and easy way to add footstep detection to your character. It provides similar configuration options as those provided in *First Person* mode and is good for applications where footstep sound and animation synchronicity is not important.

Third Person Advanced

Select this option (or Third Person Simple) if you are using Unity's ThirdPersonController. *Third Person Advanced* mode uses animation events to trigger surface analysis and footstep sound generation. This allows footstep sounds to be generated based on the character's animation and foot placement, which is far more accurate and realistic.

It is the preferred implementation for those looking for a more polished result but requires manually setting animation events that correspond with "stepping" actions. If you are

uncomfortable doing this, a sample, preconfigured *Third Person Controller* has been included for use in your projects and can be found in:

/Assets/Footsteps Scene Designer Pro/ThirdPersonController/FSDP-ThirdPersonController

Should you need to modify Unity's Third Person Controller or just want to learn how this is done, see the section entitled: Adding "Step" Events to Unity's Third Person Controller

Walking Speed

The time delay (in seconds) between footstep sounds while walking.

Running Speed

The time delay (in seconds) between footstep sounds while running.
NOTE: Running is currently tied to <Shift>+[Movement Key].

Step Offset

Allows you to adjust the distance from the player where step detection occurs when walking. By default the surface 1 meter in front of the player is tested when trying to determine which footstep sound to play.

Run Offset

Allows you to adjust the distance from the player where step detection occurs when running. By default the surface 2.3 meter in front of the player is tested when trying to determine which footstep sound to play.

Default Ground Sound

The sound to play when the detected surface is unknown.

Only Play Default Ground Sound

When selected plays the 'Default Ground Sound' for all footsteps.

Only Use Preloaded Sounds

When selected uses the preloaded sounds, even if you have added custom sounds.

Footstep Volume Control

Allows you to set the footstep volume for a given terrain/texture type (0 = min, 1 = max).

Custom Sounds

Allows you to set custom sounds to be played for each material type. This Overrides preloaded sounds for the corresponding sound type.

DetectWaterCollission.cs

Add to water planes you want to detect collisions with. When a collision occurs the “water” wade footstep sound will be played. Make sure your water object is also tagged “Water” for this to work properly.

Water Depth

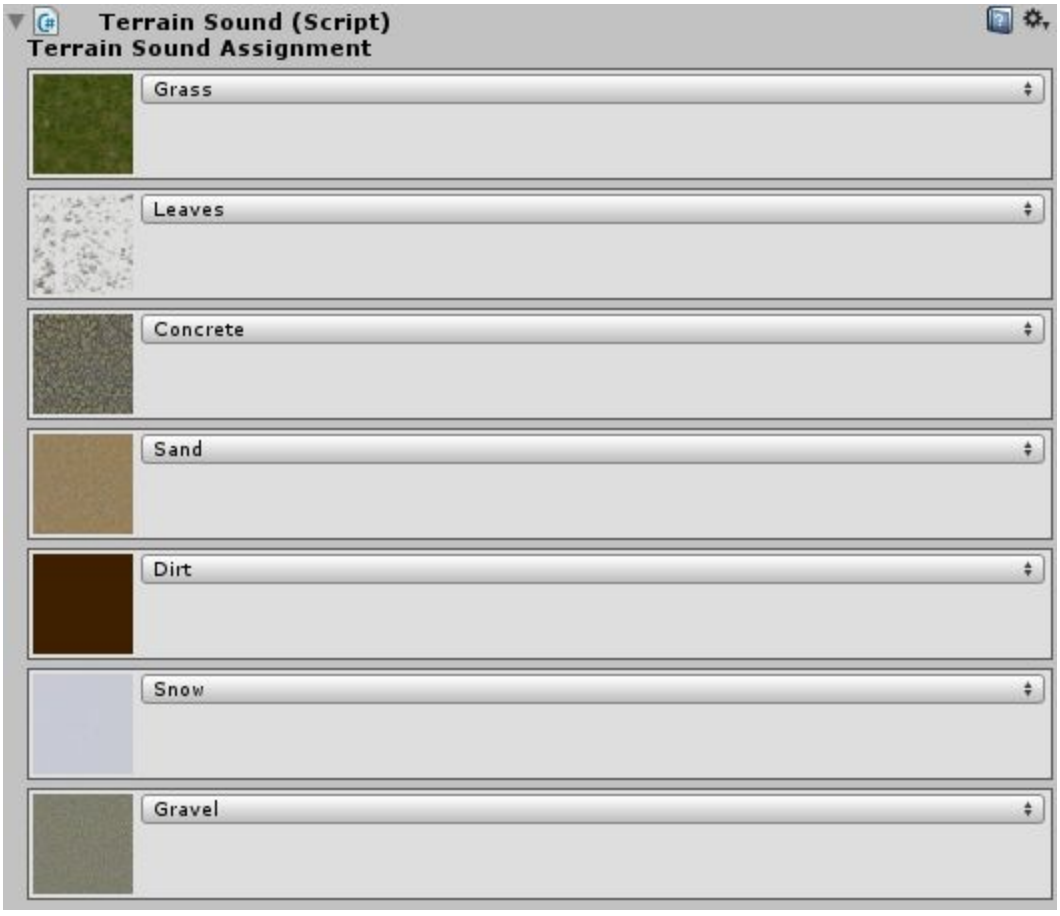
Allows you to define the depth of your water. At runtime a box collider will be added to the water plane that the DetectWaterCollision.cs script is attached to. This plane will extend downward “Water Depth” meters. This defines the area where the “water” wade sound will be played.

TerrainSound.cs

Once attached to your terrain the Terrain Sound script can be used to quickly and visually assign Footstep Sound Types to textures.

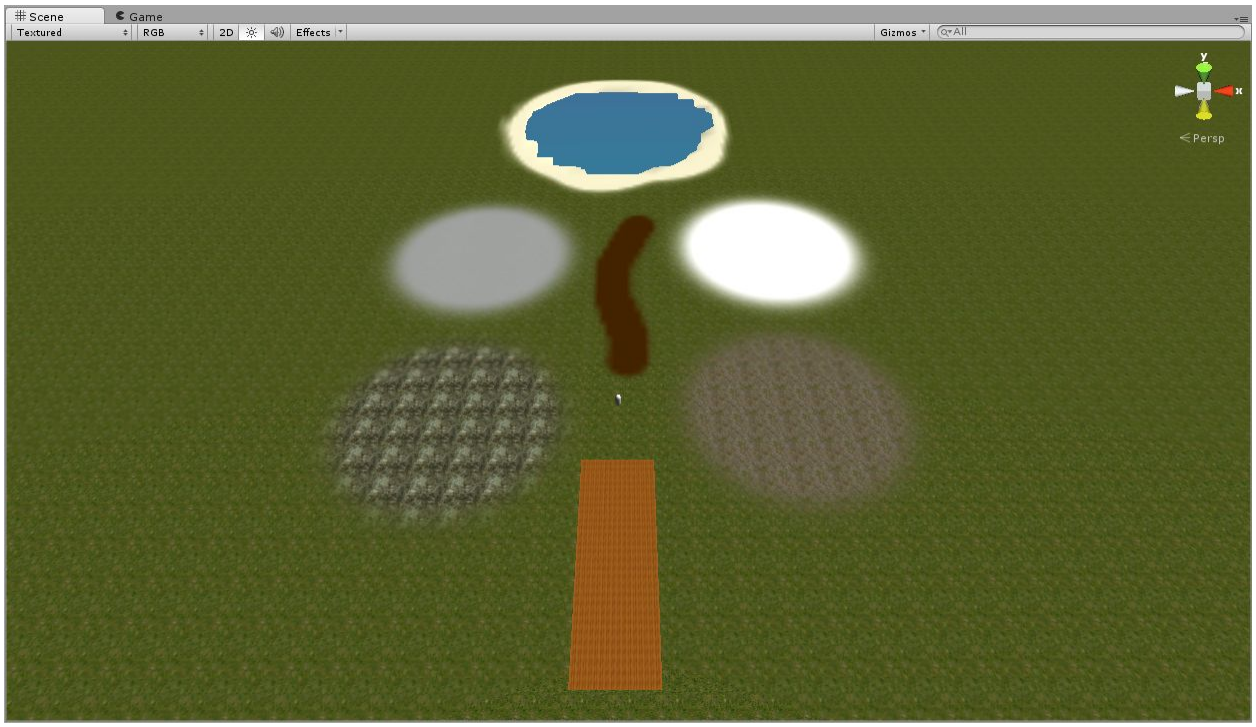
The following is a list of currently supported sound types but can easily be extended:

BrokenGlass, Concrete, Dirt, Grass, Gravel, Leaves, Puddle, Sand, Snow, Water, Wood



Demo Scenes

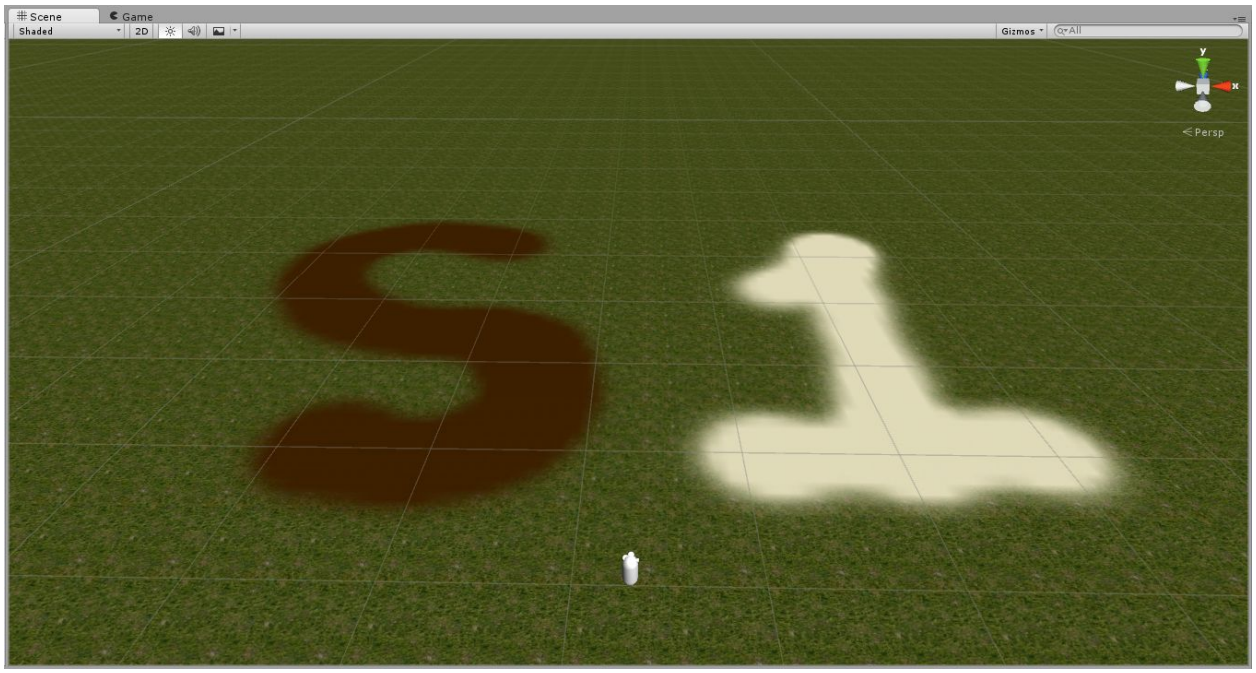
FootstepsDemo-FirstPersonController



This preconfigured scene allows you to examine and run a fully configured project. In it you can experience walking on terrains, objects and in water and hearing the sounds they produce.

You can also experiment with the various settings and customizations to see how they alter the behaviour of the scripts.

FootstepsDemo-Scene1



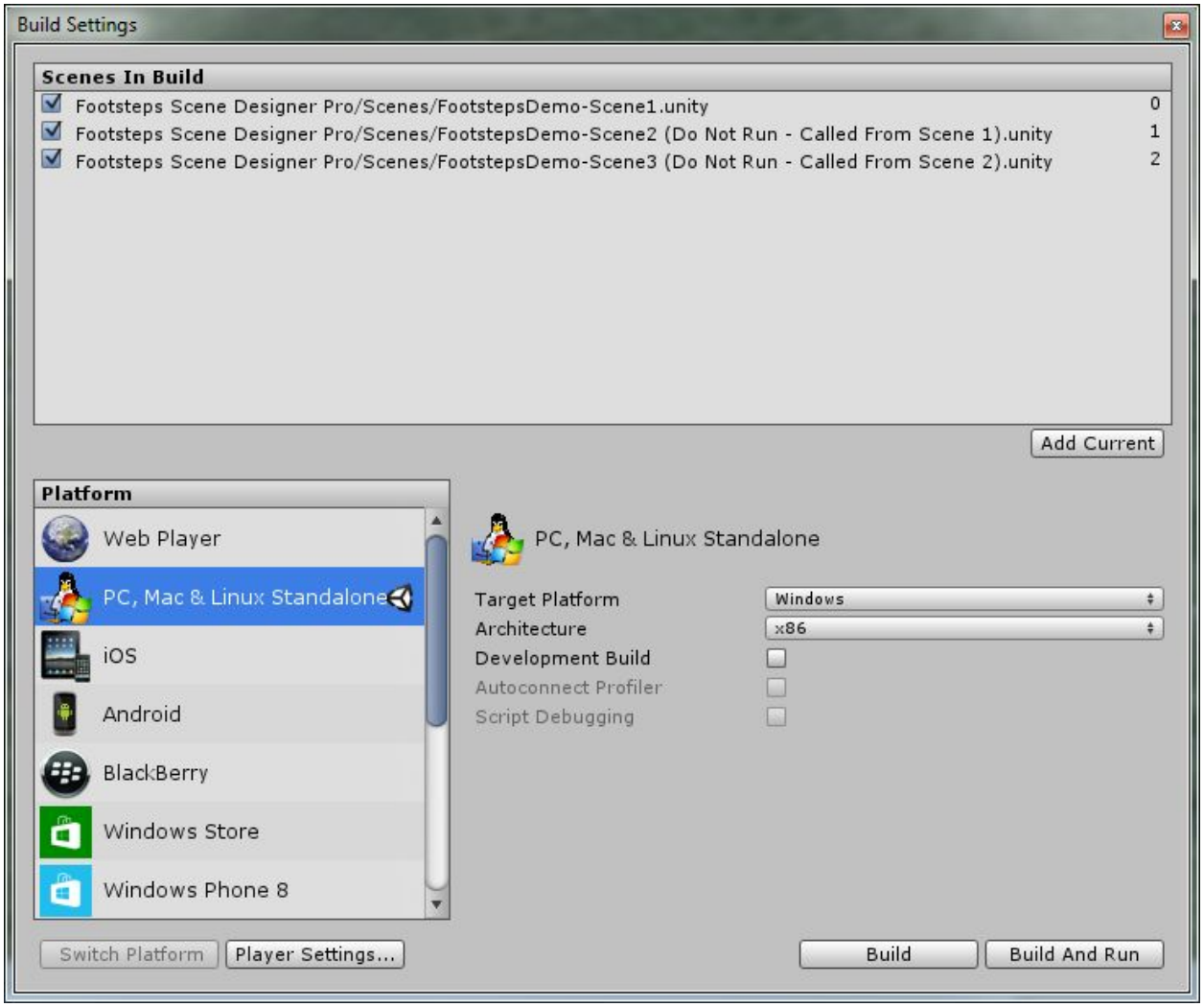
This scene demonstrates how easy it is to **setup and customize the sounds for multiple scenes** in Footsteps Scene Designer Pro. There are three textures in this scene (Grass, Dirt and Sand). Walking on them will produce their corresponding sound.

The brown cube is a transporter. Touching it will load FootstepsDemo-Scene2. A new scene with different terrain textures and sound associations.

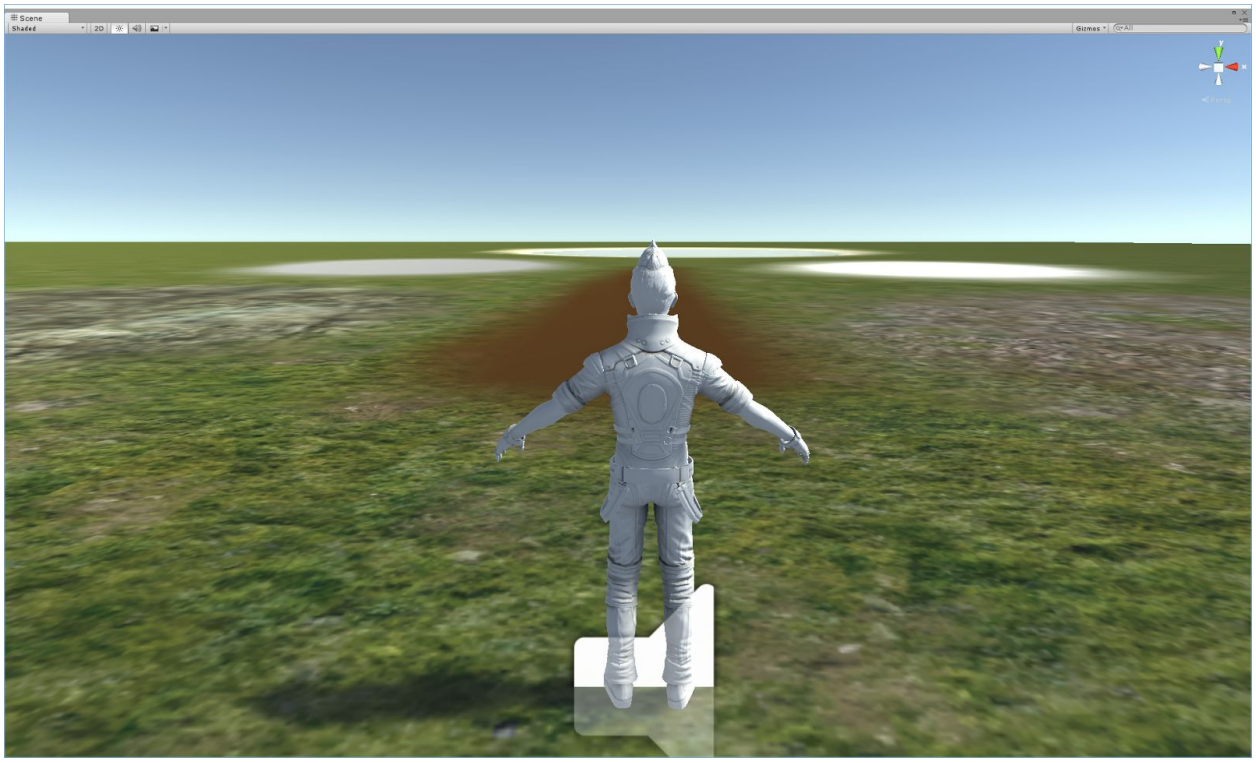
If you select a terrain in the editor for any of these scenes (FootstepsDemo-Scene1, FootstepsDemo-Scene2, and FootstepsDemo-Scene3) you can quickly change the sound associations for the textures and reload FootstepsDemo-Scene1 to test the changes.

NOTE: FootstepsDemo-Scene1 is the master scene. Running it is the only way to access FootstepsDemo-Scene2 and FootstepsDemo-Scene3, which is done by touching the blue transporter cube in each scene.

NOTE: You must add the three scenes to your build settings (File->Build Settings) in order for the "transporter" to work.



FootstepsDemo-ThirdPersonControllerSimple



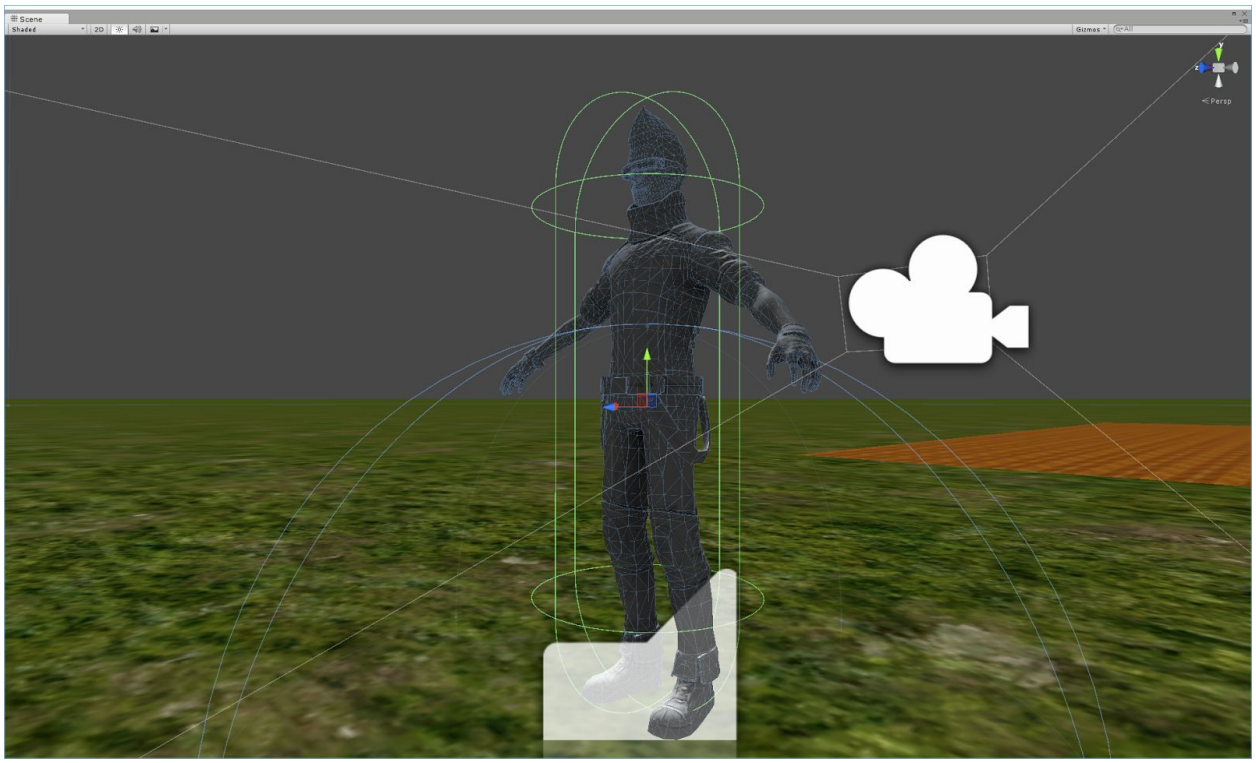
This scene demonstrates footsteps applied to Unity's ThirdPersonController using "Third Person Simple" mode.

Simple mode is easy to setup, but not as accurate as "Third Person Advanced" mode.

Here the "Walking Speed" and "Running Speed" parameters are used to adjust the footstep audio delay.

"Step Offset" and "Run Offset" are used to adjust how far ahead of the player the ground should be sampled in order to determine the next footstep sound to be played. This is generally adjusted to match the stride length of the avatar when walking/running.

FootstepsDemo-ThirdPersonControllerAdvanced



This demo scene uses Third Person Advanced mode for footstep hit detection and sound reproduction.

In this mode, audible footstep sound effects are synchronized with the character's visual animation. In addition, hit detection occurs directly under the foot that is stepping, making it more accurate than Third Person Simple mode.

It requires a bit more setup than the other modes but produces the most accurate and compelling results.

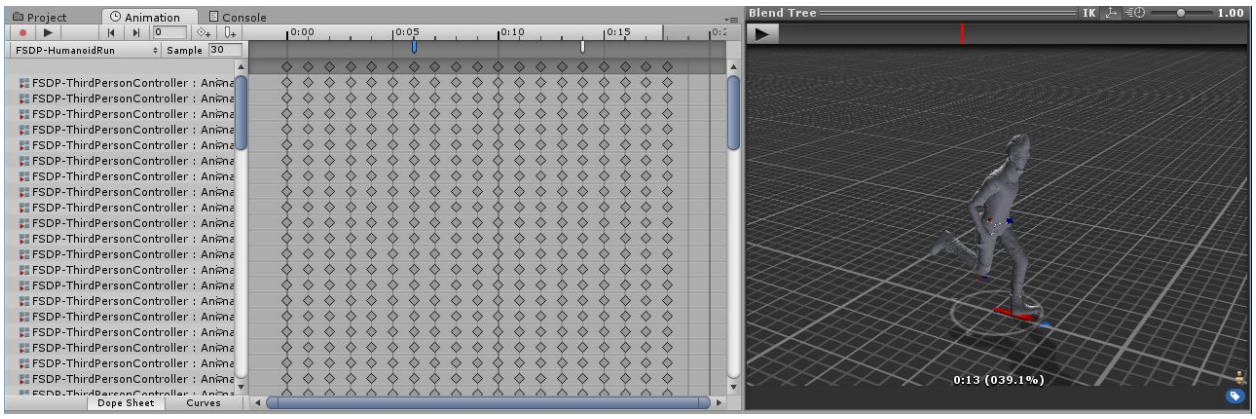
A modified version of the Third Person Controller has already been pre-configured and is used in this scene so that you may examine it more closely.

Troubleshooting

1. **No sound is heard when moving.**
   - Make sure you have an audio source attached to your FPS Controller.
   - Make sure the volume of your audio source is set to 1.
   - If You are using the standard First Person Controller provided by Unity make sure to comment out its footstep code so it doesn't override the behavior this script provides. Click here to learn how.
   - Make sure your gameObject has a collider attached to it and it is enabled
   - Make sure your gameObject is tagged with one of the following terrain sounds: BrokenGlass, Concrete, Dirt, Grass, Gravel, Leaves, Puddle, Sand, Snow, Water, Wood.
2. **No sound heard when entering water.**
   - Make sure your FPS Controller has a rigid body attached
   - Make sure your water has the DetectWaterCollision.cs script attached to it
   - Make sure water gameObject has been assigned the Tag “Water”.
3. **I hear a strange sound effect entering and exiting water or on areas bordering water.**
   - Make sure water gameObject has been assigned the Tag “Water”.
4. **I hear the water footstep sound everywhere I walk.**
   - Make sure your tags are set correctly for the objects you are walking on and are not set to “water”.
   - Make sure the boxCollider for any water in your scene is not positioned above the water. If you manually added a boxCollider it’s very possible that this is the case as they will be virtually invisible in your scene and easy to miss.
   - Make sure the “Water Depth” variable isn’t set too high for a water source that appears above where the problem occurs. If it is, you may be accidentally walking into that which will cause the “Water” wade sound to play.
5. **FootstepsDemo-Scene2 and 3 don’t load.**
   - Remember, these scenes can only be reached via “teleportation” from FootstepsDemo-Scene1 (which is the “Master” scene). See this section for more details.
   - Make sure all scenes have been added to your build settings.

Appendix 1

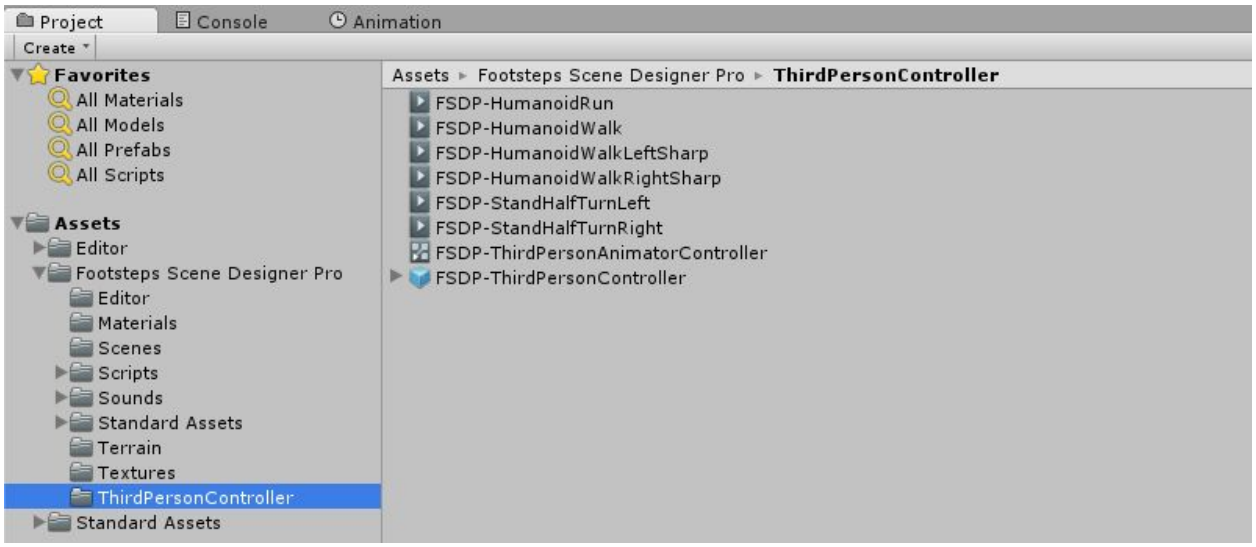
Adding “Step” Events to Unity’s Third Person Controller



Footsteps Scene Designer Pro’s *Third Person Advanced* mode allows you to synchronize animation footstep visuals to their corresponding sound. We also use the left/right foot skeletal transforms to determine the point of impact for more accurate footstep detection.
This is accomplished through animation events.

Generally speaking you will not have to do this as a preconfigured *Third Person Controller* has been included for use in your projects and can be found in:

/Assets/Footsteps Scene Designer Pro/ThirdPersonController/FSDP-ThirdPersonController



If you decide to manually add the animation events you will want to modify the following Third Person Controller Animations:

- HumanoidRun
- HumanoidWalk
- HumanoidWalkLeftSharp
- HumanoidWalkRightSharp
- StandHalfTurnLeft
- StandHalfTurnRight

Within each of these methods you will want to make two calls to the “step()” method (one for each foot). These calls are made via animation events with an Integer value of “0” to indicate left footsteps and “1” to indicate right.

To better understand the process I have created a video to visually illustrate the process.

Third Person Advanced Footstep Mode (Video)