

The Smart Footstep System is a quick & easy solution to adding realistic footstep behavior to your characters.

This document will guide you on how to use it.

1. Smart Footstep System Overview



The Smart Footstep System is very simply to use. I will explain some of the variables as seen above.

Footstep Audio is the AudioSource from which the footstep sounds will be played. Preferably, position this audio source towards your character's feet.

Ground Distance Check is the length of the ray that checks for the ground's texture. This ray will be drawn in runtime to help you figure out an appropriate value.

Ground Types is a List of your surfaces, each element contains a name, a texture and several sounds. When your character is walking over a specific element's texture, it will play sounds found in that element's "sounds" array.

2. Third Person Character

Firstly, go to Scripts/System and add the “SmartFootstepSystem.cs” to your character and make the necessary configurations.

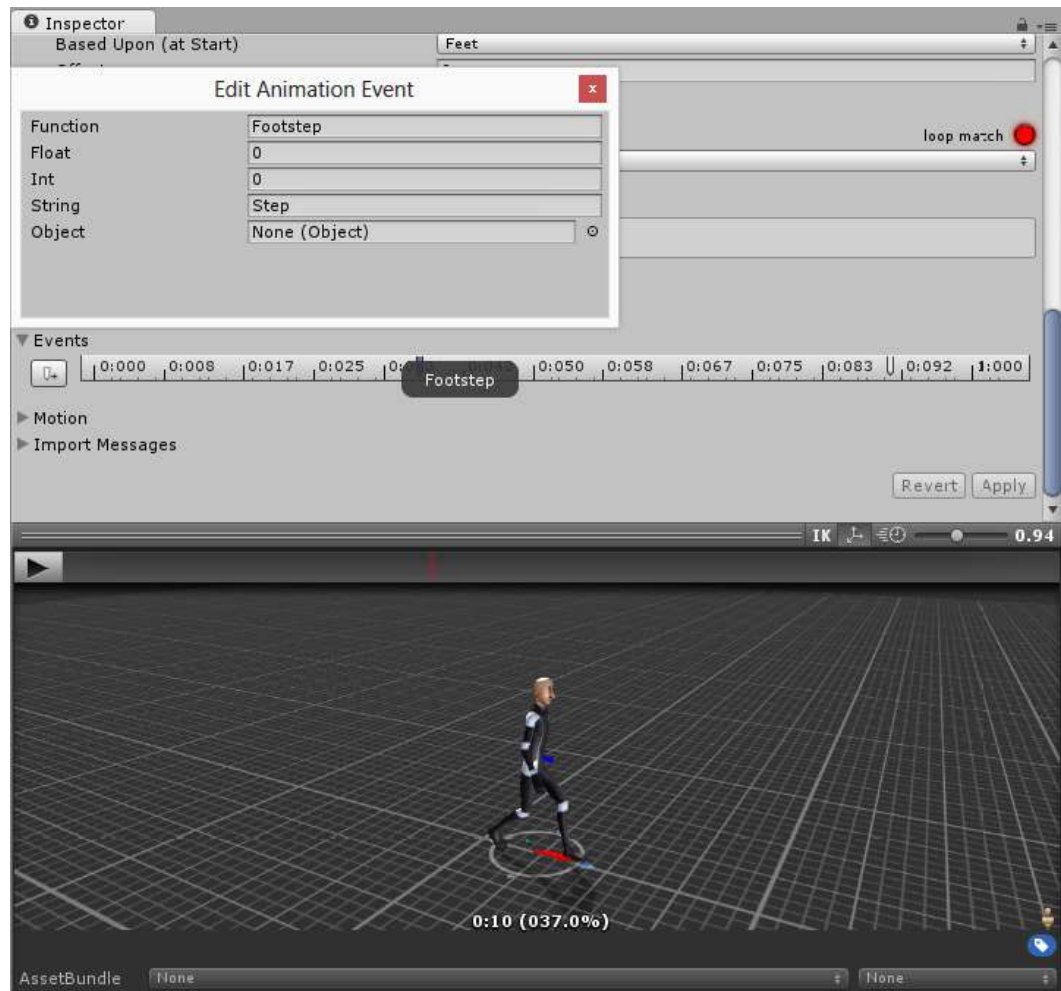
There are 2 ways of setting up the Smart Footstep System for a Third Person Character.

1. Animator Events

Using animator events is the quickest, most efficient and most accurate way of using the Smart Footstep System.

To use animator events on Mechanim Animations :

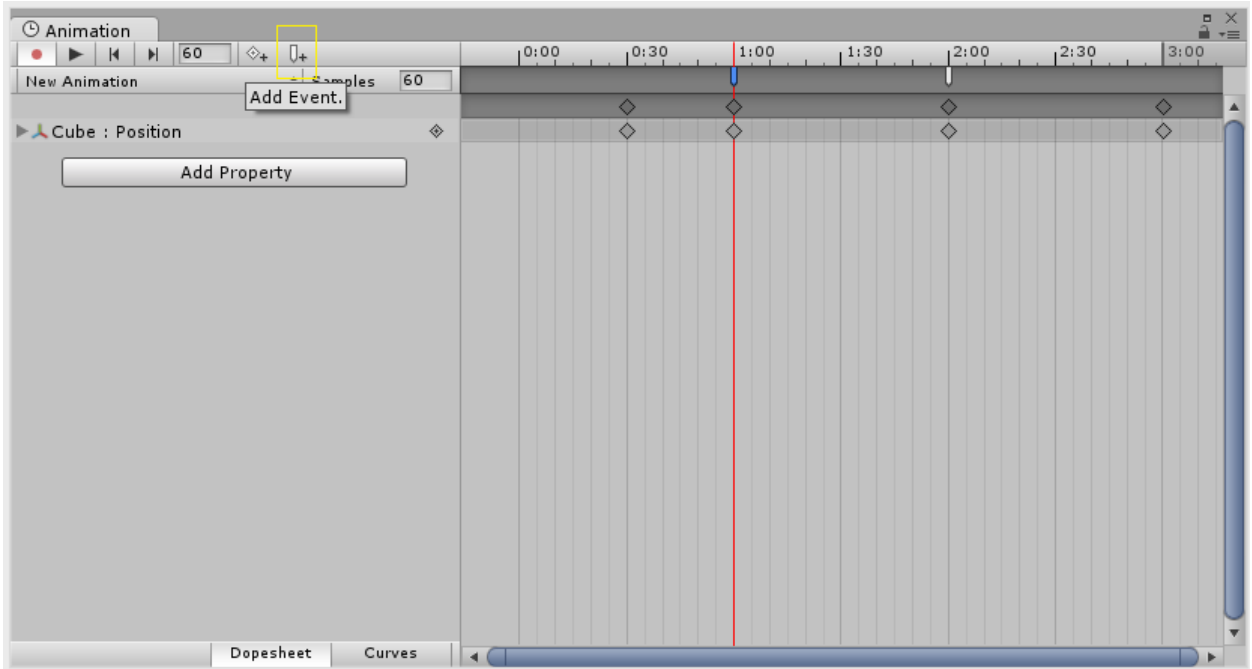
Open your animation and scroll down to Events and add a new event whenever your character takes a step. Set every event’s Function to “Footstep”.



To use animator events on Legacy Animations :

Select your character and press CTRL+6 to open the animation window.

Create a new event whenever your character takes a step as follows :



Double click on the new event and set it's Function to Footstep().

2. Footstep Intervals In Code

This is a bit more complicated but relatively easy. Setting up footstep intervals in code is primarily used in First Person Characters.

Add the following variables to your character's movement code :

```
public float footstepRate;  
private float nextFootstep;
```

footstepRate is how often a footstep sound will play when walking, running etc.

nextFootstep is the interval between footsteps. This doesn't need tweaking because it will be given values during runtime.

Next add the following code in the update method :

Note : you will need to have a bool that is determining whether your character is moving(walking or running)

```
if(isMoving && Time.time > nextFootstep){  
    nextFootstep = Time.time + footstepRate;  
    GetComponent<SmartFootstepSystem  
>().Footstep();  
}
```

You can increase or decrease the *footstepRate* depending on your character's state for more precision.

3. First Person Character

Firstly, go to Scripts/System and add the "SmartFootstepSystem.cs" to your character and make the necessary configurations.

Afterwards, you may follow Step 2 (**Footstep Intervals In Code**) above to set up the Smart Footstep System for your First Person Character.

If you encounter any problems or errors, please reference the demo scenes or contact me directly at ian.izzy94@gmail.com and I will respond promptly.

Thank you for your support!