Game Audio

Tutorial 6 – DIG6041

# Pre-requisites

* Good knowledge of FMOD Studio
* Good understanding of Character Sounds and Foley
* GitHub Account
* Basic GitHub Desktop understanding

# Topics

* Creating dynamic footstep events in FMOD
* Implementing footsteps in Unity (robot 3rd Person Game) tying them to animations
* (optional) Implementing footsteps in Unity (Viking Village) tying them to player speed
* Project definition and planning

# Preparation

* Clone GitHub Repository ([link](https://classroom.github.com/a/0XpwaAqb))
* Make sure you can play the game without errors
* Download “Sounds & Scripts” zip file from Moodle

# Using GitHub in Class

We're going to use GitHub Classroom for today's work so that you can get used to using GitHub in a safe environment (literally nothing you do can ruin anything) and we can apply some of the basic notions we introduced last week.

Every time you do "something" today, we would like you to commit your changes with a small title and brief description (if needed).

By segmenting your work in little pieces, you ensure that every little step of your work is documented, and that (worst case scenario) you can revert to a previous copy of the project without losing too much work.

Your repository should look something like this by the end of the tutorial:

A screenshot of a computer

Description automatically generated with medium confidence

# Footsteps & Foley - FMOD

Design a basic footsteps event in FMOD. For it to work *seamlessly* with the script we’re going to implement later make sure you use the following naming convention:

* Event name: Footsteps
* Parameter Sheet Name: Terrain
* The parameter should be of type *label* and have 4 labels named: Grass, Gravel, Wood, Water.

All of the above can be customised, but you need to be ready to make small amendments to the script if you decide to do so.

Once you’ve set the event up, create 4 different footsteps in the Terrain parameter sheet tab, one per label. You can be as detailed as you want, but remember: the more you refine a sound, the more immersive the experience will be! Things you might want to consider:

* Multi instrument to randomise footstep sounds
* Added foley track to enhance the footstep with other secondary sounds (leather, jackets, armour, etc)
* Nested multi instruments to have different sounds per each foot

# Footsteps & Foley – Implementation in Unity

Before we start playing around with our footsteps we need to prepare our game to be able to accept everything we are going to throw at it.

## Adding Layers to define ground types

Edit -> Project Settings -> Tags and Layers

Here, we need to add 3 new layers (Grass, Gravel, Wood if you have followed the previous instructions). Water, as you can see, is already present.

## Making our character walk on the new layers

Select the PlayerArmature and find the “Third Person Controller (Script)” component. Change the “Ground layers” variable so that all of our new layers are selected.

## Adding the raycasting script.

Add a “New Script” to the PlayerArmature called “PlayerFootsteps”. You can find the script on Moodle.

## Make the PlayerArmature animation editable

The PlayerArmature has a set of animations that make it move. Currently these animations are uneditable. To make them editable we need to create a copy of the interested animations, and re-assign them to the PlayerArmature.

* Go to Assets -> Character -> Animations and duplicate the "StarterAssetsThirdPerson" and call it "New Movement"
* Duplicate also the "Walk\_N" file inside "Locomotion--Walk\_N.anim" and call it "Walk\_New"
* Double Click the New Movement asset you just duplicated (New Movement), double click the orange box called Idle Walk Run Blend, and click on the gray empty grid.
* Drag the Walk\_New file on the right to the inspector in order to replace the existing Walk\_N
* Now select the PlayerArmature, and under "Animator" drag your "New\_Movement" to replace the old "StarterAssetsThirdPerson"

The procedure above, has assigned our “New Movement” animation to PlayerArmature, and made the walking animation (Walk\_New) editable! You can copy and replace *any* of the other animations to make them editable if you wish to sonify other movements.

# Footsteps & Foley – Tying sounds to animation keyframes

Open the animation pane (CMD + 6 or Window -> Animation -> Animation) and select our Walk New from the menu in the top left.

Between the timeline and the automation point there is a gray area, if we right click there we are able to "Add Animation Event". These events will trigger our sounds.

Create 2 events that match exactly when the foot touches the ground (if you zoom in on the legs, and move the animation playhead around, you should be able to identify the exact point really quickly).

Once you have placed your event, select it, and make sure that on the right hand side, the event has set as a function the "SelectAndPlayFootstep ()" function we declared before in the script.

The footsteps will now play in sync with the animation.

You can put as many events as you want on the animation pane, and you can (with very minor code tweaks) trigger many other sounds if you don’t want your entire foley to be tied to your main footstep trigger.

# Footsteps & Foley – Assigning Floor Types

Now, by simply “assigning” different ground layers to different ground elements, you should be able to hear the sound of the footsteps change.

* Assign a “Grass” type to the main blue ground (you might need to add a mesh collider to this for it to work)
* Assign a “Wood” layer to the orange boxes around the map.
* Assign a “Gravel” layer to the white building in the middle

Walk around the map and hear the different footsteps being triggered.

## Further implementations // Exercises

With the exact same skills learned above, you can go further and deeper in the sound implementation. Try the following tasks (in class or at home) if you want a challenge:

* Implement your personal footsteps
* Add different layers to different parts of the map to have dynamic footstep sounds
* Add a reverb zone under the bridge-like structure that affects the footstep sound (Global Parameter Sheet….!)
* Try and sonify the jump and landing of the character (no code tweaking required)

You will notice that without tweaking the code, the sonification becomes quite dull. By simply tweaking (duplicating what is already there) the code, you should be able to import new FMOD events in the script (one for landing, one for mechanical movements) and triggering them with separate functions that can be called from an animation event just like before… give it a shot!

* Create custom functions for jumping, landing and mechanical foley (code tweaking) and customise the sounds of these specific animations

# Viking Village – Homework

For those of you who wanted to continue their work on the Viking village, I have provided you with the footsteps script that works in a very similar way. This will not be covered in class (as it could be quite redundant) but should be pretty simple to follow from the following steps.

* Create a FMOD event with 4 separate tracks, one per type of ground (Wood, Dirt, Sand, Water)
* Create 4 parameter sheets, 1 per each type of ground (Wood, Dirt, Sand, Water). These can be discrete parameters, with a range of 0-1.
* Automate the volume of each track according to its parameter (Wood will have volume 0dB when the parameter Wood is at 1, Dirt will be 0dB when Dirt parameter is 1, etc..)
* Import the Footsteps script on the FlyingRigidBodyFPSController
* Set the path to your event in the event field (something like “event:/Footsteps” if you don’t have your event in folders

This should attach your designed sounds to the footsteps of your character. Note how due to the fact that we don’t see the feet, we can be more lenient with timing, and we can calculate when to play the next step just by tweaking the speed variable.

You can tweak the speed to have faster or slower footsteps.

The concepts behind the script are very similar to the ones explained in class referring to the script for the Robot 3rd Person Template.