My SCRIPT

// shows it in Unity Inspector

public FMODUnity.EventReference *FootstepsEvent*;

// declaring an event instance (= class) for parameters

FMOD.Studio.EventInstance *footsteps*;

//declaring parameter ID?

FMOD.Studio.PARAMETER.ID *surfaceID*;

Void Start()

{

// plays event at exact location

FMODUnity.RuntimeManager.PlayOneShot(*FootstepsEvent*, transform.position);

//plays attached and follows gameobject / runtimemanager can track and update an event instance

FMODUnity.RuntimeManager.PlayOneShotAttached(*FootstepsEvent*, gameObject);

FMODUnity.RuntimeManager.PlatOneShotAttached(*FootstepsEven*t, GetComponent<transform>(),GetComponent<Rigidbody>());

// creating an EventInstance via RuntimeManager

*footsteps* = FMODUnity.RuntimeManager.CreateInstance (*FoostepsEvent*)

footsteps.start()

footsteps.release()

**for event 🡪 create Instance ?**

**for param 🡪 create description ?**

// **cache a handle for a parameter used in a function/method (update or void type of**

// **function…), so you don’t have to call** // **param by name every update**

FMOD.Studio.EventDescription *surfaceEventDescription;*

// out = the API function will fill in information in this parameter

Footsteps.getDescription (out surfaceEventDescription);

//declaring PARAMETER ?

FMOD.Studio.PARAMETER\_DESCRIPTION surfaceParameterDescription;

surfaceEventDescription.getParameterDescriptionByName(“Surface”, out surfaceParameterDescription);

*surfaceID =* surfaceParameterDescription.id;

}

Void OnDestroy()

{

footsteps.release()

StopAllPlayerEvents.release()

}

Void Update()

{

// manually update instance of 3D event so it has position and velocity of gameobject

footsteps.set3DAttributes(FMODUnity.RuntimeUtils.TO3DAttributes(gameObject, *cachedRigidBody*));

//update a parameter of an instance every frame

Footsteps.setParameterByID(*surfaceID*, (float)surface??);