

Joa_Tutorial_1 Notes

Unity Project/Folder Name: joa_audio_primitives_1

Copyright: 2020 Joe Abbati. All rights reserved. It is a violation of Federal law to copy, duplicate, sell, and/or distribute in any manner, including but not limited to the internet.

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 1/12

a_ :

The default “New Scene”, which contains a Camera Object and Directional Light.

A scene is collection of Unity Game Objects

The Camera and Directional Light parameters can be viewed via the Inspector Tab.

To see the Skybox parameters, go to Unity Menu Window->Rendering->Lighting Settings.

We will be using this “empty” Scene to review the essential Editor functionality.

-> Helpful Links

<https://docs.unity3d.com/Manual/class-GameObject.html>

<https://docs.unity3d.com/Manual/LearningtheInterface.html>

<https://www.youtube.com/watch?v=QUCEcAp3h28>

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 2/12

b_ :

Create a new Game Object, simple 3D Cube...Right-Click in the Hierarchy.

Customize the Skybox by using Create-> Material (Right-Click in the Assets Panel). In the Inspector, set the Material Shader to “Skybox/Procedural” and experiment with the settings.

Examine the Transform Component of the Cube. A GO Transform is what dictates its Position, Rotation and Scale characteristics in the Scene/World.

Get comfortable with all of the Editor GUI for manipulating an Object in Scene View.

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 3/12

c_ :

Directional Light changed from Directional to Point.

Enable “Auto Generate” from the Lighting Settings Panel and then use the Light Inspector in Game View to experiment with variations of the Light Transform (Position/Rotation), Intensity, Color etc...

Notice changes made while in Play Mode will NOT be saved to the GO settings...

*** Note - We will be adding code to move the Camera and GameObject/Cube in the next Scenes.
Use Left-Click on Mouse OR Left-Ctrl to reset transforms**

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 4/12

d_ :

Add a New Script to move the Camera (DriveCameraBasics.cs) - Most notes are in the actual script here.

In addition to latching into the “Input.GetAxis()” elements that are defined in Unity’s File->Build Settings->Player Settings ->Input (see next page)

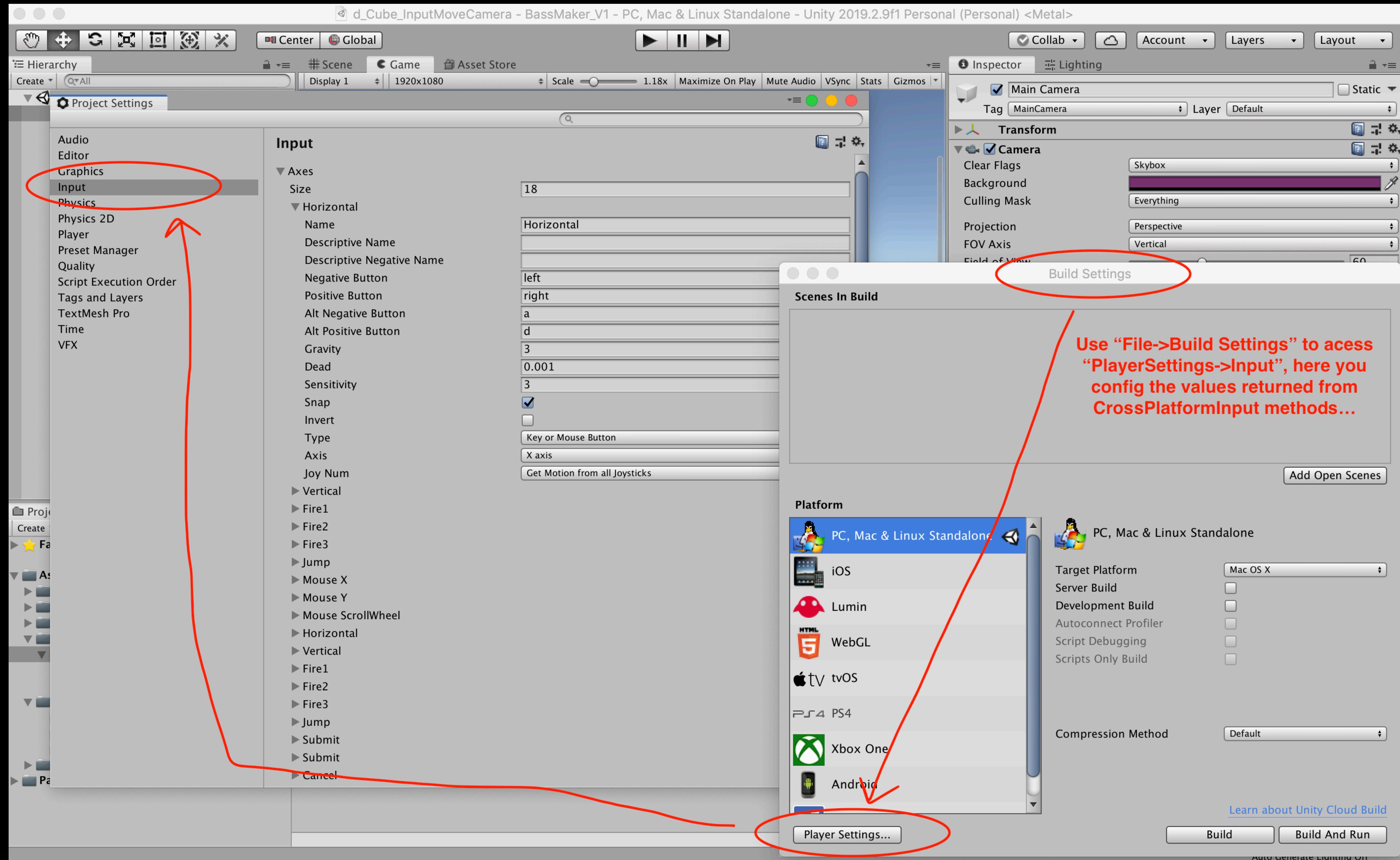
We will also demo direct “Input.GetKeyDown(KeyCode.~)” for controls. This method uses the “[SerializeField] float zChangFactor” to change zoom speed. This is best method to add tunable components to your scripts

View the Camera Transform via Inspector to see your effect on the Transform while in Play Mode.

-> Helpful Link

<https://docs.unity3d.com/ScriptReference/Transform.html>

joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 5/12



joa_audio_primitives_1

Tutorial_1_Pt1 Notes 6/12

e_ :

Add a New Script to move the Cube (ChangeCubeBasics.cs).
Again, most notes are in the actual script here.

This Script has both Transform and Translate Methods demonstrated ->

- .Translate will allow for a time factor method variable.

View the Cube Transform via Inspector and experiment with Variables/Fields of the Script Component.

“Clamping” will only work for Translate Method for now. Also, using a direct/hard Quaternionian value to reset transform.rotation for now.

-> Helpful Links

<https://docs.unity3d.com/ScriptReference/Transform.Translate.html>

<https://answers.unity.com/questions/556664/difference-between-moving-forward-with-transformtr.html>

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 7/12

f_ :

Comparing a GO with a Script Component to a Prefab. Prefabs are like Templates in many ways. You create one by Dragging from the Hierarchy into the Asset Pane. It will turn Blue in color and when you drag the Prefab back into the Hierarchy, you will notice an > symbol next to it, use to Edit the ACTUAL PREFAB. What you see In the Hierarchy is the New Instance of your Prefab

If you change an Instance of the Prefab from the Scene Inspector, you are creating Overrides, which will be shown in **Bold** (note that this does not occur when you alter the GO that was the origin of the Prefab).

A Prefab has an additional element at the Actual Object level in the Inspector to Apply/Revert Overridden values BE CAREFUL RE APPLYING TO SOURCE PREFAB.

In this Scene, the Prefab is Green and the GO is Pink - notice that Prefab is using the Translate method for movement and the not-Prefab GO is Clamped with Transform...

-> Helpful Links

<https://docs.unity3d.com/Manual/Prefabs.html>

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 8/12

g_ / h_:

Creating a new Prefab that has an AudioSource (Purple). Also adding automated movement via (MotionOscillator_v1.cs) - see notes on code.

*** Be conscious of your Prefab naming, it can be awkward to rename them after you have used a bunch of Instances!**

In regards to AudioSource, since I added the Audio Mixer Asset later in the Tutorial, it is part of the Project

****Signal flow goes from AudioClip to Audio Output to Audio Listener. You do not need the Audio Mixer Asset to use sound, it will default its output to the Audio Listener, which defaults to the Camera.**

Here want you to experiment with the AudioSource parameters, play with the MotionOscillator Vars/Fields in conjunction with changes to the “3D Sound Settings”. You have many built in options, even Keyframe based editing of spatial processing settings...

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 9/12

g_ / h_:

-> Helpful Links

<https://docs.unity3d.com/2019.1/Documentation/Manual/Audio.html>

<https://docs.unity3d.com/2019.1/Documentation/Manual/class-AudioSource.html>

<https://docs.unity3d.com/2019.1/Documentation/Manual/class-AudioClip.html>

<https://docs.unity3d.com/2019.1/Documentation/ScriptReference/ AudioSource-clip.html>

<https://docs.unity3d.com/Manual/class- AudioListener.html>

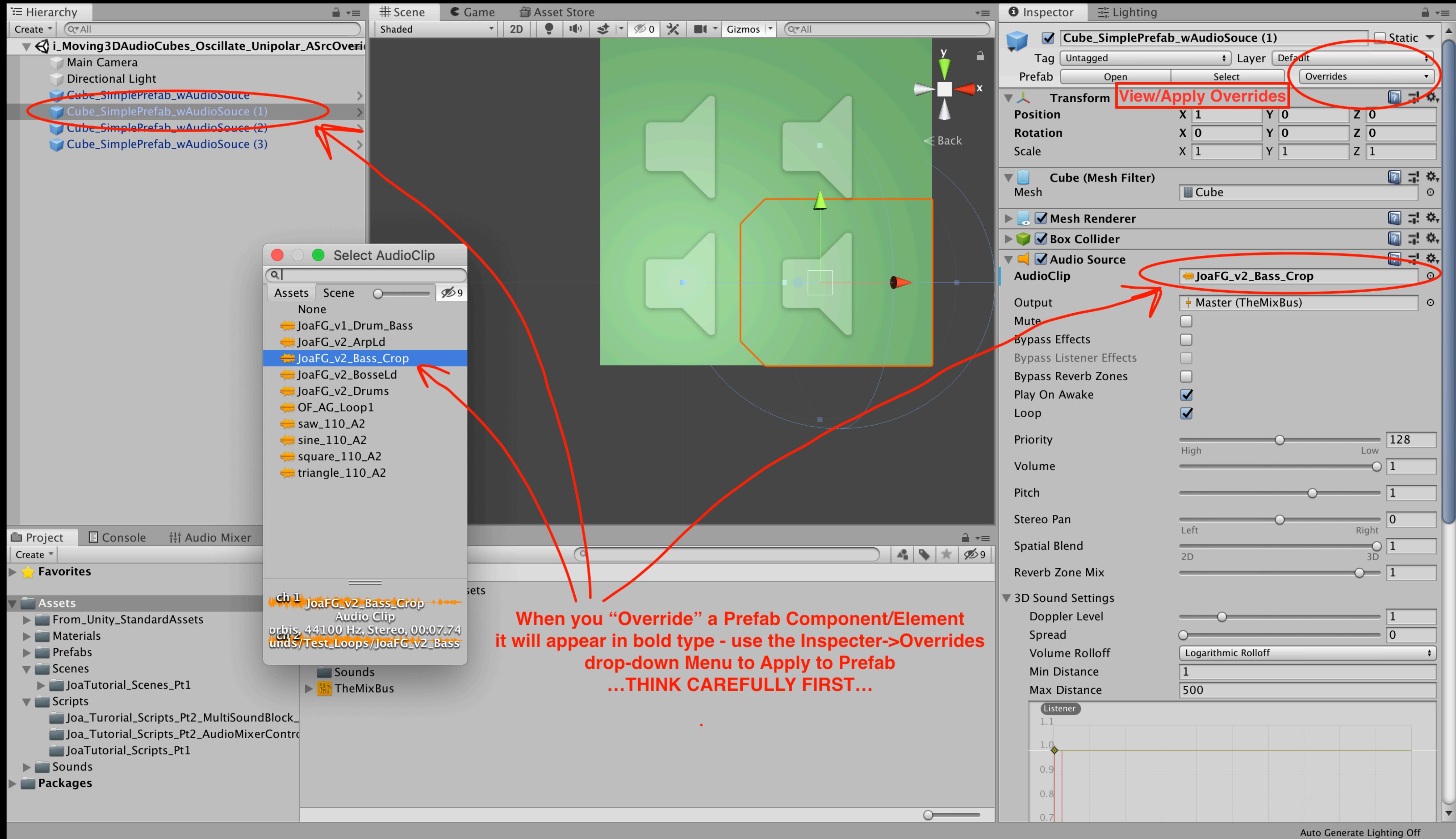
<https://docs.unity3d.com/ScriptReference/ AudioListener.html>

<https://docs.unity3d.com/ScriptReference/ UnityEngine.AudioModule.html>

i_ : Four Instances of the Prefab w/ different loops for each AudioSource - each also has its own “Osc_1_Vect” Overrides. (see next page)

Have some fun toggling the “Bi Polar Flag” and Overriding “Osc_1_Vect” Editor Fields :)

joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 10/12



joa_audio_primitives_1

Tutorial_1_Pt1 Notes 11/12

j_:

→ Before making these changes, let's go through how to Copy/Rename of Unity Project

Similar to previous Scene, but Camera movement Component/Script is put back in and I disabled the funky Doppler.

Test out the Unity mixing/spatialization processes as you “drive” the Camera around...

We will download/add the “SpaceSkies Free” 6-sided Skybox “Asset” together to pretty it up.

Time permitting, do the same with some Texture Assets for the sound/loop Prefabs.

Assets are added from the Asset Store View (Download/Import - with options of what to Import) OR Drag existing Assets into Project Pane. If it is a .unitypackage, you can Drag it into Hierarchy as well. Assets added will reside in your Project folder structure, but you may also choose to save your favorites to make access easier as needed.

joa_audio_primitives_1

Tutorial_1_Pt1 Notes 12/12

End of Tutorial_1_Pt1 Comments:

* Note we could add a script to control Start/Stop of Loops now, using a C# “List” and use:

“FindObjectsOfType<SimplePrefab_wAudioSource>();”

<https://docs.unity3d.com/ScriptReference/Object.FindObjectsOfType.html>

- This would allow use of “foreach” method, but we will doing that moving forward with integration of a master “Transport”

DOH! Just realized my spelling typo in the Prefab name, but do not want to open the Prefab/ Overrides can of worms at the moment :(

For you to do:

- Make your own loops
- Put Textures on your sound cubes
- Extra Credit - Add SpaceBar to Start/Stop Audio and Oscillation

joa_audio_primitives_1

Tutorial_1_Pt2 Notes 1/23

a_ :

Make (MultiSoundBlock_V1.cs) plus GO container prototyping a Prefab that can play different sounds by Using [SerializeField] in our script.

Demonstrate how to update TextMesh “Type” with (UpdateTextMesh.cs). Implement a (ProcessMouseInput.cs).

***Note this is not a Prefab, but a Component Type because the script is attached to GO container**

```
updateTextMesh = FindObjectOfType<UpdateTextMesh>();  
multiSndBlockV1 = FindObjectOfType<MultiSndBlockV1>();
```

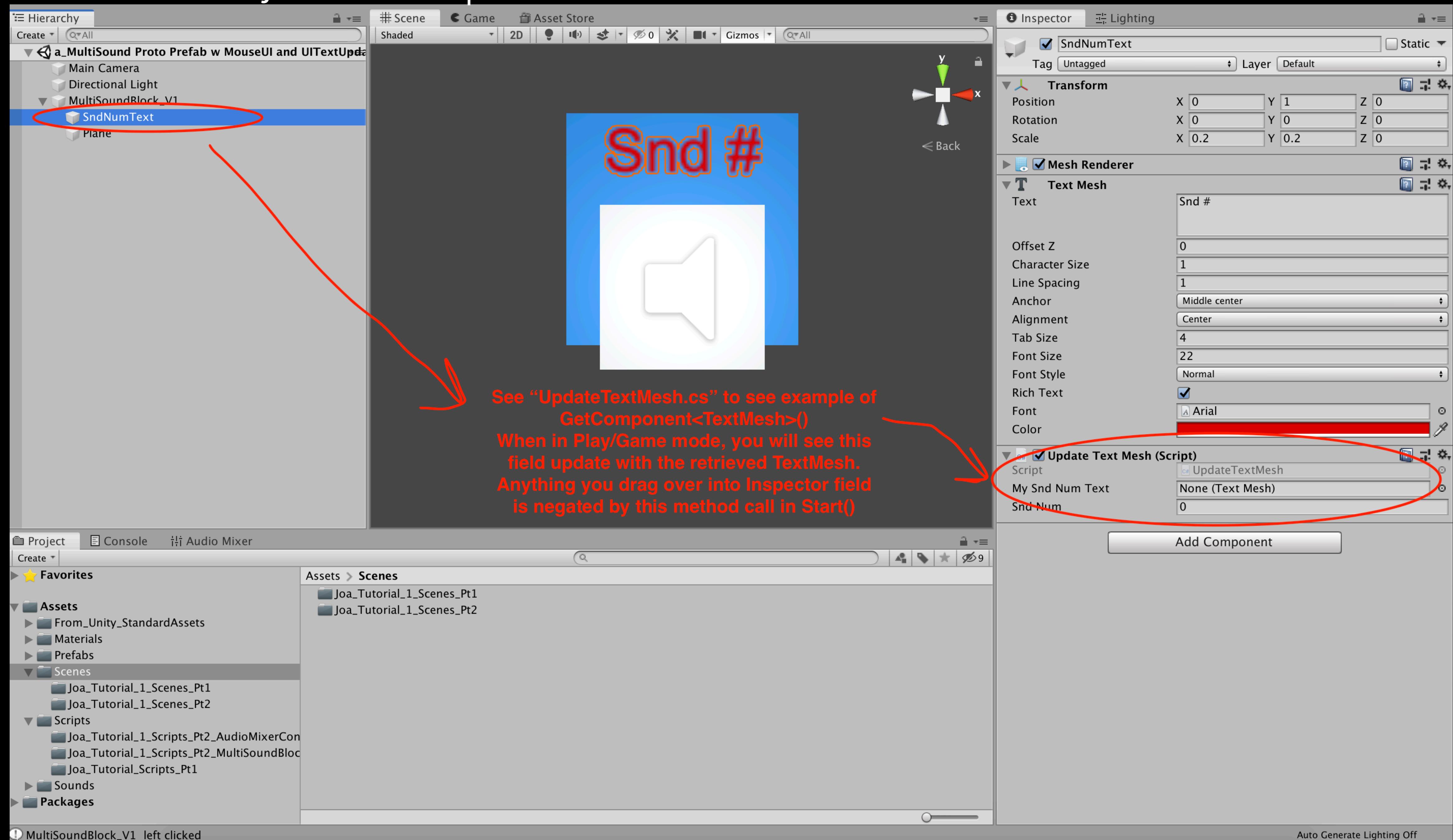
Now, using the Unity Message “OnMouseOver()” -part of the MonoBehavior Unity Base class , we can process Mouse Buttons to select what sound to play and Start/Stop...

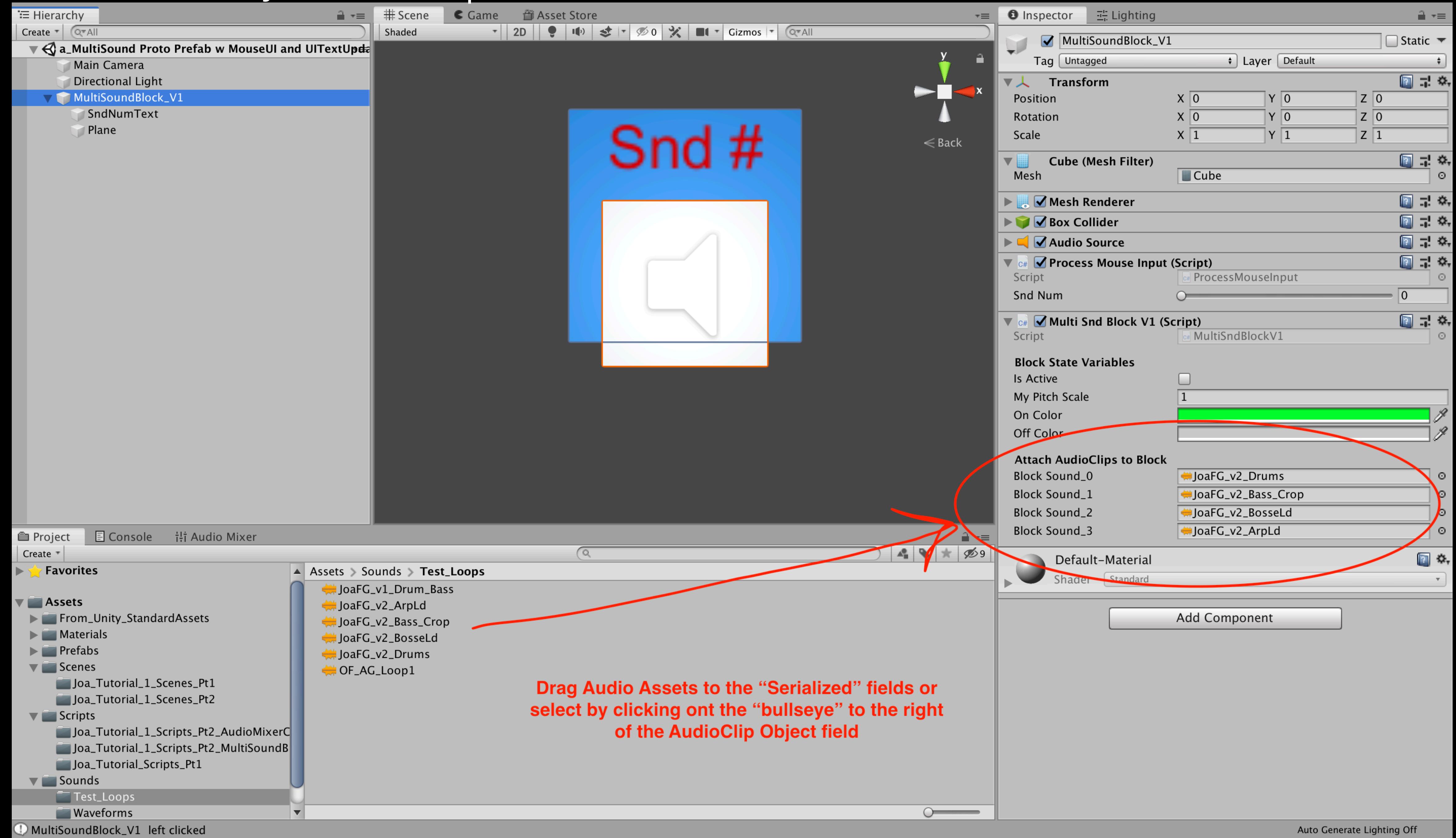
***Note that we can view the Audio Mixer Asset, from Unity menu - Window->Audio->Audio Mixer (or Command-8) I have renamed it “TheMixBus”. You can enable “Edit in Play Mode” to use the Mixer while running.**

<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>

<https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseOver.html>

joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 2/23





This Object - prototype for a Multiple Sound Prefab has its Output sent to the AudioMixer Asset, renamed "TheMixBus" AudioClip field is empty because the sounds are assigned in MultiSndBlock_V1.cs when messages from ProcessMouseInput.cs are sent to it...

Hierarchy

- a_MultiSound Proto Prefab w MouseUI and UITextUpdate
- Main Camera
- Directional Light
- MultiSoundBlock_V1

Scene

Inspector

Cube (Mesh Filter)

- Mesh: Cube
- Mesh Renderer**: Enabled
- Box Collider**: Enabled
- Audio Source**: Enabled
- AudioClip**: None (Audio Clip)
- Output**: Master (TheMixBus)
- Mute**: Off
- Bypass Effects**: Off
- Bypass Listener Effects**: Off
- Bypass Reverb Zones**: Off
- Play On Awake**: Off
- Loop**: On
- Priority**: 128
- Volume**: 0.81
- Pitch**: 1
- Stereo Pan**: 0
- Spatial Blend**: 0
- Reverb Zone Mix**: 1

Process Mouse Input (Script)

- Script: ProcessMouseInput
- Exposed Parameters (4): Snd Num

Multi Snd Block V1 (Script)

- Script: MultiSndBlockV1
- Block State Variables**
 - Is Active: On
 - My Pitch Scale: 1.0
 - On Color: Red
 - Off Color: Gray
- Attach AudioClips to Block**
 - Block Sound_0: JoaFG_v2_Drums
 - Block Sound_1: JoaFG_v2_Bass_Crop
 - Block Sound_2: JoaFG_v2_BosseLd
 - Block Sound_3: JoaFG_v2_ArpLd

Project

Console

Audio Mixer

Mixers

- TheMixBus (Audio Listener) – Inactive

Snapshots

- Snapshot

Groups

- Master

Views

- View

Master

Exposure: 0

Volume: 0

Range: -80.0 dB to 20

S M B

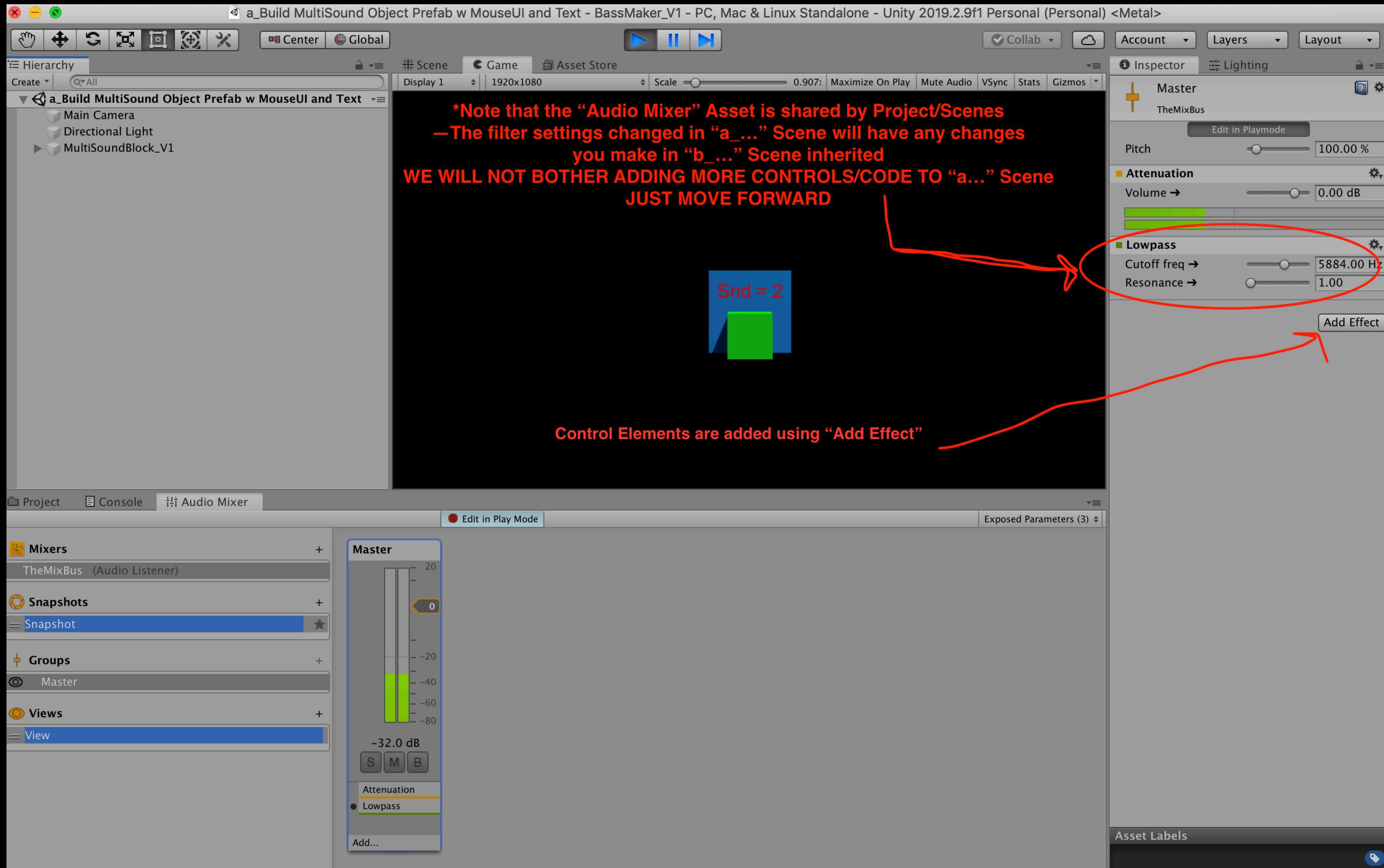
Lighting

POSITION: X: 0, Y: 0, Z: 0

ROTATION: X: 0, Y: 0, Z: 0

SCALE: X: 1, Y: 1, Z: 1

Auto Generate Lighting Off



joa_audio_primitives_1

Tutorial_1_Pt2 Notes 6/23

b_ :

Canvas and UI Concepts Introduced - `UIToggle`, `UISlider` and using “OnValueChanged” Unity Message via UI element Inspector.

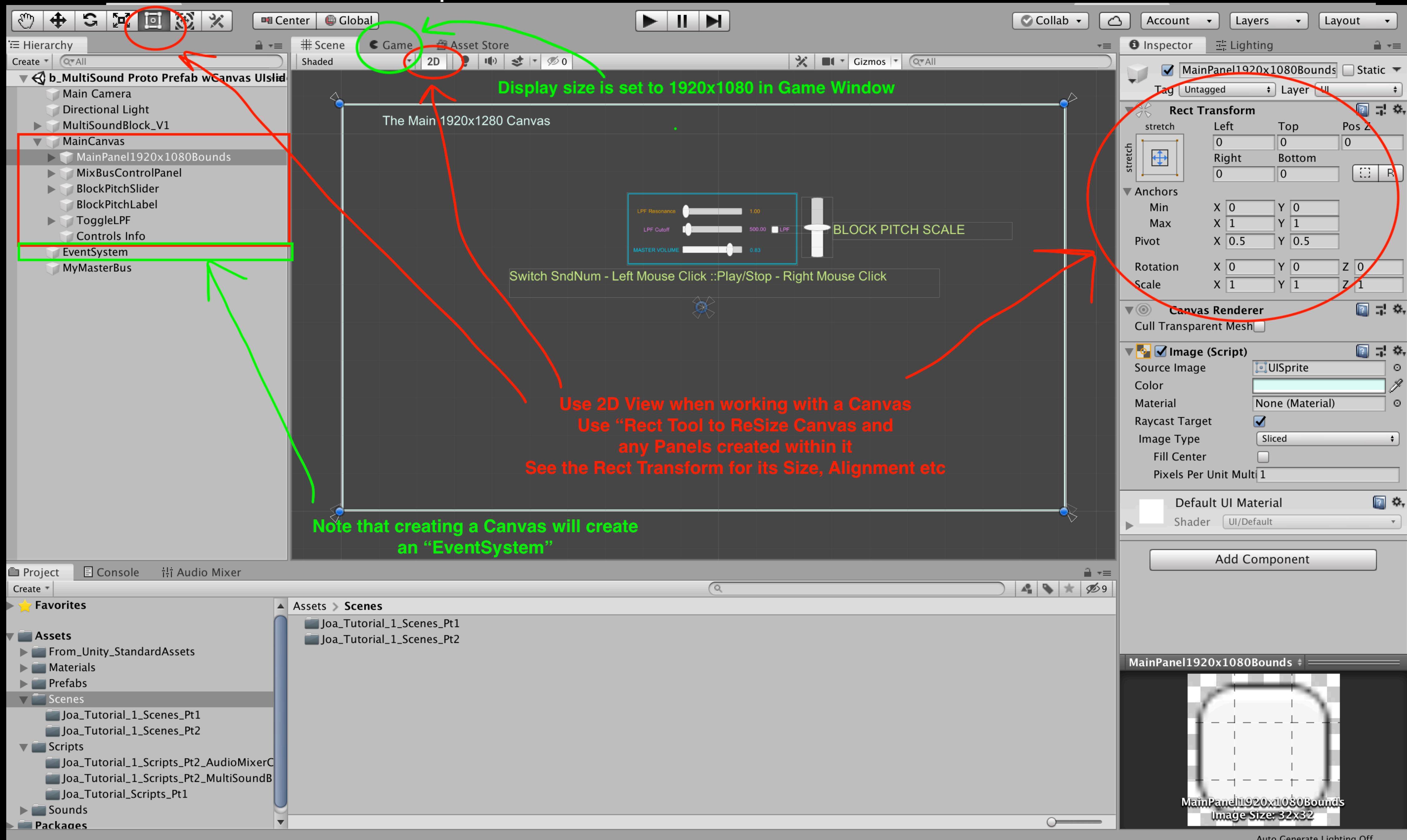
Adds (`MasterBusHandler.cs`) with LPF Frequency as “Exposed” parm for `UISlider` control -

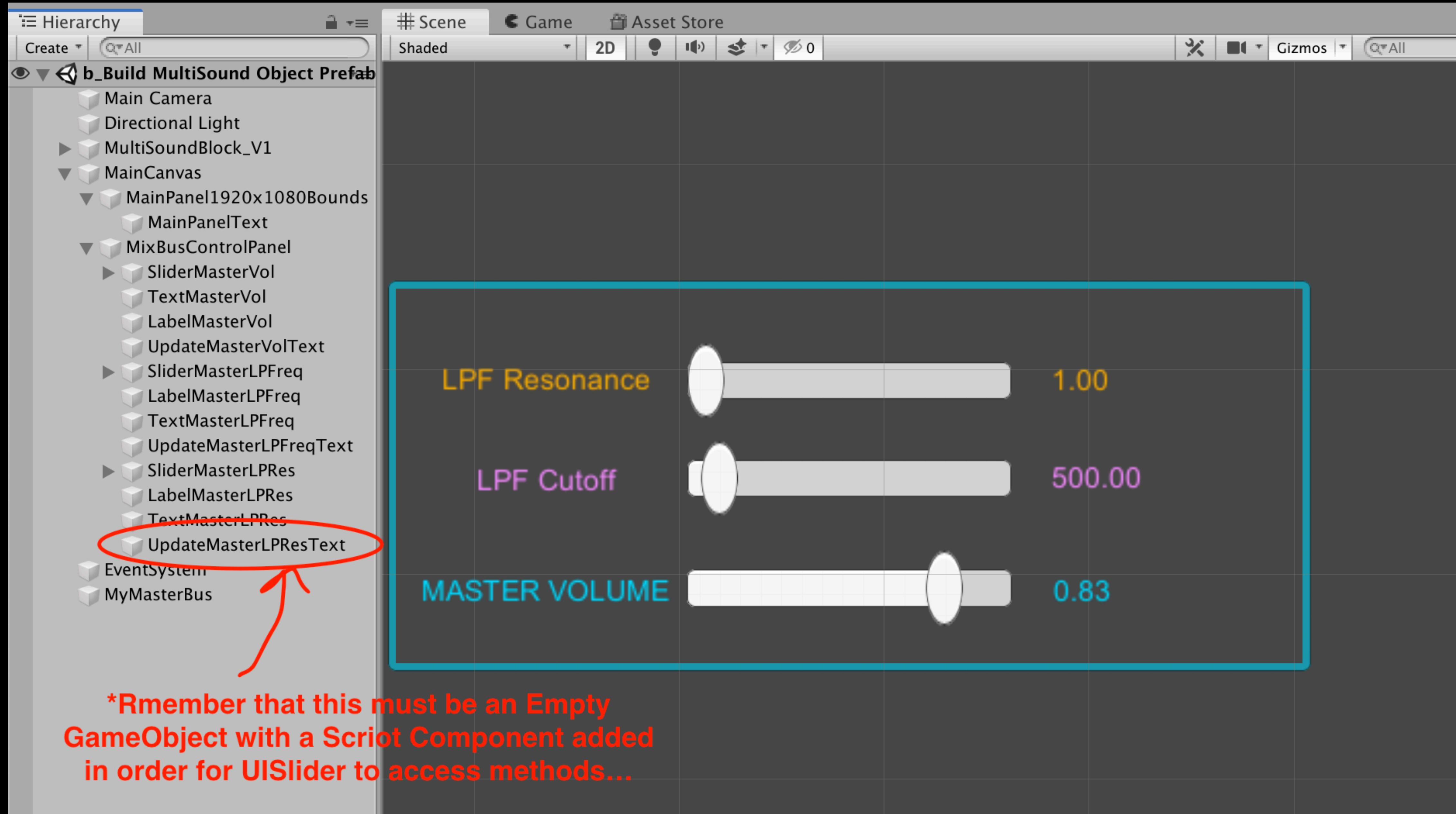
****NOTE THAT THERE WILL BE AN ERROR ON RUNNING THIS SCENE - I ADDED A “PANELHANDLER” SCRIPT TO UPDATE THE SLIDERS IN FUTURE SCENES, THIS IS A GOOD CHANCE TO SHOW THAT YOU MAY WANT TO ADD A FLAG TO INDICATE WHEN A METHOD IS USED OR NOT IN A SCENE THAT IS USING A SCRIPT THAT CHANGES IN THE FUTURE - WE WILL DEMO THIS WITH OUR SEQUENCER CODE...**

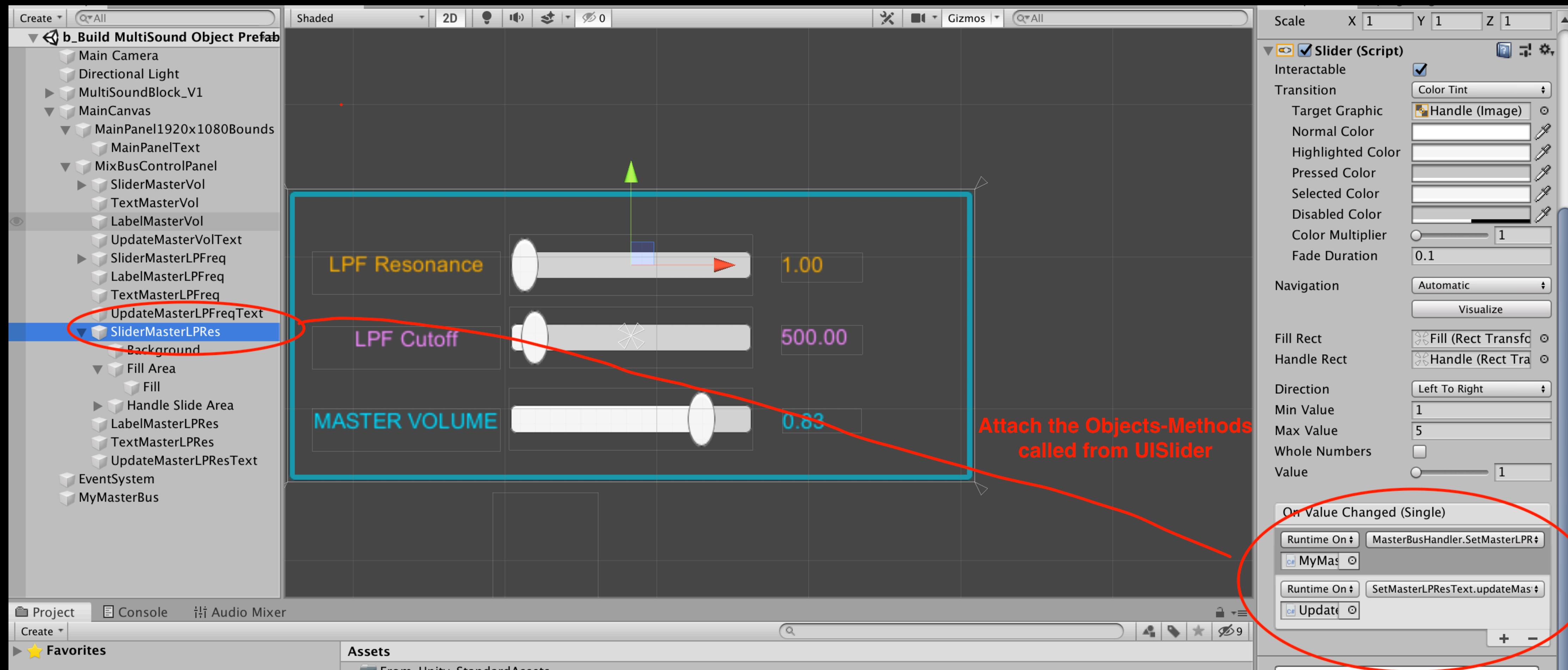
<https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html>

<https://docs.unity3d.com/2019.2/Documentation/ScriptReference/UIElements.Slider.html>

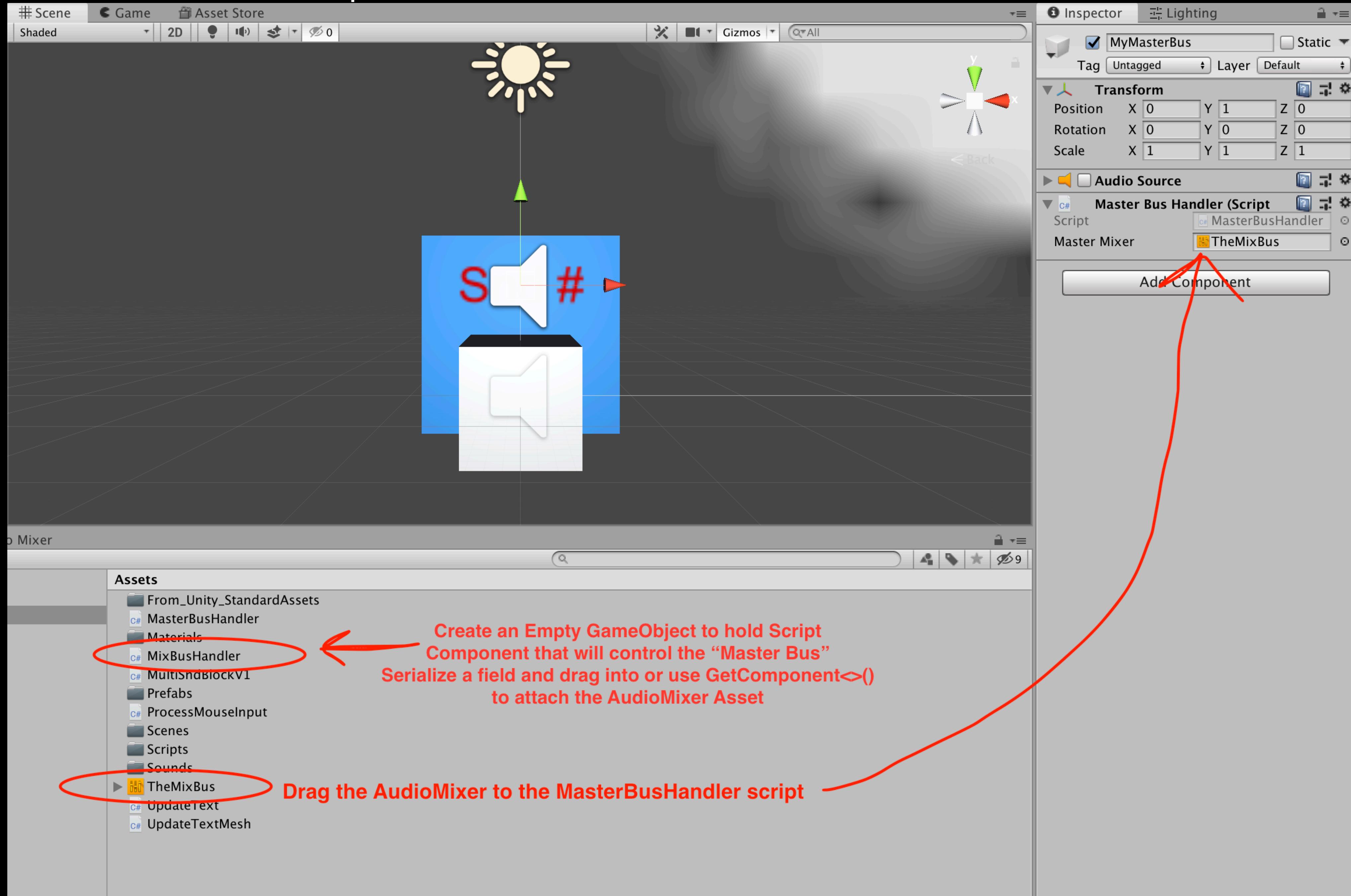
<https://docs.unity3d.com/2019.2/Documentation/ScriptReference/UIElements.Toggle.html>







joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 10/23



joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 11/23

The Main 1920x1280 Canvas

Switch SndNum - Left Mouse Click ::Play/Stop - Right Mouse Click

Snd #

BLOCK PITCH SCALE

LPF Resonance: 1.00
LPF Cutoff: 500.00 LPF
MASTER VOLUME: 0.83

MasterBusHandler.cs is a Component of EmptyGameObject that I named “MyMasterBus”

This is a Singleton aquired in the Script using FindObjectsOfType<MasterBusHanler>()

THIS IS AN ACTUAL AudioMixer CLASS...

Mixers
TheMixBus (Audio Listener) – Inactive

Snapshots
Snapshot

Groups
Master

Views

Inspector

MyMasterBus

Transform

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

Audio Listener

Audio Source

AudioClip: None (Audio Clip)
Output: Master (TheMixBus)
Mute:
Bypass Effects:
Bypass Listener Effects:
Bypass Reverb Zones:
Play On Awake:
Loop:
Priority: 128
Volume: 1
Pitch: 1
Stereo Pan: 0
Spatial Blend: 0
Reverb Zone Mix: 1

3D Sound Settings

C# **Master Bus Handler (Script)**

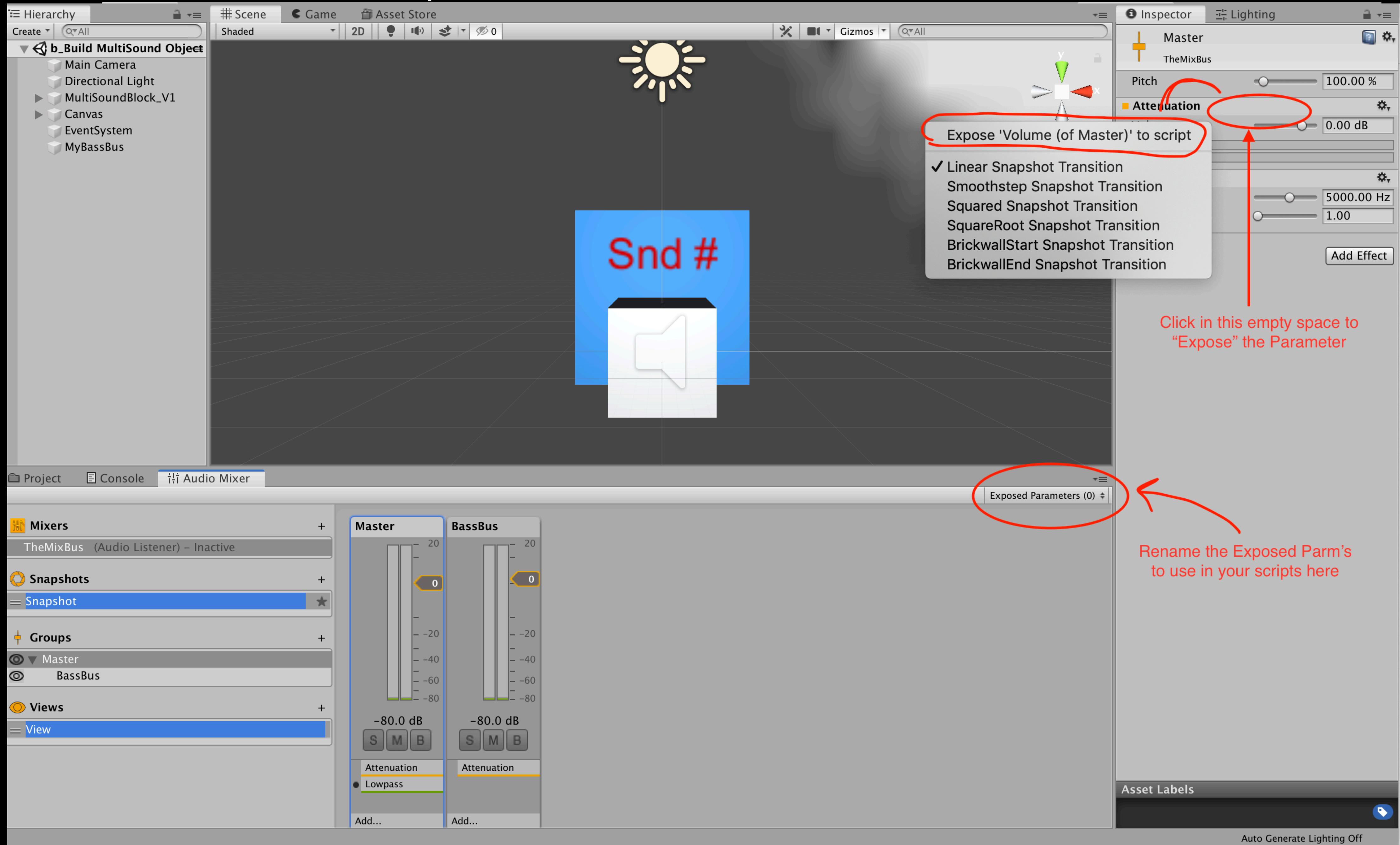
Script: MasterBusHandler

The Master Volume: 0.83
The Master LP Freq: 15245
The Master LP Reson: 1.2
Lpf Active:
Lfo Active:
Lfo Bi Polar:
Lpf Freq Stepper:
Lfo_rate: 2
Depth Factor: 50
LP Freq Steps
Size: 0
Master Mixer: TheMixBus

Add Component

Auto Generate Lighting Off

joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 12/23



joa_audio_primitives_1

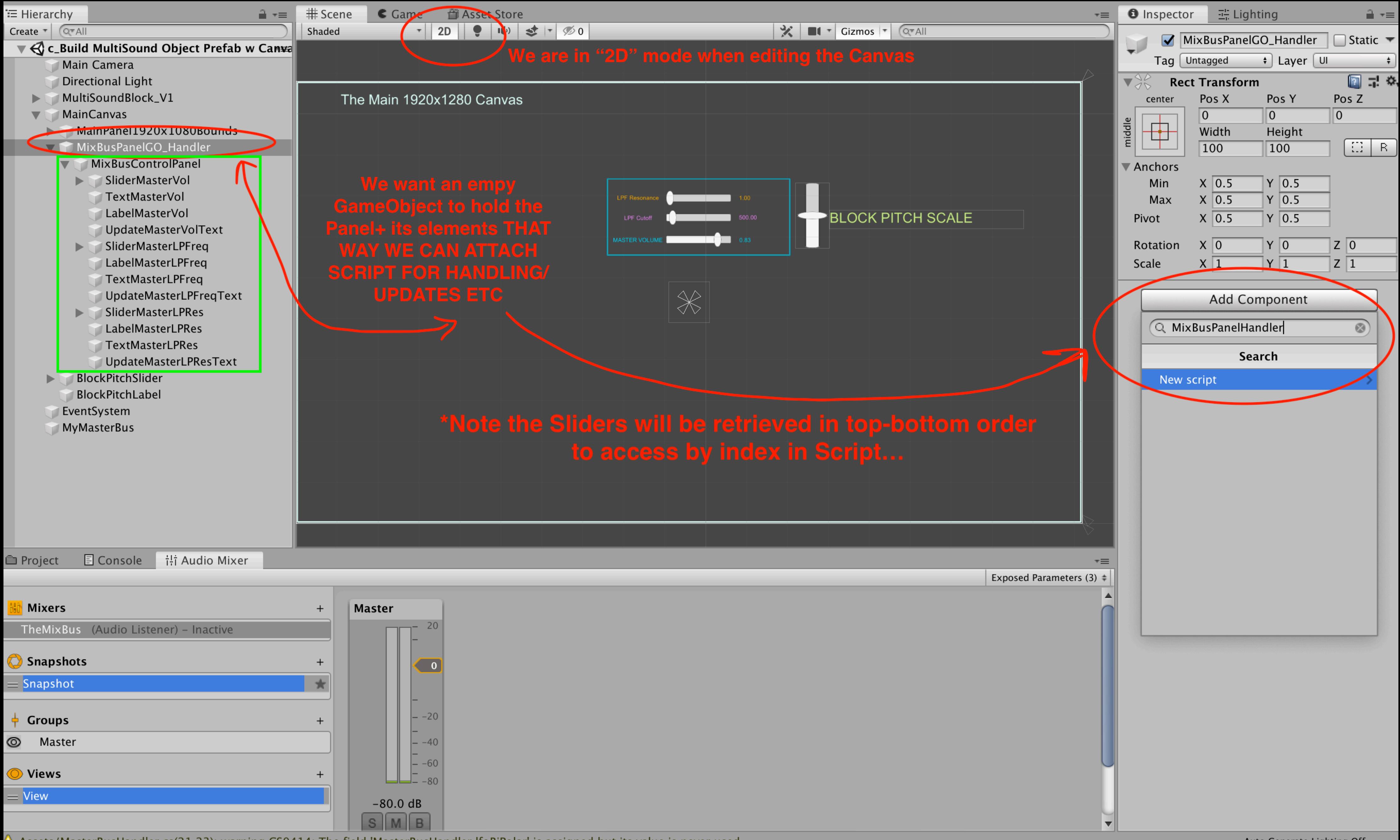
Tutorial_1_Pt2 Notes 13/23

c_ :

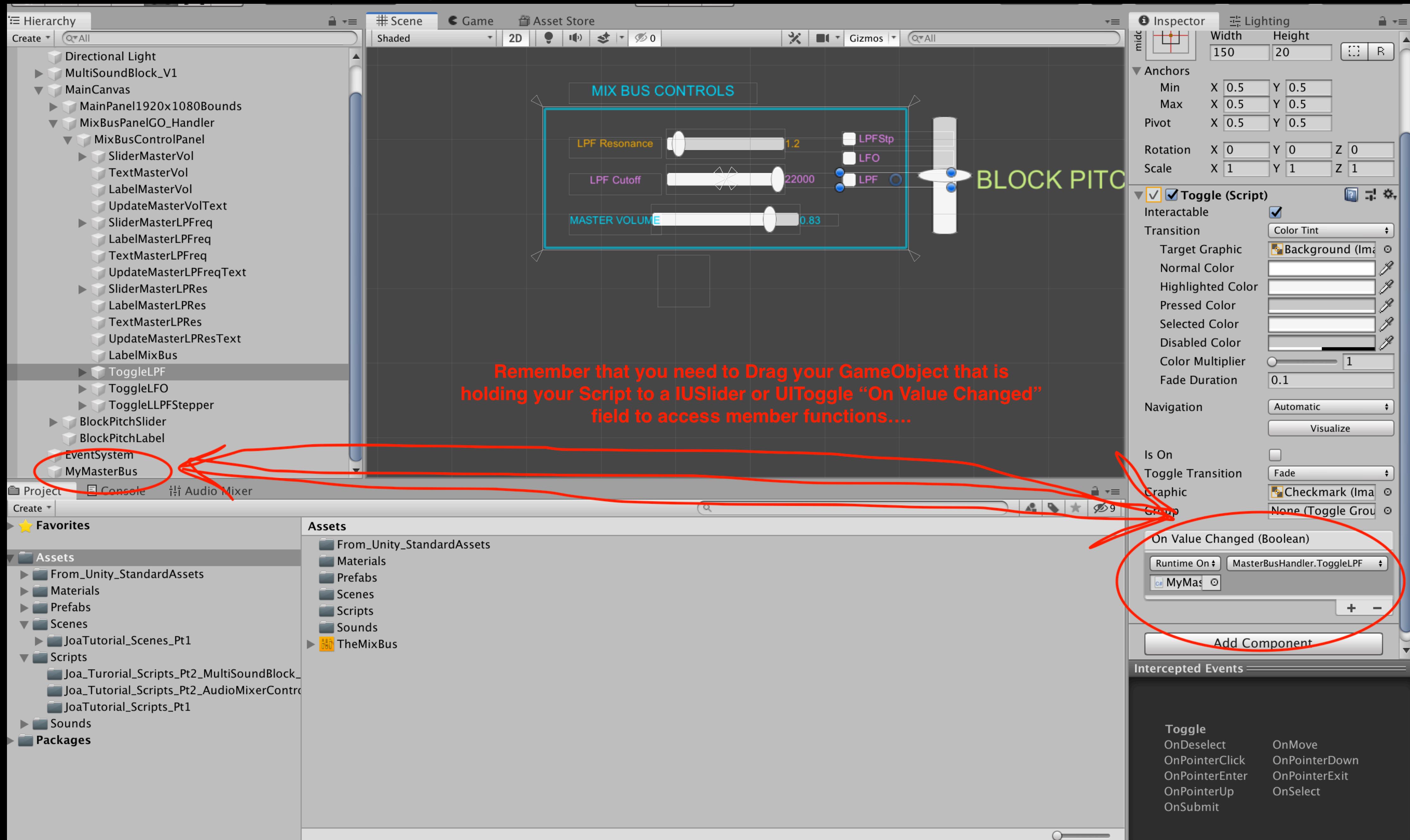
Integrates (MixBusPanelHandler.cs) for UI Slider Updates - See notes in script(s)

```
mixBusSliders = GetComponentsInChildren<Slider>();
```

<https://docs.unity3d.com/ScriptReference/Component.GetComponentsInChildren.html>



joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 15/23



joa_audio_primitives_1

Tutorial_1_Pt2 Notes 16/23

d_ :

Integrates LFO Functionality - See notes in script(s)

<https://answers.unity.com/questions/781748/using-mathfsin-to-move-an-object.html>

e_ :

Integrates a Coroutine to “step” through a <List> of “Floats” to make an LPF Frequency Stepper
Integrates LFO Functionality

<https://stackoverflow.com/questions/44701073/instantiating-a-list-of-gameobjects-in-unity-c-sharp>

<https://forum.unity.com/threads/foreach-a-list-of-objects.453140/>

<https://docs.unity3d.com/Manual/Coroutines.html>

MIX BUS CONTROLS

LPF Resonance: 1.2
LPF Cutoff: 22000
MASTER VOLUME

BLOCK PITCH

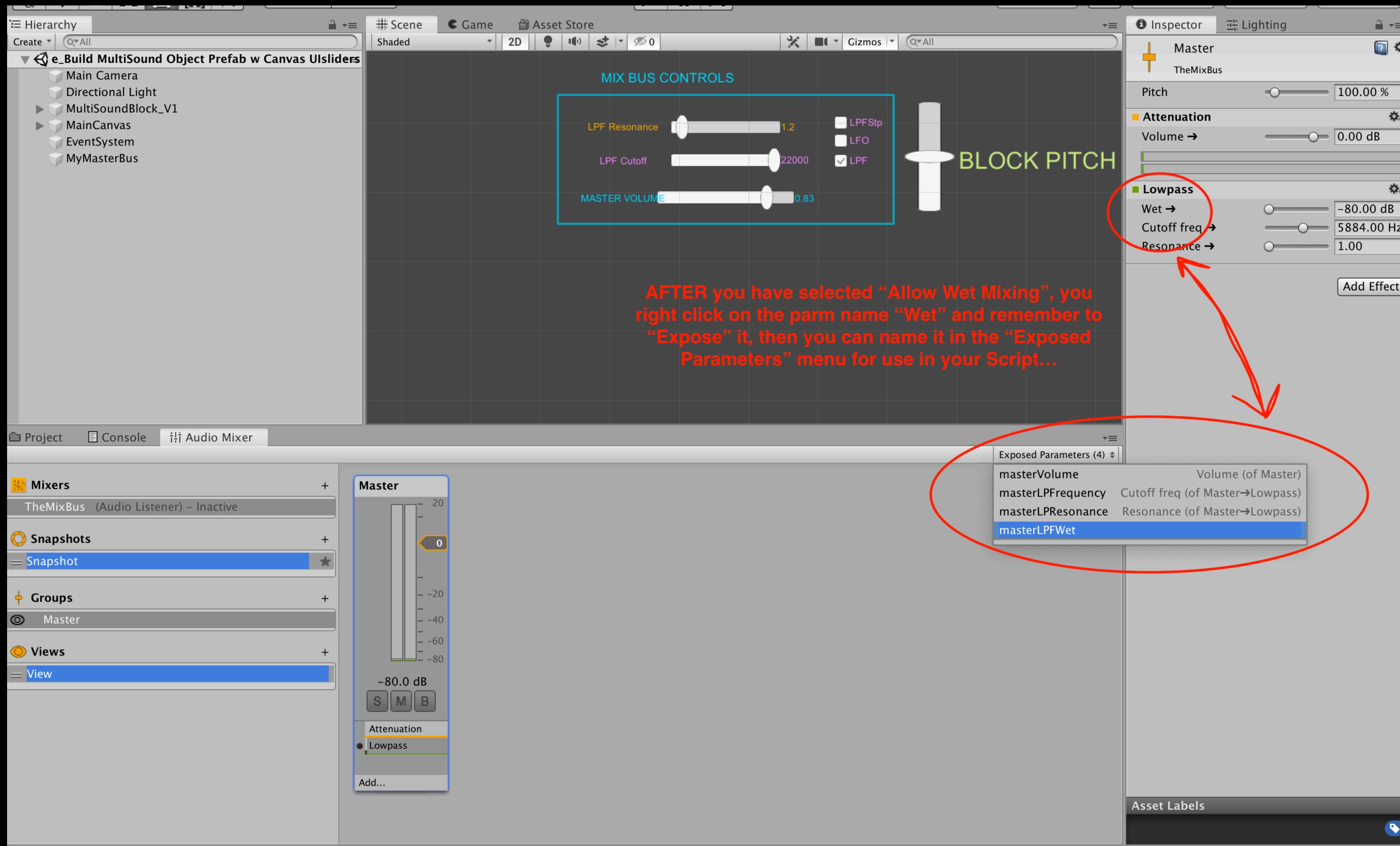
Allow Wet Mixing (causes higher memory usage)
Bypass
Copy effect settings to all snapshots
Add effect before
Add effect after
Remove this effect

Right-Click here and select "Allow Wet Mixing" to be able to Bypass via Min/Max

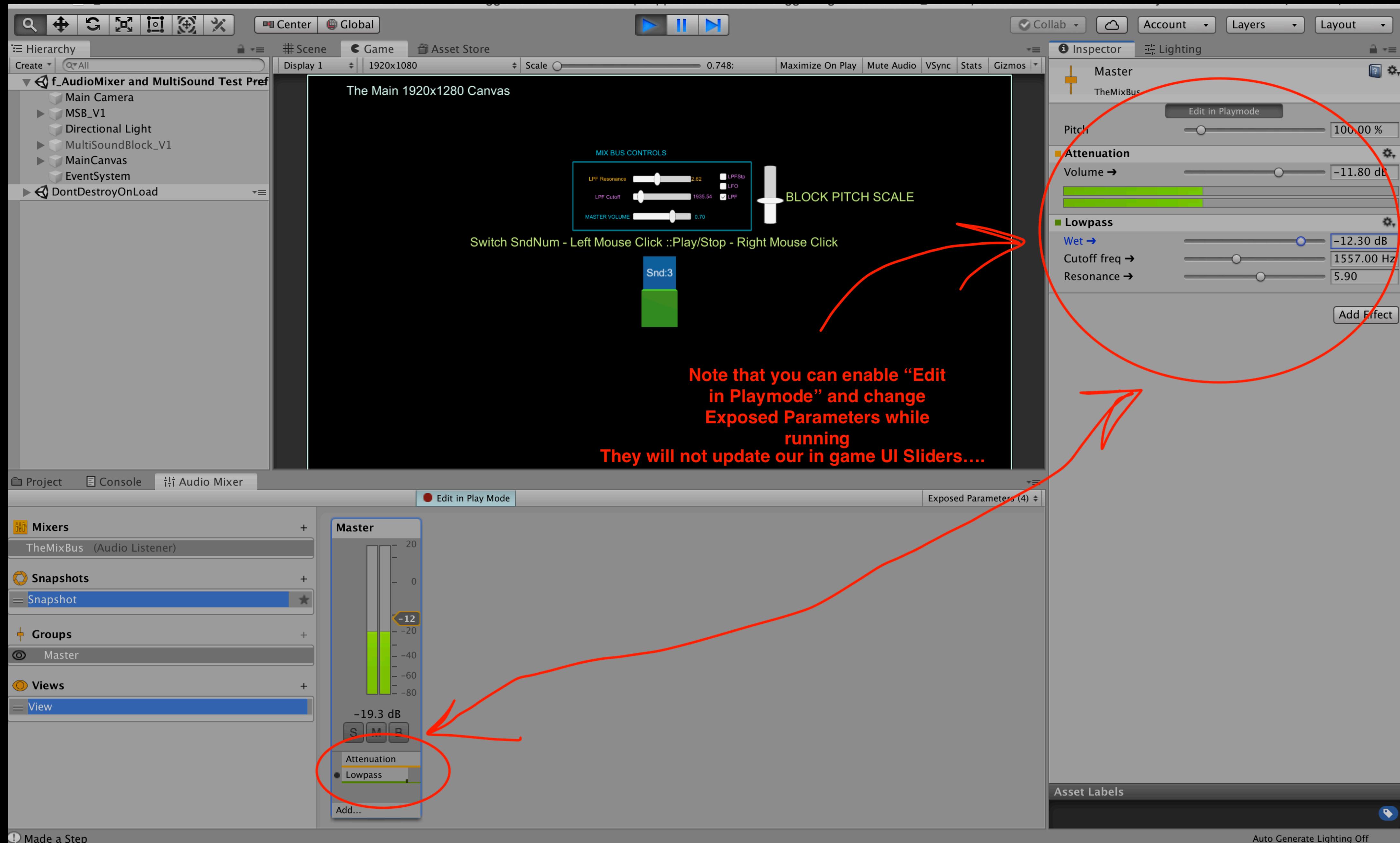
Master

Attenuation: -80.0 dB
Lowpass
Add...

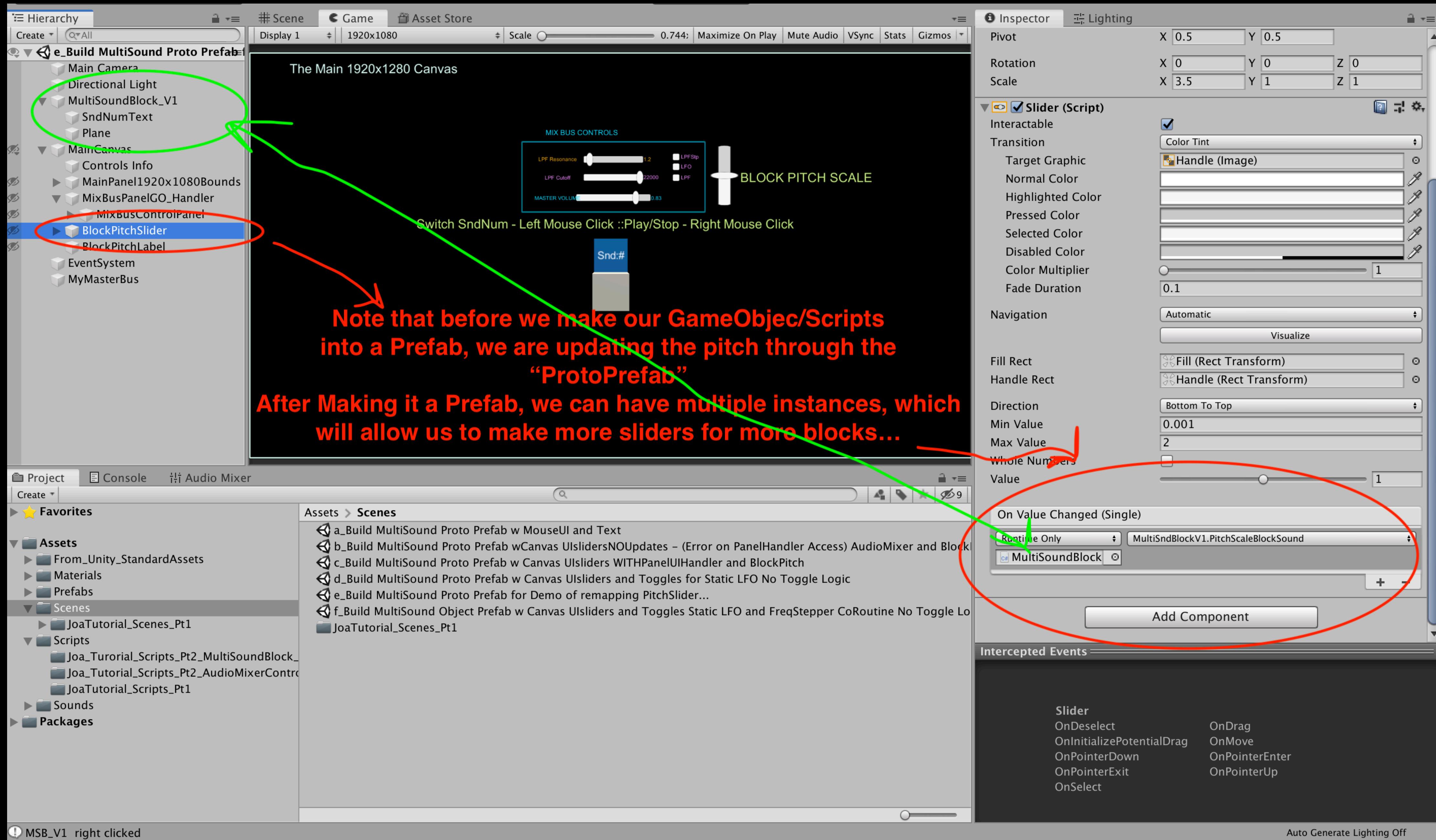
To Toggle the LPF On/Off we need to use and “Exposed Parameter” and set to Max/Min respectively in Code - It will warn you this is more taxing on the system...



joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 19/23



joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 20/23



joa_audio_primitives_1

Tutorial_1_Pt2 Notes 21/23

f_ :

Same as E but now talking to Prefab version - Works fine with one Instance, BUT we need to reassess how we process mouse input to handle multiple Instances...look what happens in g_ !

g_ :

Demonstrates need for Re-Factoring, in particular the movement of Input Processing - make it unique to each Instance of the Prefab. We will create MSB_V2 Prefab in Tutorial 2

Student Assignments

- 7/10 Create your own loops - Redesign the Master Bus Control Panel - Add a UISlider to change speed of LPStepper & Customize layout to your own design - Change Colors for sliders to match text
- 8/10 Make a BPM Tempo UISlider for LPStepper
- 9/10 Make a BPM Tempo UISlider for LPStepper - AND LFO - AND make it 8 Steps
- 10/10 Also Fix the Prefab

joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 22/23

The screenshot shows the Unity Editor interface with the following components:

- Hierarchy Panel:** Shows the scene structure, including the `f_Build MultiSound Prefab` root object, which contains the `Main Camera`, `MSB_V1`, `Directional Light`, `MultiSoundBlock_V1`, `MainCanvas`, and `MixBusPanelGO_Handler`. The `SliderMasterLPFreq` component is highlighted with a red oval.
- Scene View:** Displays the **MIX BUS CONTROLS** UI panel. It features three sliders: **LPF Resonance** (value 1.2), **LPF Cutoff** (value 22000), and **MASTER VOLUME** (value 0.83). To the right of the sliders are buttons for **LPFStp**, **LFO**, and **LPF**.
- Inspector Panel:** Focuses on the `SliderMasterLPFreq` component. Key settings include:
 - Rect Transform:** Tagged as **Untagged**, Layered under **UI**.
 - Slider (Script):** Set to **Interactable**.
 - Color Tint:** Enabled, set to **1**.
 - Handle (Image):** Set to **Visualize**.
 - Fill Rect:** Set to **Left To Right**, **Min Value** is **30**, **Max Value** is **22000**, and **Whole Numbers** is checked.
 - On Value Changed (Single):** Two methods are assigned:
 - Runtime Only:** `MasterBusHandler.SetMasterLPFrequency` (highlighted with a red oval).
 - Runtime Only:** `SetMasterLPFreqText.updateMasterLPFreqText` (highlighted with a red oval).
- Project Panel:** Shows the project structure, including `Assets`, `Scripts`, and `Packages`. The `Joa_Tutorial_1_Scripts_Pt2_AudioMixerControls` script folder is selected, containing scripts like `MasterBusHandler`, `MixBusPanelHandler`, `SetMasterLPFreqText`, `SetMasterLPResText`, and `SetMasterVolText`.

Annotations:

- A large red arrow points from the `SliderMasterLPFreq` entry in the Hierarchy panel down towards the `On Value Changed` section in the Inspector panel.
- Two specific method assignments in the `On Value Changed` section are circled in red:
 - `MasterBusHandler.SetMasterLPFrequency`
 - `SetMasterLPFreqText.updateMasterLPFreqText`

joa_audio_primitives_1 : Tutorial_1_Pt1 Notes 23/23

The Main 1920x1280 Canvas

MIX BUS CONTROLS

LPF Resonance: 1.2 LPFSlp: LFO: LPF:

LPF Cutoff: 22000

MASTER VOLUME: 0.83

BLOCK PITCH SCALE

Switch SndNum - Left Mouse Click ::Play/Stop - Right Mouse Click

Snd: #

Now that it has been made into a Prefab the UISlider for pitch can talk to that specific Instance - this means more MultiSoundBlock_V1's can be added to a Scene along with pitch sliders for each...

Here you see the Prefab below dragged into replace the "Proto" code It has been renamed MSB_V1 - the pitch slider OnValueChanged now communicates to that Instance...

****BUT - Now we need to rewrite how we process**

- Start/Stop of sound
- Change Prefab code to reference itself...
- Start "Joa_Tutorial_Primitives_2"

On Value Changed (Single)

Runtime Only: MultiSndBlockV1.PitchScaleBlockSound

MSB_V1 (MultiSn)

Intercepted Events

Slider	OnDeselect	OnDrag
OnInitializePotentialDrag	OnMove	OnPointerEnter
OnPointerDown	OnPointerExit	OnPointerUp
OnSelect		

MSB_V1 right clicked

Auto Generate Lighting Off

Joa_Tutorial_1 Notes

Unity Project/Folder Name: joa_audio_primitives_1

Student Assignments

- 7/10 Create your own loops - Redesign the Master Bus Control Panel - Add a UISlider to change speed of LPStepper & Customize layout to your own design - Change Colors for sliders to match text
- 8/10 Make a BPM Tempo UISlider for LPStepper
- 9/10 Make a BPM Tempo UISlider for LPStepper - AND LFO - AND make it 8 Steps
- 10/10 Also Fix the Prefab