

# The Scripts You need – 1

## Beginners Guide

Mostly we use these C# scripts to create common elements in games. So I have created these simple scripts, they are really useful in most developments.

Scripts List : CSharp

- Random Sounds on Collision.
- Switch between camera – Perspective to Orthographic.
- Basic Camera Zoom (FOV using Scroll).
- Destroy Game object after Time.
- Simple Text - FPS Counter
- Moving Platform (with Speed and Directions)
- Random Static Rotation of objects
- Rotating Objects (like Showcase weapons or Helicopter Propeller or Fan ).
- Torch Light with Sound
- Trigger Events ( play Audio ,Turn on Light like Alarm ,Destroy object, Load Scene.

Now How to Use it for Beginners –

### Random Sounds on Collision

1. Drag and Drop the BounceSound.cs script to the object that you want to make a collision.
2. Add AudioSource to that object for playing Audio. Keep it empty.

3. Drag all the Clips you wanted to add in the Clips array.

The image shows the Unity Inspector window for a script named 'Bounce Sounds (Script)'. The 'Clips' array is expanded, showing 4 elements: 'sound 1', 'sound 2', 'sound 1', and 'sound 2'. The 'Src' is set to 'Sphere (Audio Source)'. An arrow points from the 'Add AudioClips' button to the 'Clips' array. Another arrow points from the 'Drag and Drop same object' button to the 'Src' field. Below the 'Clips' array is the 'Audio Source' component, which has various settings like 'Mute', 'Bypass Effects', 'Bypass Listener Effe', 'Bypass Reverb Zone', 'Play On Awake', 'Loop', 'Priority', 'Volume', 'Pitch', 'Stereo Pan', 'Spatial Blend', 'Reverb Zone Mix', and '3D Sound Settings'.

**Bounce Sounds (Script)**

- Script: BounceSounds
- Clips
  - Size: 4
  - Element 0: sound 1
  - Element 1: sound 2
  - Element 2: sound 1
  - Element 3: sound 2
- Src: Sphere (Audio Source)

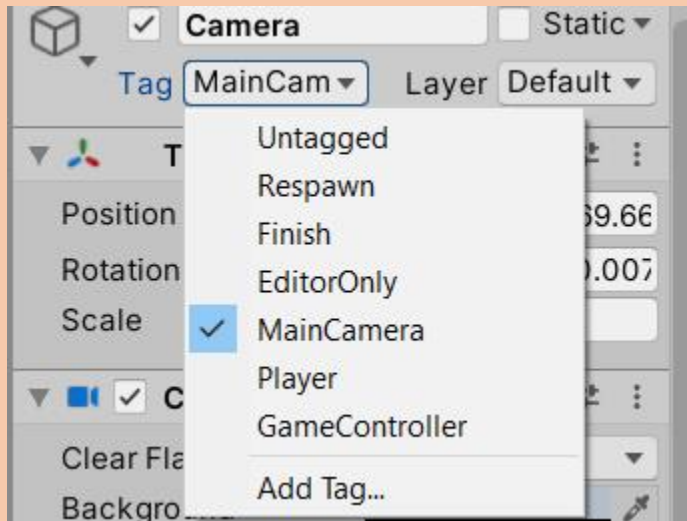
**Audio Source**

- AudioClip: None (Audio Clip)
- Output: None (Audio Mixer Group)
- Mute: ☐
- Bypass Effects: ☐
- Bypass Listener Effe: ☐
- Bypass Reverb Zone: ☐
- Play On Awake: ☒
- Loop: ☐
- Priority: 128 (High to Low)
- Volume: 1
- Pitch: 1
- Stereo Pan: 0 (Left to Right)
- Spatial Blend: 0 (2D to 3D)
- Reverb Zone Mix: 1
- 3D Sound Settings
  - Doppler Level: 1
  - Spread: 0
  - Volume Rolloff: Logarithmic Rolloff

4. There you go. When ever it collides, random sound plays.

## Switch between camera – Perspective to Orthographic

1. Create a Camera
2. Make sure the Tag is MainCamera. Or you can change it in the Scripts.



3. Drag and Drop the Script – CamSwitch.cs to the camera
4. You can change the Keycode in Script.

```
void Update()
{
    if (Input.GetKeyUp(KeyCode.Z))
    {
        if (Camera.main.orthographic == true)
        {
            Camera.main.orthographic = false;
        }
        else
        {
            Camera.main.orthographic = true;
        }
    }
}
```

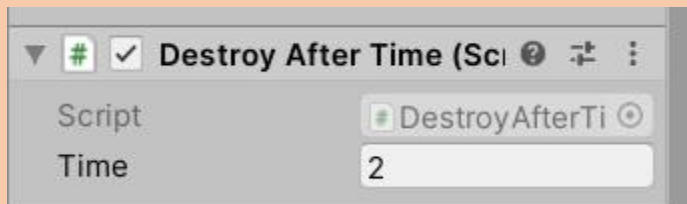
Here change Z to anything you want.

## Basic Camera Zoom (FOV using Scroll)

1. Create a Camera and change the Tag as Maincamera.
2. Drag and Drop the Script – CamZoom.cs to the Camera.
3. You can Use your Mouse Scroll up to Zoom in and Scroll down to Zoom out.

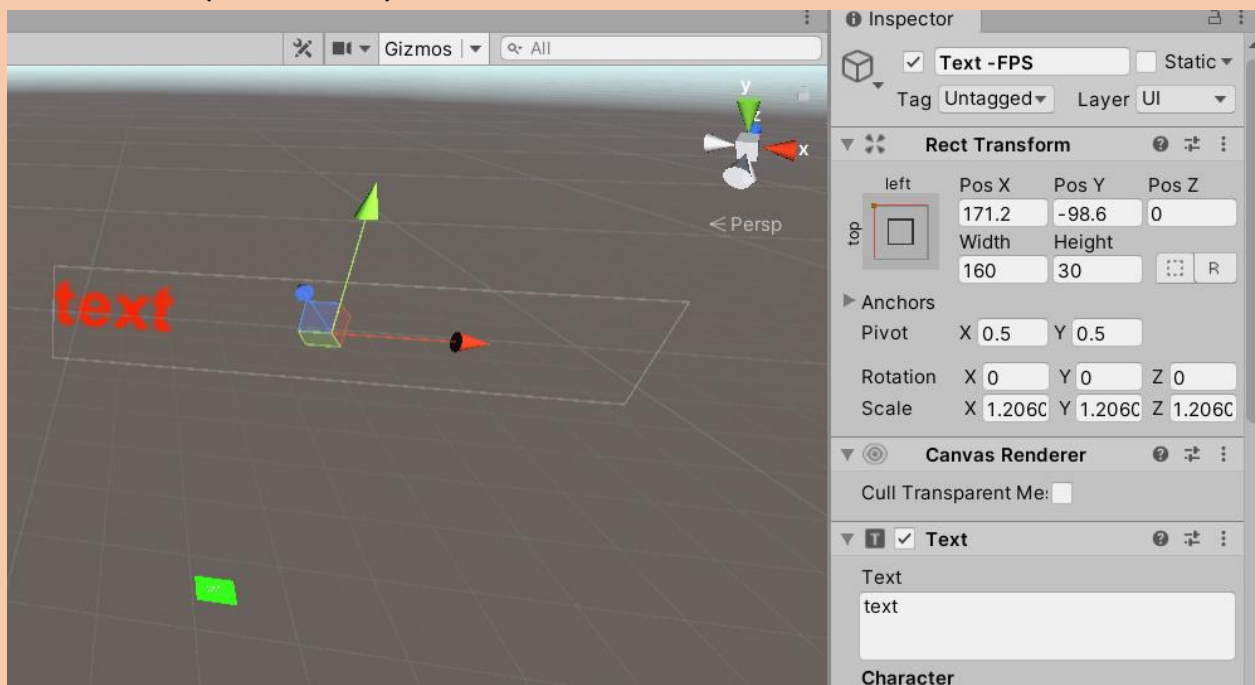
## Destroy Game object after Time

1. Add the DestroyAfterTime.cs to any object that you want to destroy.
2. Set the time as Seconds. [ setting destroy after time ]



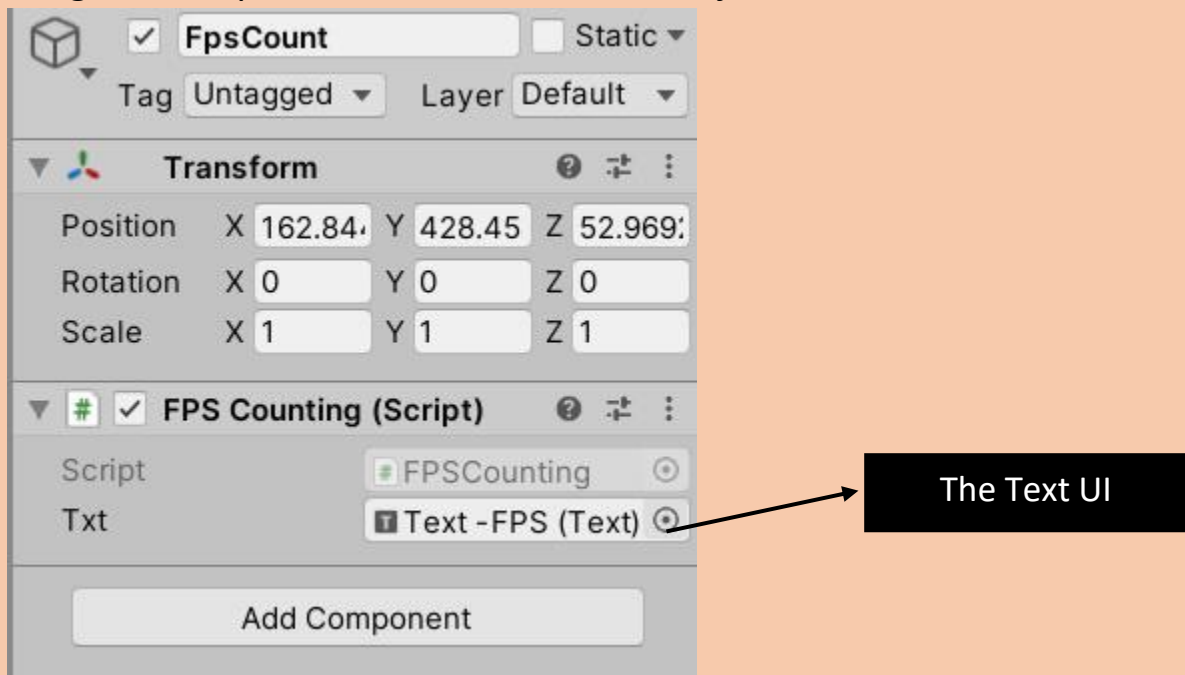
## Simple Text - FPS Counter

1. Add a Text (Canvas UI).



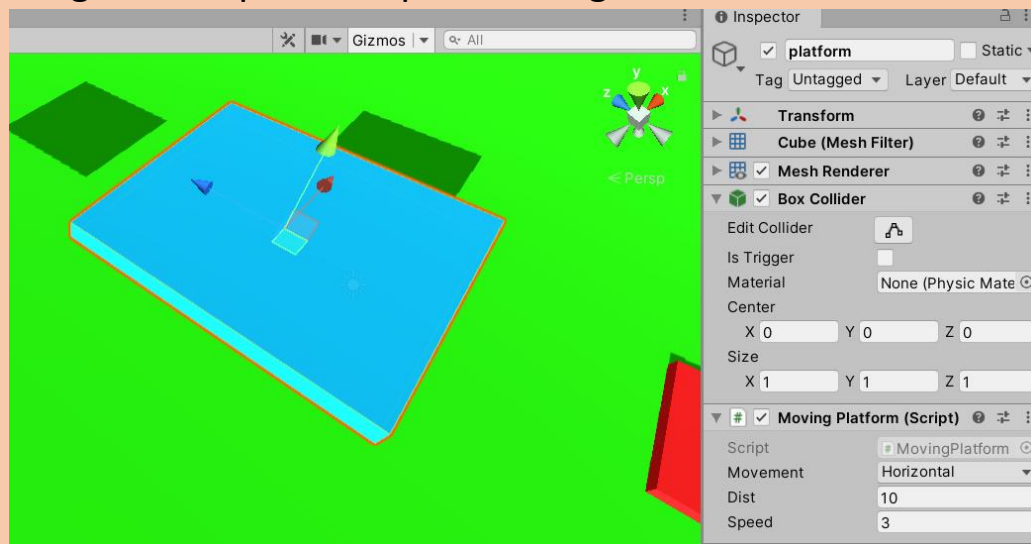
2. Create a EmptyGameObject and Drag and Drop the Script – FPSCounting.cs

### 3. Drag and Drop the Text UI to the Text Object

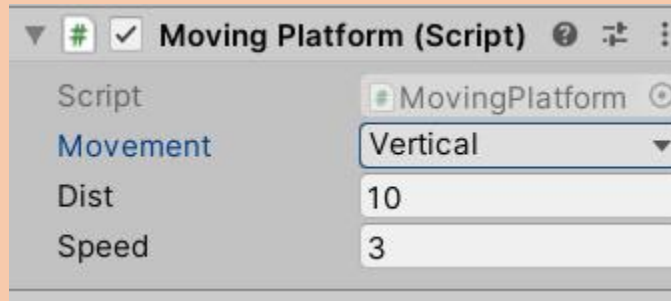


## Moving Platform (with Speed and Directions)

1. Create object like platform or maybe you already have.
2. Drag and Drop the Script – MovingPlatform.cs

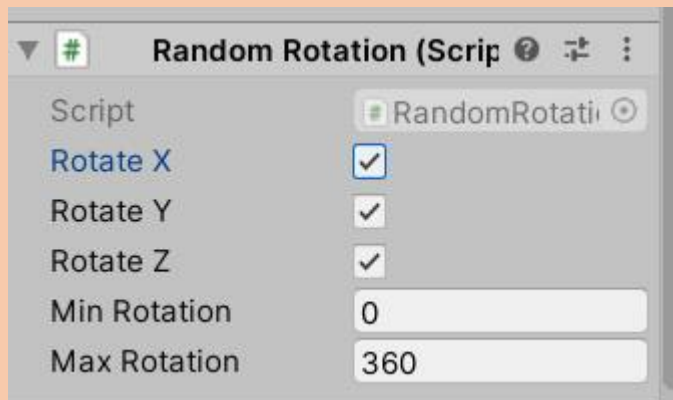


3. Movement = direction (choose vertical or Horizontal)
4. Dist = distance that the platform should move
5. Speed = Speed of the moving Platform.



## Random Static Rotation of objects

1. This creates a Random static rotation for the selected Objects.
2. Drag and Drop the RandomRotation.cs script to the object.
3. Rotate X , Y, Z – Choose which Axis you wants to rotate , If you are using Rock or Random Car you have to only use X .like Basic things.
4. Min Rotation is that which Eulerangle should the object rotated 0 – 90 , 90 – 180 – User Preferences.
5. Max Rotation is that which Eulerangle should the object rotates below the max angels like 270 – 360. – User Preference.

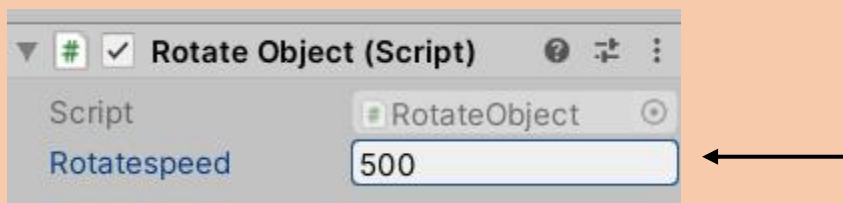


## Rotating Objects

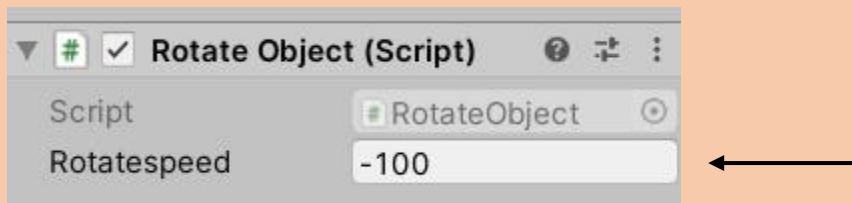
1. For example, if you want to make a Propeller fan.
2. Drag and Drop the RotateObject.cs script to the object.
3. RotateSpeed is the speed of the rotation.

-Showcase model set the speed 10 to 50

-Fan or Propeller set the speed 1000 to 1500



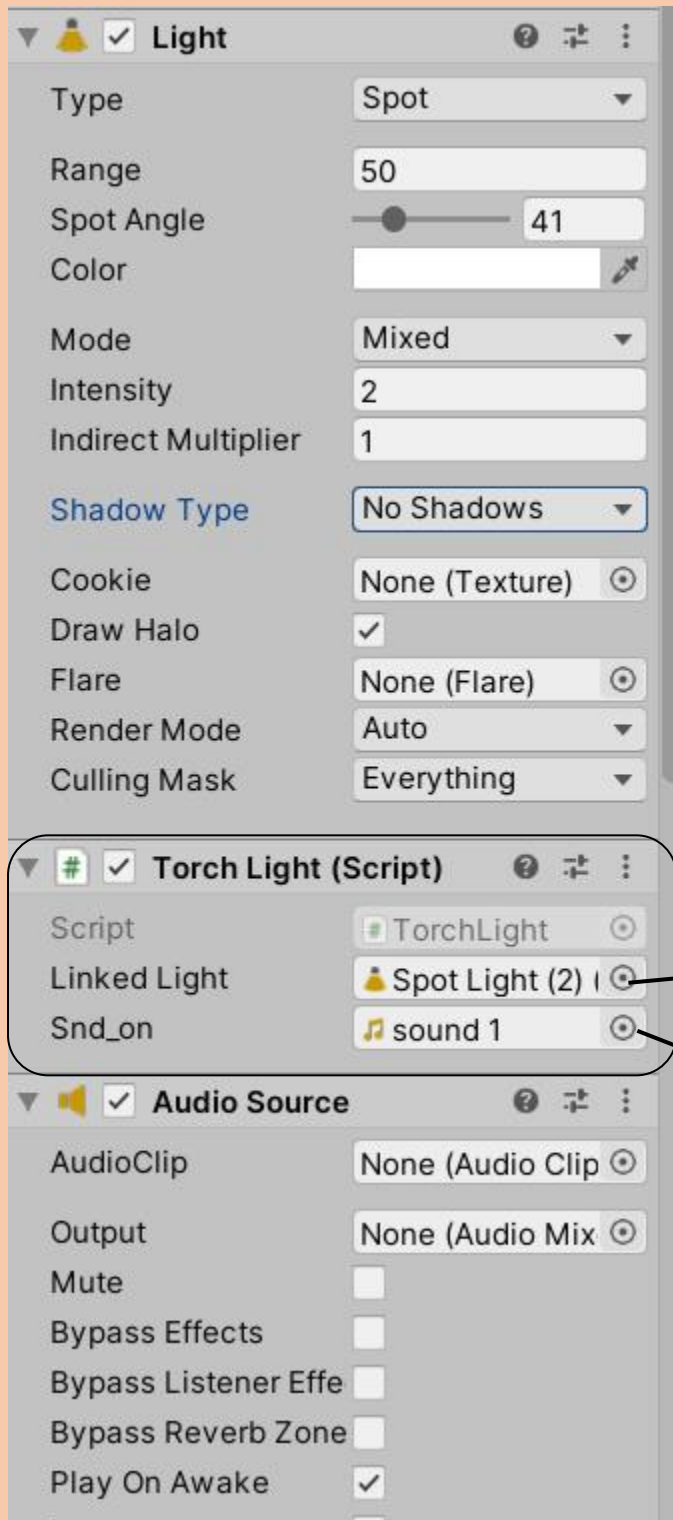
4. If you want to rotate in anti-clock wise then use minus speed



## Torch Light with Sound

1. Create a Light or you may already have.
2. Drag and drop the TorchLight.cs script to that.
3. You can also Create a empty object or maybe flashlight Model and drag the Light Object to the script.

4. Now add an AudioSource to the Light or Flashlight model.



You can also add this script to any gameobject and add a audio source, just drag and drop the light.

Drop the Light here

Add sound for on/off

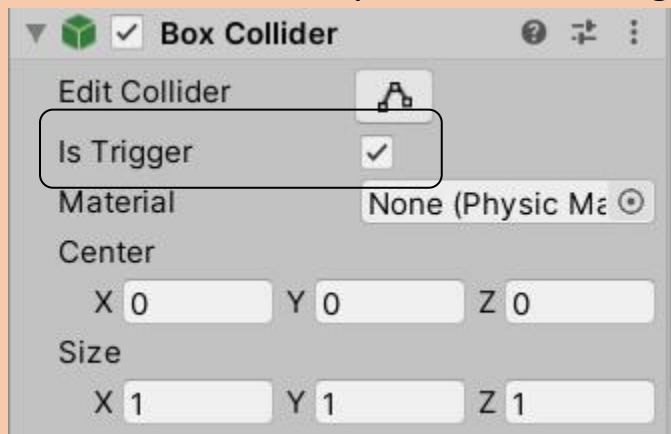


5. Now Press F or you can change it in the Script.

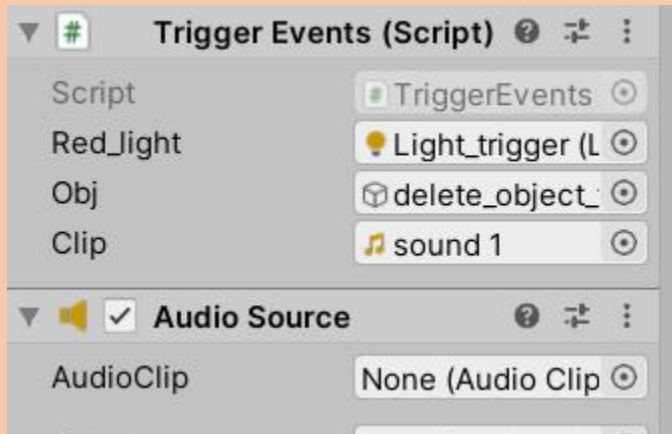
```
id Update()  
  
    if (Input.GetKeyDown(KeyCode.F))  
    {  
        linkedLight.enabled = !linkedLight.enabled;  
    }  
}
```

## Trigger Events

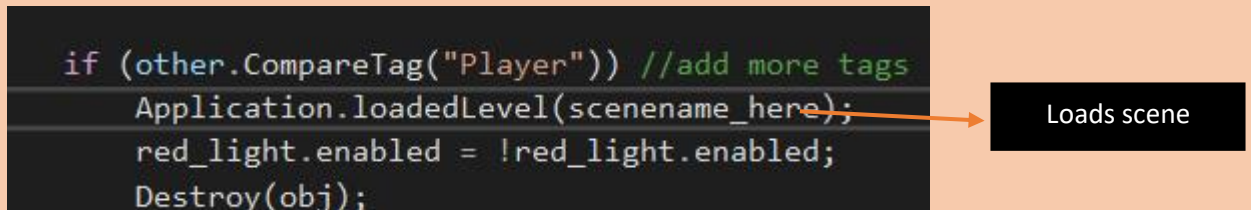
1. Trigger events is when player or any object, when it collides with the scripted object it calls an event.
2. Create a Cube or any collider with Trigger checked on .



3. Now drag and drop the Script – TriggersEvent.cs script.
4. Make sure the Player Tag is set to “Player”.
5. Now Add an Audio Source if you want trigger an audio.
6. Red\_light = Light which is turned on if the player collides.
7. Obj = If you want to destroy an Object after collision.
8. Clip = Plays Audio after collision.



9. If you want to Load Scene on Collision you can add it in scripts.



Hope its useful for you!

-APEX