The Essential Scripts
The Common Scripts that your game requires.

Update : V2

List of Scripts:

- *Switch Camera*
- *Dynamic Collision sound*
- *Camera Zoom Event*
- *Swarm Spawn system*
- *Pop after time*
- *Rotation Controller*
- *Object Transform (Platform Movement)*
- *FlashLight and Car Light system*
- *Dynamic Trigger Events*
- *Dynamic Wind System*
- *Wheel Control (power Gen)*

Switch Camera

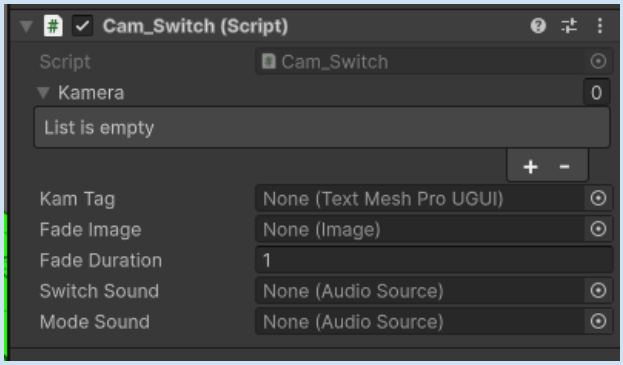
This Unity script allows users to switch between multiple cameras using keyboard inputs (Q, E & X for Mode)with smooth fade transitions, audio management, and camera mode toggling. The current camera's name and mode are displayed on-screen.

1. Create Cameras

- Go to GameObject > Camera and create multiple cameras, as much as you need.

2. Attach Script

- Create an empty GameObject (e.g., "Camera Manager").
- Drag and drop the Cam_Switch script onto it.



3. Assign Cameras

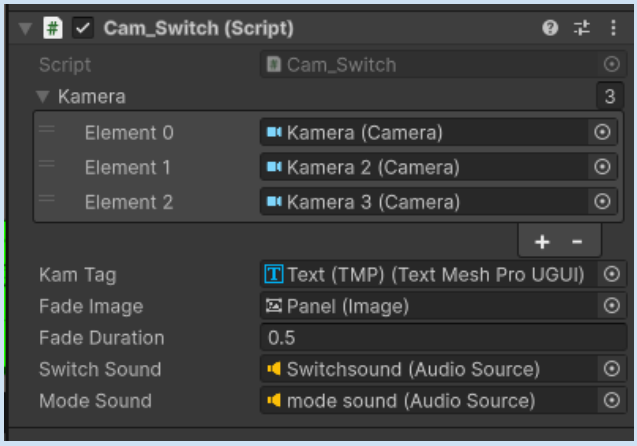
- Select the "Camera Manager" GameObject.
- In the Inspector, find the Cam_Switch script.
- Drag and drop each camera into the Kamera list.
- Now, if you have multiple Cameras, it causes Conflict (AudioListeners).so,

```
foreach (var cam in Kamera)
{
    var audioListener = cam.GetComponent<AudioListener>();
    if (audioListener != null)
    {
        audioListener.enabled = false;
    }
}
if (Kamera[0].GetComponent<AudioListener>() != null)
{
    Kamera[0].GetComponent<AudioListener>().enabled = true;
}
```

We are setting audioListener to enabled when Kamera is Enabled. Or else It will cause a 3D Spatial Listening Problem.

4. Add Audio and UI

- Create two AudioSources for switch and mode sounds. (attach Clips)
- Create a TextMeshProUGUI for the camera label.
- Create an Image for the fade effect.
- Assign these to the corresponding fields in the script.



5. Play

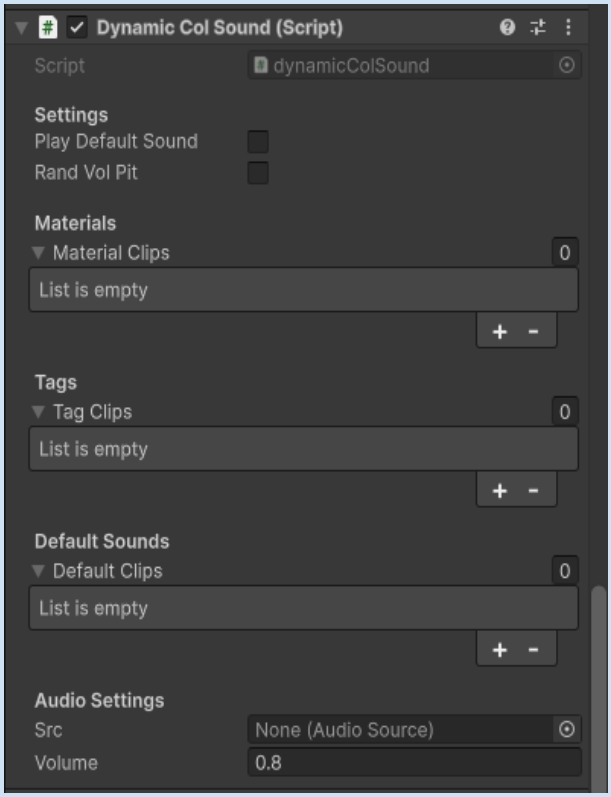
- Press Q and E to switch cameras.
- Press X to toggle orthographic/perspective mode.

Dynamic Collision Sound

This script plays sound effects on collision, using material or tag-based audio clips, with optional random volume and pitch. With default sounds.
Audio source management is handled as well.

1. Attach Script

- Create an empty GameObject.
- Drag and drop the dynamicColSound script onto it.



2. Assign Audio Source

- Create an AudioSource component on the same GameObject.
- Assign this AudioSource to the src field in the script.

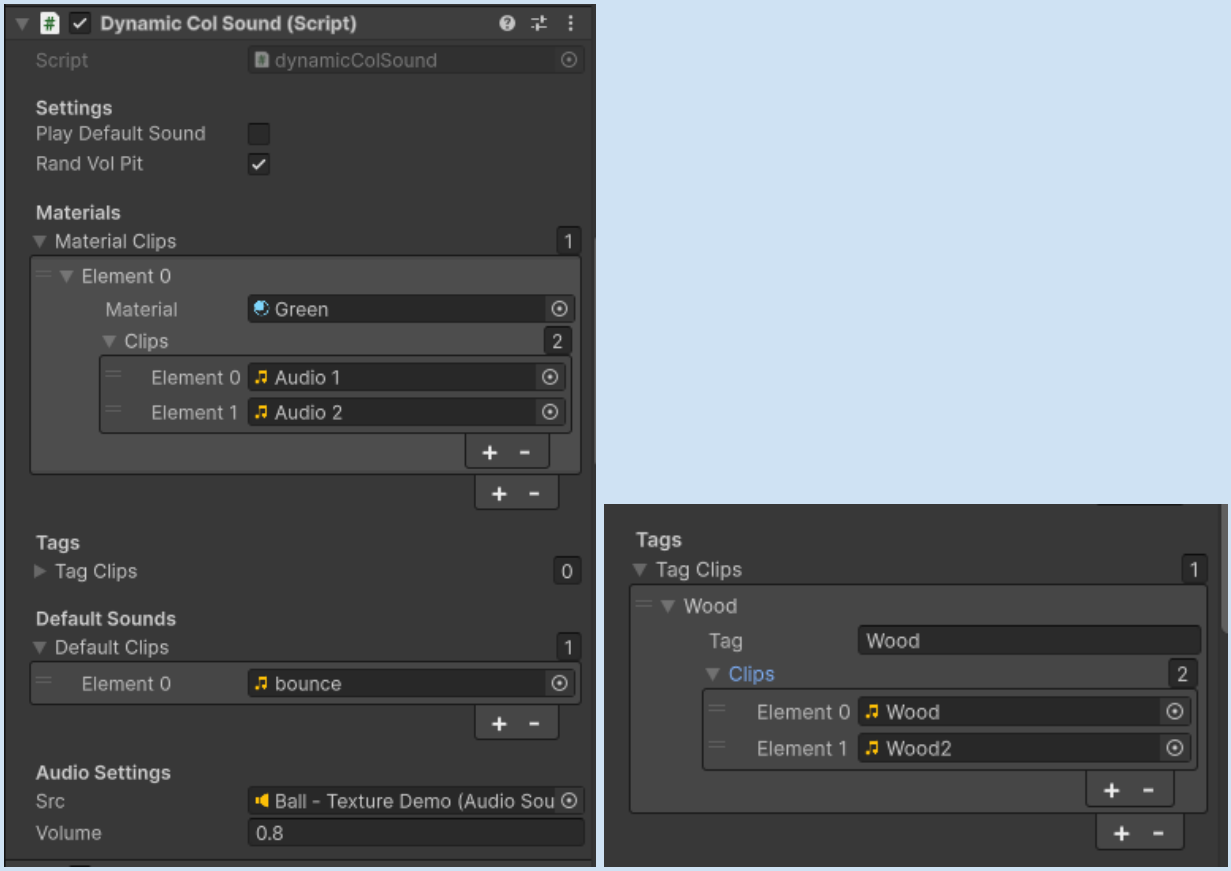
3. Add Default Sounds,

- Create a list of default audio clips.
- Drag and drop these clips into the defaultClips list.

4. Configure Material-Based Sounds.

- Create a list of materials with corresponding audio clips. For example if you have specific material, to play specific clips.

- Drag and drop these materials into the materialClips list.
- Add audio clips for each material.



5. Configure Tag-Based Sounds (if your using Tags)

- Create a list of tags with corresponding audio clips.
- Drag and drop these tags into the tagClips list.
- Add audio clips for each tag.
- If you choose, Rand Vol Pit (Random volume Pitch), each time we get a different sound effect. This will help, when we have less Audio files. Reuse of the same audio Change values here,

```
void SetRandomVolumeAndPitch()
{
    if (randVolPit)
    {
        src.volume = Random.Range(0f, 5f); // sound will be between 0 and 5
        src.pitch = Random.Range(0.5f, 1.5f); //  pitch wil between 0.5 and 1.5
    }
    else
    {
        src.volume = volume;  // normal volume
        src.pitch = 1f; // normal pitch
    }
}
```

6. Test

- Ensure collisions are detected correctly.
- Play the game to check material-based and tag-based sounds.

Note:

- *Adjust playDefaultSound and randVolPit to suit your needs.*
- *You can use volume to set the default volume.*
- *Ensure materials and tags are correctly assigned to GameObjects.*

Camera Zoom Event

*This script controls camera zoom with keyboard/mouse input. It plays zoom sound effects and enables/disables objects at minimum zoom(**When we Zoomed in**).*

1. Attach the Script

- Create an empty GameObject (e.g., "Camera Zoom Controller").
- Drag and drop the Cam_Zoom script onto it.
- Create an AudioSource component on the same GameObject.
- Assign this AudioSource to the audioSource field.

2. Configure Zoom Settings

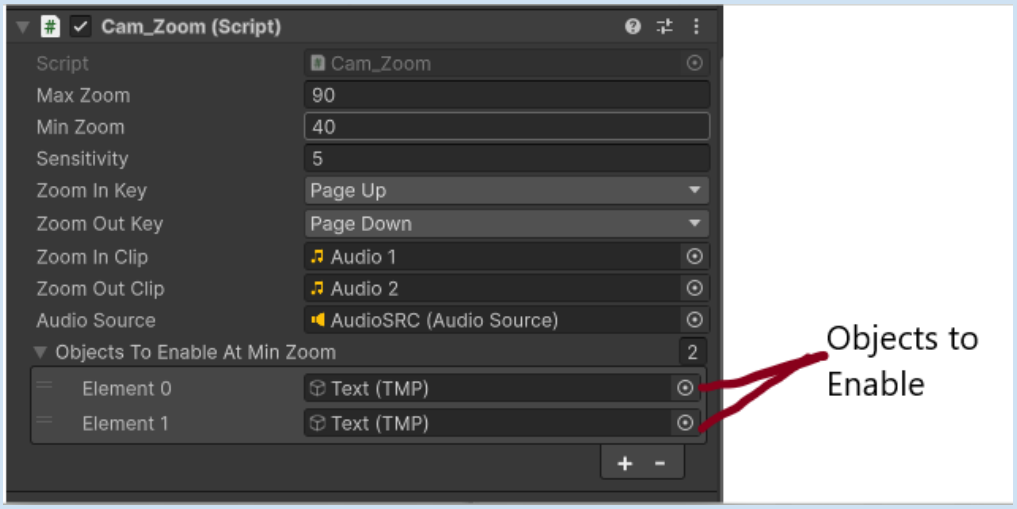
- Set maxZoom, minZoom, and sensitivity to desired values.
- Choose zoomInKey and zoomOutKey (default: PageUp and PageDown).

3. Add Zoom Sounds

- Assign audio clips to zoomInClip and zoomOutClip.

4. Objects to Enable at Min Zoom.

- Drag and drop GameObjects to enable at min zoom into objectsToEnableAtMinZoom.



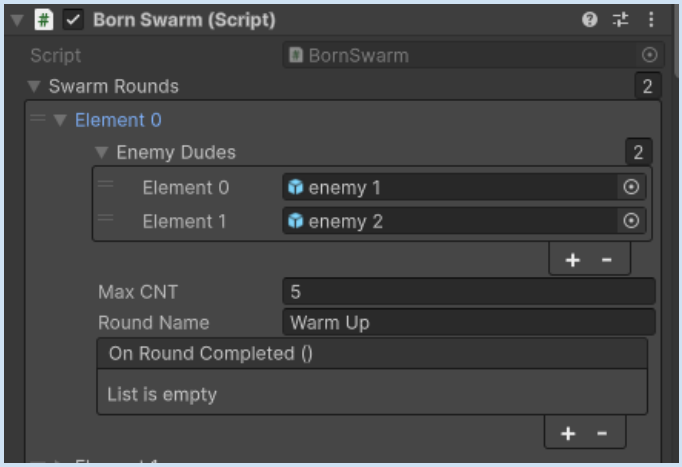
6. Play and Test Zooming

- Use zoomInKey, zoomOutKey, or mouse scroll wheel to zoom.
- Verify objects enable/disable at min zoom.

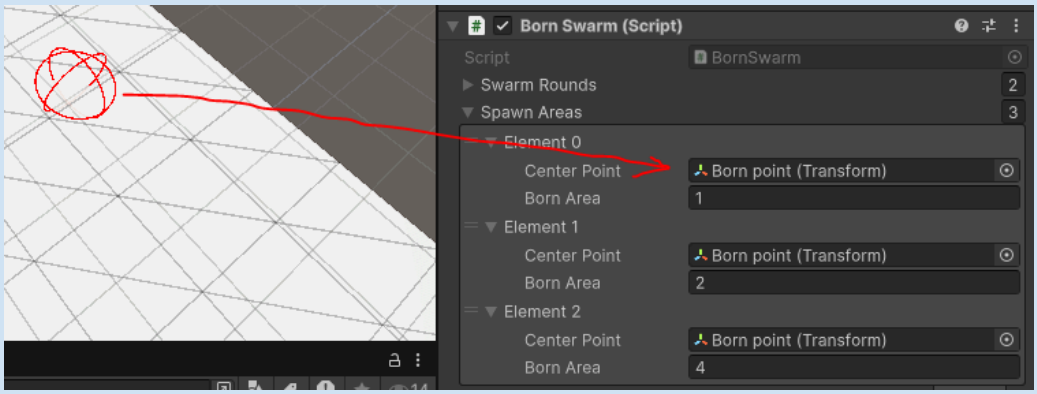
Swarm Spawn system

The script manages enemy Dudes to spawn waves with customizable rounds and spawn areas. It features real-time UI updates and audio. Supports multiple rounds with unique settings. Enables dynamic enemy wave management, each round you can control the event of any script.

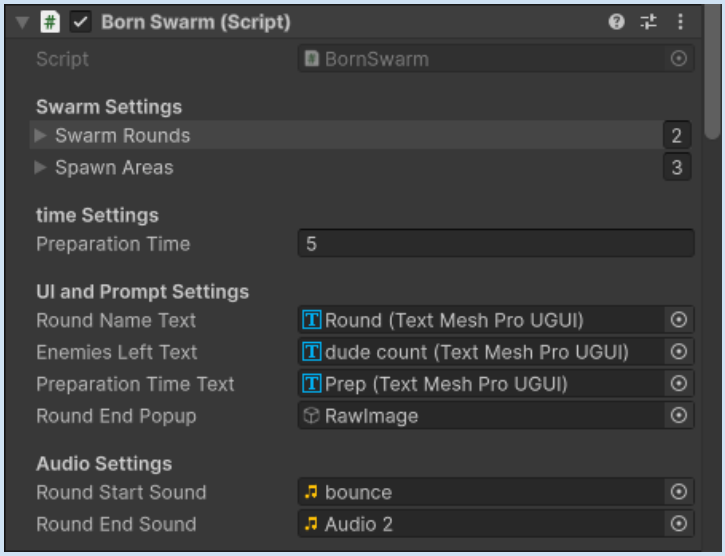
1. Attach Script: Drag and drop the script to an empty GameObject.
2. Add Audio Source: Add an AudioSource to play sounds.
3. Create Swarm Rounds: Make a list of rounds with enemy prefabs, max count, and round name.



4. Set Spawn Areas: Define areas where enemies will spawn.



5. Link UI Texts: Connect text objects to display round info.
6. Add Round Sounds: Add sounds for round start and end.
7. Set Prep Time: Set time before each round starts, like a warm up or setup the defenses.
8. Test Swarm: Check if enemies spawn and die correctly, a Round ends when all the spawned enemies die.
9. Customize Popup: Personalize the end-of-round popup.


Born Swarm (Script)
Script
BornSwarm
Swarm Settings
Swarm Rounds
2
Spawn Areas
3
time Settings
Preparation Time
5
UI and Prompt Settings
Round Name Text
Round (Text Mesh Pro UGUI)
Enemies Left Text
dude count (Text Mesh Pro UGUI)
Preparation Time Text
Prep (Text Mesh Pro UGUI)
Round End Popup
RawImage
Audio Settings
Round Start Sound
bounce
Round End Sound
Audio 2

Pop after time

1. Use the Script

1. Drag and drop the DestroyAfterTime script onto a GameObject which you want to destroy.

2. Set Destruction Time

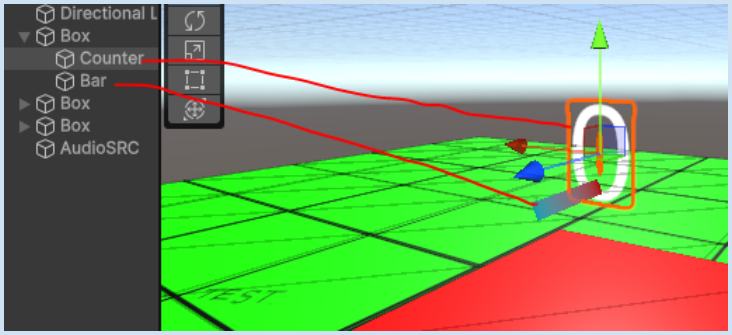
1. Set the time variable to the desired destruction time. (Lifetime)

3. Choose Destruction Method

1. Enable ShrinkDestroy to shrink the object before destruction.
2. Set shrinkDuration for the shrink effect.

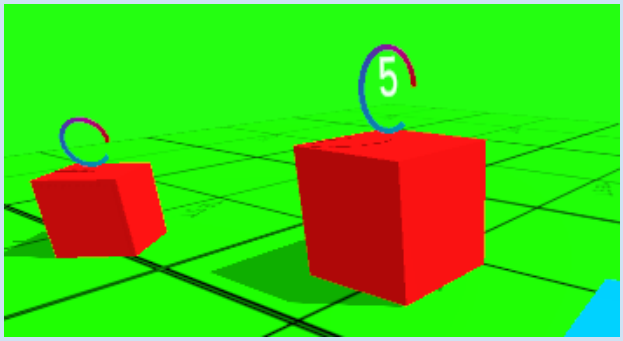
4. Configure UI (If you want to display counter)

1. Assign a TMP_Text component for countdown text (countdownText).
2. Enable Use3Dtext to display countdown text.



5. Configure Bar Graph (If you want to display a Bar)

1. Assign a LineRenderer component for the bar graph (barGraph).
2. Enable UseBarGraph to display the bar graph.

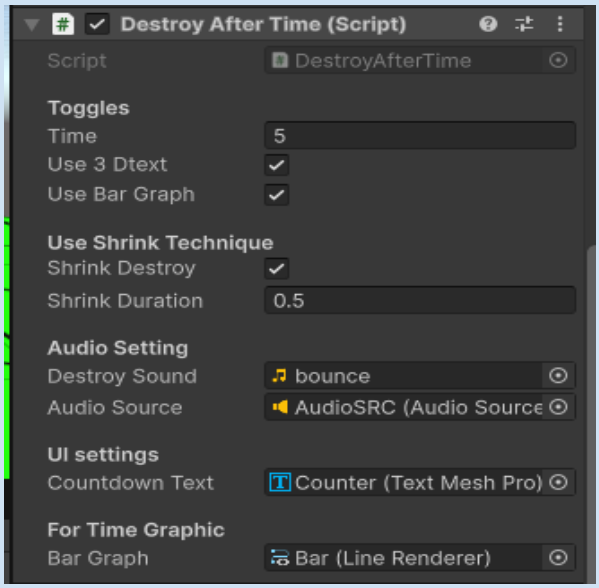


3. If you want to use the SemiCircle, instead of the Circle, Uncomment this line.

```
void UpdateCircularProgress()
{
    int segments = 50;
    float progress = time / initialTime;
    float angleStep = 360.0f / segments;
    //float angleStep = 180.0f / segments;   // if you want to use semi circle.

    barGraph.positionCount = segments + 1;
    for (int i = 0; i <= segments; i++)
    {
        float angle = angleStep * i * progress;
        float radian = Mathf.Deg2Rad * angle;
        float radius = 1.0f;
```

6. Add Destruction Sound

1. Assign an AudioClip for the destruction sound (destroySound).
2. Ensure an AudioSource is attached to the GameObject.



Rotation Controller

1. Attach Script

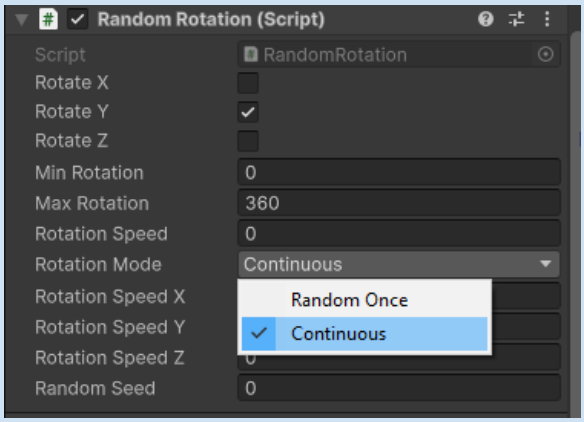
- Drag and drop the RandomRotation script onto a GameObject.

2. Choose Rotation Axes and Range.

- Enable rotateX, rotateY, and/or rotateZ to rotate around respective axes.
- Set minRotation and maxRotation for rotation range.

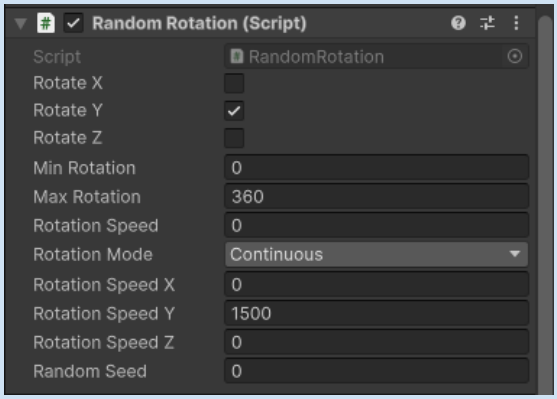
3. Configure Rotation Mode

- Choose RotationMode:
- RandomOnce (applied once at awake) - applies random rotations for a object.
- Continuous (smooth rotation in update) - like if you want to simulate a helicopter or Fan, To showcase any model.



4. Set Rotation Speed (Continuous Mode)

- Set rotationSpeedX, rotationSpeedY, and/or rotationSpeedZ, depends upon your need.



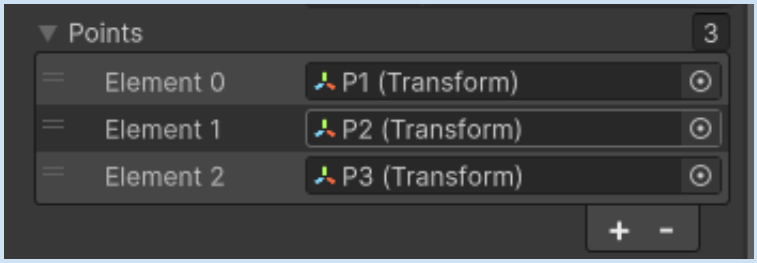
Object Transform (Platform Movement)

1. Use the Script

- Drag and drop the MovingPlatform script onto a GameObject.

2. Set Movement Points

- Create multiple Transforms to serve as movement points.
- Add these Transforms to the points list.



P1 P2 P3 are just a empty gameobjects.

3. Configure Movement Settings

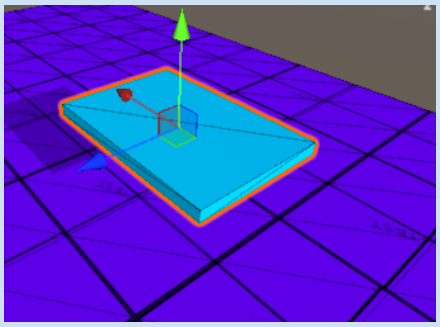
- Set maxSpeed: maximum platform speed.
- Set acceleration: rate of speed increase.
- Set deceleration: rate of speed decrease.

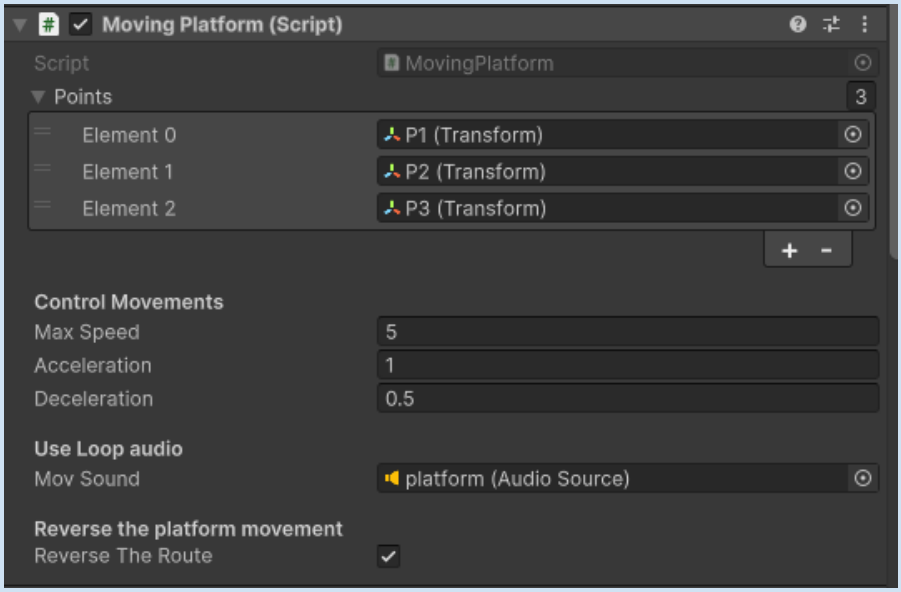
4. Add Looping Audio

- Assign an AudioSource for movement sound.
- Enable Use Loop audio.

5. Reverse Movement

- Enable ReverseTheRoute to reverse platform movement.once it reaches the end of the point. And Attach the script to a simple object.





FlashLight and Car Light system

1. Attach Script

- Drag and drop the TorchLight script onto a GameObject.

2. Link Light Component

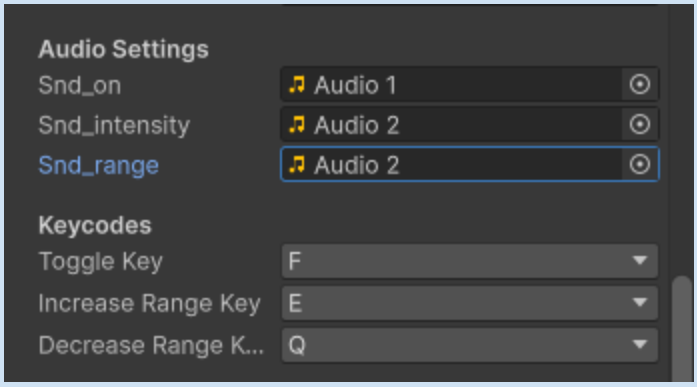
- Assign a Light component to linkedLight.

3. Configure Audio Settings

- Assign audio clips for snd_on, snd_intensity, and snd_range.
- Ensure an AudioSource is attached to the GameObject.

4. Set Keybindings

- Set toggleKey to toggle the lightSet increaseRangeKey and decreaseRangeKey to adjust range.
- You can change to anything in the inspector.



5. Configure Light Settings

- Set minIntensity and maxIntensity for intensity range.
- Set minRange and maxRange for range limits.
- Adjust intensityStep and rangeStep for incremental changes.
- Kontrol Intensity and Range - Determines, whether you are able to control these using keycodes.

CarLight

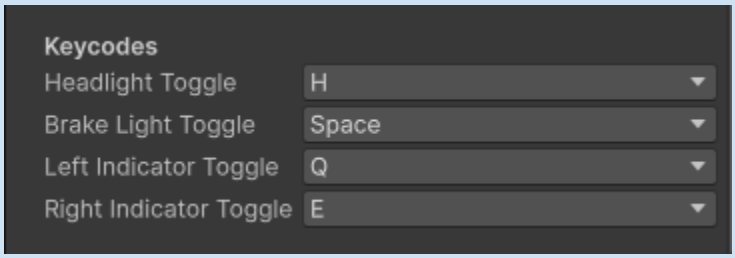
1. Attach Script

- Drag and drop the CarLight script onto a GameObject.

2. Configure Keybindings

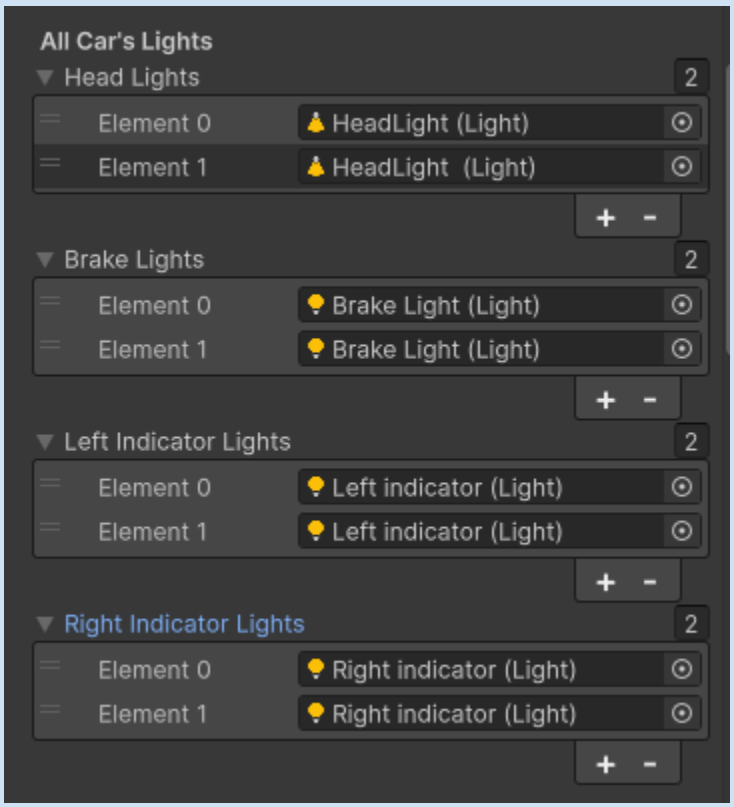
- Set headlightToggle (default: H).
- Set brakeLightToggle (default: B).
- Set leftIndicatorToggle (default: Left Arrow).

- Set rightIndicatorToggle (default: Right Arrow).
- You can Change it in the inspector.



3. Assign Lights

- Add Light components to headLights, brakeLights, leftIndicatorLights, and rightIndicatorLights lists.



Just a Demo. You can add more lights.

4. Audio Settings

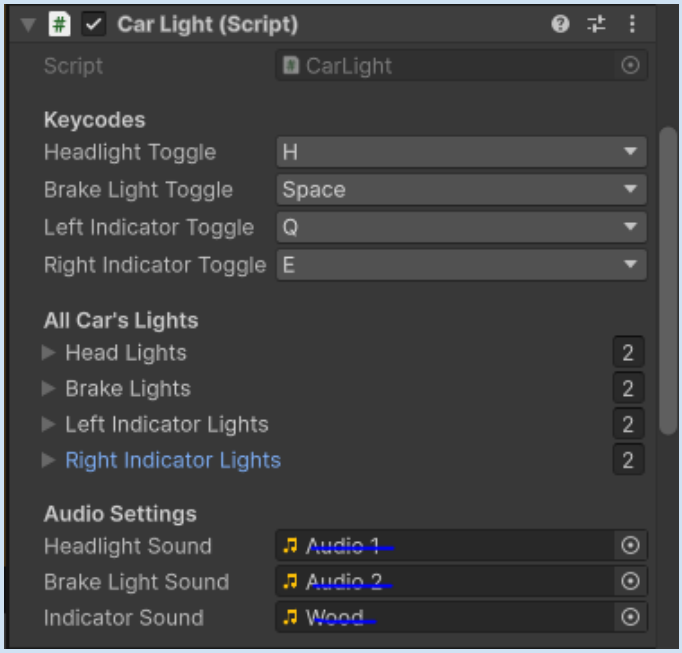
- Assign audio clips for headlightSound, brakeLightSound, and indicatorSound.
- Ensure an AudioSource is attached to the GameObject.

5. Test Light Functionality

- Test headlight toggle.
- Test brake light toggle.
- Test left and right indicator toggles.

6. Adjust Pulse Speed, For indicators.

- Adjust pulseSpeed for faster/slower indicator pulses.



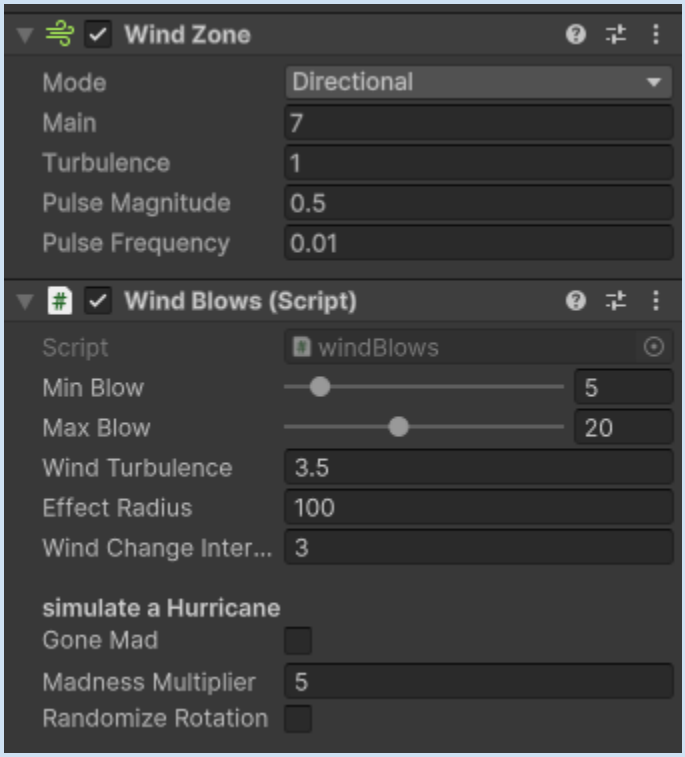

```
private float pulseTimer = 0f;
private void PulseIndicators(List<Light> indicators)
{
    float pulse = Mathf.PingPong(Time.time * pulseSpeed, 2f);
    foreach (Light light in indicators)
    {
        light.intensity = pulse;
    }

    if (Time.time - pulseTimer >= 0.5f)
    {
        pulseTimer = Time.time;
        audioSource.PlayOneShot(indicatorSound);
    }
}
```

To increase and Decrease the Pulse threshold. (Red)
To increase and Decrease the Pulse Speed (Blue)

Dynamic Wind System

1. Use Script

- Drag and drop the WindBlows script onto a WINDZONE.



2. Configure Wind Settings

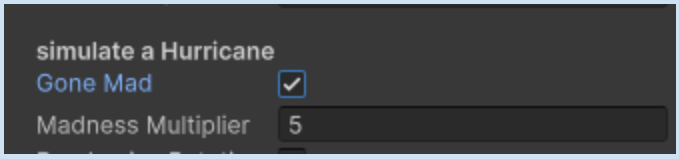
- Set minBlow and maxBlow for wind strength range.
- Set windTurbulence for wind variation.
- Set effectRadius for wind effect area.

3. Configure Audio

- Assign an AudioSource component to the GameObject.
- Ensure windAudioSource is linked to the AudioSource.

4. Enable Madness Mode

- Enable GoneMad for extreme wind behavior.
- If you want to simulate a Hurricane or Do something Crazy.

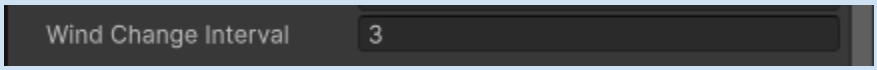


5. Randomize Rotation

- Enable randomizeRotation to randomly change wind direction

```
howLongToChange = Time.time + windChangeInterval;
```

This makes the wind to change direction for certain interval.



Default change interval is 3 secs. We can change to max secs if you want.

- ***APEX***