

Switching Sources - Rotating PIP



The description is relevant for software version [2.9.116.100](#) and later. The project has been prepared in Godot version 4.1.1.

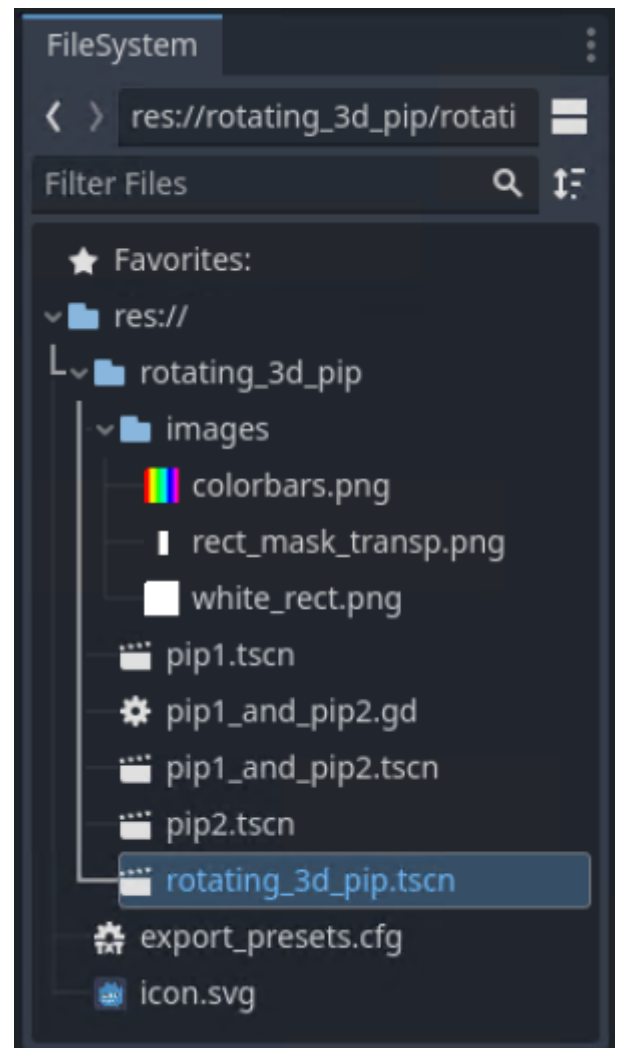
Description of the Godot demo project that implements switching between two sources during a speaker's speech in a news studio.

GodotDemo_election3d_map_and_bars.zip contains sub folders:

- **Source** - source project files to open in Godot Editor.
- **Compiled** - folder with assembled project in zip format, ready to use with Skylark Godot Engine module.
- **Parts** - additional files required for project operation:
- **toMedia** - files from this folder must be placed in the server's first local media database that has the Godot module installed.



Project Files



The "GodotDemo_rotating_3d_pip" project has 4 scenes:

- **pip1.tscn** - 2D scene with a player playing the signal from the first line.
- **pip2.tscn** - 2D scene with a player playing the signal from the second line.
- **pip1_and_pip2.tscn** - 3D scene implementing overlay of the first and second line signals to the opposite sides of the 3D box. The scene also has the box rotation animation.
- **rotating_3d_pip.tscn** - the main project scene displayed at startup. The left part of the scene outputs the signal from the studio with the host, and the right one outputs the scene with signals from pip1_and_pip2.tscn external lines.

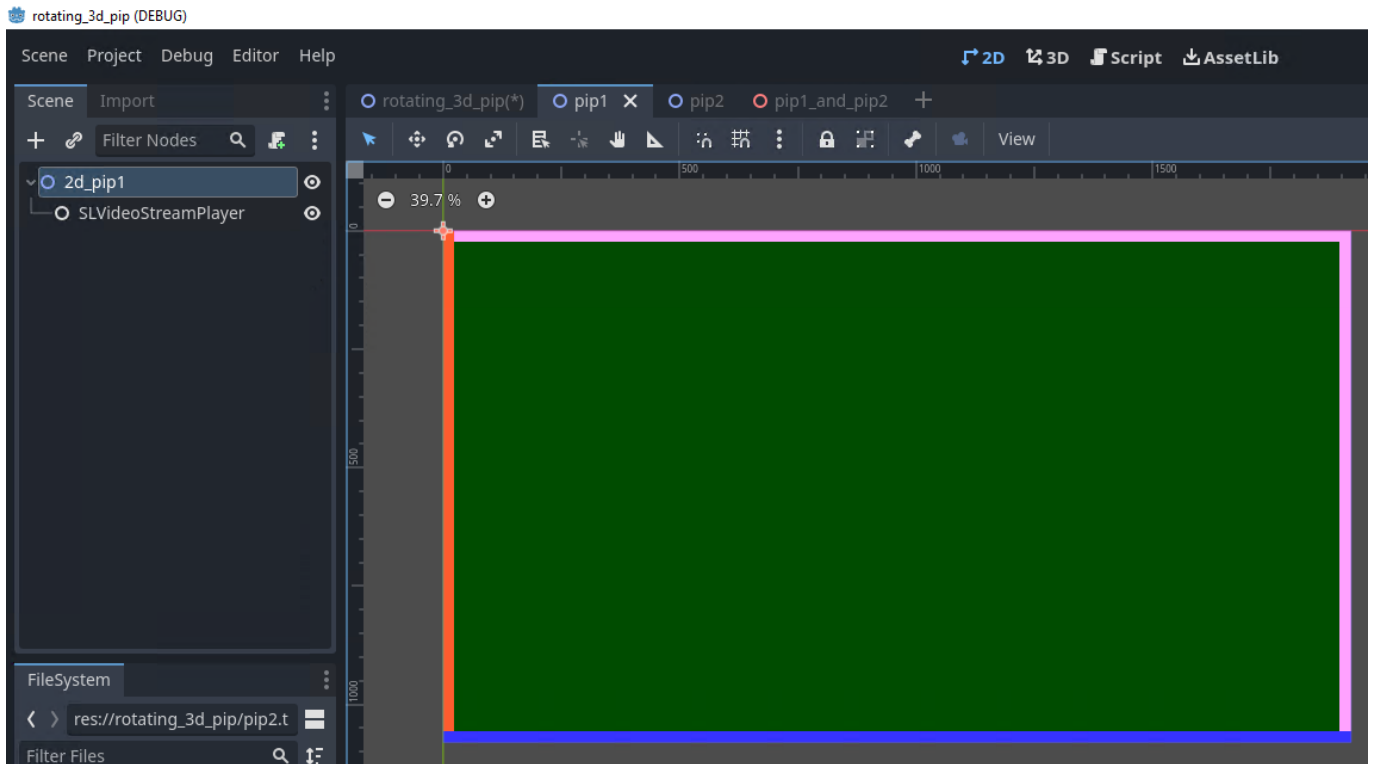
The project also includes:

- **colorbars.png** - stub for easy scaling and positioning of the Sprite3D node displaying signals from external lines.
- **rect_mask_transp.png** - mask for cutting the news host from the FullHD signal.
- **white_rect.png** - backing for the area with the news host.
- **pip1_and_pip2.gd** - GD Script file for the pip1_and_pip2.tscn scene.

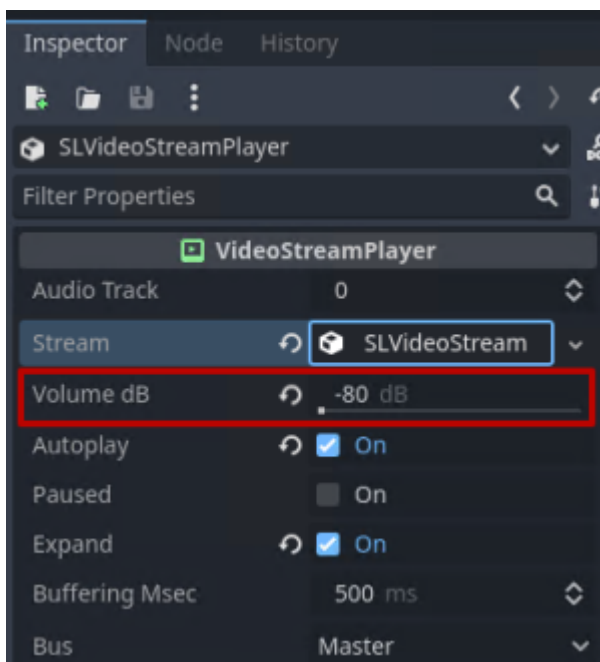
Let's look closer at the content of each scene.

Scenes pip1.tscn and pip2.tscn

2D scenes containing SLVideoStreamPlayer nodes that play signals from external lines. In the project, signals are simulated by playing the “Transportation HD” and “CityViews2 HD” files from the first local media database of the Skylark SL NEO server. To output real signals from external lines, use [Capture modules](#) and Live-clips created in the media database (“New→Local Stream...” menu in the [Media Browser window](#)).

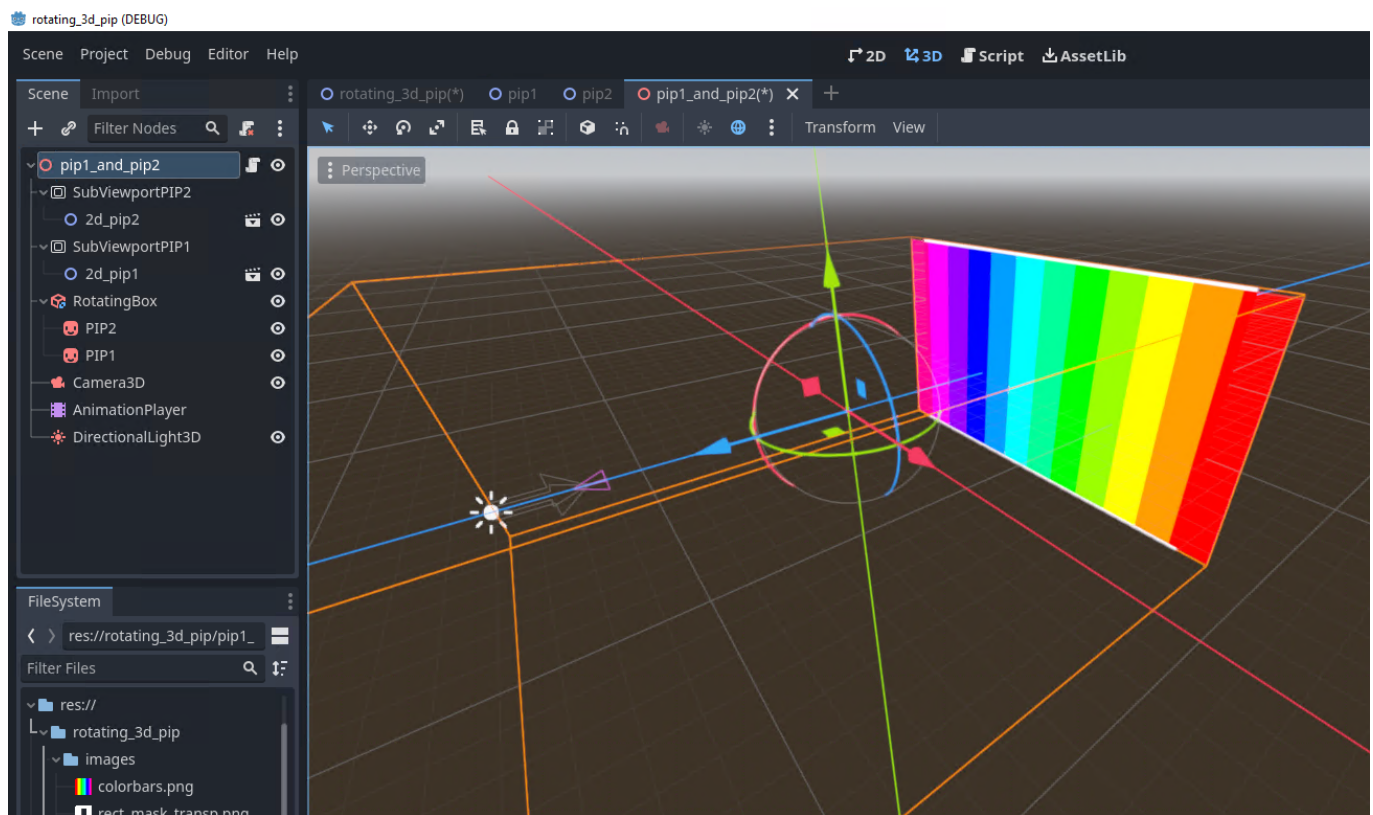


The players’ sound is muted, given that the host’s live speech is in progress. If, however, the sound must be output from the PIP being switched, use the AnimationPlayer node to control the “Volume dB” parameter.



Pip1_and_pip2.tscn Scene

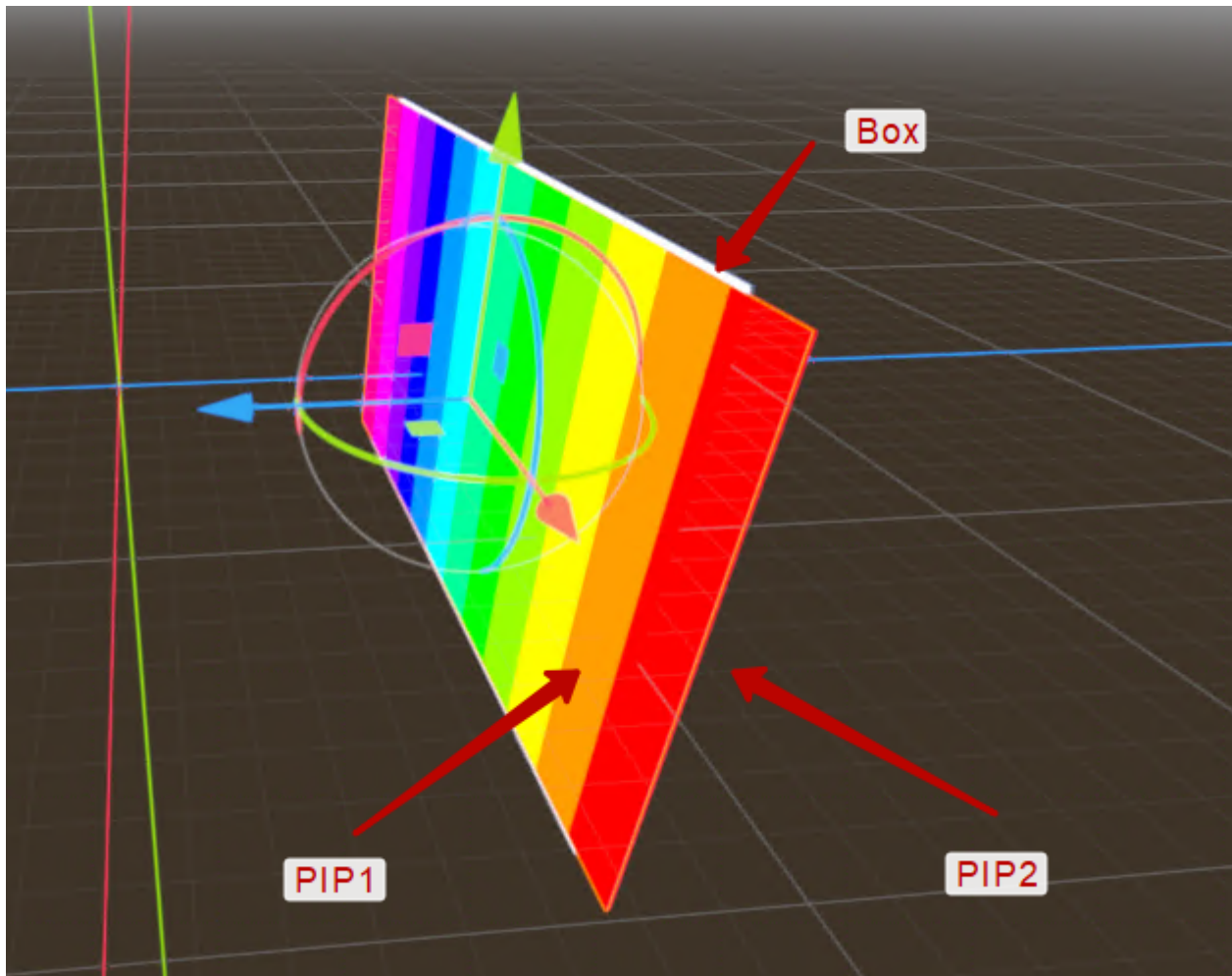
A 3D scene that provides overlaying first and second line signals on opposite sides of the 3D box and the box rotation animation.



The RotatingBox (CSGBox3D) node is the basis of the composition, performing two functions at once:

- displaying the white border around the image in static.
- creating the 3D basis during the rotation animation.

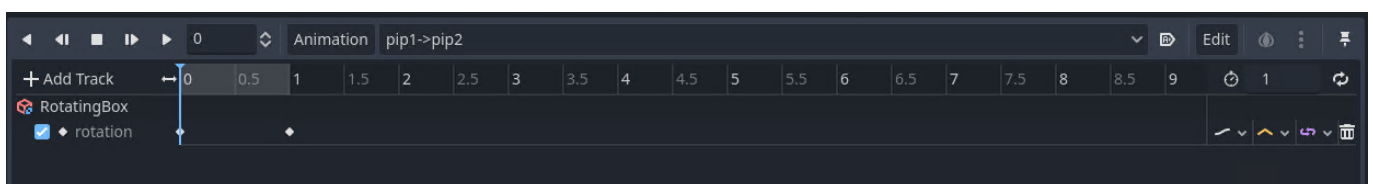
The front side of the box has the PIP1 (Sprite3D) node, and the back side has the PIP2 (Sprite3D) one. A y-axis transformation equal -180 is applied to PIP2 to ensure that the output image has the correct orientation after the box is rotated. Initially, PIP1 and PIP2 display the colorbars.png file, useful for scaling and positioning the Sprite3D node in space. When the scene is rendered, the texture will be replaced with the signal from the studio via GD Script.



PIP1 and PIP2 are embedded in RotatingBox, so animating its rotation will be enough for all objects to rotate.

The AnimationPlayer (AnimationPlayer) node has two animations:

- **pip1→pip2** - transition from the first line to the second line.
- **pip2→pip1** - transition from the second line to the first one.



The corresponding animation is called from GD Script upon receiving a user command.

SubViewportPIP1 (SubViewport) and SubViewportPIP2 (SubViewport) nodes provide capturing pip1.tscn and pip2.tscn scenes.

The DirectionalLight3D (DirectionalLight3D) node is used to create the scene lighting.

The Camera3D (Camera3D) node is used to create a perspective view of the scene objects.

The root node of the scene has a script attached that ensures assigning textures properly and calling the animation.

[pip1_and_pip2.gd](#)

```
extends Node

# Called every frame. 'delta' is the elapsed time since the previous
# frame.
func _ready():
    var texture = $SubViewportPIP1.get_texture()
    var texture2 = $SubViewportPIP2.get_texture()

    # Assigning textures to Sprite3D nodes. Allows to output the result
    # of rendering scenes loaded to Viewport
    $RotatingBox/PIP1.texture = texture
    $RotatingBox/PIP2.texture = texture2

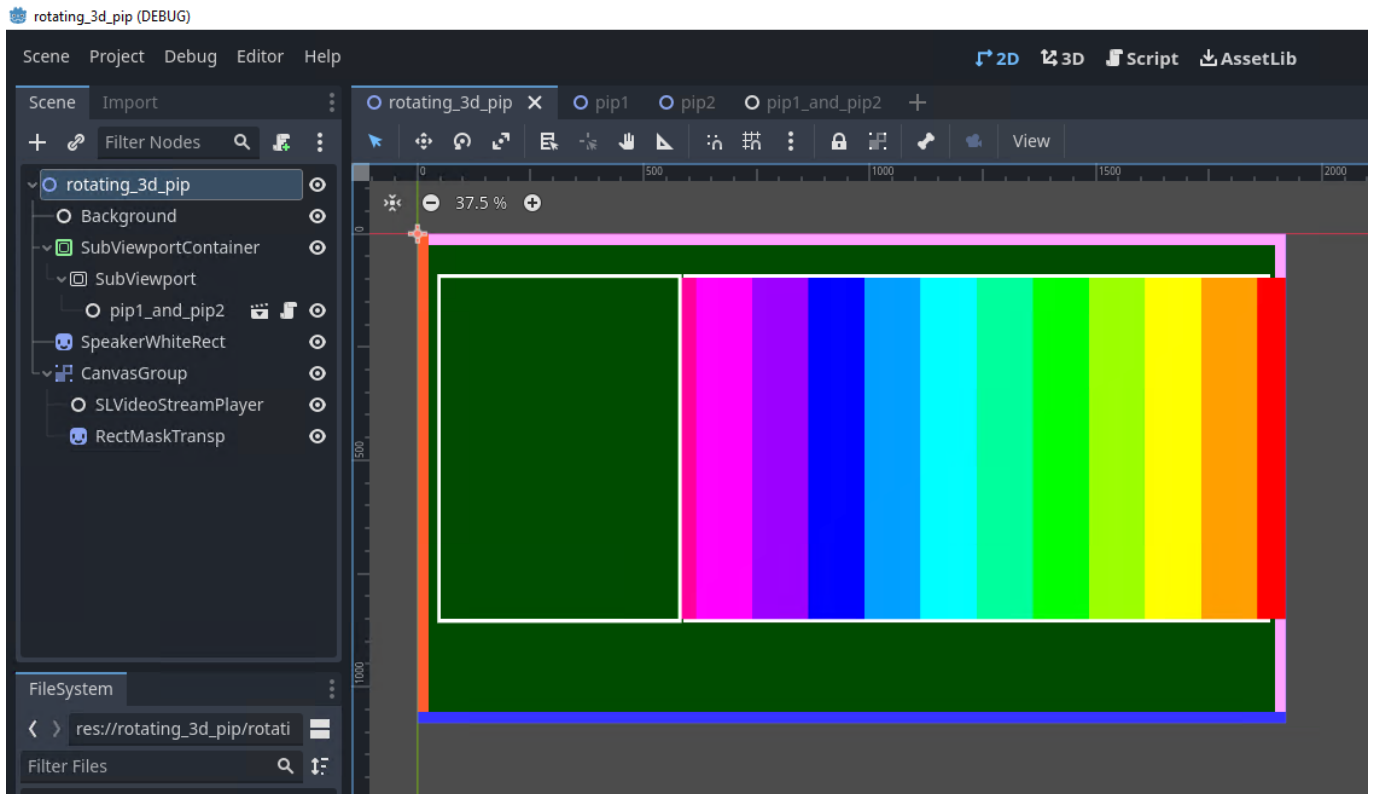
# Called when user input is detected in the Godot window
func _input(event):
    if event.is_input_event():
        print(event)

    # If the combination corresponding to the cam1 Action is pressed in
    # the Godot window
    if event.is_action_pressed("cam1"):
        $AnimationPlayer.play("pip1->pip2") # Start animation
        "pip1->pip2"

    # If the combination corresponding to the cam2 Action is pressed in
    # the Godot window
    if event.is_action_pressed("cam2"):
        $AnimationPlayer.play("pip2->pip1") # Start animation
        "pip2->pip1"
```

Rotating_3d_pip.tscn Scene

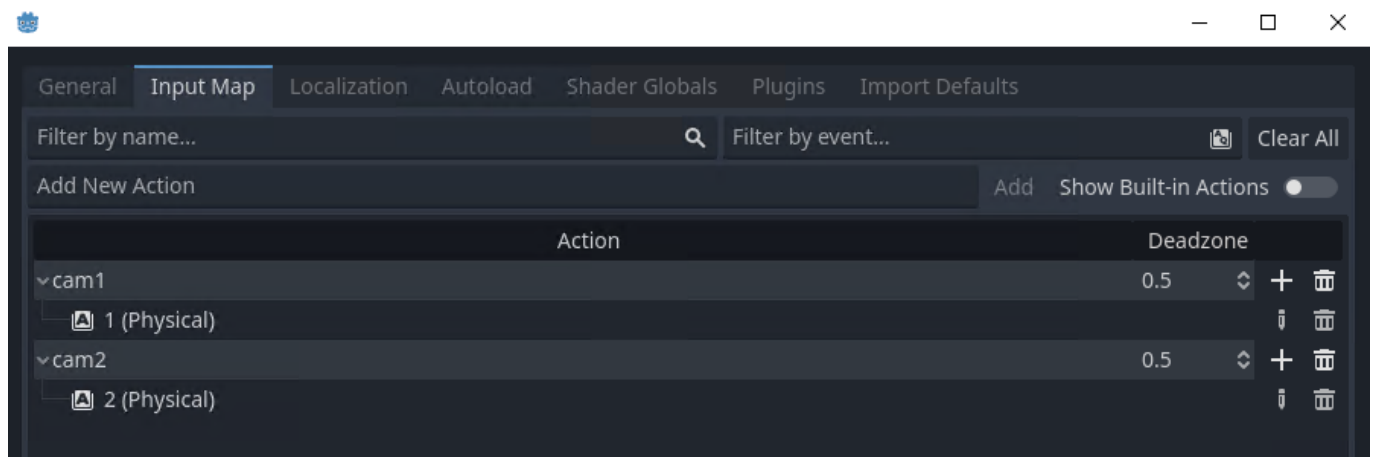
The main project scene displayed at startup. The left part of the scene outputs the signal from the studio with the host, and the right one outputs the scene with signals from pip1_and_pip2.tscn external lines.



- The Background (SLVideoStreamPlayer) node provides playback of the rotating_3d_pip_background backing clip.
- The SubViewportMap (SubViewport) node provides capture of the pip1_and_pip2 scene, and the SubViewportContainer node provides its direct output without applying textures.
- The SpeakerWhiteRect (Sprite3D) displays the white_rect.png file used to visualize the border in the signal output area of the news host.
- The CanvasGroup (CanvasGroup) node groups child nodes:
- The SLVideoStreamPlayer (SLVideoStreamPlayer) node outputs the FullHD signal from the studio. In the project, the signal is simulated by playing the "Speaker1_center_keying" file.
- The RectMaskTransp (Sprite3D) node outputs the rect_mask_transp.png file for mask cutting of the central area of the studio signal. Enabling the mask mode is implemented by the RectMaskTransp→Inspector→CanvasItem→Material→Blend Mode = Multiply parameter.

Management

"1" and "2" keys are assigned in the project settings to launch the cam1 and cam2 Actions.



The project's program logic only serves to demonstrate functionality. In real-world projects, keystroke monitoring is added to ensure the correct order and timeliness of switches.

From:

<http://wiki.skylark.tv/> - **wiki.skylark.tv**

Permanent link:

http://wiki.skylark.tv/graphics/3d_godot/samples/rotating_3d_pip

Last update: **2023/07/25 16:07**

