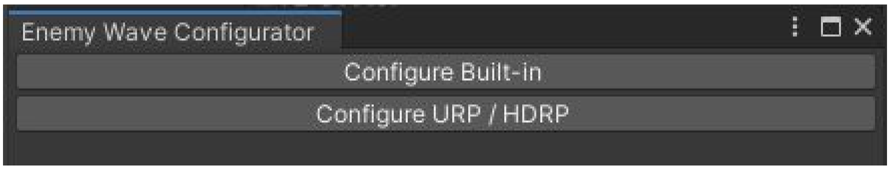
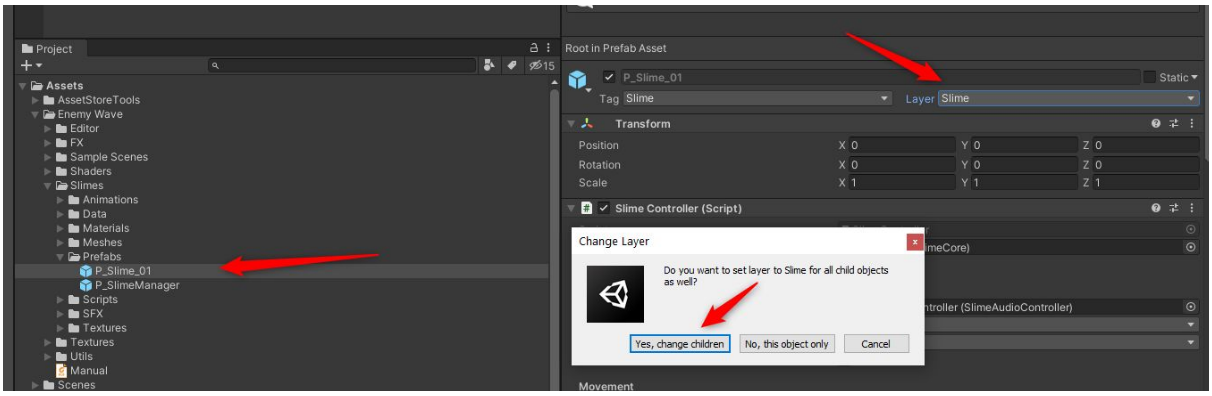
Get Started

1. Open ***Enemy Wave Configurator*** (***Tools*** → ***Enemy Wave Configurator***) and select your render configuration to set up materials correctly.



2. Go to ***Enemy Wave*** → ***Slimes*** → ***Prefabs*** folder. On ***P_Slime_01*** prefab, select a layer and apply it for all child objects.

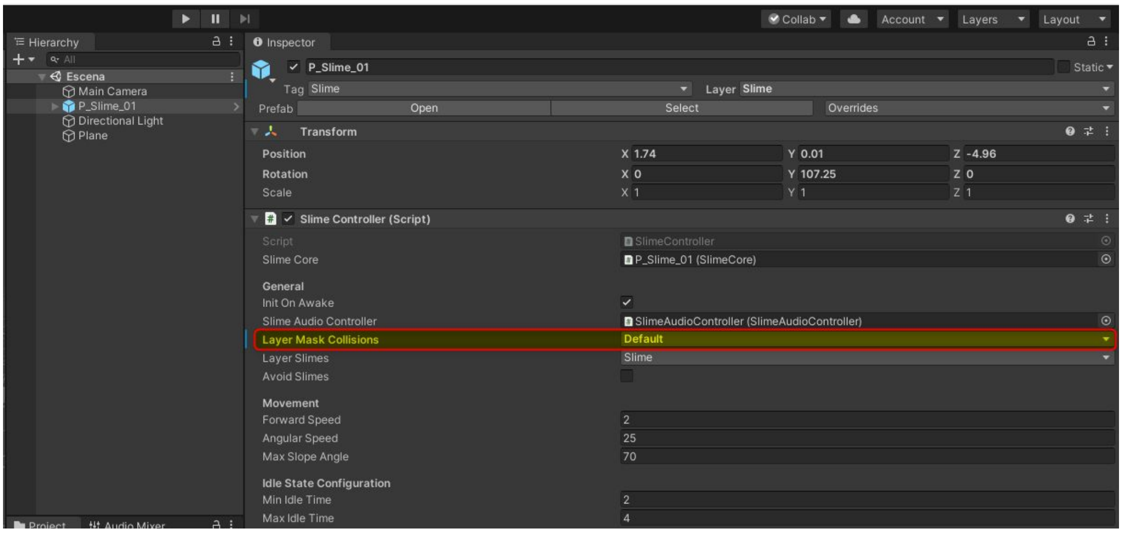


P_Slime_01
Static
Tag Slime
Layer Slime
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Change Layer

Do you want to set layer to Slime for all child objects as well?

3. We need to configure what layers will be used to detect collisions with the environment. For that, we set the public property ***Layer Mask Collisions*** on ***Slime Controller*** component (This component contains almost everything you can interact with as a user)

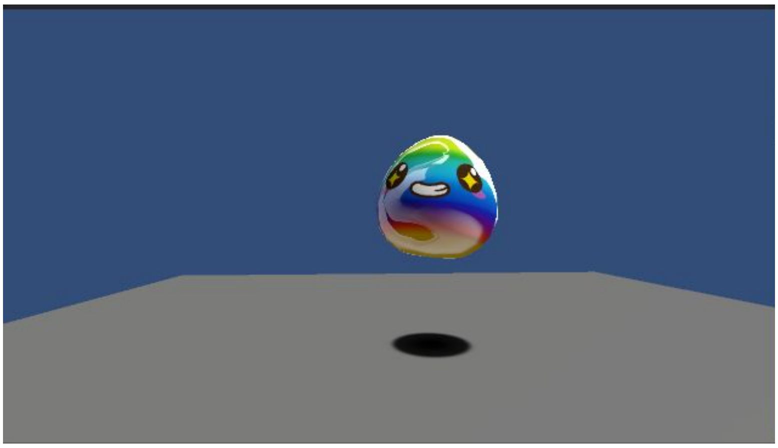


P_Slime_01
Static
Tag Slime
Layer Slime
Prefab

Position X 1.74 Y 0.01 Z -4.96
Rotation X 0 Y 107.25 Z 0
Scale X 1 Y 1 Z 1

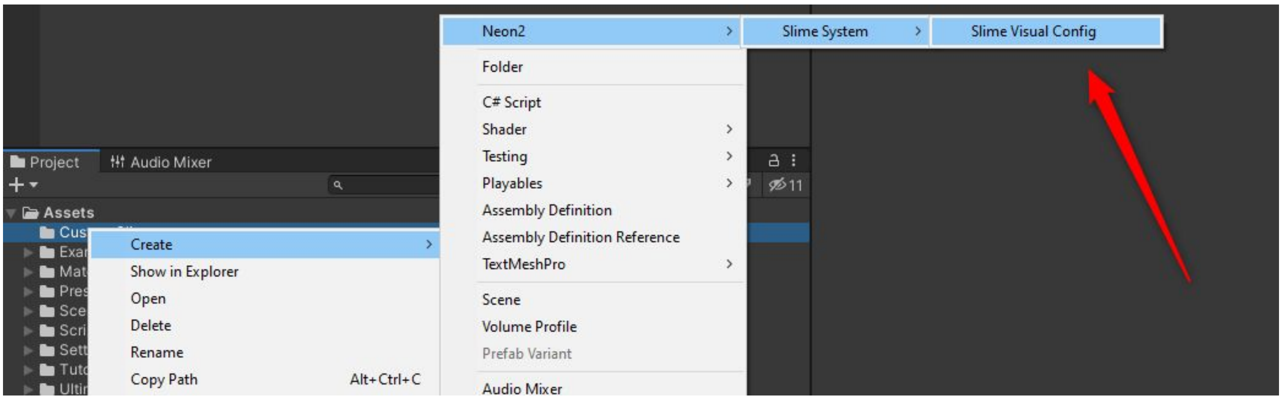
Script SlimeController
Slime Core P_Slime_01 (SlimeCore)
General
Init On Awake ☑
Slime Audio Controller SlimeAudioController (SlimeAudioController)
Layer Mask Collisions Default
Layer Slimes Slime
Avoid Slimes □
Movement
Forward Speed 2
Angular Speed 25
Max Slope Angle 70
Idle State Configuration
Min Idle Time 2
Max Idle Time 4

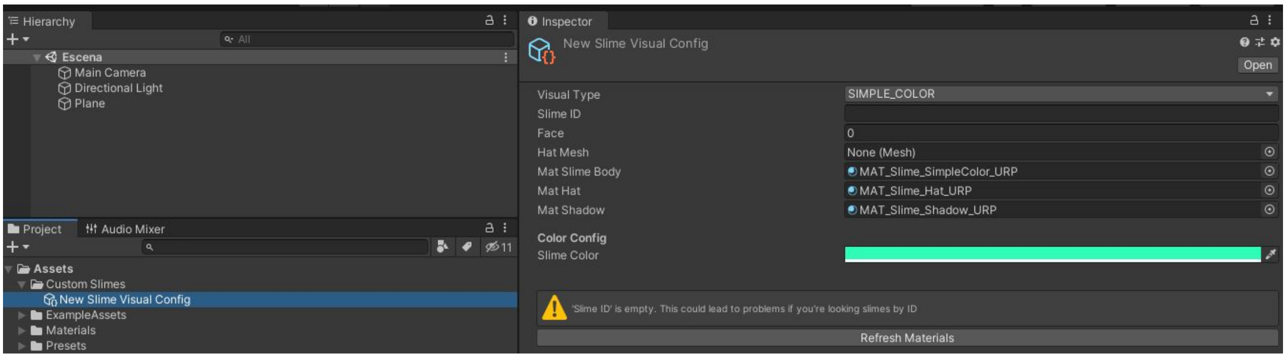
4. Once we have done all previous steps, we can drag the slime on our scene and click Play and… voilá! We have achieved a slime moving freely on our scenario (Don't forget to configure correctly all colliders and layers in your environment so the slime can move through the scene without problems)



Creating custom slimes

To create a new slime visual configuration you have to create an ***SlimeVisualConfig***. To do that, go anywhere on the project and: ***Right Click*** → ***Create*** → ***Neon2*** → ***Slime System*** → ***Slime Visual Config***.





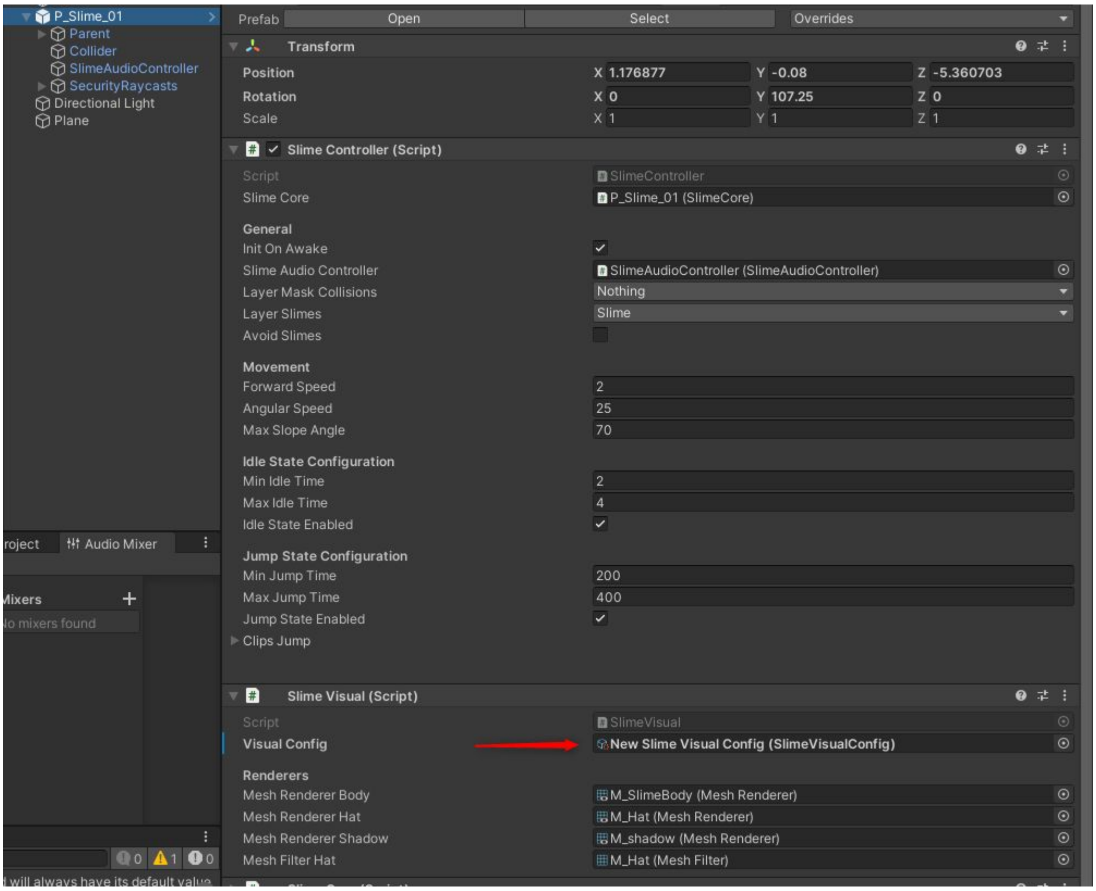
Visual Type: SIMPLE_COLOR
Slime ID:
Face: 0
Hat Mesh: None (Mesh)
Mat Slime Body: MAT_Slime_SimpleColor_URP
Mat Hat: MAT_Slime_Hat_URP
Mat Shadow: MAT_Slime_Shadow_URP

Color Config
Slime Color: 🟩

'Slime ID' is empty. This could lead to problems if you're looking slimes by ID

Refresh Materials

To apply the new configuration to the slime, reference the new file on the field ***Visual Config***, on ***Slime Visual*** component.



Prefab Overrides

Transform
Position X 1.176877 Y -0.08 Z -5.360703
Rotation X 0 Y 107.25 Z 0
Scale X 1 Y 1 Z 1

☑ Slime Controller (Script)
Script SlimeController
Slime Core P_Slime_01 (SlimeCore)

General
Init On Awake ☑
Slime Audio Controller SlimeAudioController (SlimeAudioController)
Layer Mask Collisions Nothing
Layer Slimes Slime
Avoid Slimes □

Movement
Forward Speed 2
Angular Speed 25
Max Slope Angle 70

Idle State Configuration
Min Idle Time 2
Max Idle Time 4
Idle State Enabled ☑

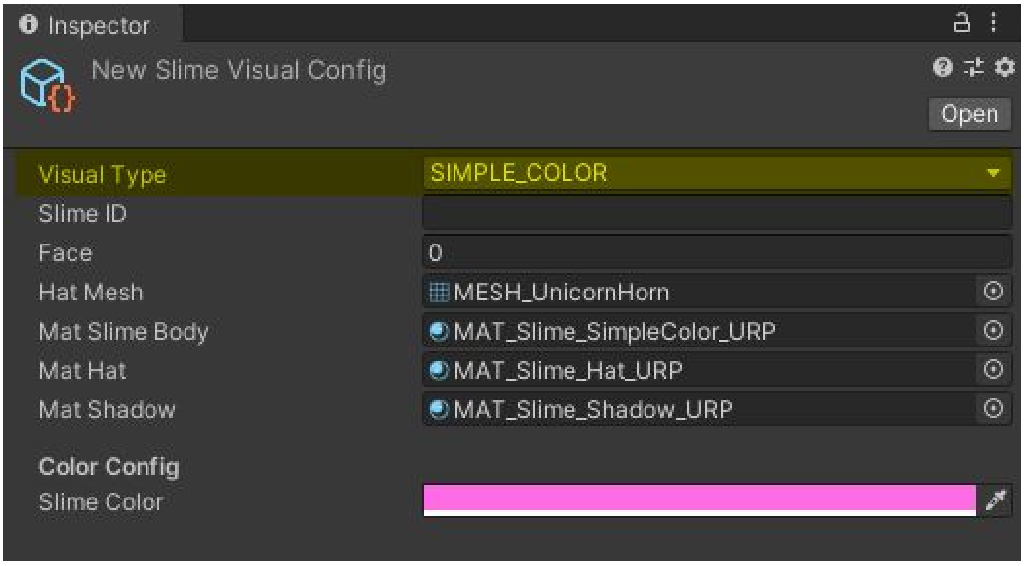
Jump State Configuration
Min Jump Time 200
Max Jump Time 400
Jump State Enabled ☑
Clips Jump

Slime Visual (Script)
Script SlimeVisual
Visual Config New Slime Visual Config (SlimeVisualConfig)

Renderers
Mesh Renderer Body M_SlimeBody (Mesh Renderer)
Mesh Renderer Hat M_Hat (Mesh Renderer)
Mesh Renderer Shadow M_shadow (Mesh Renderer)
Mesh Filter Hat M_Hat (Mesh Filter)

Types of slimes

There are 4 types of slimes: **SIMPLE_COLOR**, **BICOLOR**, **TEXTURED_RAINBOW_01**, **TEXTURED_RAINBOW_02**. You can set the slime type modifying ***Visual Type*** property.

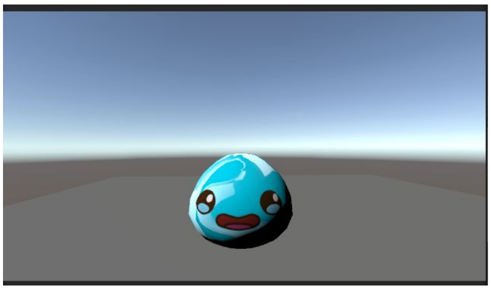


New Slime Visual Config

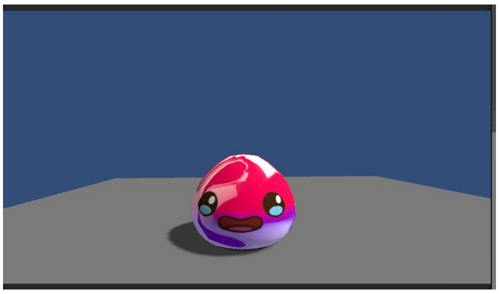
Visual Type SIMPLE_COLOR
Slime ID
Face 0
Hat Mesh MESH_UnicornHorn
Mat Slime Body MAT_Slime_SimpleColor_URP
Mat Hat MAT_Slime_Hat_URP
Mat Shadow MAT_Slime_Shadow_URP

Color Config
Slime Color 🟪

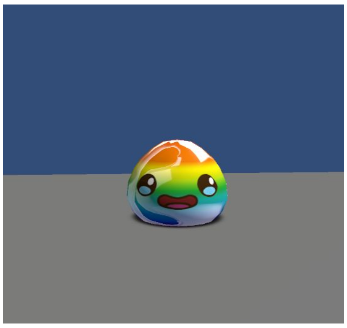
- **SIMPLE COLOR**
  Slimes with only one color. You can change the color setting ***Slime Color*** property



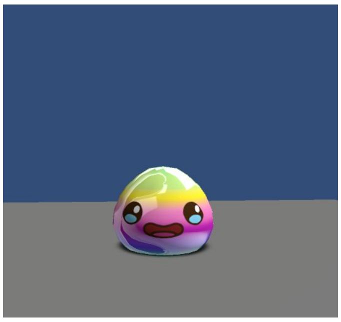
- **BICOLOR**
  Slimes with two colors. Modify ***Slime Color*** to change the top color and ***Slime Second Color*** to change bottom color.



- **TEXTURED_RAINBOW_01**

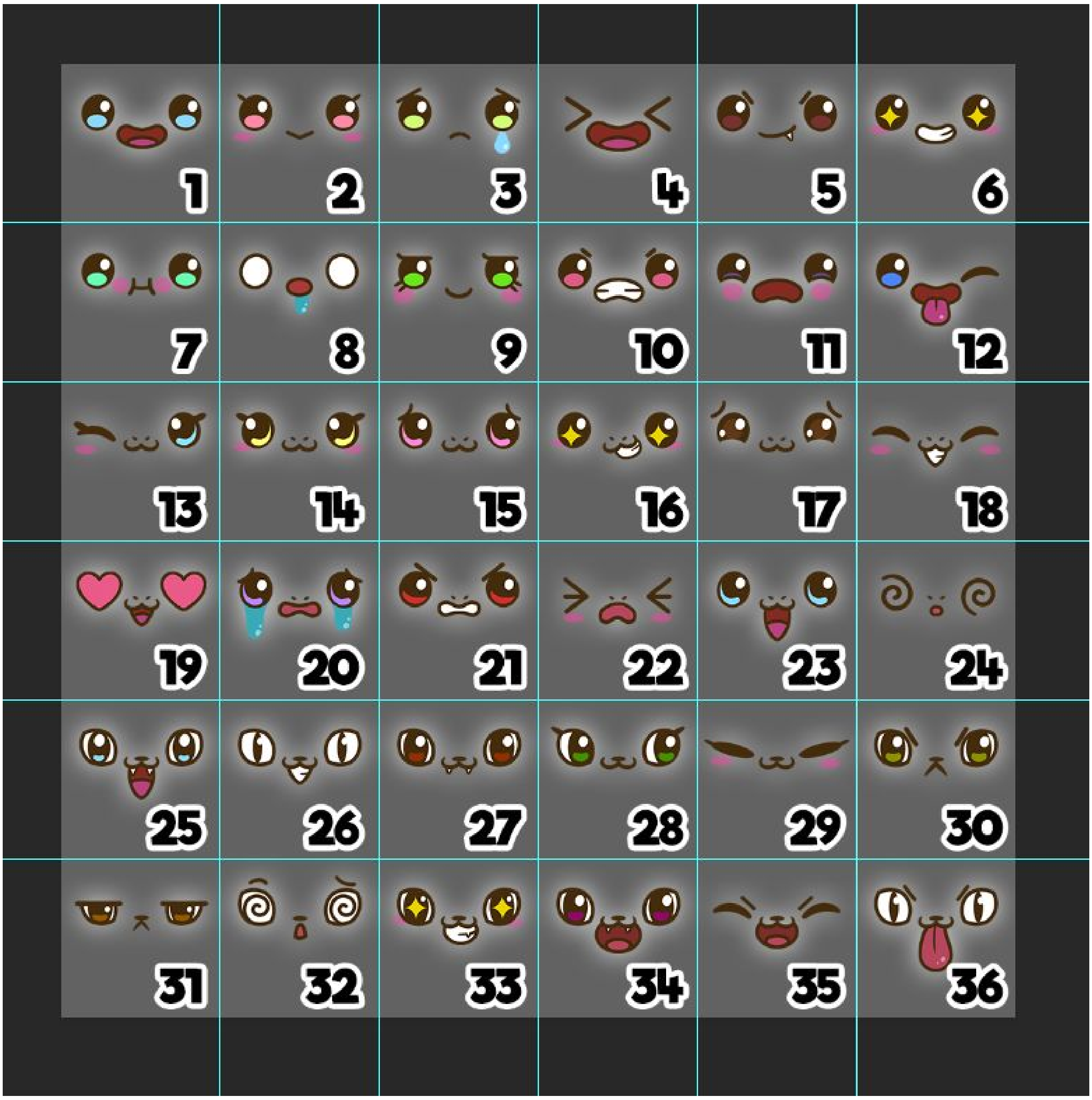


- **TEXTURED_RAINBOW_02**



Faces

There are 36 faces availables. Faces are packaged into a texture atlas as shown below.

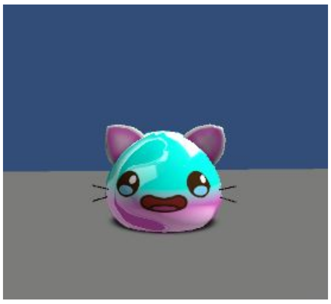


To select a face, insert in ***Face*** property the number of the face that you want.

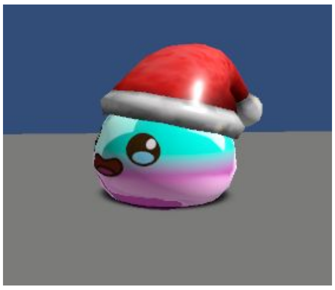
Hats

Slimes can wear a hat. There are 4 hats availables that can be found at ***Enemy Wave* → *Slimes* → *Meshes* → *Hats***. To use them, you have to reference the hat mesh in the field ***Hat Mesh***.

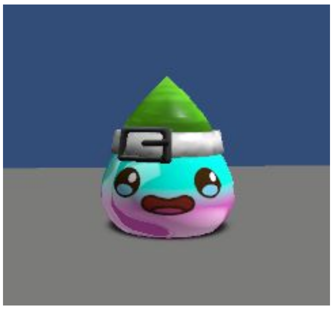
- MESH_Cat



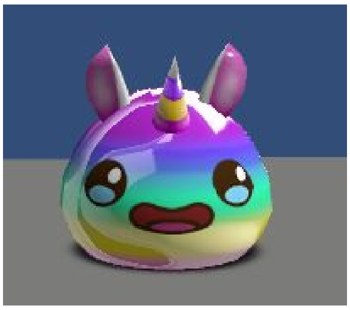
- MESH_Noel



- MESH_Dwarf



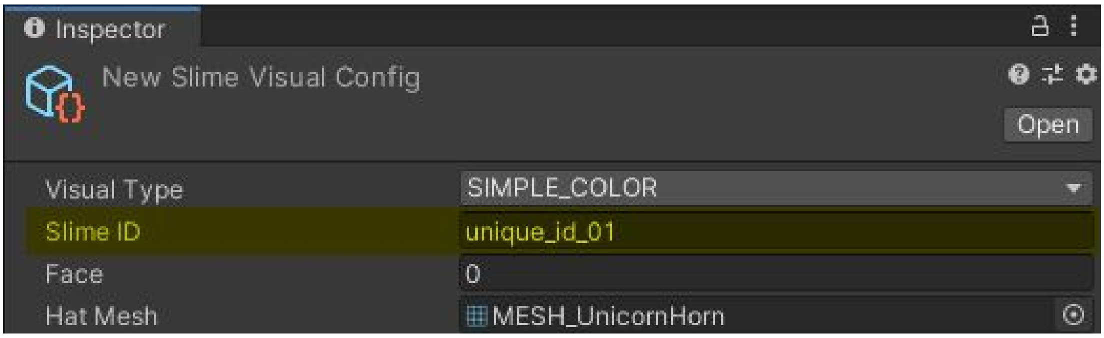
- MESH_UnicornHorn



Using custom slimes via scripting

To access the new slime via scripting you have to:

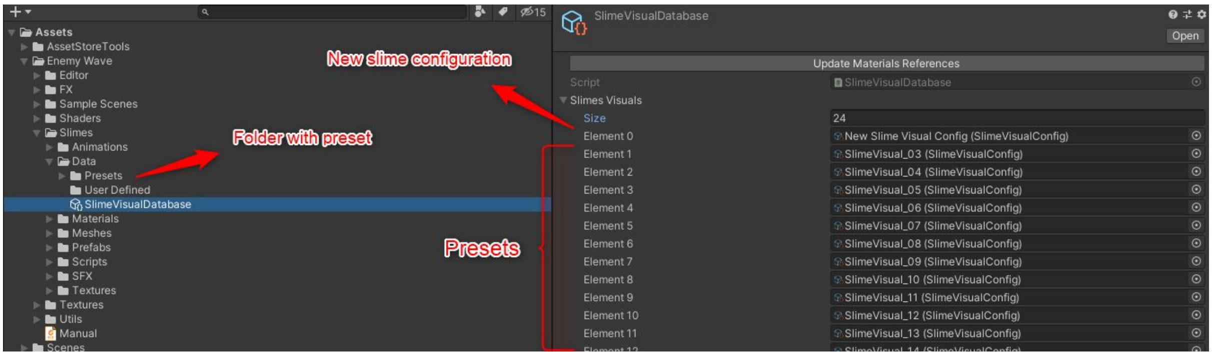
- Setup ***Slime ID***. The ID has to be unique.



New Slime Visual Config

Visual Type: SIMPLE_COLOR
Slime ID: unique_id_01
Face: 0
Hat Mesh: MESH_UnicornHorn

- Reference the file in ***SlimeVisualDatabase*** (***Enemy Wave*** → ***Slimes*** → ***Data***). By default, all preconfigured presets are referenced. You can keep them or remove them, it's up to you.



New slime configuration

Folder with preset

Presets

SlimeVisualDatabase

Script: SlimeVisualDatabase

Slimes Visuals

Size	24
Element 0	New Slime Visual Config (SlimeVisualConfig)
Element 1	SlimeVisual_03 (SlimeVisualConfig)
Element 2	SlimeVisual_04 (SlimeVisualConfig)
Element 3	SlimeVisual_05 (SlimeVisualConfig)
Element 4	SlimeVisual_06 (SlimeVisualConfig)
Element 5	SlimeVisual_07 (SlimeVisualConfig)
Element 6	SlimeVisual_08 (SlimeVisualConfig)
Element 7	SlimeVisual_09 (SlimeVisualConfig)
Element 8	SlimeVisual_10 (SlimeVisualConfig)
Element 9	SlimeVisual_11 (SlimeVisualConfig)
Element 10	SlimeVisual_12 (SlimeVisualConfig)
Element 11	SlimeVisual_13 (SlimeVisualConfig)
Element 12	SlimeVisual_14 (SlimeVisualConfig)

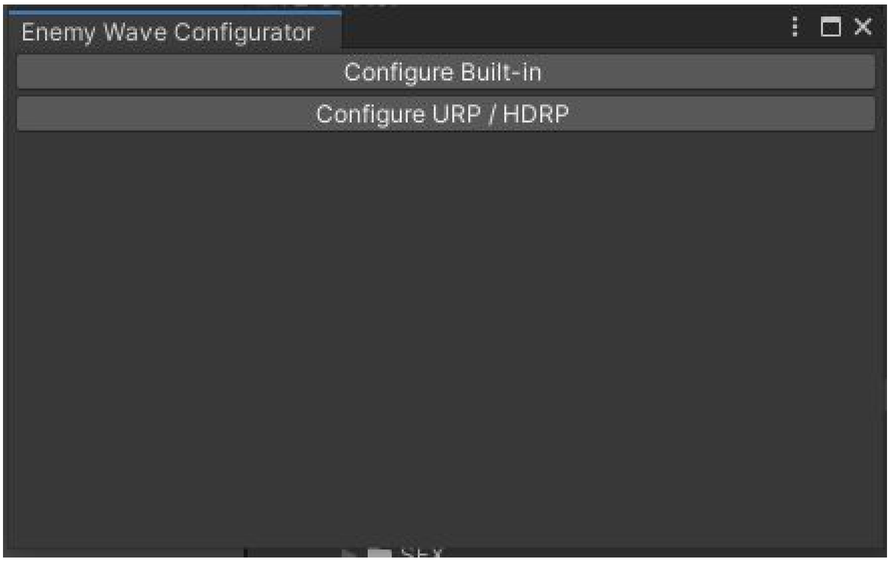
Below you can see an example of how to search and slime by ***Slime ID*** and instantiate it.

```
using Neon2.SlimeSystem;
using UnityEngine;

public class SearchAndInstantiate : MonoBehaviour
{
        //Reference to SlimeVisualDatabase
        public SlimeVisualDatabase slimeVisualDatabase;

        //The prefab to be instantiated
        public SlimeController prefabSlime;

        private void Awake()
        {

                //Serching an slime with ID 'unique_id_01'
                SlimeVisualConfig slimeVisualConfig =
                        slimeVisualDatabase.GetSlimeVisualByID("unique_id_01");

                //Instantiate the prefab
                SlimeController slimeControllerInstance = Instantiate(prefabSlime);

                //Set the instance position to Vector3.zero
                slimeControllerInstance.transform.position = Vector3.zero;

                //Set the visual configuration we retrieved previously
                slimeControllerInstance.SetSlimeVisual(slimeVisualConfig);

                /* This two last lines would not be necessary if Init On Awake is enabled */
                slimeControllerInstance.Init();
                slimeControllerInstance.GoToIdleState();
        }
}
```

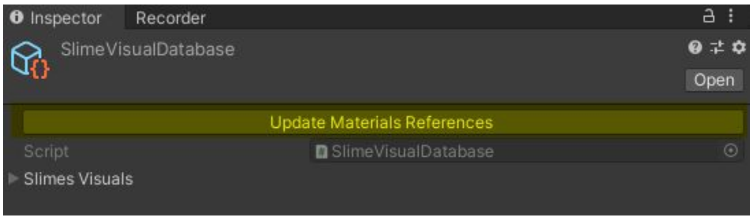
Switching between URP and Built-in

Enemy Wave! shaders and materilas support both URP/HDRP render pipeline and Built-in render pipeline. To switch between them all you need to do is update material references of your ***SlimeVisualConfig*** files.

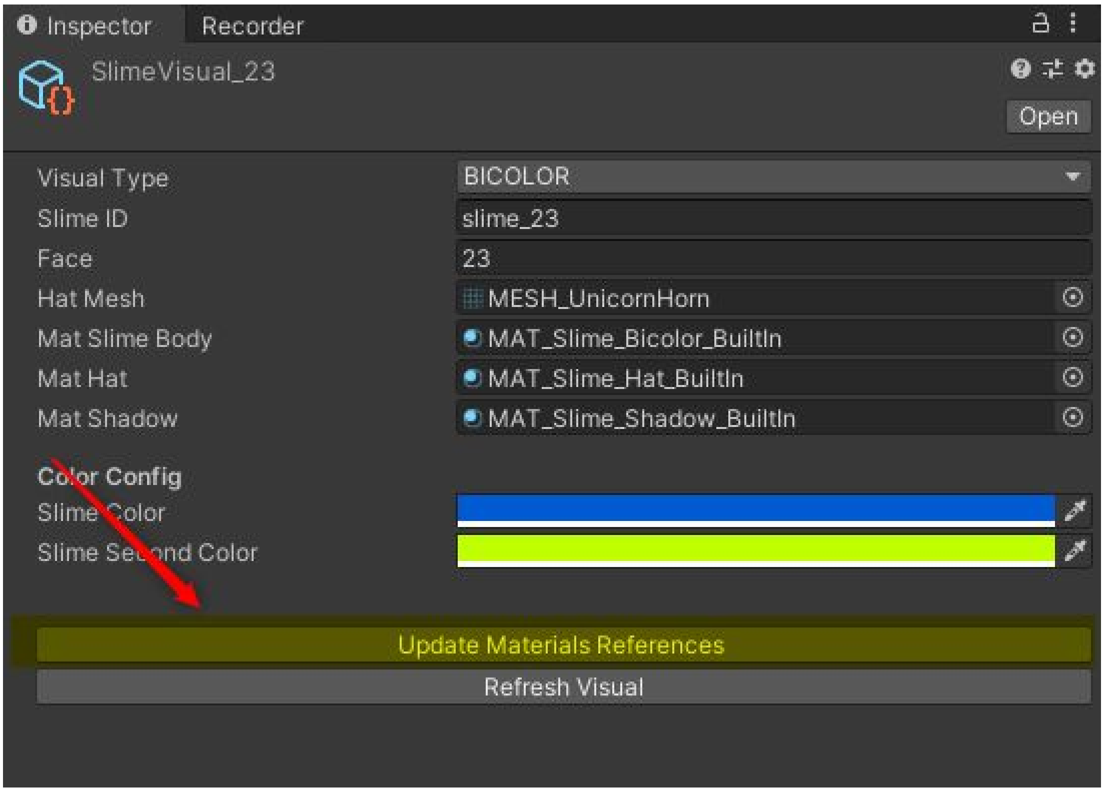
To do that, the easiest way is to open ***Enemy Wave Configurator*** (***Tools*** → ***Enemy Wave Configurator***) and click on the render configuration you are using (Built-In or URP / HDRP)



You can also do this from ***SlimeVisualDatabase*** file, clicking on "**Update Materials References**"



If you prefer update materials individually in each ***SlimVisualConfig*** file, you can click on the button with the same name (“**Update Material References**”)

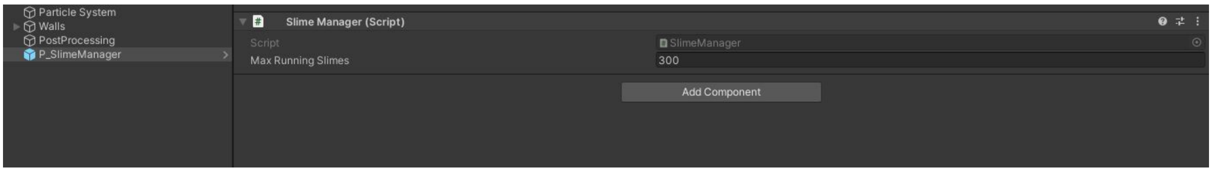


Visual Type: BICOLOR
Slime ID: slime_23
Face: 23
Hat Mesh: MESH_UnicornHorn
Mat Slime Body: MAT_Slime_Bicolor_BuiltIn
Mat Hat: MAT_Slime_Hat_BuiltIn
Mat Shadow: MAT_Slime_Shadow_BuiltIn

Color Config
Slime Color
Slime Second Color

Performance optimization

If you have a huge amount of slimes (thousands) running at the same time, depending on the target device you might want to adjust the number of slimes checking physics and doing calcs at the same time. This can be done using the component ***SlimeManager***. This component is a Singleton. Make sure that you have only one instance.
This package already has a prefab prepared to drag and drop into the scene:
P_SlimeManager (***Enemy Wave*** → ***Slimes*** → ***Prefabs***)



Script: SlimeManager
Max Running Slimes: 300

With the property ***Max Running Slimes*** you can control how many slimes can pass to Jump state. When the limit is reached, the slimes won’t be able to pass to jump state, remaining in Idle state without doing physics calculations.

Slime Controller API

Public fields

public bool initOnAwake

Controls if the slime starts moving on awake or not.

public LayerMask layerMaskCollisions

Layer mask to tell the slime which layers to consider as collision layers and move around the scene. You have to select both the ground layers and the walls layers

public LayerMask layerSlimes

Layer mask that uses the slime to dodge other slimes (In case the property ***avoidSlimes*** is enabled).

public bool avoidSlimes

If enabled, slimes dodges each other. If not enabled, the slime ignores other slimes.

public float forwardSpeed

Speed at which slime moves through the scene.

public float angularSpeed

Speed at which slime rotates.

public float maxSlopeAngle

Maximum slope angle with respect to the ground to be considered walkable slope.

public float minIdleTime

Minimum time slime can stay in Idle state.

public float maxIdleTime

Maximum time slime can stay in Idle state.

public bool idleStateEnabled

Controls if Idle state is available or not. If disabled, the slime will remain in Jump state.

public float minJumpTime

Minimum time slime can stay in Jump state.

public float maxJumpTime

Maximum time slime can stay in Jump state.

public bool jumpStateEnabled

Controls if Jump state is available or not. If disabled, the slime will remain in Idle state.

public AudioClip[] clipsJump

Array of audio clips for slime jump. A random clip is selected at each jump

Public Methods

public void Init()

Initializes the SlimeController functions. It must be called before any other method and only once per run.

public void GoToIdleState()

Slime go to Idle state

public void GoToJumpState()

Slime go to Jump state

public void SetForwardSpeed(float newForwardSpeed)

Set forwardSpeed property to newForwardSpeed

public void SetAngularSpeed(float newAngularSpeed)

Set angularSpeed property to newAngularSpeed

public void SetSlimeVisual(SlimeVisualConfig slimeVisualConfig)

Set slimeVisualConfig property

Slime Database API

Public Fields

public SlimeVisualConfig[] slimesVisuals

Array of ***SlimeVisualConfig*** files available during the game

Public Methods

public SlimeVisualConfig GetSlimeVisualByID(string id)

Get a SlimeVisualConfig by ID.

public SlimeVisualConfig GetRandomSlimeVisualConfig()

Get a random SlimeVisualConfig from all files referenced.

public SlimeVisualConfig GetSlimeVisualByIndex(int index)

Get a SlimeVisualConfig by an array index.