Skybox Lab - AI Skybox Generator

Create stunning AI-generated skybox assets within Unity for use as HDRIs in game dev and immersive projects.

Unity Versions Support

- >= 2021.x.x - works out of the box
- >= 2020.x.x - requires installing newtonsoft-json

If you are using a Unity version `2020.x.x` you would need to add a `newtonsoft-json` package to your project. Simplest way to install the package is to open `Packages/manifest.json` file of your project with your favourite editor and add the following in your `dependencies` (make sure to respect JSON commas):

```
{
 "dependencies": {
   "com.unity.nuget.newtonsoft-json": "3.0.2",
 }
}
```

Introduction

Edit the project's `manifest.json` file

Scripts in the package can be used standalone or optionally together with a Pusher websockets package. If installed, the Pusher package will use websockets to listen for any changes in the Asset Generation Process on Runtime and make updates accordingly, which in return should improve performance for your games on Runtime.

Again, easiest way to install the optional Pusher dependency is to open `Packages/manifest.json` file of your project with your favourite editor and add the following in your `dependencies` (make sure to respect JSON commas):

```
{
 "dependencies": {
  "com.pusher.pusherwebsocketunity": "https://github.com/pusher/pusher-websocket-unity.git#upm",
 }
}
```

Use git URL option

Alternatively you can go to your Unity Project, to `Window > Package Manager` and install the package using the `Add package from git URL...` option.

If using Pusher make sure to use this URL (uses the UPM branch):

`https://github.com/pusher/pusher-websocket-unity.git#upm`

Use OpenUPM-CLI

If you are using `OpenUPM-CLI`, you can easily install the Pusher package using the command below:

```
openupm add com.pusher.pusherwebsocketunity
```

For known issues after installation check the section below.

Getting Started

Samples

In the `Assets` and `Scenes` folders of the package there are samples that contain some assets and two sample scenes to get you started (Skybox and Imagine Scene).

After importing the samples load one of the above mentioned sample scenes inside your project which should be located in `Assets/Blockade Labs/Skybox Lab - AI Skybox Generator/Scenes` folder.

How to use

Pusher

On the sample scene you loaded there is game object named `Pusher`. If you don't plan on using Pusher you can safely delete it. If you plan on using Pusher on Runtime, attach the `PusherManager` script from the `Scripts` folder and that is it for that object.

Editor

Skyboxes

If you open the Skybox Scene sample, you will notice a game objects named `Skybox Sphere`. The Sphere has a Mesh Renderer component which uses a sample `Skybox Material` which in turn uses a shader of type `Skybox/Panoramic`. A texture generated with this package is assigned to the shader. You can generate a new texture that will replace the existing one on the sphere by following these steps.

1. Select the Sphere object.
2. Locate the `Blockade Imaginarium` component.
3. Add your Blockade Labs' `public` API key in the designated field first.
4. You'll notice that the object has an option `Assign to Material` ticked. Leave it as it is.
5. Click the `Get Styles` button in the `Skybox` section.
6. Select the desired style.
7. Fill the required fields (usually `prompt`) marked with an asterisk (`*`), and update the remaining fields per your preference if needed.
8. Click the `Generate Skybox` Button.
9. In about 20-30 seconds your texture will be replaced with a new texture you just created, and a folder located in `Assets/Skybox Lab Assets` will now hold your newly created sprite and texture.
10. In the `Scene` tab of the Unity editor using the `View` Tool you can position yourself inside the sphere and check out the newly generated skybox.

Imagines

Sprites

Besides skyboxes you can also generate regular 2D textures and sprites. If you open the Imagine Scene sample, you will notice 3 game objects for Character, Weapon and Environment. Also there is a disabled Cube object. You can interact with each of those objects in a similar fashion while in the Editor.

1. Select the Character object for example.
2. Locate the `Blockade Imaginarium` component.
3. Add your Blockade Labs' `public` API key in the designated field first.
4. You can leave the `Assign to sprite renderer` option ticked to assign your newly generated sprite to the current object.
5. Click the `Get Generators` button in the `Imagine` section.
6. Fill the required fields (usually `prompt`) marked with an asterisk (`*`), and update the remaining fields per your preference if needed.
7. Click the `Generate` Button.
8. In a few seconds your sprite renderer will be replaced with a new sprite you just created, and a folder located in `Assets/Skybox Lab Assets` will now hold your newly created sprite and texture.

Materials

1. Following a similar course of action as for the sprites above, you can also enable the cube object in the scene.
2. The cube object has a Mesh Renderer and a sample Material assigned.
3. Add the Api key as you would normally.
4. You'll notice that the object has an option `Assign to Material` ticked. Leave it as it is.
5. Following the same set of instructions as for the sprite, you can generate a texture that will now replace a material of the 3D object like the sample cube.

Runtime

To be able to generate assets on Runtime you just need to follow these simple steps:

1. Select the game object with the attached component of `Blockade Imaginarium`.
2. Add your Blockade Labs' `public` API key in the designated field
3. Click on the `Enable Skybox GUI` checkbox to display GUI for Skybox generation or on the `Enable GUI` checkbox for regular Imagines.
4. After you run the game a GUI will appear on top of your Game view.
5. Use the GUI in the same manner as you would in the editor (`Get Styles > Enter Prompt > Generate Skybox` or `Get Generators > Enter Prompt > Generate`).

Known Issues

If you are using an optional Pusher package, after installation on 2021.x.x versions you might get an error saying:

`Assembly 'Packages/com.pusher.pusherwebsocketunity/Packages/PusherClient.2.1.0/lib/net472/PusherClient.dll' will not be loaded due to errors: PusherClient references strong named Newtonsoft.Json Assembly references: 12.0.0.0 Found in project: 13.0.0.0.`

To resolve the issue go to `Edit > Project Settings > Player > Other Settings > Configuration > Assembly Version Validation` and disable Version Validation.