

ShapE Unity Plugin - AiKodex

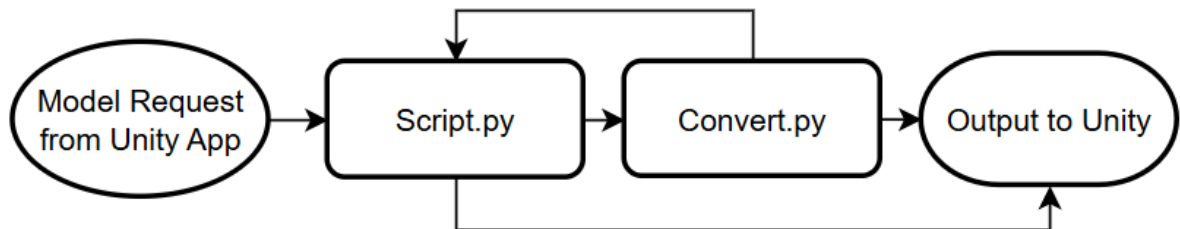
Documentation

Shap-E Installation Guide

To run Shap-E by OpenAI within Unity, you'll need to execute the model on a cloud service or your PC in the background.

Generating 3D models from text is computationally intensive, often taking several minutes even on high-end graphics cards. To maximize compatibility and avoid hardware dependency, we recommend using a cloud service. You can choose from providers like AWS, GCP, Azure, Runpod, or LambdaLabs (as of February 1, 2025). In this guide, we'll use Runpod, as it offers an easy and cost-effective starting point. Note that Runpod is **not** sponsoring us, so feel free to select any cloud provider you prefer.

Workflow:




Server Setup:

<https://www.runpod.io/console/deploy>

Deploy GPU Pod

Pod Name

Pod Template



RunPod Pytorch 2.1
runpod/pytorch:2.1.0-py3.10-cuda11.8.0-devel-ubuntu22.04

Change Template

Edit Template

GPU Count

1

2

3

4

5

Instance Pricing

On-Demand

Non-Interruptible

\$0.43/hr

Pay as you go, with costs based on actual usage time.

1 Week Savings Plan

Save \$3.36

\$0.41/hr \$68.88

Reserve a GPU for one week at a discounted hourly cost.

1 Month Savings Plan

Save \$26.88

\$0.39/hr \$262.08

Reserve a GPU for a month at a discounted hourly cost.

3 Month Savings Plan

Save \$128.10

\$0.37/hr \$789.95

Reserve a GPU for three months at a discounted hourly cost.

Spot

Interruptible

\$0.22/hr

Pay much less for an interruptible instance.

Pod FAQ


☐ Encrypt Volume

☐ SSH Terminal Access

☐ Start Jupyter Notebook

We choose an instance that already has PyTorch pre-installed.

Pod Template Overrides



Container Image

runpod/pytorch:2.1.0-py3.10-cuda11.8.0-devel-ubuntu22.04

Container Start Command

This overrides the CMD in the Docker container

Container Disk

Temporary disk space for the container

20 GB

Volume Disk

Persistent disk space mounted to the container

20 GB

Volume Mount Path

/workspace

Expose HTTP Ports (Max 10)

8888,5000

Expose TCP Ports

22

Environment Variables

+ Add Environment Variable

Clear Overrides

Set Overrides

Expose the 5000 HTTP port as our Flask app will directly connect using this port.

Libraries:

Connect to the terminal and enter the following commands:

```
cd workspace
git clone https://github.com/openai/shap-e
cd shap-e
pip install -e .
pip install bpy==3.6.0
pip install ipywidgets
pip install --ignore-installed flask
pip install --ignore-installed flask-limiter
pip install trimesh
apt-get update
apt install vim -y
apt-get install libxrender1 -y
apt-get install libxxf86vm-dev -y
apt-get install -y libxfixes-dev
apt-get install libxi6 libgconf-2-4 -y
apt-get install libxkbcommon-x11-0 -y
apt-get install libsm6 libxrender1 libfontconfig1 -y
apt-get install libglu1-mesa-dev -y
```

Screen upon completion:

```
Preparing to unpack .../libglu1-mesa-dev_9.0.2-1_amd64.deb ...
Unpacking libglu1-mesa-dev:amd64 (9.0.2-1) ...
Setting up libopengl0:amd64 (1.4.0-1) ...
Setting up libglx-dev:amd64 (1.4.0-1) ...
Setting up libglu1-mesa:amd64 (9.0.2-1) ...
Setting up libgl-dev:amd64 (1.4.0-1) ...
Setting up libglu1-mesa-dev:amd64 (9.0.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
root@8c8568f47019:/workspace/shap-e#
```

Scripts:

Enter the following command:

```
vim script.py
```

Paste the script: (use `:set paste + i`) in vim to avoid indent problems

```

import torch
import subprocess
from flask import Flask, jsonify, request
from flask_limiter import Limiter
from flask_limiter.util import get_remote_address
import requests
import io
import base64
import json
from shap_e.diffusion.sample import sample_latents
from shap_e.diffusion.gaussian_diffusion import diffusion_from_config
from shap_e.models.download import load_model, load_config
from shap_e.util.notebooks import create_pan_cameras, decode_latent_images,
gif_widget
from shap_e.util.notebooks import decode_latent_mesh

app = Flask(__name__)
limiter = Limiter(
    get_remote_address,
    app=app,
)

device = torch.device('cuda')
xm = load_model('transmitter', device=device)
model = load_model('text300M', device=device)
diffusion = diffusion_from_config(load_config('diffusion'))

@app.route('/data', methods=['POST'])
def data():
    json_data = request.get_json()
    prompt = json_data['prompt']
    cfg = json_data['cfg']
    steps = json_data['steps']
    if int(steps)>64:
        steps=64
    fileFormat = json_data['fileFormat']
    fileFormat = fileFormat.lower()
    return generate(prompt, cfg, steps, fileFormat)

def generate(prompt, cfg, steps, fileFormat):

    latents = sample_latents(
        batch_size=1,
        model=model,
        diffusion=diffusion,

```

```

        guidance_scale=int(cfg),
        model_kwargs=dict(texts=[prompt]),
        progress=True,
        clip_denoised=True,
        use_fp16=True,
        use_karras=True,
        karras_steps=int(steps),
        sigma_min=1E-3,
        sigma_max=160,
        s_churn=0,
    )

    for i, latent in enumerate(latents):
        t = decode_latent_mesh(xm, latent).tri_mesh()
        with open(f'plyFile.ply', 'wb') as f:
            t.write_ply(f)

    subprocess.run(f"python3 convert.py {fileFormat}", shell=True)

    with open(f"outputFile.{fileFormat}", "rb") as f:
        b64 = base64.b64encode(f.read()).decode("utf-8")
        return b64

if __name__ == '__main__':
    app.run(debug=False, host = '0.0.0.0')

```

Esc + :wq to save the file in VIM and exit

For Convert.py enter the following command:

```
vim convert.py
```

Paste the script:

```

import argparse
import bpy
import os
import math
import trimesh

def export_file(input_file, output_file, output_format):

    if output_format == "glb":
        mesh = trimesh.load(input_file)

```

```

rot = trimesh.transformations.rotation_matrix(-math.pi / 2, [1, 0, 0])
mesh = mesh.apply_transform(rot)
rot = trimesh.transformations.rotation_matrix(math.pi, [0, 1, 0])
mesh = mesh.apply_transform(rot)
mesh.export(output_file, file_type="glb")

else:
    bpy.ops.wm.read_factory_settings(use_empty=True)
    bpy.ops.import_mesh.ply(filepath=input_file)

# Export based on the chosen format
if output_format == "fbx":
    bpy.context.active_object.rotation_euler[0] = math.radians(270)
    bpy.ops.object.transform_apply()
    bpy.ops.export_scene.fbx(filepath=output_file, use_selection=True)
elif output_format == "blend":
    bpy.ops.wm.save_as_mainfile(filepath=output_file)
elif output_format == "obj":
    bpy.ops.export_scene.obj(filepath=output_file, use_selection=True)
elif output_format == "gltf":
    bpy.ops.export_scene.gltf(filepath=output_file, use_selection=True)
else:
    print("Invalid output file format. Please choose from fbx, blend, obj,
or gltf.")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Script to export a Blender file
to different formats.")
    parser.add_argument("output_format", choices=["fbx", "blend", "obj", "gltf",
"glb"], help="Output file format.")

    args = parser.parse_args()

    input_file = "plyFile.ply"
    output_file_format = args.output_format
    output_file = f"outputFile.{output_file_format}"


# Call the export function
export_file(input_file, output_file, output_file_format)

```


To activate the script use the command:

```
python script.py
```

Wait for the script to download the necessary models:

 **Shap-E**
Version 2.0

Mesh Generator

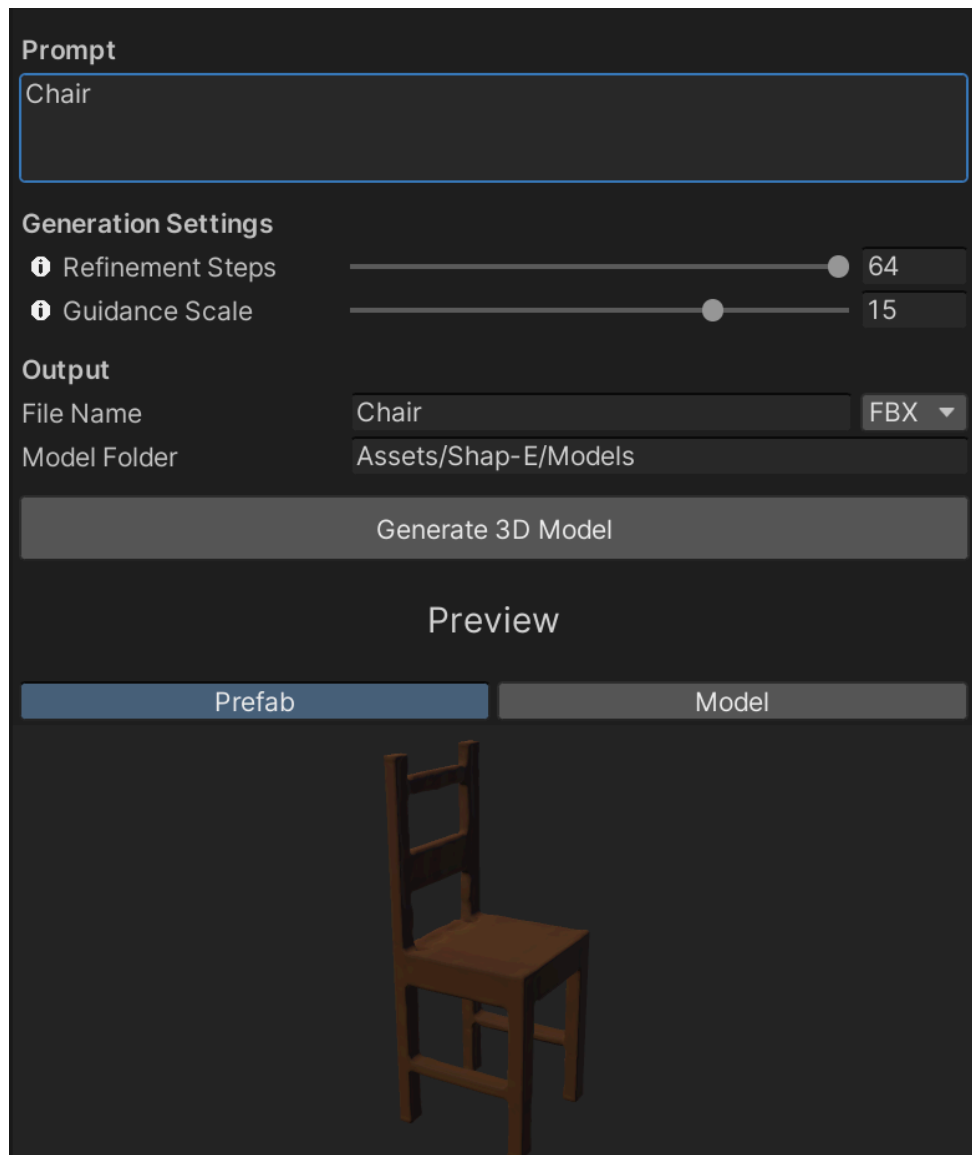
 URL

<https://cct2l48vii6f89-5000.proxy.runpod.net/data>

Save

Enter a prompt and click on generate.

```
Press CTRL+C to quit
100% | 64/64 [00:07<00:00, 8.12it/s]
/workspace/shap-e/shap_e/models/stf/renderer.py:286: UserWarning: exception rendering with PyTorch3D: No module named 'pytorch3d'
  warnings.warn(f"exception rendering with PyTorch3D: {exc}")
/workspace/shap-e/shap_e/models/stf/renderer.py:287: UserWarning: falling back on native PyTorch renderer, which does not support full gradients
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/numpy/core/getlimits.py:518: UserWarning: The value of the smallest subnormal for <class 'numpy.float64'> type is zero.
  setattr(self, word, getattr(machar, word).flat[0])
/usr/local/lib/python3.10/dist-packages/numpy/core/getlimits.py:89: UserWarning: The value of the smallest subnormal for <class 'numpy.float64'> type is zero.
  return self._float_to_str(self.smallest_subnormal)
/usr/local/lib/python3.10/dist-packages/numpy/core/getlimits.py:518: UserWarning: The value of the smallest subnormal for <class 'numpy.float32'> type is zero.
  setattr(self, word, getattr(machar, word).flat[0])
/usr/local/lib/python3.10/dist-packages/numpy/core/getlimits.py:89: UserWarning: The value of the smallest subnormal for <class 'numpy.float32'> type is zero.
  return self._float_to_str(self.smallest_subnormal)
Successfully imported 'plyFile.ply' in 1.654 sec
FBX export starting... 'outputFile.fbx'
export finished in 0.2983 sec.
100.64.0.30 - - [02/Feb/2025 16:59:05] "POST /data HTTP/1.1" 200 -
```

If you have any issues, please reach out to us at info@aikodex.com

EARLIER Version 1.0

NOTE:

Please allow Unity up to 4 hours to generate an Invoice Number from the time of your purchase. Once it is available, you can enter this number in the invoice field on top to access the model generator.

Shap-E, a conditional generative model for 3D assets, is developed by OpenAI. Shap-E-Text to 3D for Unity makes this technology easily accessible to the Unity community.

Examples



A pick ax



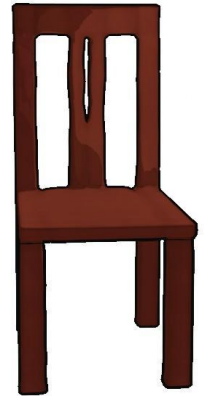
A red barrel



Futuristic Sword



Wooden Crate



A chair

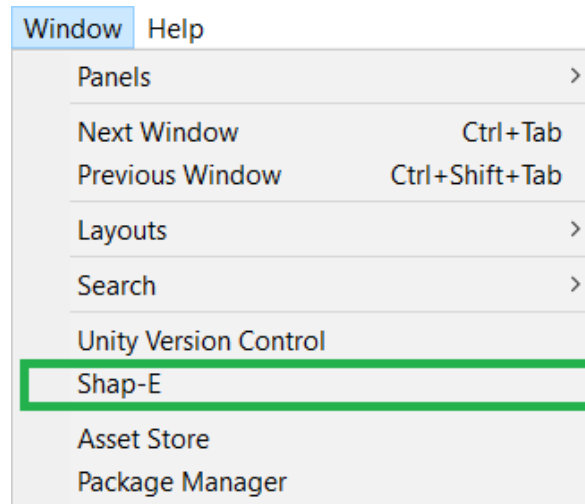
Dependencies

This asset requires the external package Editor Coroutines 1.0.0 which can be found in Window > Package Manager > Editor Coroutines. Note that Editor Coroutines


comes pre-installed in Unity Versions 2021.2 and above.

Usage

To use the Editor's Extension please go to Window > Shap-E



Editor Graphical Interface:

 **Shap-E**
Version 1.0

Mesh Generator

i Invoice Number Verify Save

Prompt

Generation Settings

i Refinement Steps 64

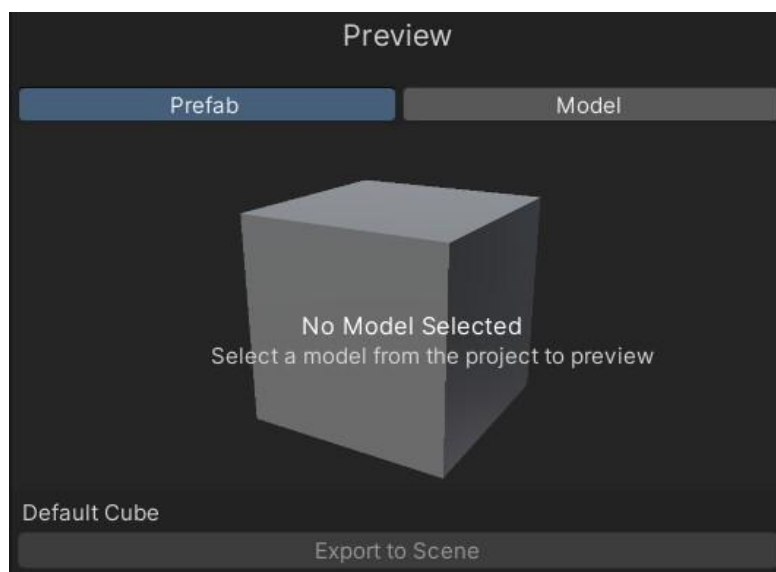
i Guidance Scale 15

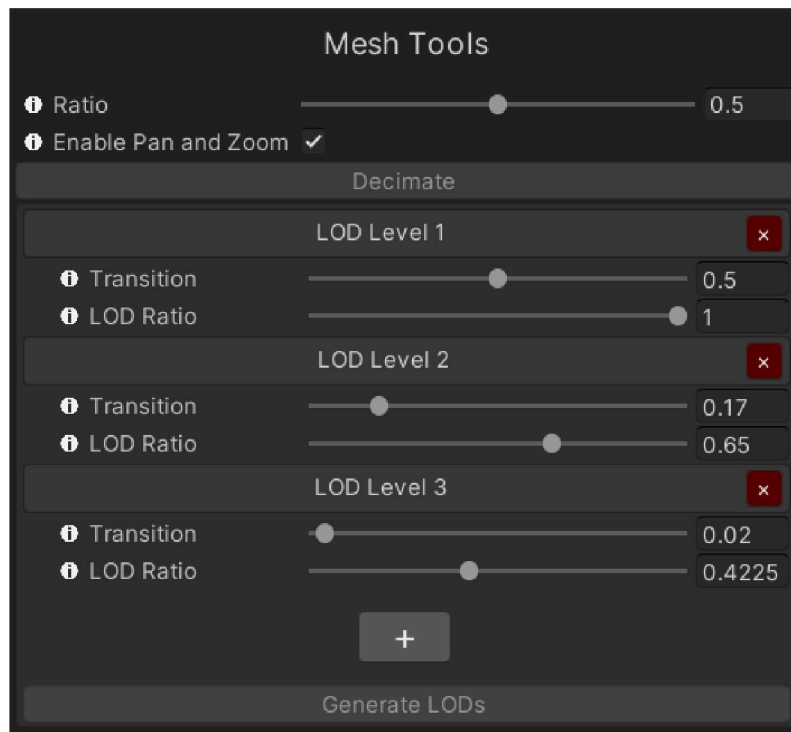
Output

File Name FBX ▾

Model Folder

Generate 3D Model

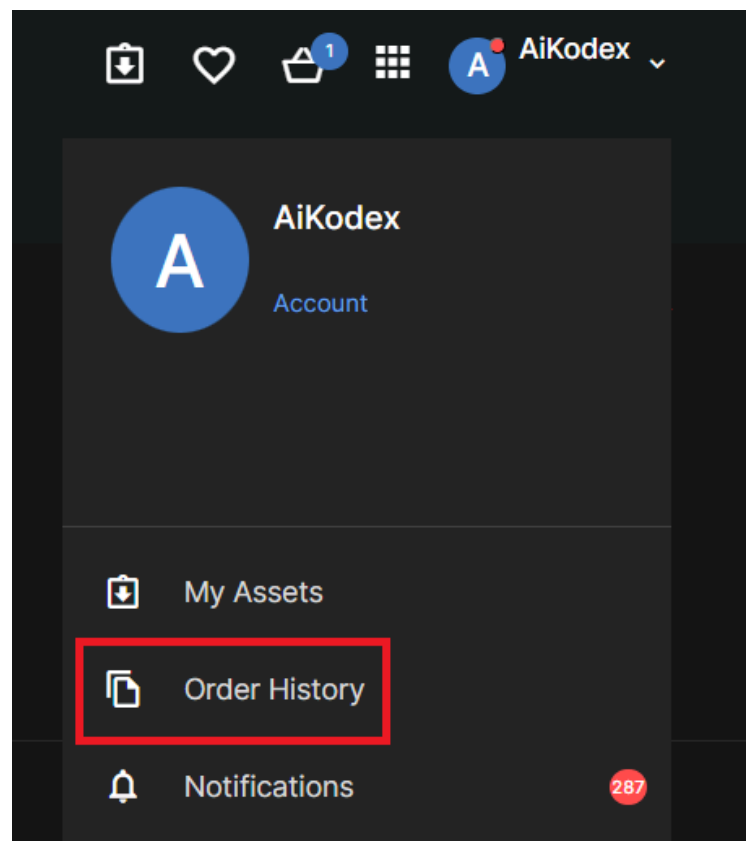





Features

Linked Invoice Number

We use this unique identifier to assign you a number of models every month. You can find the invoice number here in the My Orders section of the Unity Asset Store.



Order History						
Order Date ▾	Payment Type	Order Id	Invoice Number	Order Total	Status	
		2474	IN01000		COMPLETED	View Order Details
		2474	IN01000		COMPLETED	View Order Details
		2474	IN01000		COMPLETED	View Order Details
		2474	IN01000		COMPLETED	View Order Details



Shap-E
 Version 1.0

Mesh Generator

i Invoice Number

Prompt

Text to Mesh ID

Image to Mesh ID

From Image

None (Texture 2D)

Select

Clear

Once you have entered the number, you can then click on **verify** to check if the number you have entered is correct. If it is, you can save the number so when you launch the window again, you do not need to fill this field again. Once you have generated models with the plugin, you can check the status of generation by clicking on the Verify button. It should tell you the number of models you have generated.

Editor Window

Mesh Generator

Prompt: This field allows you to input text for mesh generation. Enter the text you want to convert to a mesh in the "Prompt" field. This would automatically change the name of the file as well. The typical guidance scale range for this is 13-17.

Refinement Steps: This number is the number of iterations performed by the model to render the object. The typical range for the best generations is 60-70 with 64 being the ideal number of steps.

Guidance Scale: This value determines how accurately the model represents the prompt.

File Name and Extension: The file name changes on the basis of the prompt. The file name also prevents overwriting of files with regenerations by including an underscore with the number of items. The extensions available are FBX, BLEND and GLB at the moment. Overwrite file check runs for every format to prevent accidental overwriting.

Model Folder: We recommend against path changing due to paths being hard coded in the editor. The default path is Assets/Shap-E/Models where all the models are stored.

Preview

The preview section is a convenient feature that allows you to preview mesh files directly within the interface, eliminating the need to switch between different windows or applications. By utilizing this functionality, you can quickly assess the information of model (geometry information, Vertex colors, wireframe) files without interrupting your workflow.



To access the preview section, simply single-click on a file within your project. Keep the plugin window active beside the scene. Hover your cursor over this panel, and you will notice that the preview section becomes enabled.

Once the preview section is activated, you can view the Prefab with the Particle Shader which exposes the vertex colors and model on the other tab which shows you the wireframe and the structure of the model.

This functionality is particularly useful when you need to review and evaluate the contents of multiple models. Instead of opening each file individually in separate 3D applications, the preview section enables you to conveniently view files directly within the Shap-E panel.

By offering a seamless and efficient way to preview model files, this feature helps streamline your workflow and enhance productivity. It provides a centralized editor window where you can organize your project's models. Whether you're a 3D designer, 3D artist or modeler, or working on any project involving meshes, the Unity preview section offers a valuable tool to facilitate your creative process and ensure the optimal outcome of your work.

Mesh Tools

Mesh Decimation

Select a model file you wish to decrease the poly count of. Once selected, use the

Ratio slider to adjust how many polygons you want to keep. In the example below, the model has a total number of 7096 triangles. Ratio is set to 0.5. After executing decimation, the model should have 3548 triangles.



You may use the Enable Pan and Zoom toggle to view the model in more detail. You can also change the lighting by right clicking and dragging the mouse.

Mesh AutoLOD

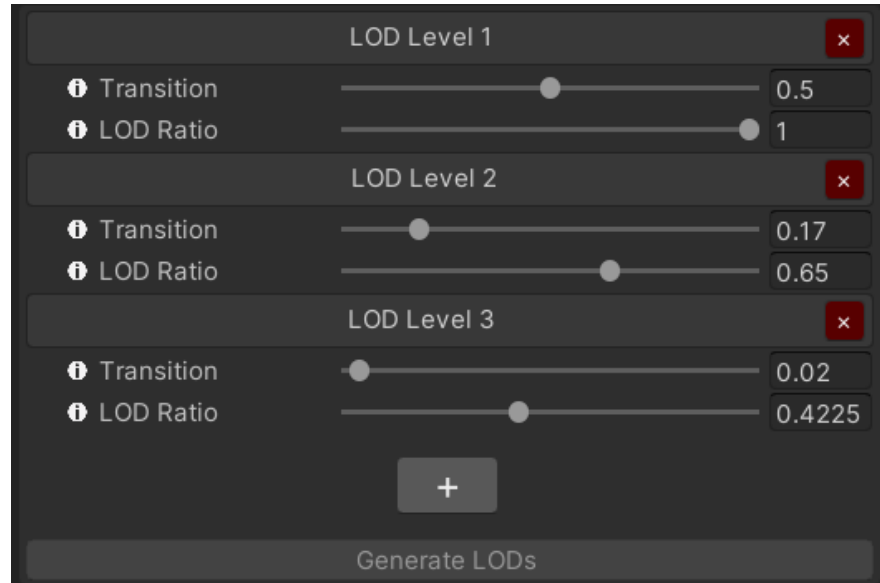
To generate LODs automatically, you need to export the object to the scene. Only exported and visible in hierarchy, you will find the option to generate LODs enabled. Click on the option to generate LODs automatically. The LODs will be saved in the Shap-E folder by the name of LODs.

You may add LODs by clicking on the + symbol at the bottom or remove LODs by pressing on the X button against them.

Transition: This variable gets and sets the screen relative height to use for the

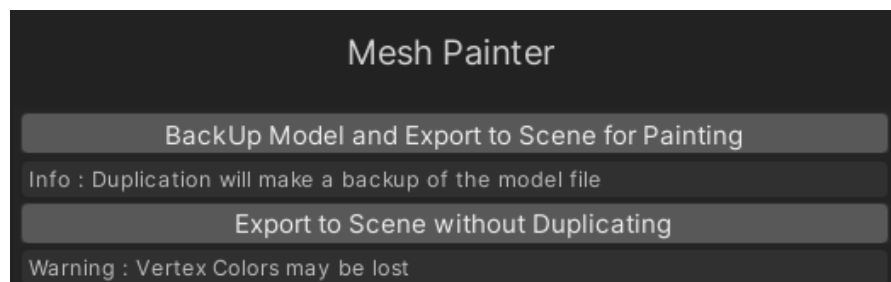
transition to the next LOD essentially acting as the parameter which makes the switch between LODs.

LOD Ratio: This refers to the quality or the decimation value of the model at the particular transition. Typically the closer the model is to the camera and scene, the higher the LOD ratio as you'd like to have a higher resolution model for a higher fidelity up close.

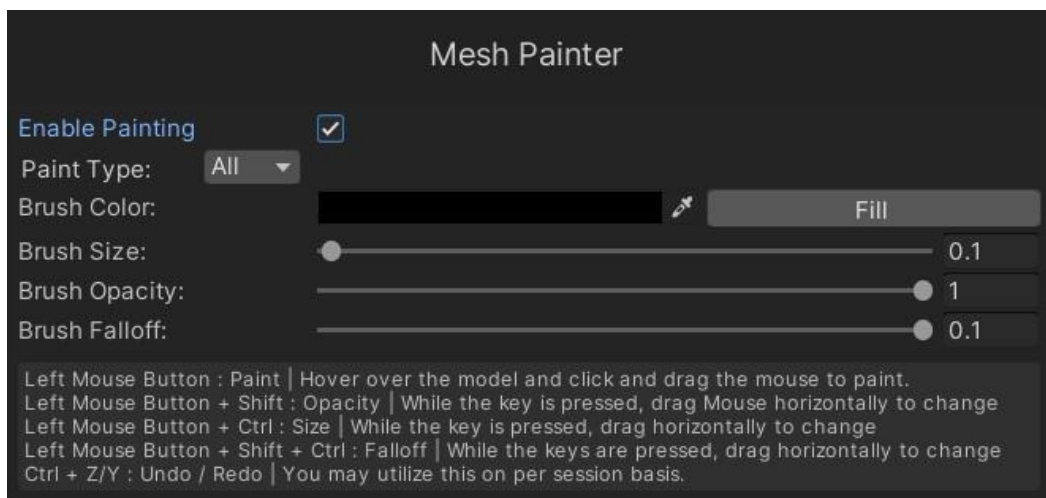


Mesh Painter

Select a model to reveal the options:



It is advisable to create a copy of the model before painting (even though Undo is available while painting in a single session). After clicking on the Export to Scene for painting, a copy of the model will be created and the model will be exported to the scene for painting. After that, click on the button enabling painting to view the menu.



You should see a cursor which will allow you to paint on the mesh. Follow the instructions and key binding on the info bar underneath to have a better control on painting style.

If you have any questions or difficulties in generation of models from text, decimation, AutoLODing or vertex painting, please reach out to info@aikodex.com

Troubleshooting:

500: Internal Server error / Cannot connect to destination host

Possible causes:

There may be quotation marks, new lines or any other illegal character in the text submitted.

There may be times that the server is down. In this unlikely event, please check the forum for announcements by us pertaining to server maintenance or contact info@aikodex.com if this issue persists for over a day. Please check your internet connection and try again in a few hours.

400: Bad Request

Possible Causes

Occurs when the information passed is not recognized either due to syntax error or other reasons. Using special characters like (""") that have a specific meaning in coding can cause this error. Please send us an email with the Unity version included outlining your issue in as much detail. A screenshot or video of the problem will help us serve you better.

Request Timeouts (408 Request Timeout response status code)

Possible Causes

The server model could have run into a generative error. When this happens, the model file becomes extremely large and is not able to be passed through the API built. Please try again with a fewer number of words.

Too many requests (429)

Possible Causes

It may be possible that a failed generation caused you to skip the timer which triggered

this error. You can try again in a couple of seconds.

Warning: Unable to allocate new scene culling mask

The cause for this is a bug in Unity which prevents the display of 3D models on the Shap-E editor. The resolution is that you can simply restart the editor and the problem should be fixed.

Privacy Policy

At AiKodex, we believe that protecting the privacy of our users is of utmost importance. We provide a secure and private environment for users to utilize our OpenAI's Model Generation services, without compromising on their privacy.

Our privacy policy is designed to ensure that we do not store any data or personal information that is shared between users and our service. We do not store input text, generated models, IP addresses or any other data, and our remote servers are configured to automatically delete any data that is left on the system.

We understand that privacy is a fundamental right, and we are committed to upholding this right for our users. We will continue to invest in the latest technologies and security measures to ensure that our users can enjoy a safe and private experience with ShapE for Unity.

Licensing

You will have the copyright of all the models you generate, thereby granting you full ownership of models you create with this unity asset. You can use the models for all commercial activities.

[ASSET STORE TERMS OF SERVICE AND EULA](#)

Original MIT license is included with the asset:

[MIT License](#)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "As is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

Prohibited Activities and Misuse

You shall not use these services, data or content provided by ShapE in any manner that is illegal, unethical, or inconsistent with the intended use of the service. You may not abuse the server by overloading requests on it or using the service from outside Unity. This may lead to termination of service.

Happy ShapEing!

- Offered By AiKodex