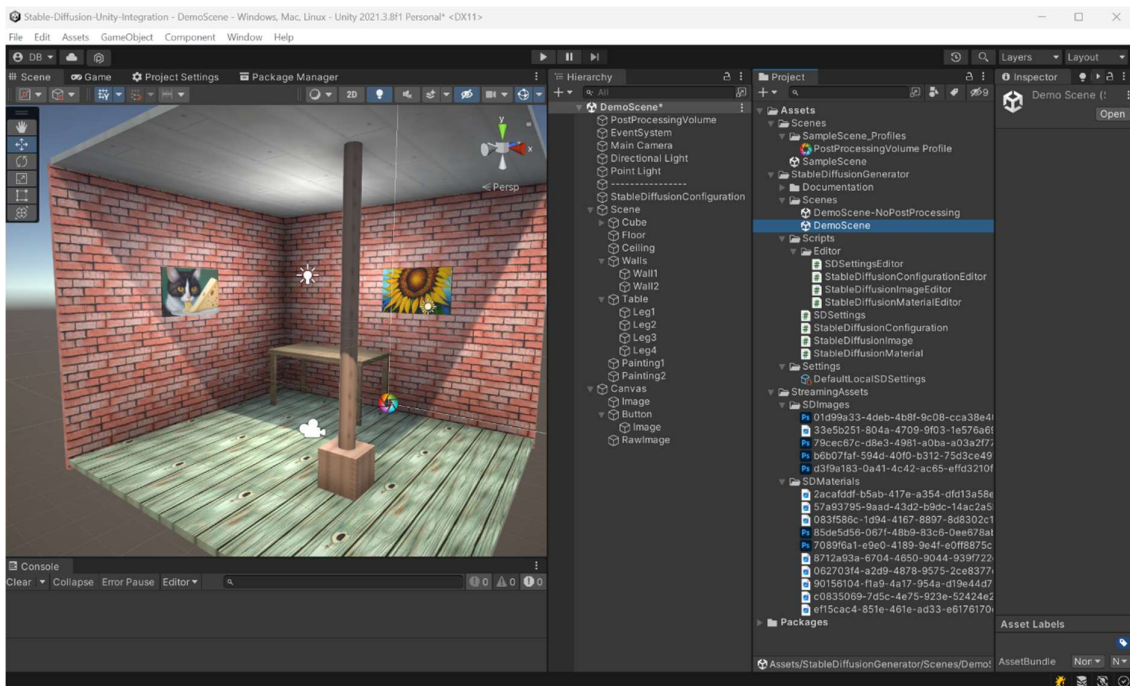


Stable Diffusion Unity Integration v1.0

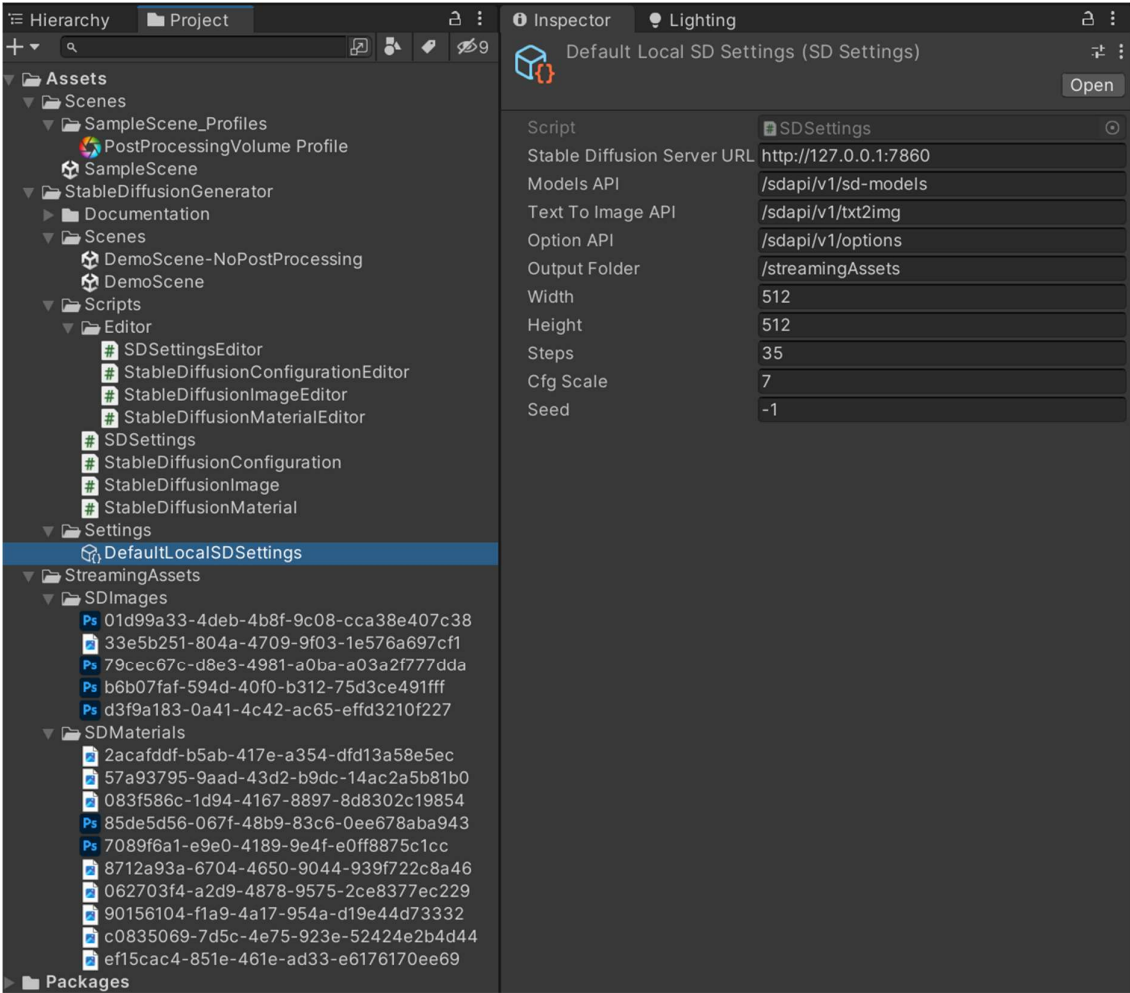
by Shadodo76

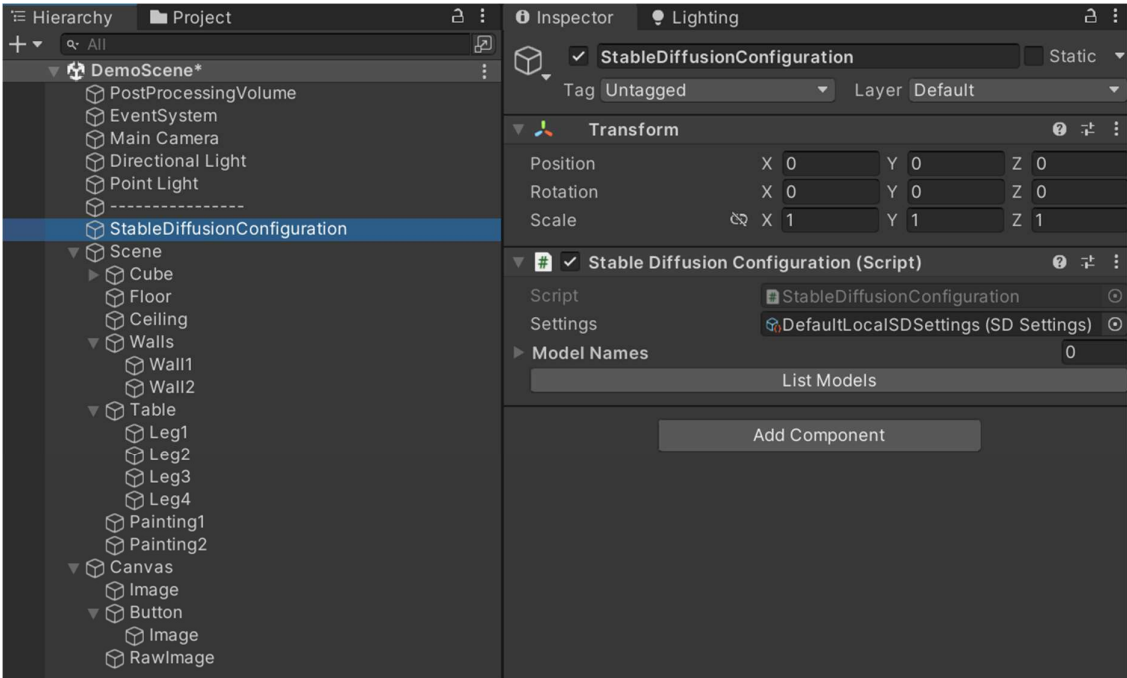
A Unity Editor Component for image generation using Stable Diffusion Automatic 1111 [webui](https://www.webui.com/) from within the Unity Editor.



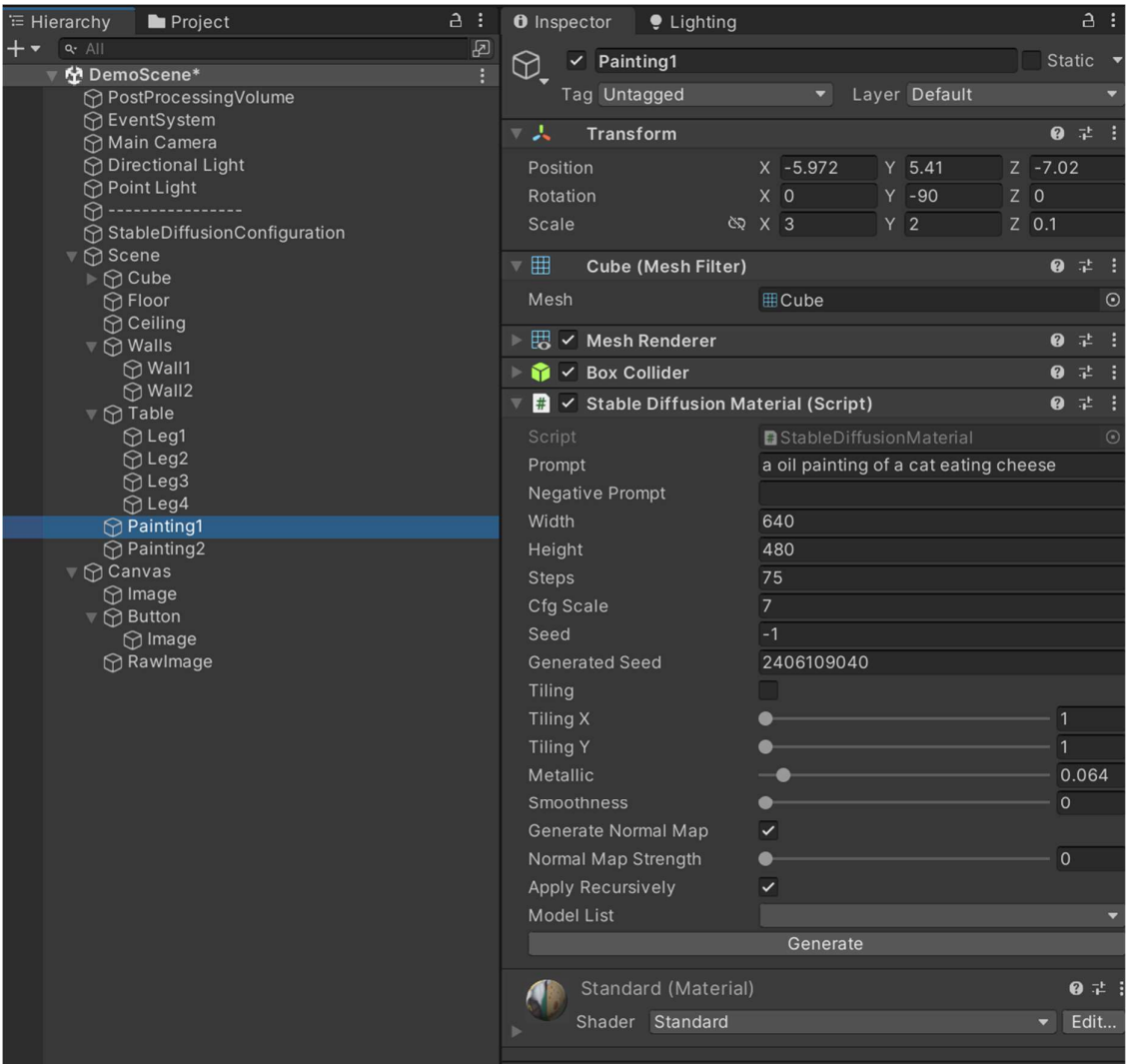
Overview

Simple server configuration with StableDiffusionConfiguration component and SDSettings ScriptableObject assets.

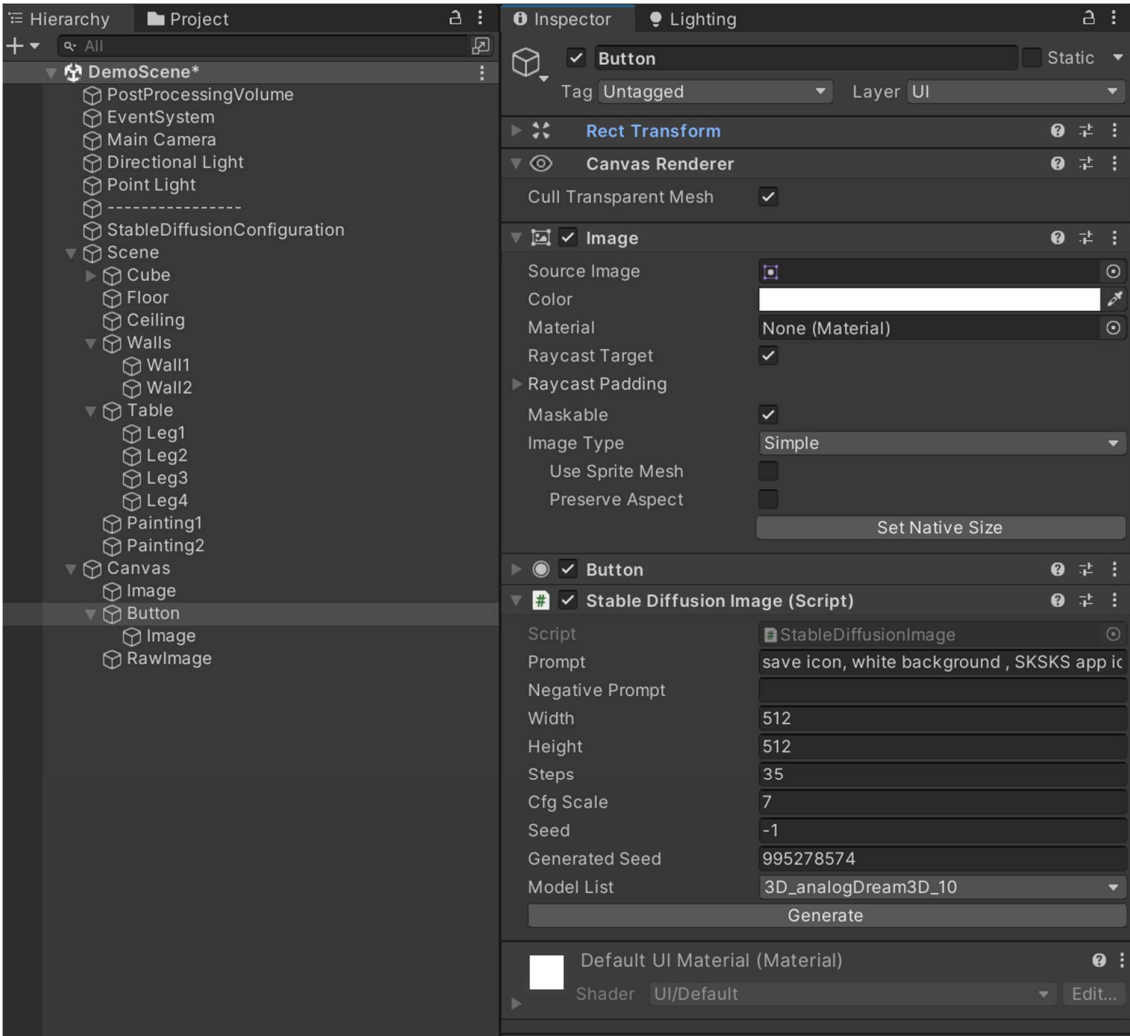




Easy Text-to-Image generation for 3D models:



Straightforward Text-to-Image generation for UI elements:



Features

- Text-to-Image generation with Prompt and Negative prompt for: a) texturing 3D models having a MeshRenderer component with tiling option and generation of Normal/Bump maps, b) UI component having a Image or RawImage component (Canvas, Panel, Image, Button, etc.)
- Editor only components for scene/level design (no runtime dependencies to Stable Diffusion)
- Image generation using any Stable Diffusion models available in the server model folder
- Standard parameters control over image generation (Prompt and Negative Prompt, Nb. of Steps, CFG Scale, Image dimension and Seed)
- All images saved in the Assets folder for persistent storage and reference

Dependencies and requirements

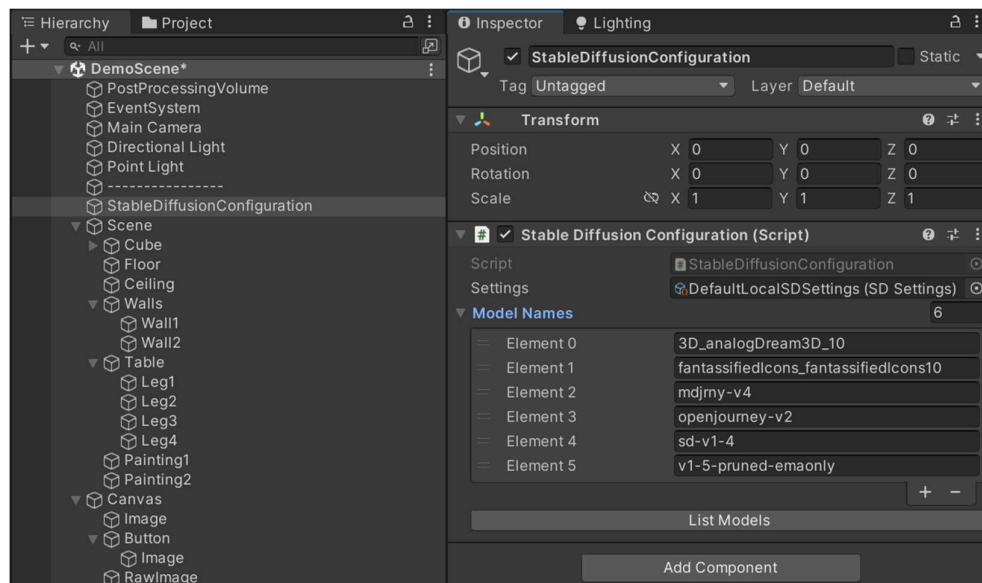
This Unity Editor tool requires access to a local Stable Diffusion Automatic 1111 server. Installation of this server is outside the scope of this documentation. Please refer to the SD WebUI repo: <https://github.com/AUTOMATIC1111/stable-diffusion-webui>

Getting Started

1. Install [Stable Diffusion WebUI Automatic 1111](https://github.com/AUTOMATIC1111/stable-diffusion-webui).
2. Launch [Stable Diffusion WebUI Automatic 1111](https://github.com/AUTOMATIC1111/stable-diffusion-webui) with the '--api' argument in the command line.
3. Either, Download the stable-diffusion-unity-integration repository, for example by running `git clone https://github.com/shadodo76/Stable-Diffusion-Unity-Integration.git`. Then, Open the stable-diffusion-unity-integration project in Unity. then open the Demo Scene provided in the package found in `StableDiffusionIntegration/Scenes/DemoScene.unity`.

At this point, the project Hierarchy should be looking something like this:

3. Or, Create and open a new Unity project for Built-in Pipeline, download and import the stable-diffusion-unity-integration.unitypackage file into the project (<https://github.com/AUTOMATIC1111/stable-diffusion-webui/stable-diffusion-unity-integration.unitypackage>). By starting from a new project and importing the package, none of the demo pre-generated images will be available in the Asset folder but the project should work as-is, since the textures and images are embedded in the Unity project.
4. Open the Demo Scene provided in the package found in `StableDiffusionIntegration/Scenes/DemoScene.unity`. There is a version with Post-Processing (you need to install the Post-Processing package, you will need to re-open the scene after installing the package) and a version without Post-Processing.
5. Select the `StableDiffusionConfiguration` component and make sure the selected setting point to the correct URL and Port (default: <http://127.0.0.1:7860/>).
6. Click on the `List Model` button to get the list of all available models from your local Stable Diffusion server.



7. Select any existing GameObject with a MeshRenderer (or that contains at least one child with a MeshRenderer) or an Image (or RawImage) and click Generate from the StableDiffusionMaterial or StableDiffusionImage component to generate a new image using the specified parameters.
8. If any error occurred, it should be caught by a try/catch and the exception displayed in the Unity Console.

Limitations

Be aware of a few limitations:

- This repo has been testing only in Unity 2019, 2020 and 2021. It may work on other versions but there is no guarantee.
- The components for generating materials are designed for the Built-in render pipeline only. It may be easy to get it to work for URP or HDRP but no effort has been done in that regard. Feel free to contribute your changes if you make it work.
- Only a local Stable Diffusion server, which requires no API key, was used in testing. This may work with remote servers if they don't require API keys. Feel free to contribute your changes if you get it to work with remote server and API keys.

Contributing

To contribute, clone the repository, make your changes, commit and push to your clone, and submit a pull request (PR).

Before submitting a PR, make sure that:

- you do not add multiple unrelated things in same PR.
- your changes do not break anything before submitting.
- you do not submit PRs where you just take existing lines and reformat them without changing what they do.

If you think you found a bug, please first make sure that you are using the latest version of both this repo (as well as the Stable Diffusion repo) and that the bug has not already been fixed. If you are submitting a bug fix, there must be a way for me to reproduce the bug. Please, provide step-by-step instructions on how to reproduce the bug.

Credits

Thanks to [UnityCoder](#) for a straightforward and functional algorithm to automatically generate a normal map from a texture.

- <https://github.com/unitycoder/NormalMapFromTexture>

Licence

This repo is under Lesser General Public License (LGPL) v2.1 , that you can find in the `License` file. Mainly, it allows you to:

- use, modify, and distribute this code base and other copyrighted work.
- create derivative works to be distributed under a different license than the original.
- requires that the source code for the derative work be made available to others.
- and that a notice is included stating that the work is based on this original LGPL-licensed code.