

Retro 3D Shaders

Version 1.1.0 | January 2019

© Kevin Foley 2018 – 2019

onemanescapeplan@gmail.com

Read First

- As with any plugin or template from the Asset Store, I **strongly recommend that you do not make any changes directly to the template files**. When I release updates in the future, **any changes that you make to files in the template may be overwritten and lost** when you download an update for this package. Instead of making changes directly to the shaders, or other files, make copies of those files in a separate folder, and edit the copies.
- **These shaders require your meshes and scenes to be configured a certain way in order to look their best**. Please see the section “Configuring Your Project for Best Results”

Configuring Your Project for Best Results

The included shaders are designed to emulate the look of classic 3D games from the 16-bit era. **In order to achieve the perfect retro look, there are some additional steps you should follow when building your game:**

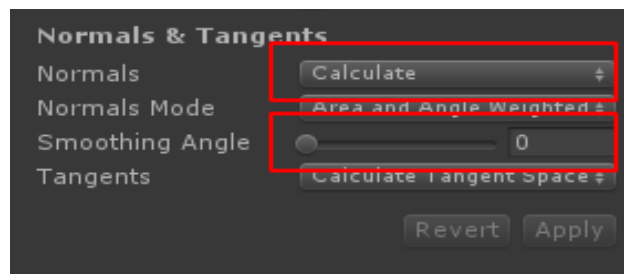
- Export your meshes from your 3D modeling software with “flat shading” instead of “smooth shading”. This applies lighting to your mesh in a way that preserves hard edges. Otherwise, the lighting engine will automatically smooth the edges of your model, making it look totally screwy.



Mesh exported with flat-shading (left) vs smooth shading (right)

If you acquired the model from someone else and don't have modeling software to re-export it, you can change the mesh's import settings in Unity to disable smooth shading:

1. Select the Mesh
2. In the Inspector, find the section “Normals and Tangents”. Change **Normals** to “**Calculate**” and set **Smoothing Angle** to **0**.



- Only use directional lights in your scene (some of the shaders do not even support other types of light)
- Disable “Soft Shadows”, either on your lights or in your project Quality settings. Enable only Hard shadows, or no shadows at all.

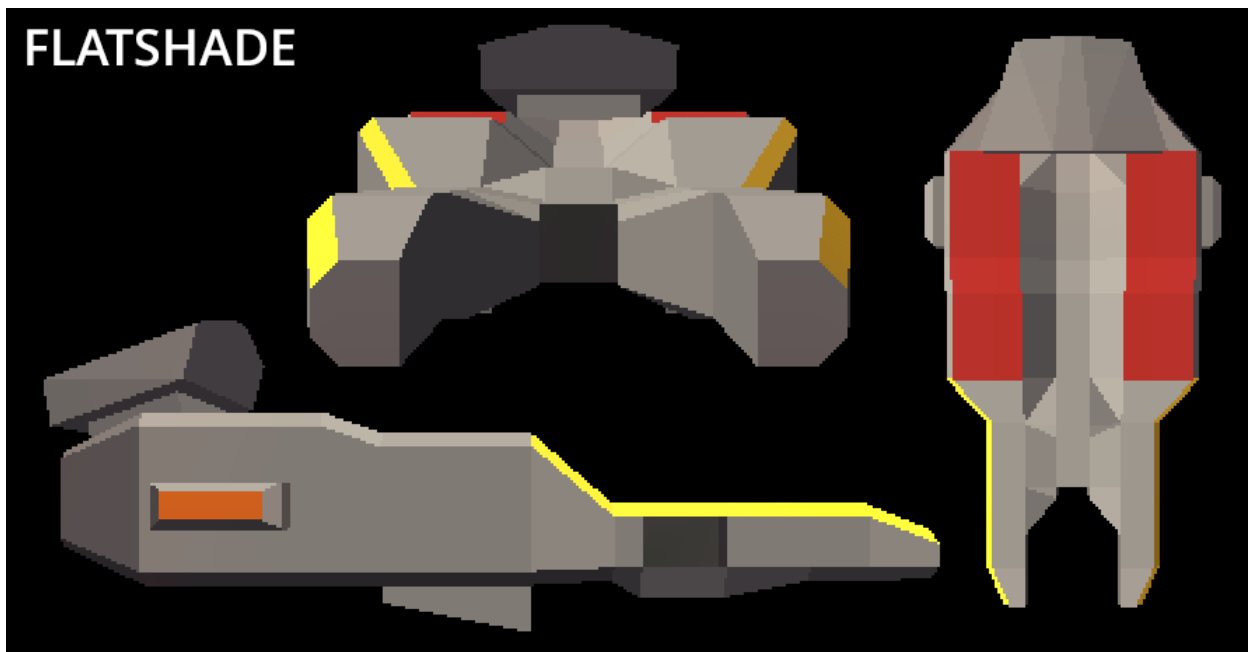
Shaders

This package comes with several retro-style shaders to emulate the look of 3D console and arcade games from the early 90s. All shaders support casting and receiving shadows.

The package includes the following shaders:

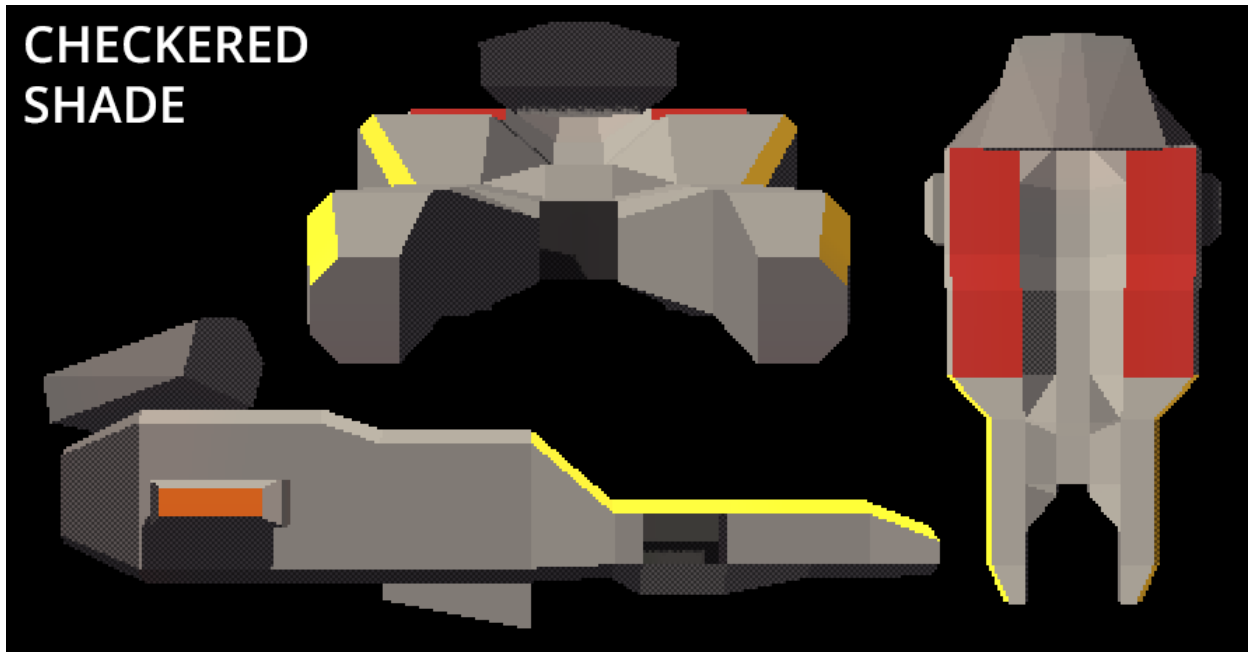
Retro Flat Shader

Renders in a flat-shaded style with no specular highlights. Supports all types of lights (directional/point/spot).



Retro Checkered Shader

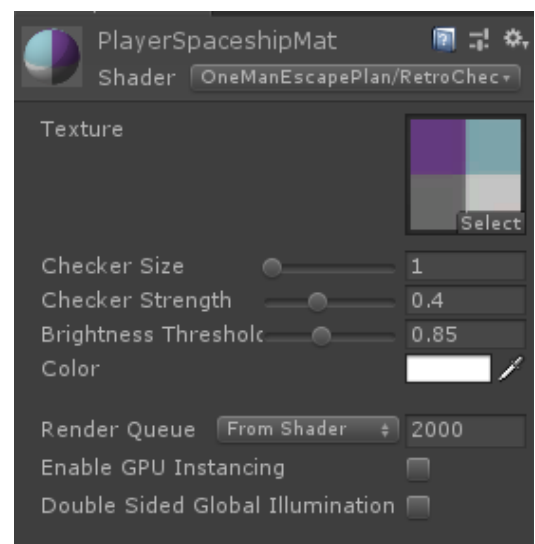
Renders in a flat-shaded style with no specular highlights, and shades in darker areas with a checker pattern. This is inspired by some early 3D games, particularly those for the SNES that use the Super FX chip. Only supports directional lights.



You may need to zoom in to be able to see the checker pattern in this image

This shader has the following settings

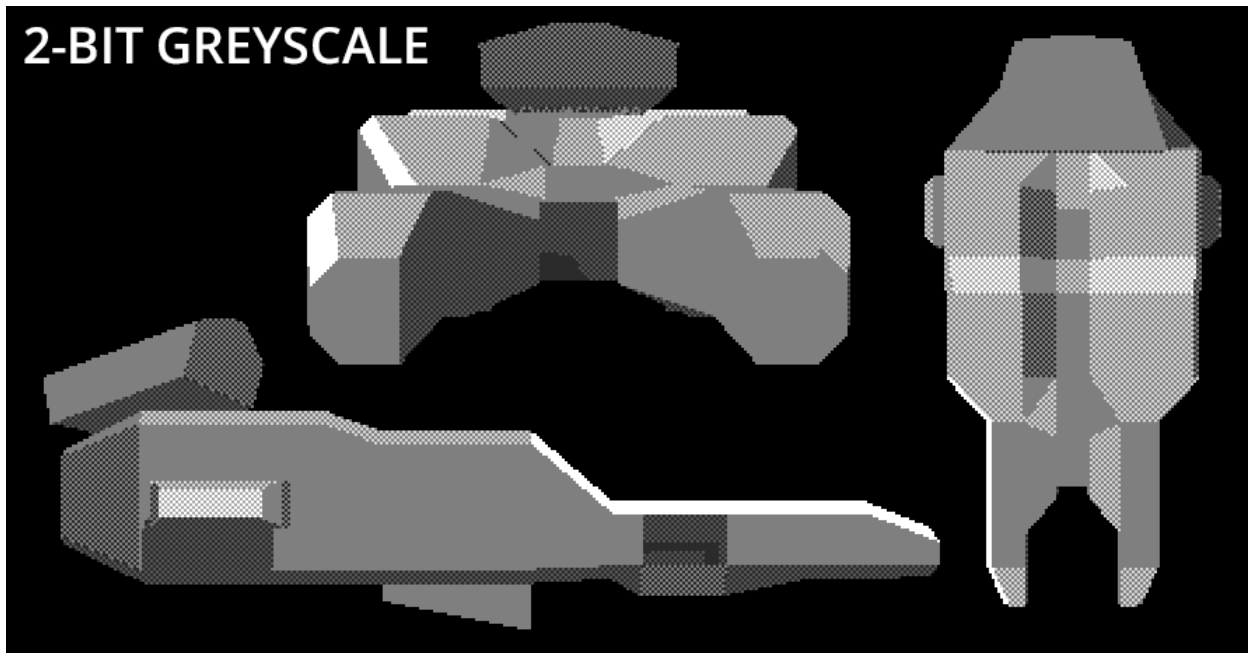
- **Checker Size:** Changes the relative size of the checker pattern (the pattern is automatically scaled for the current window resolution). You will probably want to leave this at 1.
- **Checker Strength:** The intensity (alpha) of the checkers, where 0 is invisible and 1 is pure black
- **Brightness Threshold:** Controls the minimum brightness below which checkered shading is applied to a surface. Decrease this value to only apply checkered shading to darker areas. Increase to apply it to brighter areas
- **Color:** A tint that is applied to the material color



The RetroCheckedShader settings

Retro Greyscale Checkered Shader

This is an experimental shader which renders in greyscale with a limited color palette. This was designed for emulating the appearance of a 2-bit (4-shade) greyscale screen, such as that on the original Nintendo Game Boy. Only supports directional lights.



This shader has the following settings:

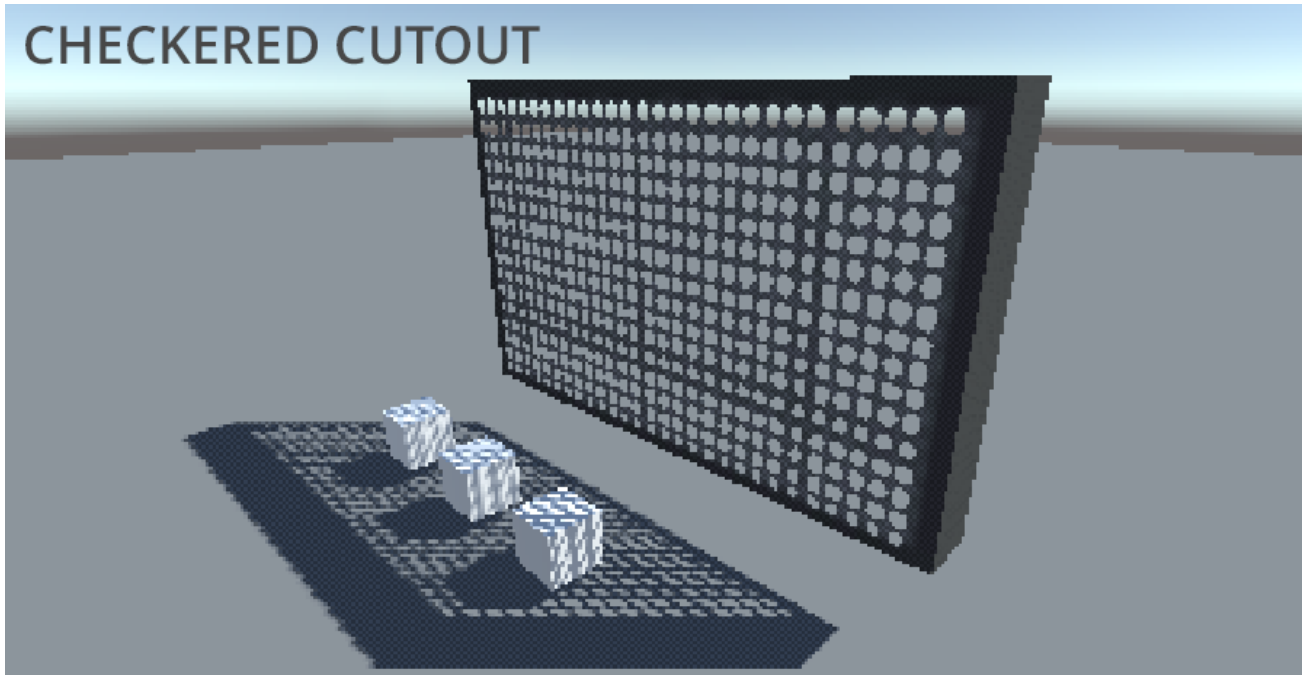
- **Checker-shading enabled:** When checkered shading is enabled on a material using this shader, additional shades of grey are simulated with checkering (for example, an alternating checker pattern of light and dark grey pixels can be used to simulate mid-grey).
- **Checker Size:** Changes the relative size of the checker pattern (the pattern is automatically scaled for the current window resolution). You will probably want to leave this at 1.
- **Min Brightness:** Adjusts the minimum brightness of any pixel rendered by the shader. Increasing this value brightens black pixels. Useful when black pixels are getting lost in a dark background.
- **Max Brightness:** Adjusts the maximum brightness of any pixel rendered by the shader. Decreasing this value darkens white pixels. Useful when white pixels are lost in a light background.



The RetroGreyscaleCheckeredShader settings

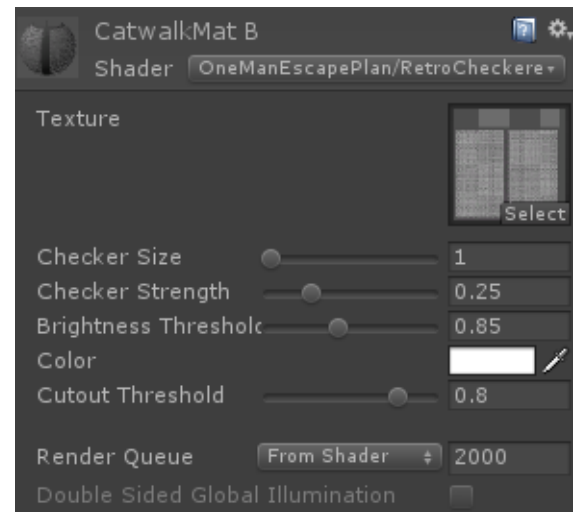
Retro Checkered Cutout Shader

This shader is used for rendering cutout surfaces (such as fences) in the checkered style. In a nod to modern graphics, it renders accurate cutout shadows. This shader is a bit more performance-intensive due to the cutout shadows.



This shader has the same settings as Retro Checkered Shader, plus one additional setting:

- **Cutout Threshold:** Controls the alpha below which pixels are discarded. For example, if the threshold is “.8” then any of the texture’s pixels with an alpha of less than .8 will be cut out when the material is rendered.



The RetroCheckeredCutoutShader settings

Demo Scenes

The project comes with three demo scenes under OneManEscapePlan/Retro_3D
Shaders/Demo/Scenes. These scenes can be used for reference to compare the appearance of the different shaders.

- **Demo:** Displays a rotating spaceship with one of the three shaders applied. Click the button to toggle to the next shader.
- **Demo2:** Displays three spaceships side-by-side, each demonstrating one of the shaders. The scene's directional light rotates, so you can study how light and shadows move across the ship.
- **Cutout Demo:** Demonstrates the Retro Checkered Cutout shader using a rotating metal catwalk.