

Trouble viewing this?

- Use [Typora](#) to view this file (which is what I used to edit)
- Or, view the [PDF](#) version of this file

Still need to complete:

- GitHub link
- Contact form
- Itch link
- "A Deeper Look" section (needs polishing)

## Table of Contents:

### Quick Intro

### Ways to Contribute

[Asset Store](#)

[GitHub](#)

[Itch](#)

### Quick Start

[Imports](#)

[Global Variables](#)

[Start Method](#)

[Check File](#)

[Find Image](#)

### A Deeper Look

[How it Works](#)

### FAQ

# Quick Intro

---

Highlights:

- As of writing this, 5 stars on the Asset Store
- Extensive documentation
- Super easy way to edit photos
- FREE!

Why I made this:

- Initially needed a way to manipulate pictures programmatically
- Couldn't find another implementation

What this documentation is intended to be:

- Easy to follow/understand
- Straight to the point
- Clear and concise

# Ways to Contribute

---

## Asset Store

---

- Leave a good rating
- Let others know what you experienced

## GitHub

---

- Give a star
- Report issues
- Make requests
- Submit pull requests

## Itch

---

- Give a donation
- Leave a rating
- Leave feedback via a comment

# Quick Start

---

## Imports

---

First make a new C# script, then set up the imports:

```
using Picture_Editor_v2.Scripts;           // Contains Texture2DEditor class
using Picture_Editor_v2.Scripts.Commands;  // Contains all of the commands
```

## Global Variables

---

Make the global variables:

```
public MeshRenderer RendererToManipulate; // Place to store the edited texture
public Texture2D Tex;                     // Original texture
private Texture2DEditor _myTextureEditorVariable; // Changes (a copy of) the texture
```

## Start Method

---

Used to make the edits to the photo:

```

void Start ()
{
    // Initialize the editor with the texture
    _myTextureEditorVariable = new Texture2DEditor(Tex);

    // Add commands in the order they should execute
    _myTextureEditorVariable.AddCommand(new Filter(Filters.Sepia)); // Adds a sepia filter
    _myTextureEditorVariable.AddCommand(new GaussianBlur(2));      // Next, blurs the image

    // Call "GetTexture2D" to get the new texture
    RendererToManipulate.sharedMaterial.mainTexture = _myTextureEditorVariable.GetTexture2D();
}

```

## Check File

What your file should look like:

```

using Picture_Editor_v2.Scripts;           // Contains Texture2DEditor class
using Picture_Editor_v2.Scripts.Commands; // Contains all of the commands
using UnityEngine;

public class Test : MonoBehaviour
{
    public MeshRenderer RendererToManipulate; // Place to store the edited texture
    public Texture2D Tex;                     // Original texture
    private Texture2DEditor _myTextureEditorVariable; // Changes (a copy of) the texture

    void Start ()
    {
        // Initialize the editor with the texture
        _myTextureEditorVariable = new Texture2DEditor(Tex);

        // Add commands in the order they should execute
        _myTextureEditorVariable.AddCommand(new Filter(Filters.Sepia)); // Adds a sepia filter
        _myTextureEditorVariable.AddCommand(new GaussianBlur(2));      // Next, blurs the image

        // Call "GetTexture2D" to get the new texture
        RendererToManipulate.sharedMaterial.mainTexture = _myTextureEditorVariable.GetTexture2D();
    }
}

```

## Find Image

Locate the image you want to edit

## A Deeper Look

# How it Works

---

*a   b   c*  
*d   e   f*  
*h   i   j*

# FAQ

---

**Will Photo Editor make changes to the original file?**

No,