ChangeLog

1.1.1

- support for **Volume Framework** - new custom CRTVolumeComponent with all CRT Filter attributes
- added legacy (CG) versions of shaders for Unity 2022.x or Unity 6 in compatibility mode

1.1.0

- updated for the **new Unity 6 render pipeline** and it's Core RP Library 17.0.3 with IRenderGraphRecorder
- updated bliping, adjusted shader, using proper texture handles
- **new shader with limited features to improve performance on mobile devices** (see 'Improving performance' chapter below)
- adjusted ranges for various parameters

1.0.5

- new feature - shadowlines can be set to negative values to change orientation from horizontal to vertical

1.0.4

- updated for URP 14.0.8
- updated for Unity 2022.3.8f1
- removed unused libraries which may cause some issues with latest unity version
- fixed "upside-down" camera render behavior during edit time and pause
- added injection point to parameters for better control if used in other situation than full screen with pixel perfect
- removed parameter passes that are not necessary with URP 14.0.8 (cleaner code)

1.0.3

- updated for URP 14.0.7
- use of RTHandles instead of Textures

Overview

Customizable CRT effect usable either as a full screen effect or on any camera in the project using URP. It uses high-performance single pass shader to create many CRT "artifacts". This shader can be used separately as well in advanced scenarios. **Please read 'Installation instructions' below to properly configure the filter.** Thanks for using my CRT Filter for URP. If you need any support, please contact me at: curio124@gmail.com

Package contents:

This package comes with the following asset files:

- SampleScene folder – sample scene and assets required for it (screenshot from the game Cursed Castilla - credits to Locomalito, Gryzor87 - Abylight Studios) **please note, that the sample scene will not work properly until you configure your URP pipeline**. For more info see 'Installation instructions' below
- **Scripts\CRTRendererFeature** script - main script for the ScriptableRendererFeature
- **Scripts\CRTVolumeComponent** script - custom CRT Volume Component to be used with Unity's Volume Framework
- Scripts\ReadMeInfo script - script used in sample scene to point you to this documentation
- **Shaders\CRTFilter shader** - main shader for the filter
- **Shaders\CRTFilterLite shader** - shader with limited features to improve performance on mobile devices
- Shaders\CRTFilterUnityCG shader - unityCG version of the shader for Unity 2022.x.x
- Shaders\CRTFilterUnityCGLite shader - unityCG version of the lite shader for Unity 2022.x.x
- Settings\ExampleCRTFilter2DRenderer - example of renderer settings. For more info see 'Installation instructions' below
- Settings\ExampleUniversalRPSettings - example of URP settings. For more info see 'Installation instructions' below

Installation instructions for full screen CRT effect using Pixel Perfect Camera:

1. **Create a new 2DRenderer or adjust provided ExampleCRTFilter2DRenderer** You can either create **a new 2DRenderer** anywhere in your assets folder by right-click in the project explorer under Assets section and selecting **Create->Rendering->URP 2D Renderer**, or you can use (optionally also rename and relocate) provided **ExampleCRTFilter2DRenderer** file from the Settings subfolder.
2. **URP settings location** This package is compatible only with the URP, as it uses URP way of adding renderer passes. If the project is configured properly for the URP, there is an existing URP settings. It's usually located either in the root Assets folder or in the Settings subfolder with a default name **"UniversalRP"**. If you have any difficulties to locate URP settings file, open "Project Settings" and select section "Graphics". Used URP settings file is visible in the first configuration item - "Default Render Pipeline" (or before Unity 6 - "Scriptable Render Pipeline Settings"). By clicking on the item, URP settings file will be located in the project explorer.
3. **URP settings configuration** note: URP use render passes also for the scene editor. If you change default renderer (first item in the "Renderer List"), CRT filter will be applied also to the scene editor. Better option is to add another renderer, that will be used only with selected camera(s), what is described in this step. Open settings file from the step 2. in the inspector and **add a new renderer to a "Renderer List" by clicking "+" button** at the end of the "Renderer List". **Add reference to a rendered from the step 1** to a newly created record in the renderer list.

4. **Camera configuration - necessary for the sample scene as well** Once a new renderer was added to the renderer list in the URP configuration (step 3.), it can be selected for any camera in the project. **If you wish to see the CRT Filter in the sample scene, camera in the sample scena has to be adjusted as well.**
    i. select the camera to which you would like to apply the CRT filter
    ii. find the **"Camera"** component and expand **"Rendering"** section
    iii. select proper renderer (from the step 1.) in the **"Renderer"** setting
5. **Configure 2DRenderer** note: if you've used provided ExampleCRTFilter2DRenderer, it already contains CRT Renderer Feature with proper configuration, so you may skip this step and all it's sub steps.
    i. open 2DRenderer from the step 1. in the inspector and scroll down to the section "Renderer Features"
    ii. add CRT Filter Renderer Feature by clicking **"Add Renderer Feature"** button at the end of settings list and choosing **"CRT Renderer Feature"**
    iii. expand section "CRT Renderer Feature" if it's collapsed
    iv. rename the name of the feature (there was a known issue, if the feature has the same name as the object). Name it "CRT Filter" for example.
    v. set the **"Shader"** setting for the **"CRTFilter"** (the provided shader)
    vi. set the **"Shader Lite"** setting for the **"CRTFilter"** (the provided lite shader)
6. **Configure CRT Filter** note: CRT Filter is configured in the 2DRenderer. You may create multiple 2DRenderers with different CRT Filter configuration if you need it for your game (valid option for in-game monitors).
    i. open 2DRenderer from the step 1. in the inspector and scroll down to the section "Renderer Features"
    ii. expand section "CRT Renderer Feature" if it's collapsed
    iii. choose any preset to see CRT effect on all properly configured cameras - with selected preset, most values can't be changed
    iv. or choose a **"custom" preset** and modify any values to your liking - for more info see 'Reference' below
    v. **!!!IMPORTANT!!!!** for pixel effects (blur, smidge, bleed and to some extent also scanlines) you need to provide proper pixel resolution (x and y). This CRT effect takes the texture and no matter what is its real resolution (can be upscaled) it processes each pixel on it by this resolution. This makes sense only if you use the filter for pixel/retro graphic style, when your game looks like for example 320x240px game even tough resolution is 4x higher (each pixel in the game is in reality 4x4 pixels with the same color). CRT filter needs to know that there are 320x240 pixels in the texture to work properly without any artefacts.
7. **Custom color spaces** If you don't use standard color space, you may want to change the color space for the temporary render color buffer in the **CRTRendererFeature.cs** class in the OnCameraSetup method.

ADVANCED - Improving performance (mostly for mobile devices)

This CRTFilter is based on a simple **single-pass shader**. If the GPU is presented, impact on the performance is neglectable. On the mobile device (or on a device without GPU in general), mathematical calculations required for every pixel may decrease the performance. The math itself can't be omitted and it's already optimized to my best knowledge. What can be done tough is to skip the calculation for effects that are not required/used. Version 1.1.0 introduced **new CRTFilterLite shader that can be selected in the configuration**. You can either select this lite shader using a checkbox "Use Shader Lite" or by referencing CRTFilterLite shader in the "Shader" option. Checking "Use Shader Lite" checkbox also disable sliders that are not supported by the lite shader. (if changing sliders doesn't effect camera renderer, restart the scene to create new material in the pipeline after changing checkbox value or shader)

Lite shader keeps the most important effects:

- vignette and screen geometry
- bleed
- scanlines, shadowlines and noise and removes subtle effects that are probably not required on the mobile device anyway:
- blur and smidge
- aperture lines
- image adjustments (brightness, contrast, gamma, etc.)

If you don't use effects that are still in the Lite version, you can further increase the performance by skipping the math in the shader. Just edit the Lite version of the shader and comment/uncomment lines in ApplyCRT method around the line 218.

ADVANCED - Using Volumes

CRTFilter from version 1.1.1 supports volumes and provides CRTVolumeComponent. This component contains all variables of the CRT filter and may be used as any other VolumeControl with VolumeProfiles. Use of volumes is configured by parameter **Use Volume Component**. Once enabled, all atributes in the CRT Renderer Feature are ignored (can't be even modified as the profile is automatically set to 'None') and values from calculated volume are used.

If volumes doesn't work or CRT filter reports error - reffer to the Troubleshooting section bellow.

Usage of VolumeControls and VolumeProfiles is out of the scope of this documentation - please refer to public Unity Volume Framework documentation.

ADVANCED - Using CRT effect on other than main camera

You can use CRT filter on any camera in the project and each can have different CRT effect configuration. You may proceed similarly as in **installation instructions** above and just add proper

2DRenderer to any camera. If more cameras use the same renderer, they will share CRT filter settings obviously. If you need different settings for different camera(s), you need to create more 2DRenderers.

ADVANCED - Using CRT filter on non pixel/retro game or camera

This filter can be used on any graphic style (even 3D), although it was created specifically for 2D pixel game using Pixel Perfect Camera. Specific for 2D games are just pixel effects (blur, smidge, bleed and to some extent also scanlines). Shader checks each VISUAL pixel (as user sees them - e.g. 320x240) and apply math to effect each TEXTURE pixel (by texture resolution - e.g. 1920x1080). If the texture is not "pixelated" and this virtual resolution (e.g. 320x240) is not properly set in CRT filter settings, results are not guaranteed and based on graphical style may be better or worse. Try to play with settings to reach acceptable results.

ADVANCED - Usage of shader

Main value of this CRT filter is its shader. If you know how, you may use it in your own pipelines to achieve nice effect in some advanced scenarios. To describe use of shader is out of scope of this doc and also of the CRT filter package itself.

Requirements

CRT filter requires URP to work properly. Shader can be used also in other pipelines, but provided script and instructions are implemented for URP.

Limitations

CRT filter can't be used without component that creates camera texture such as PixelPerfectCamera or postprocessor on the camera. Any other component that renders to camera texture can be used, but the filter works best with PixelPerfectCamera and with resolution set to same values in the CRT filter and PixelPerfectCamera componet.

Reference

CRT filter is configured in the 2DRenderer asset (see 'Installation instructions' above). Most parameters are either self-explanatory or their effect can be easily seen if changed. Some important notes:

- if preset is set to any value except "custom", most parameters can't be changed and are fixed by selected preset
- you may choose any preset and then change preset to "custom" to modify all values
- or you may choose preset "none" and then change it to "custom" to start from scratch
- Pixel Resolution X & Y has to be set to the same values as in PixelPerfectCamera component - otherwise some effects may be misaligned (see some explanations above)

- smidge effect is design to be used only with the bleed effect, without bleed, smidge doesn't look right
- to set custom offsets for R, G and B, Chromatic Aberration has to be set to 0
- see values used for presets for inspiration

Unity 2022.x.x or Unity 6 Compatibility Mode (without Render Graph)

The latest version of this asset is adjusted to Unity 6 and it's new and improved rendering pipeline using Render Graph. If you would like to use CRT filter with previous version of Unity or in Unity 6 with Compatibility Mode enabled (without Render Graph), please download previous version of this asset (1.0.5) or adjust CRTRendererFeature.cs class by commenting out existing RecordRenderGraph and Dispose methods and uncommenting section at the end and by using UnityCG variants of both full and lite shaders. instructions are provided directly in the CRTRendererFeature.cs class.

Troubleshooting

- **values in CRT renderder feature can't be modified**
  - values can be modified only if the profile is set to 'Custom'. Any other profile overwrites all values and prevents its modification
  - if "Use Volume Control" option is enabled, 'None' profile is forced and values can't be modified. If this option is enabled, values in Renderer feature are ignored anyway and values from CRTVolumeComponent are used instead
- **sample scene doesn't use CRT filter**
  - if you've opened sample scene in your existing project, your URP settings still use standard renderer. You have to configure your URP settings (see 'settings' above) and properly set camera in the sample scene to use a renderer that you've properly configured
- **no output of the camera**
  - disable CRT filter in the 2DRenderer to be sure, that camera is working properly without the filter
  - check the console, if there isn't a warning regarding not existing camera texture. If there is the message, make sure there is some component that renders to camera texture (such as PixelPerfectCamera with crop enabled or standard camera with postprocessing enabled)
  - try change any other value in 2DRenderer (for example disable and enable post-processing)
- **camera is 'upside-down' during the edit mode**
  - there was a bug introduce with a specific combination of system + unity editor version + URP version. It was fixed in the version 1.0.4. If it persist or you have any other issue, pls. contact me at: curio124@gmail.com
- **CRT effect is not visible**
  - make sure, that camera is using proper renderer (camera settings in camera inspector)
  - make sure, that renderer has added CRT Filter and it's enabled

    - make sure, that filter has some settings (if everything is set to 0, there is no effect from the filter)
- **there is an error in the console**
    - try to fix it by yourself - both renderer class and shader is available to you
    - contact me: curio124@gmail.com
- **CRT filter decreases the performance of the game**
    - try to use Lite version of the shader - see chapter 'ADVANCED - Improving performance (mostly for mobile devices)' above
- **CRT filter not working in Unity 6**
    - make sure, "Compatibility Mode" in Project Settings -> Graphics -> section "Render Graph" is disabled
    - if "Compatibility Mode" is required, follow instructions in chapter 'Unity 2022.x.x or Unity 6 Compatibility Mode' above
- **VolumeControl doesn't work or CRTFilter reports an error that CRTVolumeComponent can't be retrieved**
    - final volume with calculated values to be used for CRTFilter is retrieved from VolumeManager. Based on the specific settings of your project and/or usage of multiple cameras, there may be some issues connected with proper VolumeControl retrieval. Pls. adjust the code in the CRTRendererFeature.cs class in the AddRenderPasses and retrieve the volume based on your specific situation