CMDRSpirit / **URPTemporalAA**   Public

🔔 Notifications   ⑂ Fork **3**   ☆ Star **22**   ⌄

<> Code   ⊙ Issues **3**   � Pull requests   ⊙ Actions   ⊞ Projects   ⛨ Security   ⚲ Insights

⎇ main ⌄   ⑂ **2** branches   🏷 **2** tags   Go to file   Code ⌄

🔴 **CMDRSpirit** Updated readme with new suggested settings.   f3d7621 · on Oct 1   ⟳ **28** commits

📁 TemporalAA	Updated readme with new suggested settings.	last month
📁 res	Added reference to paper	7 months ago
📄 LICENSE	Create LICENSE	7 months ago
📄 README.md	Updated readme with new suggested settings.	last month

≣ README.md

Unity URP Temporal Anti-Aliasing

This is a Temporal Anti-Aliasing (TAA) solution for Unity's Universal render pipeline. URP does not have a TAA solution yet, so this may solve aliasing issues for devs using URP.

NOTE: URP does not support true motion vectors, so we rely on Neighborhood Clipping to deal with objects in motion. It should be fine though when the game runs at 60+ FPS, but this can definately depend on the use case.

This implementation is based on the Siggraph2014 talk by Brian Karis: High Quality Temporal Supersampling
https://de45xmedrsdbp.cloudfront.net/Resources/files/TemporalAA_small-59732822.pdf

Limitations

* Some pixel flickering at thin lines
* Does not work with active MSAA
* Only one camera with TAA is supported at the moment

Comparison

You can easily see that FXAA is more or less a blurry mess everywhere. SMAA is much cleaner but still has issues with very thin details, like the rope. TAA fixes those issues and efficiently super samples the details of the image.

A Video is available on my youtube channel: https://www.youtube.com/watch?v=0D_8q_3q0_s

Usage

* Render Pipeline Asset:
  * Make sure MSAA is disabled
  * Enable Depth / Opaque Textures
* Camera:
  * Disable any anti-aliasing method on your camera
* Renderer Asset:
  * Add Temporal AA Feature to your renderer
* Done!

Suggested Settings

* TemporalFade: 0.95 -> Lower leads to the current value being more represented in the final image, but more jittering is visible.
* MovementBlending: 100 -> Higher leads to more aggressive pixel rejection
* UseFlipUV: Sometimes, post processing and other settings flip the image. Toggle this to fix upside down results.

A Halton length of 8 should be enough (roughly 8x super sampling), larger values seem to make the jittering more obvious.

Requirements

* Unity 2021.2+ with URP 12 -> Should also work with most other versions, I just didn't test it.
* Unity.Mathematics (https://docs.unity3d.com/Packages/com.unity.mathematics@1.1/)

About

Temporal AA solution for Unity's Universal render pipeline

`unity` `anti` `urp` `aliasing` `taa`

📖 Readme
⚖ MIT license
☆ 22 stars
👁 2 watching
⑂ 3 forks

Releases 2

🏷 **Temporal AA v1.1** `Latest`
on Apr 24

+ 1 release

Packages

No packages published

Languages

● C# 75.9%   ● ShaderLab 24.1%