

Antialiasing Sample

*This sample is compatible with the Microsoft Game Development Kit (June 2020)*

# Description

# This sample shows different antialiasing methods (SMAA, SMAA2x, and FXAA).



# Building the sample

If using an Xbox One devkit, set the active solution platform to Gaming.Xbox.XboxOne.x64.

If using the Xbox Series X|S devkit, set the active solution platform to Gaming.Xbox.Scarlett.x64.

*For more information, see* Running samples*, in the GDK documentation.*

# Using the sample

|  |  |
| --- | --- |
| Action | Gamepad |
| Cycle AA techniques | A button / X button |
| Toggle hardware AA | B button |
| Cycle MSAA count | Y button |
| Select SMAA edge detection technique | DPad Left, Down, Right |
| Rotate view | Left thumbstick |
| Reset view | Left thumbstick (click) |
| Exit | View Button |

# Implementation notes

This sample implements post-processing shader techniques for anti-aliasing.

## SMAA

The SMAA algorithm is explained in detail at <http://www.iryoku.com/smaa/>. The scene is rendered and passed to the algorithm which is executed in 3 passes:

* Edge Detection Pass: This can be done using depth, luma or color values. Depth runs the fastest and gives better results.
* Blending Weight Pass
* Neighborhood Blending Pass

For SMAA 2x the scene is first rendered using MSAA 2x and then the above passes are run separately for each of the mesh renders generated from the multi-sample.

## FXAA

Just render the scene and pass it off to the shader. The FXAA algorithm is explained in detail at <http://developer.download.nvidia.com/assets/gamedev/files/sdk/11/FXAA_WhitePaper.pdf>.

*See the MiniEngine demo for an implementation of FXAA using DirectCompute*.

***Be sure to read ThirdPartyNotices.txt if you wish to implement these techniques in your title.***

See the **SimpleMSAA** sample for a demonstration of the basics of using the built-it multi-sampling hardware, and **Multisampling** for more detailed exploration of MSAA.

# Update history

The original version of the sample was written using the XSF-based framework. It was rewritten to use the ATG sample templates in June 2020.

# Privacy Statement

When compiling and running a sample, the file name of the sample executable will be sent to Microsoft to help track sample usage. To opt-out of this data collection, you can remove the block of code in Main.cpp labeled “Sample Usage Telemetry”.

For more information about Microsoft’s privacy policies in general, see the [Microsoft Privacy Statement](https://privacy.microsoft.com/en-us/privacystatement/).