

ShaderGlass v1.2

Overlay for running GPU shaders on top of Windows desktop

Copyright (c) 2021-2025 mausimus (mausimus.net)

Distributed under GNU General Public License v3.0

Features

- applies shader effects on top of desktop, in a floating window or full-screen
- built-in **RetroArch** shader library (1200+ shaders!) covering:
 - CRT monitor simulation
 - image upscaling
 - TV, VHS and handheld simulation
 - softening, denoising, blur, sharpen and many more
- works with most emulators, retro platforms and pixel art editors including:
 - **DOSBox**, **FS-UAE**, **Altirra**, **ScummVM**, **AGS**, **VICE**, **Aseprite** etc.
- excellent companion for pixel art showing shaded and/or aspect-ratio corrected preview
- you can even use it on top of YouTube, Twitch or modern games
- allows capture from a USB source (webcam or capture card)
- saving and loading profiles and custom shaders
- high customizability with various options, operating modes and shader parameters
- can be captured by OBS (using Game Capture source)

Download

GitHub: <https://github.com/mausimus/ShaderGlass>

Itch: <https://mausimus.itch.io/shaderglass>

Steam: <https://store.steampowered.com/app/3613770/ShaderGlass/>

Requirements

- **Windows 10, version 2004** (build 19041) or **Windows 11**
- will work on version 1903 but in limited capacity (no Desktop Glass mode)
- Windows 11 allows the **removal of yellow border** (see FAQ for tips on avoiding it on Win 10)
- DirectX 11-capable GPU

Menu Options

Currently supported options are:

Processing

- **Global Hotkeys**
 - **Enable** - toggle use of global hotkeys (if they conflict with another app)
 - ... - set global hotkey for each of the possible actions
- **Toggle Menu** - toggle menu bar ('m' or global hotkey); this will also work in fullscreen
- **Remember Position** - remember previous window position and size on startup
- **GPU** - shows the GPU which Windows assigned to ShaderGlass (read-only)
- **FPS** - by default ShaderGlass runs at V-Sync, you can reduce GPU load by lowering the frame rate (for pixel art etc.)
 - **Remember FPS** - tick so that ShaderGlass remembers and always uses current FPS setting (this overrides any preset)
- **Advanced** - extra options for lowering latency; these options require restarting ShaderGlass
 - **Use Flip Mode** - uses DXGI Flip Presentation mode (as opposed to BitBlt) which is faster but a little unstable (especially on Windows 10)
 - **Allow Tearing** - removes explicit V-Sync, required for VRR to work
 - **Max Capture Rate** - remove capture frame limit of WGC, only supported since Windows 11 24H2
 - **Use HDR** - use HDR-compatible RGBAf capture and display formats (experimental)
- **Load/Save/Recent Profiles** - you can save ShaderGlass' configuration into a profile file and load later
- **Set as default/Remove default** - make current settings the default when starting up

Input

- **Desktop** - captures the whole desktop or an individual monitor, defaults to Glass mode
 - **Lock Current Area** - always capture current area even if ShaderGlass window is moved
- **Window** - captures the selected window, defaults to Clone mode
 - **Crop** - remove border around captured window, if your emulator has a toolbar for example
- **Device** - captures an external/USB device (webcam or capture card)
- **File** - load a .png/.jpg/.gif file as input
- **Pixel Size** - indicates the size of input pixels (pre-scaling), i.e. if you run a game in x3 scaling mode set this to x3 as well so that ShaderGlass can tell the original resolution
 - **Adjust for DPI Scale** - if your source always applies Windows DPI Scaling when displaying images
(for example browsers) enable this option to take it into account; most emulators don't do this however

- **Capture Cursor** - whether to include mouse cursor in capture; this will not work 100% of the time unfortunately, it will only hide system cursors but not custom cursors some games set; see FAQ for workarounds
- **Remove Yellow Border** - disables yellow frame around captured area (forced security feature); currently only supported on Windows 11

Output

- **Mode** - overrides default mode for the input type:
 - **Glass** (default for Desktop) - ShaderGlass window appears transparent, you have to position it over the window or area you'd like to capture
 - **Clone** (default for Window) - ShaderGlass copies the content of capture so you can position it anywhere; this mode is also faster and more compatible
- **Window** - override mouse behavior:
 - **Solid** (default for Clone) - ShaderGlass window area is solid (traps mouse events), with the only exception of passing focus to captured window when clicked
 - **Click-through** (default for Glass) - ShaderGlass window area is transparent and clickable/scrollable-through to window(s) underneath
- **Flip** - flip output image horizontally and/or vertically
- **Scale** - apply additional scaling to the output if you'd like it to be larger; using no pre-scaling and only output scaling should result in best performance
 - **Free** - allow manual resizing of ShaderGlass window (Window Clone and File modes only)
 - **Retain** - prevent scale from resetting when switching inputs
- **Aspect Ratio Correction** - presets for common aspect ratio correction factors (DOS, etc.), applied horizontally to preserve scanline count

If you select *Custom* you can enter your own correction ratio expressed as pixel height (1.0 being square pixel). For example, in MS-DOS era a picture of 320x200 resolution (16:10) was displayed on a 4:3 screen meaning each pixel was 1.2x tall on screen, $1.2 = (16/10) / (4/3)$
- **Orientation** - direction of shader effect, "Vertical" emulates display rotated 90 degrees, like in arcade cabinets
- **Fullscreen** (Ctrl+Shift+G) - turn ShaderGlass into a topmost fullscreen borderless window, in Glass mode you will still see yellow outline around the screen but if you can use Window Glass (surrounding black bars) or Window Clone (top-left aligned) with your source then you can avoid yellow edges; press Ctrl+Shift+G to revert
- **Take Snapshot** - export current image in .png format

Shader

- **Choose from Library** - open Shader Browser to switch the current shader:
 - **Personal Favorites** - mark shaders as favorites using Add/Remove Favorite buttons
 - **Imported** - custom .slangp/.slang shaders you imported (these are NOT persisted between restarts)
 - **Community Favorites** - selection of popular shaders
 - **RetroArch Library** - built-in shaders from libretro repository
- **Next** - switch to the next Shader
- **Random** - choose a random Shader
- **Active** - temporarily switch to 'none' Shader to see the difference (hold TAB)
- **Import custom** - load and compile an external .slangp/.slang shader
- **Parameters** - show and modify active shader's parameters

You can save and load profile files which will store all the options. It's also possible to pass profile file path on the command line and ShaderGlass will load it on startup.

If you associate ShaderGlass with .sgp extension in Windows (double-click .sgp file and choose ShaderGlass) you will be able to start it up by double-clicking a profile file.

Command Line

```
ShaderGlass.exe [-p|--paused] [-f|--fullscreen] [profile.sgp]
```

You can pass profile filename as a command-line parameter to ShaderGlass.exe and it will be auto-loaded. If your profile file name contains spaces please put it in quotes.

In addition -p will launch in paused mode, and -f will launch in fullscreen mode.

Tuning

In order to achieve the best effect it's necessary to tune parameters to match your input:

- if you use scaling in your input, use nearest-neighbour (i.e. no smoothing) and set *Input -> Pixel Size* to match
- moving/resizing the window slightly might also help find the best fit for downscaling pixels
- use Window Clone mode if you can, it's fastest and most compatible; if you need mouse click-through use Glass mode

Always check that input is crisply pixelated using the "none" shader as there is often implicit scaling happening (for example even when Chrome displays an image at 100%, Windows DPI scaling is still applied). The none shader should display a pixelated image with no smoothing whatsoever, try to match Input Pixel Size setting with your input's size to achieve that.

Parameters

All shaders start using default Parameters as defined by their authors, but you can also tweak them using *Shader -> Parameters* menu option.

Emulators

Most emulators (DOSBox, FS-UAE, Altirra etc.) will capture mouse by default so you can use them in Window Clone mode. ShaderGlass window will remain topmost so just position it over the game window and Alt-Tab to the game to have mouse and focus captured.

DOSBox

DOSBox often defaults to exclusive fullscreen - for best results, run it in Windowed or Borderless Fullscreen.

You may also find mouse input doesn't work correctly with DOSBox. To remedy, tick Output -> Window -> Click-through in ShaderGlass.

ScummVM

ScummVM doesn't capture mouse cursor by default so for best results follow below steps:

1. Use Window Clone mode, set Input/Output/Shader options to your liking
2. Switch Output Window to Click-through
3. Click or Alt-Tab to ScummVM game window so that it has focus
4. Press Ctrl-M which will force ScummVM to capture mouse

This way you should have the mouse captured by ScummVM so that it remains within the game window until you press Ctrl-M again. You can apply output scaling and/or aspect ratio correction in ShaderGlass and enjoy the game.

Importing Custom Shaders

ShaderGlass comes with a large built-in library of shaders, but you can also import custom ones written in [RetroArch shader system](#). It accepts both individual shaders (*.slang*) and multi-pass presets (*.slangp*). When working on a custom shader, you can use *Recent imports* menu to quickly reload it.

RetroArch shader/preset packs

A lot of preset packs out there rely on base RetroArch shaders so to import them into ShaderGlass install them into RetroArch first, and then Import from RetroArch's shaders directory.

Please note some shaders might require Vulkan (ShaderGlass is DirectX 11 only) or rely on RetroArch features that ShaderGlass doesn't support so might not work just as well.

Writing your own shaders

ShaderGlass lets you easily experiment with your own shaders! *CustomShader.slang* is a minimal example of a shader that swaps color channels and applies a boost, meant as an example from which you could start writing your own shaders and have them applied to your desktop. Check out [RetroArch shader repository](#) for tons of examples.

Modifying RetroArch shaders

To modify shaders from RetroArch library, download their source code [from here](#), make changes and import.

Frequently Asked Questions

> I'm seeing black only

Make sure you're using **Windows 10, version 2004** (build 19041) or **Windows 11**.

> Can I use it with Reshade?

While there is no special support for it, you can inject Reshade into ShaderGlass just like into any other DirectX 11 app and stack Reshade shaders on top.

> I see yellow border around captured screen/window

Windows 10

Unfortunately yellow border is enforced on Windows 10 and there's no way to remove it. There are some ways to avoid it, for example:

- start up your game in Windowed mode
- in ShaderGlass choose Input -> Window -> choose your game's window
- click onto your game window to give it focus
- press Ctrl+Shift+G to make ShaderGlass fullscreen

Windows 11

It's possible to disable yellow border but only on Windows 11. If you are using Windows 11 and still seeing the border, open Windows Settings app and search for "screenshot borders". Make sure all the options and ShaderGlass are allowed to remove the border.

> I can't capture using OBS

OBS can capture ShaderGlass but only as a Game Capture source (not Window Capture nor Desktop Capture). Use "Capture specific window" within Game Capture to select ShaderGlass window only.

> Text/letters are all blurry or pixelated

For maximum sharpness set Input -> Pixel Size -> x1 but note that a lot of shaders expect pixelated input so set this option accordingly to each scenario.

> I don't see my favourite RetroArch shader

ShaderGlass includes majority of RetroArch repository at time of release, but some shaders had to be excluded because of features SG doesn't yet support. Differences versus upstream RetroArch shader repository are tracked [in this PR](#).

> ShaderGlass can't capture some windows

Unfortunately some old-style rendering windows are not supported by modern Windows Capture API. Please use Desktop mode to capture those.

> Mouse cursor isn't properly captured (duplicate and/or wrong image)

This is the case when a Windows-native game sets a custom cursor shape; ShaderGlass cannot override such cursors so they will remain visible. There are two possible workarounds:

- Use Window Clone mode - if you have another display, run your game in a window on that display and ShaderGlass on another display.
 1. Use Input -> Window with Capture Cursor enabled, and move mouse over to the game window, while watching the result in ShaderGlass window.
- Use a Virtual Display - install an open-source virtual monitor driver and run the game on there; then in ShaderGlass use Display Clone mode to see the game with mouse cursor properly captured:
 1. Download and install [Virtual Display Driver](#)
 2. Configure the virtual display to the same resolution and refresh rate as your main display using Windows Display Settings
 3. Start up your game in borderless fullscreen and use Win+Shift+Arrow keys to position it on the virtual display
 4. Set Input -> Desktop to the virtual display in ShaderGlass
 5. Set Output -> Mode -> Clone in ShaderGlass
 6. Make sure Capture Cursor is ticked
 7. Go to Fullscreen in ShaderGlass (Ctrl+Shift+G)
 8. Move mouse over to the virtual display and play the game

Virtual Display Driver can also be used to capture popup windows like menus or notifications that normally ShaderGlass cannot get in front of.

> A different issue...

Check [Issues section on GitHub](#) for known problems and workarounds.

Notices

- ShaderGlass application is provided under [GNU General Public License v3.0](#)
- Includes precompiled shaders from [libretro/RetroArch shader repository](#). Please refer to copyright notes within shader code for detailed copyright and license information about each shader.
- App icon courtesy of Icons-Land
- Big kudos to RetroArch team, emulator developers and the wide retro community!