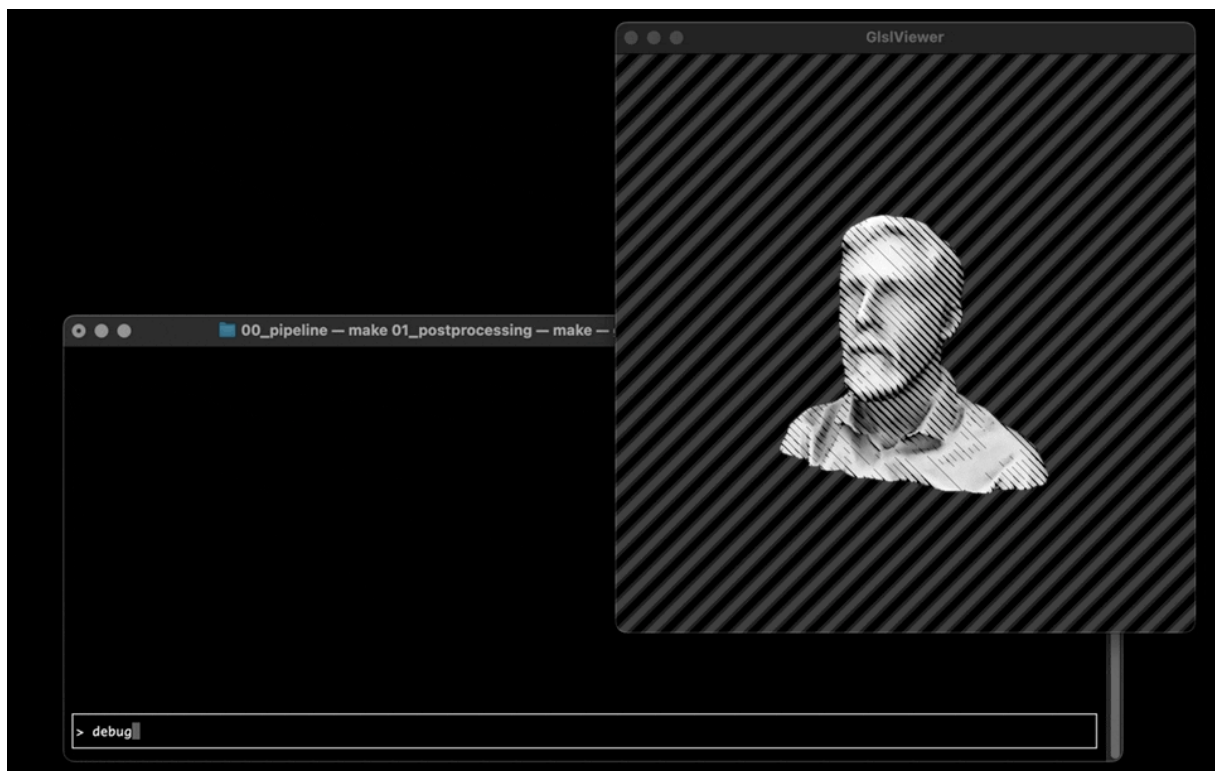


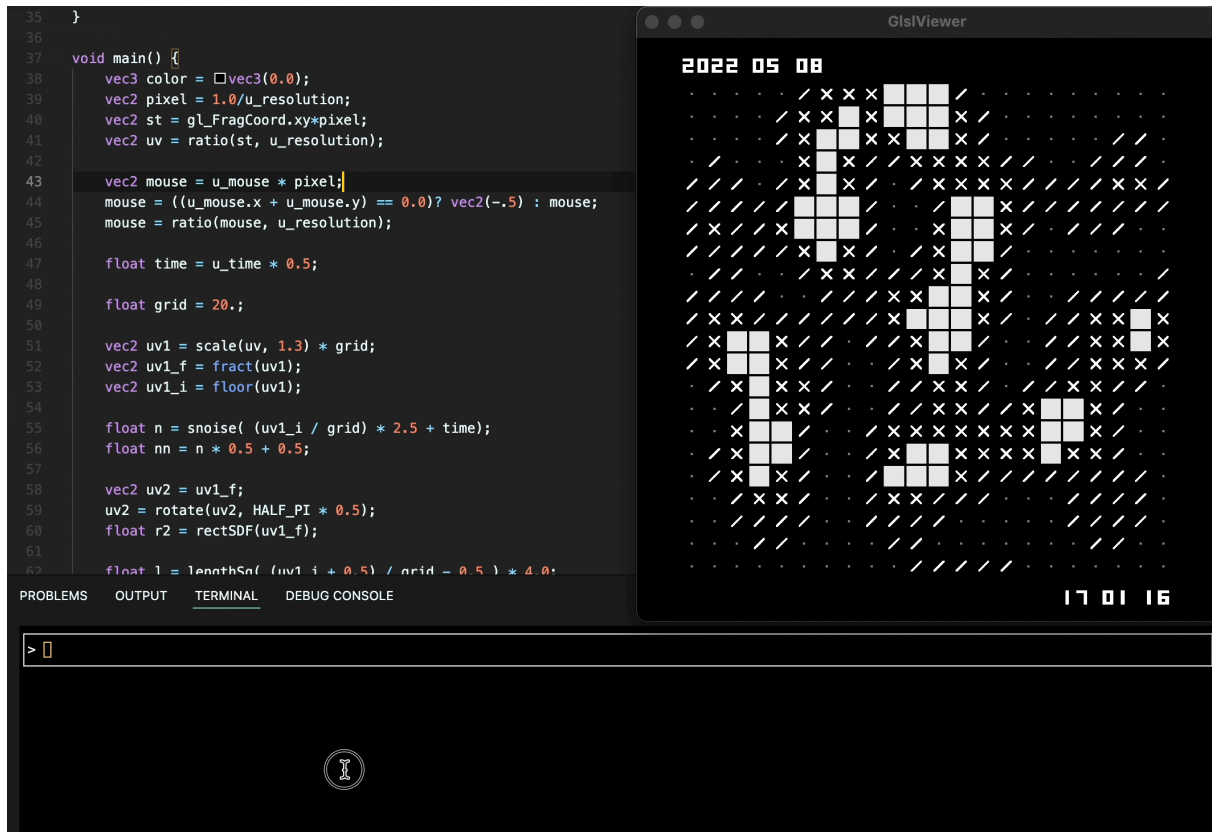
**glslViewer**  Sponsor  **Donate**



GlsViewer is a flexible console-base OpenGL Sandbox to display 2D/3D GLSL shaders without the need

of an UI. You can definitely make your own UI or wrapper using the Python Module (include) or any other tool that communicates back/forth with glslViewer through the standard POSIX console In/Out or OSC.

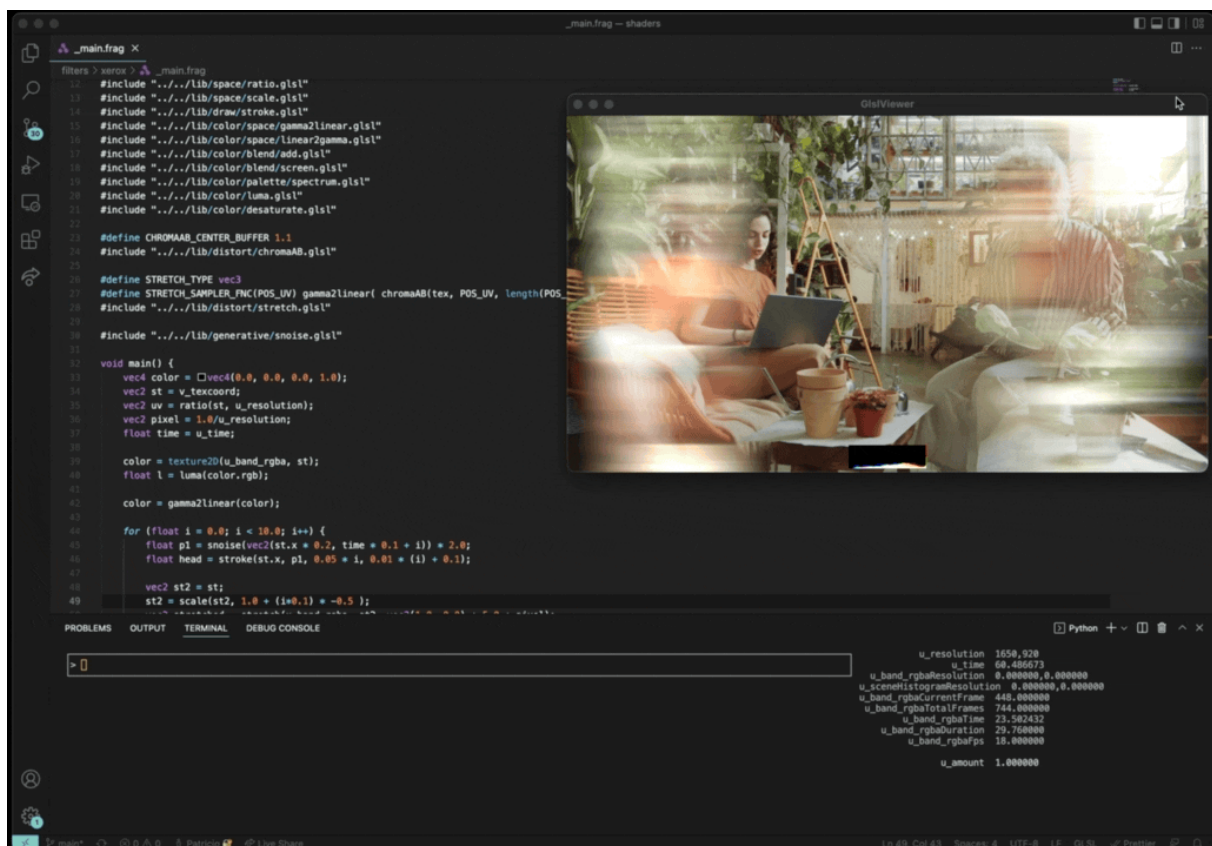
Joining #GslViewer channel on shader.zone discord to learn how to use it, share work and get help.

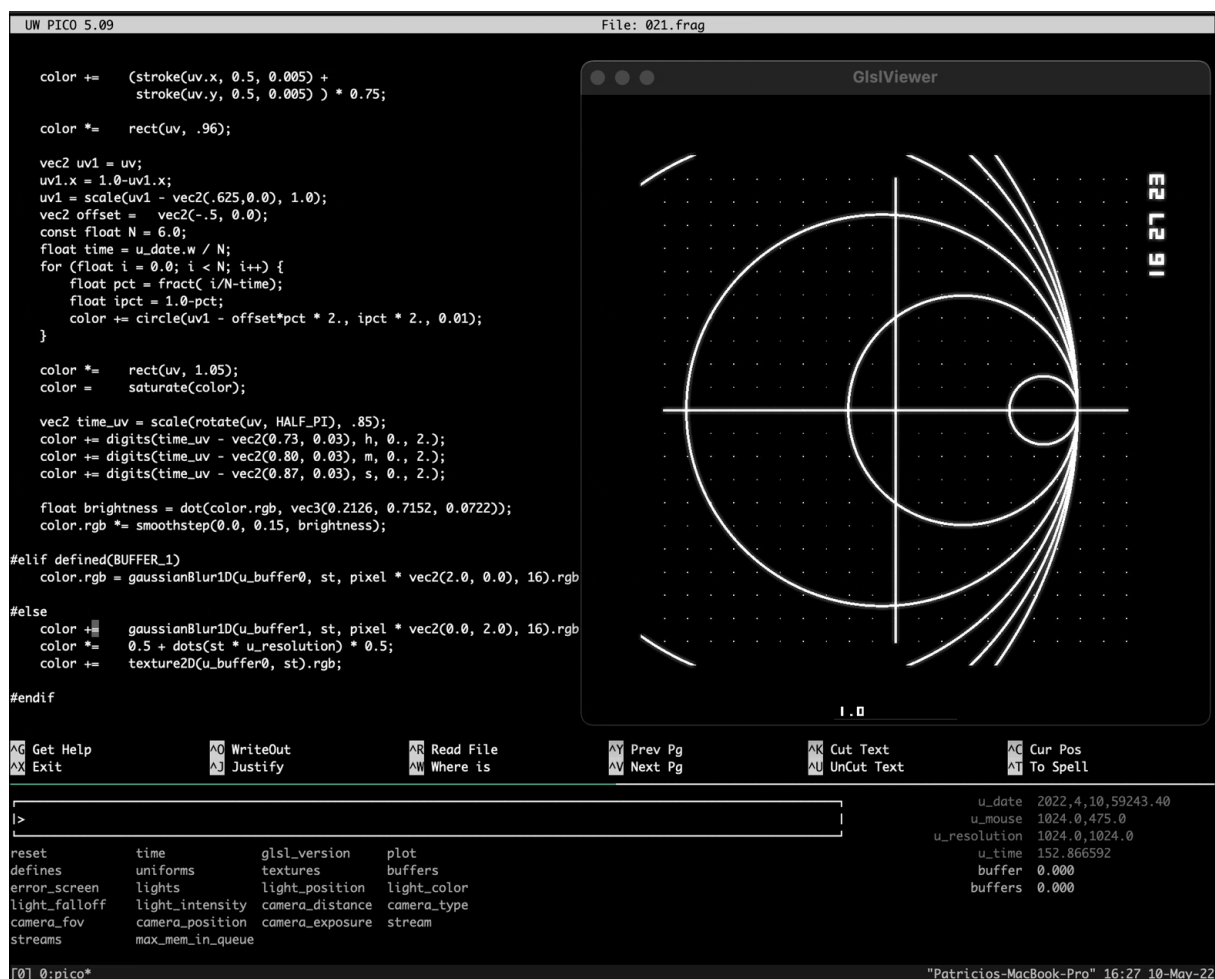


GslViewer gives support to:

- fragment and vertex shaders on GLSL
- resolve #include dependencies
- adding/deleting #define keys through console IN commands and OSC
- automatically generated set of defines based on the platform, buffer, render pass, geometry attributes and materials properties.
- passing custom uniforms (float, int, vec2, vec3 and vec4) through console IN or OSC
- Different kinds of Textures (png, bmp, jpg, tga, hdr, gif, mp4, mov, rtc, rtsp, local camera devices and audio textures)
- import of Cubemaps and spherical harmonics (png, jpg, tga, hdr)
- import LST, PLY, OBJ or GLTF files (and their dependencies)
- default vert/frag shaders for 2D shader and 3D material shaders with PBR lighting model
- hot reload of files on changes

- One default light and one default camera
- Interactive commands through POSIX console IN/OUT or OSC
- different debug modes (histogram, textures, buffers, bounding box, etc)
- shadow maps
- headless rendering
- fullscreen and screensaver mode
- HoloPlay rendering on LookingGlass Display
- image export
- PNG sequence export
- WASM crosscompiling





## Wiki themes

- Installing
  - Compile on Linux
  - Compile on MacOS
  - Compile on Windows
  - Compile as WebAssembly
- Using GlslViewer
  - Loading a single Fragment shader
  - Loading geometry and a vertex shader
  - Load a model and edit the default PBR shader
  - Loading Textures
  - Audio and video Textures

- 
- Other arguments
  - Console IN commands
  - Convention:
    - Defines
      - ★ Platform
      - ★ Buffers & Render Passes
      - ★ Geometry Attributes
      - ★ Materials
    - Uniforms

## Author

Patricio Gonzalez Vivo: [github](#) | [twitter](#) | [instagram](#) | [website](#)

## Acknowledgements

Thanks to:

- Mihai Sebea and Bertrand Carré for making the Windows compile happen
- Karim Naaki lot of concept and code was inspired by this two projects: fragTool and hdreffects
- Doug Moen he help to add the compatibility to ShaderToy shaders and some RayMarching features were added for his integration with his project: curv.
- Wray implement the OSC listener feature, opening GlslViewer to an entire new ways of communicating and interacting with other apps, devices and ecosystems.
- Yvan Sraka for putting the code in shape and setting it up for TravisCI.
- Andsz for Spherical Harmonics code from Spherical Harmonics Playground
- Syoyo Fujita for the work on tinyobjloader v1.0.x
- Morgan McGuire's for the OBJ models on Computer Graphics Archive
- Philip Rideout and Romain Guy general generosity to share their code and experience
- Sergei B for adding support for Audio textures