Functional 3D graphics for the browser in scheme

Theo Tyburn

TU Berlin ● Fraunhofer Institute ● Geometry School

Fri 13 March 2026

What is `sacr3d` ?

`sacr3d` is a scheme library for 3D graphics targeting the browser.

It focuses on

- Mathematical Visualization,
- Procedural Computer Graphics,

and features

- GPU programming in scheme,
- an embedded DSL for shader programming,
- compilation to WebAssembly (guile-hoot).

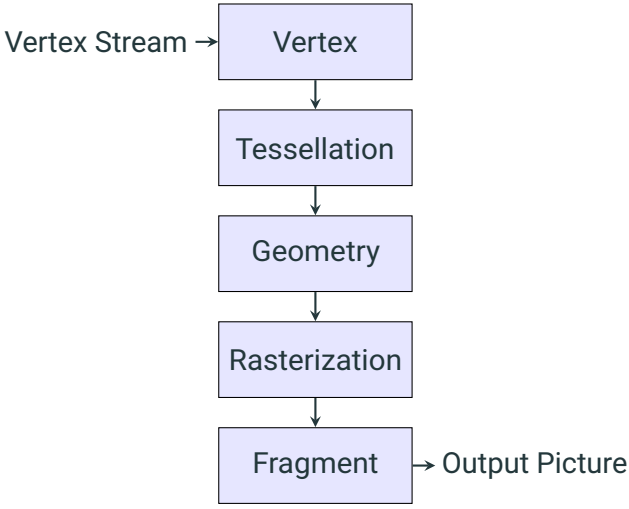
Design Goals

The design goals of `sacr3d` are

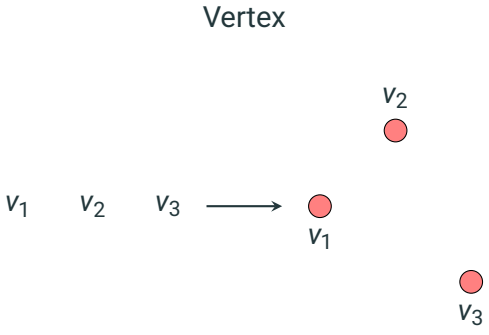
- Expressivity: embrace different visualization paradigms,
- Portability: share visualizations as web applications,
- Modularity: reuse graphics pipelines and mathematical libraries,
- Simplicity: functional graphics pipelines.

Non-goal (yet): Performance.

The OpenGL graphics pipeline

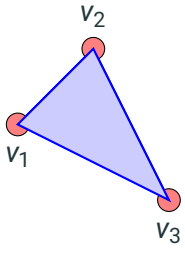


Hello Triangle in the OpenGL graphics pipeline



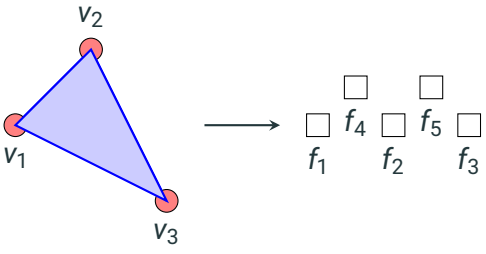
Hello Triangle in the OpenGL graphics pipeline

Tesselation



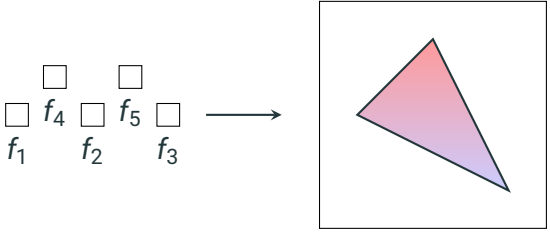
Hello Triangle in the OpenGL graphics pipeline

Rasterize

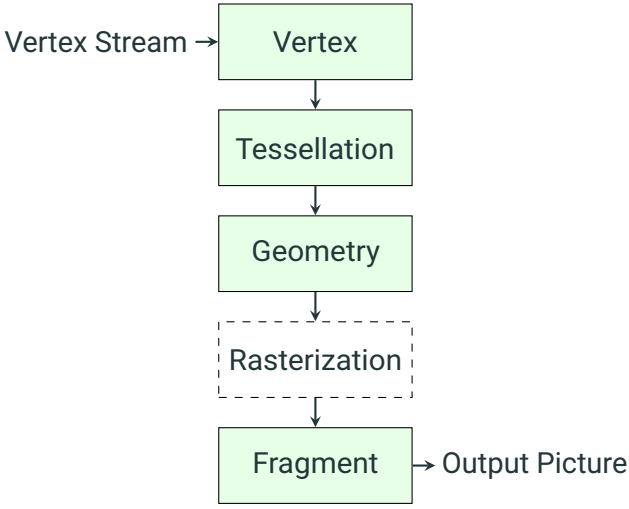


Hello Triangle in the OpenGL graphics pipeline

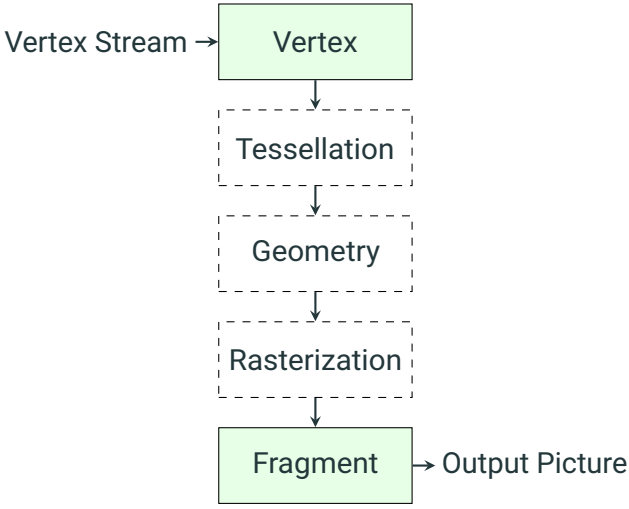
Fragment



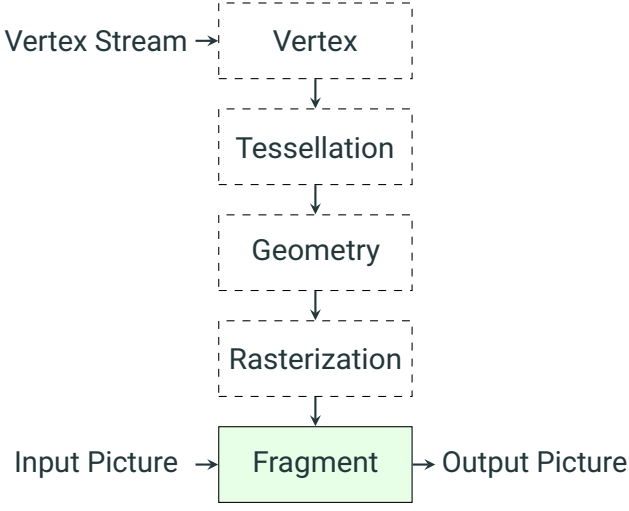
The programmable OpenGL graphics pipeline



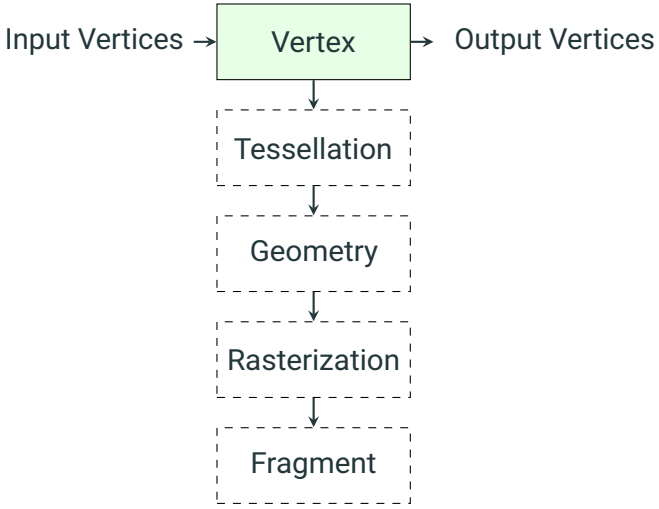
The programmable sacr3d graphics pipeline



The image processing graphics pipeline



The particle system graphics pipeline



Practice

Let’s try it out!

- Source : `https://codeberg.org/theottm/sacr3d`
- Branch : `bobkonf-2026`
- File : `examples/template.scm`
- Code snippets : `examples/snippets.scm`

Setup

- Download Docker image: `docker pull teddd/sacr3d`
- Clone repository: `git clone https://codeberg.org/theottm/sacr3d`
- Inside the repository build the example: `docker run -rm -v $(pwd):/project teddd/sacr3d make -C project build build-source=examples/template`
- In another terminal, run the webserver: `docker run -rm -v $(pwd):/project -p 8088:8088 -it teddd/sacr3d make -C project serve`

Vertex shader

Drawing using parameterizations.

Fragment shader

Drawing using screen coordinates.

The Julia set fractal

Screen coordinates as complex numbers.

Dynamical Systems

Feedback loops.

The future

- Particle systems
- High level rendering API (scene graph, cameras, lights, material)
- Extend eDSL: Proper Interpreter, Functional Reactive UI.

Useful links

- Project page : `https://teddd.srht.site/sacr3d/`
- Source : `https://codeberg.org/theottm/sacr3d`
- GLSL docs: `https://docs.gl`
- Shadertoy : `https://www.shadertoy.com/`

The Geometry School

We offer workshops in geometry and visualization.

`https://geometry-school.com/`

Contact us if you are interested !

THANK YOU FOR YOUR ATTENTION!