PORTING AN EDUCATIONAL GRAPHICS ENGINE TO A WEB PLATFORM USING WEBGL

Eldar Satyndiev

INTORDUCTION

- OpenGL ES 2.0 (C++)
- Educational tool for Computer Graphics
- Core principles (Shaders, Vertex, Fragment, Transormations)

PROBLEM

Development Environment

Prepare the development environment at home.

- Install the latest drivers for your GPU.
- Install Git VCS https://git-scm.com.
- Install Conan Package Manager https://conan.io.
- Install CMake Build System https://cmake.org.
- On macOS, install Xcode IDE https://developer.apple.com/xcode.
- On Windows, install Visual Studio 2022 Community IDE https://visualstudio.microsoft.com legally.

You can use any other IDEs (like CLion) or code editors (like VS Code, Vim, Emacs) that you lik provide support for Visual Studio 2022 on Windows and Xcode on macOS.

You can also work on Linux, but you must figure out the driver/tools/editor-installation process

Installation

Debug

```
/Applications/CLion.app/Contents/bin/cmake/mac/aarch64/bin/cmake -DCMAKE_BUILD_TYPE=Debug -DC
CMake Error: CMAKE_PROJECT_TOP_LEVEL_INCLUDES file does not exist: conan_provider.cmake Debug
CMake Error: CMAKE_C_COMPILER not set, after EnableLanguage Debug
CMake Error: CMAKE_CXX_COMPILER not set, after EnableLanguage Debug
CMake Error at CMakeLists.txt:10 (find_package): Debug
  By not providing "FindSDL2.cmake" in CMAKE_MODULE_PATH this project has
  asked CMake to find a package configuration file provided by "SDL2", but
  CMake did not find one.

  Could not find a package configuration file provided by "SDL2" with any of Fix...
  the following names:

    SDL2Config.cmake
    sdl2-config.cmake

  Add the installation prefix of "SDL2" to CMAKE_PREFIX_PATH or set
  "SDL2_DIR" to a directory containing one of the above files.  If "SDL2"
  provides a separate development package or SDK, be sure it has been
```

Errors

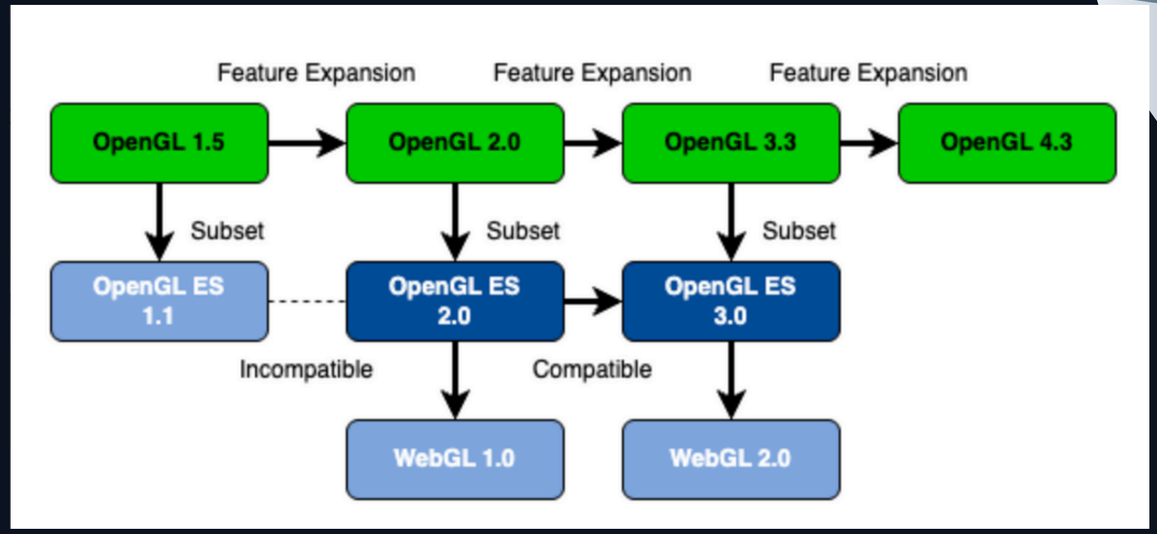
SOLUTION

Simplify the engine by porting it to Web platform, allowing students to run it directly in a browser with no installations.

LITERATURE REVIEW

Main Objective:

- No installation required
- Full browser compatibility
- Easy setup for students



Vulkan	Three.js	WebGL
Metal	Bobylon.js	WebGPU

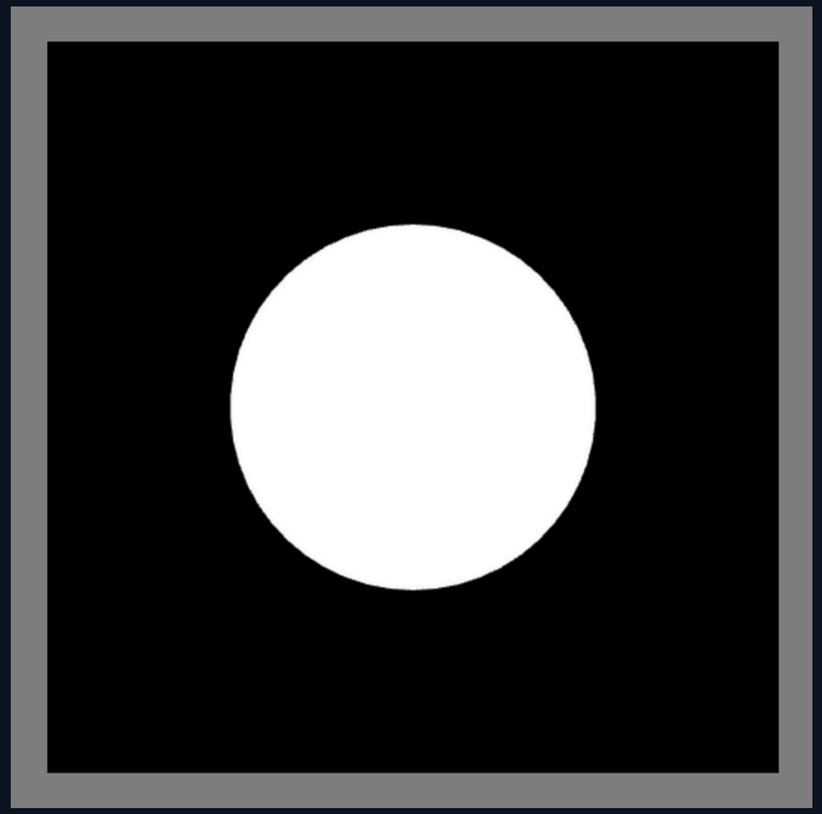
ENGINE STRUCTURE

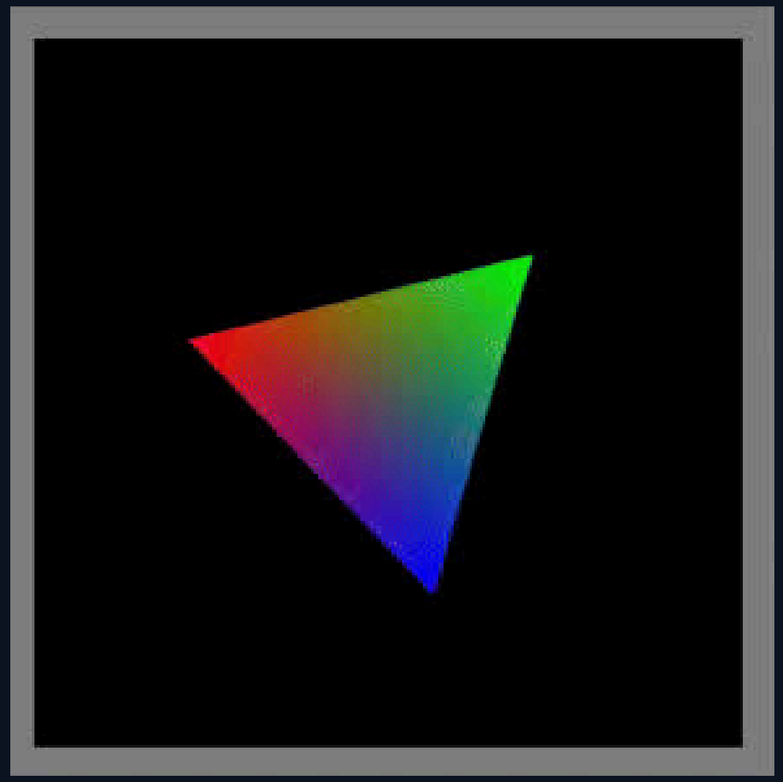
Versions:

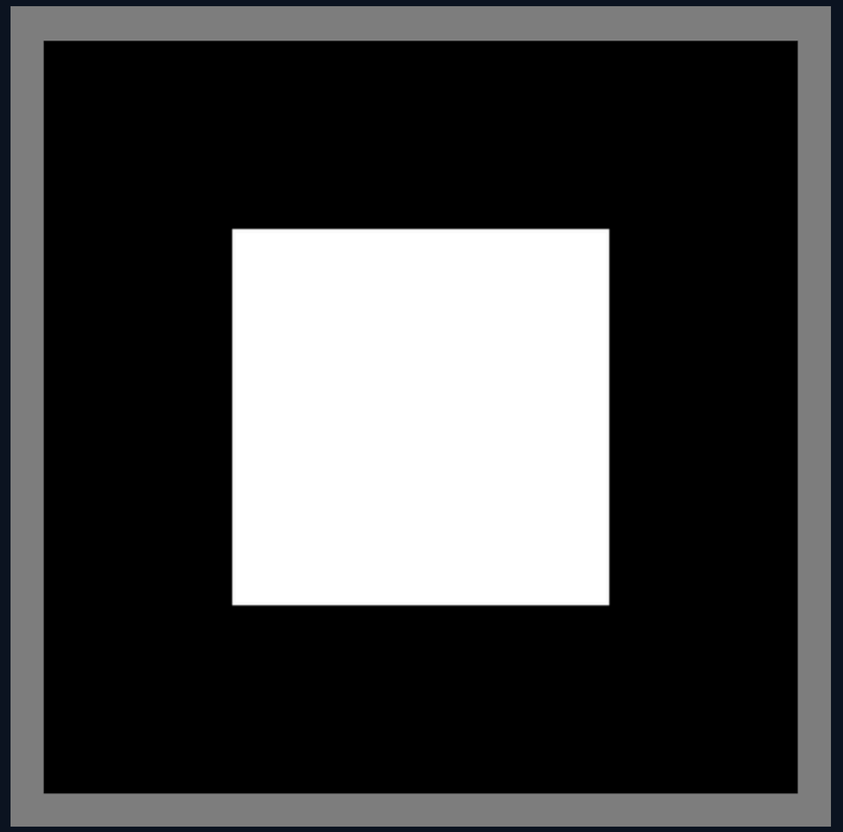
Incremental builds, each adding new features:

- **asr-1.0 – Basic 2D rendering**
- **asr-1.1 – 3D geometry + indexing + keyboard handling + matrix usage support**
- **asr-1.2 – Texture support**
- **asr-1.3 – Instancing and lighting**

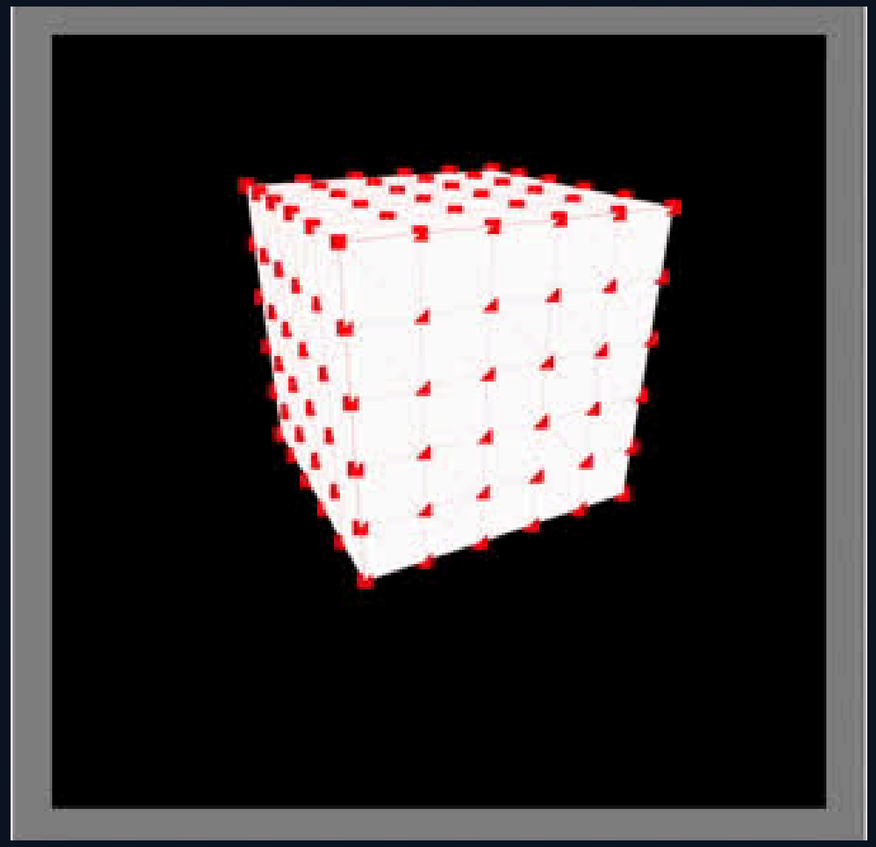
VERSION 1.0

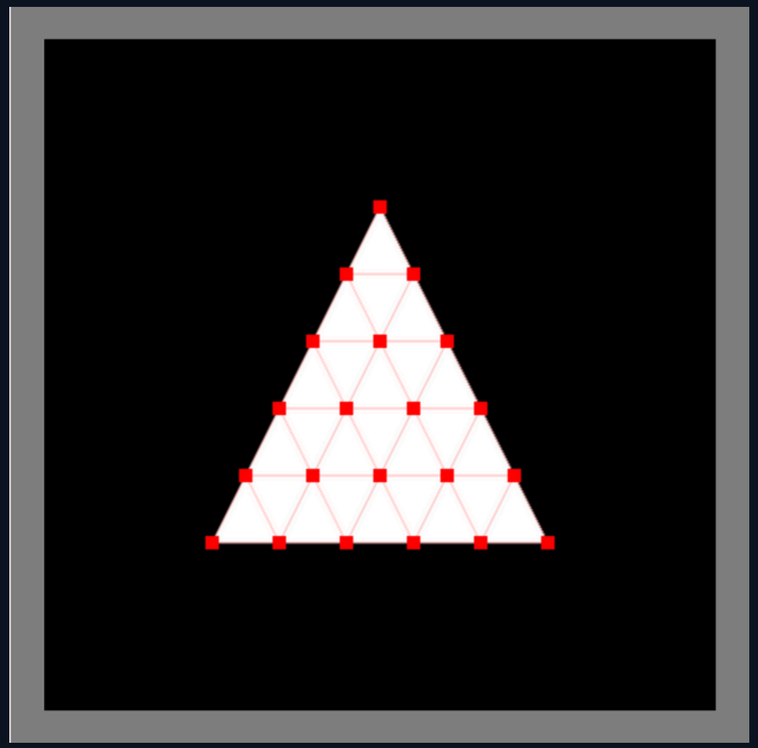


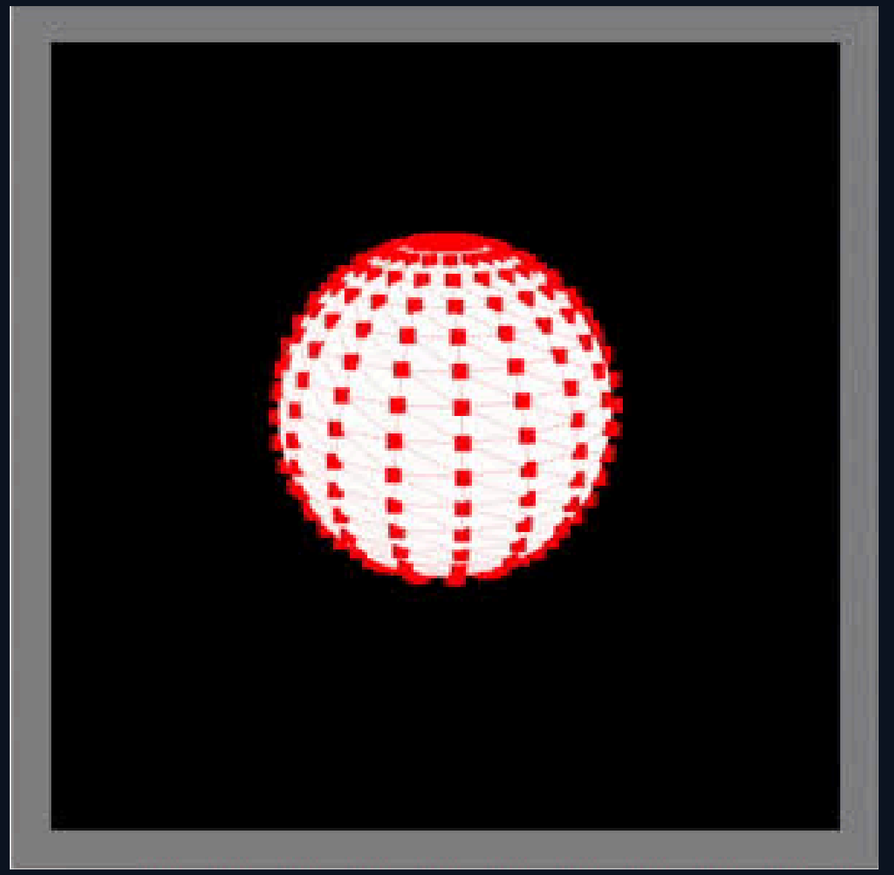




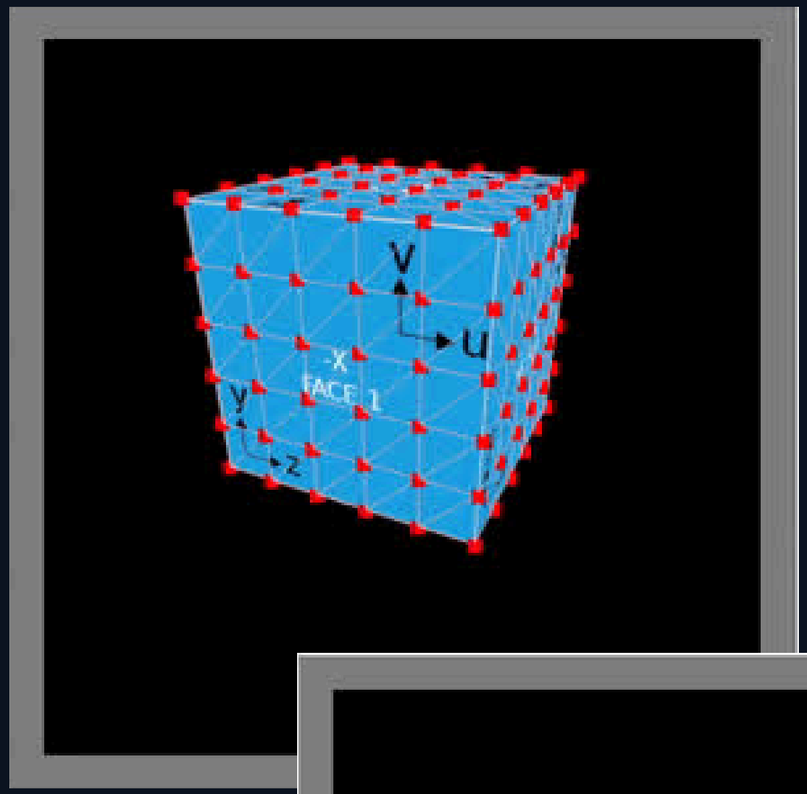
VERSION 1.1

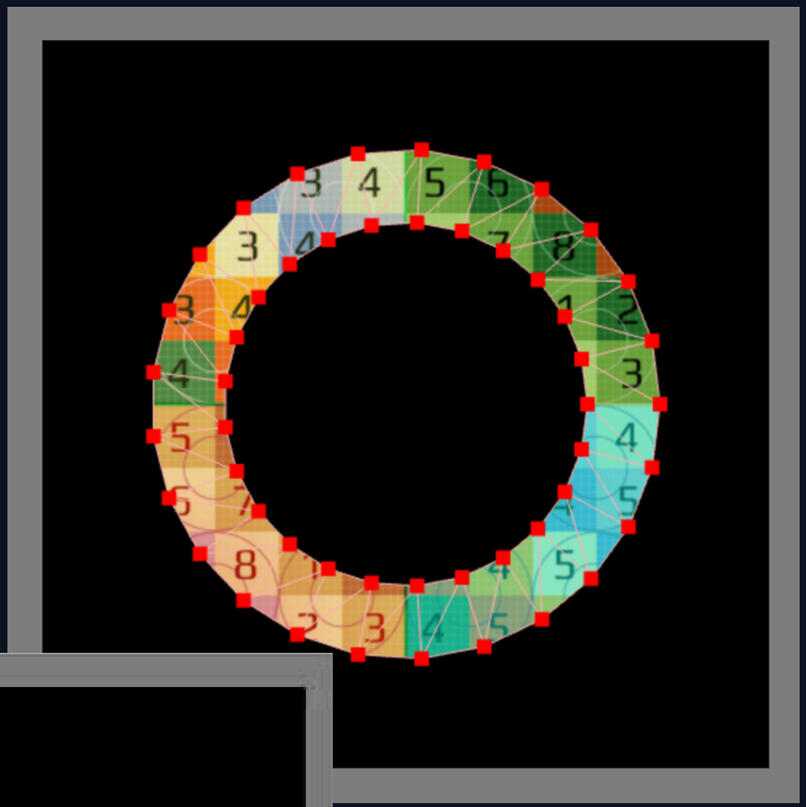


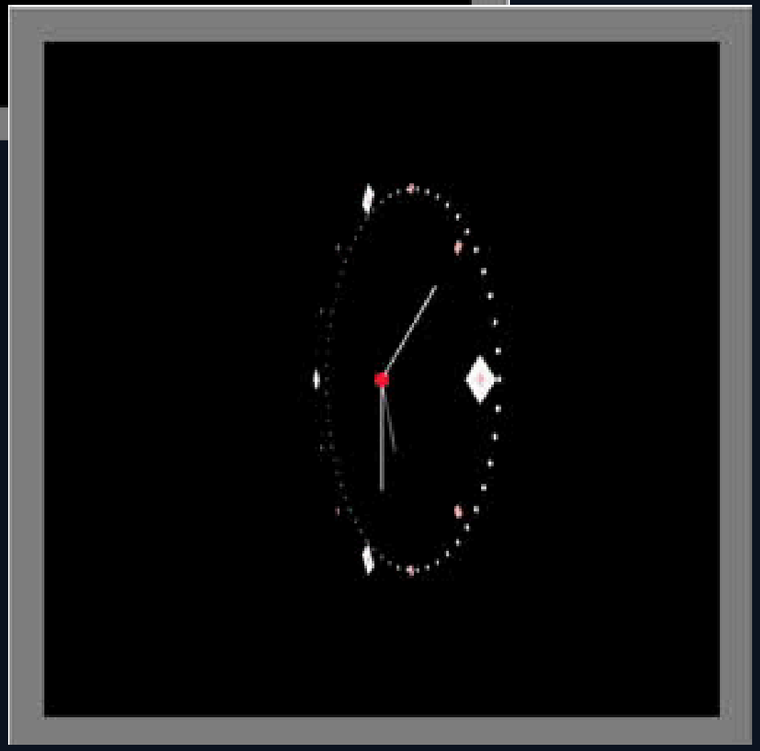


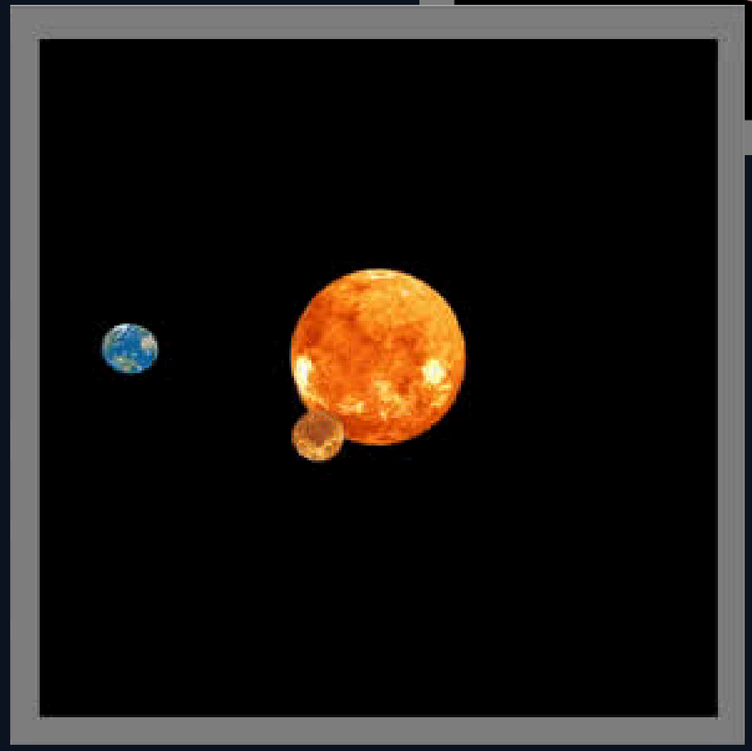


VERSION 1.2

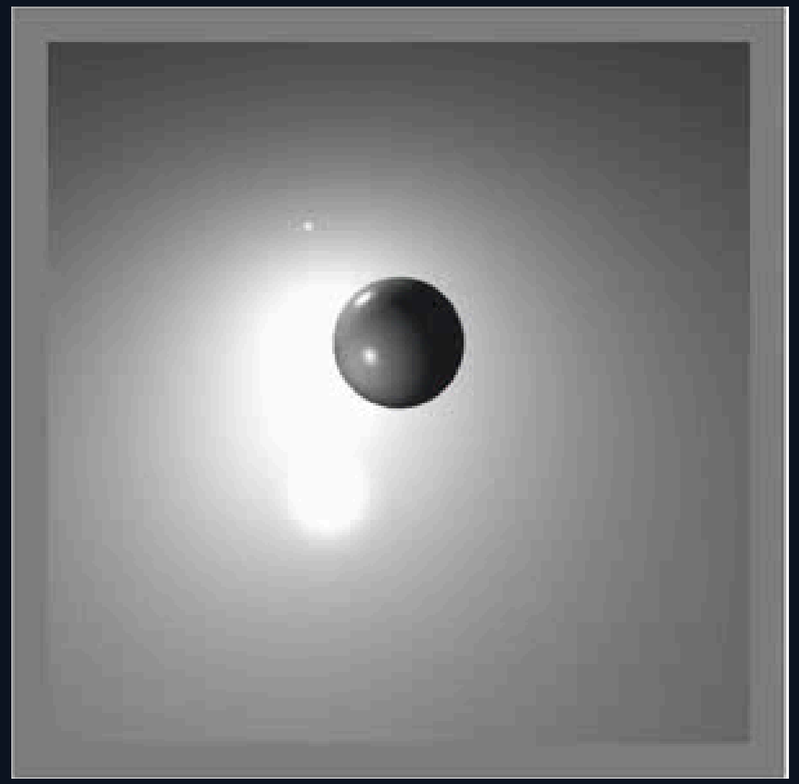


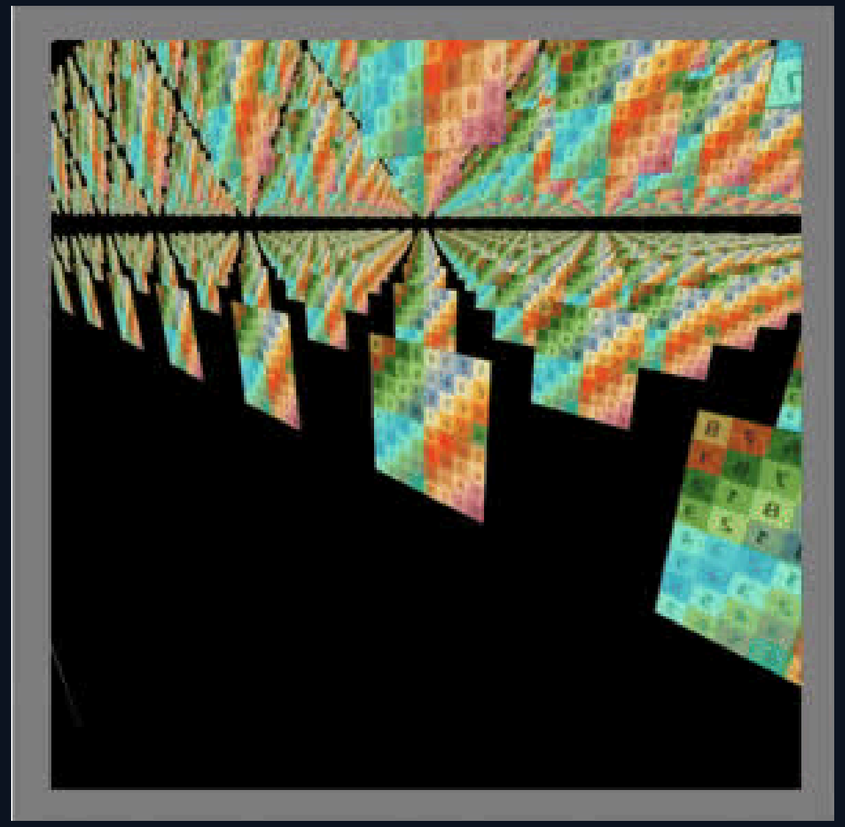






VERSION 1.3





MAJOR CHALLENGES

- JavaScript lacks built-in support for GL-style math
- Complex handling of various uniform types
- Keyboard handling
- Browsers restrict file access (CORS issues)

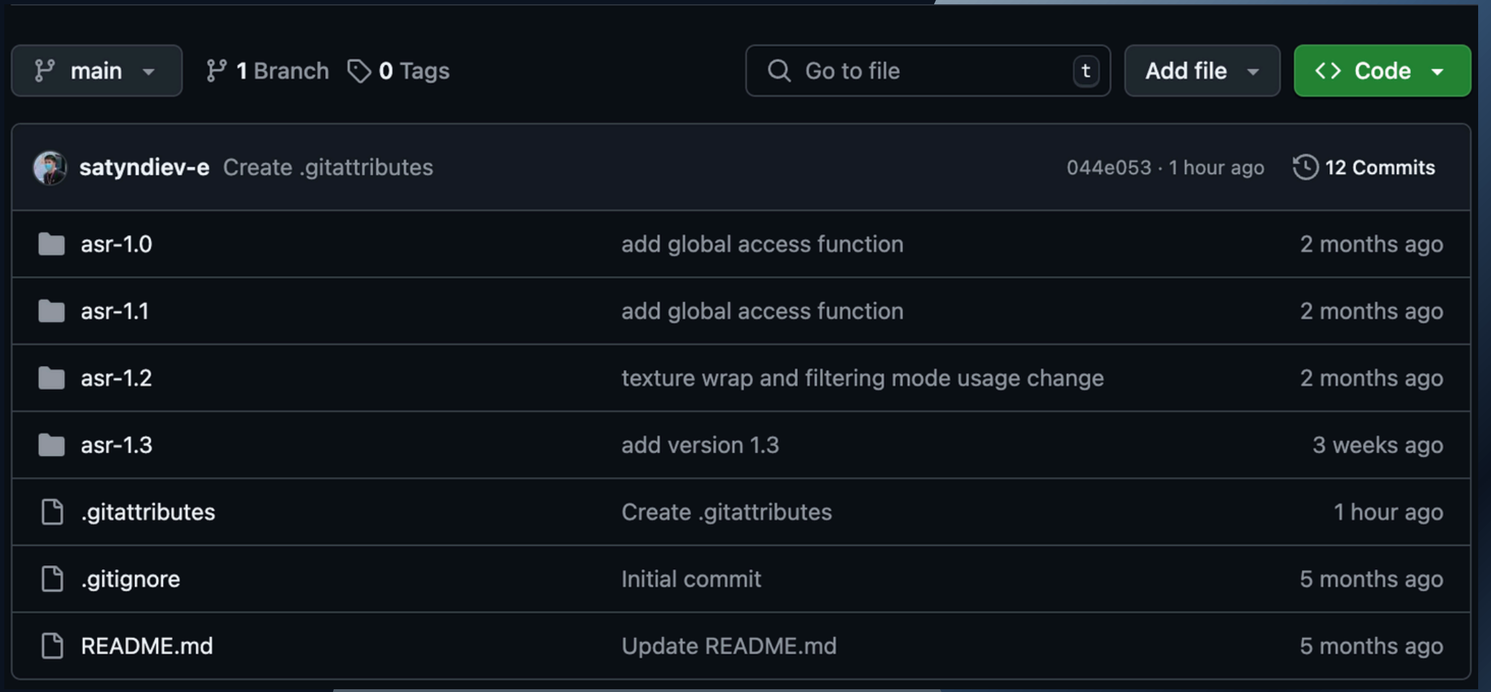
RESULTS

Fully Functional WebGL Engine

- All versions migrated, ready for educational use

Runs Without Server

- Students can launch the engine by simply opening the HTML file in a browser
- No setup or local server needed



FUTURE WORK

Modular Expansion of the Engine

- Future versions could include UI panels, object manipulation tools, or real-time editors

Exploring Libraries like Three.js

- It could be used for creating a more complex visual version of the engine in future iterations

DEMONSTRATION

THANK YOU FOR YOUR ATTENTION

satyndiev_e@auca.kg