The University of Pennsylvania
Department of Computer Science
CIS 400/401 Senior Design 2017 – 2018

Exploring High Performance Graphics with Modern Web Technology

Austin Eng (aen@seas.upenn.edu)
Advisor: Patrick Cozzi (pjcozzi@gmail.com)
Group #12

OBJECTIVE

Create a high-performance graphics application using modern web technologies to demonstrate the viability of the web as a platform capable of computationally intensive applications.

MOTIVATION

- The web has historically been a low-performance platform used mainly for display and delivery of simple content and media.
- The advent of newer web technologies offer the potential for superior *near-native performance* on an exceptionally *accesible* and *portable* platform.
- High performance graphics applications are currently limited to desktop and game-console platforms.

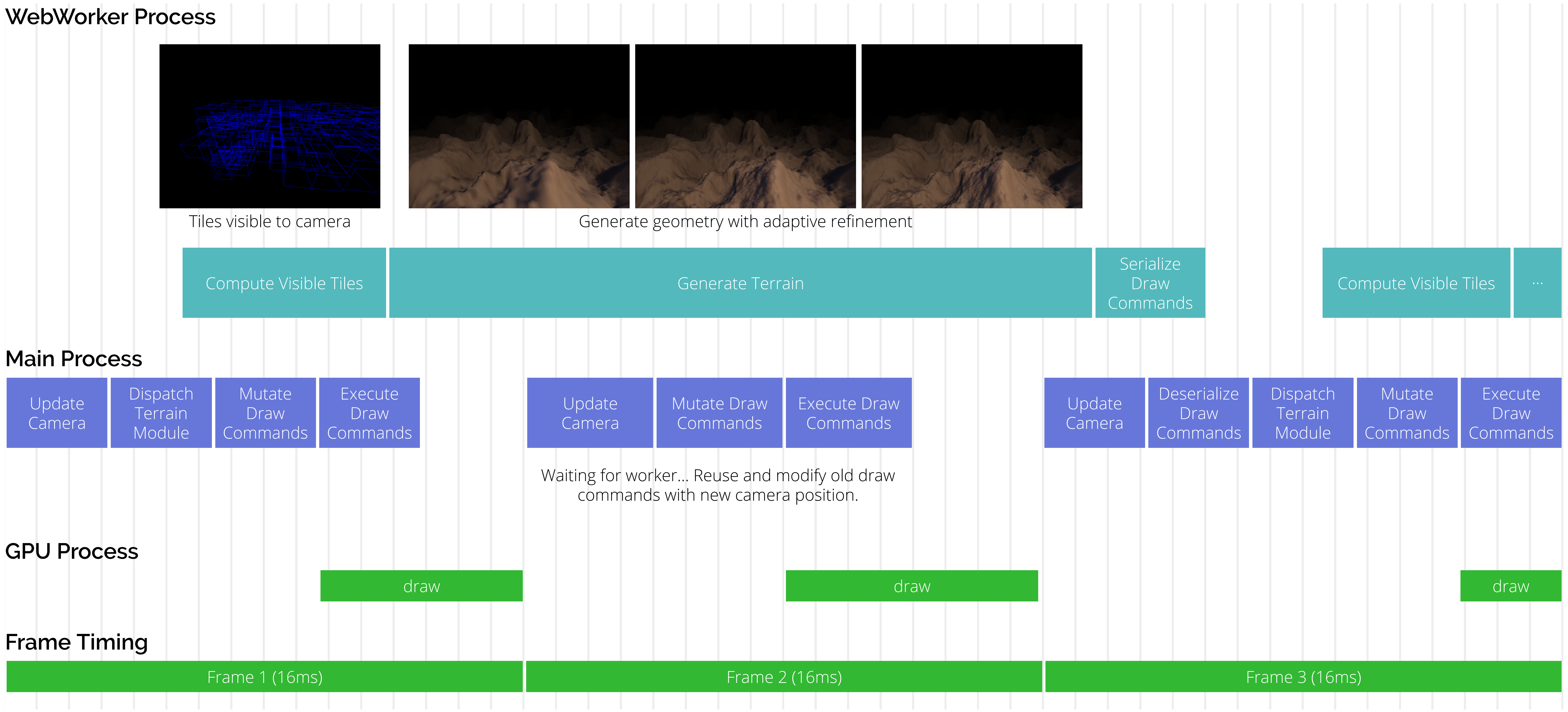
GOALS

- Build a sample application which utilizes WebWorkers and WebAssembly to demonstrate increased computational power on the web.
- Understand and investigate best practices and techniques for maximizing framerate in online graphics applications.
- Compare and evaluate WebAssembly and JavaScript performance.
- Investigate WebWorkers for multiprocess computation and evaluate performance implications.

IMPLEMENTATION

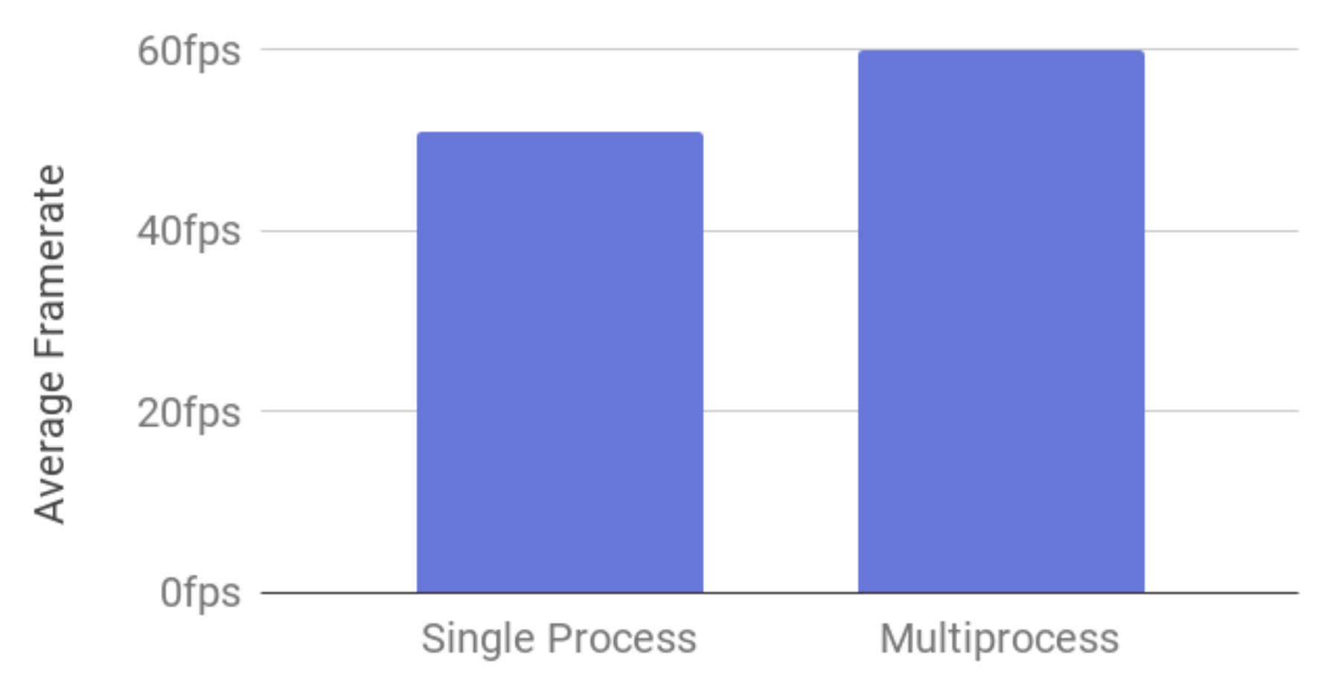
C++ → LLVM → WebAssembly

Code is compiled using the Emscripten toolchain. This first compiles code to LLVM with clang, allowing for significant compiler optimizations, before outputing WebAssembly.



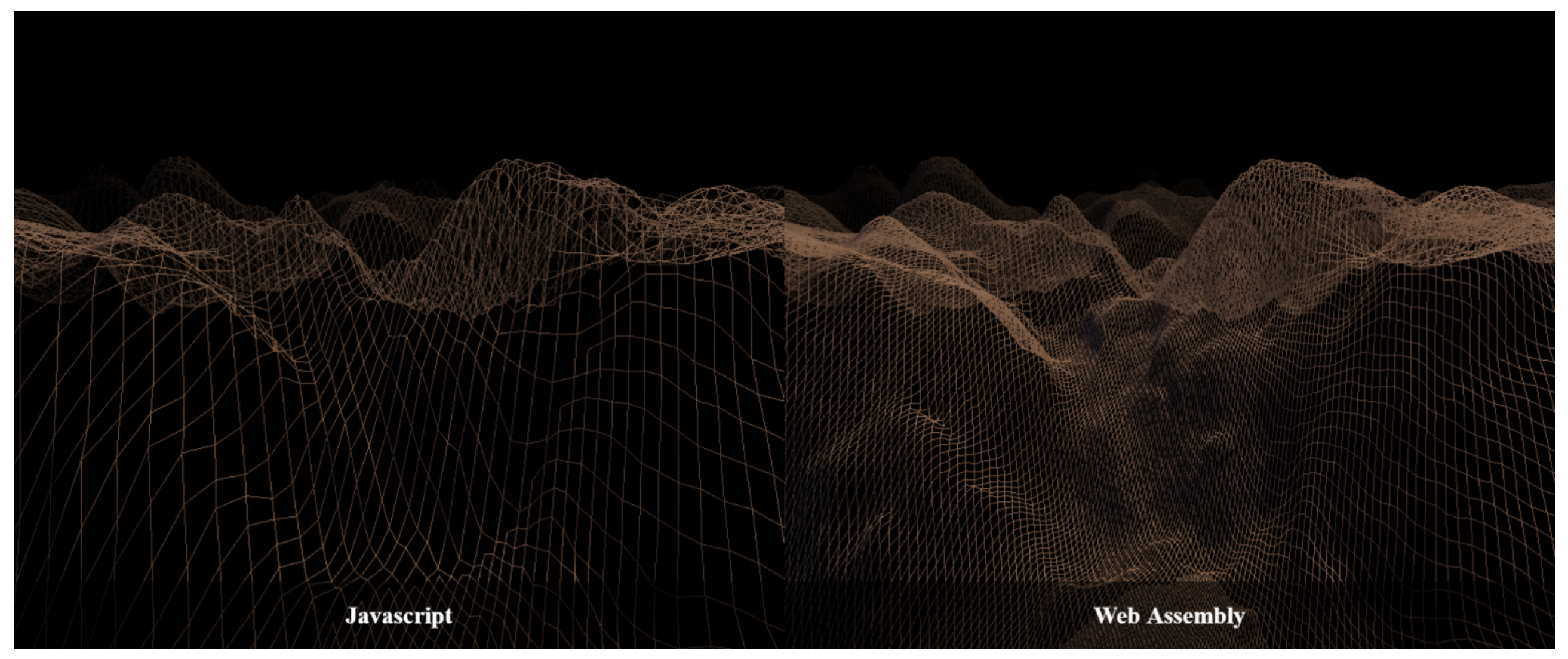
EVALUATION

Lightweight Main Process with Command Reuse

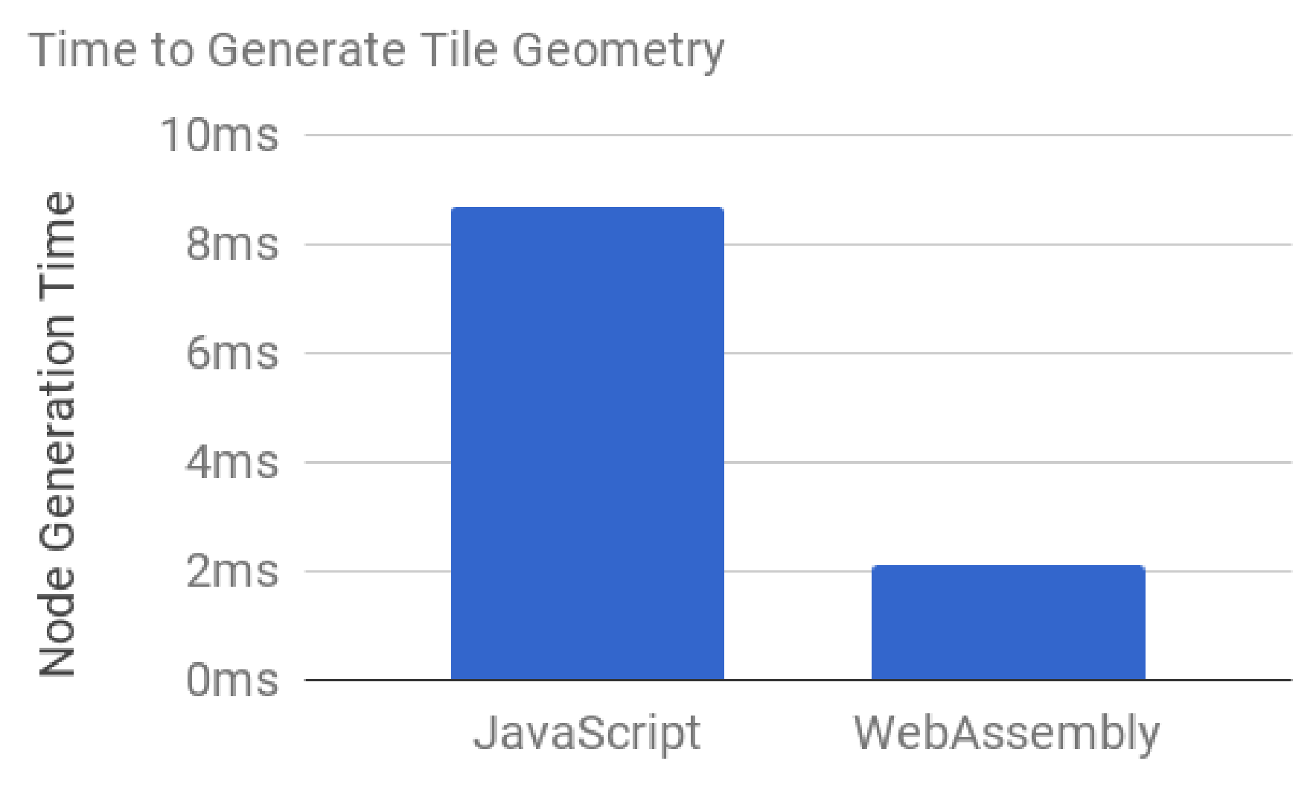


The main process strictly handles camera interaction and submitting commands to the GPU. Old commands are reused and "warped" if the new frame is not ready.

JavaScript vs. WebAssembly



Comparison of the rendering quality of JS and WASM builds. WebAssembly generates content faster and is able to output higher resolution geometry.



Content generation with WebAssembly is roughly four to five times faster than JavaScript.