An Interactive Introduction to WebGL

Educators Forum SIGGRAPH 2020

Edward Angel
University of New Mexico
angel@cs.unm.edu

Dave Shreiner
Unity Technologies
shreiner@siggraph.org

ACM Reference Format:
Edward Angel and Dave Shreiner. 2020. An Interactive Introduction to WebGL: Educators Forum SIGGRAPH 2020. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Educator's Forum (SIGGRAPH '20 Educator's Forum), August 17, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3388530.3408878

1 AN INTERACTIVE INTRODUCTION TO WEBGL

For more than 15 years OpenGL and its derivatives have been the most widely used APIs for creating interactive computer graphics applications across all major platforms. Their uses span virtually all application areas and reveal the most up-to-date features of modern graphics hardware.

Over the past four years there has been extraordinary interest in WebGL, the JavaScript implementation of OpenGL ES 2.0. Applications developed using the WebGL run in all major browsers. Programs can be hosted remotely, downloaded using a URL, and run locally, and still take advantage of all a system's local resources, in particular its GPU. In addition, applications can be integrated with other Web content and libraries. Higher level APIs such as three.js have been built on top of WebGL.

This course provides an accelerated introduction to WebGL. Our goal is to present participants with the capabilities of the API and help them navigate the plethora of available APIs. We will present simple coding examples using WebGL. Although we do not expect participants will know all the functionality of the API after a few hours of the course, we expect that they will leave knowing its capabilities and how to start on the path of creating their own applications.

This version of the course—introductory OpenGL courses have been presented at most SIGGRAPHs and SIGGRAPH Asia's for over 15 years—will have a significant focus on using shaders to make full use of the GPU through the WebGL 2.0's shaders. WebGL application programs allow the application to access the GPU directly, through the use of buffers and shaders. Although most attendees at recent courses have some familiarity with OpenGL/WebGL, many of them are still users of older versions of the API and others see the API only indirectly through a high-level API, such as three.js. Consequently, the majority of the attendees have little direct experience with shaders and thus cannot exploit the full capabilities of the API and their GPUs.

The course starts with a brief overview of the architectural models and development of the OpenGL family of APIs. Next, we present an introduction to WebGL programming, focusing on a simple but complete example. It will show on how a typical WebGL application is comprised of HTML, JavaScript, and GLSL files.

We will provide an introduction to the OpenGL Shading Language (GLSL). Participants will have access to all the example code on the web and will be able to see the full code and run the examples during the course.

The second half of the course will focus on application areas that show the power of the API. Examples will be a level that can be understood with a knowledge equivalent to the fundamentals course and some programming experience. These include:

- Transformations: A quick overview of the usual coordinate systems used in OpenGL and how the matrices are sent to the shaders. Examples will focus on lighting and lighting models can be implemented in either the vertex shader or the fragment shader.
- Texture mapping: Overview of 2D and 3D texture mapping with an example of 3D texture mapping used in medical imaging.
- GPGPU: In this section, we will show the advantages of using the multiple computing units in GPUs for scientific computing with image processing examples.
- Off-screen and multi-rendering: Many of the most powerful applications of WebGL are based on being able to render a scene multiple times, usually employing rendering to an off-screen buffer whose contents can be used as texture for the next rendering to the visible frame buffer.

We wrap up with a discussion of the status of WebGL 2, the differences between the WebGL and other related APIs such as three.js. We will also discuss how these APIs fit in with other graphics APIs such as desktop OpenGL, Direct X, Vulkan and Metal. Finally, we will provide resources for support of topics we covered.

SIGGRAPH '20 Educator's Forum, August 17, 2020, Virtual Event, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7966-3/20/08.
https://doi.org/10.1145/3388530.3408878