Three.js fundamentals

Beuren F. Bechlin @bfbechlin

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INSTITUTO DE INFORMÁTICA UFRGS

Summary

- Prerequisites
  - OpenGL 2+ knowledge
  - Visual Studio Code (preferable)
    - Live Server Plugin ritwickdey.liveserver
  - Chrome (preferable)
    - Three.js Developer Tools jsantell
- Examples
- Background
- WebGL
  - Example
- Three.js
  - Tutorial
  - Developer Tools
  - Caveats

Examples

- Explorable Videos
- Interland Game
- Google Cloud Infrastructure
- Harp API Maps
- Google experiment

- Three.js:
  - Teapot
  - Bumpmap
  - Envmaps
  - Ocean

Background

- HTML
  - Basics
  - Metadata
  - Vector images: optional.
- Javascript
  - Overview
  - First Code
  - Prototypes: optional.
  - Events
  - Asynchronous
  - Event Queue: event-queue.html
- Web-API
  - Introduction
  - Drawing graphics

WebGL

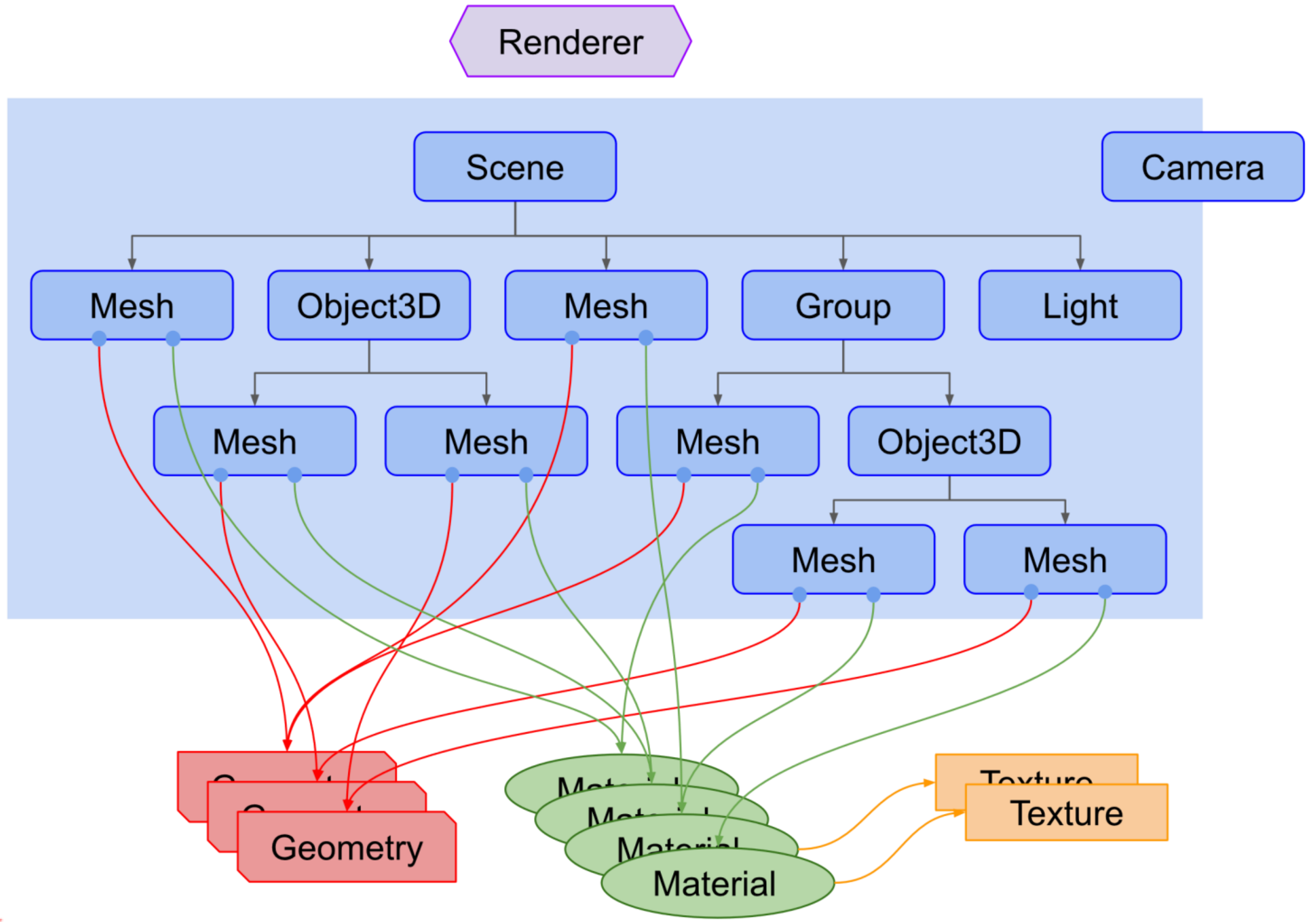
- Around 2006–2007, Mozilla started work on an experimental 3D canvas implementation.
- This became WebGL, which gained traction among browser vendors, and was standardized around 2009–2010.
- Development of the **WebGL 2** specification started in 2013 with final in January 2017.
  - Based on OpenGL ES 3.0.
  - Implementation of newer OpenGL specifications such as 3D textures, Multi-Sampled render buffers, and so on...
- Limitations:
  - No tessellation shaders.
  - No geometry shaders.
- Example
  - webgl.html

Three.js

- Three.js is a 3D library that tries to make it as easy as possible to get 3D content on a webpage.
- Three.js is often confused with WebGL since more often than not, but not always, three.js uses WebGL to draw 3D.
  - ASCII
- Middleware between an engine and raw WebGL*.
- Useful links:
  - Documentation
  - Examples
  - Repository

Following slide shows about Three.js basic entities.

Three.js



UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Three.js – Tutorial

The tutorial is incremental, so in order to follow these steps, please look at files threejs-X.html, on which X is the step’s number.

1.Scene setup with one cube.

2.Cube’s animation.

3.Lights to the scene and using a material that is affected by lights.

4.More objects to the scene.

5.Camera controls.

6.Grid and axes helpers.

7.Box highlights.

8.Crate texture.

9.Box selection.

10.Transforming tools.

A diff tool between the files is highly recommended for comprehension.

Three.js – Caveats

- How to update parameters (position/texture)
  - Some parameters are uniforms/inputs to shaders under the hood, but other such as presence of texture in a material requires a shader program compilation which will lead to a completely different time of execution.
- How to manage GPU memory
  - When adding objects to scene they are automatically transferred to GPU memory.
  - When deleting objects from scene they’re NOT automatically delete from GPU memory.
    - We must use .dispose method available in all entities that send data to GPU.
  - Tracking GPU memory in Chrome:
    - More Tools -> Task manager
- Example
  - threejs-caveats.html

References:

https://developer.mozilla.org/en-US/docs/Learn

https://threejs.org/

https://threejsfundamentals.org/

● Thank you!

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf INSTITUTO DE INFORMÁTICA UFRGS