WebGLStudio
a Pipeline for WebGL Scene Creation

Javi Agenjo, Alun Evans, Josep Blat
Universitat Pompeu Fabra
Motivations

Paradigm shifting from Desktop to browser in all lightweight daily applications
Industry Needs

Our department was working on industry projects that required browser based 3D editors 2 years ago: Started developing a proof of concept 3D editor, meant to test the possibilities of the web for high-performance
What is an editor?

An **environment** where users can import resources, and arrange them to construct a semantic scene to share.

In a 3D environment the resources are **meshes, textures, materials, and other scene components** (lights, cameras).

The result is an **interactive 3D Scene** available in any browser.
What's out there?

Realtime editors / Engines
- Unity
- Unreal Engine
- Blender

Offline 3D Editors
- Maya
- 3Ds Max
- Cinema 4D
Web 3D Engines

- Unity (now very common for cross platform 3D)
- Flash 3D
- Unreal Engine (WebGL version as of March 2013)
- Three.JS
- Other WebGL Custom Engines...

VRML...
WebGLStudio overview

- Core Engine
- Render Pipeline
- Editor
- Resources Manager
- Post-processing effects
- Particle Engine
- Realtime mesh painting
- Scene saving / export
Core Engine

- Lightweight low level library to wrap most of WebGL functionality, called LiteGL
  - Classes for basic components in 3D: Mesh, Texture, Buffer, Shader
  - Mathematics: uses glMatrix, adds other classes like Octree, RayTesting, AABB, ...

- LiteScene library to handle the Scene Graph
  - Classes: Scene, SceneNode, Component.
  - Camera, Light, ...
  - ResourcesManager
Core Engine main features

- SceneGraph
- Component based
- Resource Management
- Binary formats for fast loading
- Meant to high quality rendering
- Works in Firefox and Chrome (not tested fully on Safari)
Core Engine
Render Pipeline

- Generates the shader according to the material properties (ubershader)

- Supports multiple lights (omni, spot and direct) with projective texture, realtime reflections, shadows with shadowmaps, postprocessing FX, multiple textures per material.

- Events system to hook to any rendering stage
Editor

- Allows to construct scenes from a set of meshes and textures
- Pixel-perfect picking for selection
- Familiar 'desktop app' interface (Menus, Tools, Gizmos, Attributes editor, keyboard shortcuts ...)
- Easy to add new assets from the hard drive (drag and drop to upload to server)
- Easy to extend (modular architecture)
Resources Manager

- Sync'd with the server

- Allow to easily upload any kind of asset
  - Textures
  - Meshes
  - Materials
  - Scenes
  - Prefabs

- Generates a preview image to easy browsing

- Allows to add comments or other info
Resources Manager
Post-processing FX

- Some basic post-processing effects using Color and Depth information
  - Color curves
  - Depth of Field
  - Edges
  - Lens distortion
Particle Engine

- Particle emisor with several configurable settings:
  - Alpha and size curves to control over time
  - Animated textures
  - Plane, Sphere and Mesh emisors
Particles
Realtime mesh painting

- Allows to paint any channel of a material directly on the browser
- Octree to test collision
- Paints pixel around the area of collision using an unwrap of the mesh.
Mesh Painter
Performance

We achieve 60fps in scenes with several lights and meshes using mid-specs computers.

The performance drops when having tens of meshes if the shader is too complex but we are very happy with the results.
Conclusions

WebGL + aspects of Javascripts = very suitable match for creating a 3D Editor

"Democratization" of 3D technology - give 'non-technical' users the ability to create interactive 3D scenes that can be distributed for the web

Open source ASAP…
Thanks!