Project Release Plan

Baskin Engineering
UC SANTA CRUZ

Trace.js

January 17, 2015

The Trace.js Team

Chris Camargo (Product Owner)
Shahar Zimmerman
Steven Esser
Katherine Barsaloux
John Haytko

Project Release Plan

Baskin Engineering
UC SANTA CRUZ

Trace.js

We aim to create a Ray Tracer that allows users to define scenes and render them with realistic detail, all in the Web Browser. This deviates from traditional renderers which usually come in the form of large client-side applications.

It’s kind of like RenderMan or Mental Ray in the Browser.

Project Release Plan - User Stories

Baskin Engineering
UC SANTA CRUZ

Trace.js

Sprint 1 User Stories Examples:

- As a developer, I want a Point3D class so that I have a mathematical representation for points in a 3D scene.
- As a developer, I want a Vector3D class so that I have a mathematical representation for vectors in a 3D scene.
- As a developer, I want a Normal class so that I have a mathematical representation for normals in a 3D scene.
- As a user, I want to visit a web page and see an orthographic ray-traced sphere so that I know the ray tracer is working properly.

Project Release Plan - User Stories

Baskin Engineering
UC SANTA CRUZ

Trace.js

Sprint 2 User Stories Examples:

- As a developer, I want a Sampler base-class so that I have an interface to which all Samplers adhere.
- As a developer, I want a Regular class that extends Sampler so that we can have regular sampling.
- As a developer, I want a MultiJittered class that extends Sampler so that we can have Multi Jittered sampling.
- As a user, I want to be able to specify the sampler I use so that I have more options for image quality.

Project Release Plan - User Stories

Baskin Engineering
UC SANTA CRUZ

Trace.js

Sprint 3 User Stories Examples:

- As a developer, I want a simple scene definition language so that I have an abstract way to represent scenes in text format.
- As a user, I want to be able to load scene language files with scene definitions and run them in the tracer so I don’t have to hard-code scenes.
- As a developer, I want to implement shadow casting in existing hit points so that scenes can contain shadows.
- As a user, I want to be able to turn shadows on and off so that I can trace scenes with or without shadows.

Project Release Plan - Architecture

Baskin Engineering
UC SANTA CRUZ

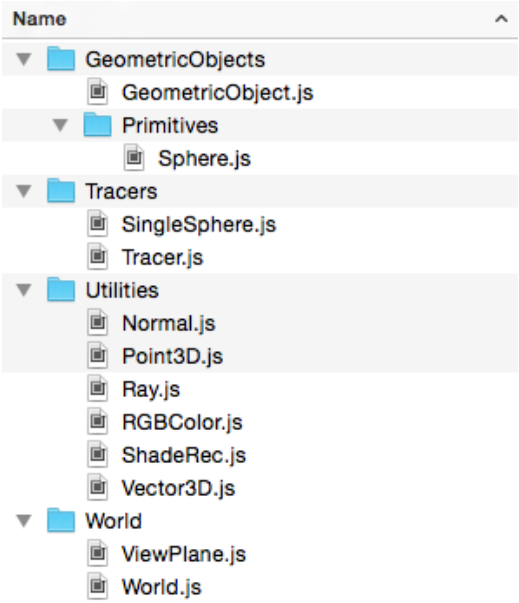
Trace.js

Geometric Objects	**Tracers**	**Utility**	**World**
GeometricObject (base-class)	Tracer (base-class)	Point3D	ViewPlane
Sphere (extends GeometricObject)	SingleSphere (extends Tracer)	Vector3D	World
		Ray	
		Normal	
		ShadeRec	
		RGBColor	

Project Release Plan - Architecture

Baskin Engineering
UC SANTA CRUZ

Trace.js



Project Release Plan - Challenges / Risks

Baskin Engineering
UC SANTA CRUZ

Trace.js

- Developing an efficient Ray Tracer in the web browser using JavaScript.
- Coding all the required classes like the Geometric Objects, Tracers, Utility, and World.
- Making sure Trace.js team understands how to use JavaScript to build a prototype (i.e. Translating known C++ code of Ray Tracer to JavaScript to be compatible on a web browser).
- Completing a working prototype by the end of the course.

Project Release Plan - Technologies

Baskin Engineering
UC SANTA CRUZ

Trace.js

- JavaScript
- GitHub