

# Final Presentation

## Trace.js

March 6, 2015

### The Trace.js Team

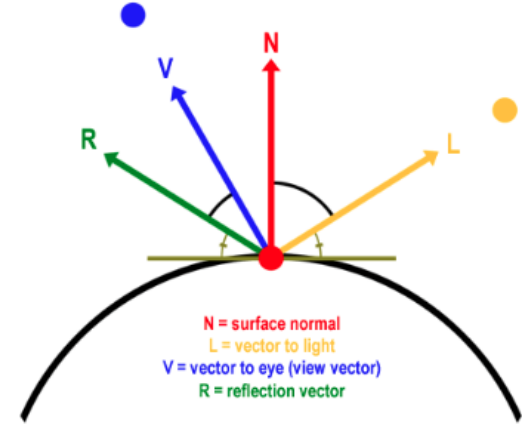
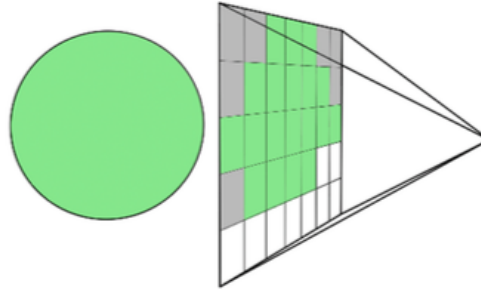
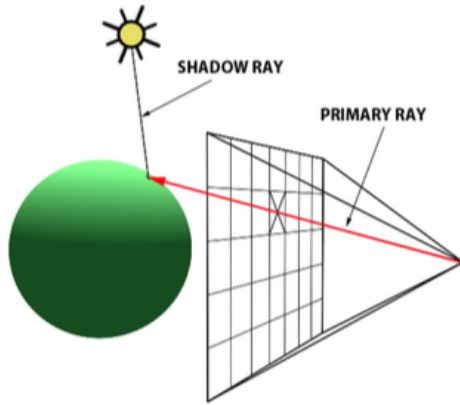
Chris Camargo (Product Owner)  
Shahar Zimmerman  
Steven Esser  
Katherine Barsaloux  
John Haytko

# Trace.js - Main Goal

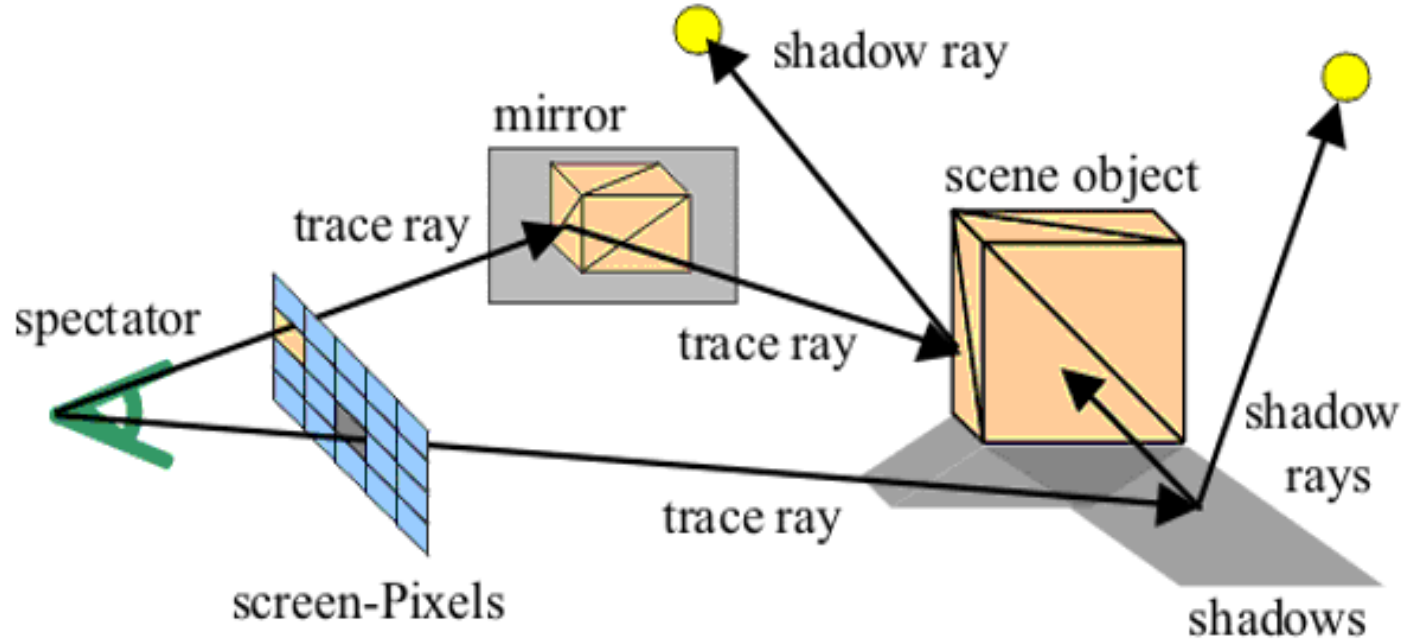
To create a Ray Tracer that allows users to define scenes and render them with custom configurations, all in the Web Browser. This deviates from traditional renderers which usually come in the form of large client-side applications.

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

# Trace.js - Ray Tracing Basics



# Trace.js - Ray Tracing Basics



# Trace.js - Language Decision

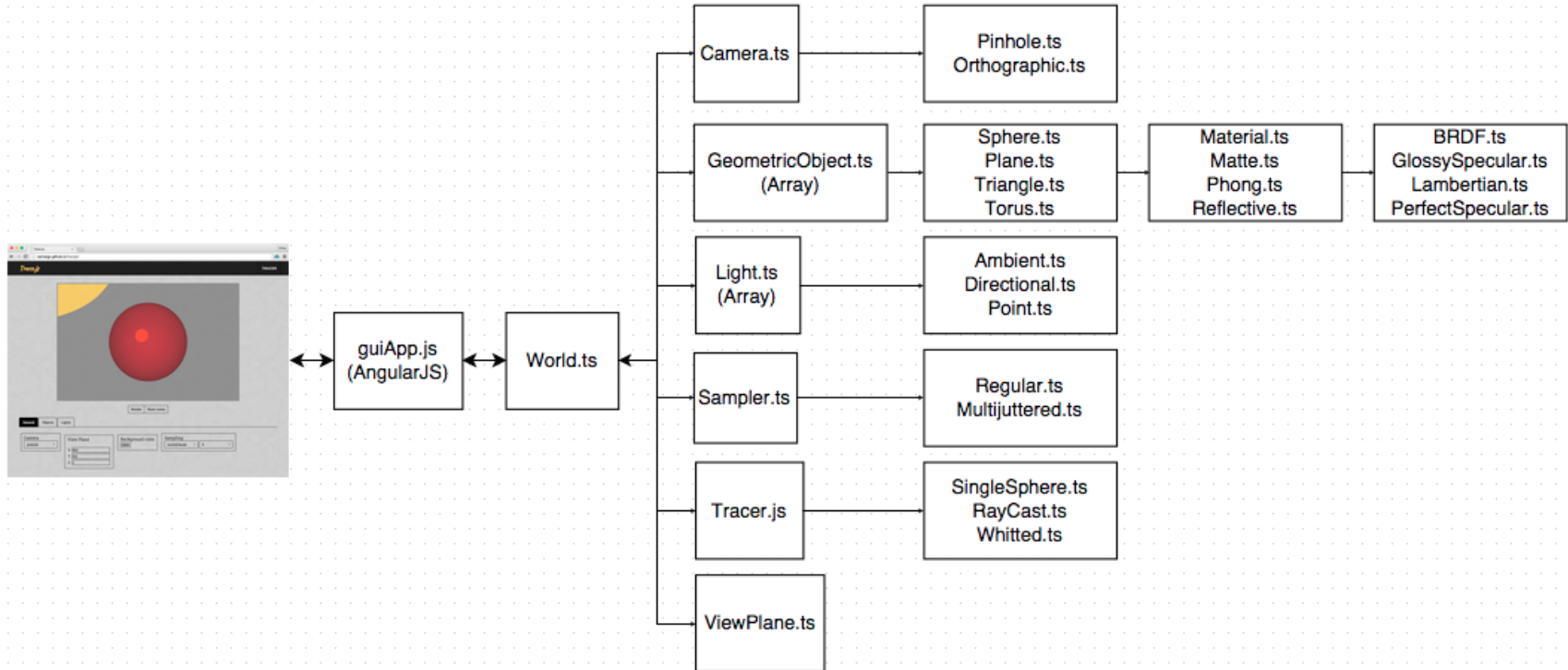
- Wrote the main Trace.js classes using TypeScript (<http://www.typescriptlang.org/>)
  - JavaScript is not a statically typed language

```
1 // Trace.js - Material.ts
2
3 /// <reference path="../../Utilities/RGBColor.ts" />
4 /// <reference path="../../Utilities/ShadeRec.ts" />
5
6 module Tracejs {
7     export class Material {
8         shade(sr : ShadeRec) : RGBColor {
9             //Set RGBColor to black
10            var black = new Tracejs.RGBColor(0,0,0);
11            return black;
12        }
13
14        area_light_shade(sr : ShadeRec) : RGBColor {
15            //Set RGBColor to black
16            var black = new Tracejs.RGBColor(0,0,0);
17            return black;
18        }
19
20        path_shade(sr : ShadeRec) : RGBColor {
21            //Set RGBColor to black
22            var black = new Tracejs.RGBColor(0,0,0);
23            return black;
24        }
25    }
26 }
```

```
1 var Tracejs;
2 (function (Tracejs) {
3     var Material = (function () {
4         function Material() {
5         }
6         Material.prototype.shade = function (sr) {
7             var black = new Tracejs.RGBColor(0, 0, 0);
8             return black;
9         };
10        Material.prototype.area_light_shade = function (sr) {
11            var black = new Tracejs.RGBColor(0, 0, 0);
12            return black;
13        };
14        Material.prototype.path_shade = function (sr) {
15            var black = new Tracejs.RGBColor(0, 0, 0);
16            return black;
17        };
18        return Material;
19    })();
20    Tracejs.Material = Material;
21 })(Tracejs || (Tracejs = {}));
```



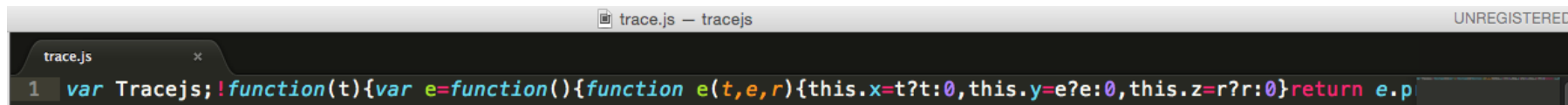
# Trace.js - Architecture



# Trace.js - Building



- Built project using Gulp (<http://gulpjs.com/>)
  - This allowed us to streamline complex build tasks like:
    - i. TypeScript compilation
    - ii. JavaScript concatenation and minification

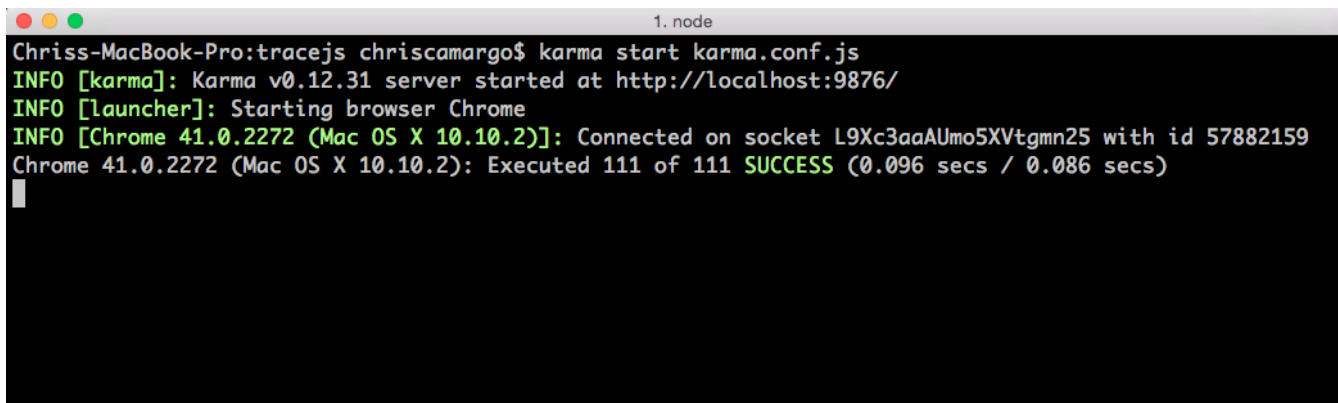


```
trace.js — tracejs UNREGISTERED
trace.js
1 var Tracejs;!function(t){var e=function(){function e(t,e,r){this.x=t?t:0,this.y=e?e:0,this.z=r?r:0}return e.p
```



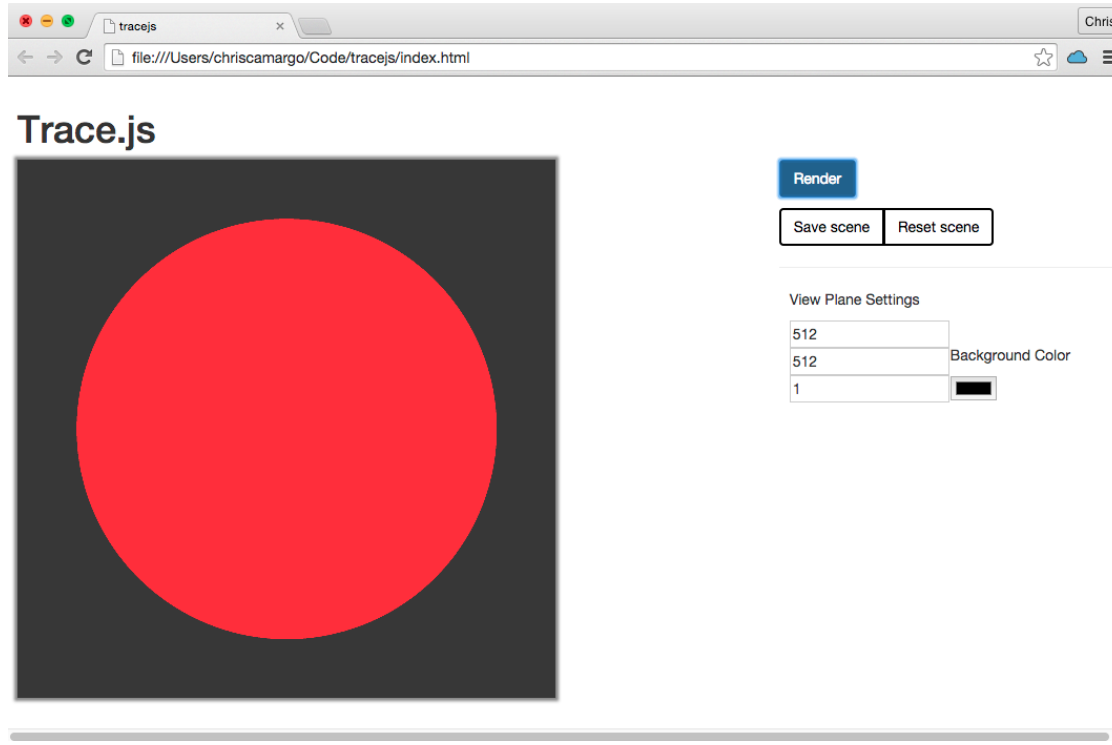
# Trace.js - Unit Testing

- Jasmine (<http://jasmine.github.io/>) to write our unit tests
- Karma (<http://karma-runner.github.io/0.12/>) to run our unit tests
- At the time of writing our code-base has 111 unit tests

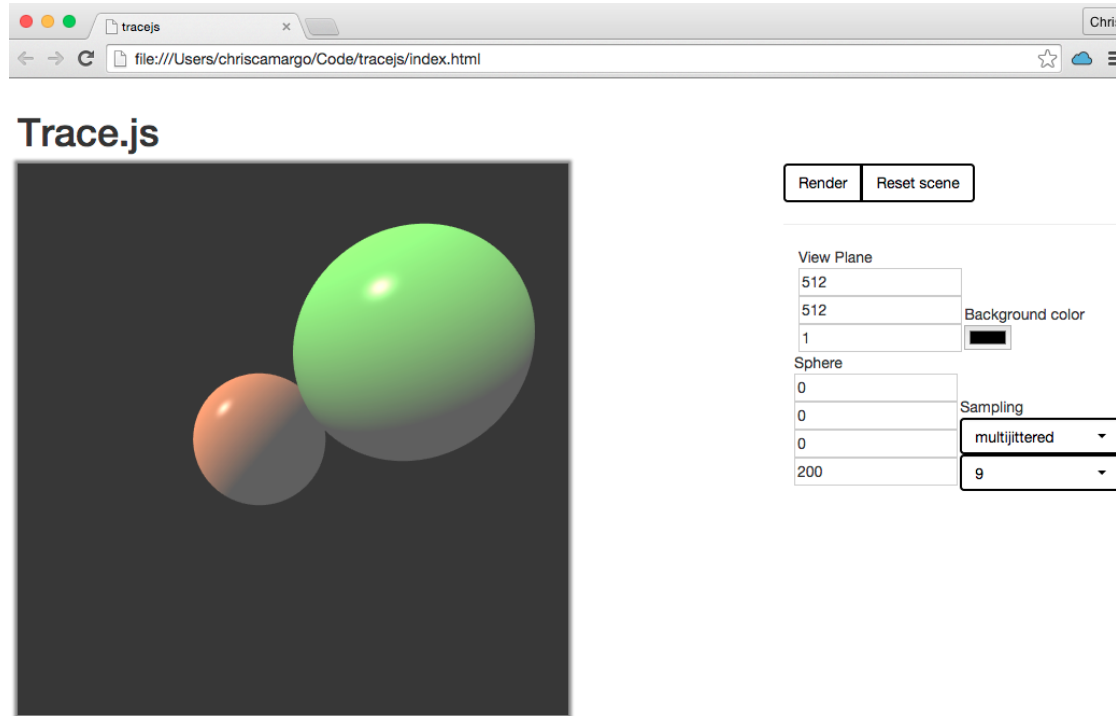


```
1. node
Chriss-MacBook-Pro:tracejs chriscamargo$ karma start karma.conf.js
INFO [karma]: Karma v0.12.31 server started at http://localhost:9876/
INFO [launcher]: Starting browser Chrome
INFO [Chrome 41.0.2272 (Mac OS X 10.10.2)]: Connected on socket L9Xc3aaAUmo5XVtgmn25 with id 57882159
Chrome 41.0.2272 (Mac OS X 10.10.2): Executed 111 of 111 SUCCESS (0.096 secs / 0.086 secs)
```

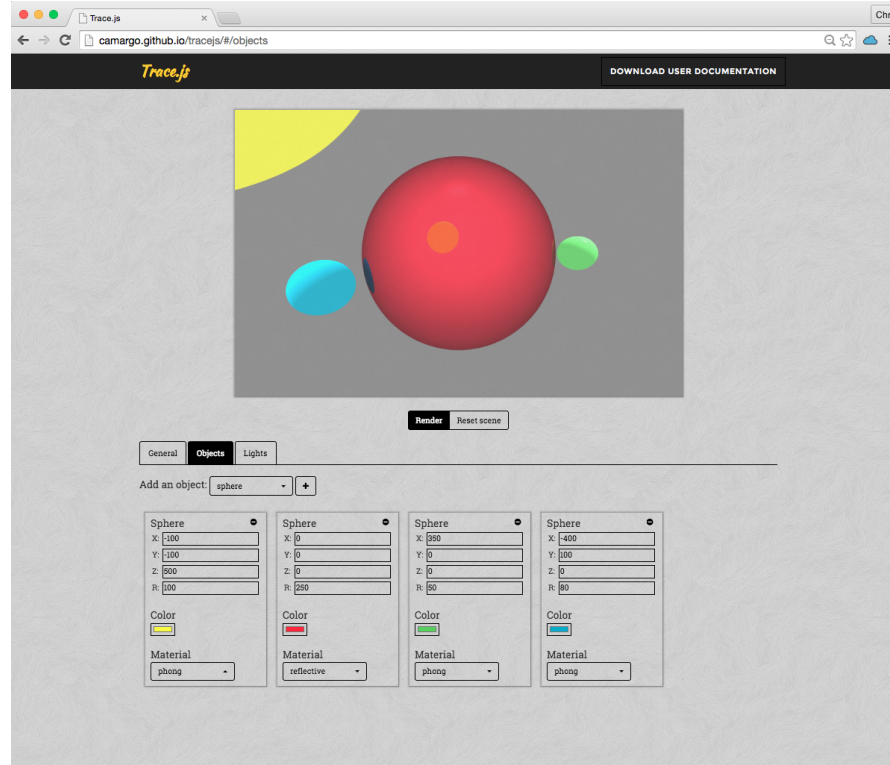
# Trace.js - End of Sprint 1



# Trace.js - End of Sprint 2



# Trace.js - End of Sprint 3



# Trace.js - Demo

- Live Demo: <http://camargo.github.io/tracejs/#/>
- Github Repo: <https://github.com/camargo/tracejs>
- Scrum Board: <https://trello.com/b/hDYuaHtJ/tracejs-scrum-board>