Release Plan

Trace.js
The Trace.js Team
3/12/15
Rev. 1 :: 1/19/15

High-Level Goals

Render a graphical sphere in the web browser using a ray tracer written in Javascript. Users eventually will have the ability to render different objects and scenes in our ray tracing web application.

User Stories for Release (everyone put a size after each story)

- **Sprint 1**
    - As a developer, I want a Point3D class so that I have a mathematical representation for points in a 3D scene. (5)
    - As a developer, I want a Vector3D class so that I have a mathematical representation for vectors in a 3D scene. (5)
    - As a developer, I want a Normal class so that I have a mathematical representation for normals in a 3D scene. (5)
    - As a developer, I want a RGBColor class so that I have a simple representation for an RGB color. (4)
    - As a developer, I want a Ray class so that I can programmatically trace rays. (5)
    - As a developer, I want a GeometricObject base-class so that I have an interface for which all geometric shapes in a scene will adhere too. (4)
    - As a developer, I want a ShadeRec class so that I can record shading information in ray-object intersections easily. (4)
    - As a developer, I want a Sphere class that extends the GeometricObject class so that I have a mathematical representation for spheres in a 3D scene. (4)
    - As a developer, I want a Tracer base-class so that I have an interface for which all tracers in a scene will adhere too. (4)
    - As a developer, I want a SingleSphere class that extends the Tracer class so that I can ray trace a single sphere in a scene. (4)
    - As a developer, I want a ViewPlane class so that I have a representation for the final ray traced image. (5)
    - As a developer, I want a World class so that I have a simple way to represent all the objects in a given scene. (4)
    - 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. (3)

- As a user, I want to see that the ray-traced sphere is an accurate representation with correct proportions, color, and size. (2)

- **Sprint 2**
  - As a developer, I want a Sampler base-class so that I have an interface for which all Samplers adhere too.
  - 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.
  - As a developer I want a Camera base-class so that I have an interface for which all Cameras adhere too.
  - As a developer, I want a Orthographic class that extends Camera so that I can view images orthographically.
  - As developer, I want a Pinhole class that extends Camera so that I can view images under perspective projection.
  - As a user, I want to be able to specify the viewing method so that I have options for how I view a scene..
  - As a developer, I want a BRDF base-class so that I have an interface for which all BRDFs adhere too.
  - As a developer, I want a Lambertian class that extends BRDF so that I can do lambertian reflection on a surface.
  - As a developer, I want a PerfectSpecular class that extends BRDF so that I can do perfect specular reflection on a surface.
  - As a developer, I want a GlossySpecular class that extends BRDF so that I can do glossy specular reflection on a surface.
  - As a user, I want to be able to specify the BRDF used on a surface so that I have more control over the reflection in the final image.
  - As a developer, I want a Light base-class so that I have an interface for which all Lights adhere.
  - As a developer, I want an Ambient class that extends Light so that we have ambient light in our scenes.
  - As a developer, I want a Directional light class that extends Light so that we have directional lights in our scenes.
  - As a developer, I want a PointLight class that extends Light so that we have point lights in our scene.
  - As a user, I want to be able to specify the type of light in our scene so that I have more options for how I view a scene.
  - As a developer, I want a Whitted class that extends Tracer so that we can implement whitted ray tracing.

- As a developer, I want a PathTrace class that extends Tracer so that we can implement path tracing.
- As a developer, I want Material base-class so that I have an interface for which all materials adhere too.
- As a developer, I want a Matte class that extends Material so that objects can have matte shading (e.g. pure diffuse).
- As a developer, I want a Phong class that extends Material so that objects can have phong shading (e.g. ambient + diffuse + specular).
- As a user, I want to be able to specify the type of surface objects in the scene have so that I have more options for how a scene is viewed.

- **Sprint 3**
  - TODO: Stories for shadows, transparency, ambient occlusion (maybe), area lights (maybe), global illumination (maybe)
  - 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.