Introduction to 3D Graphics with three.js

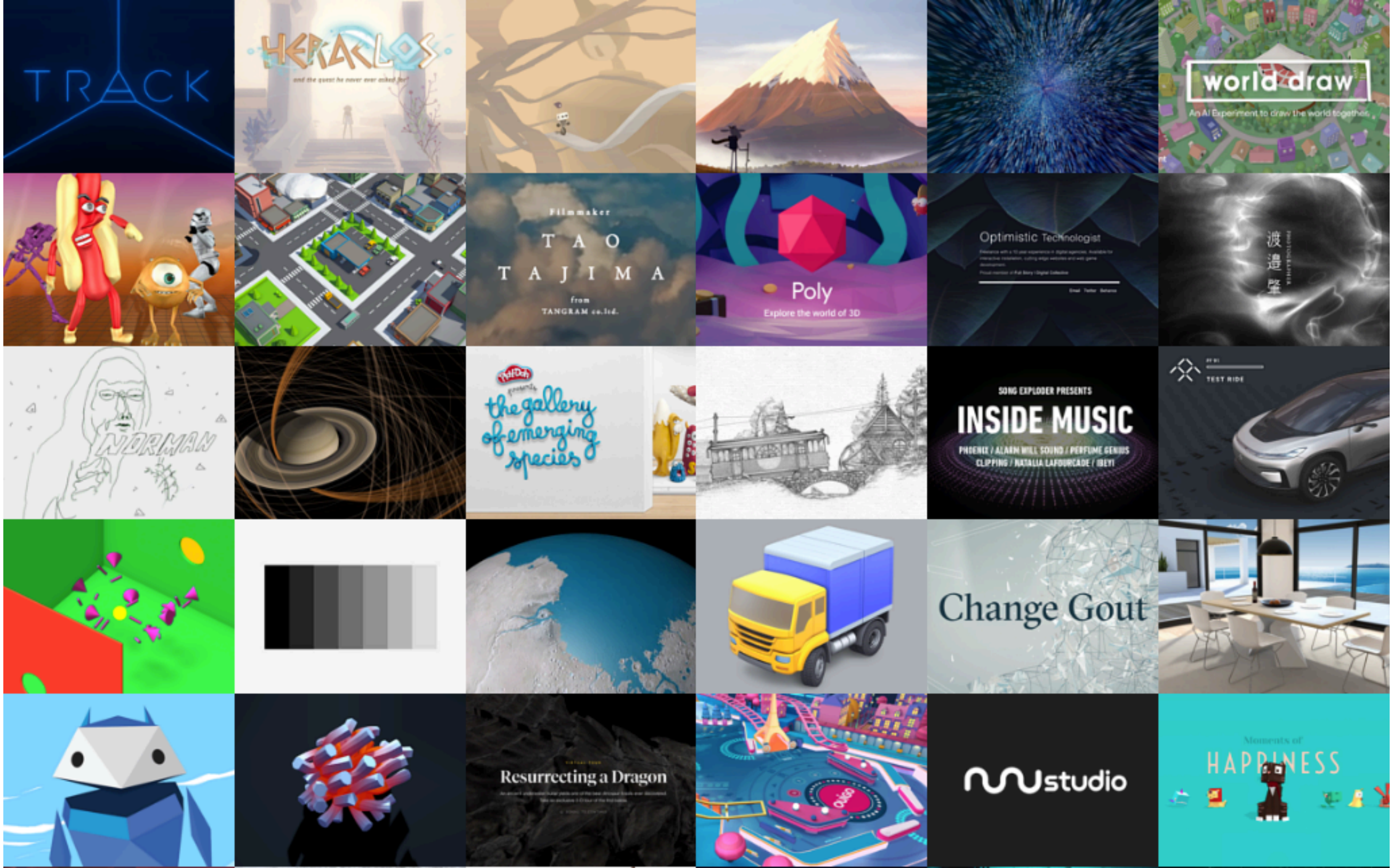
Adapted from slides by Crystal Hess

Learning Goals

- What is three.js?
- What is “the documentation”?
- How does the coordinate system work in 3D?
- What can a for-loop be used for?
- Note similarities between programming languages used to code in three.js (JavaScript) and Arduino (c/c++)

What is three.js?

- JavaScript language
- Runs in a web browser
- Uses WebGL to run graphics calculations off of the graphics card
- Relatively accessible and (mostly) platform-agnostic 3D graphics


TRACK
HERACLES
and the quest he never ever asked for
world draw
An AI Experiment to draw the world together.
Filmmaker
TAO TAJIMA
from
TANGRAM co.ltd.
Poly
Explore the world of 3D
Optimistic Technologist
渡邊肇
PHOTOGRAPHER
NORMAN
Play-Doh
presents
the gallery of emerging species
SONG EXPLODER PRESENTS
INSIDE MUSIC
PHOENIX / ALARM WILL SOUND / PERFUME GENIUS
CLIPPING / NATALIA LAFOURCADE / IBEYI
TEST RIDE
Change Gout
Resurrecting a Dragon
studio
Moments of
HAPPINESS

Basic Lingo

- Scene
    - Where the objects exist
- Camera
    - How we view the scene
- Lightning
    - How anything is visible
- Renderer
    - The technical details for how a virtual scene becomes something we see of the screen

Example: Basic

- Download the class repo and open the web->three_projects->00_basic folder
- Open three.html with a browser
- Open project.js with a text editor
- **Play with the code!**
  - Try changing some camera numbers
  - What happens if you set the mesh material to wireframe: false?
  - De-comment the axesHelper to see the scene's axes drawn for you

- How does js look similar to the Arduino language?
- How does it look different?

The Camera

- Graphics cameras are modeled like old pinhole cameras

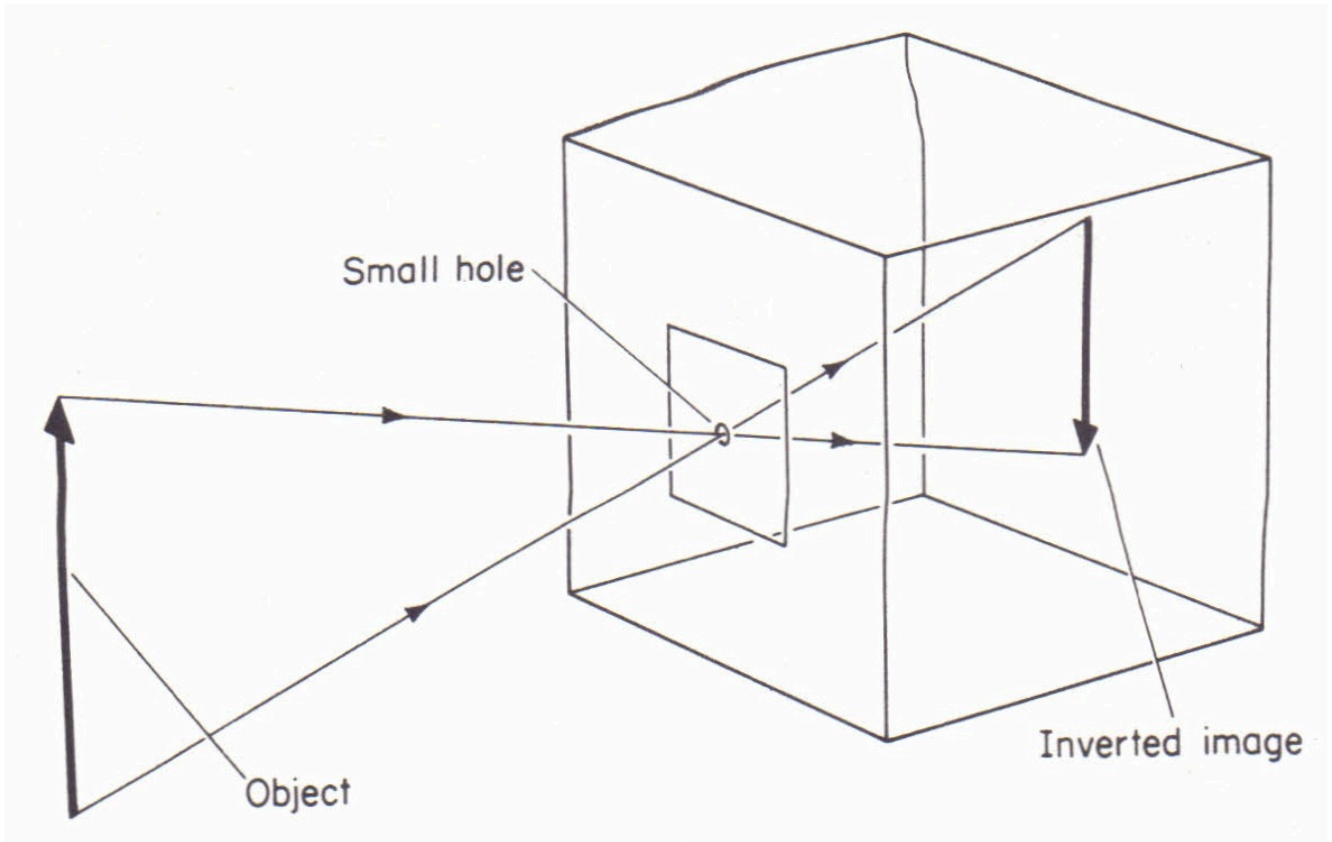
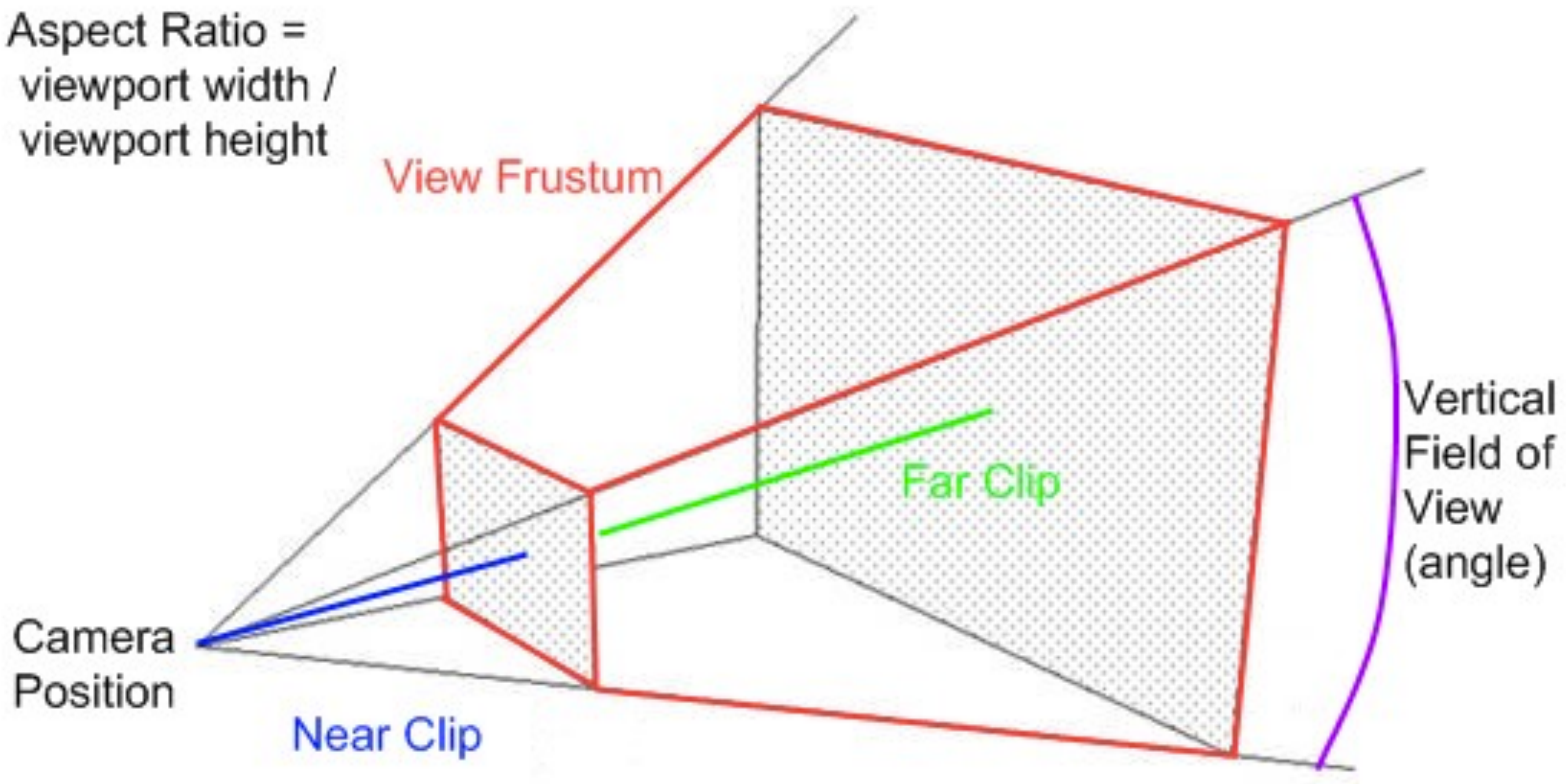
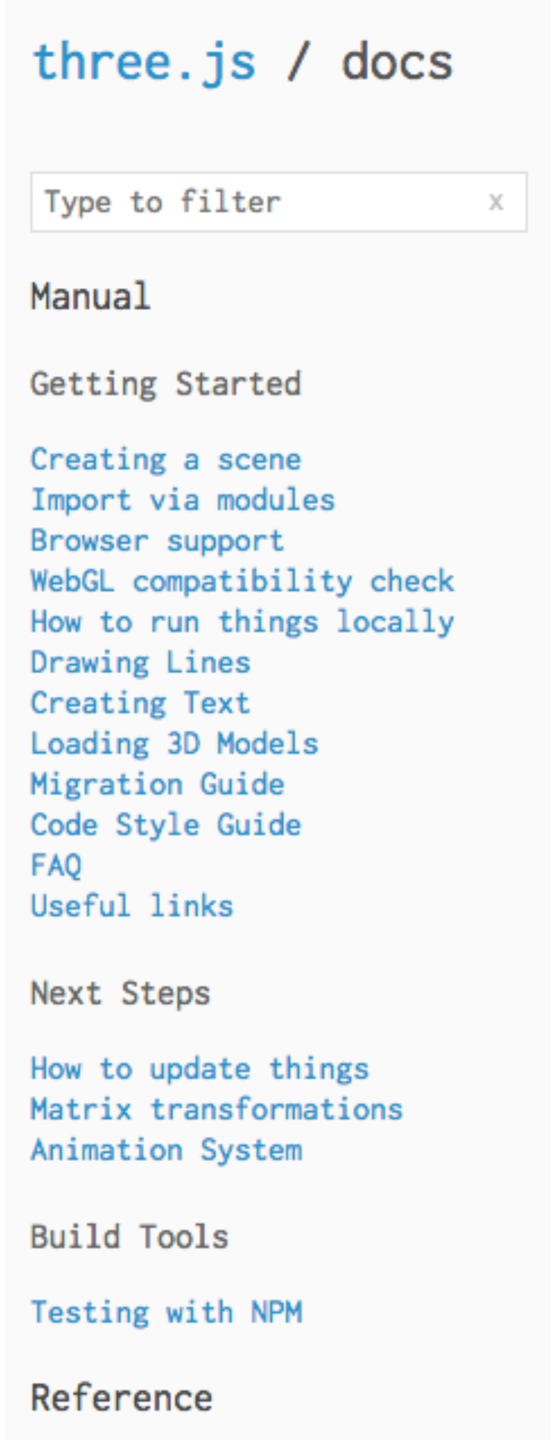


Fig. 21.2. Image formation by a pinhole camera

The Synthetic Camera





three.js / docs

Type to filter

Manual

Getting Started

Creating a scene
Import via modules
Browser support
WebGL compatibility check
How to run things locally
Drawing Lines
Creating Text
Loading 3D Models
Migration Guide
Code Style Guide
FAQ
Useful links

Next Steps

How to update things
Matrix transformations
Animation System

Build Tools

Testing with NPM

Reference

The three.js API

Remember the Documentation!

https://threejs.org/docs/index.html

Example: Lighting

- Now open the 01_sphere_with_lighting project
- What's different?
- Try commenting out the ambient light in the code
- Turn that back on and try it without the point light code
- Add a new point light
- Play with the sphere's metal-ness and rough-ness values (range 0.0-1.0)
  - What do you think they do?
  - Try it with the flat shading turned off
- **Added Fun**
  - Replace the point light with a directional light
  - Make the sphere use a different material than MeshStandardMaterial

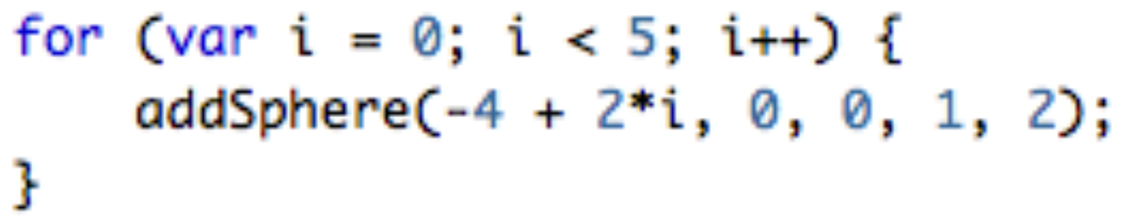
Common Light Types

- Ambient
  - That dim glow that's everywhere
- Point Light
  - Just like a lightbulb
- Directional Light
  - Light coming from one direction, everywhere
  - Looks like daylight
- Spotlight
  - Just like you would think
  - What would this be most similar to?

Material Lighting Properties

- Diffuse (aka “roughness”)
  - Like a matte finish
  - Brighter when the light source is perpendicular to the surface
- Specular (aka “metalness”)
  - Makes a thing look shiny
  - Brighter when the light source is a the same angle to the surface as the camera
- Together with ambient lighting these are commonly known as the “Phong” model of graphics lighting
- Other lighting stuff (these mostly require ray-casting):
  - Reflection
  - Refraction
  - Caustics

For Loops



```
for (var i = 0; i < 5; i++) {
    addSphere(-4 + 2*i, 0, 0, 1, 2);
}
```

- Open the 02 project, Loops and Events
- Consider the code here
  1. Can you make the line of spheres longer?
  2. Can you make the spheres stack vertically instead of line up?
  3. Can you space the spheres farther apart?
  4. Can you make the spheres larger?
  5. Can you make each one larger than the last?

- **Added Fun**
  - Make each sphere be a different random color
  - Make each object be a random shape

Shadows

- Shadows are tricky, they cost a lot to compute
- 4 steps required
  - Let the renderer know we are using shadows
  - Enable shadows on the light source (and decide how detailed they will be)
  - Make at least one object that can cast a shadow
  - Make at least one object that can receive the shadow
- What actually happens to make a shadow?
  - The light source creates a new camera that's only b+w
  - Only objects that cast shadows show up in the camera
  - The "screen" of this camera is shown on objects that accept shadows

Playing with Shadows

- Open the 03 Shadow project
- Try changing the shadowMap size to something smaller (like 256)
  - These values are basically the "resolution" of the shadow
- Change the object to something else, see if the shadow changes
- De-comment the shadowCameraHelper to see the area the light's shadow camera is looking
- Add a second light source and make multiple shadows appear