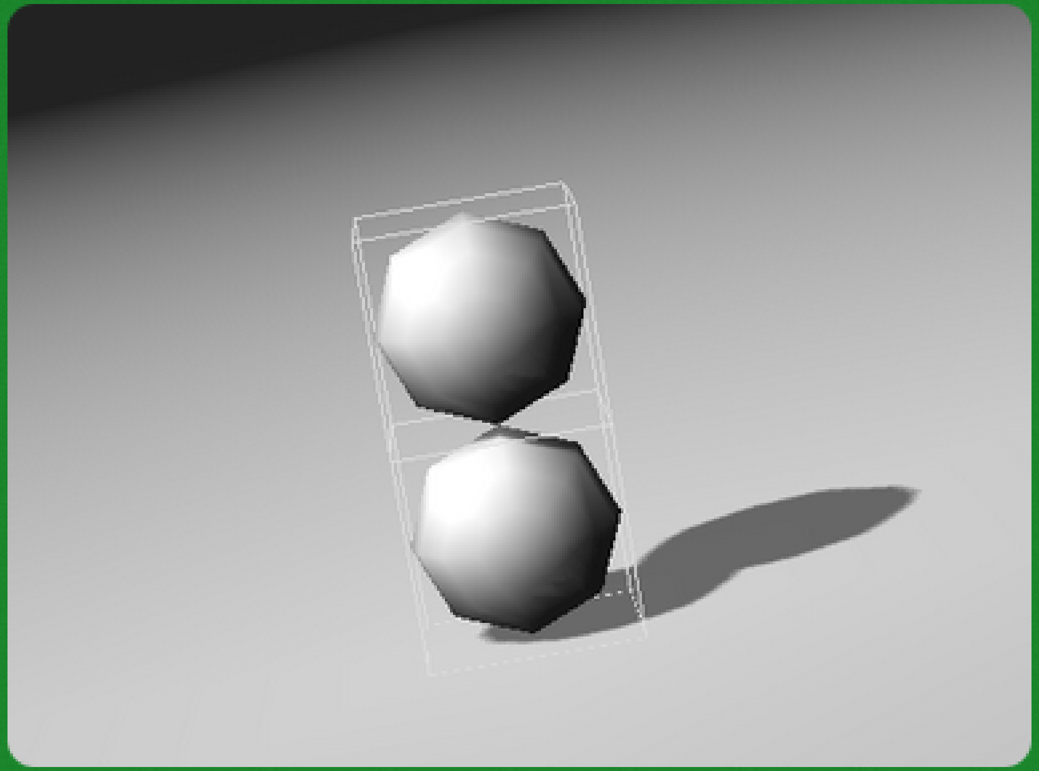
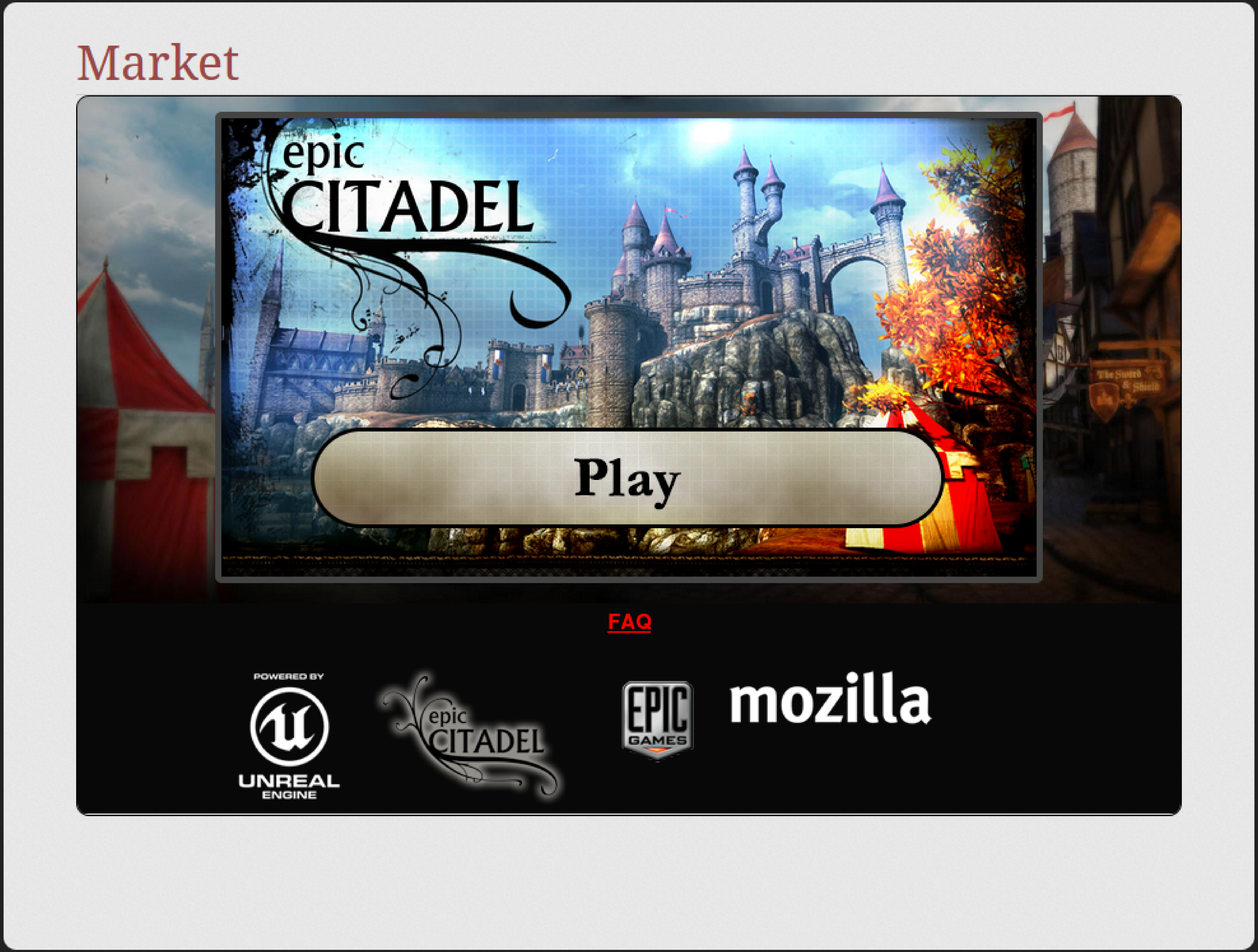
Physics 3D

discover a new dimension





Market

epic CITADEL
The Sword & Shield
Play
FAQ
POWERED BY
UNREAL ENGINE
epic CITADEL
EPIC GAMES
mozilla

Overview

Engine

Differences

Library

Simulation

Aspects

physics related math

Aspects

physics related math

rigid body dynamics

Aspects

physics related math

rigid body dynamics

soft body dynamics

Aspects

physics related math

rigid body dynamics

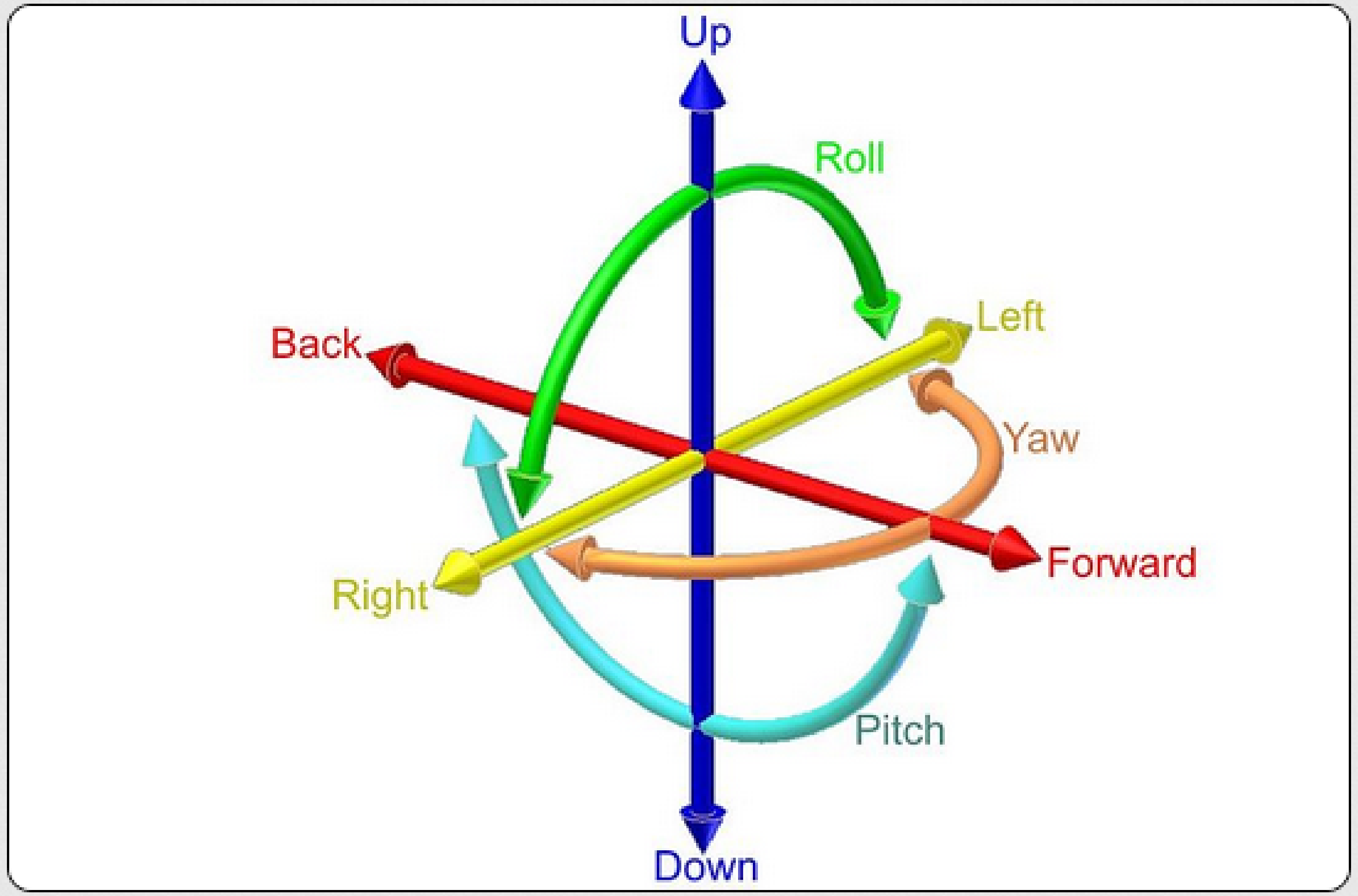
soft body dynamics

collision detection

World

Position

6 degrees of freedom



Relation

camera position

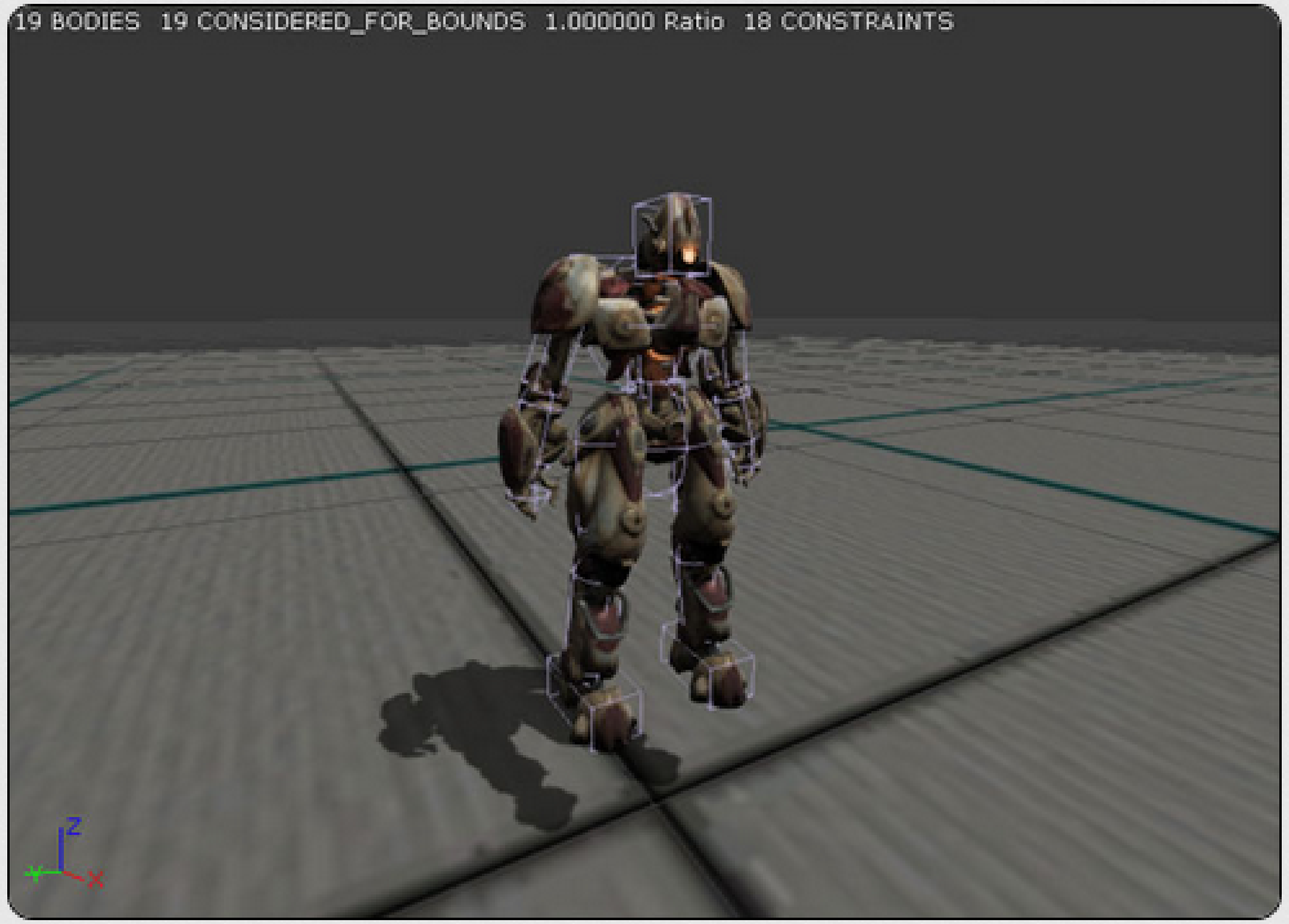
Relation

camera position

perspective + movement

Model

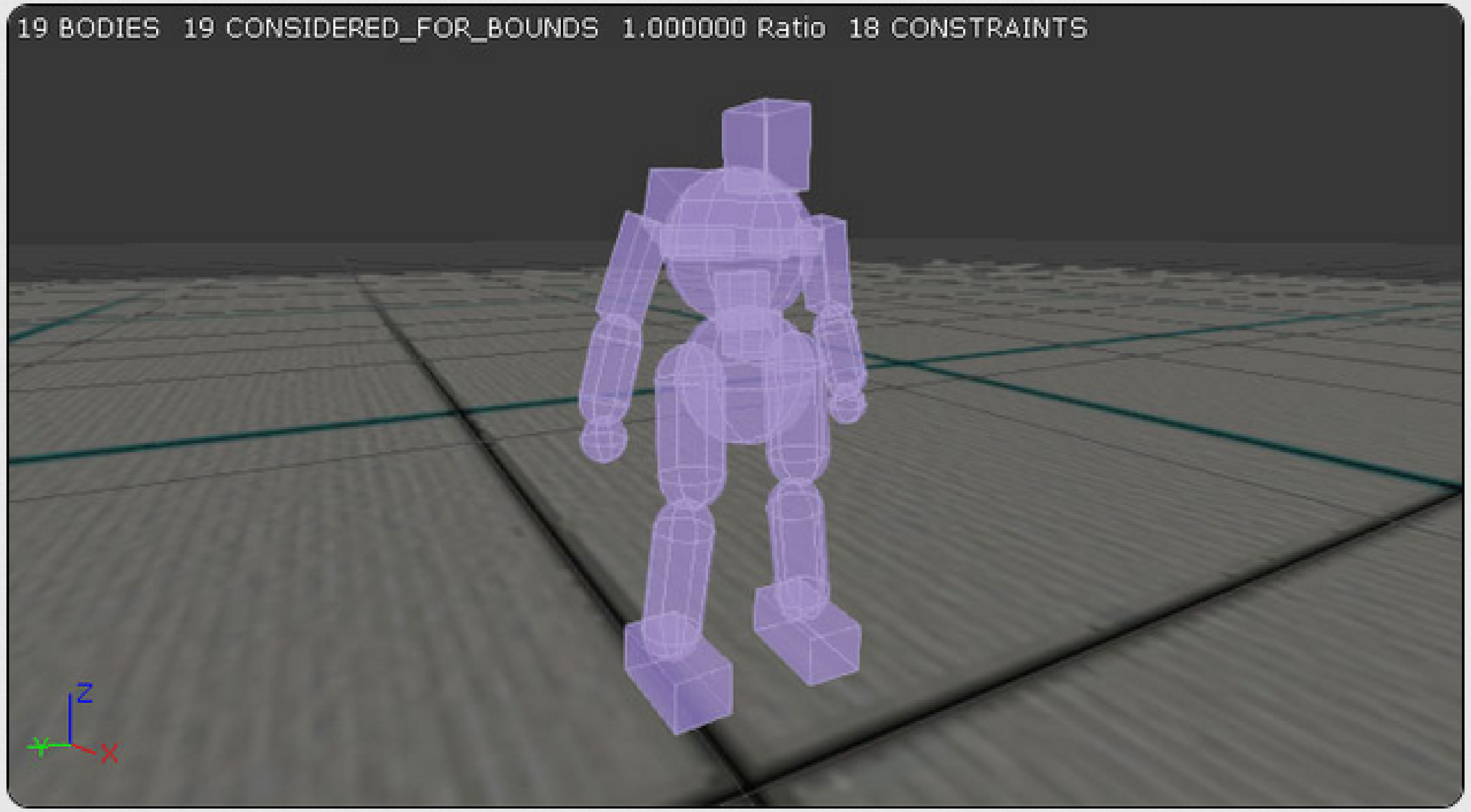
mesh & shapes



```
19 BODIES  19 CONSIDERED_FOR_BOUNDS  1.000000 Ratio  18 CONSTRAINTS
```

Model

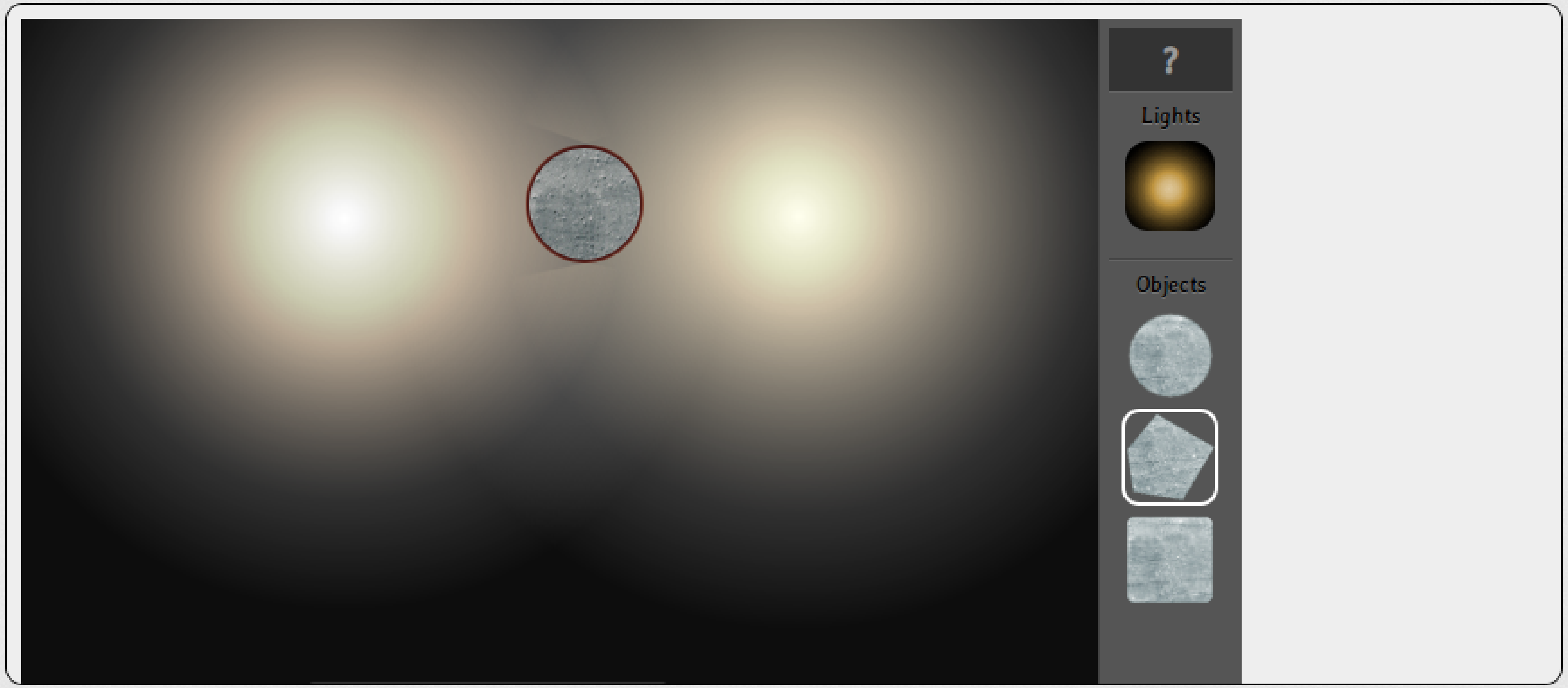
mesh & shapes



```
19 BODIES  19 CONSIDERED_FOR_BOUNDS  1.000000 Ratio  18 CONSTRAINTS
```

Effects

light + shadow



illuminated.js

Performance

hardware requirements

Performance

hardware requirements

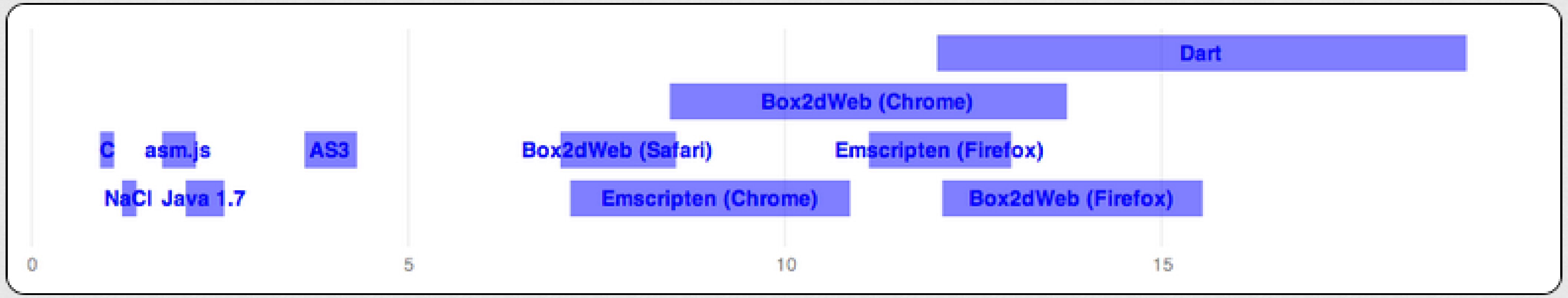
↓

frustum culling

Browser

Comparision

Box2d performance



C (1x) - Java (2.3x) - AS (4x) - JS(8x)

Stats

Cannon.js

"Lightweight 3D physics for the web"

2012 by **@schteppe**

Development: 220kb

Production: 68kb

Github

Features

rigid body physic engine

Features

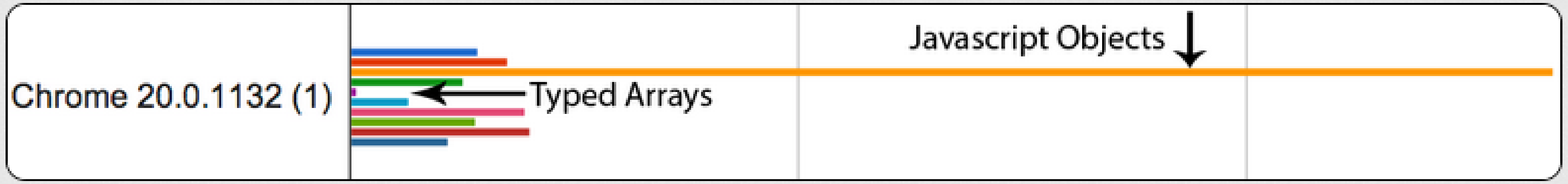
rigid body physic engine

written from scratch

Challenge

"Tried replacing the CANNON.Vec3 with [Typed Arrays]. Sadly that time was wasted... The code ran significantly slower!"

*- **Stefan Hedman** (Cannon.js)*



Vector

Coordinates

```
/** vector container **/

CANNON.Vec3 = function(x,y,z){

    this.x = x||0.0;
    this.y = y||0.0;
    this.z = z||0.0;
};
```

Vector

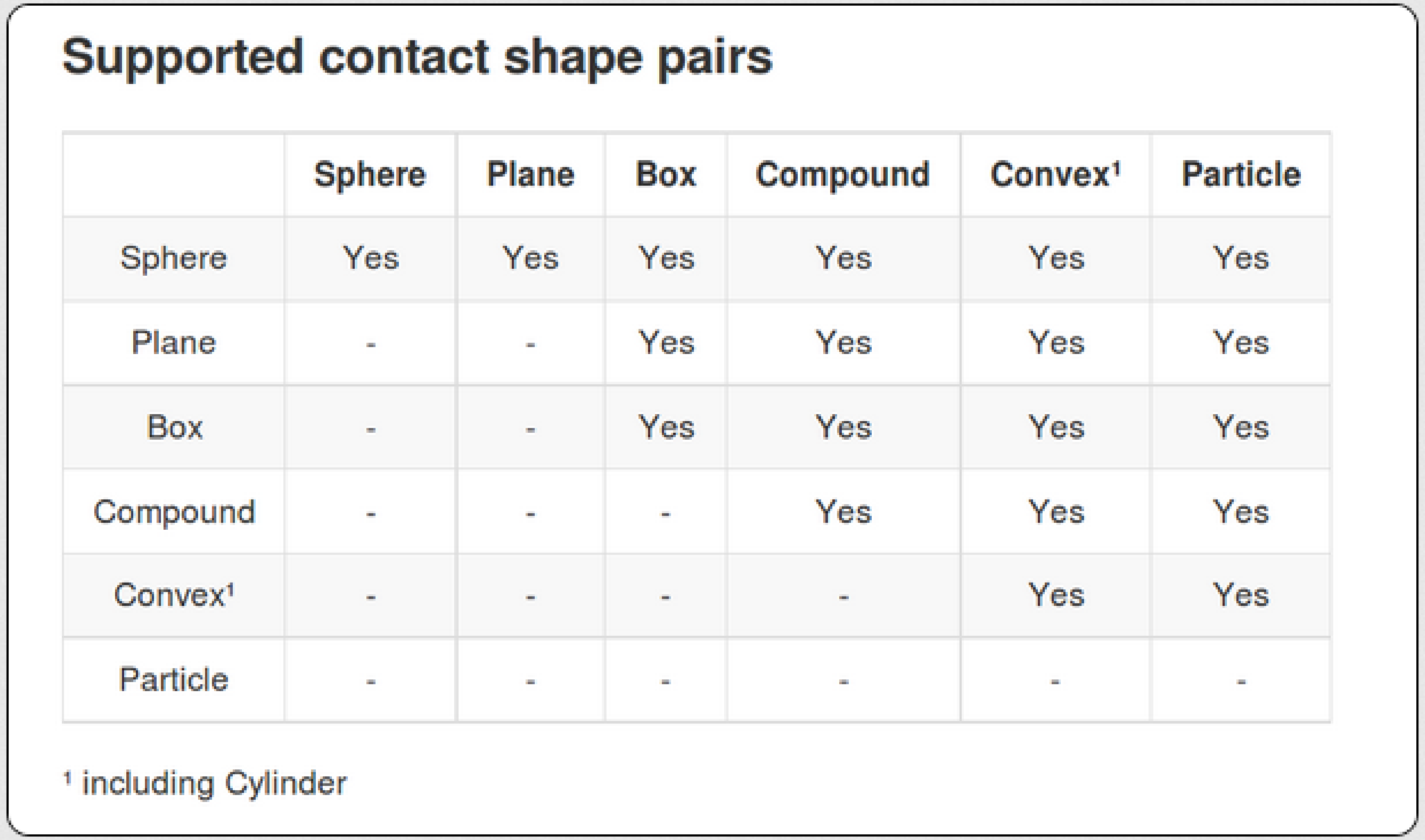
Coordinates

```
/** vector container **/

CANNON.Vec3 = function(x,y,z){

    this.x = x||0.0;
    this.y = y||0.0;
    this.z = z||0.0;
};
```

Collision



Supported contact shape pairs

	Sphere	Plane	Box	Compound	Convex[1]	Particle
Sphere	Yes	Yes	Yes	Yes	Yes	Yes
Plane	-	-	Yes	Yes	Yes	Yes
Box	-	-	Yes	Yes	Yes	Yes
Compound	-	-	-	Yes	Yes	Yes
Convex[1]	-	-	-	-	Yes	Yes
Particle	-	-	-	-	-	-

[1] including Cylinder

Math

"Gauss-Seidel" (**#1** | **#2**) constraint solver

Math

"Gauss-Seidel" (**#1** | **#2**) constraint solver

transformation matrix

Example

Guide

1.) Invoke the Physic World

Guide

1.) Invoke the Physic World

2.) Create an Element

Guide

1.) Invoke the Physic World

2.) Create an Element

3.) Define Limits

Guide

1.) Invoke the Physic World

2.) Create an Element

3.) Define Limits

4.) Update through Rendering

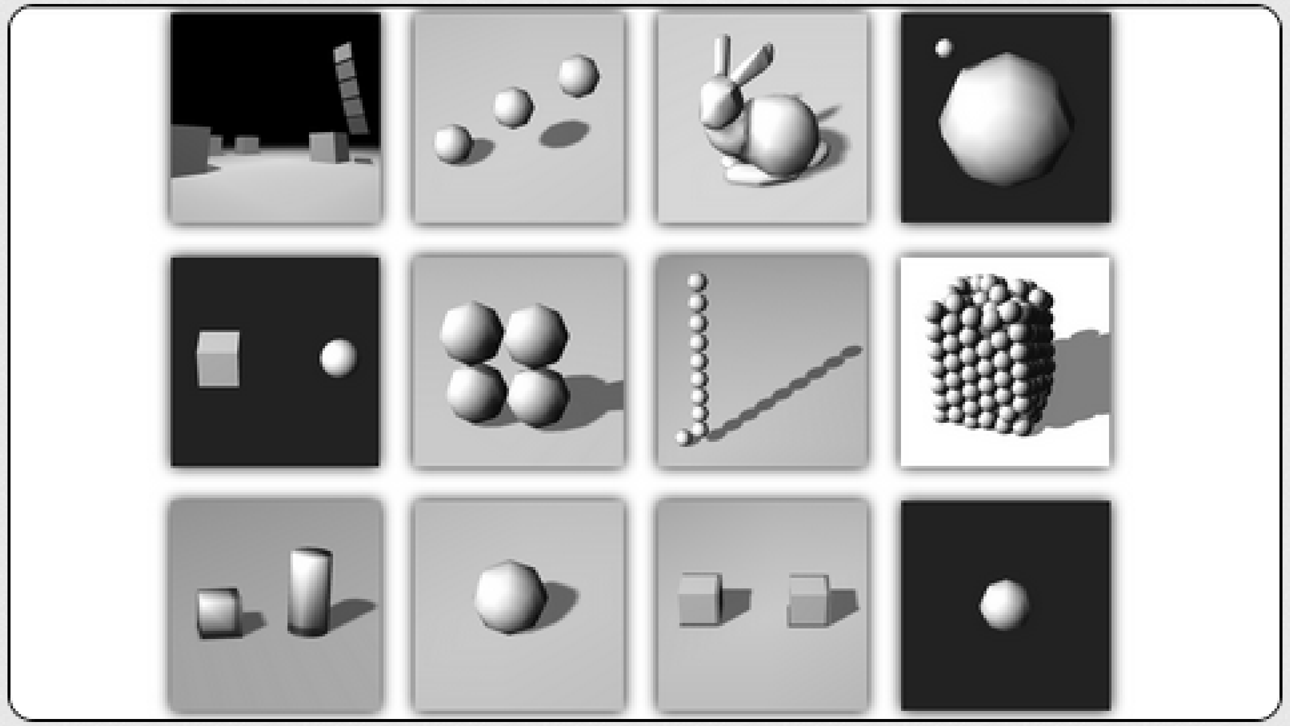
Visual

WebGL - Canvas - CSS3

Three.js

Demo

Simple World



Demos

Questions
&
Discussion