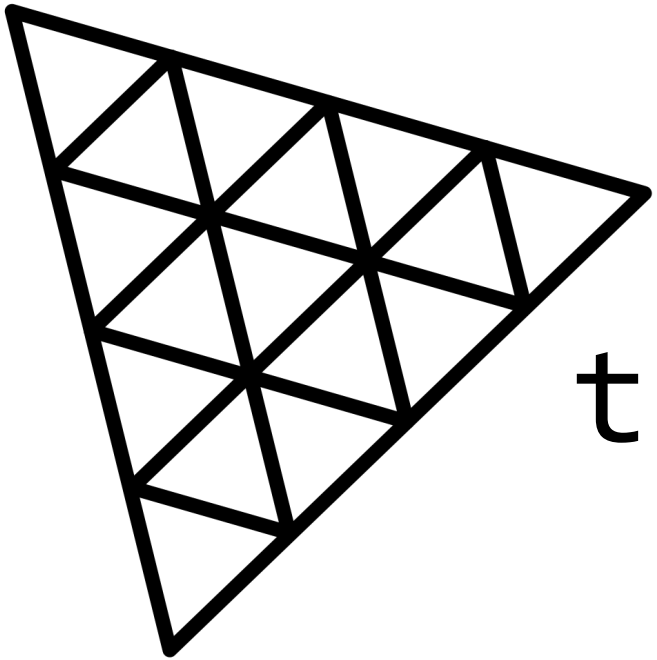
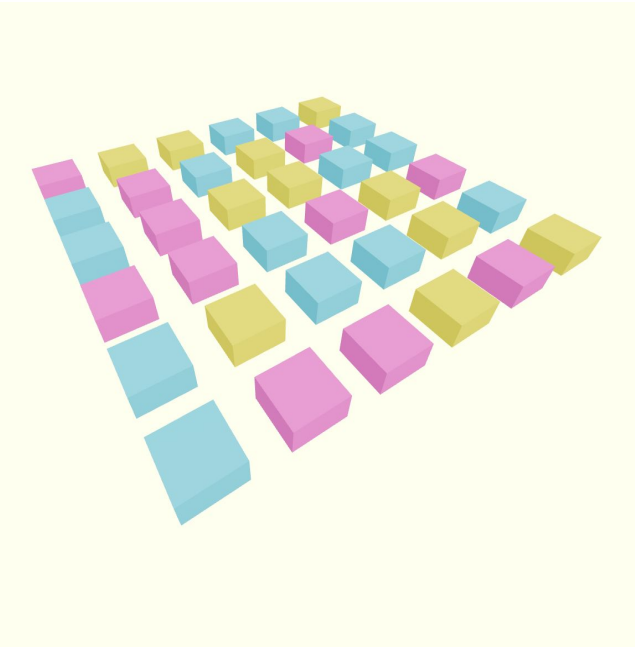


three.js

Lien Cai Ting

Agenda for 27 July



0/ background

what is webgl?

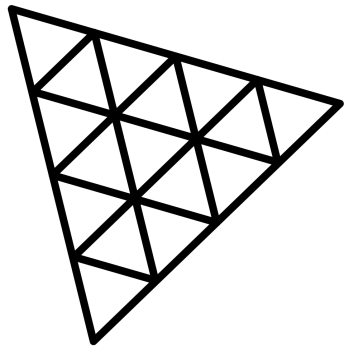
WebGL (Web Graphics Library)

- **JavaScript API** for rendering **interactive 2D and 3D graphics** within a web browser without the use of plugins
- Supported by most major web browsers
- Uses **<canvas>** to draw graphics

what is three.js?

three.js

- JavaScript library and API used to create and display animated 3D computer graphics in a web browser using WebGL
- Layer of abstraction on top of WebGL

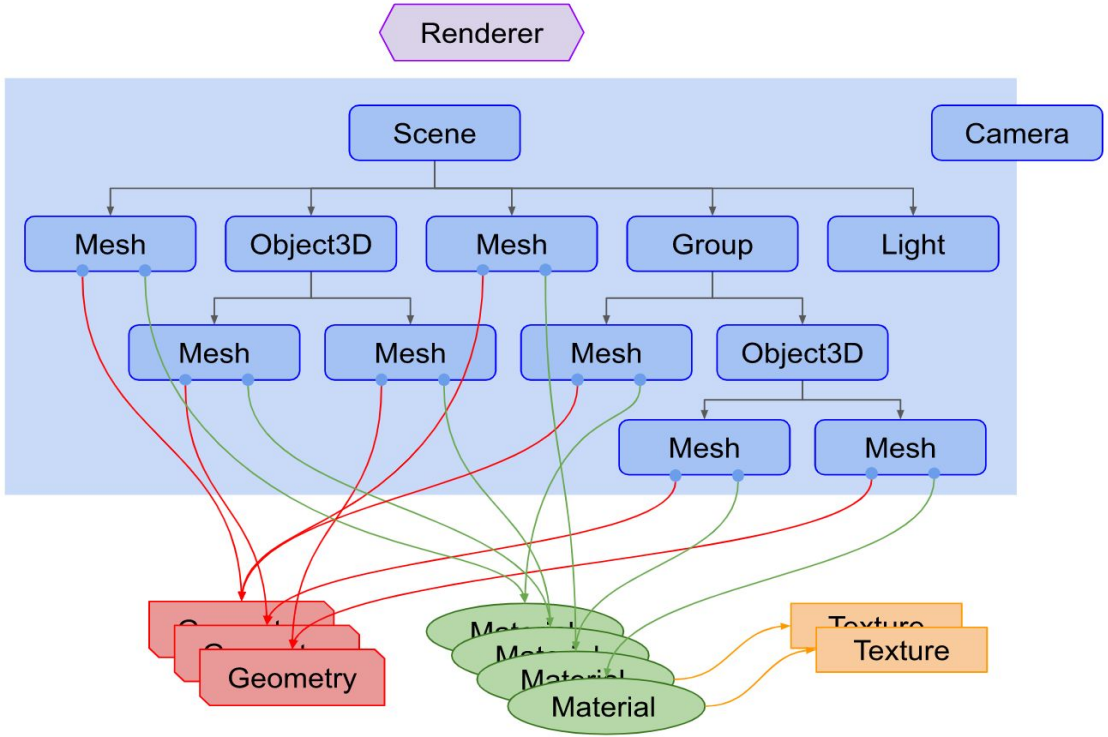


what is react-three-fiber?

react-three-fiber

- React renderer for three.js
- Everything that works in three.js will work here

three.js is hierarchical



what can i do with three.js?

https://chartogne-taillet.com/en

https://renaultespace.littleworkshop.fr/

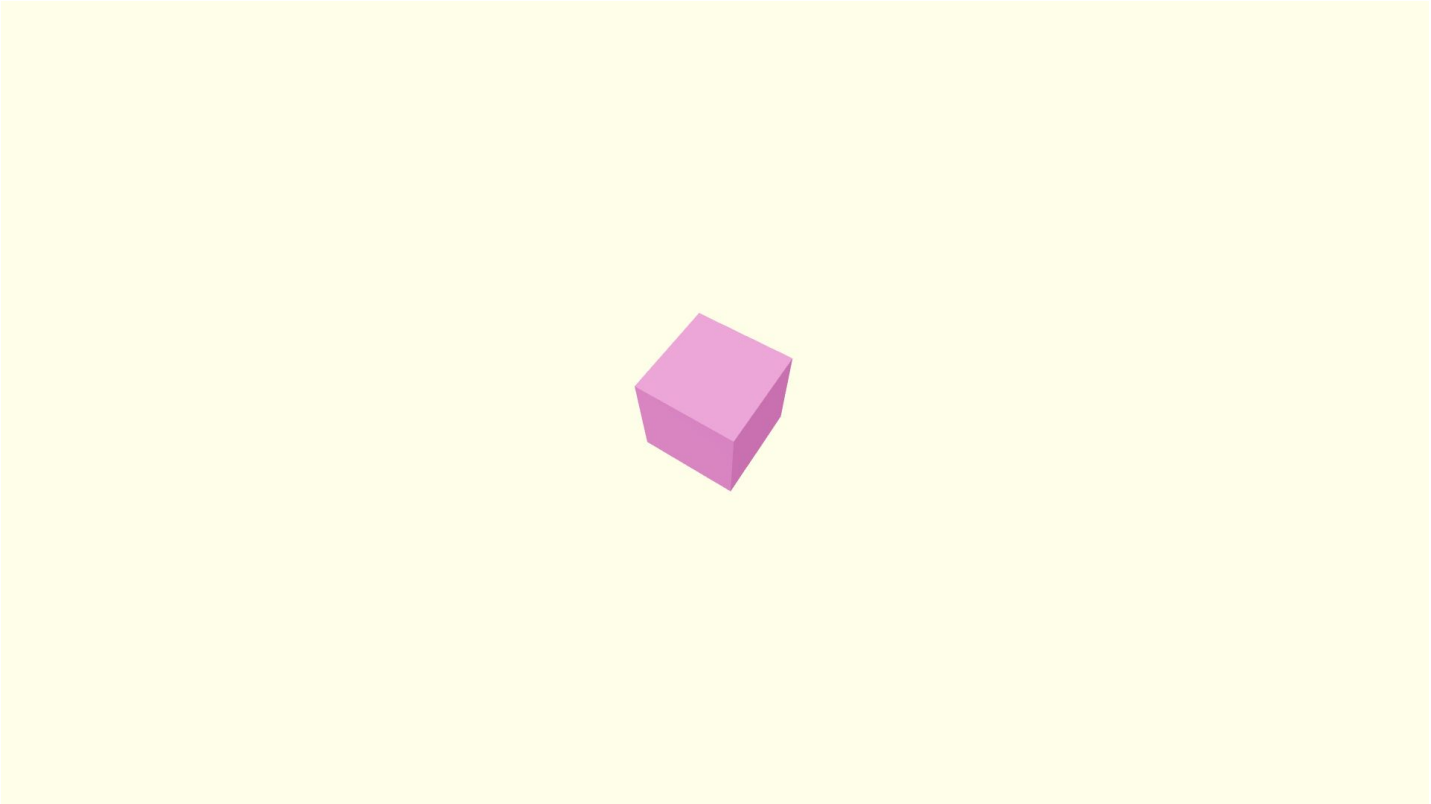
https://bruno-simon.com/

// https://threejs.org/examples

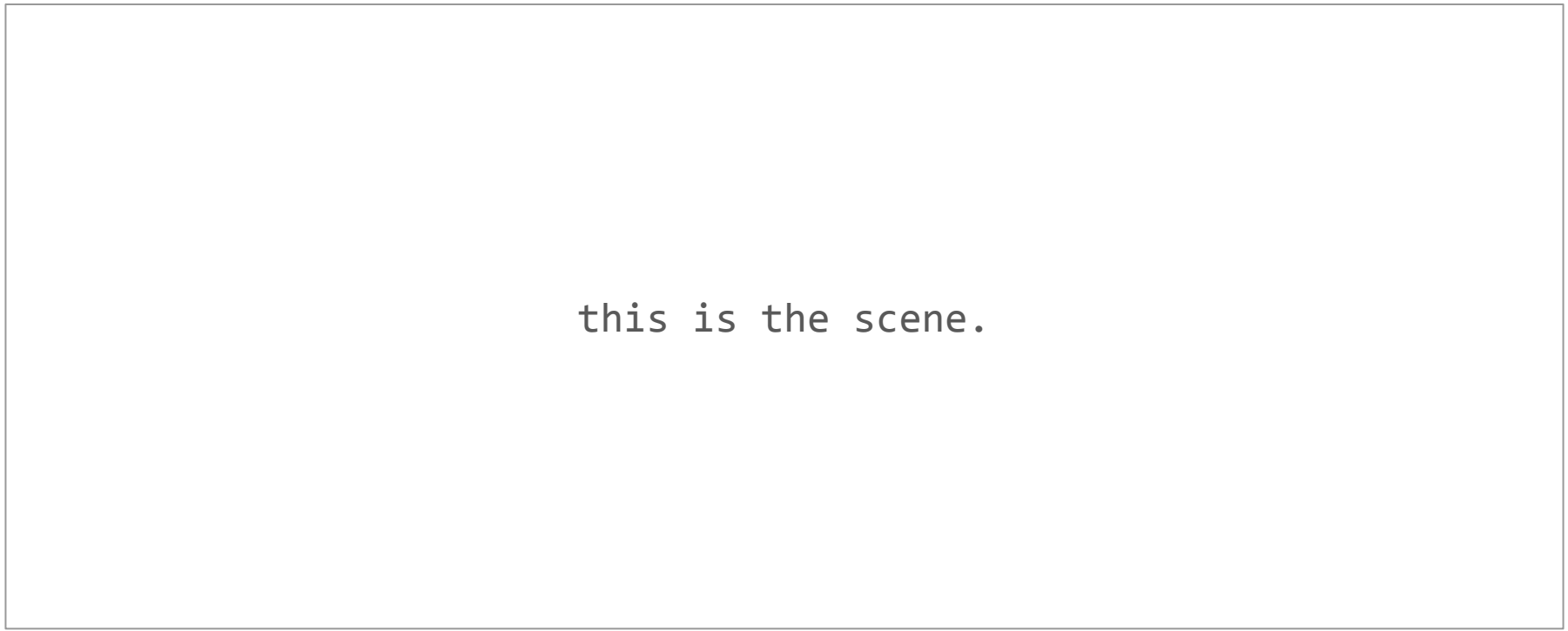
1/ hello world

goal: introduction

(here’s what we want to achieve) **hello cube**



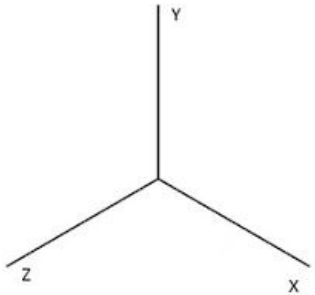
the basics: setting the scene



the basics: setting the scene

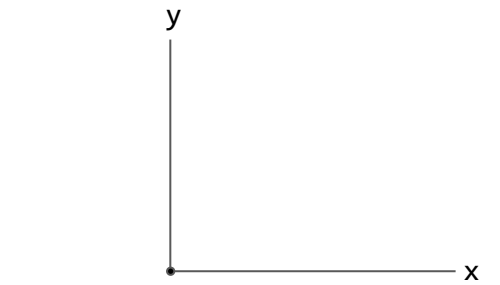
the basics: setting the scene

Cartesian Coordinate System



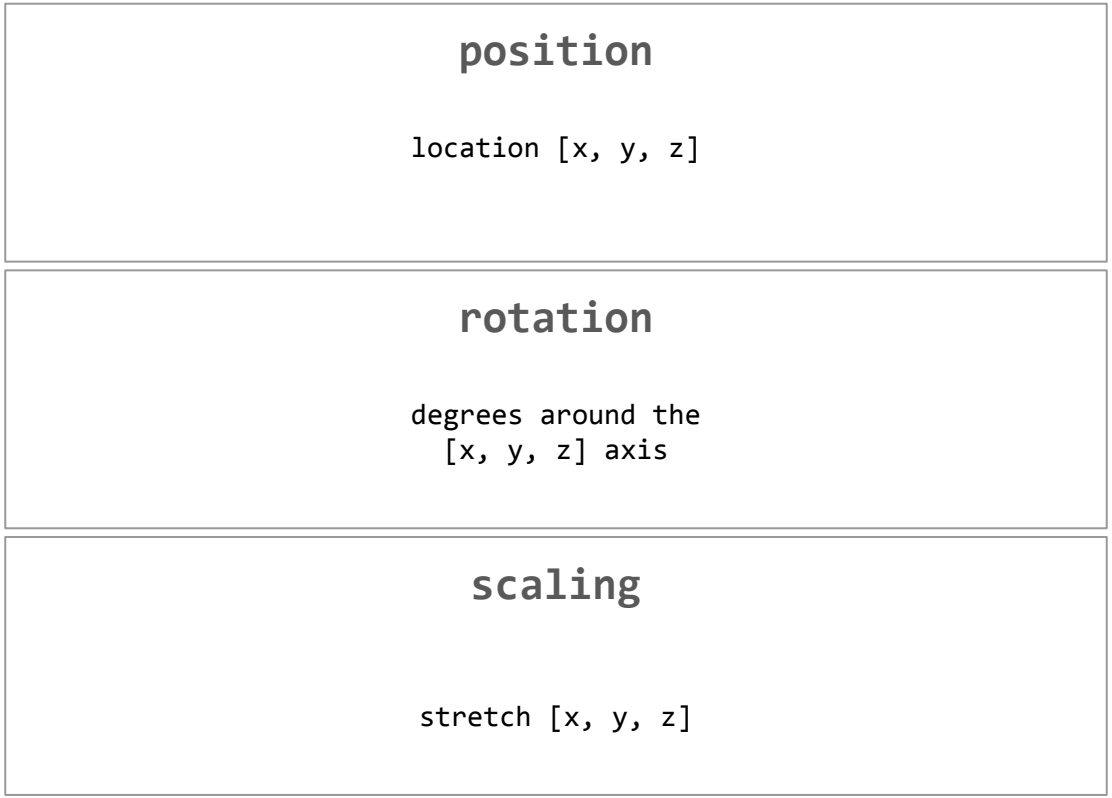
the basics: setting the scene

more accurately



z pointing out of screen

the basics: transformations



the basics: lights, camera, action 🎬

LIGHTS 💡

- **PointLight(**
  - **color : Integer,**
  - **intensity : Float,**
  - **distance : Number,**
  - **decay : Float)**
- Light that gets emitted from a single point in all directions
- There’s also directional light ☀️ (distant source of light) and ambient light (lights all objects evenly) and some others

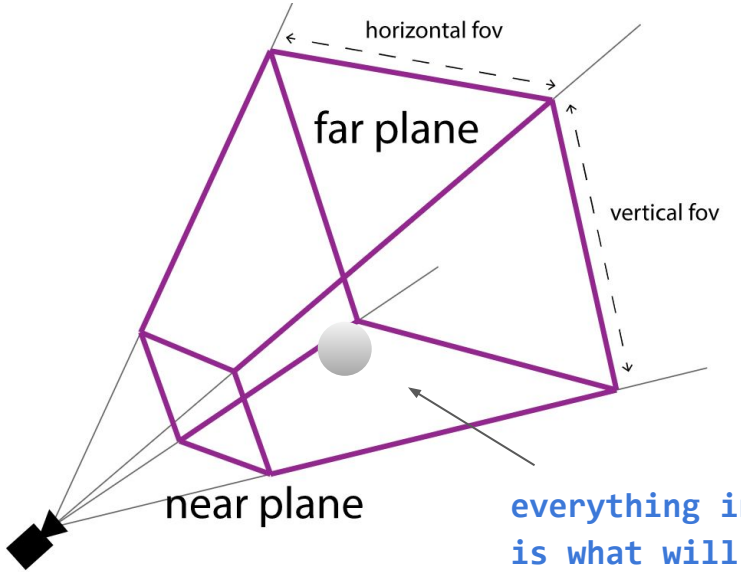
the basics: lights, camera, action 🎬

CAMERA 📷

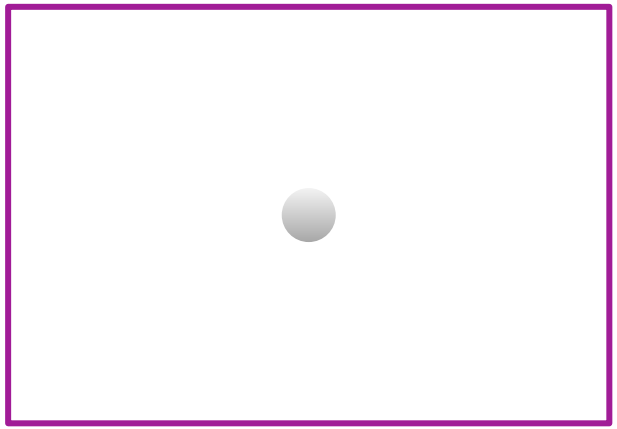
- PerspectiveCamera(
  - fov : Number, // field of view (in degrees)
  - aspect : Number, // aspect ratio: width / height
  - near : Number, // near clipping
  - far : Number) // far clipping
- Camera mimics the way the human eye sees

the basics: lights, camera, action 🎬

CAMERA 📷



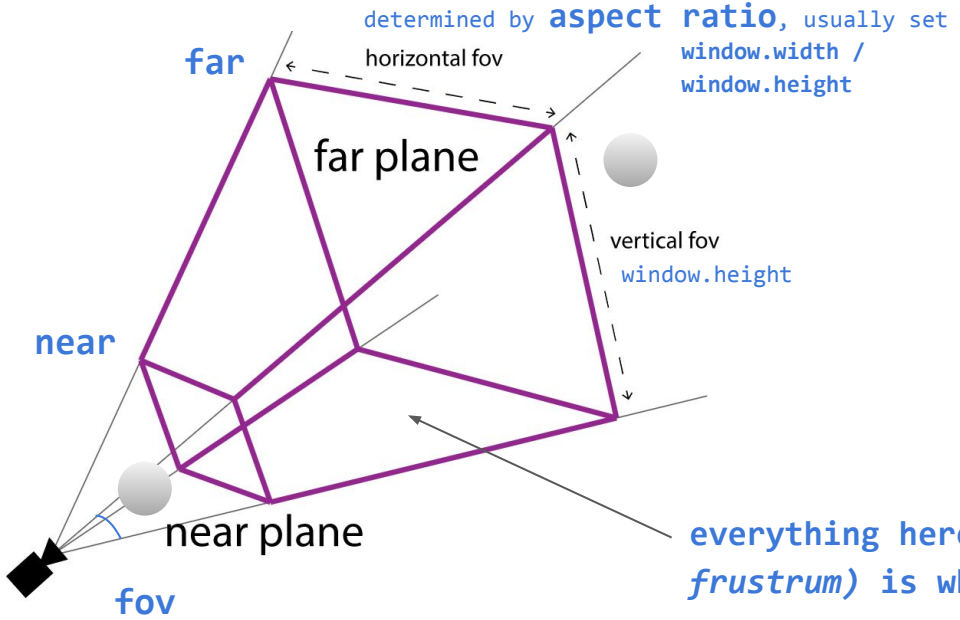
screen



everything in this frustrum
is what will be seen

the basics: lights, camera, action 🎬

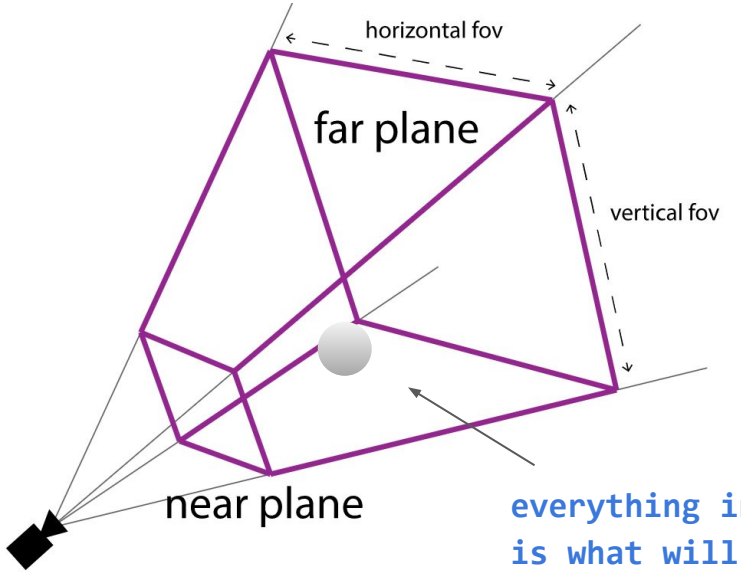
CAMERA 📷



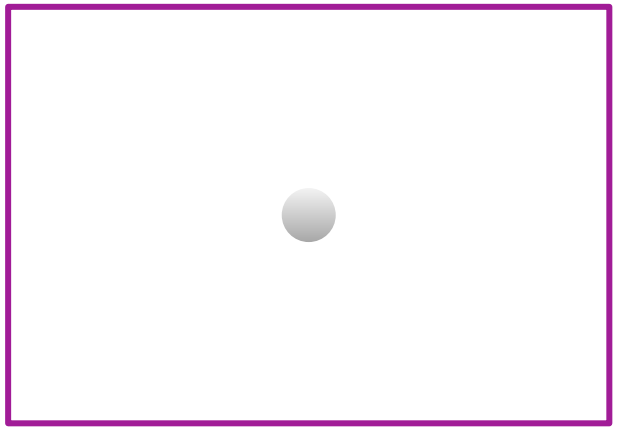
- PerspectiveCamera(
  - fov : Number,
    // field of view (in degrees)
  - aspect : Number,
    // aspect ratio: width / height
  - near : Number,
    // near clipping
  - far : Number)
    // far clipping

the basics: lights, camera, action 🎬

CAMERA 📷



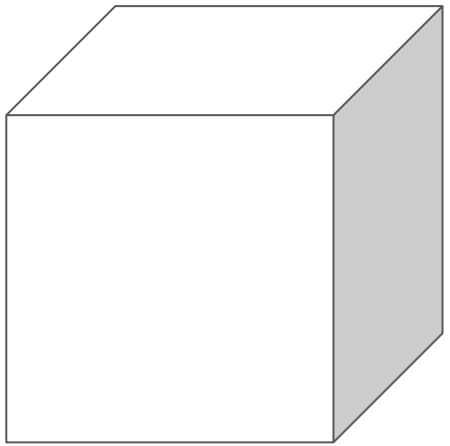
screen



everything in this frustrum
is what will be seen

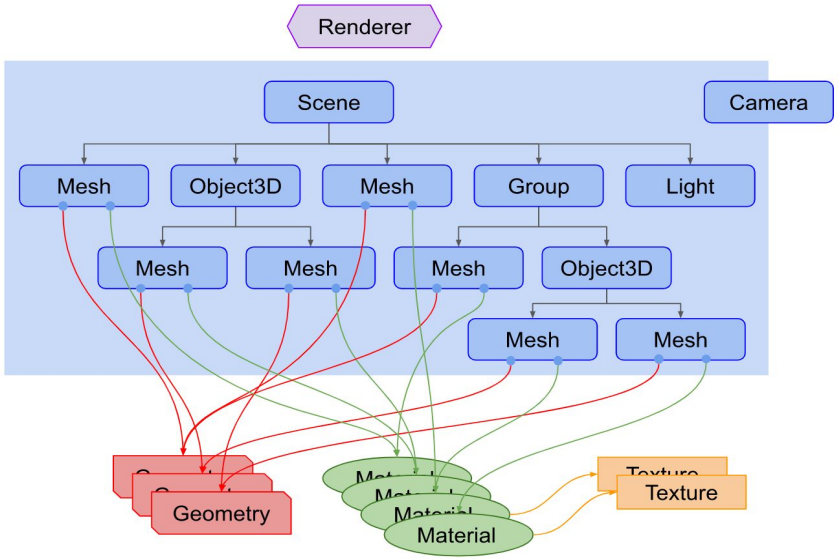
the basics: adding an object (mesh)

- **Mesh: geometry** (shape) and **material** (appearance)
- **Geometry**: Using **Primitives**
  - We will be using Primitives --> basic 3D shapes
  - Specifically, a cube (**BoxGeometry)**
    - **BoxGeometry(width, height, depth);**
- **Material:**
  - **MeshBasicMaterial**
    - {color: ## }
- **Mesh**
  - **Mesh(geometry, material);**



how it all comes together

- There is a `Renderer`. This is arguably the main object of three.js. You pass a `Scene` and a `Camera` to a `Renderer` and it renders (draws) the portion of the 3D scene that is inside the *frustum* of the camera as a 2D image to a canvas.



`renderer`, `scene`, `group`, `object`, `mesh`, `geometry`, `materials`, `texture`

- the `renderer` controls what is shown on the screen
- `scenes` are a collection of `meshes` and `objects`
- `groups` are a collection of `objects` and/or `meshes`
- `objects` are a combination of `meshes`
- `meshes` are a combination of `materials` and `geometry`
- `materials` may have `texture`

Activity #1

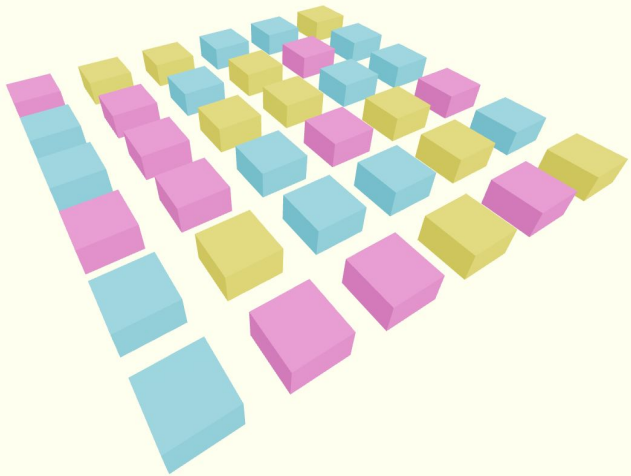
Create a grid of cubes.

Tip #1 💡
<boxGeometry args={[1,1,1]}/>

Specifies dimensions of cube

Tip #2 💡 **<group>**

Can use this to group individual meshes

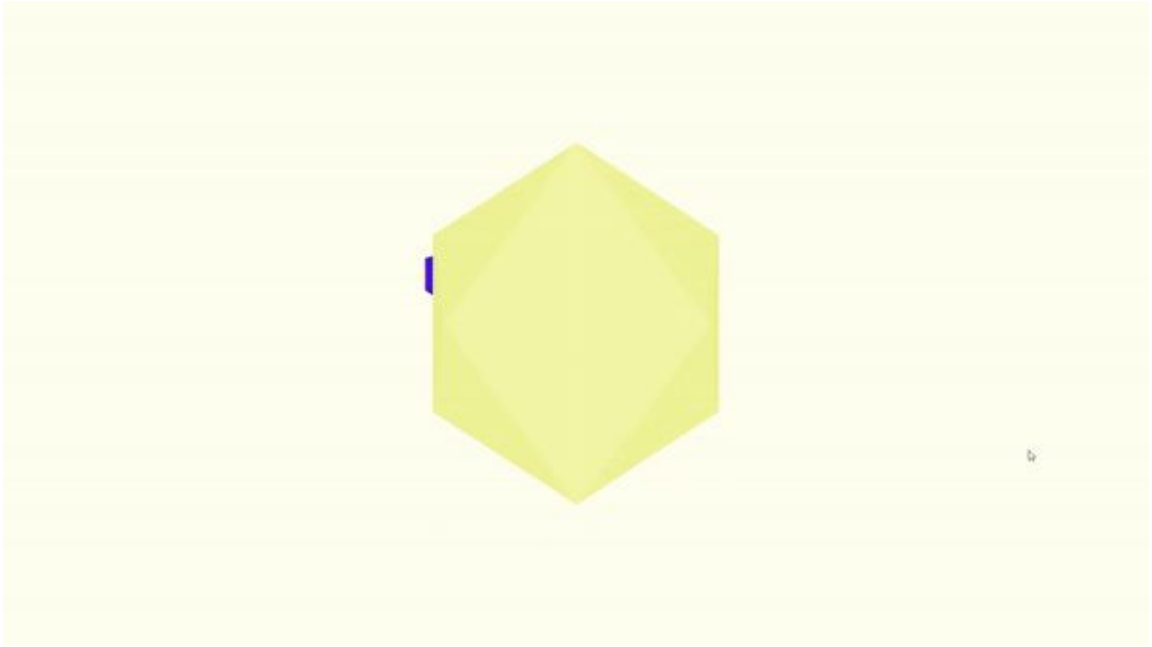


```
["#FF88DD", "#6EE1FF",
"#FFF053"]
```

2/ world

goal: how to animate things

(here’s what we want to achieve) **solar system**



useFrame

Allows execution of code on every rendered frame, like running effects & updating controls.

Will be using this to animate objects.

useRef

React hook, persists values between renders.

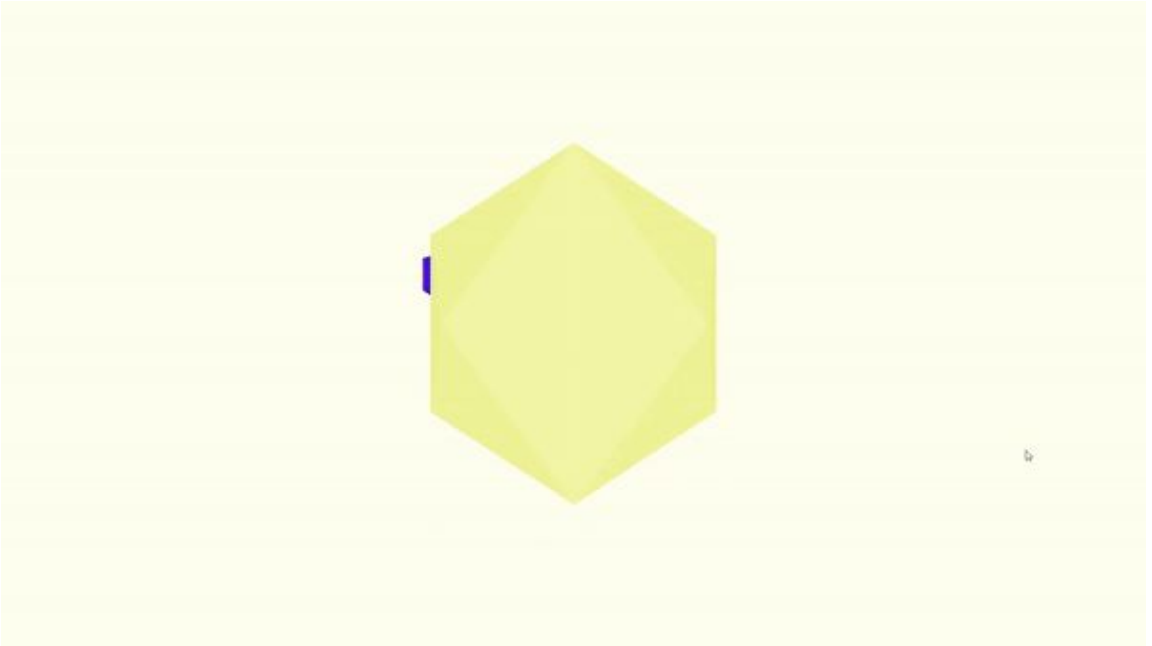
object3D

Used to group objects together to create a new object.

Position of children objects are relative to parents.

Activity #2

Add a moon.



OrbitControls

Lets the user spin or orbit the camera around some point.

useThree

Hook that gives access to the state model

State model contains default renderer, the scene, your camera, etc.

3/ goodbye world

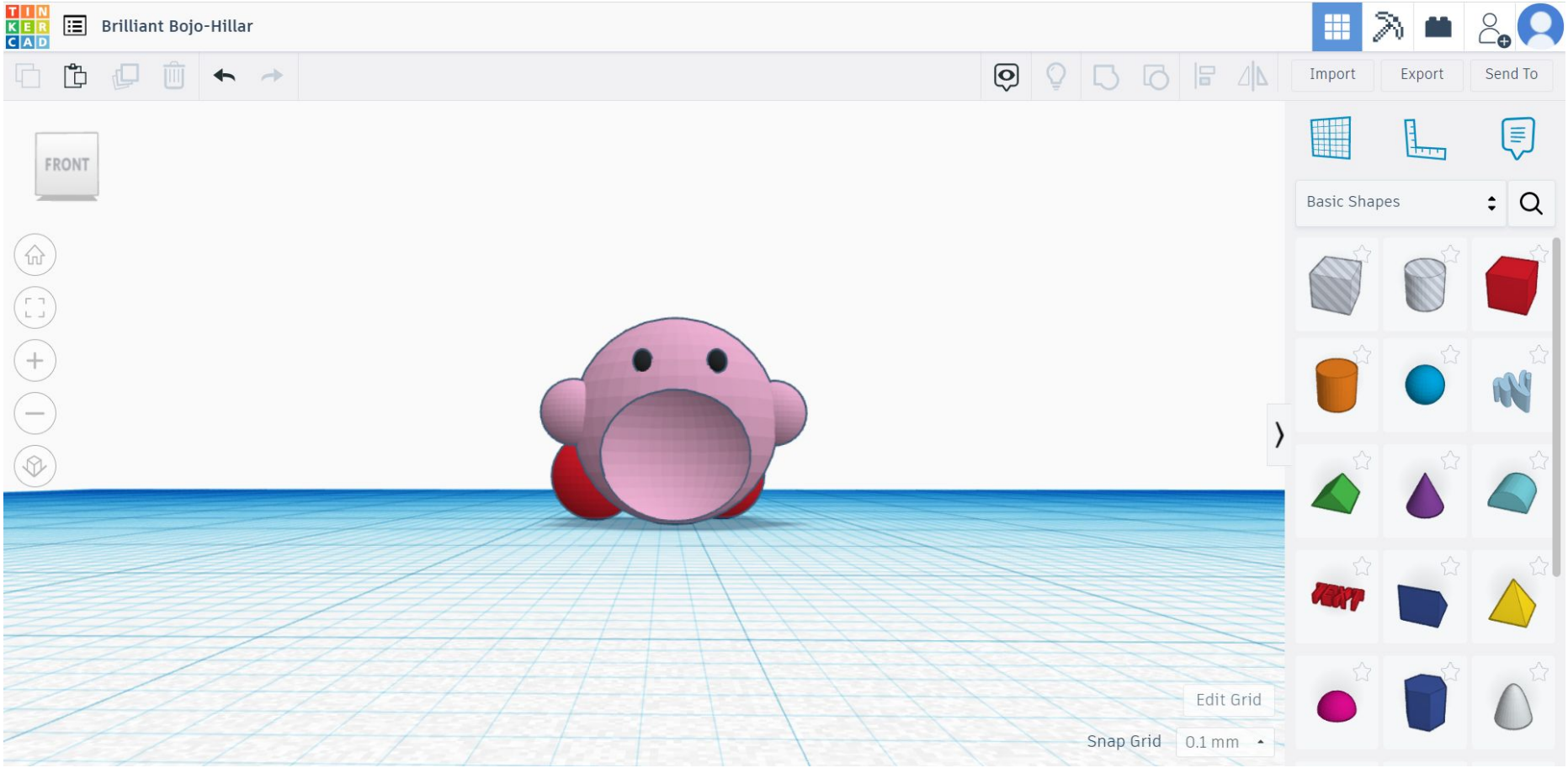
goal: importing models and controlling objects

moving away from primitives

- Primitives mainly use for learning or testing
- More complex models usually created somewhere else and imported
- https://threejs.org/examples

3D modelling software

- Blender
  - Look for donut tutorial 🍩
  - Free & open source!
- Maya (free for uni students ;D)
- ...
- Tinkercad?


Brilliant Bojo-Hillar
Import
Export
Send To
FRONT
Basic Shapes
Edit Grid
Snap Grid
0.1 mm

Activity #3

Use arrow keys to move the kirby.

Bonus: Spacebar to fire projectiles (useState)

thanks for
listening

some good resources

S

https://discoverthreejs.com/book/introduction/

A

https://threejs.org/manual/#en/fundamentals