↳ Found In Translation - Taiyo Watanabe

Overview

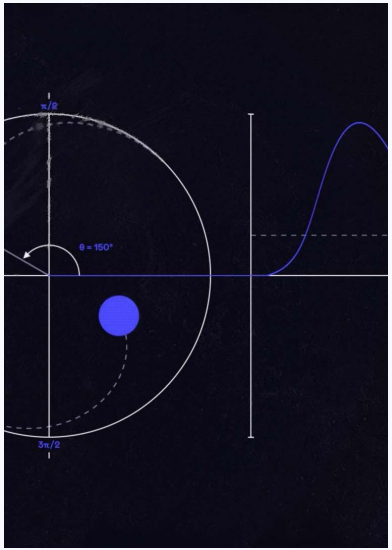
Theory

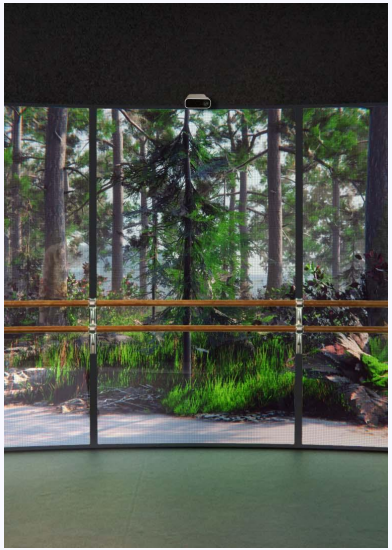
- ➔ A brief history
- ➔ Graphic Programming
- ➔ Disclaimers
- ➔ 3D Models
- ➔ Textures
- ➔ Lighting
- ➔ Cameras

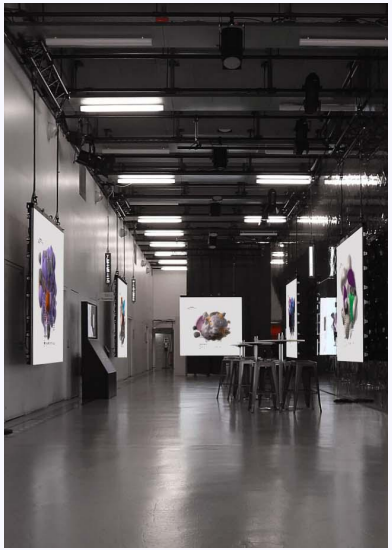
Practice

- ➔ Review the examples
- ➔ Create our first scene
- ➔ Load a 3d model
- ➔ Make it beautiful
- ➔ Instantiate million objects
- ➔ Projects
- ➔ Anything we want

Introduction





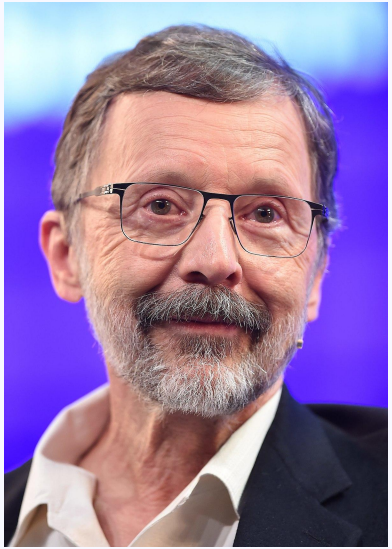


↳ My portfolio

A BRIEF HISTORY

↳ A Computer Animated Hand - Ed Catmull

Hall of fame

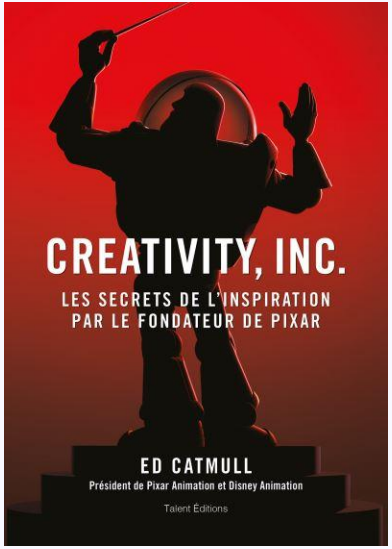


↳ Ed Catmull

↳ Ken Perlin

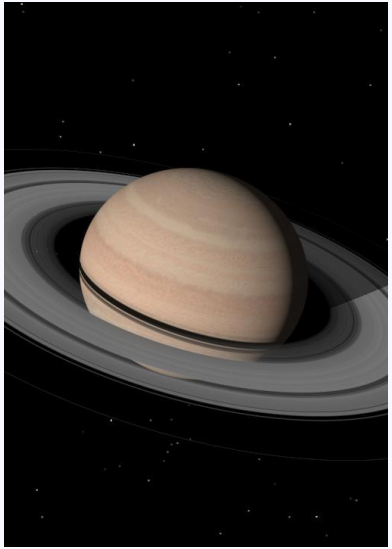
↳ Jim Blinn

Also known as

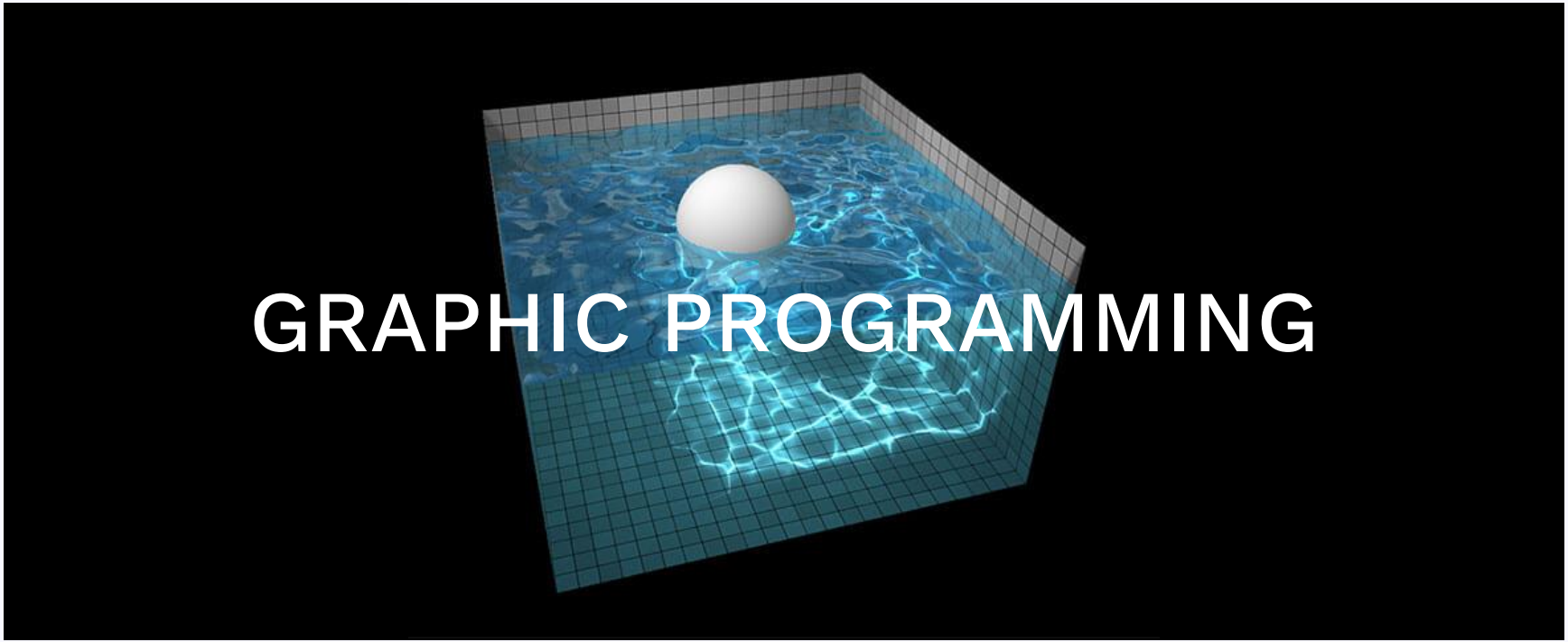


↳ Pixar co-founder

↳ Creator of Perlin noise



↳ NASA 3dman



↳ WebGL Water - Evan wallace

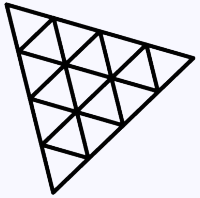
OpenGL and WebGL

OpenGL (Open Graphic Library) is a standardized set of functions for computing 2D and 3D images launched by Silicon Graphics in 1992.

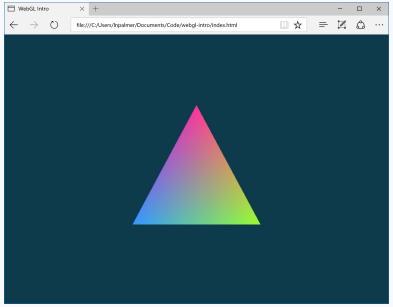
WebGL is created by Khronos Group in 2011 and allows to use the OpenGL ES standard within HTML5 pages and applications.

Both draw their potential from the graphics card which makes them excellent tools

Three.js



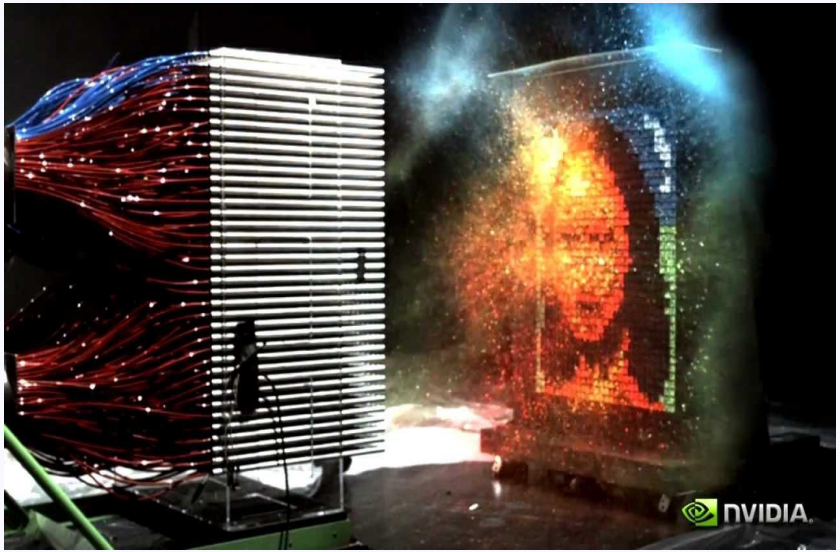
Three.js is a JavaScript library created by mrDoob in 2010, its principle is to be accessible to everyone, it allows to design and render 2D/3D scenes in a web browser in WebGL, CSS3D and SVG.



Its accessibility will make it an excellent tool in most cases, the best way to prove it to you is maybe to try to getting started only with WebGL.

↳ WebGL tutorial result

Parallel computing

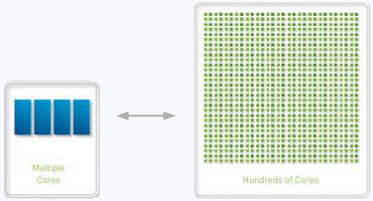


↳ Demo GPU vs CPU - Mythbusters

A CPU consists of four to eight CPU cores, while the GPU consists of hundreds of smaller cores.

This massively parallel architecture is what gives the GPU its high compute performance.

three.js gets its strength from this parallelization on the GPU.



↳ CPU/GPU cores

Shaders

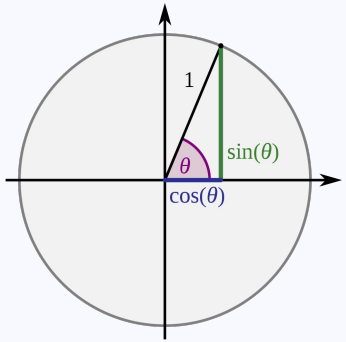
Shaders are computer programs, used to parameterize part of the rendering process carried out by a graphics card.

They can be used to describe the absorption/diffusion of light, textures, reflection/refraction, post-processing...

In three.js they are written in GLSL. They are ideal candidates for parallel execution on a GPU.

↳ Shader toy - Inigo Quilez

Mathematics

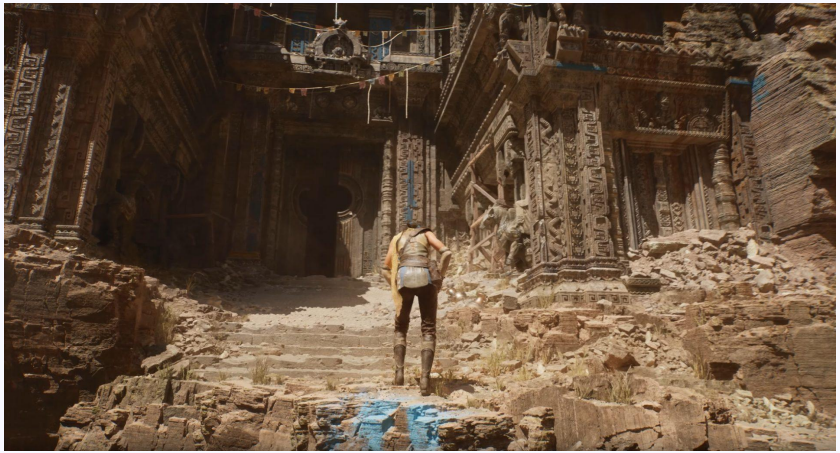


Fundamentals in mathematics will be useful, at least some trigonometry, if possible some linear algebra.

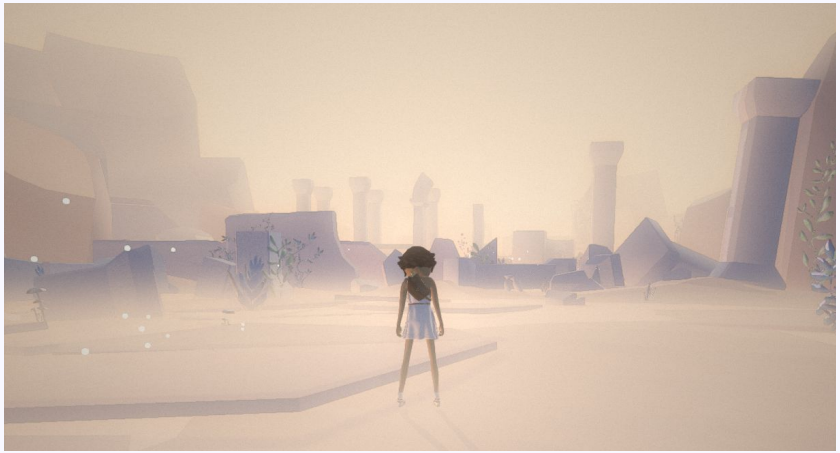
Concretely, this will allow you to understand rotation systems, to calculate a distance, to code an animation, to use matrix transformations, one day we will even be able to explain what a quaternion is.

↳ Ugly - Nikita Diakur

Expectations/reality



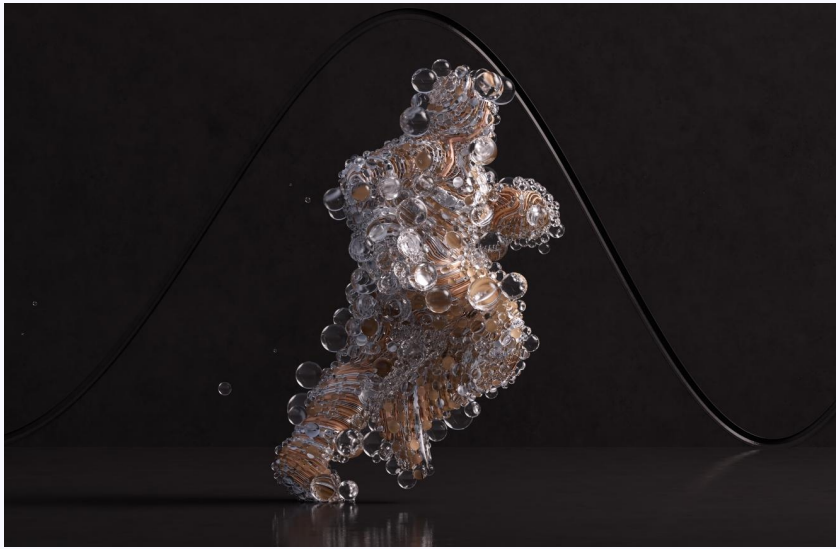
↳ Lumen in the Land of Nanite (Unreal Engine 5)



↳ Heraclos (three.js)

Three.js is behind WebGl which is behind OpenGl...
Let's estimate 4 to 8 years of 3D delay to avoid getting into trouble.

It's a trap!

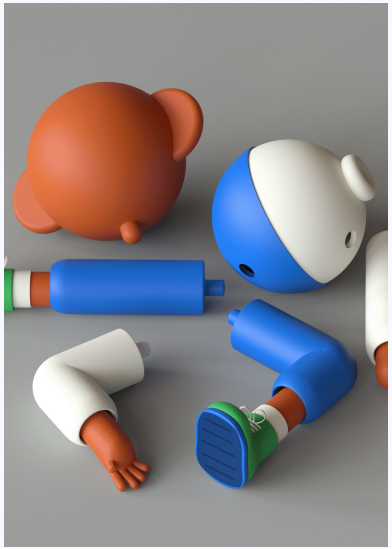


↳ Run forever - Universal Everything

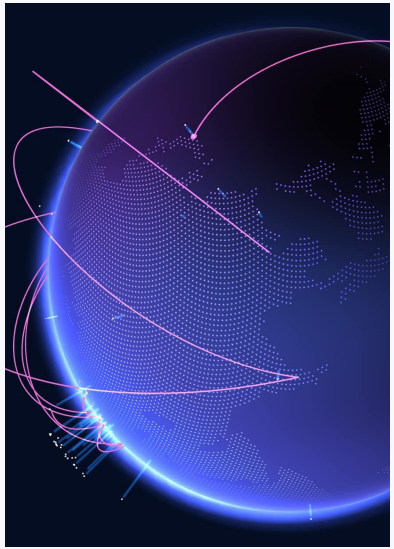
Photorealism, transparency, fur, physics, 3D scan, subsurface scattering...

Beware this project is going to be tricky.

Keep it simple (but look good)

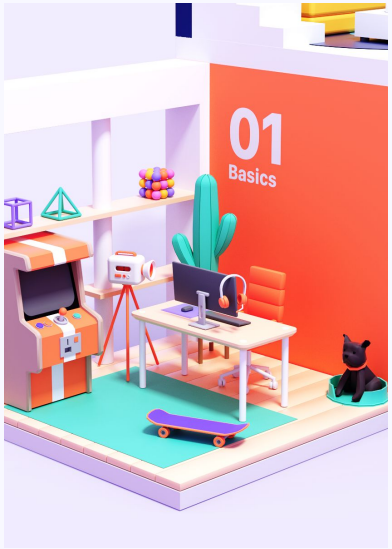


↳ César Pelizer

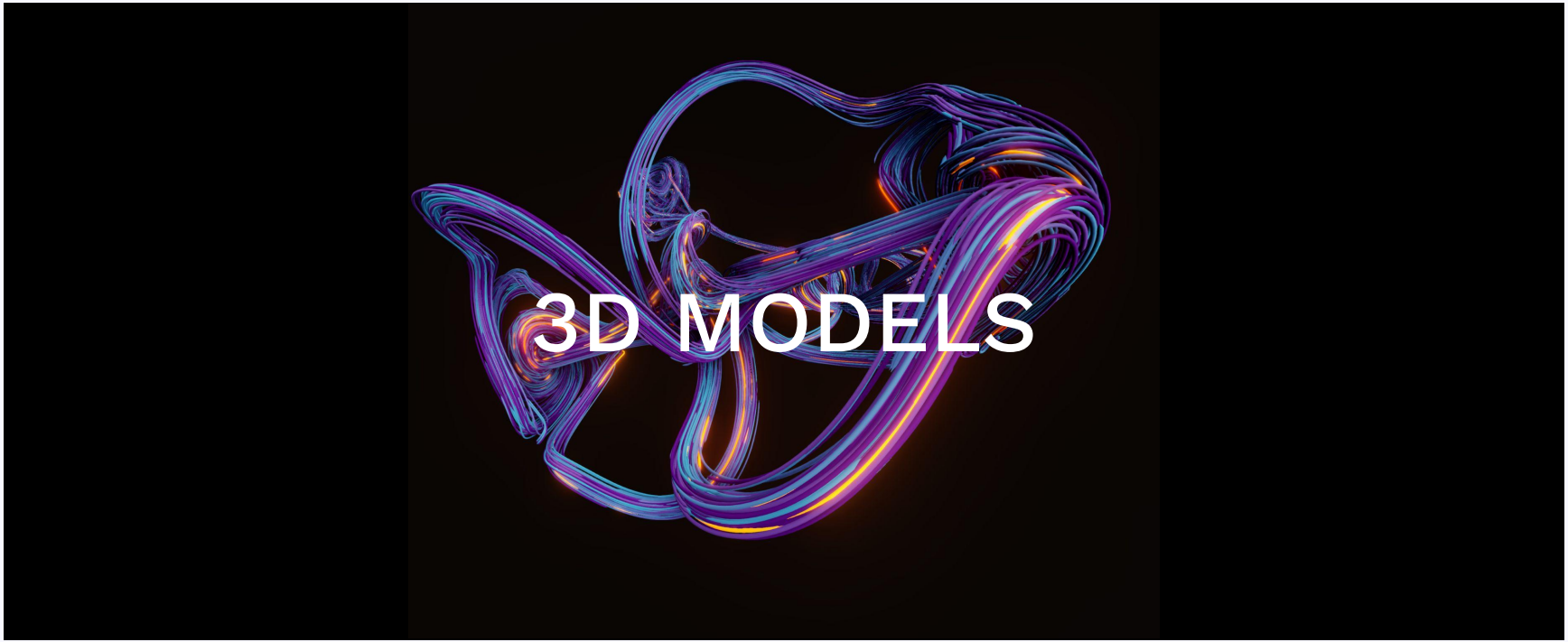


↳ Github globe

↳ Omar Aqil

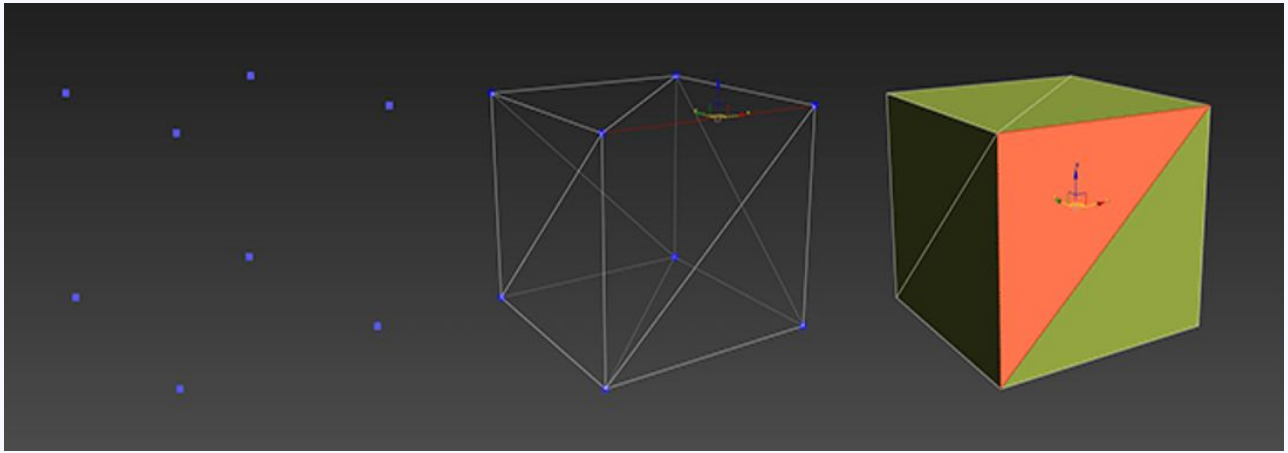


↳ Three.js journey

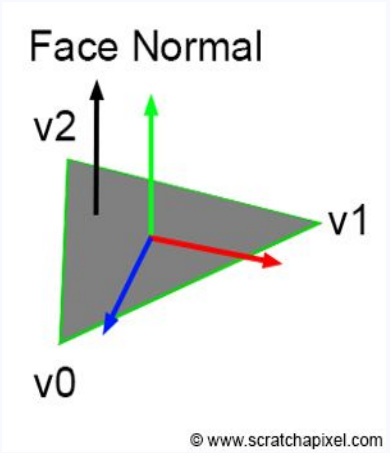


↳ Critical Dialogues - Variable.io

Back to basics



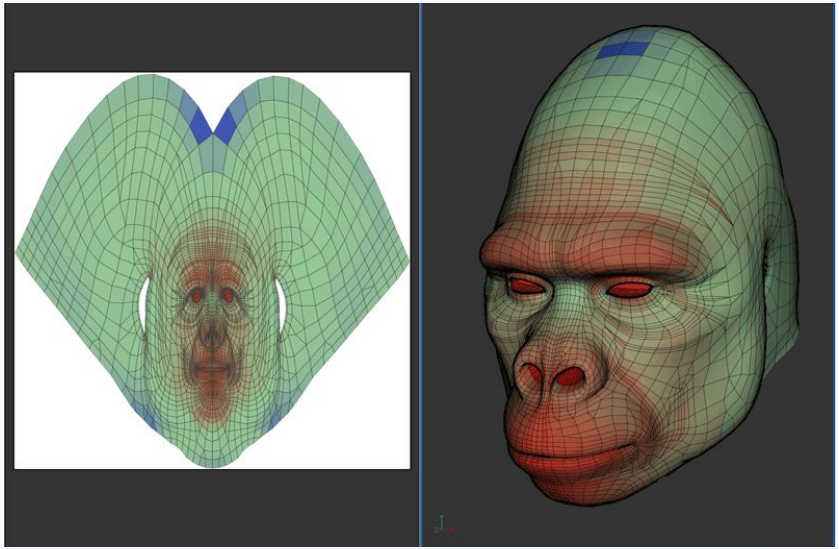
↳ Vertices, edges, triangles



↳ Face normal

Vertices, segments, triangles and normals are part of the basis of 3D, they are used to describe a 3D model.

Textures coordinates



↳ UV map example

When it is not about shaders, textures will often be images of 512, 1024, 2048 pixels (cf : hexadecimal system)...

The UV coordinate system is used to map a 2d texture to a 3D model, these values are encoded for each triangle vertex.

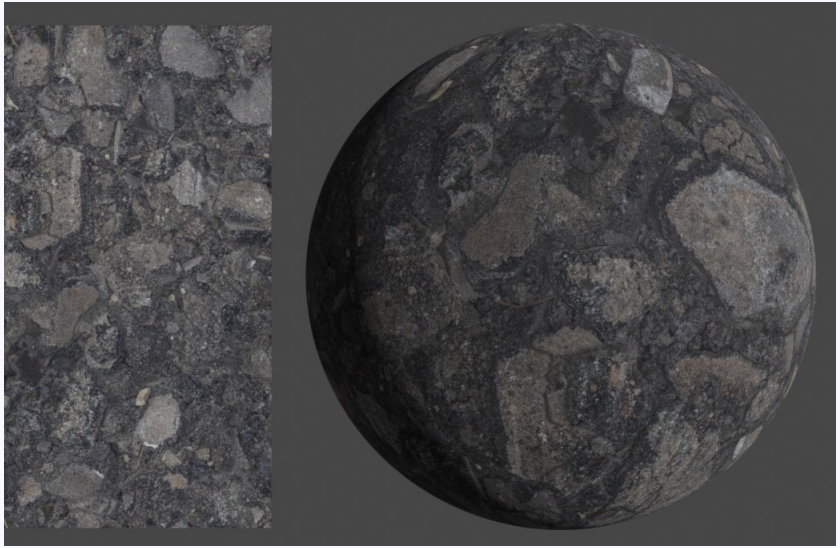
3D formats

FBX (Filmbox), OBJ (object file) and glTF (GL Transmission Format) are the main 3d formats I was confronted with in the communication, these files contain all the data mentioned above.

In the workshop we will focus on the FBX format because it is often accessible and compatible with Unity. However, three.js favors the use of the glTF format because it is focused on runtime assets delivery, it is compact to transmit and fast to load.

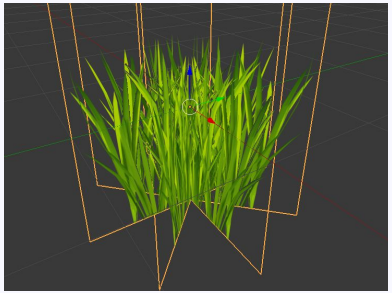
↳ Geologies - Oscar Gonzales Diez

Diffuse and alpha maps



↳ Diffuse map example

The diffuse map is used to color your model. The alpha map is used to designate the amount of transparency or translucency of its surface.

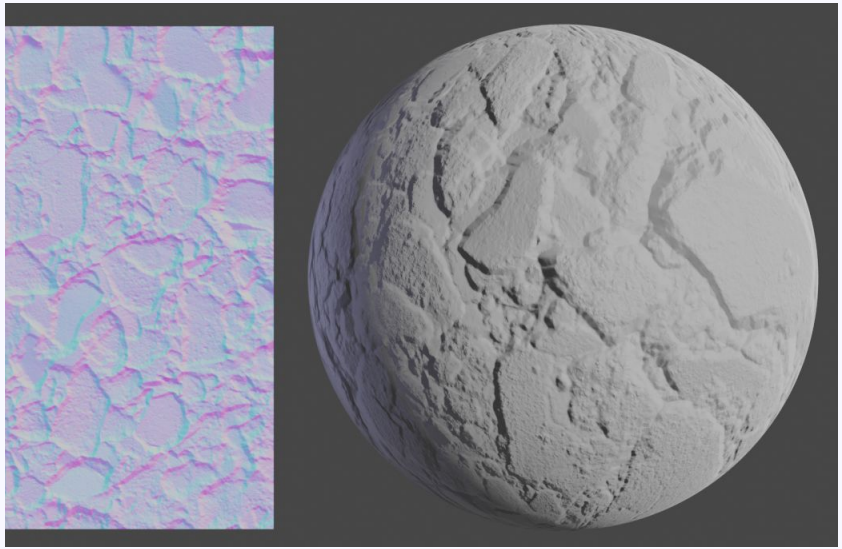


↳ Alpha map example

Normal map

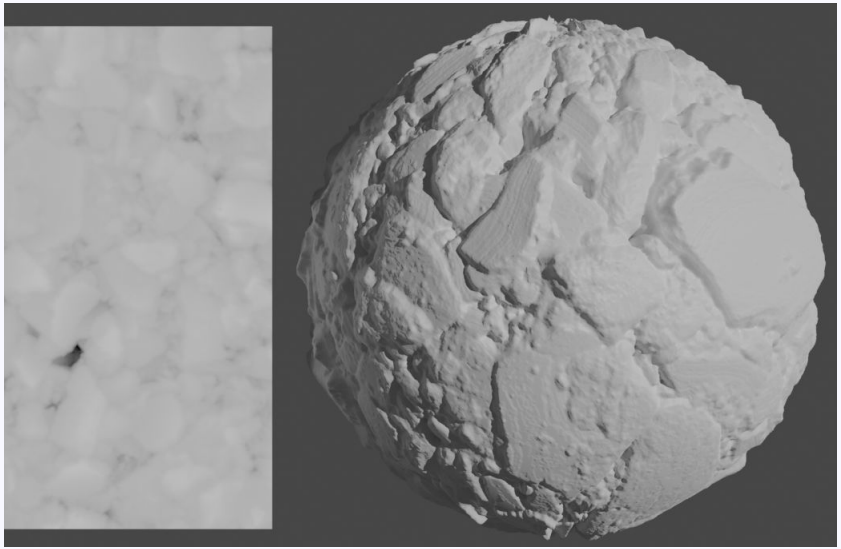
The normal map is used to represent a highly detailed surface mesh using a more simplified low polygon count version of the mesh.

Each color represents a different axis of direction. So essentially, the computer can easily understand the surface.



↳ Normal map example

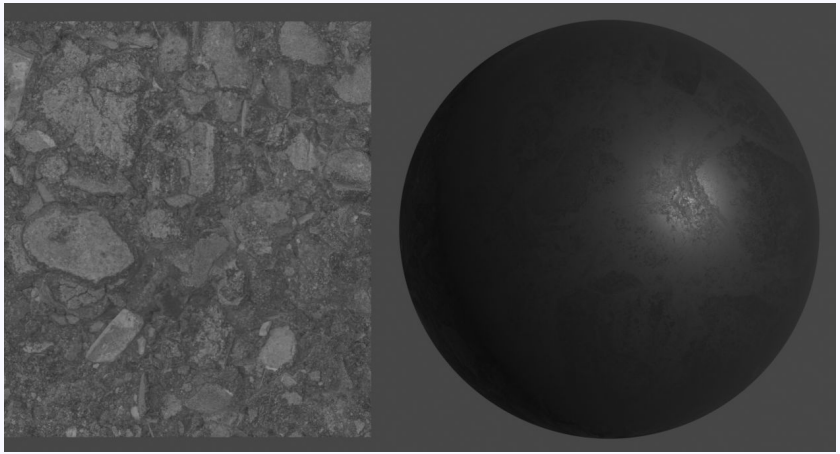
Displacement map



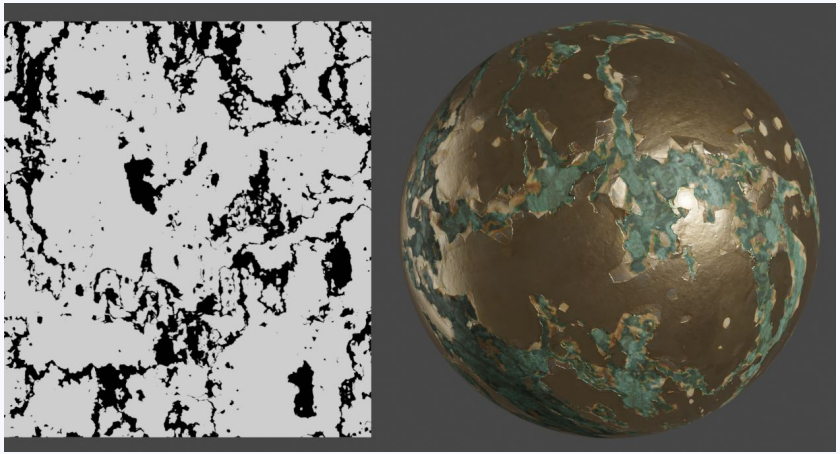
↳ Displacement map example

The displacement map allows a model to represent visual detail not modeled or sculpted into the original mesh and can be used as a method for transferring sculpted detail from one model to another.

Roughness and metallic maps



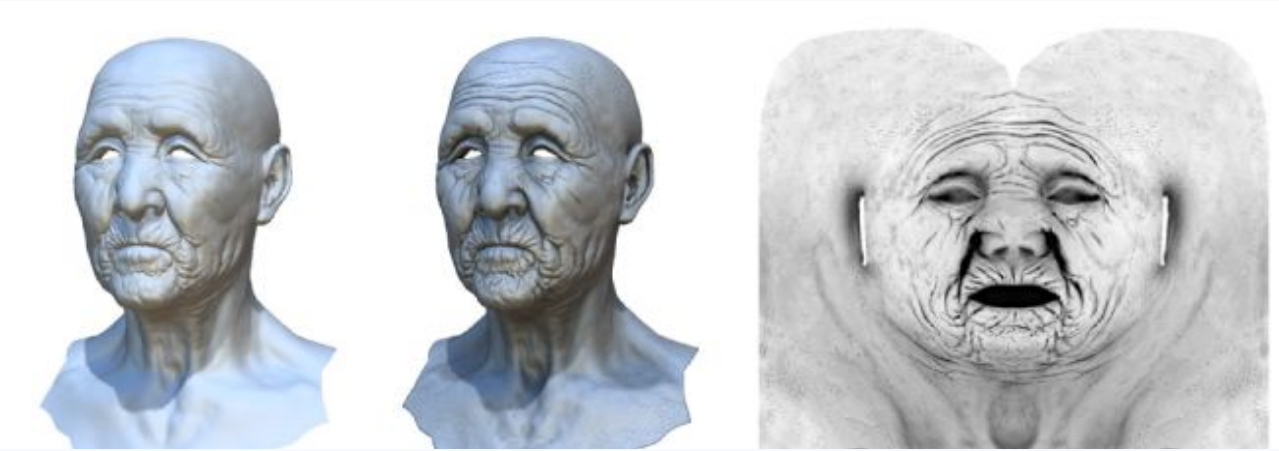
↳ Roughness map example



↳ Metallic map example

The roughness map (aka gloss map) describes the surface irregularities that cause light diffusion. The metallic map is used to define which areas of a material denote raw metal.

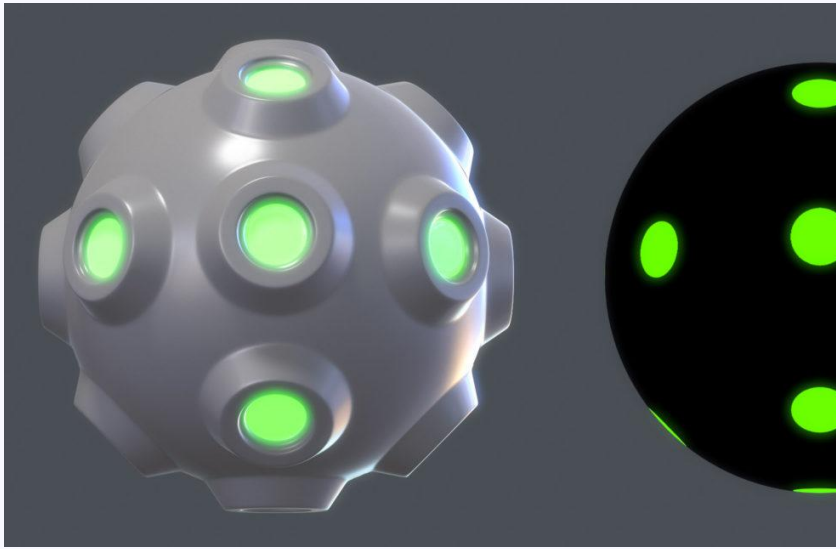
Ambiant occlusion map



↳ Ambiant occlusion example

AO map define the attenuation and occlusion of light between surface details and other objects in the scene and then adds realistic shading to models.

Emissive map

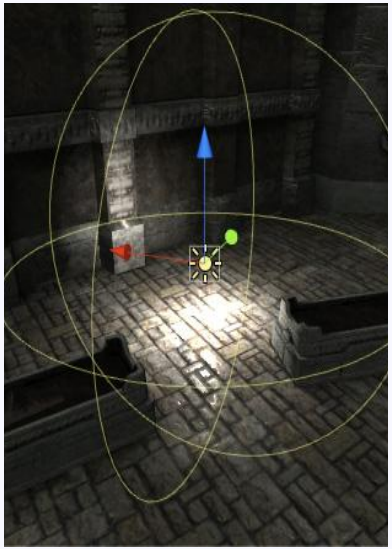


↳ Emissive map example

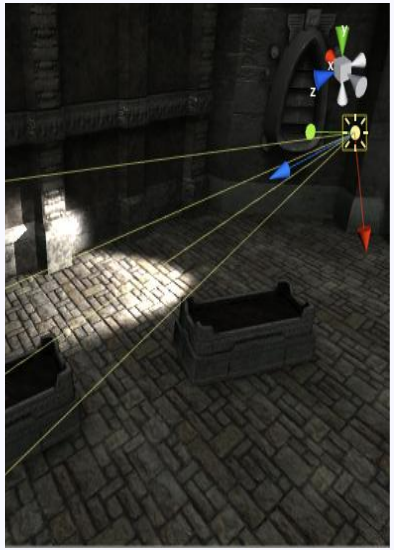
The emissive map is used to make your model emit light.

↳ Light transformations - Media.Work

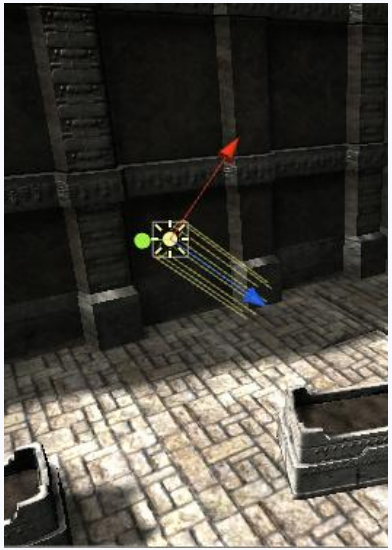
Main types of light



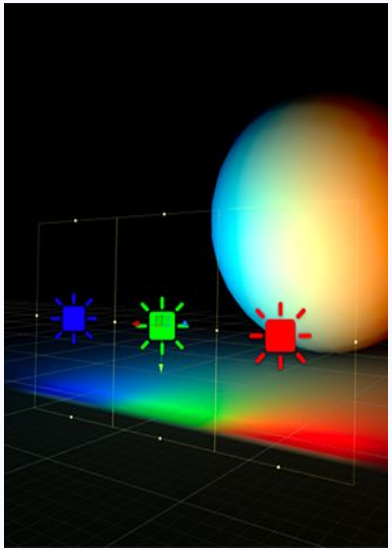
↳ Point light



↳ Spotlight



↳ Directional light



↳ Area light

HDRI

↳ HDRI environment

HDRI (High Dynamic Range Image) are used in graphics as an alternative to artificial lighting.

They allow you to recreate the colors and lights of the photograph in the 3D scene, with each pixel of the HDR image serving as a light source.

↳ Hyperbolic - CodeParade

Perspective and orthographic cameras

↳ Perspective camera

↳ Orthographic camera

The perspective and orthographic modes of viewing a scene are known as camera projections. Both perspective and orthographic cameras have a limit on how far they can “see” from their current position, this is known as the far clipping plane.

Resources

- Intro to three.js
- Public 3D assets library
- Realistic lighting setup on three.js
- HDRI example on three.js
- Git and Github for Poets
- The Coding Train
- Resources for creative coding