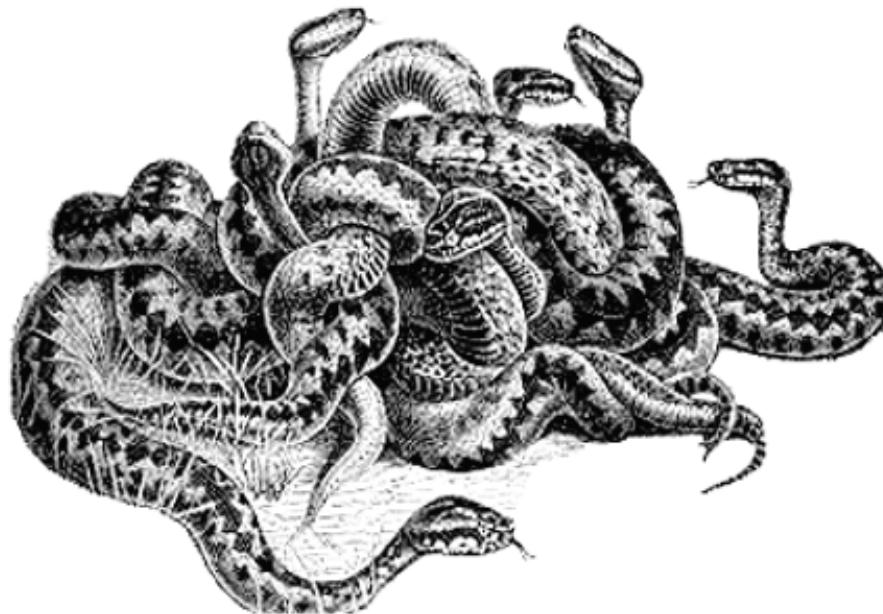


Ah, ¿que al final lo de las matrices servía para algo?



Destripando gráficos de los videojuegos

The Definitive Guide

O RLY?



@AdrianensisEC

Dificultad 'Easy'



Conceptos útiles

50% Matemáticas:

Modelo 3D, malla, vértices, vector, punto, matriz, transformaciones afines, rotación, escalado, translación

50% Programación:

APIs de renderizado: OpenGL / Directx, Motor gráfico, Buffer



¡DOBLE SABOR!
¡SENSACIÓN ÚNICA!



Welcome to my nightmare.

¿Qué vamos a ver?

¿Qué son los gráficos por ordenador?

Evolución de los gráficos

Bajo el capó: El motor (gráfico)

Cómo funciona WebGL

¿Qué son los gráficos por ordenador?

“usually referring to image data created by a computer specifically with help from specialized graphical hardware and software” - Wikipedia

PUES VALE



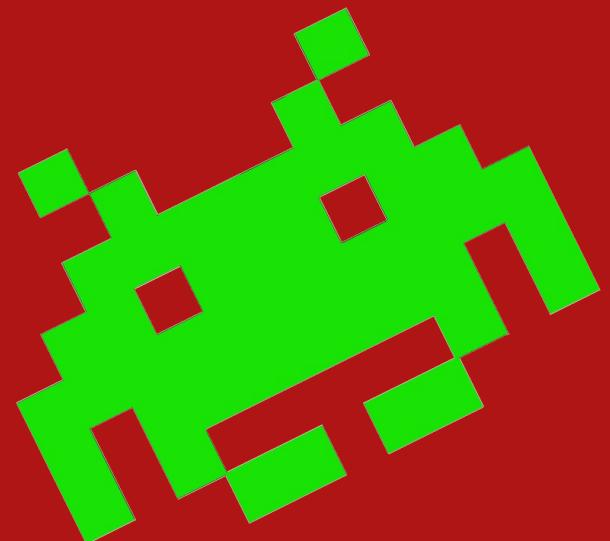
Evolución de los gráficos

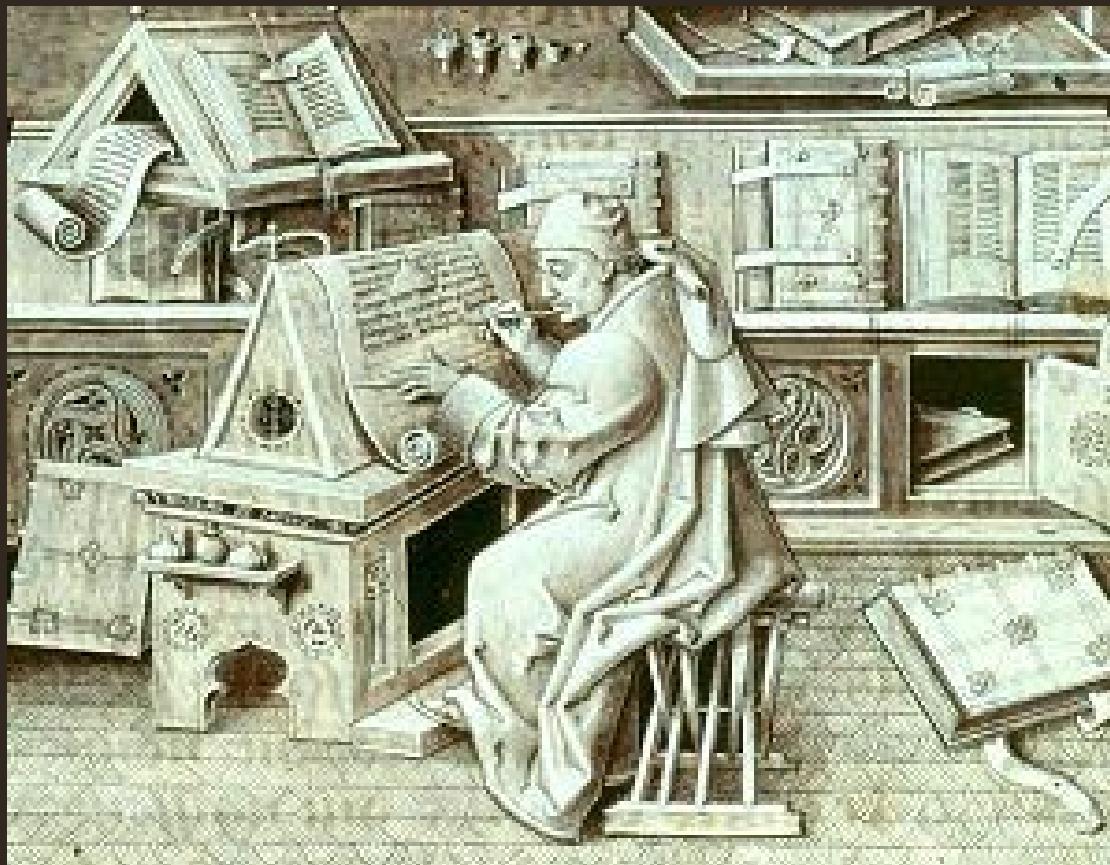
Años 70

Sprites

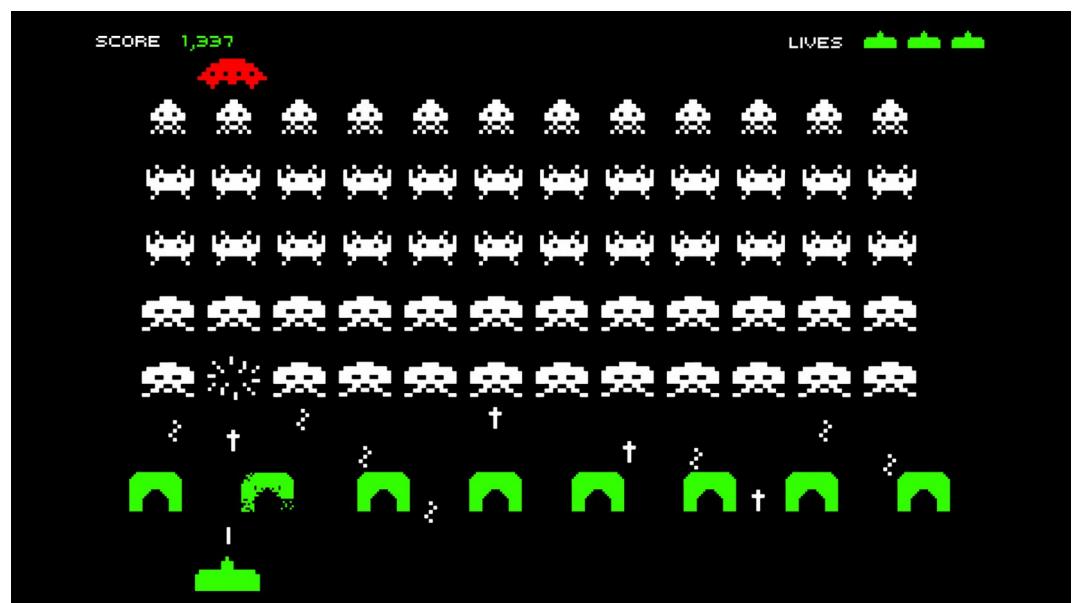
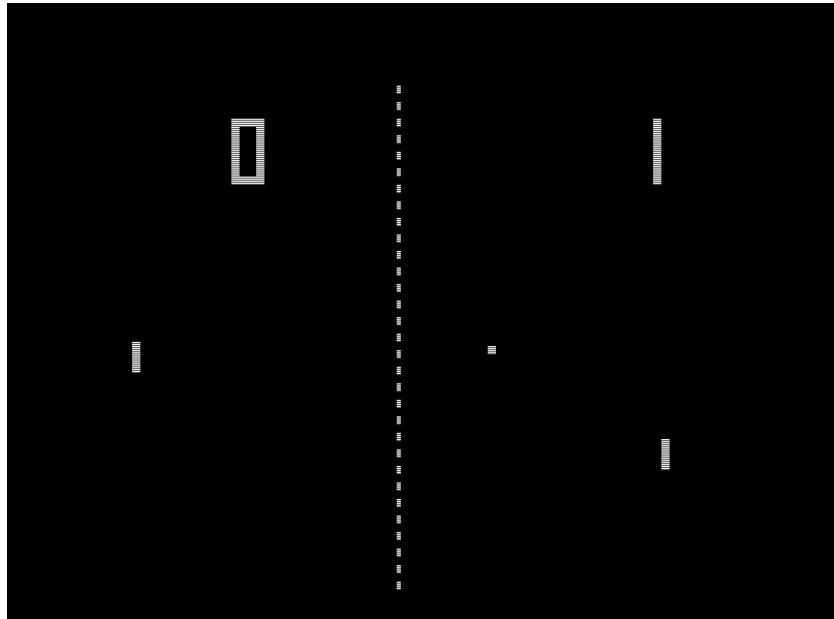
Polígonos

Aceleración por hardware

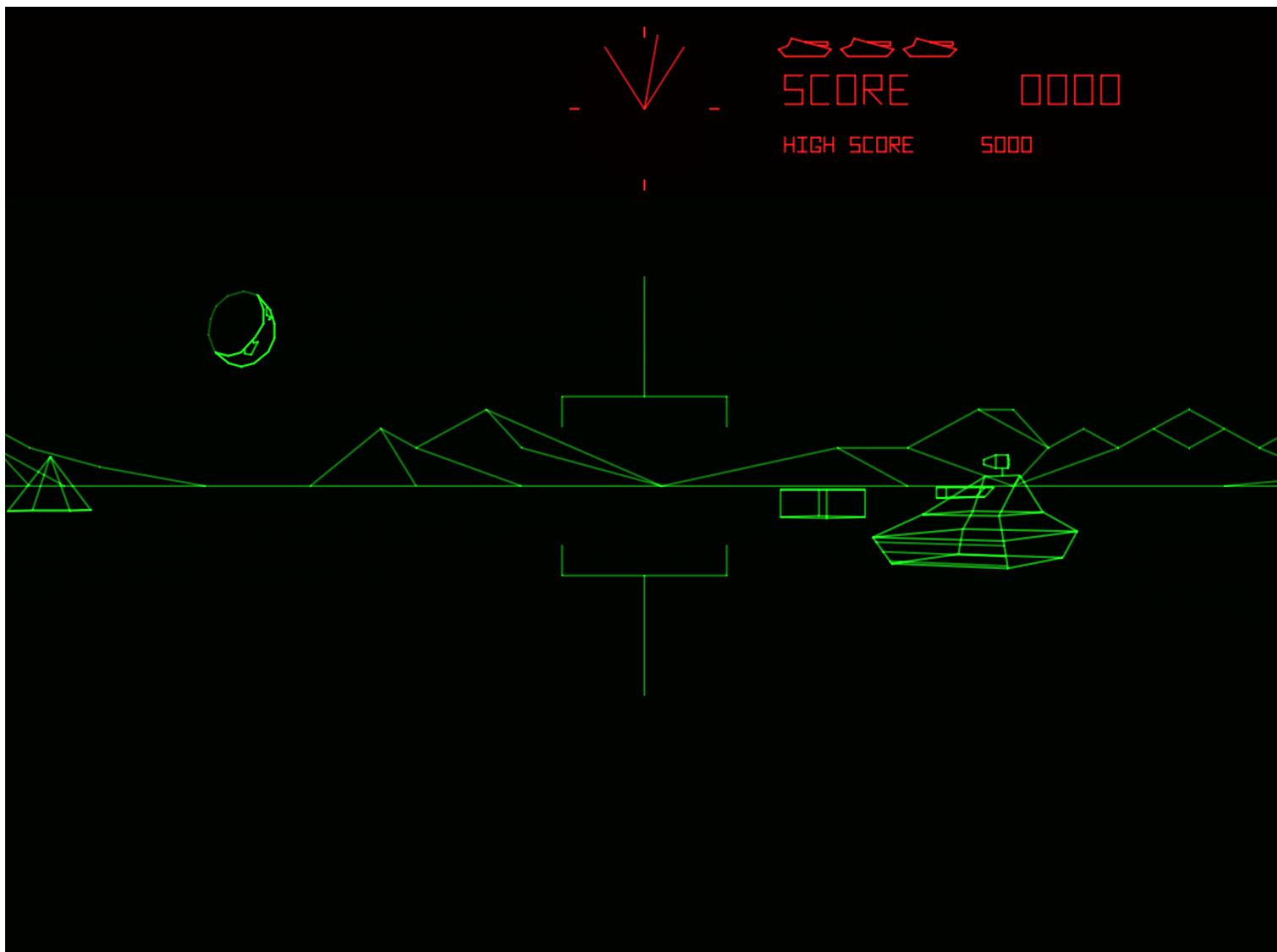




Grabado en madera en el que se puede apreciar a Matusalén escribiendo el código del pong, en las oficinas de Atari.



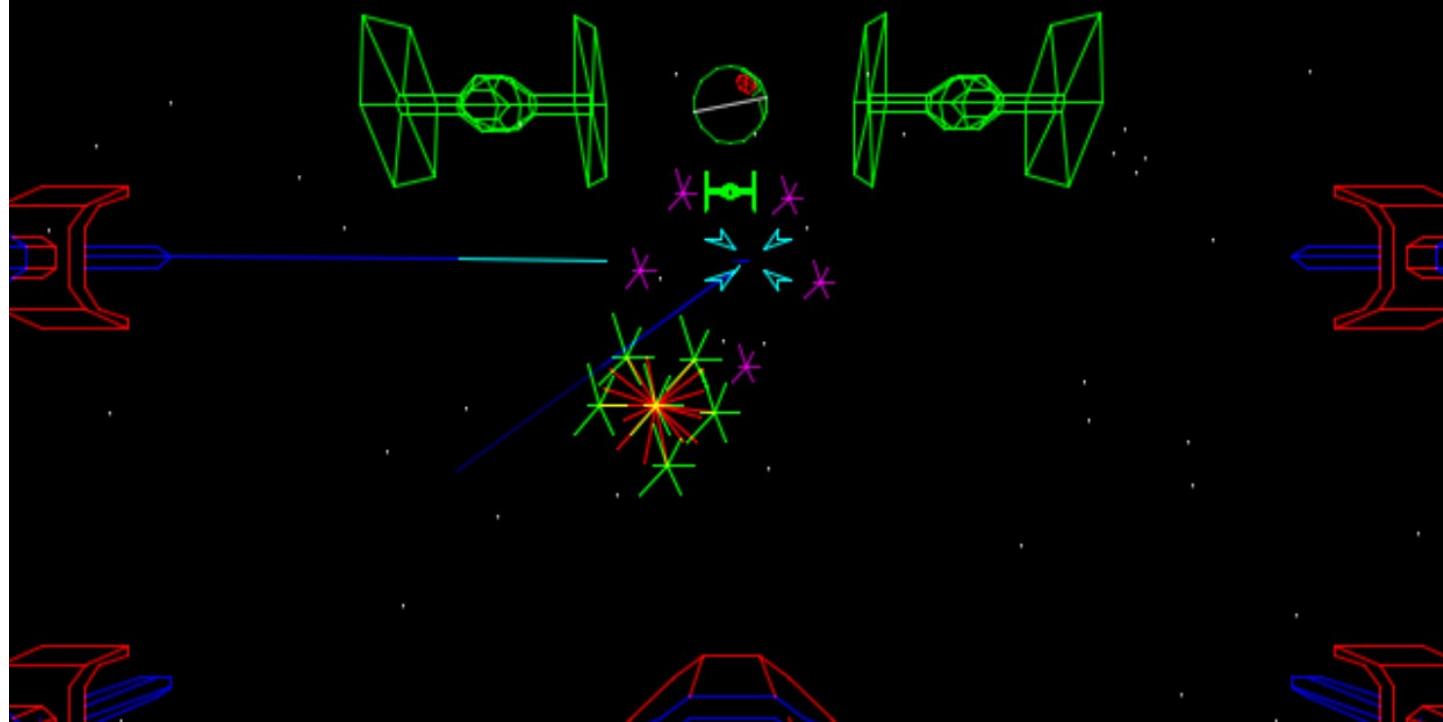




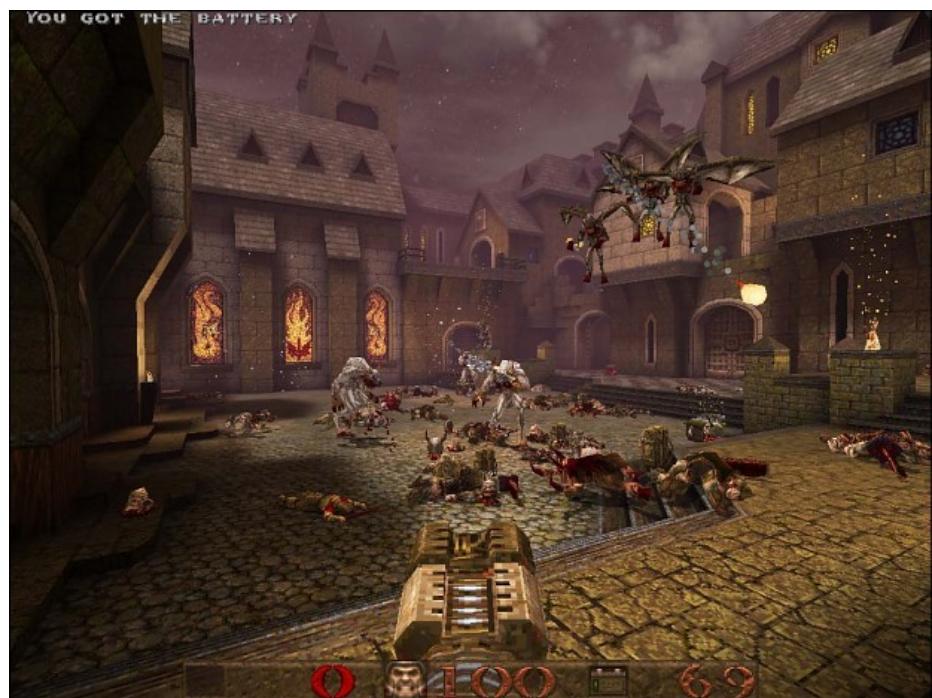
SCORE
58,066
33



2 WAVE









Bajo el capó: El motor (gráfico)

Gestiona:

Objetos 3D / 2D

Texturas

Animaciones

Escenas

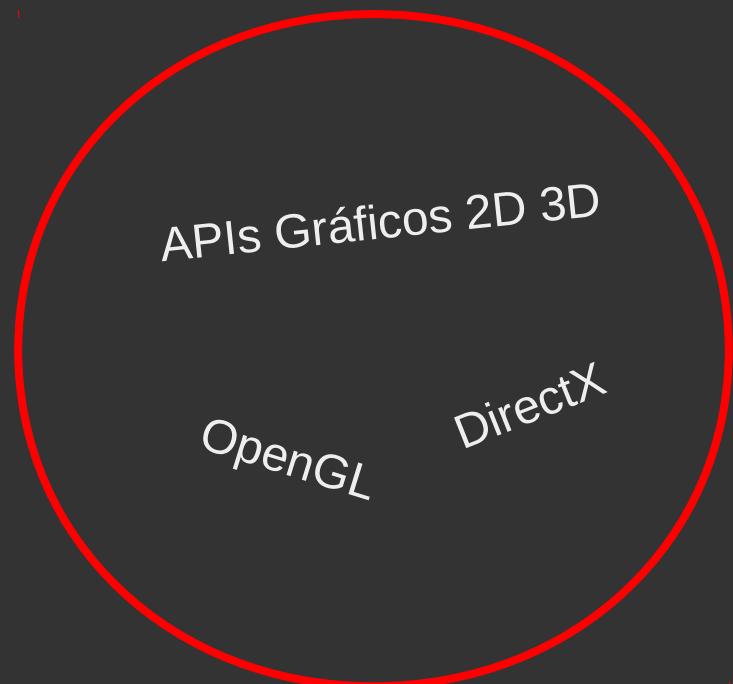
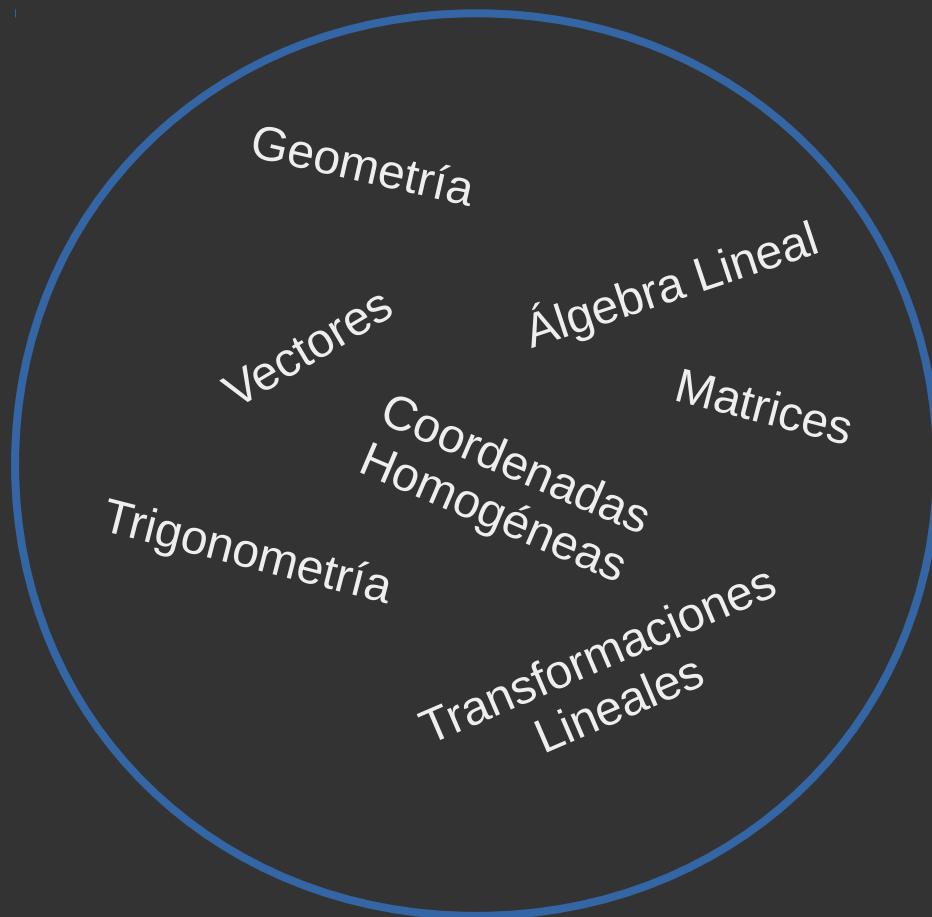
Luces

Cámaras

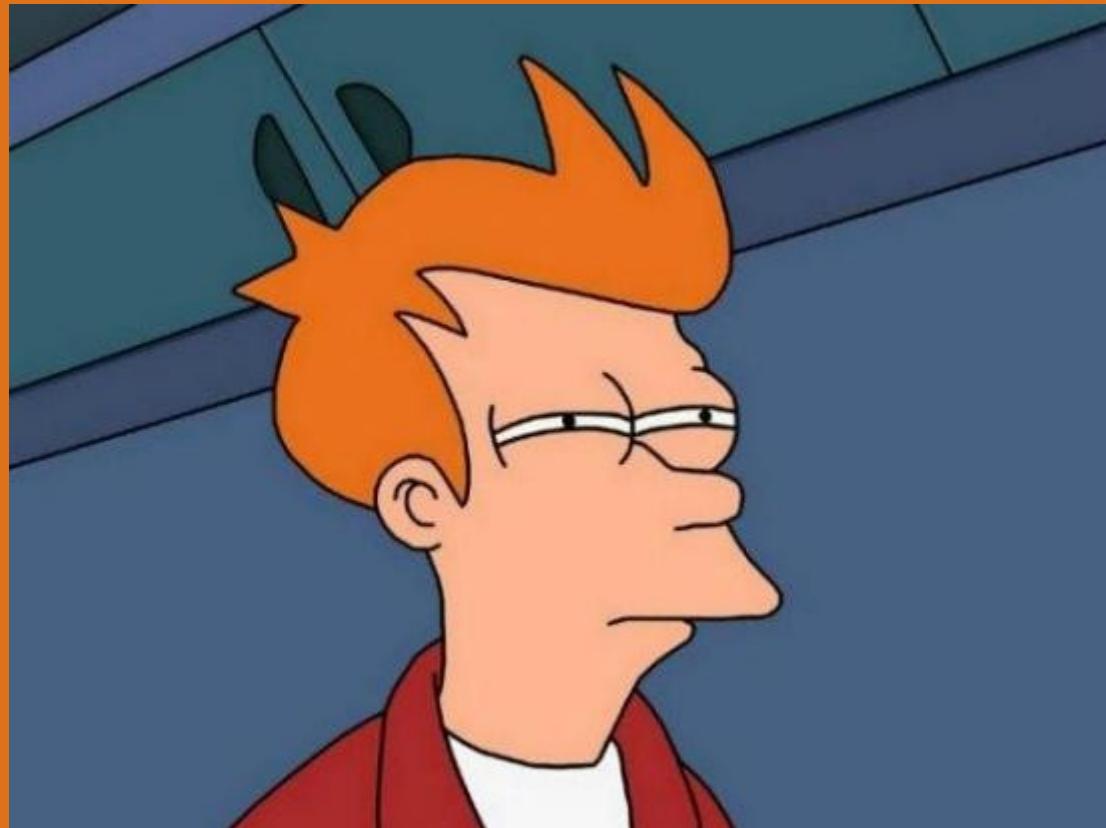
Efectos Especiales



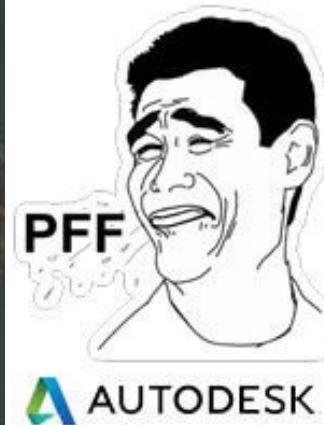
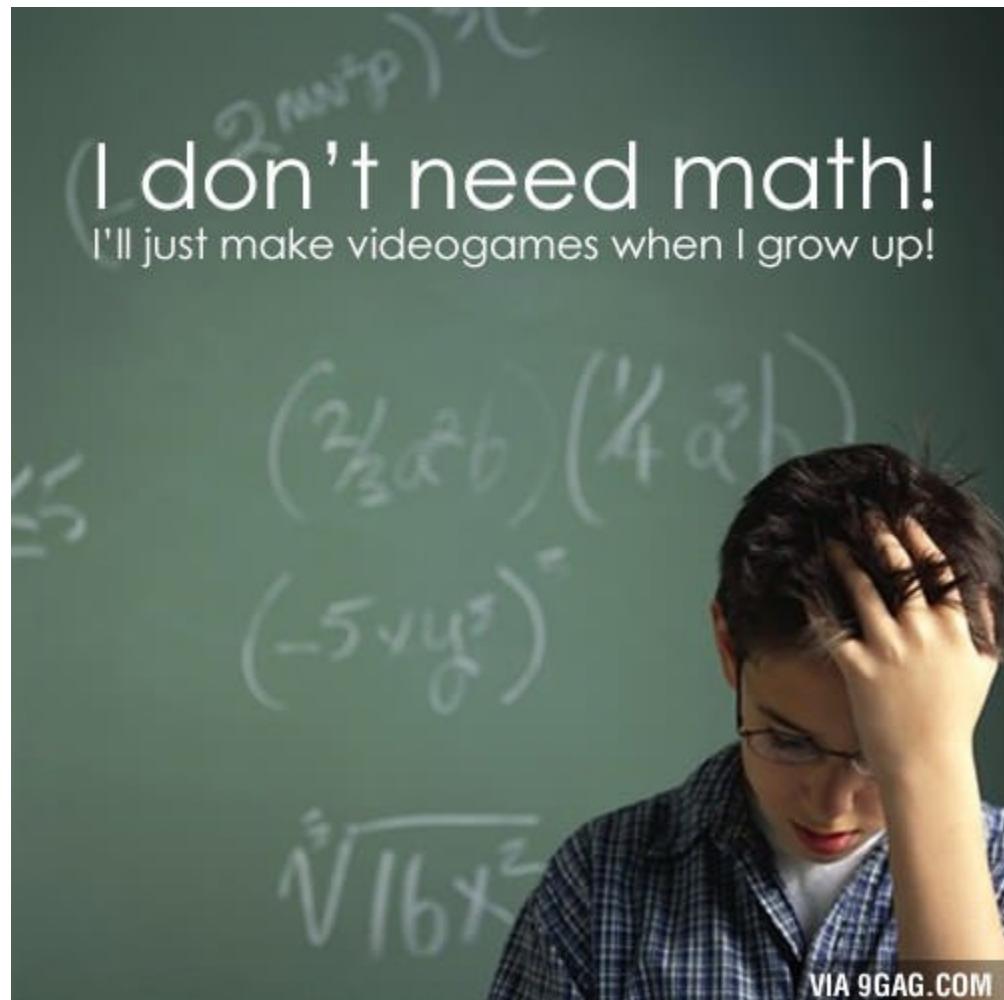
¿Qué usa el motor gráfico para funcionar?



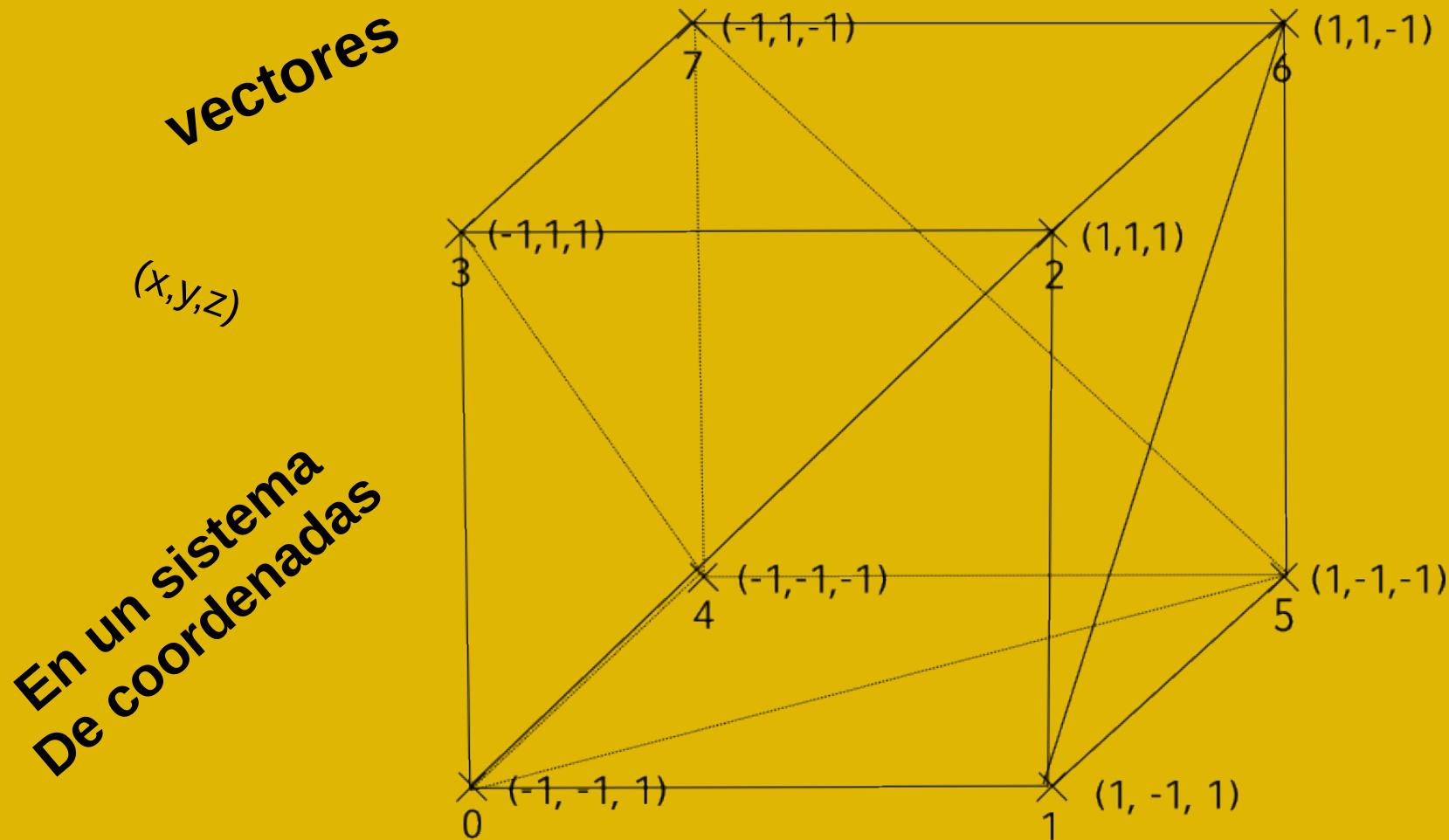
Pero si esto son...

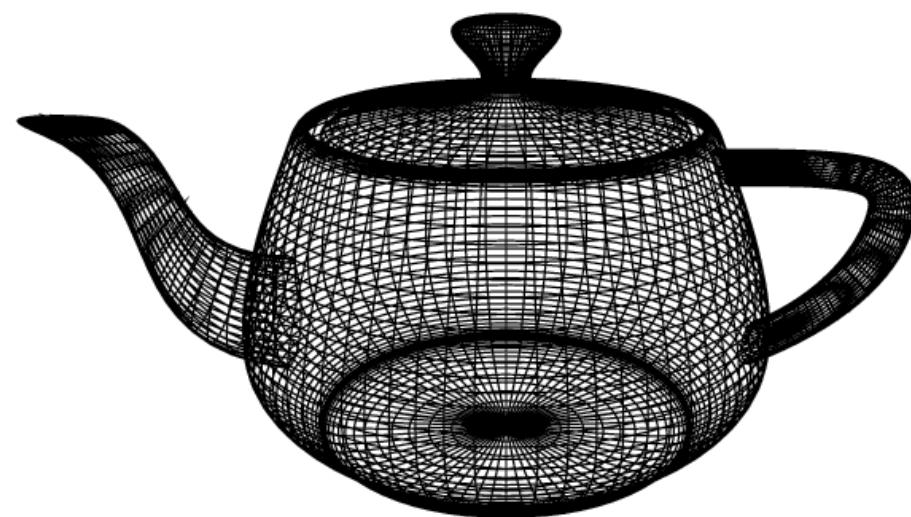
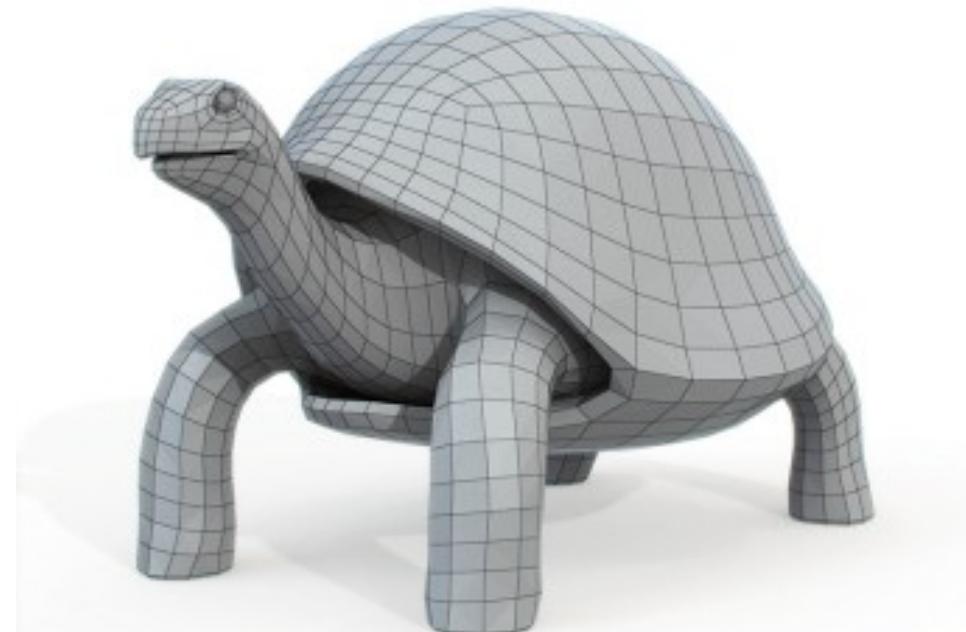
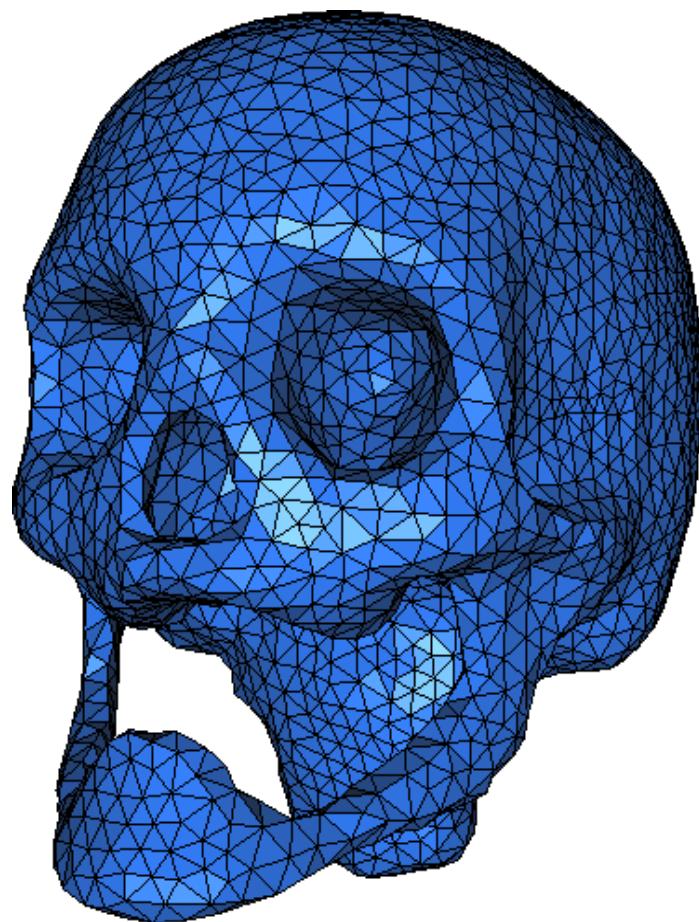


¡Las matemáticas que
dí en primero!

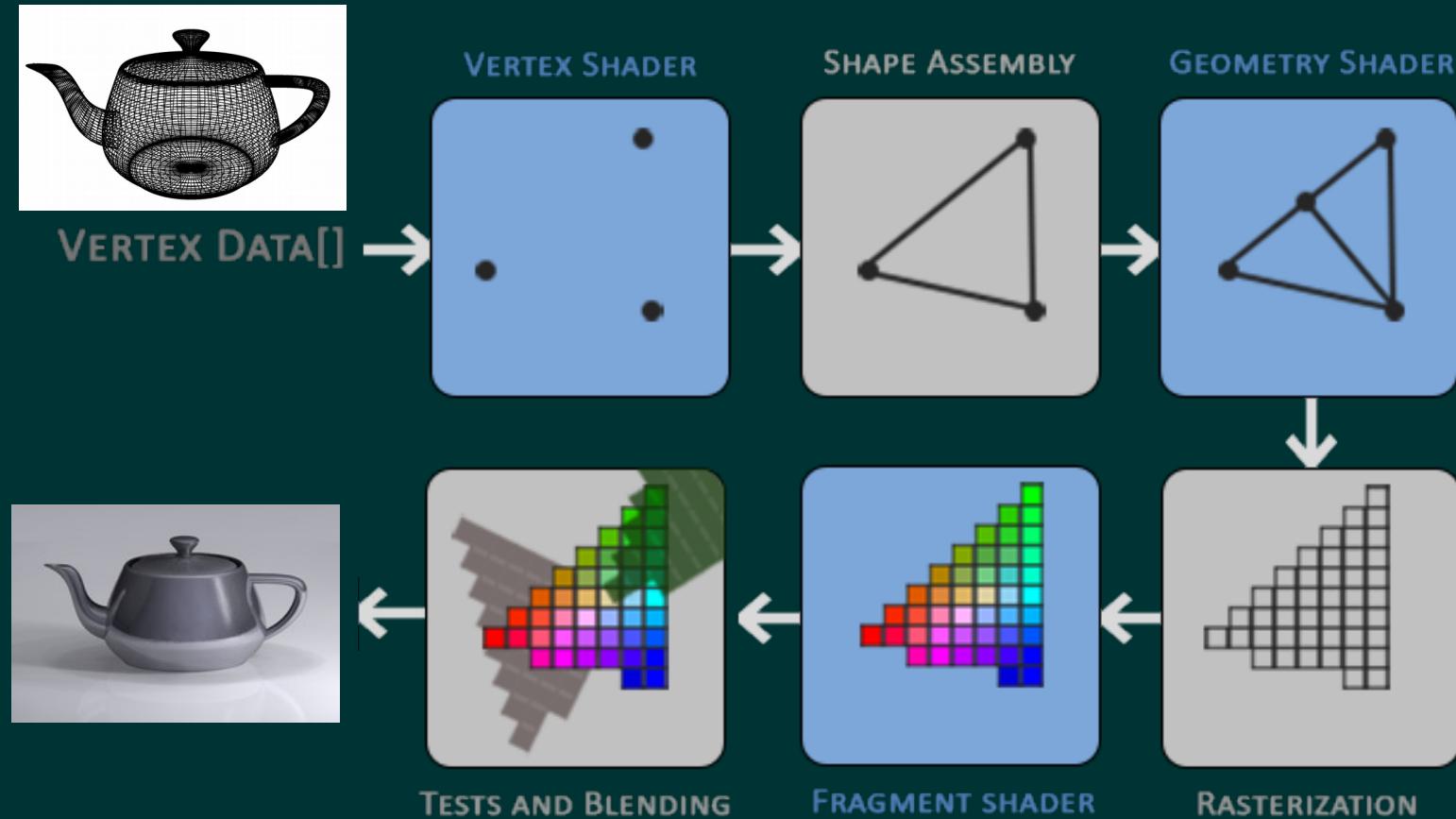


Representar un objeto





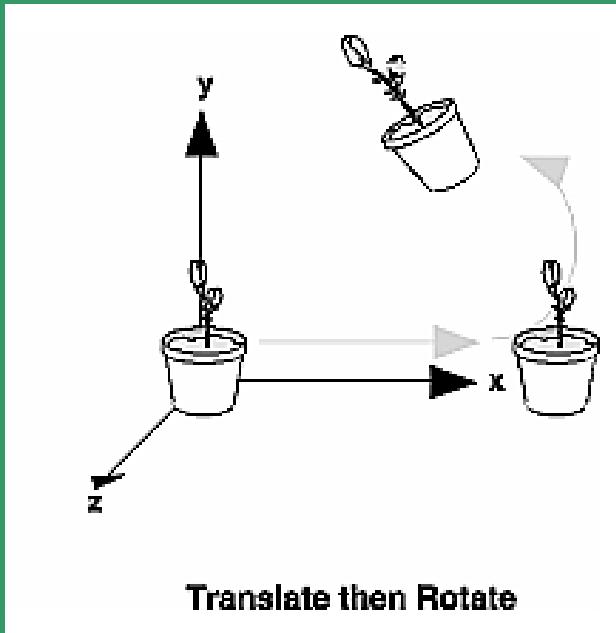
¿Cómo se dibujan los objetos en mi pantalla?



¿Pero porqué no se mueve?



Transformaciones



Translation matrix

$$\begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + dx \\ y + dy \\ z + dz \\ 1 \end{bmatrix}$$

Existen más transformaciones: rotación, escalado, perspectiva, ortogonal...

Las transformaciones
....!

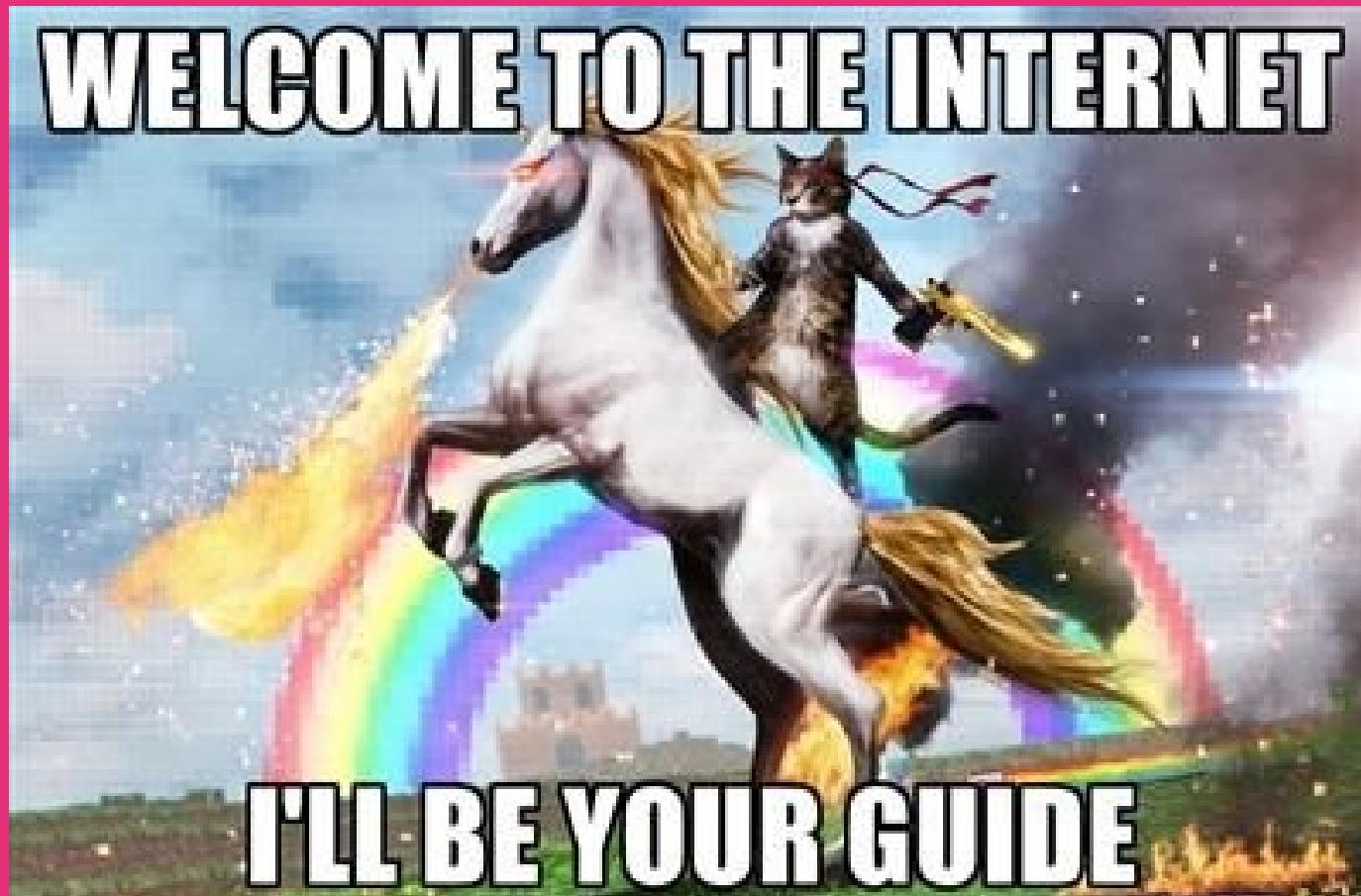
¡Ya basta de tantas
matemáticas!



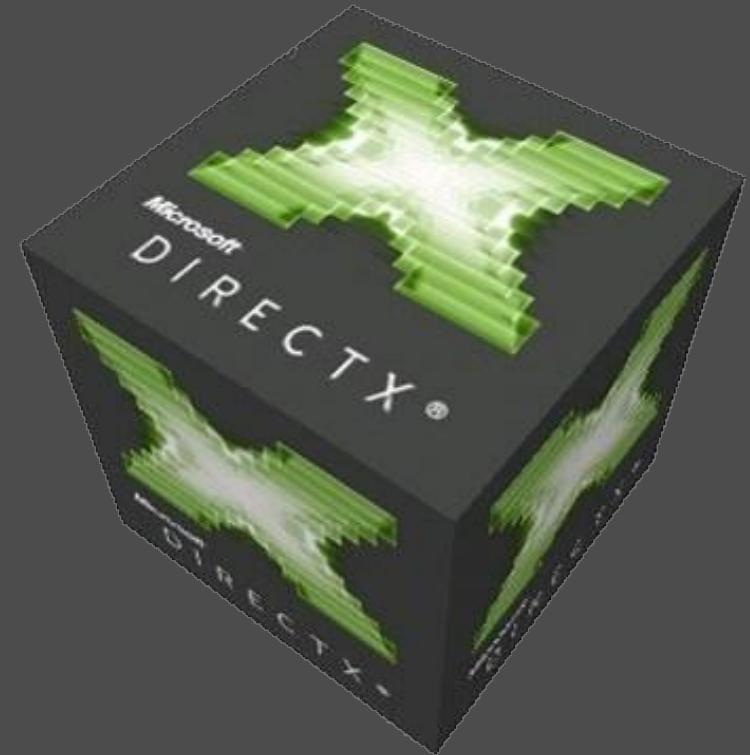
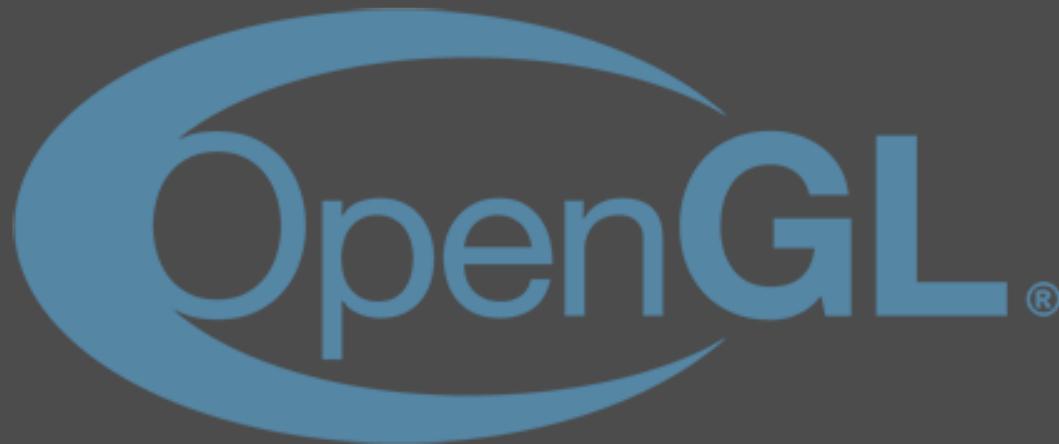
Talk is cheap. Show me the code.

— *Linus Torvalds* —

Bienvenidos al maravilloso mundo de las APIs de gráficos 3D



¿Qué es una API de gráficos 3D?





The logo for WebGL features the word "WebGL" in a bold, dark red sans-serif font. The letters are partially enclosed by a thick, dark red oval that has a slight upward curve at the bottom right corner. The background is a solid yellow color.

WebGL



Let's code!



¿Cómo se dibuja en un videojuego?

Enviar datos a memoria gráfica.

Bucle del juego:

Se **limpia** la pantalla.

Se **actualizan** los objetos

Se **dibujan** los objetos.

Hello World con WebGL

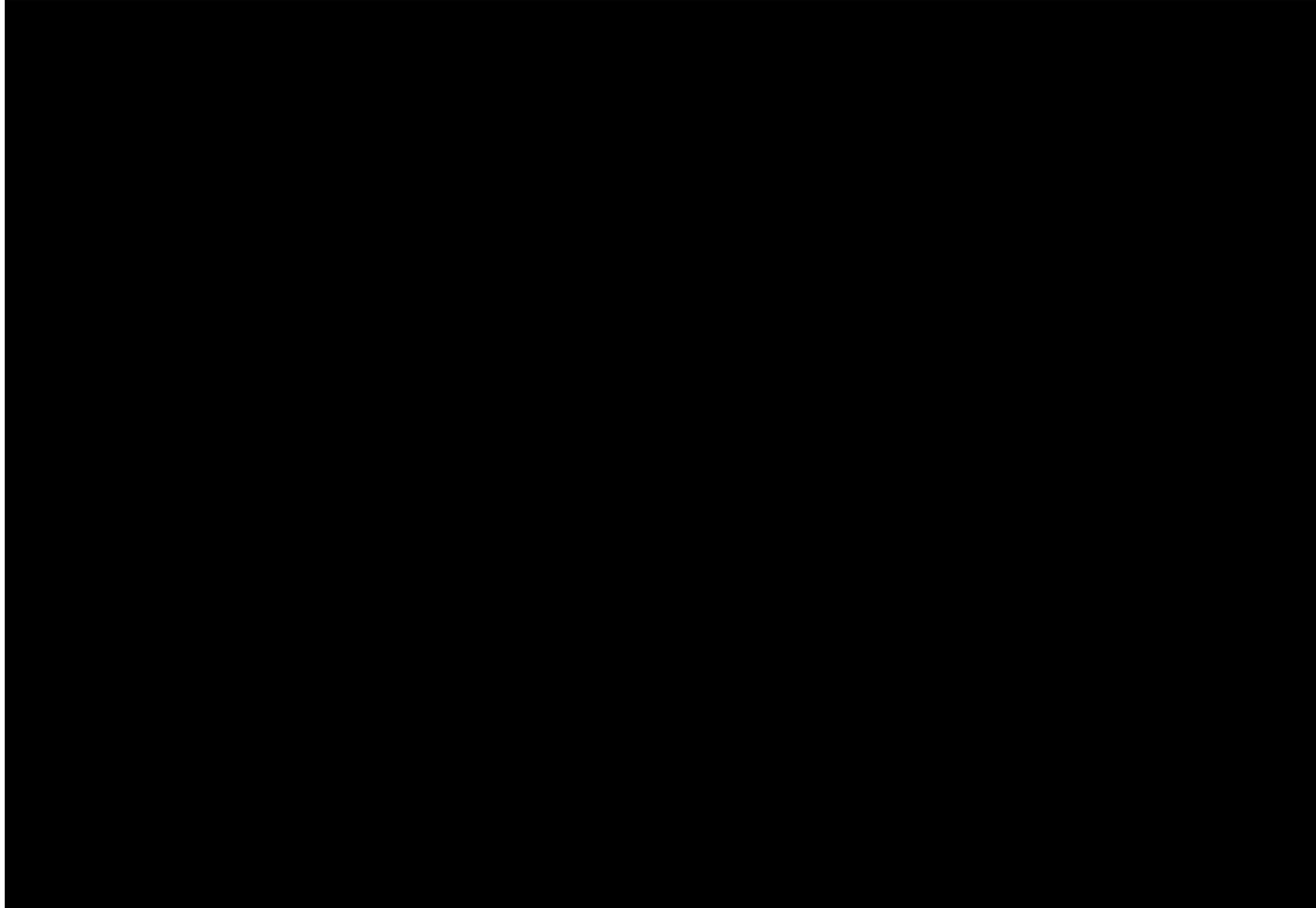
```
gl.viewport(0, 0, gl.drawingBufferWidth, gl.drawingBufferHeight);
gl.clearColor(0,0,0,1); // Set clear color.

gl.enable(gl.DEPTH_TEST); // Enable depth testing
gl.depthFunc(gl.LEQUAL); // Near things obscure far things
gl.enable(gl.CULL_FACE); // BACK by default
gl.cullFace(gl.BACK);

gl.blendFunc(gl.SRC_ALPHA, gl.ONE_MINUS_SRC_ALPHA);

gl.clear(gl.COLOR_BUFFER_BIT|gl.DEPTH_BUFFER_BIT); // Clear
```

¿Cuál es el resultado...?



Well okay

Pongamos a WebGL a trabajar

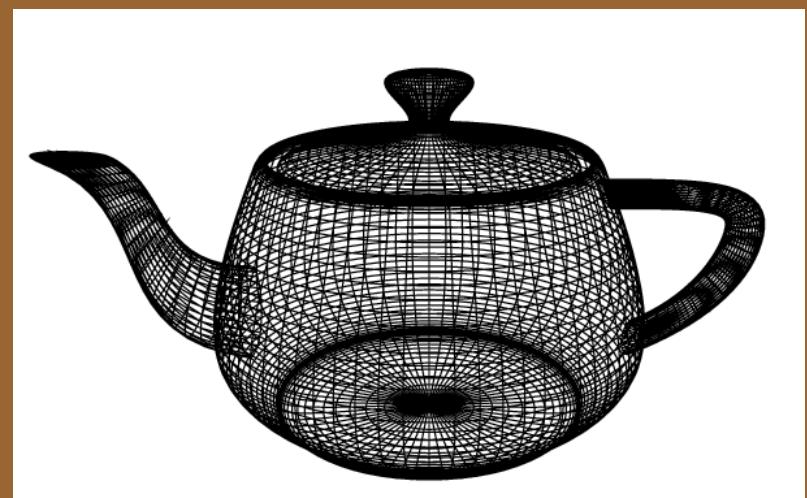


WebGL necesita materiales para trabajar

vértices

colores para cada vértice

Otros: coordenadas de texturas, normales...



¿Cómo le pasamos los datos a WebGL?

Estructuras de datos.

VAO y VBO.

Son simplemente buffers.

VBO

Vertex Buffer Object.

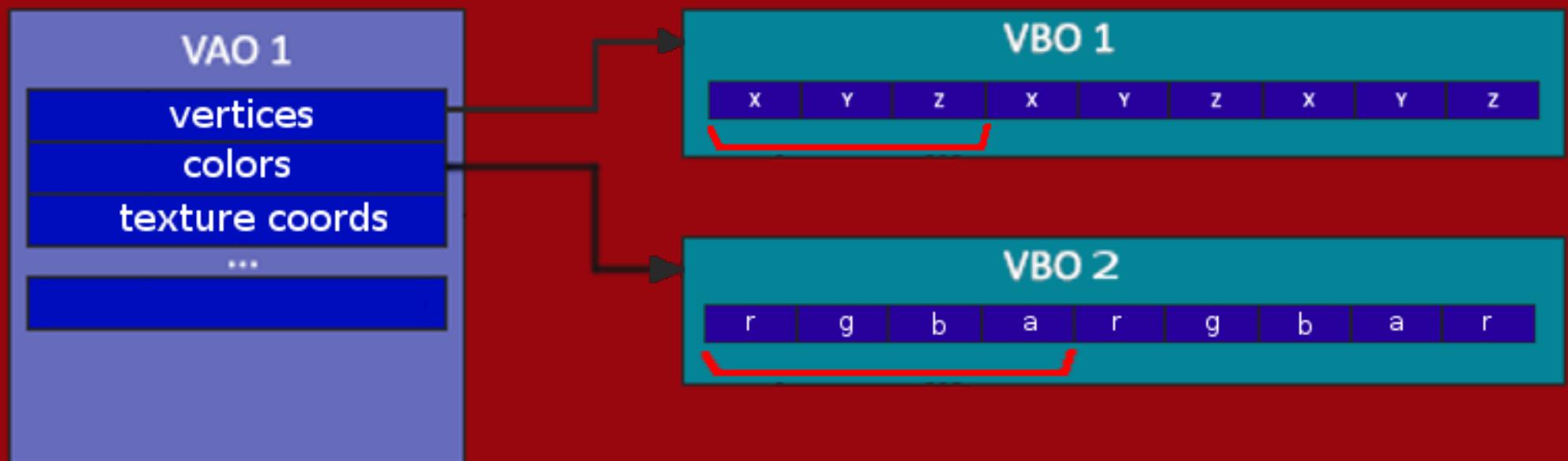
Para **enviar los atributos** de un objeto: vértices, colores, coordenadas de texturas, etc.

x	y	z	x	y	z	x	y	z
---	---	---	---	---	---	---	---	---

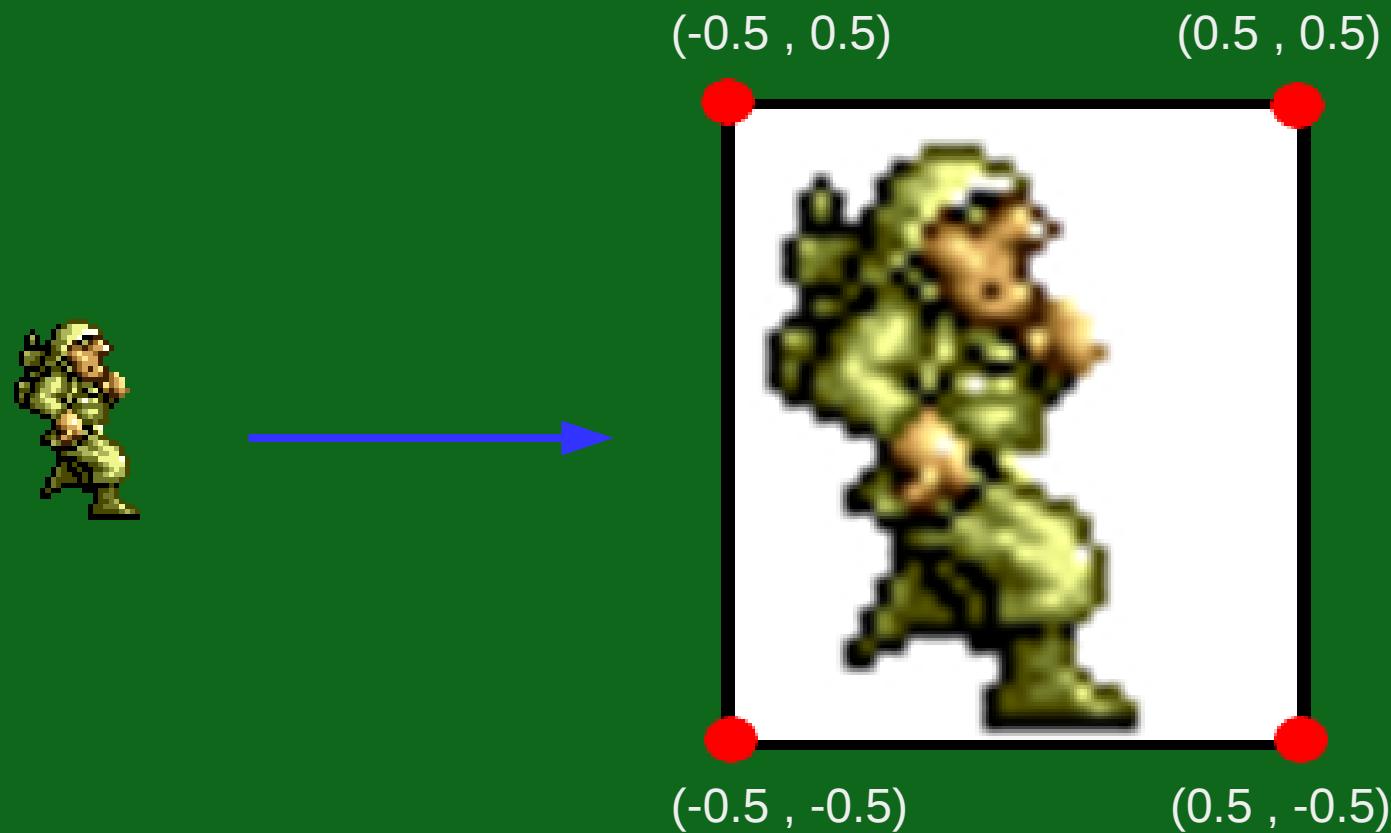
VAO

Vertex Array Object.

Es una colección de VBOs.



Enviar nuestro primer objeto a WebGL



Crear el VAO para nuestro objeto

```
var vao = vao_ext.createVertexArrayOES();
vao_ext.bindVertexArrayOES(vao);

// POSITION
gl.enableVertexAttribArray(0);
var vboPosition = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, vboPosition);
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(mesh.getVerticesData()), gl.STATIC_DRAW);
gl.vertexAttribPointer(0, 4, gl.FLOAT, false, 0, 0);

// TEXTURE
// ...

// NORMALS
// ...

// Finished setting up VAO
vao_ext.bindVertexArrayOES(null);
```

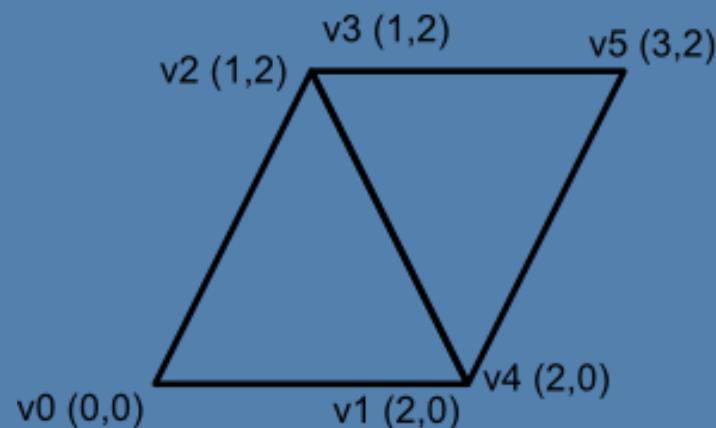


Power Up ! - Optimizando el dibujado



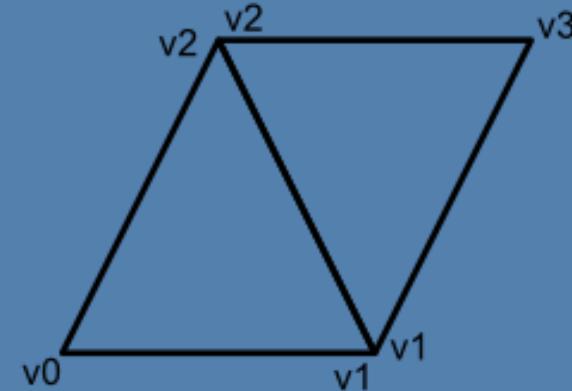
Indexando los vértices

Without indexing



[0,0, 2,0, 1,2, 1,2, 2,0, 3,2]

With indexing

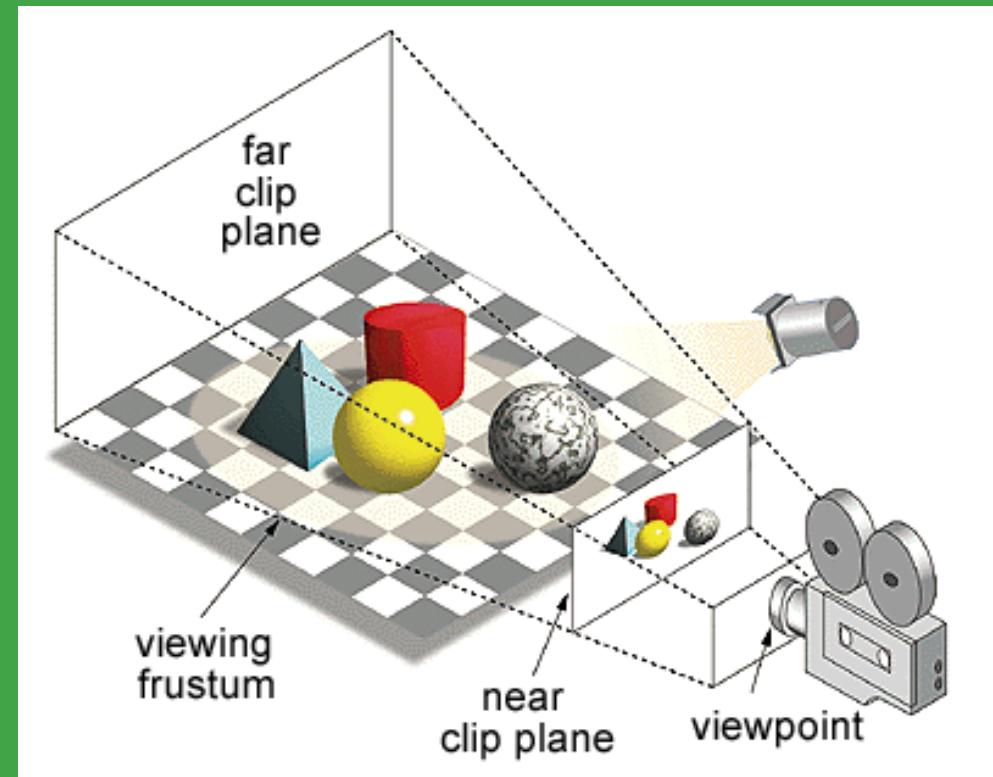
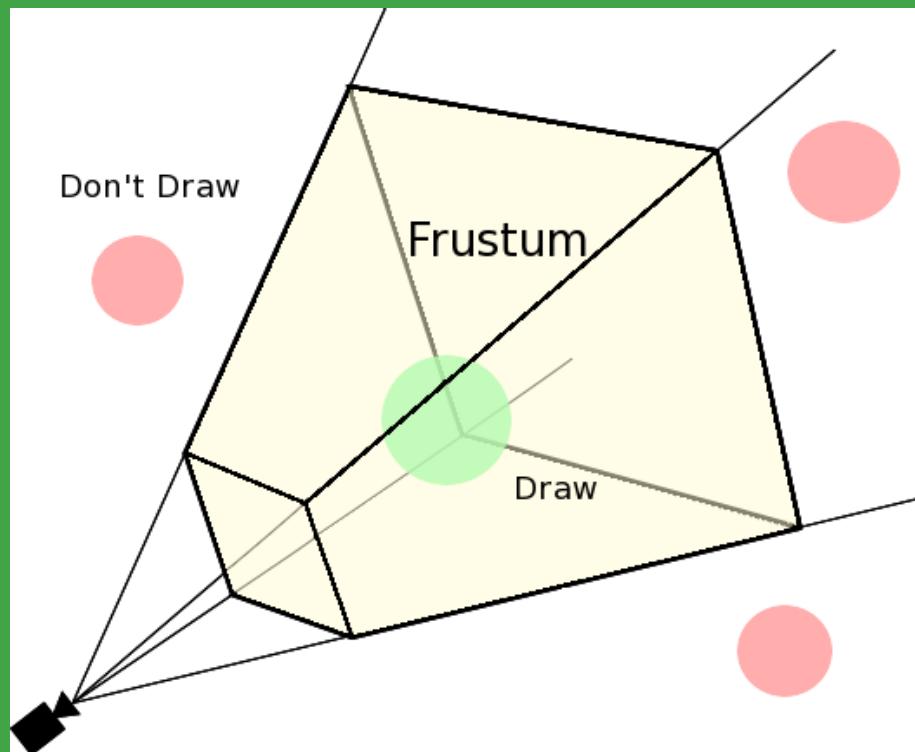


[0,1,2, 2,1,3]
[0,0, 2,0, 1,2, 3,2]

Vertices
reused
twice

Frustum Culling

Todo lo que se situa fuera del campo de visión de la cámara no se dibuja.



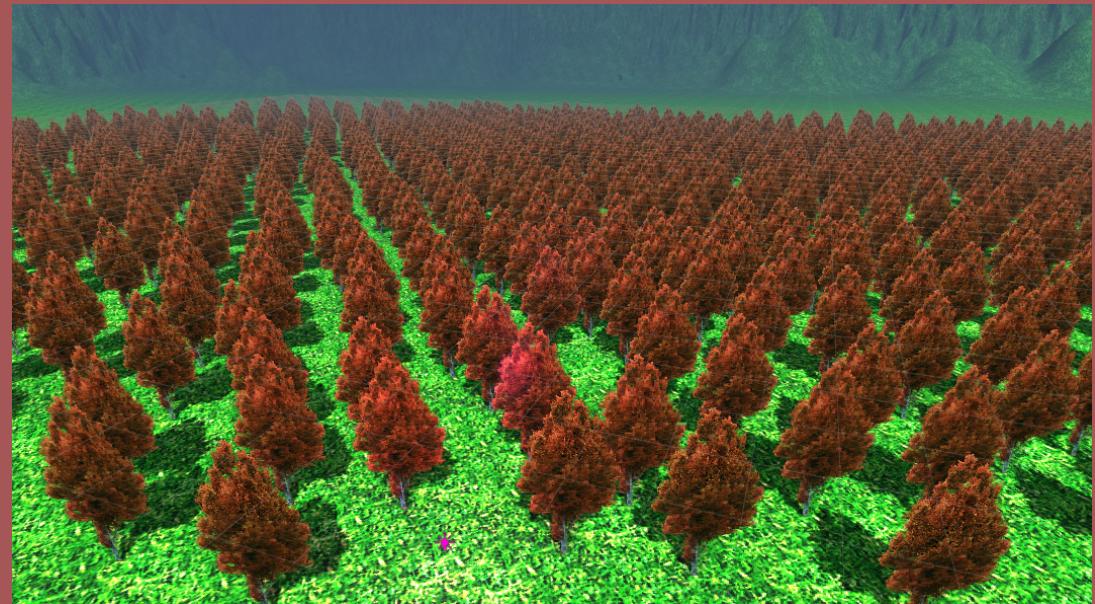
Hay más tipos de culling, como occlusion culling...

Compartir → BIEN Duplicar → MAL

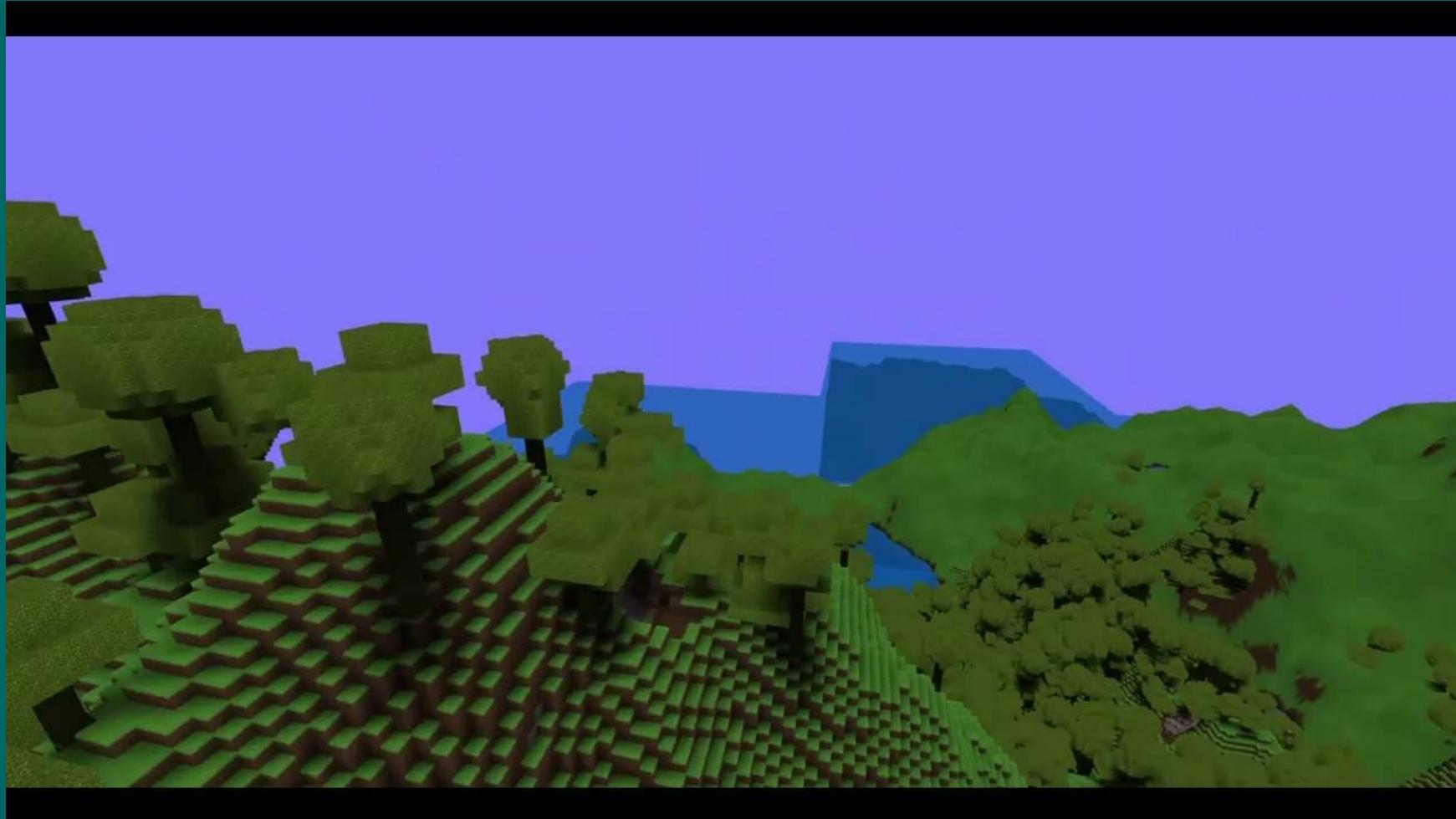
Texture Atlas



Batching



Chunks

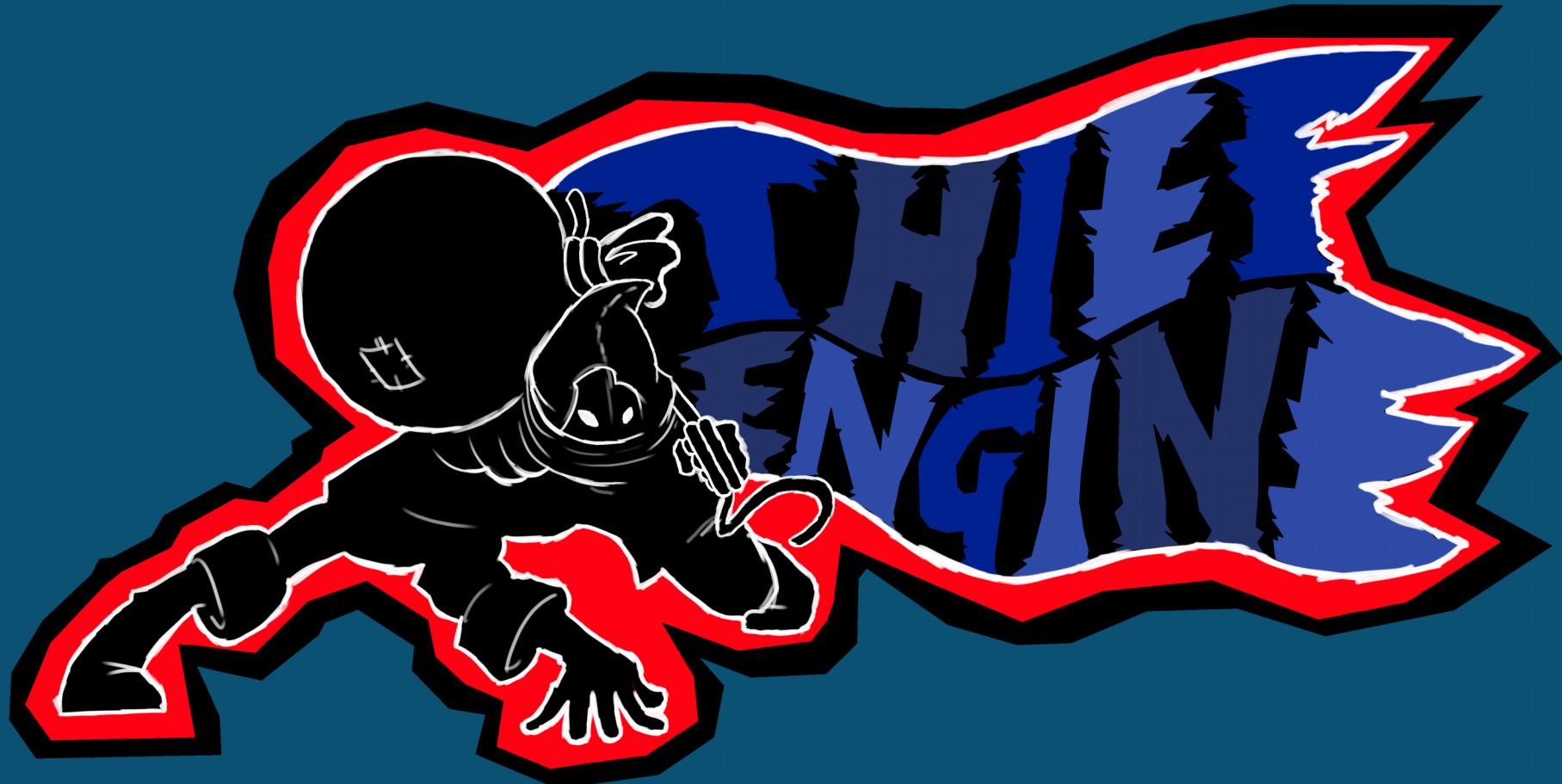


Cada vez que alguien no optimiza su motor



Muere un gatito

Mi libro motor



<https://github.com/AdrianECom/ThiefEngine>



@AdrianensisEC

¡ MUCHAS GRACIAS !

¿Preguntas?

