

Introducción a WebGL

2022-01-31

Que es WebGL

WebGL es un motor de rasterizado (otra forma de decir dibujar). Contiene primitivas como Puntos, Lineas y Triangulos. Basado en el código que se escriba se dibujaran cosas en el espacio.

Características

- Corre en la GPU
- Esta basado en OpenGL (Otra API para generación de elementos 3D)
- En la versión actual, se escribe en un lenguaje conocido como GLSL (GL Shader Language)
- Un aplicativo WebGL es la combinación de 2 Shaders
 - Vertex Shader
 - Fragment Shader

Vertex Shader

- Calcula las posiciones de los vertices de lo que queremos pintar.
- Con las posiciones podemos rasterizar distintas primitivas.

Fragment Shader

Calcula el color de cada pixel mientras se está rasterizando una primitiva.

Las aplicaciones en WebGL se centran en organizar los estados de estas dos funciones (la descripción de la escena) para luego ser dibujadas en el canvas.

Organizar Información para escena + `gl.drawArrays`

Formas de pasar información a los shaders

1. Atributos y buffers

- a. Un buffer es un arreglo de datos binarios, que contienen posiciones, normales, coordenadas de texturas, colores de los vertices, etc.
- b. Los atributos definen como se leen los datos del buffer.

2. Uniforms

- a. Variables globales que pueden acceder los shaders

3. Texturas

- a. Arreglos de acceso aleatorio, tradicionalmente con imágenes de texturas.

4. Varyings

- a. Ofrecen comunicación entre Vertex y Fragment shaders. Cuando el valor cambia en el Vertex Shader, se interpola en el Fragment Shader

A WebGL solo le importan 2 Cosas

- Las coordenadas en el “clip space”
- Los colores de los pixeles

Ejemplo de un Vertex Shader

```
// Vertex shader program

const vsSource = `
    attribute vec4 aVertexPosition;

    uniform mat4 uModelViewMatrix;
    uniform mat4 uProjectionMatrix;

    void main() {
        gl_Position = uProjectionMatrix * uModelViewMatrix * aVertexPosition;
    }
`;
```

Ejemplo de un Fragment Shader

```
const fsSource = `  
    void main() {  
        gl_FragColor = vec4(1.0, 1.0, 1.0, 1.0);  
    }  
`;  
;
```