

Tema 8: Interacción y Animación

José Ribelles

Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I

SIU020 - Síntesis de Imagen y Animación

Contenido

- 1 Introducción
- 2 Selección de objetos 3D
- 3 Animación
 - Timers
 - On/Off
 - Texturas
 - Desplazamiento
 - Mezcla entre dos valores
- 4 Sistemas de partículas

Hoy veremos...

- 1 **Introducción**
- 2 Selección de objetos 3D
- 3 Animación
- 4 Sistemas de partículas

Introducción

¿Es difícil añadir animación utilizando Shaders?

- Implementar animación en un Shader es sencillo ya que es en el Shader donde se procesan los vértices.
- Sólo has de modificar algún atributo a lo largo del tiempo.
- En este tema verás diversos ejemplos.



Hoy veremos...

- 1 Introducción
- 2 Selección de objetos 3D
- 3 Animación
- 4 Sistemas de partículas

Selección de objetos 3D

¿Cómo proceder?

- Cuando el usuario haga la selección, es decir, que por ejemplo sitúe el cursor del ratón sobre un objeto de la escena y entonces pulse el botón izquierdo del ratón.
- Obtén en qué píxel hizo el 'click'.
- Borra el canvas y pinta cada objeto seleccionable de color plano y diferente.
- Averigua el color del píxel seleccionado.
- Sabiendo el color sabrás sobre qué objeto se hizo el 'click'.
- Borra de nuevo el canvas y pinta por completo la escena normal.

Selección de objetos 3D

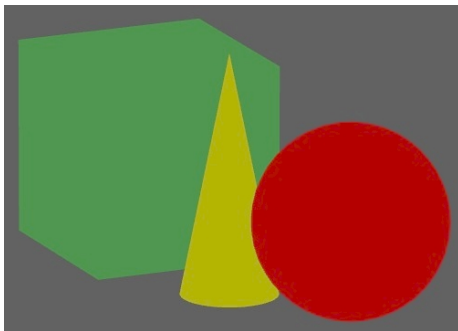
Obtén en qué píxel hizo el 'click'

- En el navegador, ¿dónde está el origen?
- Y para WebGL, ¿dónde está el origen?
- Entonces, ¿qué operación debo hacer?
 - Si, `rect = event.target.getBoundingBoxClientRect();`
 - `x_in_canvas = (event.clientX - rect.left);`
 - `y_in_canvas = (rect.bottom - event.clientY);`

Selección de objetos 3D

Pinta los objetos con colores planos y diferentes

- En primer lugar no olvides:
`gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);`
- Y después pinta algo así:



Selección de objetos 3D

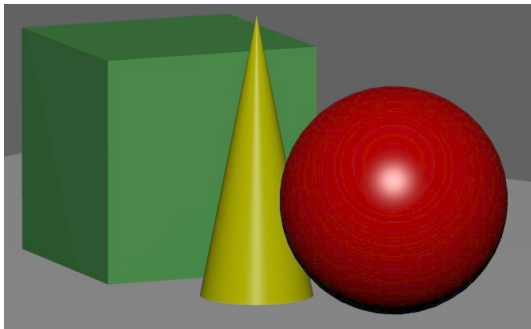
Averigua el color del píxel seleccionado

- Simplemente consulta el color:
 - `var pixels = new Uint8Array(4);`
 - `gl.readPixels(x_in_canvas, y_in_canvas, 1, 1, gl.RGBA, gl.UNSIGNED_BYTE, pixels);`
- El valor del color lo contendrá la variable `pixels`.

Selección de objetos 3D

Borra de nuevo el canvas y pinta por completo la escena normal.

- De nuevo: `gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);`
- Y después pinta la escena completa y normal:



Selección de objetos 3D

Listado 1: Resumen

```
function drawScene() {  
  
    var pixels = new Uint8Array(4);  
  
    setProjection();  
  
    // dibujo la escena con colores planos  
    gl.uniform1i (program.PickingIndex, true);  
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);  
    drawObjects();  
  
    // aqui leo el color del pixel  
    gl.readPixels(x_in_canvas, y_in_canvas, 1, 1, gl.RGBA, gl.UNSIGNED_BYTE, pixels);  
  
    // vuelvo a dibujar la escena usando el modelo de Phong  
    gl.uniform1i (program.PickingIndex, false);  
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);  
    drawObjects();  
  
}
```

Ver ejemplo en el *AulaVirtual*

Selección de objetos 3D usando un FBO

¿Cómo proceder?

- La idea es la misma, dibujar con colores planos...
- ... pero en un FBO no visible al usuario.
- Cuando el usuario selecciona, ya tienes en el FBO dónde mirar el qué.
- De esta manera el usuario selecciona y se consulta sobre lo ya dibujado.

Ver ejemplo en el *AulaVirtual*

Hoy veremos...

- 1 Introducción
- 2 Selección de objetos 3D
- 3 Animación**
 - Timers
 - On/Off
 - Texturas
 - Desplazamiento
 - Mezcla entre dos valores
- 4 Sistemas de partículas

Timers en JavaScript

¿Cómo proceder?

- Utiliza: `myVar = setInterval(drawScene, 40);`
 - Y cada 40ms se ejecutará `drawScene()`.
- Utiliza: `myVar = setTimeout(drawScene, 40);`
 - Y en 40ms se ejecutará `drawScene()` una sola vez.
- En ambos casos, para parar utiliza: `clearInterval(myVar);`

On/Off

¿Cómo proceder?

- Simula un parpadeo de una luz al encenderse.
- O de un tubo fluorescente estropeado...
- En general, variar una propiedad cualquiera entre dos valores dados.
- Utiliza una variable en el *Shader* para indicar cómo se ha dibujar el objeto.

On/Off

Listado 2: On/Off

```
// Por ejemplo, en Javascript,  
// antes de llamar a drawSolid() llama a esta funcion  
function setFlickering(value) {  
  
    if (Math.random() > value) // Math.random devuelve un valor en [0..1]  
        gl.uniform1i (program.ModelIndex, false);  
    else  
        gl.uniform1i (program.ModelIndex, true);  
}
```

Listado 3: On/Off

```
// Por ejemplo,  
// en el fragment shader algo asi  
...  
uniform bool mode;  
...  
void main() {  
    ...  
    fragmentColor = (mode == false) ? vec4( phong(n,L,V), 1.0) * 1.6 : vec4 ( Material.Ka, 1.0);  
}
```

Ver ejemplo en el *AulaVirtual*

Texturas

¿Cómo proceder?

- Modifica las coordenadas de textura.
- Súmales por ejemplo un valor de desplazamiento cada 40ms.



Ver ejemplo en el *AulaVirtual*

Texturas

```
// Por ejemplo, en Javascript,
var Offset = 0.0, Velocity = 0.01;

function updateOffset() {
    Offset += Velocity;
    gl.uniform1f(program.OffsetIndex, Offset);

    requestAnimationFrame(drawScene);
}

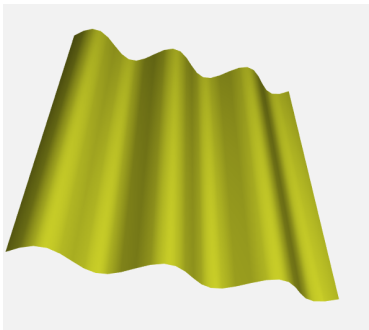
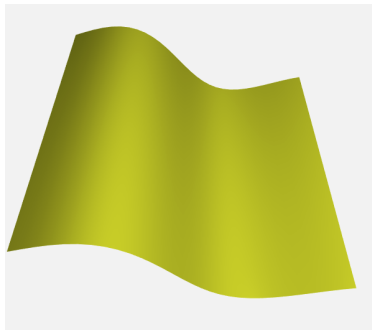
function initWebGL() {
    ...
    setInterval(updateOffset,40);
}
```

```
// Por ejemplo, en el fragment shader algo así
...
uniform float Offset;
...
void main() {
    ...
    float newTexCoords = texCoords + Offset;
    ...
}
```

Desplazamiento

¿Cómo proceder?

- Modifica las coordenadas de los vértices.
- Súmales un desplazamiento en dirección de la normal cada t ms.
- En este ejemplo se usa la función seno sobre un plano teselado.



Desplazamiento

Listado 4: Ejemplo usando la función seno

```
// Vertex shader
...
uniform float Time;
uniform float K;           // wavenumber
uniform float Velocity;   // velocidad
uniform float Amp;        // amplitud

void main() {

    vec4 pos = vec4(VertexPosition, 1.0);
    float u = K * (pos.x - Velocity * Time);
    pos.z    = Amp * sin(u);

    vec3 n = vec3(0.0);
    n.xz    = normalize(vec2(-K*Amp*cos(u), 1.0));

    vec4 ecPosition = modelViewMatrix * pos;
    N = normalize(normalMatrix * n);

    ec = vec3(ecPosition);

    gl_Position = projectionMatrix * ecPosition;
}
```

Ver ejemplo en el *AulaVirtual*

Desplazamiento

Otro ejemplo...

- Modifica las coordenadas de los vértices pero ...
- ... a partir de una función de ruido.



Otros efectos de mezcla

Ejemplos

- Utiliza una variable que varía con el tiempo para obtener el valor definitivo como interpolación lineal de dos valores.
- Por ejemplo, variar la Kd del material..., o asignar dos Kd a un objeto y ...
- En métodos de texturas procedurales, modificar con el tiempo alguno de los parámetros.
- O la componente de transparencia para desvanecer objetos, o para hacerlos aparecer.

Ver ejemplo en el *AulaVirtual*

Hoy veremos...

- 1 Introducción
- 2 Selección de objetos 3D
- 3 Animación
- 4 Sistemas de partículas**

Sistemas de partículas

¿Qué son?

- Técnica de modelado para 'Fuzzy objects'.
 - Objetos con forma no bien-definida.
 - Que además cambia con el tiempo.
 - Objetos dinámicos: fuego, humo, spray, ...
- Nube de partículas que definen un ... volumen!!
- Una partícula: nace, evoluciona y muere.
- Las condiciones iniciales establecen cuándo y cómo, pero no la forma.
- La forma y la apariencia es no determinista.

Sistemas de partículas

Rendering

- Para simplificarlo, se asume que las partículas:
 - No colisionan entre sí.
 - No reflejan luz, pero si emiten.
 - No producen sombras sobre otras partículas.
- Atributos habituales: posición, color, transparencia, velocidad, tamaño y tiempo de vida.
- La posición de una partícula se actualiza en función de un vector inicial de velocidad y efectos como: gravedad, viento, fricción etc.
- Los parámetros de una partícula cambian en función del tiempo, su edad, su velocidad, etc.

Ver ejemplo en el *AulaVirtual*

Ejemplo

Listado 5: Creación de las partículas

```
var nParticles    = 10000;
function initParticleSystem () {

    var particlesData = [];

    for (var i= 0; i < nParticles; i++) {

        // angulos del cono
        var theta = Math.PI / 6.0 * Math.random();
        var phi   = 2.0 * Math.PI * Math.random();

        // direccion
        var x = Math.sin(theta) * Math.cos(phi);
        var y = Math.cos(theta);
        var z = Math.sin(theta) * Math.sin(phi);

        // velocidad
        var alpha = Math.random();
        var velocity = (1.25 * alpha) + (1.50 * (1.0 - alpha));

        particlesData[i * 4 + 0] = x * velocity;
        particlesData[i * 4 + 1] = y * velocity;
        particlesData[i * 4 + 2] = z * velocity;
        particlesData[i * 4 + 3] = i * 0.00075; // instante de nacimiento
    }

    program.idBufferVertices = gl.createBuffer ();
    gl.bindBuffer (gl.ARRAY_BUFFER, program.idBufferVertices);
    gl.bufferData (gl.ARRAY_BUFFER, new Float32Array(particlesData), gl.STATIC_DRAW);
}
```

Ejemplo

Listado 6: Rendering

```
function updateTime(){
    time += 0.01;
    gl.uniform1f (program.TiempoIndex, time);
    requestAnimationFrame(drawScene);
}

function drawParticleSystem() {
    gl.bindBuffer (gl.ARRAY_BUFFER, program.idBufferVertices);
    gl.vertexAttribPointer (program.vertexVelocityAttribute, 3, gl.FLOAT, false, 4*4, 0);
    gl.vertexAttribPointer (program.vertexStartTimeAttribute, 1, gl.FLOAT, false, 4*4, 3*4);

    gl.drawArrays (gl.POINTS, 0, nParticles);
}

function drawObjects() {
    gl.uniformMatrix4fv(program.modelViewMatrixIndex, false, getCameraMatrix());
    drawParticleSystem();
}

function drawScene() {
    setProjection();
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
    drawObjects();
}
```

Ejemplo

Listado 7: El Vertex Shader

```
...
in vec3  VertexVelocity;
in float VertexStartTime;
uniform float Time;
out float alpha;

void main() {
    vec3 pos = vec3(0.0);
    if (Time > VertexStartTime) { // si ha nacido ya
        float t = Time - VertexStartTime;
        if (t < 1.0) { // si aun vive
            pos = pos + (VertexVelocity * t);
            alpha = 1.0 - t;
        }
    }
    vec4 ecPosition= modelViewMatrix * vec4(pos,1.0);
    gl_Position = projectionMatrix * ecPosition;
    gl_PointSize = 3.0;
}
```

Listado 8: El Fragment Shader

```
precision mediump float;
in float alpha;
out vec4  fragmentColor;

void main() {
    fragmentColor = vec4(vec3(0.0), alpha);
}
```

Modifica la trayectoria de la partícula

$$P(t) = P_0 + v_0t + \frac{1}{2}at^2 \quad (1)$$

Listado 9: El Vertex Shader

```
...  
  
void main() {  
    ...  
    if (t < 1.0) { // si aun vive  
        pos = pos + (VertexVelocity * t) + 9.81 * 0.5 * t * t;  
        alpha = 1.0 - t;  
    }  
    ...  
}
```

Ver ejemplo en el *AulaVirtual*

Otro Ejemplo

Listado 10: Cortina de partículas

```
var nParticles    = 10000;
function initParticleSystem() {

    var particlesData = [];

    for (var i= 0; i < nParticles; i++) {

        // velocidad
        var alpha = Math.random();
        var velocity = (0.1 * alpha) + (0.5 * (1.0 - alpha));

        // posicion
        var x = Math.random();
        var y = velocity;
        var z = 0.0;

        particlesData[i * 4 + 0] = x;
        particlesData[i * 4 + 1] = y;
        particlesData[i * 4 + 2] = z;
        particlesData[i * 4 + 3] = i * 0.00075; // instante de nacimiento
    }

    program.idBufferVertices = gl.createBuffer ();
    gl.bindBuffer (gl.ARRAY_BUFFER, program.idBufferVertices);
    gl.bufferData (gl.ARRAY_BUFFER, new Float32Array(particlesData), gl.STATIC_DRAW);
}
```

Ver ejemplo en el *AulaVirtual*

Otro Ejemplo

Listado 11: El Vertex Shader

```
...
in vec3  VertexVelocity;
in float VertexStartTime;

void main() {

    vec3 pos = vec3(0.0);

    if (Time > VertexStartTime) {
        float t = Time - VertexStartTime;
        if (t < 2.0) { // si aun vive
            pos.x = VertexVelocity.x;
            pos.y = VertexVelocity.y * t;
            alpha = 1.0 - t/2.0;
        }
    }
    vec4 ecPosition = modelViewMatrix * vec4(pos,1.0);

    gl_Position      = projectionMatrix * ecPosition;
    gl_PointSize     = 6.0;
}
```

Por último

Añade una textura: Ver ejemplo en el *AulaVirtual*