SIMULACIÓN DEL FUEGO

Joel Oswaldo Gallegos Guillén

Resumen

Se realiza la simulación del fuego, para ello se utiliza un sistema de partículas, en donde se puede ir agregando y eliminando partículas. El sistema de partículas se utiliza por la necesidad de actualizar los datos, por ejemplo, cuando se emite el fuego, una partícula comienza con una determinada posición, y luego va a ir cambiando, así como también se le asocia un tiempo de vida, etc. y es por ello que necesitamos saber el cambio que tiene cada partícula en sus frame anterior, para poder generar su frame siguiente, es por ello que utilizamos una función de opengl, que nos devuelve dicha información

Como se menciono, también se utilizan algunos conceptos de como se da el proceso de generación del fuego, con la finalidad de intentar dar un efecto más realista.

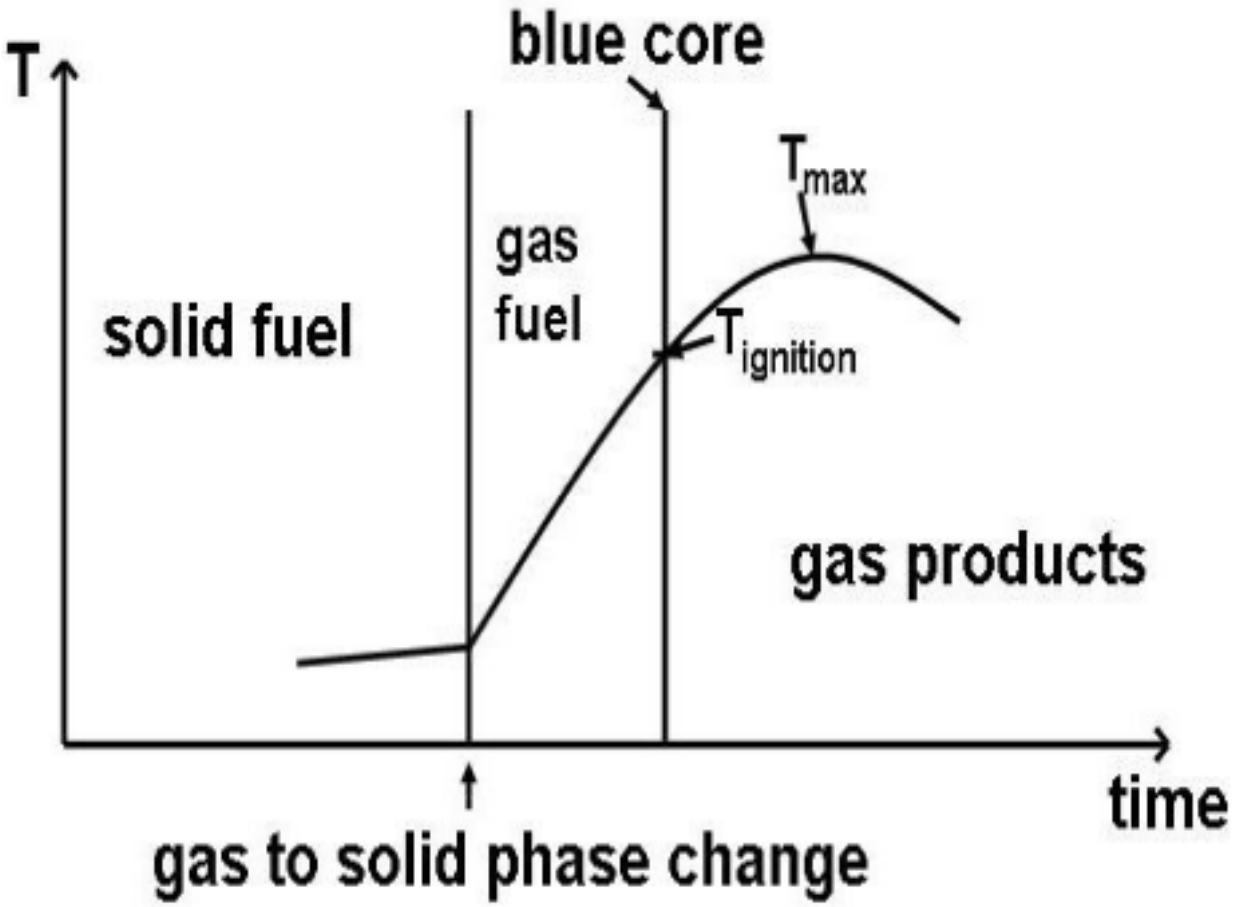
CONCEPTOS: El Fuego

Los principales conceptos que se debe tener es que el fuego esta compuesta por miles de partículas, en el caso del fuego una particula de fuego se genera como combustible, hasta el momento de su fin de vida.

En cada partícula del fuego se debe de considerar:

- **Posición inicial.**
- **Velocidad y dirección Inicial.**
- **Tamaño Inicial.**
- **Color Inicial.**
- **Velocidad de combustión**

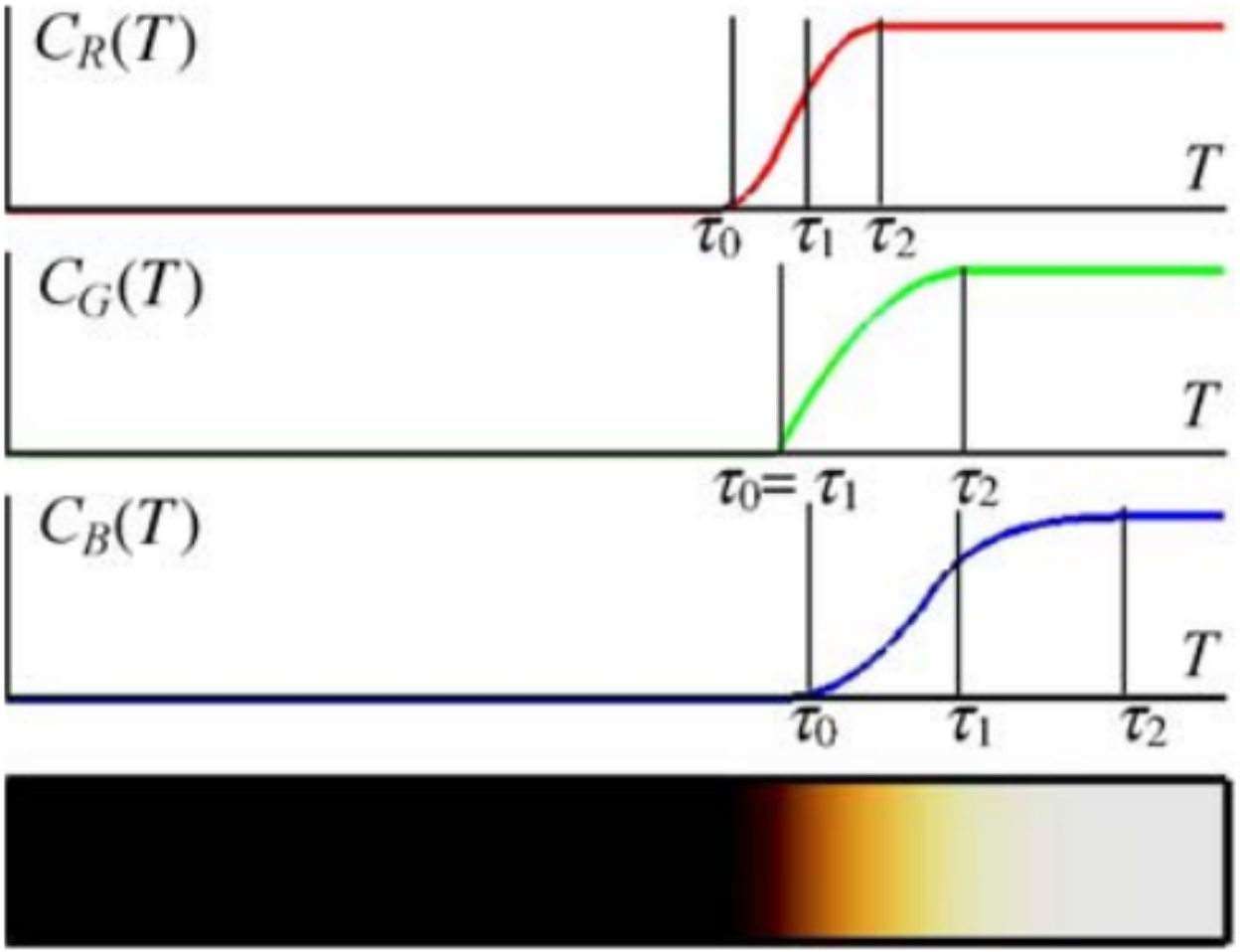
CONCEPTOS: El Fuego



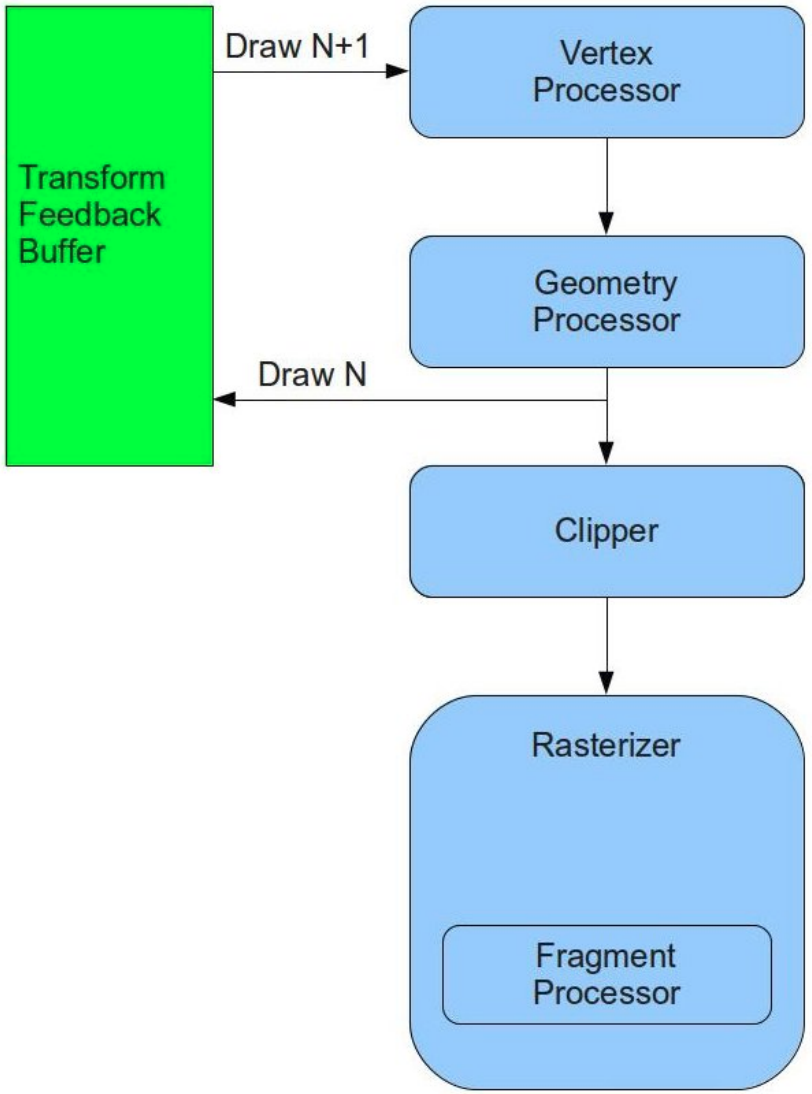
Ecuacioón de Navier-Stokes

$$\rho\left[\frac{du}{dt} + u.\nabla u\right] = -\nabla p + \mu\nabla^2 u + f \quad [1]$$

Colores



OPENGL (GEOMETRY SHADER - TRANSFORM FEEDBACK)



Implementación: Geometry Shader

Vertex Shader

```
out vec3 vVelocidad;
out vec3 vDireccion;
out float vEdad;
out float vMaxEdad;
out float vLimite;
```

Geometry Shader

```
layout(points) in;
layout(points, max_vertices = 2) out;

in vec3 vVelocidad[];
in vec3 vDireccion[];
in float vEdad[];
in float vMaxEdad[];
in float vLimite[];
```

Implementación: Transform Feedback

Vertex Shader

```
out vec3 posicion;
out vec3 direccion;
out vec3 velocidad;
out float edad;
out float maxEdad;
out float limite;
```

Geometry Shader

```
velocidad = vVelocidad[0];
direccion = dirNueva;
edad = vEdad[0] + deltaTime;
maxEdad = vMaxEdad[0];
limite = miLimite-limitePro;

EmitVertex();
EndPrimitive();
```

Implementación: Transform Feedback

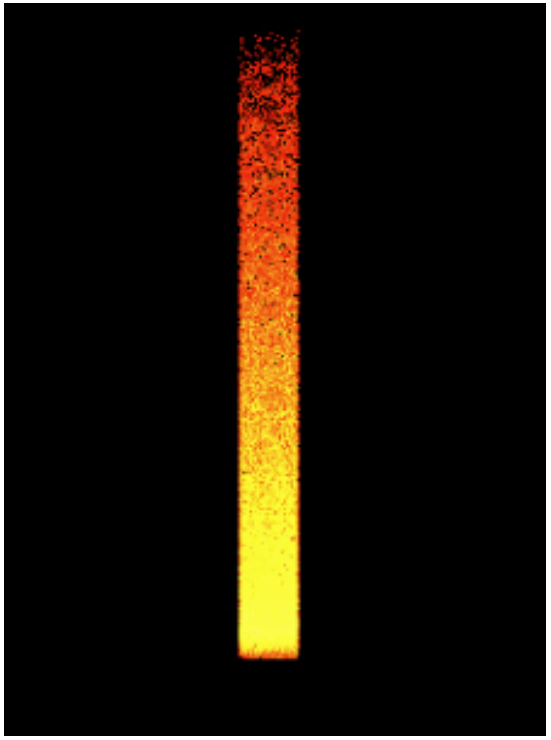
```
const GLchar* feedbackVaryings[] = {"posicion", "direccion",
"velocidad", "edad", "maxEdad", "limite"};
```

```
glTransformFeedbackVaryings(program, 6, feedbackVaryings, GL_INTERLEAVED_ATTRIBS);
```

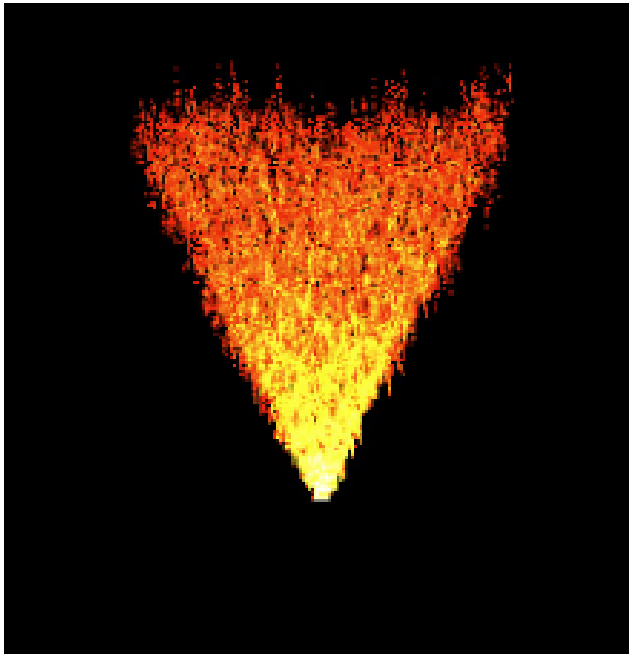
```
glLinkProgram(program);
```

Resultados

Caso inicial, solo modifica posiciones.

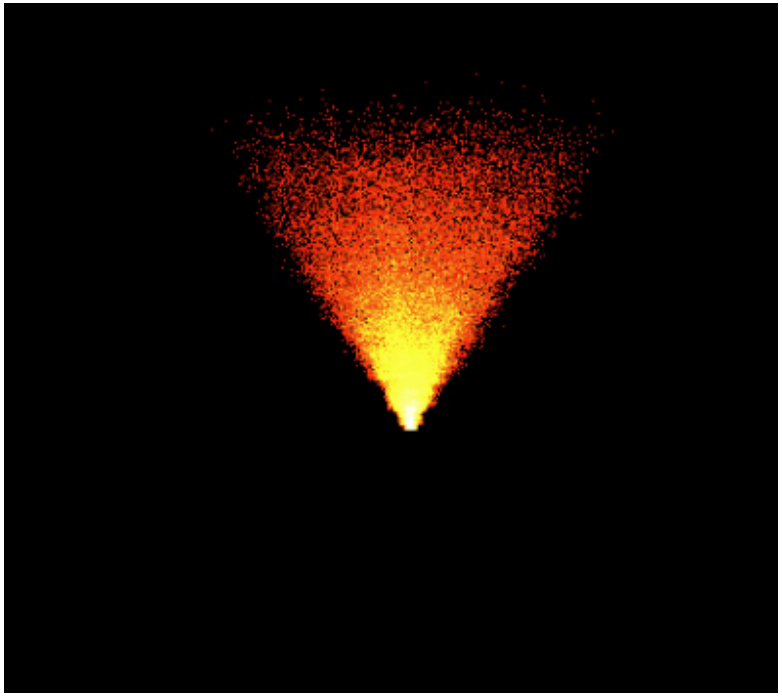


Se utiliza dirección

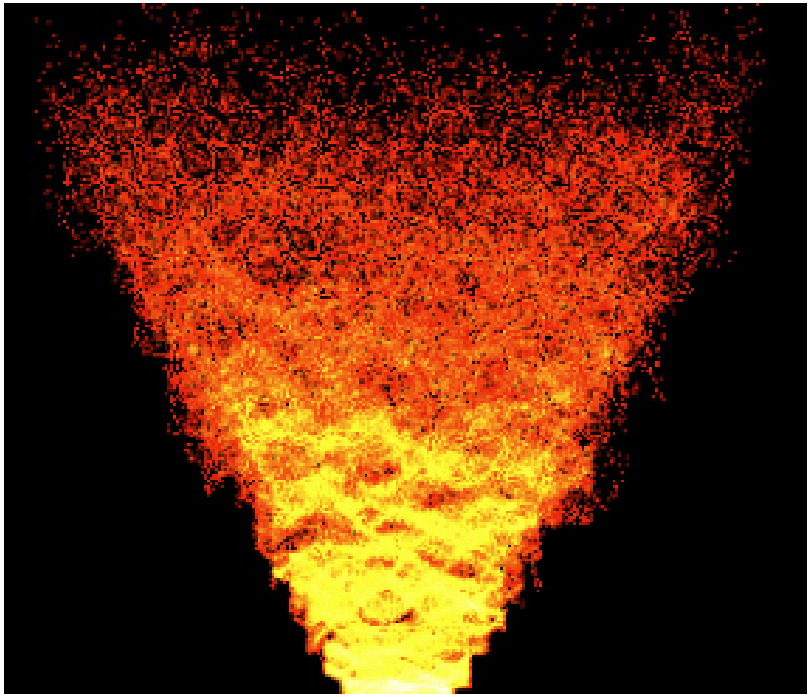


Resultados

Se modifica tiempo de vida máximo

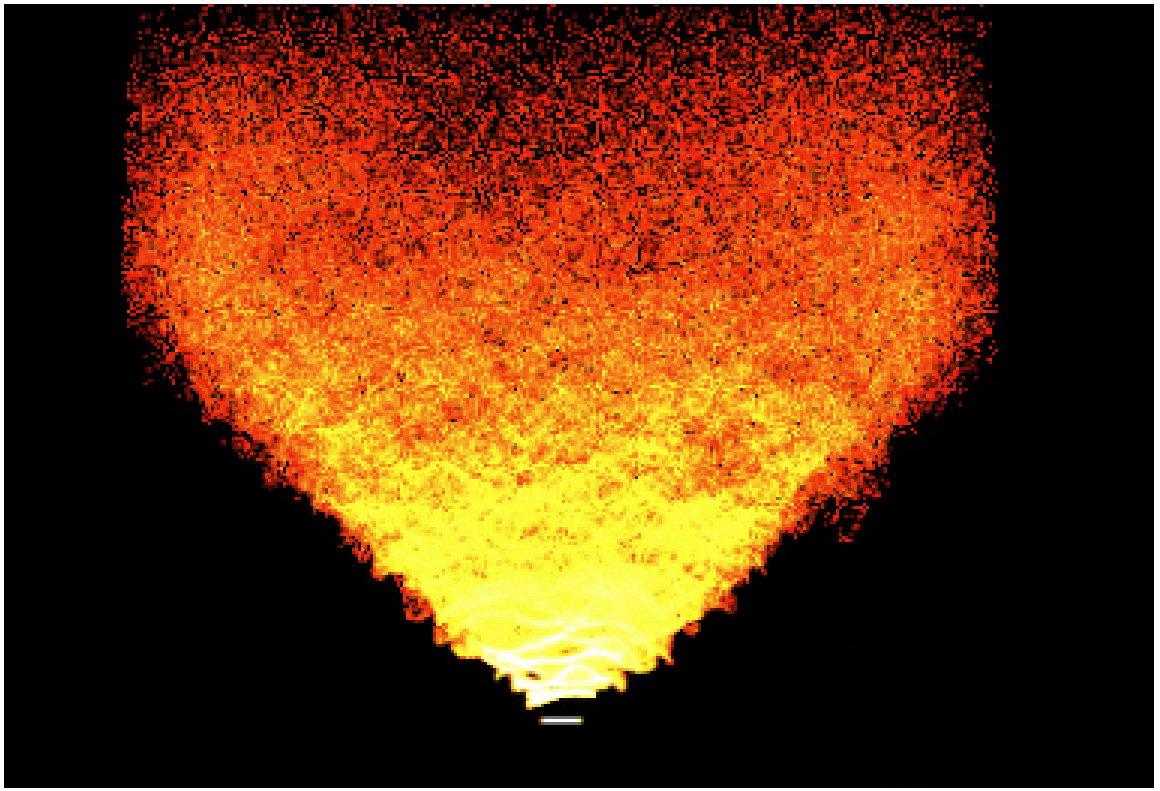


Se utiliza un ruido para dar el efecto del movimiento.



Resultados

Intento de efecto de agrupación.



Videos

https://youtu.be/PY_YivM5z3o

https://youtu.be/kf7K7e2bEA0

https://youtu.be/vFj-MZe6S9A

https://youtu.be/c0BDfC_Dkww

https://youtu.be/JtQHoiOOzEk