

Alumno: Rubén Garrido Valdepeñas  
Asignatura: Efectos visuales y sonoros  
Curso: 2022-2023  
Profesor: Rafael Pérez Vidal

## **Pec 1: Generación de efectos en tiempo real de procesamiento**

### Índice

#### [1. Stars](#)

[1.1. Inicialización](#)

[1.2. Update](#)

[1.3. Render](#)

[1.4. Close](#)

#### [2. RotoZoom](#)

[2.1. Inicialización](#)

[2.2. Update](#)

[2.3. Render](#)

[2.4. Close](#)

#### [4. Túnel](#)

[4.1. Inicialización](#)

[4.2. Update](#)

[4.3. Render](#)

[4.4. Close](#)

#### [5. Plasma](#)

[5.1. Inicialización](#)

[5.2. Update](#)

[5.3. Render](#)

[5.4. Close](#)

#### [6. Transiciones](#)

[6.1. Transición Fade-in Fade-out](#)

[6.2. Transición Puertas](#)

#### [7. Mixer](#)

# 1. Stars

## 1.1. Inicialización

En la inicialización primero reservo el espacio que necesito para todas las estrellas. Seguidamente por cada estrella le doy una dirección entre -1 y 1. Este valor lo utilizaré más tarde para modificar el eje x de la posición de la estrella. Para la posición de la estrella tengo un struct para guardar el eje X e Y. Finalmente he querido añadir una cola de píxeles para dar la sensación de movimiento..

## 1.2. Update

Aquí toca actualizar la posición de todas las estrellas para simular el movimiento. En mi caso he querido simular que las estrellas están cayendo. El método consiste en actualizar el eje Y para simular que la estrella está cayendo y el eje X con la dirección. Se pueden dar el caso de que la estrella se salga por uno de los extremos laterales o por el inferior por lo que cuando pasa esto le indico una posición nueva. Finalmente queda actualizar la cola donde cada valor se mueve a la siguiente posición y la última se pierde.

## 1.3. Render

Cada vez que se llama a este paso se tiene que llenar toda la pantalla de negro. De esta manera se podrá actualizar las posiciones de las estrellas. En este apartado he añadido los planos mostrados en el ejemplo dado que me ha gustado como quedaba en pantalla el tener varios colores. Finalmente solo queda mostrar por pantalla los píxeles de la estrella. Para esto cojo la posición inicial de los píxeles de la pantalla y le añado el desplazamiento donde se encuentra la estrella. En este caso la posición x se multiplicaría por el tamaño de píxel y la Y por el pitch. Ya solo queda poner en la posición el color.

## 1.4. Close

Al terminar el efecto he añadido un método que limpie el puntero utilizado para almacenar las estrellas.

## 2. RotoZoom

### 2.1. Inicialización

Para la inicialización sólo es necesario cargar la textura en la surface.

### 2.2. Update

En esta etapa se tienen que calcular varios puntos en el espacio para representar la posición de la textura y la profundidad.

Para esto necesito una variable que me defina la distancia, en mi caso he utilizado el tamaño de la textura y el currentTime para que el valor vaya cambiando.

Seguidamente hace falta un valor para indicar la rotación a las posiciones que se van a calcular después. En este caso solo uso el currentTime. Si este valor fuera positivo el efecto de giro sería hacia la derecha. En mi caso lo he dejado en negativo.

Finalmente queda calcular los puntos intermedios de la textura. Para esto se necesita un conjunto de tres posiciones.

Estos valores utilizan como base el tamaño de la textura y se multiplican por el ángulo (para indicar la dirección) y por el valor de profundidad.

### 2.3. Render

Con los valores generados en el Update se van a generar los deltas que se usarán en el pintado de todas las líneas. Estos deltas son valores que se irán añadiendo en cada posición de la pantalla que se va a pintar. Esto se hace incrementando el valor de los puntos con los delta en cada vuelta del pintado de pantalla (en líneas y filas).

En cada iteración de la fila se actualiza el valor de la textura con los primeros deltas (estos valores se deben restaurar al final del pintado de cada línea).

### 2.4. Close

En este apartado no es necesario hacer nada ya que no tenemos ningún puntero que liberar de memoria. No cuento la surface porque será reemplazada por el siguiente efecto.

## 3. Distorsión

### 3.1. Inicialización

En la inicialización reservo el espacio en memoria de los dos buffers y cargo la textura. Seguidamente se hace un precálculo del efecto en los dos buffers.

### 3.2. Update

En este paso tengo que calcular los deltas que se van a usar para acceder a una posición concreta de los buffers. Este valor debe ir desde 0 hasta el tamaño del buffer. Estos deltas se calculan a partir de una posición fija sumado a una parte variable. En este caso oscilante mediante el uso del currentTime como ángulo en senos y cosenos.

### 3.3. Render

Utilizando los valores calculados se accede a cada buffer y se añade el valor a la posición de la textura que queremos mostrar por pantalla. hay que vigilar que no estemos saliendo fuera del rango de la pantalla.

### 3.4. Close

Aquí libero de memoria los dos buffers.

## 4. Túnel

### 4.1. Inicialización

En esta primera parte el primer paso es reservar la memoria para el buffer del túnel. En este caso casi todos los cálculos pueden ser realizados de antemano.

Para este efecto necesitamos dos valores, uno para guardar el ángulo de la posición de la textura y otro para indicar el punto en el espacio de este ángulo.

Para el punto en el espacio empiezo con unos valores estáticos de la pantalla. En mi caso ha sido el centro de la pantalla. Seguidamente hay que calcular una posición con los valores normalizados. Este valor ya se puede guardar en el buffer, ahora solo falta obtener el ángulo para esa posición utilizando esos valores.

### 4.2. Update

No hace falta hacer nada en este apartado porque se han hecho todos los precálculos en la inicialización.

## 4.3. Render

Para mostrar el efecto por pantalla se utilizan los valores del precálculo con un factor de corrección. En mi caso el `currentTime` y el valor de los FPS máximos.

## 4.4. Close

Aquí libero memoria del buffer del túnel.

# 5. Plasma

## 5.1. Inicialización

En este efecto se utilizan dos buffers. Cada uno tendrá una generación diferente.

## 5.2. Update

Aquí se deben hacer dos cosas. Generar una paleta de colores y obtener el punto inicial del efecto.

Para generar la paleta de colores he visto que se puede hacer de dos formas. Por un lado se puede “crear a mano” en un fichero a parte. De esta manera no hay que realizar ningún cálculo y solo cargar los valores en memoria. La siguiente forma es crear la paleta en tiempo real en cada frame. Para eso necesitamos definir el RGB de cada posición de la paleta.

Para generar cada color se tiene que tener en cuenta que el valor debe ir entre 0 y 256. Como valor base uso 256 puesto que es el tope de valor que puede tener nuestros colores. Ahora hay que multiplicarlo por un valor oscilante entre 0 y 1. Hay que tener en cuenta que el seno, por ejemplo, tiene un rango de -1 a 1 por lo que la parte negativa no sirve. Por esto los valores se van a multiplicar por  $\pi$  de esta manera el valor no será inferior a 0. Ahora le añado el `currentTime` para añadir algo de variación en el tiempo. Esto no funcionaba bien y el cambio iba muy rápido por lo que he añadido un divisor grande.

Finalmente solo queda generar una posición para acceder a los buffers. En mi caso lo he cogido de forma aleatoria tomando el `currentTime` como variación.

## 5.3. Render

En el render por cada posición de la surface se usa la combinación de los valores de cada buffer en una posición determinada (los valores aleatorios obtenidos en el Update). Estos índices se van incrementando mientras se va recorriendo la surface.

## 5.4. Close

Aquí solo he tenido que liberar los dos buffers.

# 6. Transiciones

## 6.1. Transición Fade-in Fade-out

En esta transición la pantalla se va oscureciendo poco a poco hasta quedar completamente en negro. Seguidamente vuelve al estado inicial cuando el nuevo efecto se ha cargado. Para hacer este efecto tengo una surface en la que dependiendo de si estoy en un ciclo de sombreado o clareado voy modificando la transparencia.

## 6.2. Transición Puertas

Con esta transición quiero simular las puertas correderas automáticas que hay en las entradas de los establecimientos.

Para hacer este efecto utilizo un surface extra al que añado al surface principal mediante blend.

El efecto consiste en la inclusión de dos rectángulos en la pantalla. Uno en cada extremo. Durante el desarrollo del efecto el rectángulo situado a la izquierda aumenta su ancho (hasta llegar a la mitad de la pantalla). Por otro lado, el rectángulo derecho, a parte de aumentar su ancho, disminuye su posición X. Así da la sensación de que se mueve a la izquierda.

En el momento en el que ha llegado al centro se pasa a la fase de apertura. En este punto el efecto ya se ha cambiado y tenemos el nuevo. Ahora solo toca invertir decreciendo el tamaño de los rectángulos.

## 7. Mixer

No me ha dado tiempo de implementar el Mixer de audio correctamente. He cogido el ejemplo de clase y lo he utilizado al menos para reproducir audio. No he conseguido que cuadre el inicio de transición con el cambio de efecto en el momento en el que sale el tope de heart beats.