

Arquitecturas Paralelas

Animaciones con SSE

Versión 2.0

25 de febrero de 2019

1. Objetivo

El objetivo de esta práctica es crear una animación utilizando las instrucciones, las aritméticas y los tipos de datos que proporciona la tecnología SIMD de Intel (MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, AES y AVX).

En la serie de juegos Angry Birds versión PC, en su directorio de instalación se pueden encontrar un par de carpetas interesantes. En la carpeta data/imagenes están los personajes y escenarios de las animaciones. En la carpeta data/audio se localizan la música y los efectos de las animaciones.

Los ficheros INGAME_BIRDS_1.png y THEME_EASTER_SKY_1.png, figuras 1 y 2 respectivamente, son dos ejemplos de Angry Birds Seasons. En el primero se pueden ver algunos de los personajes y en el segundo uno de sus escenarios.



Figura 1: Personajes



Figura 2: Fondo

Queremos aquí agradecer a Rovio, que tal como veis en el correo electrónico de la figura 1, nos da todas las facilidades para utilizar sus personajes en nuestras clases de AP.

La práctica se realizará en los siguientes pasos:

Katariina, Feb 19 03:37:

Hello Matias,

and thank you for contacting us!

We are happy to hear that you would like to use our games as examples to your students. You can gladly use the games as examples and inspirations for your students, as long as there is no profit gained. We are always happy have new people learning about the great world of animation and coding!

We wish you a good course!

Best regards and greetings from the Angry Birds!

Katariina

Rovio Support Team

More about Rovio

Newsletter - <http://rovio.rovionews>

Facebook - <http://www.facebook.com/RovioEntertainment>

Twitter - <http://twitter.com/Rovio>

www.rovio.com

Figura 3: We are always happy have new people learning about the great world of animation and coding!

1. Mostrar las dos imágenes en pantalla
2. Utilizando el canal de alfa de las imágenes png, superponer uno de los personajes sobre el fondo.
3. Utilizando la tecnología SSE2, superponer uno de los personajes sobre el fondo.

2. Mostrando las dos imágenes

Las dos imágenes proporcionadas son imágenes tipo PNG. Dichas imágenes se caracterizan por ser un formato sin pérdida de calidad con alta compresión, y además admite un canal alfa. Con este cuarto canal un píxel se almacena en 4 Bytes según el formato de la figura 2:

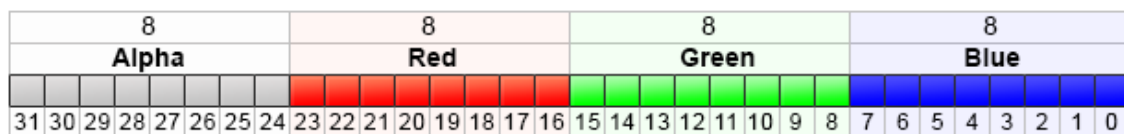


Figura 4: Formato ARGB.

Los 32 bits de un píxel se distribuyen entre 4 canales, un canal alfa de 8 bits y los ya conocidos canales RGB de también 8 bits. Este cuarto canal alfa añade información sobre transparencia, es decir, "Se puede establecer el valor de 0 (el color del píxel puesto encima no cambia en absoluto la visualización del píxel del fondo) hasta 255 (el color se pone encima íntegramente y cubre totalmente el color del píxel del fondo). El color de cada píxel para la imagen resultante se calcula según la siguiente fórmula: el aporte de la imagen de fondo es $(1 - \text{alfa}/255) * 100\%$ y el aporte de la imagen de primer plano es $(\text{alfa}/255) * 100\%$. Es decir, cuanto menor sea el valor de la composición alfa, más transparente será el color." de <https://www.mql5.com/es/docs/convert/colortoargb>

3. Animación sin SSE2

Vamos a mostrar sobre el fondo THEME_EASTER_SKY_1.png al personaje Red que se puede encontrar en INGAME_BIRDS_1.png a partir de la fila 908, columna 780, con alto 48 filas y de ancho 56 columnas.

El canal alfa pondera la superposición del personaje sobre el fondo. En el fichero INGAME_BIRDS_1.png el canal alfa es como una máscara, pero no es una máscara binaria como en la práctica 3. Para algunos píxeles la máscara vale 0, en otros 255, pero también hay píxeles con valores intermedios. Por lo tanto, cada píxel de la animación se compone del píxel del personaje ponderado por $\text{alfa}/255$ más el píxel del fondo ponderado por $(1 - \text{alfa}/255)$.



Figura 5: Detalle de la animación sin SSE2 utilizando el canal alfa.

4. Animación con SSE2

Vamos a hacer inicialmente la animación con SSE2 con una máscara binaria, de esta forma será más sencillo el algoritmo. Es decir, si el canal alfa de un pixel del personaje es 0, entonces en la animación aparece el pixel del fondo o escenario; si el canal alfa de un pixel del personaje es 255, entonces en la animación aparece el pixel del personaje.

Para ello tenéis la ayuda de la figura 6:

Registros SSE2																MEMORIA	
byte 15	byte 14	byte 13	byte 12	byte 11	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0		
pixel 3				pixel 2				pixel 1				pixel 0				B	
Ap=00	Rp	Gp	Bp	Ap=00	Rp	Gp	Bp	Ap=1F	Rp	Gp	Bp	Ap=FF	Rp	Gp	Bp	R1=4 pixels personaje	G pixel 0
FF	00	00	00	FF	00	00	00	FF	00	00	00	FF	00	00	00	R2=mascara para canal alfa	R pixel 1
Ap=00	00	00	00	Ap=00	00	00	00	Ap=1F	00	00	00	Ap=FF	00	00	00	R3=R1&R2,solo canal alfa	A pixel 2
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	R4= todo ceros	R pixel 3
FF	FF	FF	FF	FF	FF	FF	FF	00	00	00	00	00	00	00	00	R5=(R3 == R4) a nivel INT	B
Af	Rf	Gf	Bf	Af	Rf	Gf	Bf	Af	Rf	Gf	Bf	Af	Rf	Gf	Bf	R6= 4 pixels fondo	G
00	00	00	00	00	00	00	00	Ap=1F	Rp	Gp	Bp	Ap=FF	Rp	Gp	Bp	R7= not R5 and R1	R
Af	Rf	Gf	Bf	Af	Rf	Gf	Bf	00	00	00	00	00	00	00	00	R8= R5 and R6	A
Af	Rf	Gf	Bf	Af	Rf	Gf	Bf	Ap=1F	Rp	Gp	Bp	Ap=FF	Rp	Gp	Bp	R9 =R7 or R8	

Figura 6: Algoritmo de la animación con SSE2.

Es importante utilizar las funciones `_mm_loadu_si128` y `_mm_storeu_si128` para cargar un registro con datos de memoria y para almacenar un registro en memoria respectivamente. Se hace distinto a la practica 4, pues no tenemos garantizado que los datos de la imagen estén alineados en memoria a nivel de 16 bytes.

El resultado puede verse en la imagen 7. El resultado no es igual al del apartado anterior, porque se ha hecho la simplificación de la máscara binaria.

Las funciones a utilizar son:

- `_mm_loadu_si128`
- `_mm_set1_epi32`



Figura 7: Detalle de la animación con SSE2 utilizando mascara binaria.

- `_mm_cmpeq_epi32`
- `_mm_andnot_si128`
- `_mm_and_si128`
- `_mm_or_si128`
- `_mm_storeu_si128`

5. Varios

El tipo de datos `__m128i` y las funciones SSE suelen aparecer como errores en el editor del netbeans. Para solucionar esto, podemos hacer botón derecho del ratón sobre el nombre del proyecto, ir a Code Assistance, y después Reparse Project. A partir de este momento, ya no se verán como errores en la pantalla del editor.