



Módulo práctico 2: Sistemas de partículas mediante shaders de cómputo

Procesadores Gráficos y Aplicaciones en Tiempo Real

Profesores: Alberto Sánchez, Óscar D. Robles y Juan P. Brito

**Máster en
Informática Gráfica,
Juegos y Realidad Virtual**

Escuela Técnica Superior
de Ingeniería Informática



Universidad
Rey Juan Carlos

Sistemas de partículas mediante shaders de cómputo

- Objetivo: Familiarización con los shaders de cómputo en un supuesto práctico basado **en sistemas de partículas**.



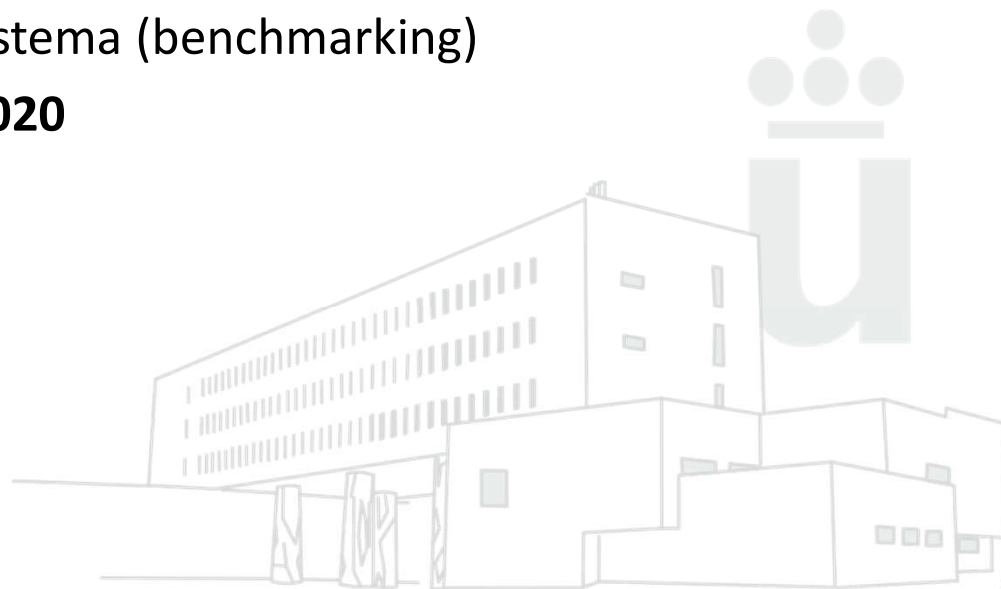
Vídeo

- Star Trek II: The Wrath of Khan



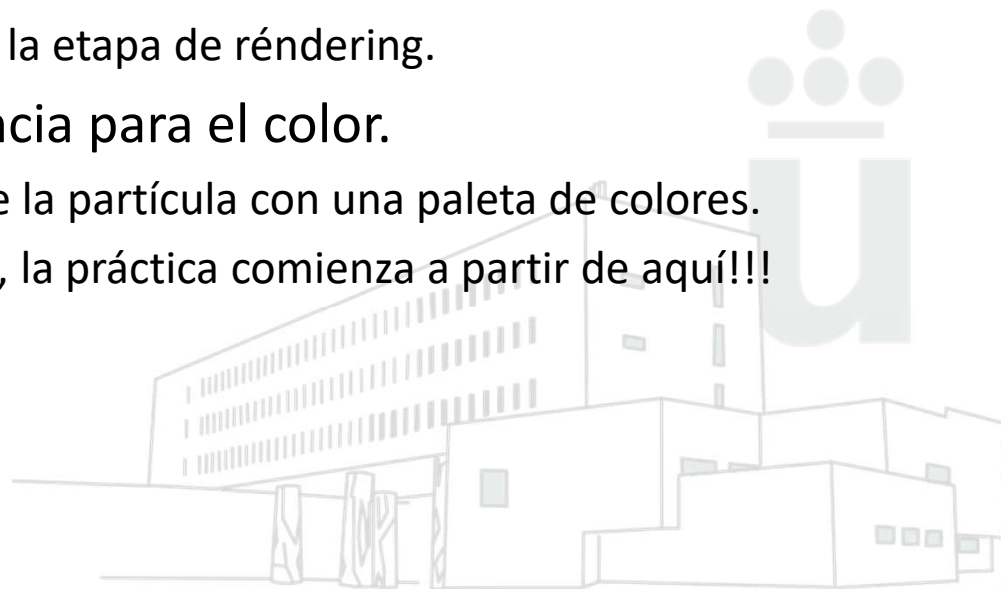
Desglose en sub-módulos

- Implementación de un sistema básico de partículas.
 - Utilizando shaders de cómputo
- Renderizado en orden:
 - Renderizado de las partículas para lograr un correcto funcionamiento del blending
- Cambio del comportamiento de las partículas:
 - Modificación de los parámetros que gobiernan el comportamiento de las partículas.
- Medir el rendimiento del sistema (benchmarking)
- Fecha de entrega: **22/03/2020**



Sistema de partículas básico

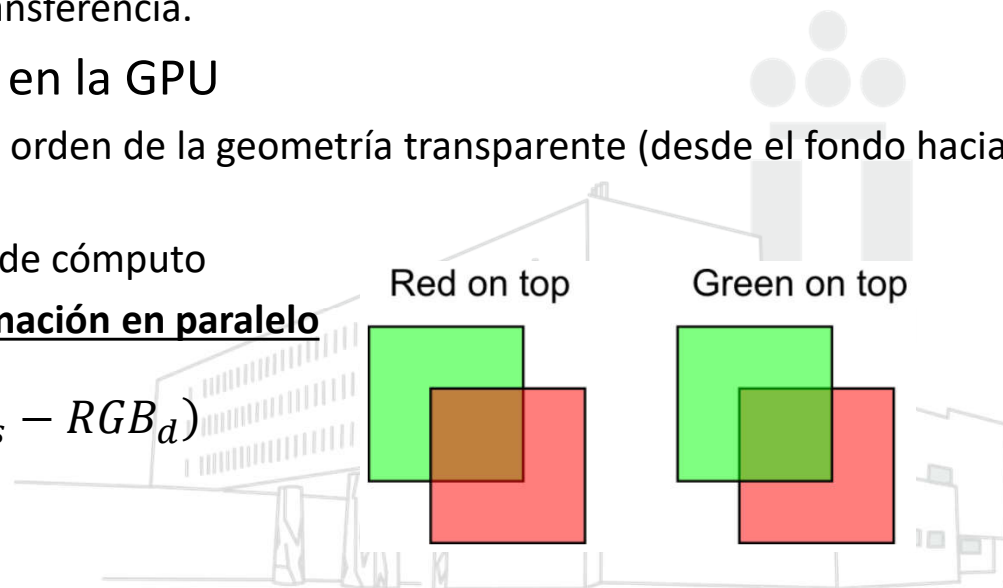
- Sistema de partículas muy básico pero ...
 - Se puede partir del proyecto visto en clase
 - Pero también se puede desarrollar uno propio
 - Todo el procesamiento debe hacerse en GPU.
 - **Debe utilizar** shaders de compute para el procesamiento de las partículas.
 - Para el benchmarkin: Se podría comparar con el mismo procesamiento en CPU.
 - Debe estar conectado con la etapa de réndering.
 - Función de transferencia para el color.
 - Eg. Codificar la vida de la partícula con una paleta de colores.
 - Esta parte no es evaluable, la práctica comienza a partir de aquí!!!
 - Ejercicios propuestos
 - Benchmarking



Renderizado en orden

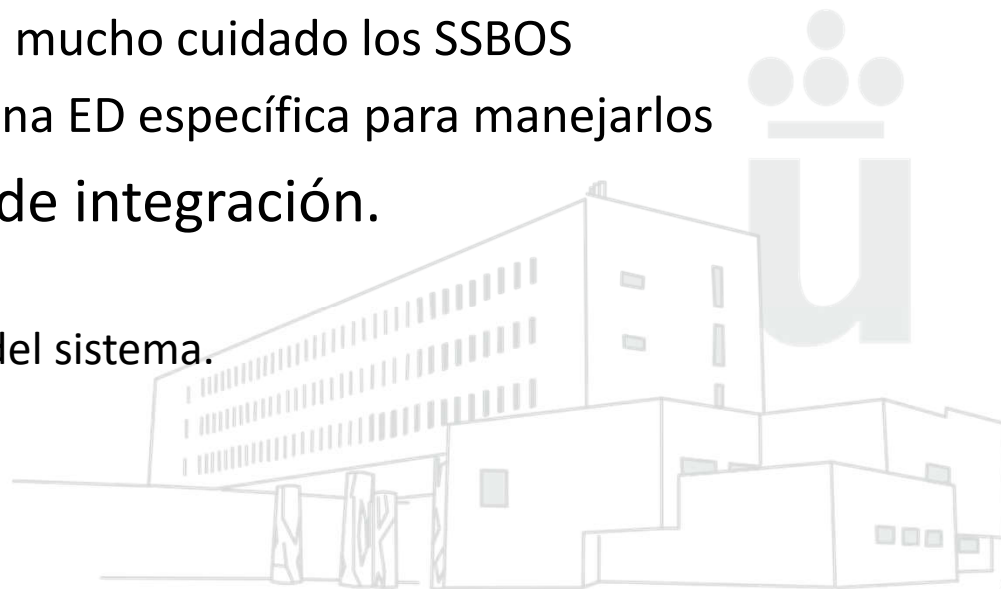
- Renderizado en orden (back to front).
 - Mejorar la calidad del rendering
 - Sustituir las partículas como billboards (quads orientados hacia la cámara)
 - Aprovechar etapa de geometría
 - Colorear o texturar estos quads pero con componente Alpha.
 - Adaptar la función de transferencia.
 - Utilización del blending en la GPU
 - Requiere renderizado en orden de la geometría transparente (desde el fondo hacia adelante)
 - Perfecto para un shader de cómputo
 - **Algoritmo de ordenación en paralelo**

$$RGB_d = RGB_d + A_s * (RGB_s - RGB_d)$$



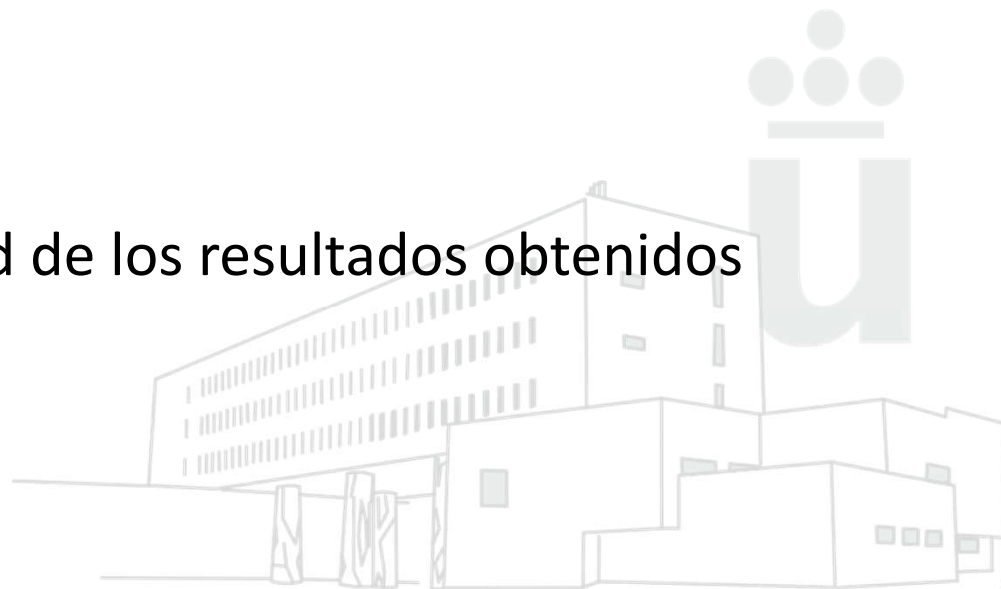
Modificación del comportamiento de las partículas

- Cambiar el modelo físico de comportamiento de las partículas.
 - Diferentes comportamientos
 - Explosiones.
 - Fuego.
 - Fuentes.
 - Añadir varios emisores y atractores
 - Ojo!, configurar con mucho cuidado los SSBOS
 - Puede requerir alguna ED específica para manejarlos
 - Cambiar el modelo de integración.
 - Verlet, RK4 ...
 - Probar la estabilidad del sistema.



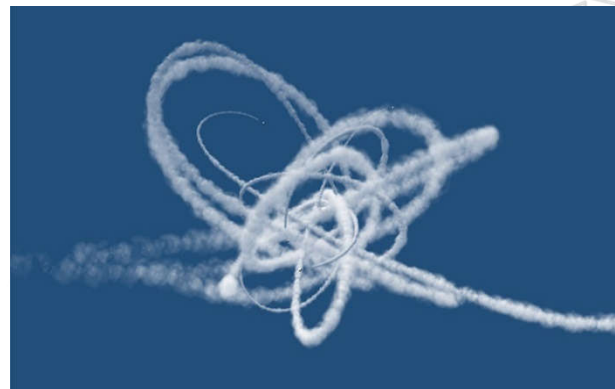
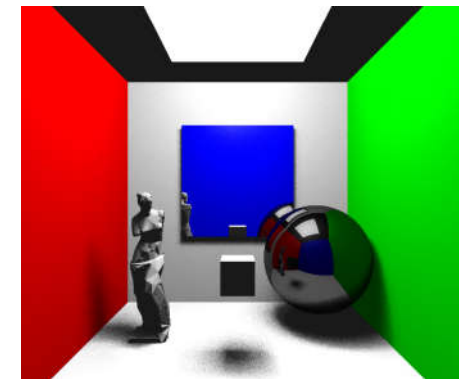
Benchmarking

- Debe hacerse un análisis de rendimiento
- Medir parámetros susceptibles de influir en el rendimiento
 - Número de partículas
 - Configuración del GRID
 - Algoritmos de ordenación
 - Número de emisores y atractores
 - Métodos de integración
 - ...
- CPU VS GPU
- Análisis en profundidad de los resultados obtenidos
 - Conclusiones.



Ampliación del módulo práctico 2

- Simulación de fenómenos naturales
 - Humo, Galaxias, etc.
- Simulación de objetos deformables
 - Pelo, hierba, tejidos etc.
- Filtros visuales (procesamiento en espacio de imagen)
 - Convoluciones, blurring, nprs etc.
- Traza de rayos.
- Otras técnicas que en las que os gustaría trabajar.
- ... Vuestras propias propuestas 😊.



Bibliografía

- Reeves, W. T. (1983). Particle systems—a technique for modeling a class of fuzzy objects. ACM Transactions on Graphics (TOG), 2(2), 91-108.
- Everitt, C. (2001). Interactive order-independent transparency. White paper, nVIDIA, 2(6), 7.
- Bavoil, L., & Myers, K. (2008). Order independent transparency with dual depth peeling. NVIDIA OpenGL SDK, 1-12.
- Bailey, M., & Cunningham, S. (2016). Graphics shaders: theory and practice. CRC Press.

