Estudio e implementación de técnicas de renderizado en tiempo real sobre un prototipo de motor gráfico

Alejandro Catalán Cebollada
Treball Fi de Grau en Enginyeria Informatica
Escola Superior Politècnica UPF
Any 2015
Director del treball: Javier Agenjo, Departament GTI

Resumen

Análisis de diferentes técnicas de renderizado en tiempo real e implementación de Deferred Shading sobre un engine WebGL.

Indice

1. Introducción

Introducción

La gran evolución de los navegadores web nos permite renderizar imágenes 2D/3D.

2 subcampos dentro de la generación de imágenes:

- **Generación off-line:** proceso mas elaborado y detallado.
- **Generación en tiempo real:** a partir de un modelo 2D o 3D, utilizando la GPU para la mayoría de los cálculos.

Introducción

- **Motivación:** estudiar diferentes técnicas de renderizado, y su rendimiento en WebGL sobre un engine desarrollado expresamente para el proyecto.

- **Background personal:** gran interés por la Computación Gráfica y la Ingeniería de Videojuegos.

2. La Web

La Web

World **W**ide **W**eb (**WWW**): conjunto de documentos y otros recursos interconectados.

Documentos web inicialmente estáticos formados por texto plano.

Con los años aparecieron lenguajes de programación, librerías y elementos que mejoraron las capacidades de la web.

La Web

- **HyperText Markup Language (HTML)**
- **Cascading Style Sheets (CSS)**
- **JavaScript**
- **<canvas>**
- **CSS3**
- **WebGL**

3. Graphics Pipeline

Graphics Pipeline

Con junto de pasos a realizar para transformar una escena 3D en una imagen 2D.

Modelos de Iluminación y Sombreado

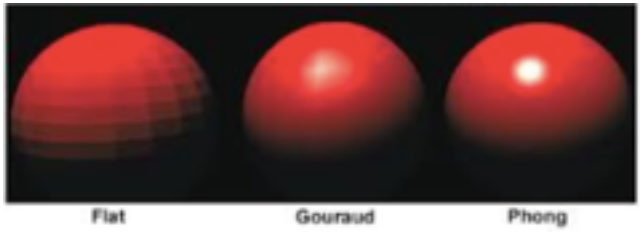
Calculo de la intensidad de color de cada punto de la escena.

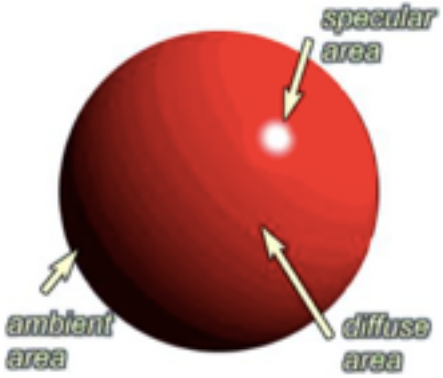
Tipos de luz:

- **Luz Direccional**
- **Luz Posicional**
  - **Luz Focal**

Modelo de iluminación de Phong: los objetos no emiten luz, solo reflejan la luz que les llega.

Sombreado: Se provee de un modelo de iluminación para la visualización de los polígonos del objeto.





Multiples Luces

- **Multipass Lighting: calcular iluminación de cada luz sobre el objeto y sumarla a la imagen real.**
- **Varias luces por shader: el shader recibe la información de un grupo de luces para realizar el calculo de iluminación.**
- **Lightmaps: estructura de datos precomputada que contiene el brillo de la superficie del objeto.**

3.1 Forward Rendering

Forward Rendering

- Técnica estándar mas utilizada por los engines gráficos.
- `O(N_fragmentos_geometria x L_luces)`
- Desventajas
  - +Luces +Objetos -> Gran caída de rendimiento
  - Repite trabajo en cada pasada
  - Trabajo inútil con polígonos ocluidos

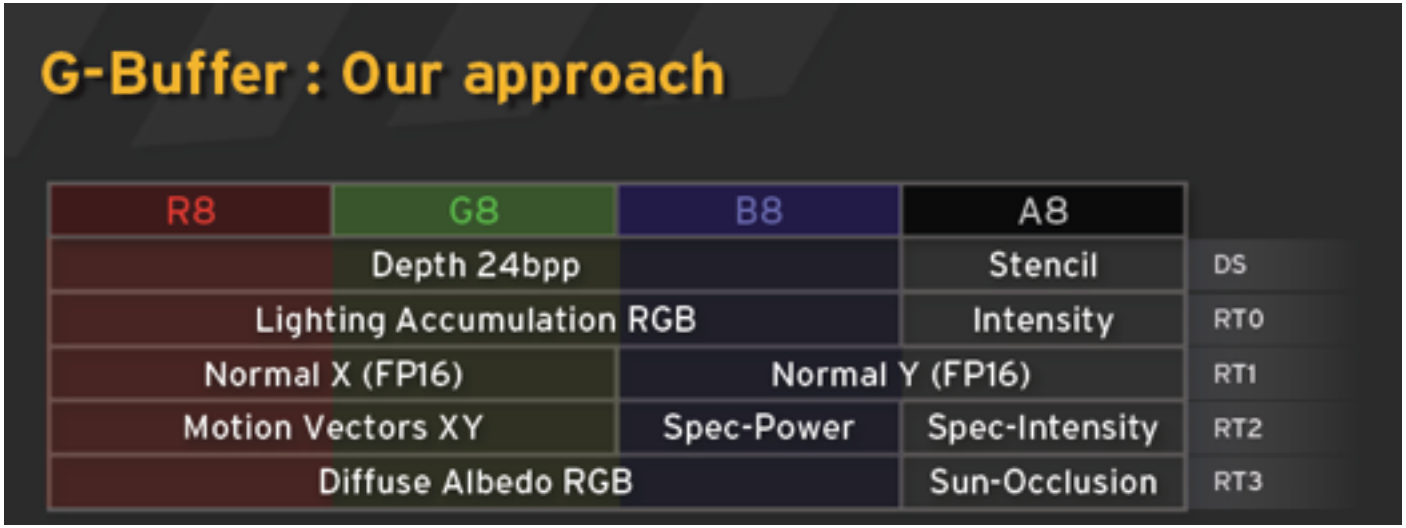
3.2 Deferred Rendering

Deferred Rendering

- Una primera pasada para generar el G-Buffer
- Una segunda pasada para realizar el calculo de iluminación
- `O(N_fragmentos_geometria + L_luces)`
- Desventajas
  - Multiples render targets
  - Gran ancho de banda para gestionar los buffers
  - No hay transparencias
  - Anti-Aliasing generado por hardware
  - Solo un tipo de material

Deferred Rendering

G-Buffer



Deferred Rendering

Tipos de Deferred Rendering

- Deferred Shading
- Deferred Lighting/Light pre-pass
- Inferred Rendering
- Tiled Shading
- Forward+

4. WebGL

WebGL

Web Graphics Library

- API JavaScript para el render de gráficos 2D/ 3D
- Basado en OpenGL ES 2.0, Shaders en GLSL
- Integrado con la interfaz de Document Object Model (DOM) de HTML
- Gestión de memoria automática gracias a JavaScript

WebGL

Ventajas

- API basada en OpenGL, familiar y aceptado
- Compatibilidad entre plataformas como entre navegadores
- Integración con HTML
- Gráficos 3D acelerados por hardware
- Entorno de scripting JavaScript permite ejecutar y debugar sin tener que compilar y linkar código.

Desventajas

- JavaScript es un lenguaje no tipado
- No tiene Geometry Buffer, no tiene Geometry Shaders
- No es capaz de gestionar buffers de indices 32 bits, no es capaz de tener buffers de mas de 65536 vertices, implica dividir el Vertex buffer para meshes grandes.

5. Engine Desarrollado

5.1. Diseño

Diseño del Engine Desarrollado

El engine se desarrolla sobre un entorno web utilizando WebGL.

Nos permite una fácil ejecución y debug, sin depender de grandes IDE, sobre cualquier dispositivo con acceso a la web y desarrollado sobre cualquier bloc de notas.

Requerimientos del Engine basados en intentar emular las características de los grandes engines comerciales.

Diseño del Engine Desarrollado

G-Buffer

R	G	B	A
posición.x	posición.y	posición.z	
normal.x	normal.y	normal.z	
color_textura.r	color_textura.g	color_textura.b	

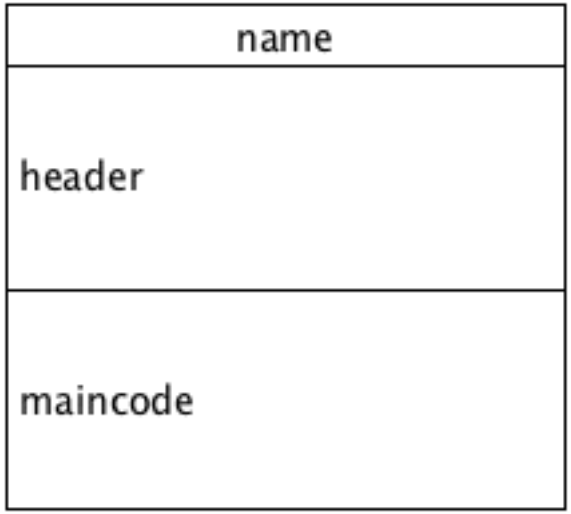
Diseño del Engine Desarrollado

G-Buffer

R	G	B	A
profundidad			
normal.x	normal.y	normal.z	
color_textura.r	color_textura.g	color_textura.b	

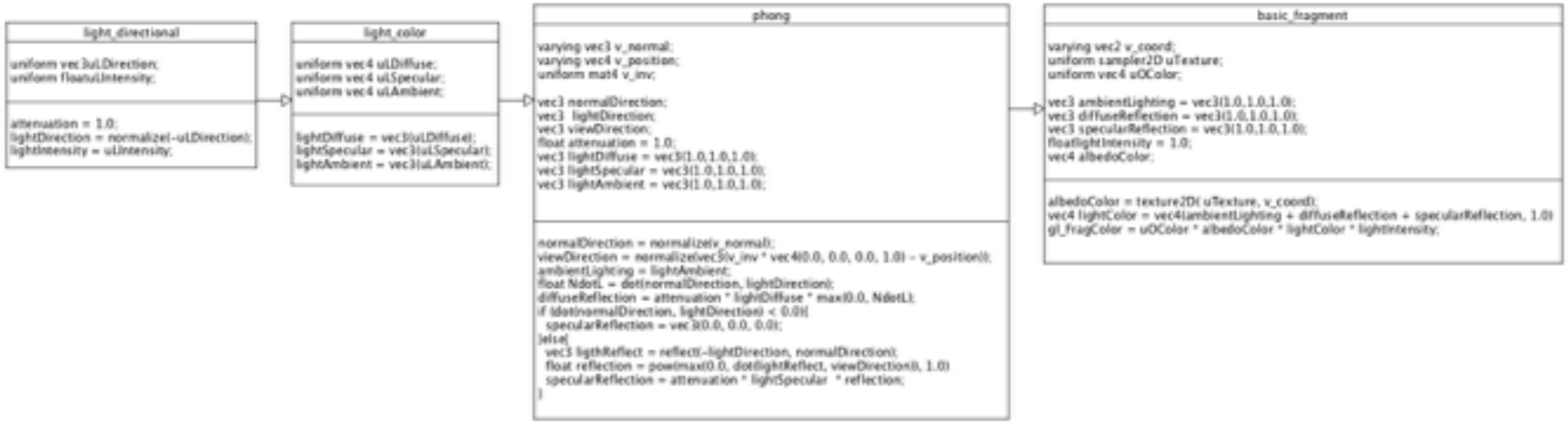
Diseño del Engine Desarrollado

MicroShader Manager



Diseño del Engine Desarrollado

MicroShader Manager



5.2. Estructura

Estructura del Engine Desarrollado

- **App:** inicializa y pone en funcionamiento la aplicación, gestiona el bucle principal definiendo el comportamiento y guarda el objeto Scene.
- **Scene:** contenedor de todos los objetos, gestiona la actualización y propagación de eventos a los objetos.
- **Renderer:** gestiona la Pipeline, recibe los objetos, luces y cámara, y ejecuta las funciones de pintado Forward o Deferred según la configuración del Engine.

Estructura del Engine Desarrollado

GameObject + Components

- Transform
- ObjectRenderer
- Light
- Camera

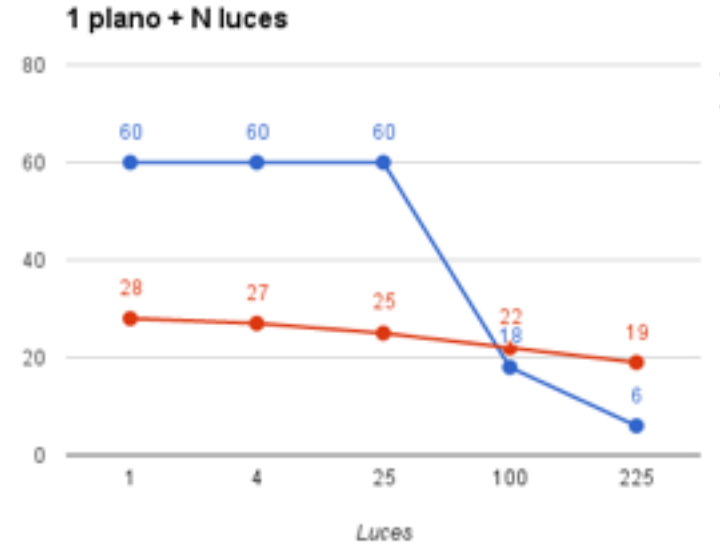
- KeyController, MouseController & RandomMovement

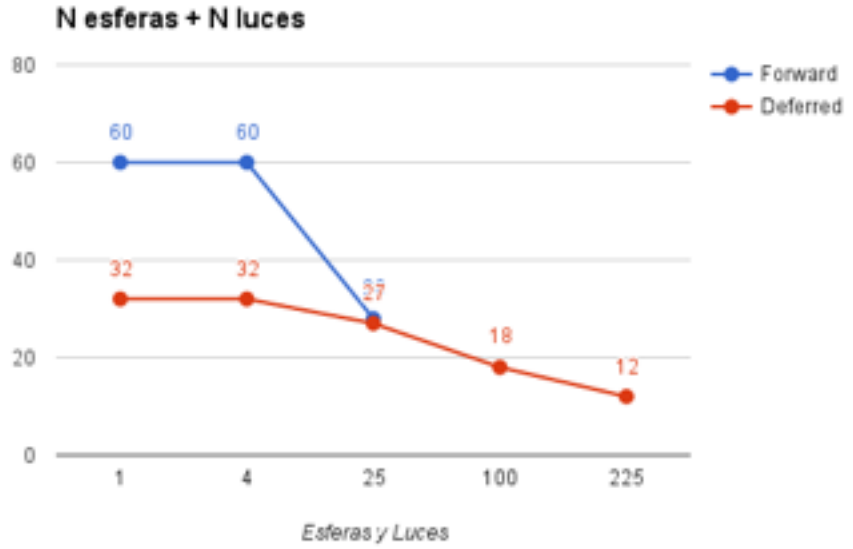
Estructura del Engine Desarrollado

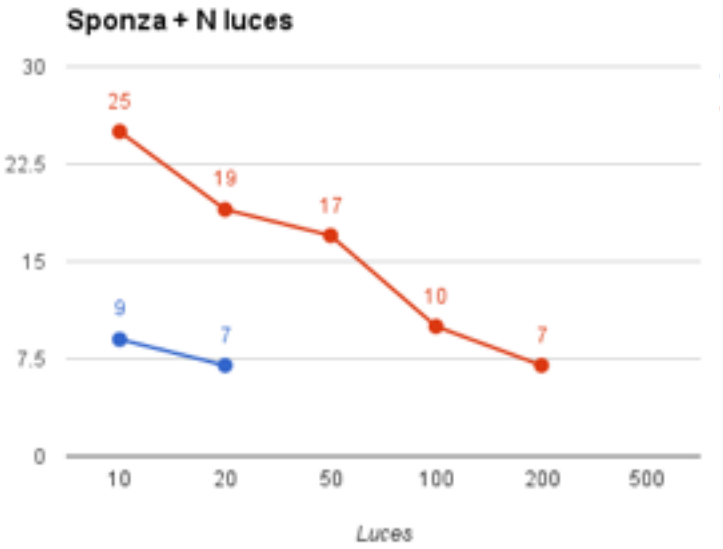
- **MicroShaderManager:** obtiene los fragmentos de shader de un XML y devuelve un shader completo.
- **GUI:** crea un interfaz de usuario para la aplicación, para su fácil manejo.
- **Benchmark:** provee a la escena de composiciones de objetos para poner a prueba el rendimiento.

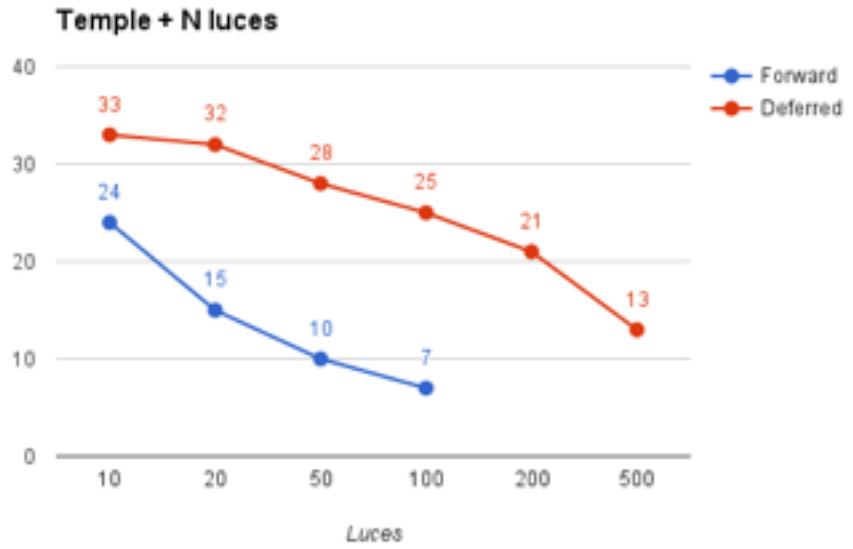
6. Resultados

Resultados


1 plano + N luces
80
60
40
20
0
60
60
60
28
27
25
22
18
19
6
1
4
25
100
225
Luces


N esferas + N luces
80
60
40
20
0
60
60
32
32
27
18
12
1
4
25
100
225
Esferas y Luces
Forward
Deferred


Sponza + N luces
30
22.5
15
7.5
0
25
19
17
10
7
9
7
10
20
50
100
200
500
Luces


Temple + N luces
40
30
20
10
0
33
32
28
25
21
13
24
15
10
7
10
20
50
100
200
500
Luces
Forward
Deferred

7. Conclusiones

Conclusiones

- Deferred es capaz de renderizar escenas con un numero mucho mayor de luces que Forward Rendering.
- G-Buffer supone un coste de rendimiento aceptable.
- G-Buffer supone un handicap para el detalle de la imagen.

Conclusiones

Posibles Mejoras

- Generación de escenas en tiempo de ejecución.
- Sistema de partículas, motor de físicas, animación de meshes, etc.
- Proyección de sombras y uso de materiales.
- Adaptación del Engine a dispositivos móviles.

Trabajo Futuro

- Implementación de las mejoras mencionadas.
- Implementación y análisis de las técnicas de Deferred estudiadas.

UNIVERSITAT
POMPEU FABRA