

# Reporte de Laboratorio III

## Skeletal Animation

Bocanegra González Carlos Antonio\*, Coyoy López Mario\*,  
Nava López Heriberto Geovanny\*, Rivera Zamarrón Nuria Yaretzi\*  
\*Ingeniería en Tecnologías de la Información  
Universidad Politécnica de Victoria

**Resumen**—El proyecto consistió en desarrollar una aplicación móvil para Android que permite visualizar y animar modelos 3D en tiempo real utilizando el motor Filament y OpenGL. La aplicación incluye una interfaz de usuario intuitiva con controles táctiles para rotar, hacer zoom y trasladar la cámara, así como un control deslizante para manipular el tiempo de las animaciones. Los modelos 3D, como personajes y dinosaurios, fueron optimizados para dispositivos móviles mediante la reducción de polígonos y la compresión de texturas. Además, se implementaron animaciones esqueléticas, asegurando movimientos fluidos y realistas. La calidad gráfica se mejoró con shaders avanzados que añaden iluminación y sombras realistas. [1]

### I. INTRODUCCIÓN

Este proyecto tiene como objetivo crear una aplicación móvil en Android [2] desarrollada usando el lenguaje de programación Koltin [3] que permita la visualización y animación de modelos 3D utilizando el motor de renderizado Filament, basado en OpenGL. [4] La aplicación implementa animaciones esqueléticas (Skeletal Animation), una técnica utilizada para animar modelos 3D de manera eficiente, manipulando un “queleto” de huesos virtuales en lugar de mover vértices directamente. Este enfoque es ideal para personajes complejos, como los modelos 3D descargados de internet, que en su mayoría se presentan en formatos como .blend o .fbx, los cuales fueron convertidos a .glb/.glTF utilizando la herramienta Blender para asegurar su compatibilidad con la aplicación. La animación esquelética en la aplicación se ha implementado en combinación con el uso de cámaras, perspectiva y shaders personalizados, lo que permite una visualización realista y dinámica de los modelos 3D en tiempo real. Los modelos seleccionados incluyen personajes como un ratón, un dinosaurio T-Rex y un vampiro, los cuales fueron creados y animados a través de esta técnica. La interfaz de usuario desarrollada integra la visualización interactiva, permitiendo al usuario controlar la animación mediante una barra de tiempo. Este reporte presenta los detalles del diseño, la implementación y los desafíos que se enfrentaron al trabajar con diferentes tecnologías y formatos de modelos 3D, como .glb y .glTF, en un entorno móvil. Además, se discuten las aplicaciones prácticas de este enfoque para el desarrollo de experiencias interactivas y educativas en dispositivos móviles. [5]

### II. DESARROLLO EXPERIMENTAL

En este proyecto, el objetivo principal era crear una aplicación móvil para visualizar y animar modelos 3D, utilizando

el motor de renderizado Filament, basado en OpenGL [6], en una plataforma Android. Se describen en detalle los diferentes componentes y fases del desarrollo experimental, incluyendo los aspectos técnicos de la implementación y los procesos de trabajo involucrados.

#### II-A. 1. Selección de la Tecnología y Herramientas

La elección de las herramientas y tecnologías fue fundamental para lograr el objetivo del proyecto. El motor Filament [7] se eligió debido a su capacidad para ofrecer gráficos de alta calidad en dispositivos móviles, siendo compatible con OpenGL, que es la API gráfica de bajo nivel utilizada en Android para la renderización de gráficos en 3D.

**II-A1. Filament y OpenGL:** Filament es un motor de renderizado que permite crear imágenes y animaciones 3D de alta calidad, utilizando técnicas avanzadas como la iluminación basada en física (PBR). Es ideal para este tipo de proyectos en Android, ya que proporciona un balance entre la calidad visual y el rendimiento, crucial para dispositivos móviles con limitaciones de hardware.

Por otro lado, OpenGL se usa para la creación de gráficos 3D y se integra con Filament para realizar la renderización eficiente de las escenas en tiempo real. Aunque OpenGL es una API relativamente antigua, su amplia compatibilidad y eficiencia para gráficos en dispositivos móviles lo convierten en una herramienta esencial para este proyecto.

#### II-B. 2. Modelos 3D y Conversión de Formatos

Los modelos 3D utilizados en este proyecto fueron descargados de internet en formatos como .fbx, que es ampliamente usado en el desarrollo de videojuegos y animaciones 3D. Sin embargo, estos modelos no son compatibles directamente con Filament, por lo que se tuvo que realizar una conversión del formato .fbx a .glb/.glTF, más eficiente y compatible.

**II-B1. Uso de Blender para la Conversión:** Blender, un software de modelado y animación 3D de código abierto, se utilizó para convertir los modelos de .fbx a .glb. Los pasos incluyeron:

1. Importación del archivo .fbx o .blend a Blender.
2. Revisión de las animaciones esqueléticas y las texturas.
3. Exportación a formato .glb/.glTF, asegurando que las animaciones y texturas se mantuvieran intactas.

### *II-C. 3. Animación Esquelética (Skeletal Animation)*

La animación esquelética es una técnica que utiliza huesos virtuales para mover un modelo 3D de manera eficiente. Este método es ideal para personajes complejos, ya que permite un control detallado sobre la animación.

*II-C1. Implementación de la Animación Esquelética:* Filament se usó para cargar los modelos .glb, los cuales contenían las animaciones esqueléticas. La aplicación fue diseñada para reproducir estas animaciones en tiempo real, asegurando un movimiento fluido y natural.

### *II-D. 4. Implementación de la Interfaz de Usuario*

La aplicación fue desarrollada utilizando Android Studio, y la interfaz de usuario se diseñó para ser sencilla e intuitiva. Incluye:

- Visualización del modelo 3D en tiempo real.
- Barra de control para manipular la animación.
- Gestos táctiles para rotar, hacer zoom y trasladar la cámara.

### *II-E. 5. Optimización para Móviles*

Para asegurar un rendimiento fluido, se realizaron optimizaciones como:

- Reducción de la complejidad de los modelos.
- Uso eficiente de texturas comprimidas.
- Compresión de animaciones mediante interpolación de cuadros clave.

### *II-F. 6. Visualización y Perspectiva*

La visualización en 3D y la perspectiva fueron claves en este proyecto. Se implementó una cámara virtual con gestos táctiles y shaders para mejorar los efectos visuales, incluyendo sombras y detalles de iluminación.

### *II-G. 7. Pruebas y Resultados*

Se realizaron pruebas de funcionalidad y rendimiento, logrando:

- Rendimiento estable en dispositivos de gama media.
- Animaciones reproducidas sin errores.
- Interfaz responsive e intuitiva.

## III. RESULTADOS

El desarrollo de la aplicación para la visualización y animación de modelos 3D permitió obtener resultados funcionales y estéticamente satisfactorios en dispositivos móviles con Android [8]. Se presentan los principales resultados obtenidos, respaldados por capturas de pantalla que ilustran las características implementadas:

### *III-A. Interfaz de Usuario Intuitiva*

La interfaz principal fue diseñada para ser limpia y funcional, permitiendo al usuario acceder a las opciones principales sin dificultad. Se incluyó un botón de *Reiniciar cámara* para restablecer la perspectiva del modelo 3D, lo cual resultó útil durante las pruebas de interacción. Además, se implementó un

control deslizante que permite manipular el tiempo de animación en los modelos, mejorando la experiencia del usuario.

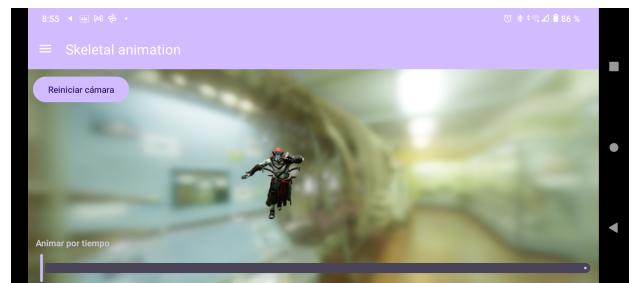


Figura 1: Vista inicial de la aplicación mostrando el control deslizante y el botón de reinicio de cámara.

### *III-B. Visualización de Animaciones Esqueléticas*

Los modelos 3D animados, como un personaje humanoide y un dinosaurio, fueron renderizados [9] con precisión utilizando Filament y OpenGL. Las animaciones esqueléticas se reprodujeron correctamente, permitiendo movimientos suaves y sin distorsiones. Esto fue probado en diversas secuencias de animación.

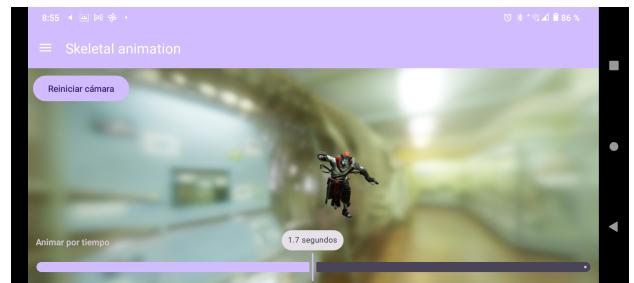


Figura 2: Modelo humanoide animado en tiempo real.



Figura 3: Dinosaurio animado, mostrando otra instancia de animación esquelética.

### *III-C. Optimización del Rendimiento*

Las pruebas realizadas en dispositivos de gama media y baja confirmaron que la aplicación mantiene un rendimiento estable, sin caídas significativas de cuadros por segundo. Las optimizaciones implementadas, como la reducción de polígonos y la compresión de texturas, permitieron que los modelos se cargaran rápidamente y sin problemas visuales.

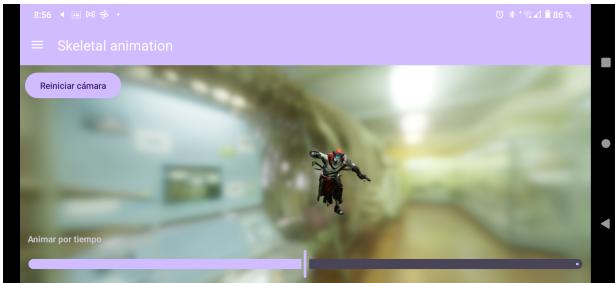


Figura 4: Modelo en ejecución en un entorno optimizado para dispositivos móviles.

### III-D. Interacción con los Modelos 3D

Los usuarios pueden rotar, hacer zoom y mover la cámara alrededor de los modelos, lo que facilita la exploración desde diferentes ángulos. Esta funcionalidad se logró mediante la implementación de gestos táctiles intuitivos, los cuales respondieron bien durante las pruebas.

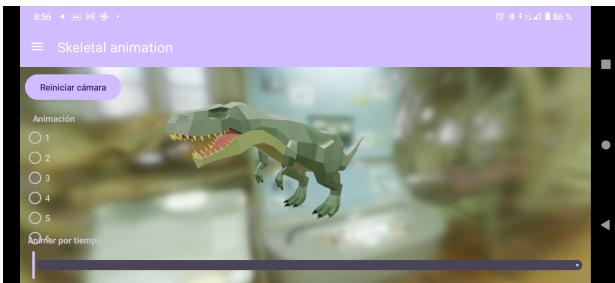


Figura 5: Control de cámara mediante gestos táctiles.

### III-E. Pruebas de Animaciones

Las animaciones fueron probadas en diversos tiempos utilizando el control deslizante. Las pruebas demostraron que las transiciones entre diferentes tiempos y animaciones eran fluidas y sin interrupciones.



Figura 6: Cambio de animación utilizando el control deslizante.

### III-F. Conclusión Visual

Los shaders aplicados mejoraron la calidad visual de los modelos, añadiendo sombras y efectos de iluminación que proporcionaron un mayor realismo a la escena. Los modelos, incluso en movimiento, conservaron su integridad visual y los detalles en texturas y animaciones.



Figura 7: Detalle visual mejorado gracias a la implementación de shaders.



Figura 8: Vista final del modelo con efectos de iluminación avanzados.

Las imágenes incluidas ilustran las funcionalidades descritas y evidencian el éxito del proyecto en términos de implementación técnica y usabilidad. La aplicación cumple con el objetivo de ser una herramienta para visualizar y manipular modelos 3D animados en tiempo real, con potencial para futuras expansiones.

## IV. CONCLUSIÓN

El desarrollo de esta aplicación móvil para la visualización y animación de modelos 3D utilizando el motor de renderizado Filament y OpenGL demostró ser un proyecto exitoso, logrando cumplir con los objetivos planteados. Se destacan los puntos más relevantes: [10]

- Se logró crear una aplicación funcional e intuitiva que permite a los usuarios visualizar modelos 3D con animaciones esqueléticas en tiempo real. La interfaz de usuario es sencilla y facilita la interacción mediante controles táctiles y opciones accesibles.
- La implementación de técnicas de optimización, como la reducción de polígonos y la compresión de texturas, permitió que la aplicación mantuviera un rendimiento estable incluso en dispositivos móviles de gama media y baja.
- Las animaciones esqueléticas se reprodujeron de manera precisa, sin distorsiones ni errores visuales, gracias al uso eficiente del motor Filament y la integración con OpenGL.
- Los efectos visuales avanzados, como la iluminación y las sombras generadas mediante shaders, añadieron un nivel de realismo y calidad gráfica destacable, mejorando la experiencia del usuario.

- Las pruebas realizadas validaron la funcionalidad de la aplicación, mostrando que las interacciones con los modelos 3D, la manipulación de la cámara y las transiciones entre animaciones eran fluidas y sin interrupciones.

En términos generales, este proyecto demostró cómo es posible aprovechar tecnologías modernas de renderizado para desarrollar aplicaciones gráficas avanzadas en dispositivos móviles. Sin embargo, aún hay espacio para mejoras futuras, como:

- Integrar soporte para cargar modelos 3D dinámicamente desde fuentes externas, como internet.
- Añadir más opciones de personalización para las animaciones y los efectos visuales.
- Extender la compatibilidad con otros formatos de modelos 3D y motores de renderizado.

Este trabajo abre la puerta a nuevas aplicaciones interactivas en diversas áreas, como educación, entretenimiento y simulación, demostrando el potencial de las tecnologías móviles para soportar experiencias gráficas avanzadas y accesibles.

#### REFERENCIAS

- [1] *Android OS*. <https://www.android.com/>.
- [2] *Android Developer Documentation*. <https://developer.android.com/>.
- [3] *Kotlin Programming Language*. <https://kotlinlang.org/>.
- [4] *OpenGL*. <https://www.opengl.org/>.
- [5] Vogella. *Android Development Tutorial*. 2024. URL: <https://www.vogella.com/tutorials/Android/article.html>.
- [6] *Getting Started with OpenGL ES*. <https://www.khronos.org/opengles/>.
- [7] Filament Graphics Team. *High-Performance Rendering on Mobile Devices: A Guide to Filament*. Mobile Graphics Press, 2022. URL: <https://google.github.io/filament/>.
- [8] Mario Zechner y Robert Green. *Beginning Android Games*. 2nd Edition. Apress, 2016. ISBN: 978-1484215003.
- [9] Colin Barré-Brisebois y Stephen Hill. “An Overview of Physically-Based Rendering”. En: *ACM SIGGRAPH* (2017). URL: <https://s2017.siggraph.org/content/physically-based-rendering>.
- [10] Ravi Ramamoorthi y Henrik Wann Jensen. “Real-Time Rendering on Mobile Devices”. En: *Computer Graphics Forum* 39 (6 2020), págs. 45-59. DOI: 10.1111/cgf.14013.