

Proyecto 4: The Ray Tracing Challenge

Profesor: Ernesto Rivera Alvarado

I. INTRODUCCIÓN

En este proyecto, el estudiante creará un motor de sintetizado de imágenes tridimensionales utilizando la técnica de ray-tracing. Toda la programación se hará en C sobre Linux.

II. RAY TRACER DE ESFERAS

Los estudiantes diseñarán y crearán una escena estética y creativa que represente algún escenario haciendo uso de las primitivas: esferas, discos, cilindros, conos en distintas posiciones, tamaños y colores.

La escena contará con varias (más de una) fuentes de luz, sombras, texturas y espejos.

La salida del programa será un archivo gráfico que no pierda calidad (se sugiere el formato .avs que se debe convertir a formato .jpg). Para el día de la revisión se tendrá ya preparada y desplegada la imagen final. También la escena se podrá generar en tiempo real en una ventana de OpenGL/Vulkan o SDL. El modo de generación de la escena será especificado en el archivo de texto de entrada.

III. DESPLIEGUE

El sistema generará un archivo en formato de imagen (png, jpeg, avs, ppm, bmp) en donde se guardará la imagen. La misma no debe perder calidad.

De igual manera, mientras la imagen se esté sintetizando, se mostrará en pantalla (SDL/OpenGL/Vulkan) los píxeles que han sido calculados.

IV. SOBRE LA REVISIÓN Y COMPLETITUD DEL PROYECTO

La evaluación del proyecto se dividirá de la siguiente manera:

- Ray tracer de n esferas con iluminación difusa y especular. que guarda la imagen sintetizada en un formato adecuado y muestra en pantalla el proceso de síntesis. 30%.
- Implementación de conos, discos, cilindros y planos de corte con iluminación difusa y especular. 10%.
- Implementación de sombras, espejos y transparencias en todas las primitivas descritas anteriormente. 10%.
- Implementación de texturas. 10%.
- Escena que incorpora **todos** los elementos descritos anteriormente en una escena compleja y elaborada. 40%.

V. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto “no revisable” y recibe un 0 de calificación inmediata:

- Todo el código debe de estar escrito en C sobre Linux.

- Las funciones de OpenGL/Vulkan o SDL se limitan a crear una ventana y pintar píxeles en colores definidos.
- No debe presentarse “segmentation fault” por ninguna razón.
- La colaboración entre grupos se considera fraude académico.
- Todo el código debe estar escrito en C (no C++).
- El proyecto debe compilar y ejecutar en Linux. Todo debe estar **integrado**, explicaciones del tipo “*todo está bien pero no pudimos pegarlo*”¹ provocan la cancelación automática de la revisión.
- Todos los algoritmos gráficos deben haber sido desarrollados por los estudiantes. El uso de alguna biblioteca (MESA / GLUT / OpenGL / Vulkan / SDL o equivalente) se limita a crear una ventana, alterar el color de los píxeles, encender el píxel.
- La presentación debe ser de mucha calidad.
- Hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.
- La presentación debe hacerse en clase en la computadora del estudiante.
- Debe seguir a cabalidad lo descrito en la **Sección II y III** del presente documento.
- Los estudiantes deben demostrar dominio completo de su proyecto en la presentación oral.
- Los estudiantes deben utilizar el sistema de control de versiones Git tal como se describe en este documento.

La evaluación de este proyecto está sujeta a la presentación oral del mismo, en la que el estudiante debe mostrar un dominio completo del trabajo realizado a nivel de código. En caso de que el estudiante no muestre un dominio completo del trabajo, se considerará el trabajo como “no revisable” y se asignará la nota de cero. La defensa del proyecto será grabada.

En caso de trabajar en pareja, ambos integrantes deberán tener un dominio completo del trabajo que están presentando. El profesor podrá hacerle preguntas a cualquiera de los integrantes y este deberá contestar acordemente. En caso de que se evidencia que uno de los integrantes desconoce el funcionamiento de alguna parte del trabajo presentado, se le asignará la nota de cero a ambos.

VI. DESARROLLO DEL PROYECTO

El proyecto está pensado para desarrollarse individualmente, sin embargo, los estudiantes que deseen reforzar la habilidad de trabajo en equipo pueden entregar el proyecto en parejas. El profesor les hace la aclaración de que el trabajo con

¹esto incluye los supuestos casos cuando alguien del grupo de trabajo no hizo su parte – el profesor no está interesado en sus problemas de organización.

un compañero conlleva dificultades de coordinación, división de trabajo y sobre todo de “pegar o juntar ambas partes”. En experiencias propias del profesor, se les comenta que en ocasiones el trabajo de juntar, acoplar y corregir partes desarrolladas por diferentes personas conlleva más tiempo y trabajo que la realización individual.

VII. GIT

Los estudiantes deberán utilizar el sistema de control de versiones Git, específicamente a través de GitLab. Para ello configurarán desde sus máquinas Gitlab, para poder hacer cualquier acción (ejemplo *push*, *pull*, *commit*) desde **consola de texto**, sin la necesidad de ingresar la contraseña constantemente. Además agregarán al repositorio del proyecto al usuario Ernesto.cursos, donde el profesor podrá observar los avances y desarrollos del proyecto por semana, día, hora, minuto y segundo.

VIII. CONSIDERACIONES SOBRE PLAGIO

El trabajo presentado por los estudiantes (ya sea a nivel individual y de parejas) debe ser de su propia autoría. No se permite utilizar código realizado por un tercero (sin importar la licencia de dicho código) o entre compañeros del mismo curso, y en caso de que esto ocurra, será considerado plagio por lo que se seguirá el proceso correspondiente. Códigos encontrados en “github” de los cuales solo se tomaron “partes” será considerado plagio, el estudiante es el responsable de desarrollar absolutamente todo su código, a excepción del código que se menciona como base en el presente documento. De igual manera, cualquier código que el estudiante no tenga la capacidad de explicar, será considerado como plagio.

IX. FECHA DE ENTREGA

Las demostraciones se harán en la **semana 15** asignadas por citas distribuidas aleatoriamente en las dos lecciones de la semana.

La carpeta comprimida `.zip` de su proyecto debe contener los archivos fuentes `.c` y complementarios, además del archivo `Makefile` que genera el ejecutable del código fuente. El nombre de la carpeta comprimida debe de ser los apellidos de integrantes del grupo de trabajo (por ejemplo `rivera-alvarado.zip`) y este será subido al correo del profesor y al Tec Digital el día de la revisión a antes de las 6 am. Los estudiantes deberán defender el trabajo realizado y mostrar un dominio completo del trabajo desarrollado. El no tener conocimiento de cómo funcionan partes del proyecto invalida la entrega y se asignará la nota de cero. El título del correo será [Proyecto 4 - Gráficos por Computadora].