



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2912 – Computación Gráfica (I/2013)

Tarea 1: 1era Parte (40 %)

16 marzo de 2013 - 29 marzo de 2013

Fecha de entrega: Viernes 29 de marzo 18:00

1. Descripción

En esta primera parte de la tarea, usted deberá implementar una versión parcial del algoritmo de ray-tracing, considerando sólo la intersección de los rayos con los objetos, y no el shading de estos. Su programa deberá ser capaz de revisar la intersección para todos los triángulos y esferas en la escena y en caso de detectar intersección, pintar el pixel con el color difuso del objeto intersectado.

Para desarrollar la tarea se les proveerá con una serie de clases útiles para manejar escenas 3D cargadas desde un archivo. Las clases relevantes para esta tarea son:

- Clase **Scene** maneja los elementos de la escena incluyendo un color de fondo, los objetos, materiales y la cámara. La escena es cargada desde un archivo XML. Un ejemplo de escena es provisto en el proyecto de la tarea publicado en la página con el nombre «cornellBox.xml»
- Clase **SceneBackground** maneja la información del color de fondo y luz de ambiente. **Para esta tarea solo se utilizará el color de fondo**
- Clase **SceneCamera** maneja la información de la cámara incluyendo su posición y orientación.
- Clase **SceneObject** representa un objeto 3D a ser desplegado en la escena. Contiene información de la traslación, rotación y escalamiento del objeto. Existen tres clases que heredan de este, **SceneSphere**, **SceneTriangle** y **SceneModel**. **Para esta tarea solo se utilizarán SceneTriangle y SceneSphere**
- Clase **SceneTriangle** representa un triángulo 3D a ser desplegado en la escena, conteniendo una lista de 3 vértices y de 3 materiales (que a su vez contienen los colores asociados a cada vértice).
- Clase **SceneMaterial** contiene información de iluminación y textura aplicable a un objeto. **Para esta tarea solo se utilizará el atributo Diffuse el cual corresponde a un color RGB**
- Clases **Matrix** y **Vector** con implementaciones de matrices y vectores de 4 dimensiones, permitiendo todas las operaciones básicas entre ellos.
- Clase **OpenGLRenderer** la cual corresponde al renderizado 3d de la escena ocupando las herramientas de OpenGL. La idea de este renderer es que usted lo utilice como punto de comparación para su propio renderer. **No se debe modificar este código, y no es necesario que lo entiendan para esta tarea.**
- Clase **RaytraceRenderer**, la cual usted deberá completar con su algoritmo. Esta clase contiene los siguientes métodos:

- **void Render()** Realiza el rendering ocupando el algoritmo de raytracing. Pinta los pixeles con los colores almacenados en la matriz `Vector [,] buffer`. **No es necesario modificar este método.**
- **void Update()** Método llamado continuamente para ir actualizando progresivamente el display, y así no tener que esperar tener la imagen completa para ir viendo el resultado del algoritmo. Realiza una llamada al método `Vector CalculatePixel(int x, int y)` almacenando el resultado para cada pixel en el `buffer`. **No es necesario modificar este método.**
- **Vector CalculatePixel(int i, int j)** Método que usted deberá implementar con su algoritmo de raytracing que devuelva el color correspondiente al pixel. Este método es llamado para cada pixel (i, j) de la imagen en el método `Update`. Para esta parte de la tarea en que no se exige shading, este método deberá devolver el color difuso asociado al material del objeto intersectado.
- **void SaveImage(Vector[,] buffer, string fileName)** Guarda un buffer de pixeles en un archivo de imagen. Este método no se debe utilizar aún, será utilizado en la segunda parte de la tarea.

En la clase `Program` se pueden modificar los valores de `width` y `height` para cambiar la resolución de la escena. Esto es de mucha utilidad para probar rápidamente el algoritmo de raytracing que puede demorarse un tiempo considerable. Otras funcionalidades útiles que vienen implementadas son:

- Pueden elegir que renderer ocupar para generar la escena: presionando la tecla '1' se selecciona el renderer OpenGL que viene implementado en el código base, y que les sirve para comparar; presionando la tecla '2' se selecciona el Raytrace renderer que ustedes deben implementar.
- Pueden modificar los parámetros de posición y target de la cámara ocupando las teclas 'a', 's', 'd', 'w', 'i', 'j', 'k', 'l' moviendo el punto de vista de la escena. Lo recomendado es realizar estos movimientos estando en modo renderer OpenGL de manera de ver en tiempo real los cambios.

2. Entrega y Consultas

La entrega deberá ser realizada enviando un mail a iic2912@gmail.com con asunto «Tarea 1a - Nombre Apellido». Debe entregar un archivo comprimido que contenga todos los archivos necesarios para ejecutar su programa. La tarea deberá ser entregada a más tardar el **Viernes 29 de marzo a las 18:00**.

Las consultas deben hacerlas a través del foro del sitio Web del curso. Este es el único medio oficial para dar respuesta a sus preguntas sobre la tarea.

3. Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería¹.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno copia un trabajo, obtendrá nota final 1,1 en el curso y se solicitará a la Dirección de Docencia de la Escuela de Ingeniería que no le permita retirar el

¹http://dsrd.uc.cl/images/stories/pdf/AlumnosUC/Info_Academ/Reglamentos/alumnopregrado.pdf

curso de la carga académica semestral. Por “copia” se entiende incluir en el trabajo presentado como propio partes hechas por otra persona.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile². Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

²https://intrawww.ing.puc.cl/siding/dirdes/web_docencia/pre_grado/reglamentos_int/integridad_academica.phtml