

GRÁFICOS Y **VISUALIZACIÓN DE DATOS**

Práctica 2 **(Ray Traicing)**

Víctor Fernández
David García
31/05/2016

Sumario

Objetivos.....	3
Resumen de la práctica.....	3
Implementar intersect.....	3
Calculo del rayo primario.....	3
Calculo de sombras.....	4
Calculo de reflexiones.....	4
Opcionales y puntos implementados.....	4
Opcional.....	4
Puntos implementados.....	5
Resultado y tomas falsas.....	5
Resultado.....	5
Tomas falsas.....	6

Objetivos

En esta práctica aprenderemos a usar el algoritmo de Ray Tracing. El algoritmo trata generar un rayo que vaya desde el observador a los objetos, píxel a píxel. Para entender el algoritmo primero hay que entender que son las coordenadas locales, de mundo, de cámara, de window y de viewport. Además se ha de entender como pasar de unas coordenadas a otras.

También aprendemos a calcular las sombras y las reflexiones de los objetos.

Resumen de la práctica

Implementar intersect

Este apartado trata de generar las funciones en los objetos para ver si, dado un punto del objeto, el rayo lo atraviesa. También hay que mirar que el objeto esta delante de la cámara y no detrás.

Antes de generar esas funciones hay que generar los objetos que habrá en el escenario. En este caso serán esferas, planos y triángulos. Una vez comprobamos que el rayo atraviesa el objeto tenemos que comprobar que objeto queda mas cerca de la cámara.

Calculo del rayo primario

Para poder aplicar el calculo del rayo primario, hay que decirle a la escena que profundidad tiene, esto se hace con el zNear y zFar, también hay que generar dos matrices.

La matriz de model View, que nos permite pasar de coordenadas de mundo a coordenadas de cámara. Y la matriz de proyección, con la que podemos pasar de coordenadas de cámara a coordenadas de Window.

Pero nosotros lo que tenemos que hacer es ir de Viewport a coordenadas de mundo, para ello debemos obtener los pixeles del Viewport y pasarlos a coordenadas de Window. Una vez en coordenadas de Window tenemos que hacer la inversa de la matriz de proyección para obtener las coordenadas de cámara. Estando en cámara hay que hacer la inversa de la matriz de model View para llegar a las coordenadas de mundo.

De esta forma podemos pasar del pixel al objeto con el rayo primario.

Calculo de sombras

Para obtener las sombras debemos aplicar primero de todo el algoritmo de Blinn-Phong, que como vimos en la practica anterior, es un algoritmo para obtener el color de la escena a partir de las luces y los materiales de los objetos.

La forma de comprobar si en un punto existe sombra, es generando un nuevo rayo. Donde el origen del rayo es el punto actual del objeto y el vector dirección es hacia donde esta la luz. Si al ir hacia la luz choca con un objeto quiere decir que en el punto donde estábamos mirando hay sombra.

De esta forma el color resultante es la si no hay objeto el color es Blinn-Phong, en caso contrario el color es la luz ambiental multiplicado por el k_a , para una única luz. Para más luces es la acumulación de la formula pero apuntando a cada una de las luces.

Calculo de reflexiones

Las reflexiones son un punto importante en esta practica ya que proporciona más realismo en la escena, porque cada objeto muestra una parte de otro objeto.

Las reflexiones se calculan una vez calculado sombras. Se necesita generar un nuevo rayo que vaya desde el punto actual del objeto hacia toda la escena, dependiendo del angulo de entrada del rayo primario al objeto. Para poder calcular este angulo se necesita la normal y el vector de salida del objeto (vector secundario) que es dos veces el angulo de entrada.

Una vez tenemos el vector dirección del de la reflexión, hay que hacer una llamada recursiva para calcular el color que refleja de este nuevo objeto en el anterior.

De esta forma la formula de color es la acumulación de todos los colores que reflejan en el objeto principal.

Opcionales y puntos implementados

Opcional

Como punto opcional hemos implementado mas de una luz, aunque son del mismo tipo (puntual) y un nuevo objeto que es de tipo triangulo.

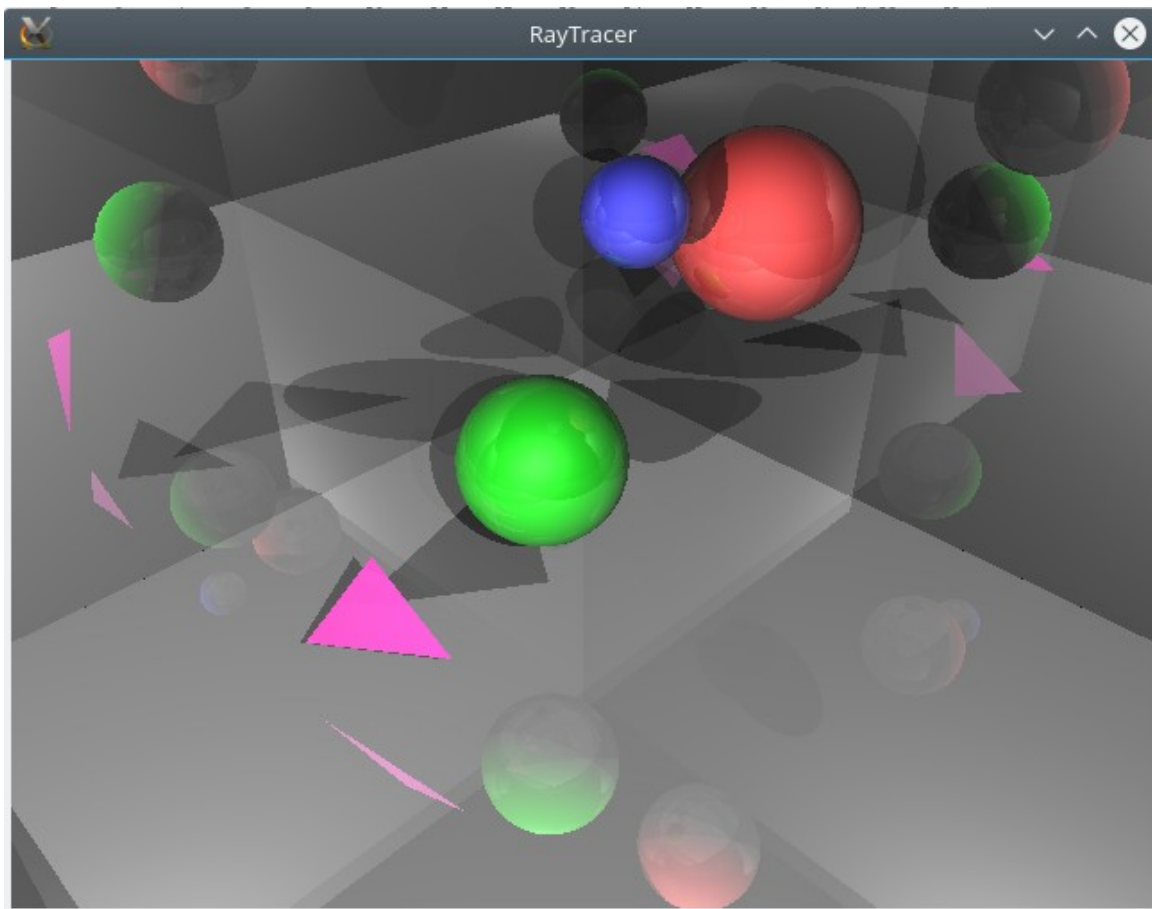
Puntos implementados

Los puntos implementados son:

- Funciones necesarias para calcular si un rayo intersecta con un objeto.
- Comprobar para todos los objetos en los que intersecta el rayo, cual es el más cercano al observador.
- Aplicar el zNear y zFar del mundo y la generación del rayo.
- Aplicar sombras a partir del rayo de luz.
- Aplicar reflexiones a partir del rayo secundario.
- Añadir más de una luz.
- Añadir un nuevo objeto de tipo triángulo.

Resultado y tomas falsas

Resultado



Tomas falsas

