

# Pràctica Gràfics 2 (curs 2015/16)

---

## Whitted Ray Tracer

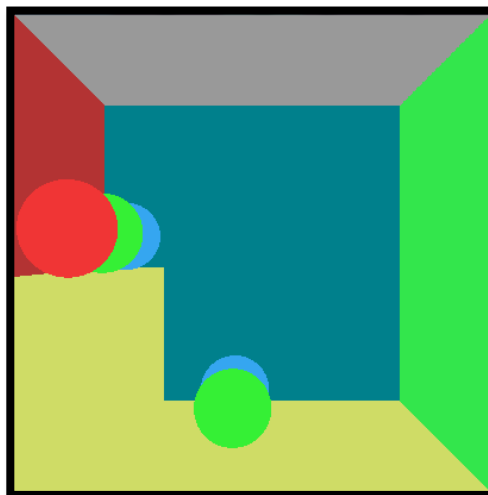
Whitted Ray Tracing és un algorisme per generar imatges 2D a partir d'una escena 3D. En el nostre cas generarem una imatge partir d'un fitxer anomenat escena.sdf. L'algorisme Whitted Ray Tracer treballa amb reflexions i refraccions ideals, així com també llums puntuals (sense àrea).



## La pràctica

Modifica el programa "Ray Tracer GFX II" que se t'ha entregat tot omplint les funcions **Triangle::Intersect**, **Raytracer::Shade**. El programa està força documentat internament amb comentaris, així que no hauries de tenir massa problemes per entendre l'estructura del programa. La vostra versió ha d'implementar un Whitted Ray Tracer amb el model d'il·luminació de Phong (ambient, difusa i especular) i ombres. Ha de suportar varies fonts de llum puntuals i reflexions. El raytracer ha de ser capaç de renderitzar una escena amb triangles.

El codi està disponible a l'eStudy de l'assignatura. Tot el codi està escrit en C++ i es pot compilar i executar tal com s'entrega. Verifica immediatament que pots compilar i executar el programa i es mostra la imatge de més avall.



**Color Raytracer::Shade**( **const HitInfo** &hit, **const Scene** &scene, **int** max\_tree\_depth )

### Entrades

- **const HitInfo** &hit

Rebem l'objecte **HitInfo**, que emmagatzema la geometria del raig que ha col·lisionat amb la superfície i el material de la mateixa.

- **const Scene** &scene

Rebem l'escena on es descriu l'escena global, incloent les llums.

- **int** max\_tree\_depth

Rebem també un enter que delimita el nombre de rebots que ha de fer el raig de llum indirecte per tal d'evitar una recursió infinita.

### Sortida

Tornarem un **Color** que ha de tenir tan la component directe com la indirecta.

### Pseudocodi

```
1 Color = negre
2
3 Si hi ha intersecció entre ray i l'escena
4     Selecciona el punt d'intersecció més proper -> inter
5     Calcula el raig amb origen a inter i que passa per la posició de la llum -> light
6
7     Si NO hi ha intersecció entre light i l'escena
8         Color = Calcula llum directa
9     fisi
10
11     Si el material és reflectant
12         Calcula el raig amb origen a inter i en la direcció de reflexió ideal -> reflected
13         Color += Calcula la llum reflectada
14     fisi
15 fisi
16
17 Retorna Color
```

## Funcions implementades que haureu d'utilitzar

**Color** Trace(**const Ray** &ray, **const Scene** &scene, **int** max\_tree\_depth);

Trace es una funció que torna la llum indirecte d'un punt cap a una direcció de reflexió descrita en el raig.

**int** Cast( **const Ray** &ray, **const Scene** &scene, **HitInfo** &hitinfo, **Object** \*ignore )

Cast és una funció que busca el primer punt d'intersecció (si n'hi ha algun) entre el raig i la llista d'objectes de l'escena. Retorna **true** (si xoca contra algun objecte) o **false** en cas contrari.

**bool Triangle::Intersect( const Ray &ray, HitGeom &hitgeom ) const;**

### Entrades

- **const HitGeom &hitgeom**

Rebem l'objecte **HitGeom**, que és buit. L'haurem d'omplir quan trobem una intersecció.

- **const Ray &ray**

Rebem informació del raig, com la direcció o l'origen.

### Sortida

Tornarem un **bool** que ha de valdre **true** en cas que la intersecció es produeixi dins el triangle o **false** en cas contrari.

### Pseudocodi

```
1 Defineix el pla del triangle
2 Calcula la distancia entre l'origen del raig i el pla -> dist
3
4 Si (dist > 0) && (dist < distancia de xoc)
5     Busca el punt P del pla que conté el triangle
6     Projecta el les coordenades del punt a un pla 2D en funció de l'eix dominant del triangle
7     Calcula les coordenades baricéntriques
8
9     Si totes les coordenades barixentriques són entre 0 i 1
10         Emplenem els camps de hitgeom
11         Retorna true
12     fisi
13
14 fisi
15 Retorna false
```

## Ajuda amb baricéntriques

Si necessiteu invertir una matriu, podeu utilitzar aquesta sentència:

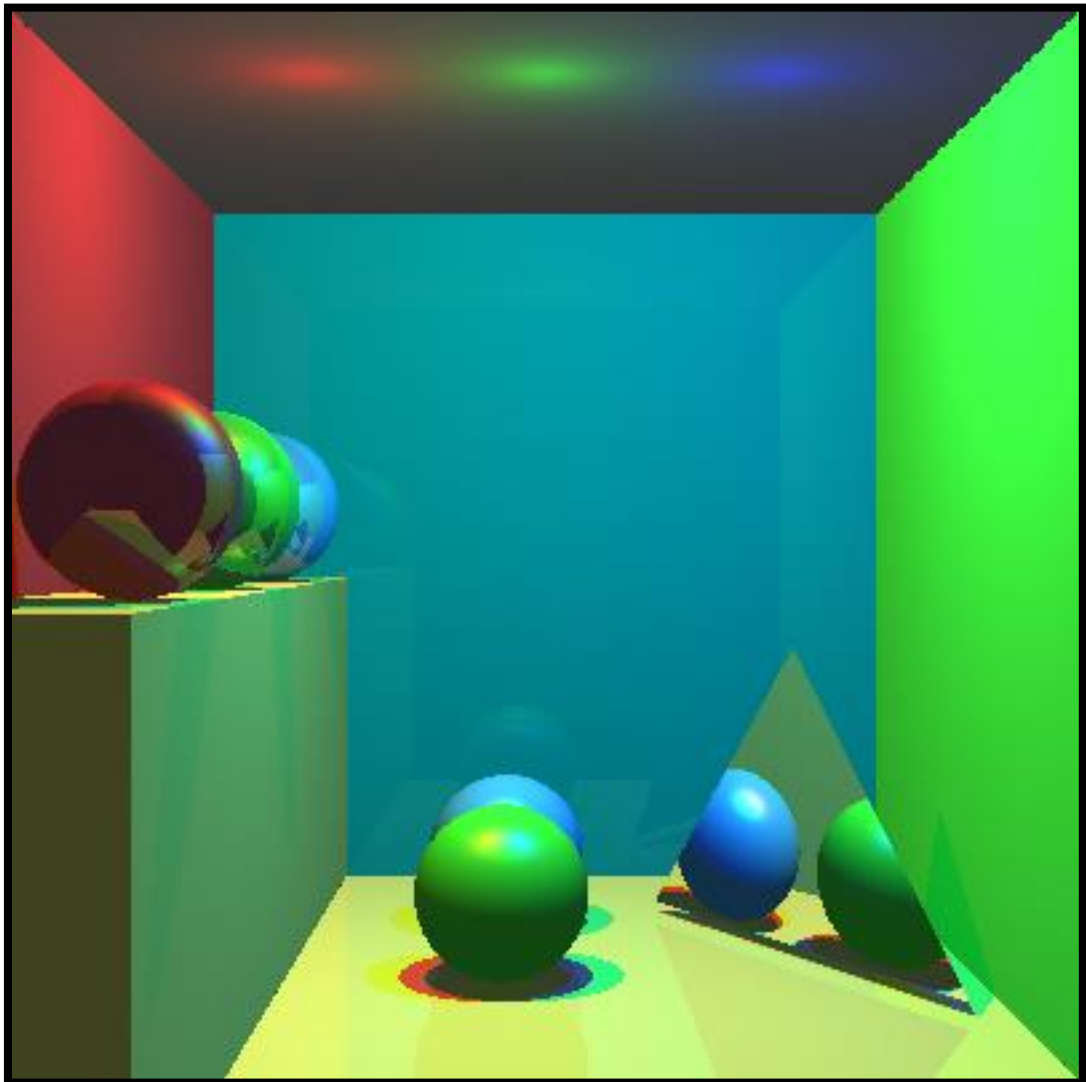
$$\text{Minv} = ( 1 / \det( M ) ) * \text{Transpose}( \text{Adjoint}( M ) );$$

## Com visualitzar el resultat

Quan executeu el codi veureu que es crea una finestra on es visualitza el resultat, però a més a més, es generarà un arxiu d'imatge PPM. Necessites un visor d'imatges per veure el PPM, per exemple en Windows pots utilitzar "IrfanView" que és un programa gratuït o "Photoshop".

[http://www.irfanview.com/main\\_download\\_engl.htm](http://www.irfanview.com/main_download_engl.htm)

El resultat final hauria de ser aquest:



## Entrega

S'ha de lliurar el codi font amb projecte de Visual Studio i el fitxer de sortida (Resultat.PPM) generat. El lliurement s'ha de realitzar a través del pou pertinent de l'eStudy. Crea un únic fitxer ZIP amb tots els continguts de l'entrega.

L'entrega s'ha de realitzar com a molt tard el **diumenge 6 de març de 2016**. Aquesta pràctica s'ha de realitzar en grups de **dues persones**.