

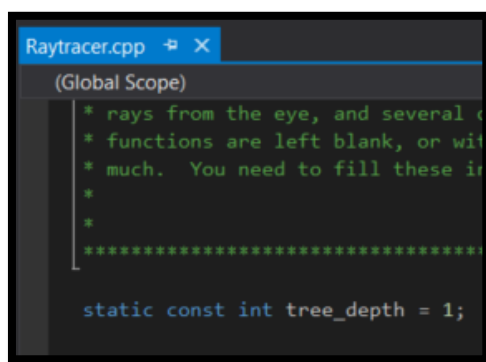
Pràctica Gràfics 2 (curs 2015/16)

Consideracions inicials

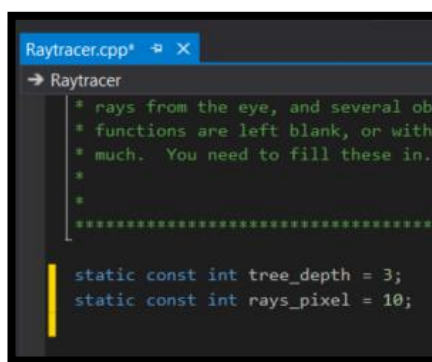
L'objectiu d'aquesta pràctica és modificar el codi de l'entrega anterior per tal d'aconseguir 3 objectius. Cada objectiu que assolis contarà de manera separada i el resultat final variarà.

Abans de començar a anomenar els objectius heu de modificar una petita part del codi que us varem proporcionar a l'enunciat de la pràctica anterior. Dins de la funció Raytracer.cpp heu d'afegir una nova variable global (rays_pixel).

```
static const int rays_pixel = 10;
```



```
Raytracer.cpp [x]
(Global Scope)
* rays from the eye, and several o
* functions are left blank, or wit
* much. You need to fill these in
*
*
*****
static const int tree_depth = 1;
```

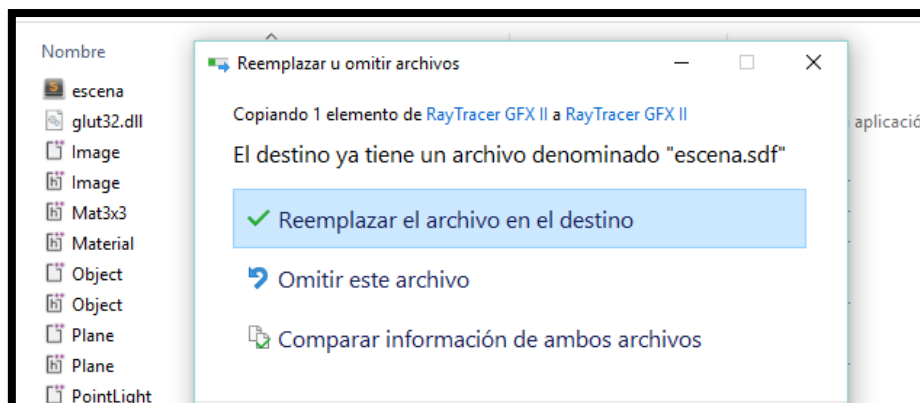


```
Raytracer.cpp* [x]
→ Raytracer
* rays from the eye, and several ob
* functions are left blank, or with
* much. You need to fill these in.
*
*
*****
static const int tree_depth = 3;
static const int rays_pixel = 10;
```

Noteu que la variable tree_depth ara val 3. Les imatges d'exemple que es mostren en aquest document s'han generat utilitzant aquest valor.

A més heu de canviar tot el codi de la funció cast_line definida a Raytracer.cpp per un de nou que us hem adjuntat (cast_line.cpp). Per canviar-lo simplement obriu la funció amb un editor i sobreescriu la part del codi de Raytracer.cpp que defineix la funció cast_line.

Finalment l'escena no serà la mateixa que la de la pràctica anterior, per això l'haureu de sobreescriure l'escena anterior per la nova.



Tots aquets nous arxius excepte les variables globals (que podeu fer Copy-Paste d'aquest pdf) són a l'eStudy, juntament amb l'enunciat de la pràctica.

A continuació es mostren els enunciats de cadascun dels objectius. Estan dissenyats per que es resolguin de forma incremental, es a dir, començar per el primer, després el segon i finalment el tercer. També s'indica la puntuació de cada un dels objectius entre parèntesis.

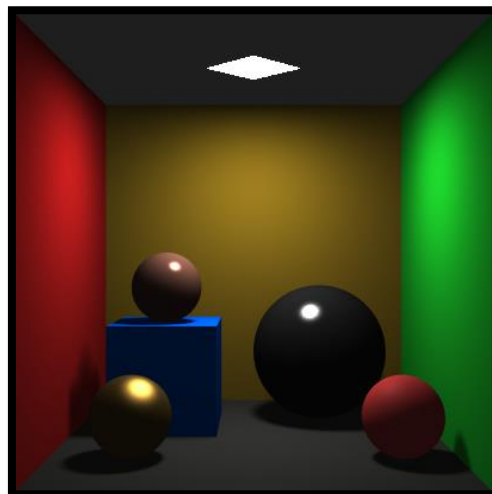
Objectiu 1 (4p). Il·luminació directa

En aquest primer objectiu haureu de calcular la il·luminació directa (difusa i especular) utilitzant el model de reflectivitat de Phong. Per extraure un punt d'una superfície que emet llum heu de cridar a la funció "GetSample" de l'objecte en qüestió. En l'escena d'exemple les fonts de llum són triangles. La seva funció GetSample està buida, per tant s'haurà d'implementar.

Pseudocodi de la funció Shade

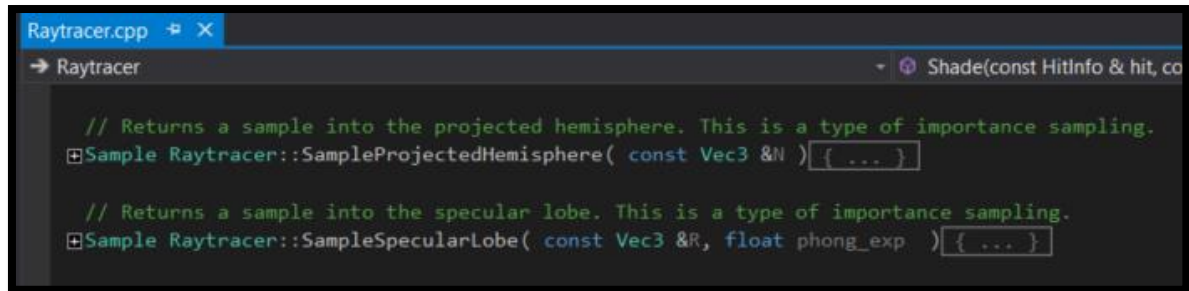
```
1 Si hem xocat amb una llum
2     retornem el color que emet
3 Sino
4     Per tots els objectes de l'escena
5         Si l'objecte emet llum
6             Extreiem un punt de llum mitjançant obj->GetSample.
7             Construïm un raig cap aquest punt.
8             Si no hi ha intersecció entre raig-punt.
9                 Càlcul de irradiància
10                Càlcul del component difús.
11                Càlcul del component especular.
12                directa += (difusa + especular) * irradiància;
13         Fisi
14     Fisi
15 Fiper
16     retornem el color directe + Ambient*kd
17 Fisi
```

Resultat



Objectiu 2 (4p). Il·luminació indirecta

En el segon objectiu afegirem la component indirecta. Haurem de crear dues funcions anomenades `SampleProjectedHemisphere` i `SampleSpecularLobe` que teniu definides en el `Raytracer.h` pero que no han estat implementades en el `Raytracer.cpp`.



```
Raytracer.cpp  + - x
→ Raytracer  Shade(const HitInfo & hit, co

// Returns a sample into the projected hemisphere. This is a type of importance sampling.
Sample Raytracer::SampleProjectedHemisphere( const Vec3 &N ) { ... }

// Returns a sample into the specular lobe. This is a type of importance sampling.
Sample Raytracer::SampleSpecularLobe( const Vec3 &R, float phong_exp ) { ... }
```

Pseudocodi de la funció Shade

```
1 Si hem xocat amb una llum
2   retornem el color que emet
3 Sino
4   Per tots els objectes de l'escena
5     Si l'objecte emet llum
6       Extreiem un punt de llum mitjançant obj->GetSample.
7       Construïm un raig cap aquest punt.
8       Si no hi ha intersecció entre raig-punt.
9         Càlcul de irradiància
10        Càlcul del component difús.
11        Càlcul del component especular.
12        directa += (difusa + especular) * irradiància;
13      Fisi
14    Fisi
15  Fiper
16
17  Extreure una mostra del hemisferi projectat
18  Construïm un raig amb direcció a la mostra
19  Càlcul reflexió difusa.
20
21  Extreure una mostra del lòbul especular
22  Construïm un raig amb direcció a la mostra
23  Càlcul reflexió especular.
24
25  indirecte = reflexió difusa + reflexió especular
26
27  retornem el color directe + indirecte
28 Fisi
```

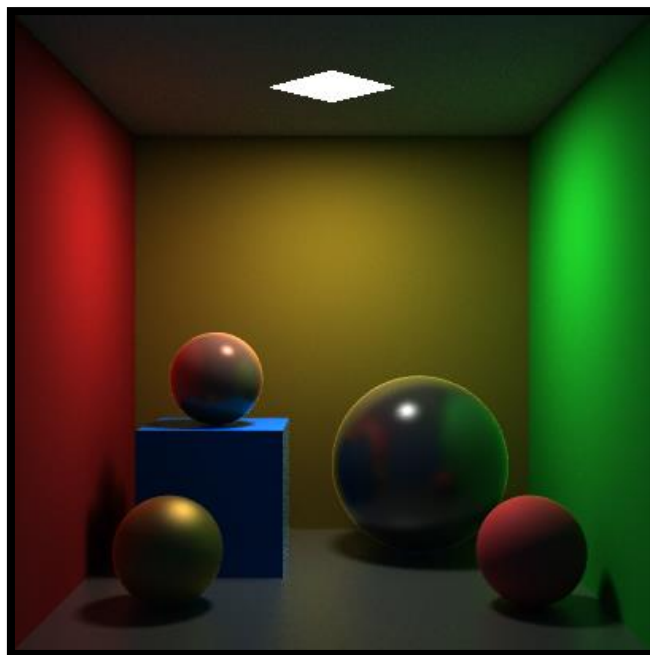
Resultat



Objectiu 3 (2p). Model de reflectivitat Torrance-Sparrow

Finalment, l'últim objectiu es basarà en modificar la il·luminació especular directa i la il·luminació especular indirecta per tal d'utilitzar el model basat en micro-estructures de Torrance-Sparrow.

Resultat



Entrega

S'ha de lliurar el codi font amb projecte de Visual Studio i el fitxer de sortida 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 17 d'abril de 2016**. Aquesta pràctica es pot realitzar en grups de **dues persones**.