Projet de lancer de rayon – Phase 1

La première tâche est de lire et comprendre ce qui se passe dans le code.

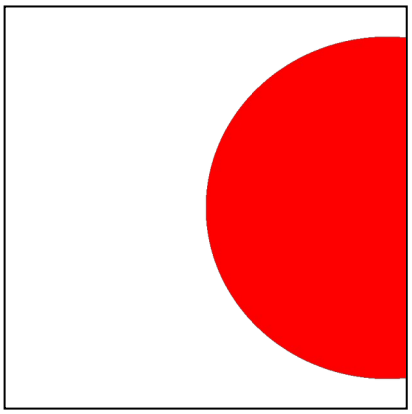
Pour la première partie de ce projet vous devez remplir les fonctions suivantes dans Scene.h, Square.h et Sphere.h :

1. `rayTrace ()`
   Cette fonction est appelée pour chaque rayon et devrait calculer l'intersection avec tous les objets de votre scène. Initialement, (cela changera par la suite) elle devra retourner la couleur de l'objet touché le plus proche. La couleur se trouve dans meshes[i].material.diffuse_material pour l'objet i. Notez que vous pouvez utiliser (ce n'est pas le cas actuellement) objectIndex dans RaySceneIntersection pour enregistrer et retrouver l'objet touché correspond à chaque RaySceneIntersection.
2. Intersect () dans Sphere.h
   Calcule l'intersection d'un rayon avec une sphère
3. Intersect () dans Square.h
   Calcul l'intersection d'un rayon avec un carré
4. `computeIntersection()`
   Remplir le résultat de RaySceneIntersection avec l'intersection la plus proche

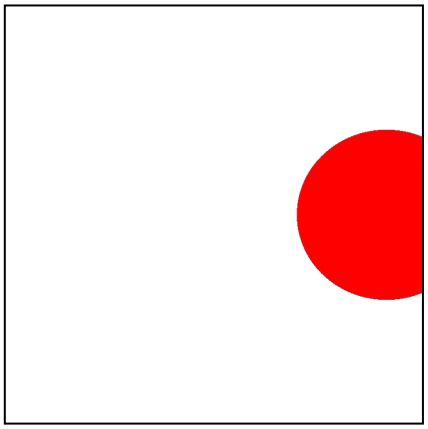
Voici quelques suggestions d'étapes à réaliser pour s'assurer du bon fonctionnement du projet.

Commencez par la scène avec une seule sphère. Dans ce cas, la fonction computeIntersection() reste simple pour l'instant, vous pouvez donc vous concentrer sur les fonctions intersect. Si elles fonctionnent correctement, vous devez obtenir un cercle rouge au centre de votre image.

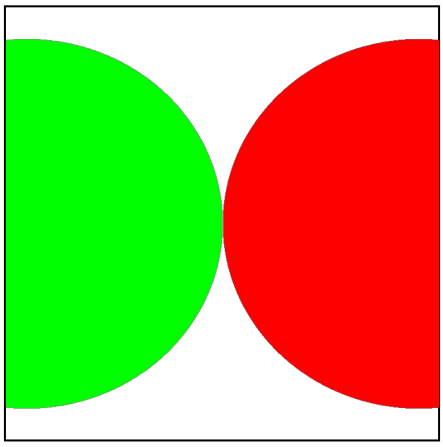
- Changer la position initiale de la sphère en changeant son centre de position à vec3(1.0, 0.0, 0.0)). Vous devez obtenir l'image suivante :



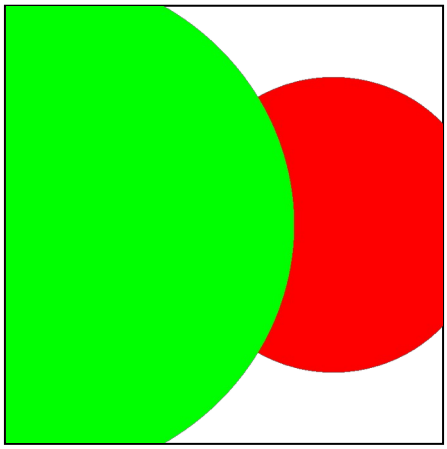
- Changée maintenant à vec3(1.0, 0.0, 0.0) et radius 0.5. La sortie doit ressembler à l'image suivante :



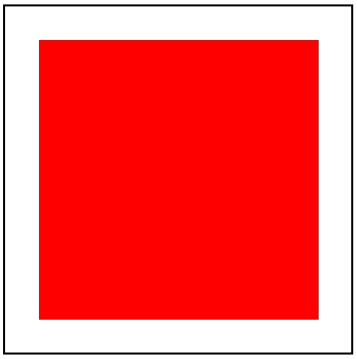
- Ajouter une second sphère (de couleur verte), appliquer une translation à l'une d'entre elles center (1.0, 0.0, 0.0) et la deuxième center (1.0, 0.0, 0.0). Votre résultat doit ressembler à cela :



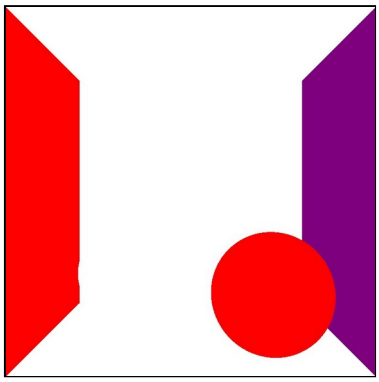
- Bougez, tournez :



- Passez à la scène représentant un carré. Maintenant vous devez écrire le code d'intersection d'un carré unité sur un plan XYplane avec Z=0. C'est ce que Square::intersect doit faire (Calculer l'intersection d'une droite et un plan ensuite limiter au carré unité). Maintenant mettre à jour Square::intersect() comme pour la sphère. Vous devriez maintenant avoir quelque chose comme l'image suivante (si vous changez la couleur du carré à rouge). Notez que les bords du carré ne vont pas jusqu'au bord de l'image.



- Bougez, tournez, comme la sphère, cela devrait fonctionner de la même façon.
- Maintenant passez à la scène 2 et VOILA une boite de cornell apparaît.



En changeant les couleurs :