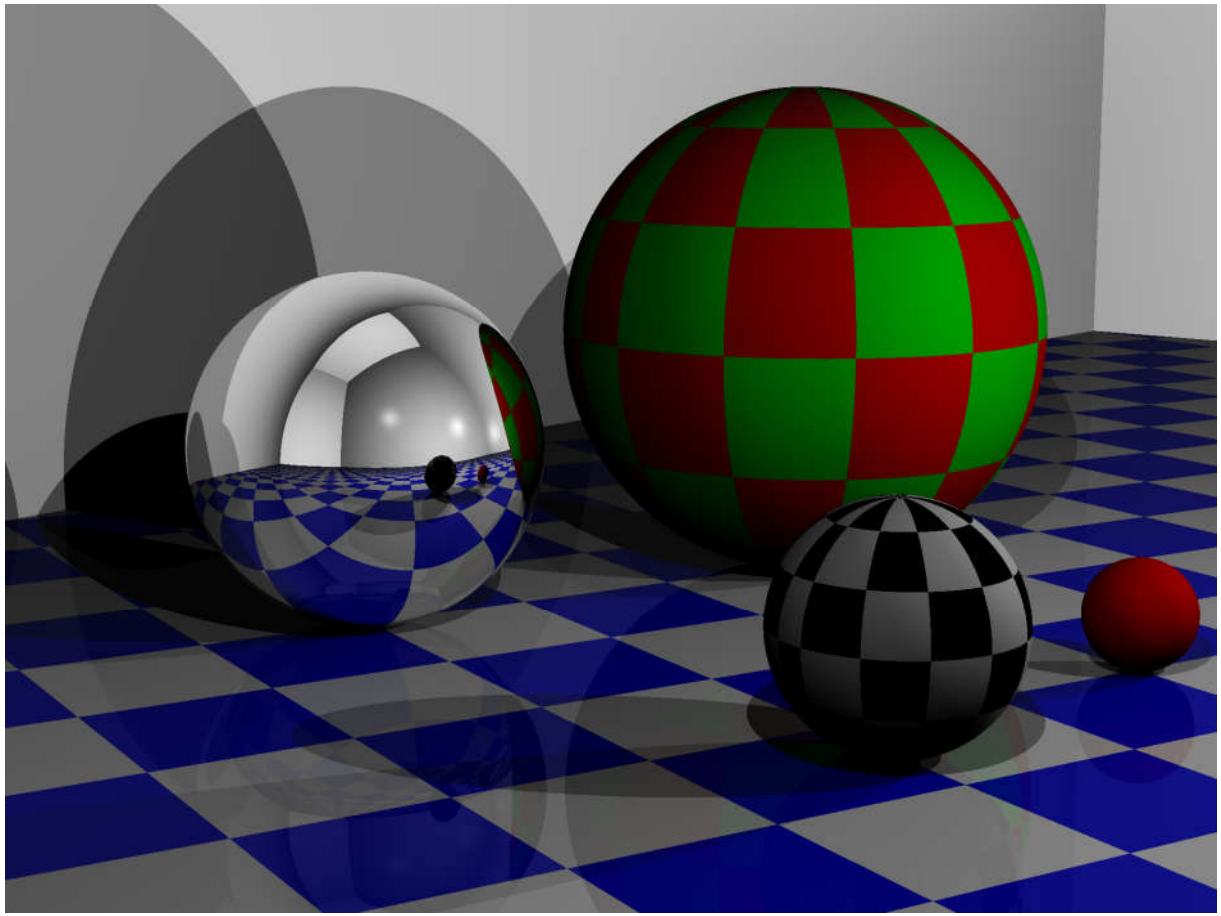
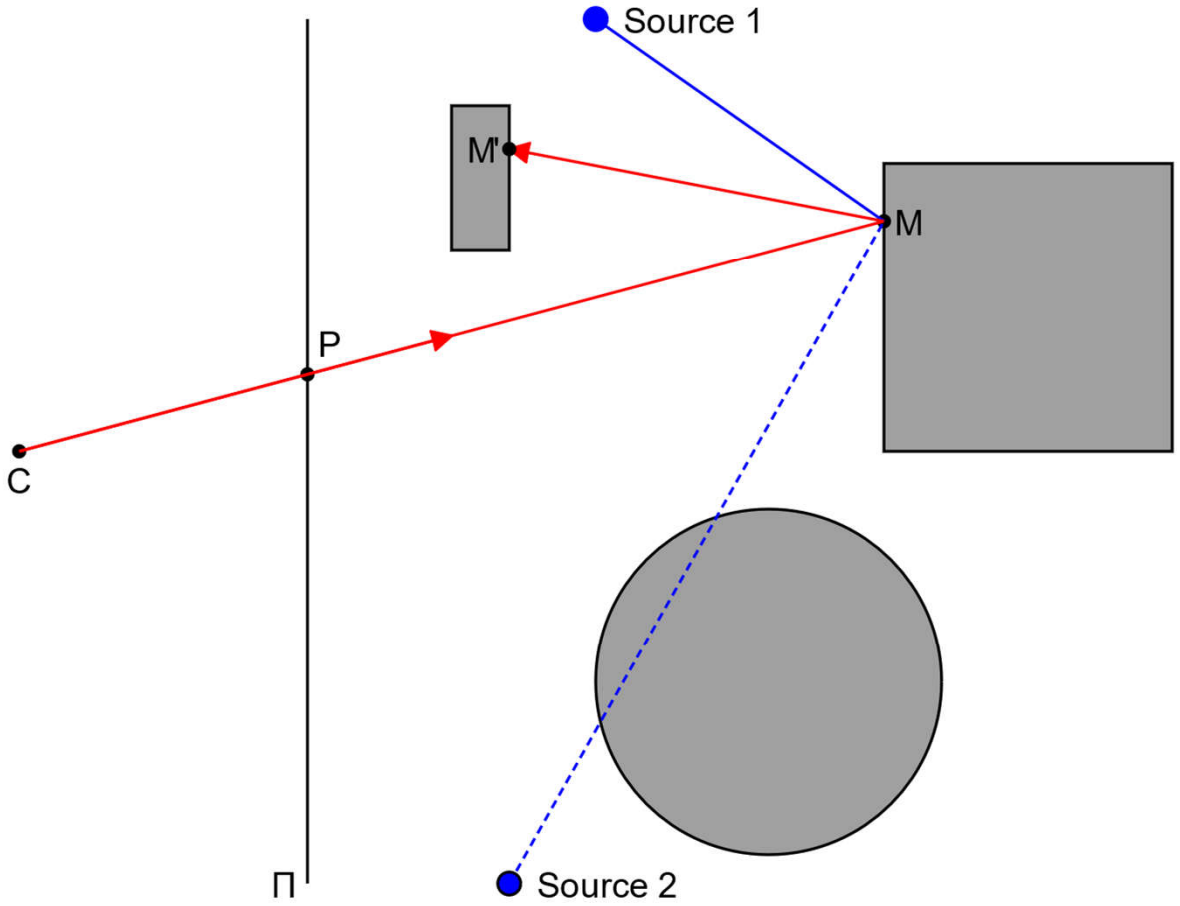
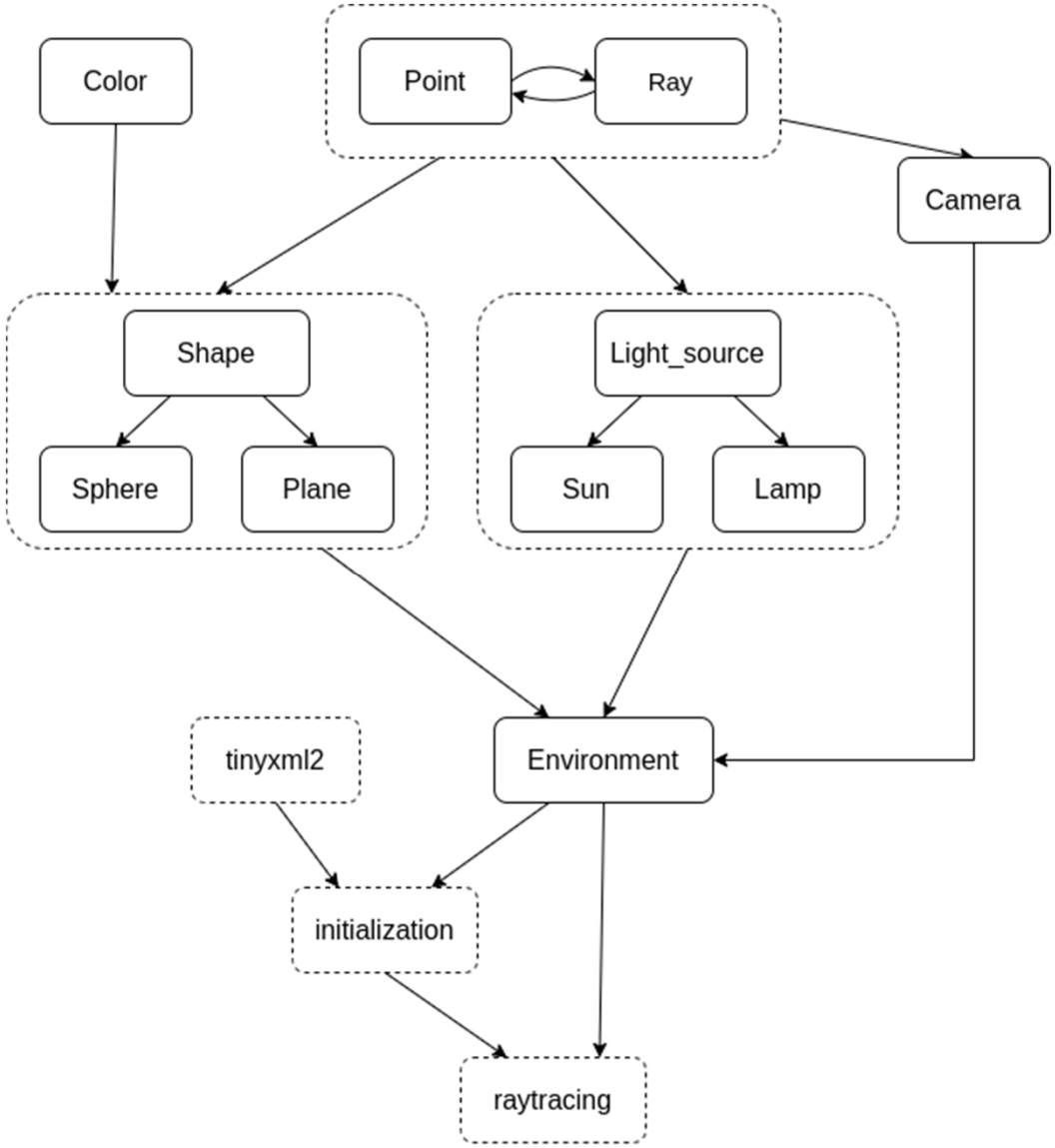
IN204
Lancer de rayons orienté objet



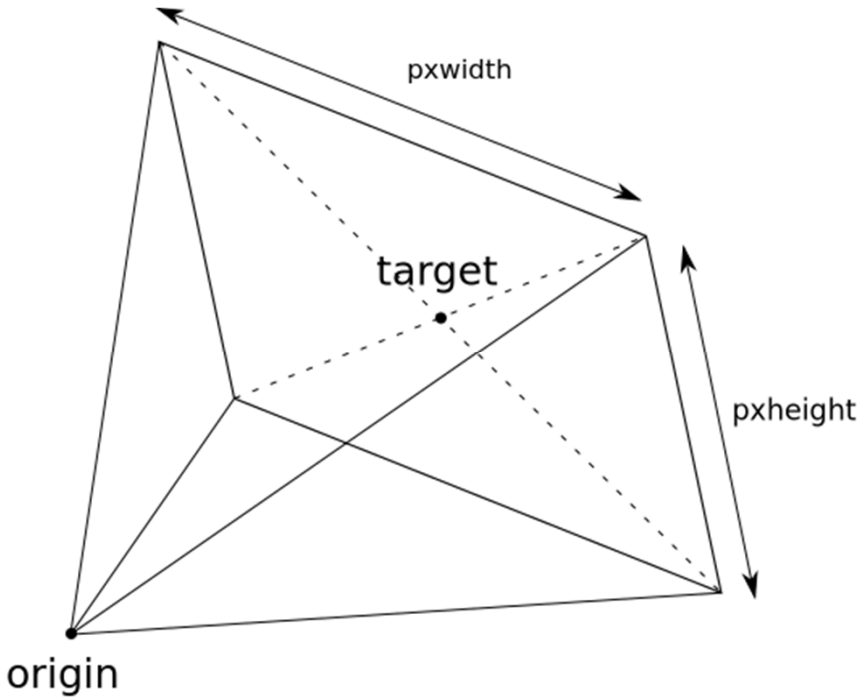
Description de l’algorithme



Structure du projet



La classe Camera



Elle est constituée des champs :

Point origin, target

float pxwidth, pxheight

float width, height

La classe Environment

Cette classe comporte les méthodes suivantes :

int find_first_intersect(const Ray& r, Point& I) const

float lighting(const Point&, unsigned) const

Ray ray_from_pixel(unsigned x, unsigned y) const

Color<float> recursive_color_from_ray(Ray r, float coeff, unsigned counter) const

void recursive_raytracing() const

Color<float> color_from_ray(Ray r) const et void raytracing() const

Initialisation avec TinyXML2

```
<world>
  <camera pxwidth="800" pxheight="800" width="4">
    <origin x="0" y="0" z="0"/>
    <target x="0" y="7" z="0"/>
  </camera>
  <lights>
    <lamp brightness="1">
      <point x="0" y="7" z="2"/>
    </lamp>
    <sun brightness="2">
      <point x="30" y="4" z="-10"/>
    </sun>
  </lights>
  <shapes>
    <sphere color="white" gloss="0" size="0.3" chess="default">
      <point x="0" y="10" z="0"/>
    </sphere>

    <plane color="blue" gloss="0.5">
      <origin x="1.4" y="0" z="0"/>
      <normal x="-10" y="-1" z="0"/>
    </plane>
    <plane color="red" gloss="0" chess="white">
      <origin x="0" y="0" z="-4"/>
      <normal x="0" y="0" z="1"/>
    </plane>
  </shapes>
</world>
```

Caméra

Lampe

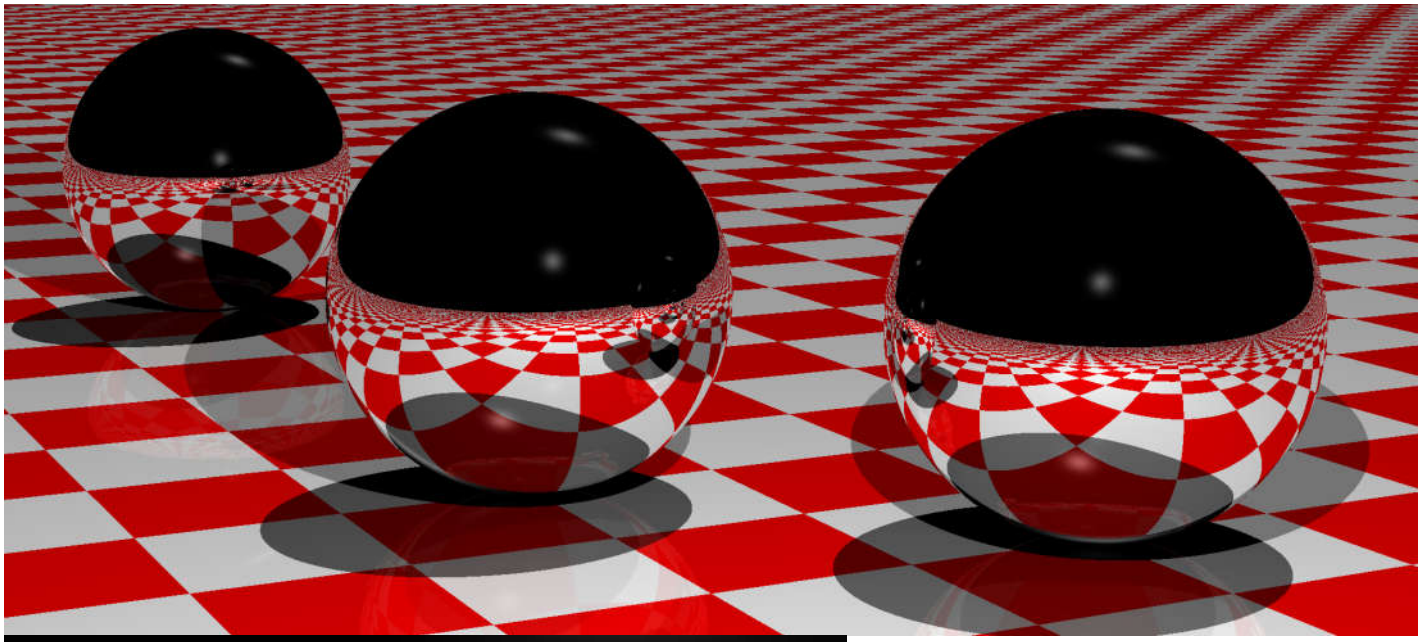
Soleil

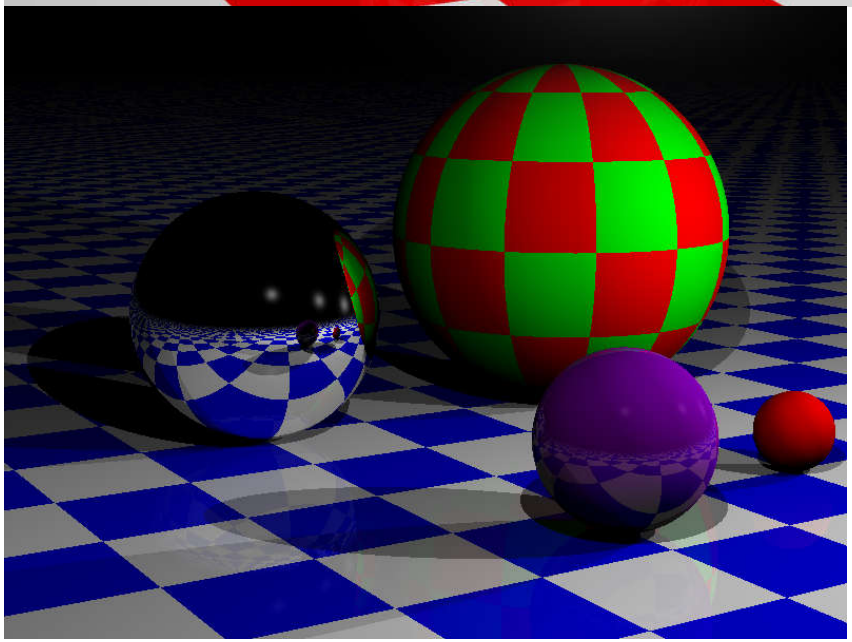
Sphère

Plan

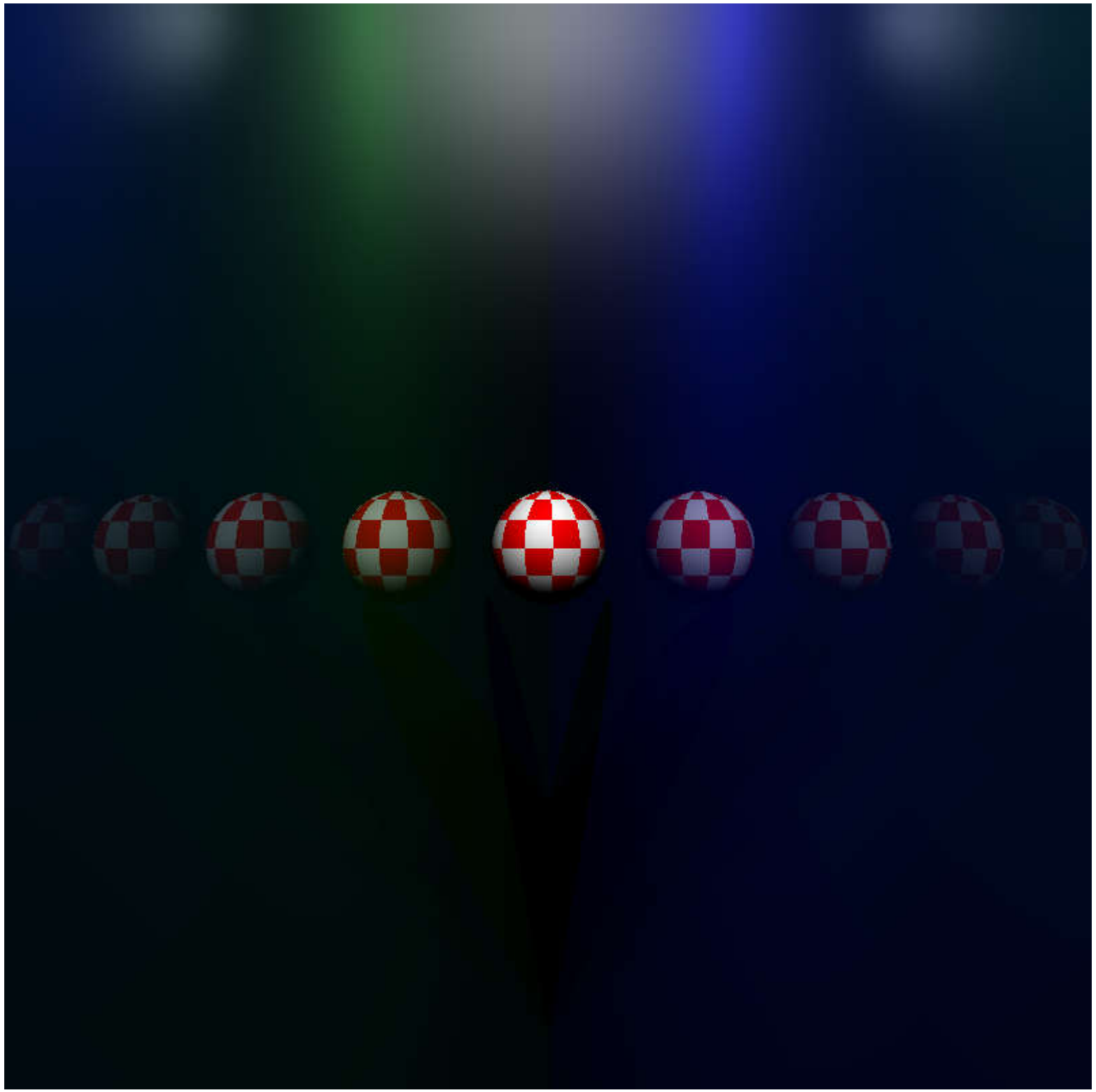
Plan

Quelques rendus





Quelques rendus



Améliorations possibles

- Autres textures et formes
- Optimisation des ressources en temps (et mémoire ?)
- Possibilité de spécifier la couleur d'un objet en RGB ou par son nom
- Calcul de l'éclairage en prenant en compte les réflexions
- Lumière diffuse pour une lampe
- Orientation automatique de l'objet « plan infini »
- Possibilité de créer des objets transparents (compliqué, nécessite un lancer de rayons bidirectionnel)