Ray Tracing

Proyecto: Luciérnagas en una botella

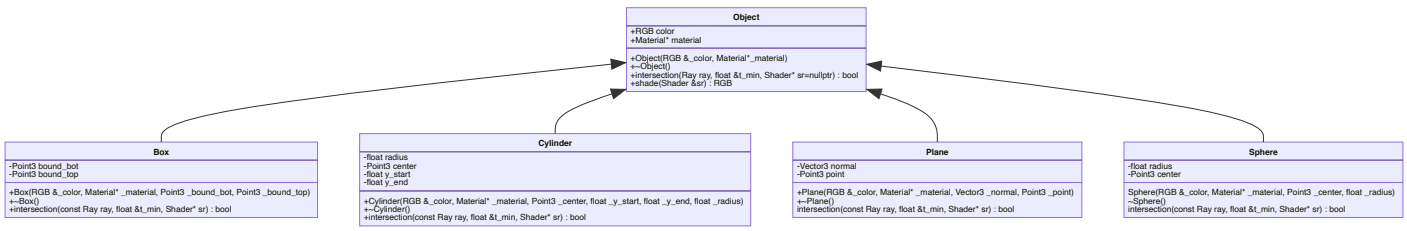
UTEC-CS2501 2021-1

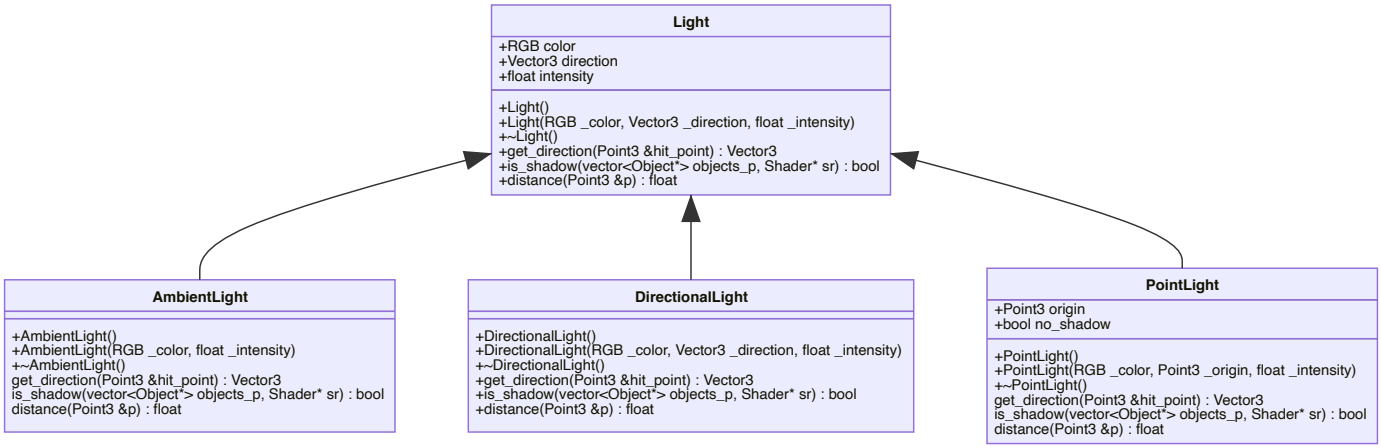
Nombre: Andrea Velásquez Gushiken
Tema: Ray Tracing Basics

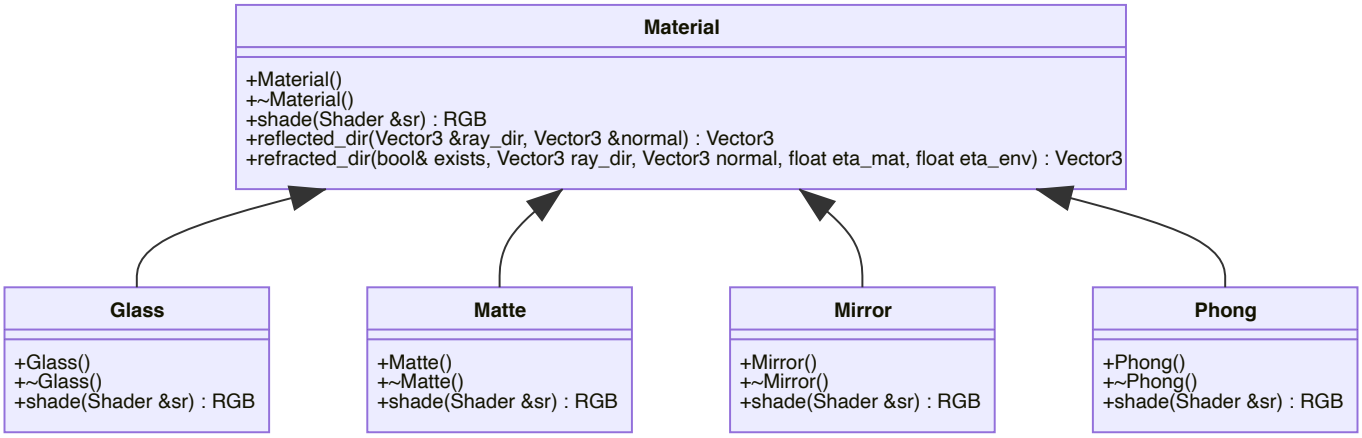
Descripción del proyecto

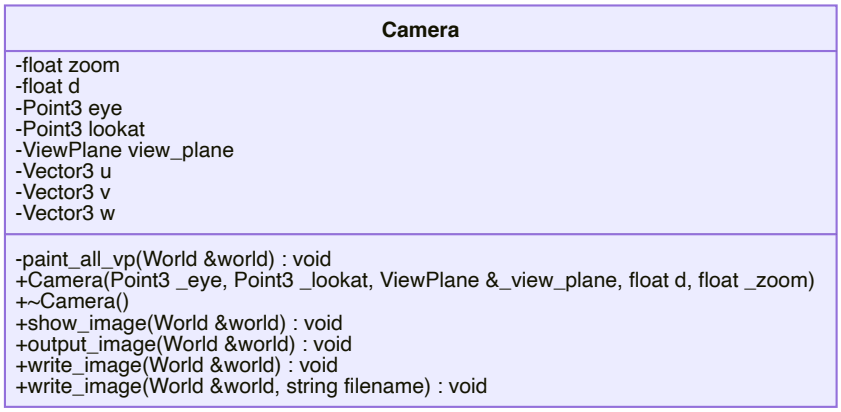
Se presenta un cilindro blanco de vidrio (botella) con esferas luminosas amarillas (luciérnagas). Ambos objetos fueron construidos luego de implementar los fenómenos de reflexión, refracción y luz especular y difusa con la técnica Ray Tracing.

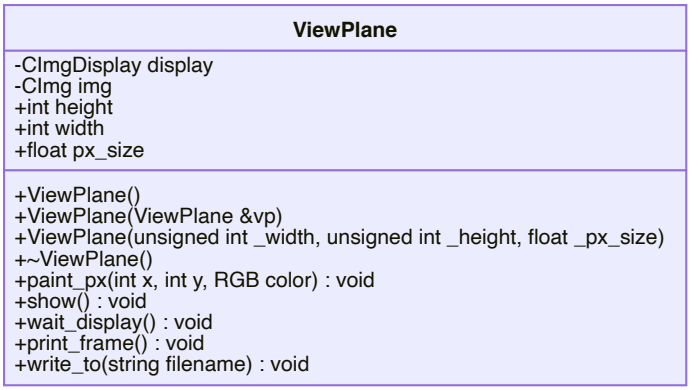
Diseño de clases

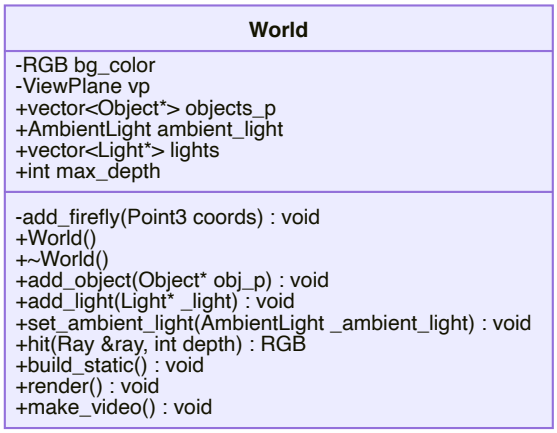


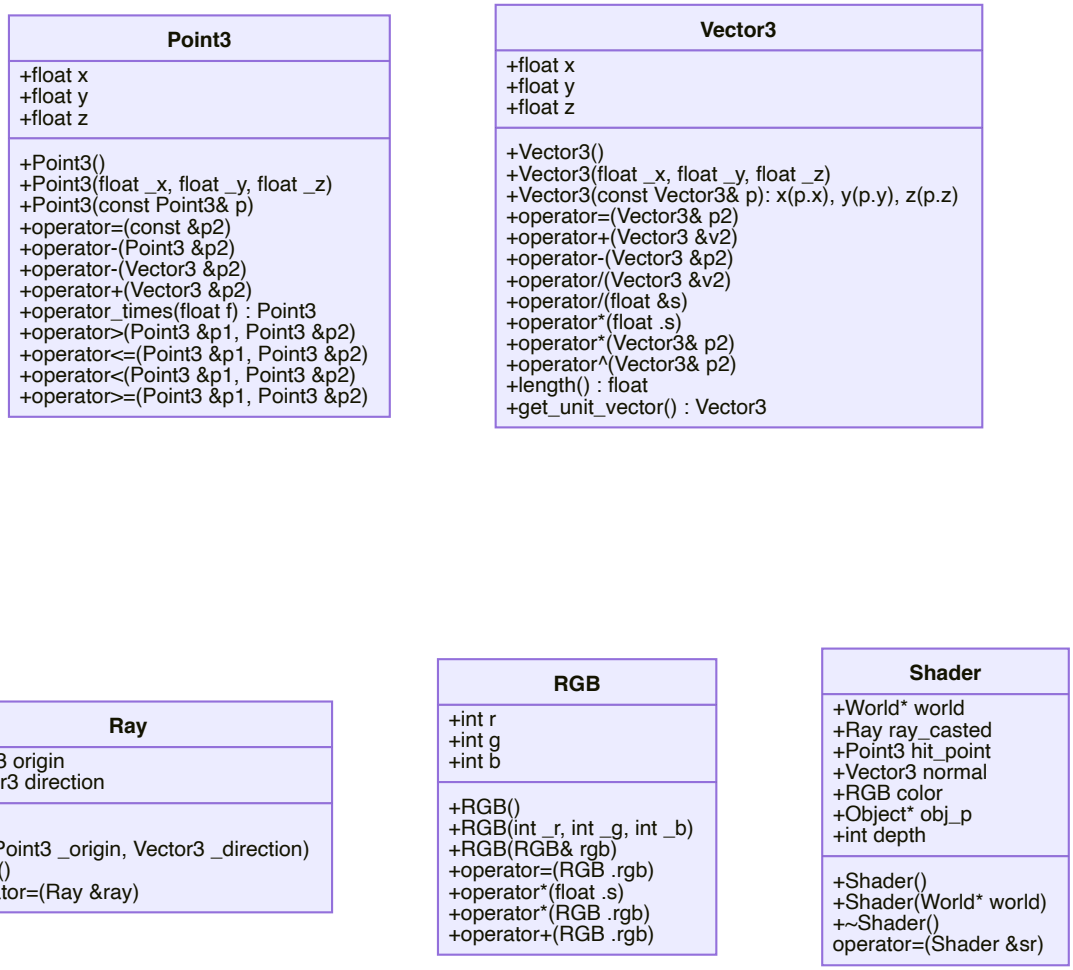

Light
+RGB color
+Vector3 direction
+float intensity
+Light()
+Light(RGB _color, Vector3 _direction, float _intensity)
+~Light()
+get_direction(Point3 &hit_point) : Vector3
+is_shadow(vector<Object*> objects_p, Shader* sr) : bool
+distance(Point3 &p) : float
AmbientLight
+AmbientLight()
+AmbientLight(RGB _color, float _intensity)
+~AmbientLight()
get_direction(Point3 &hit_point) : Vector3
is_shadow(vector<Object*> objects_p, Shader* sr) : bool
distance(Point3 &p) : float
DirectionalLight
+DirectionalLight()
+DirectionalLight(RGB _color, Vector3 _direction, float _intensity)
+~DirectionalLight()
+get_direction(Point3 &hit_point) : Vector3
+is_shadow(vector<Object*> objects_p, Shader* sr) : bool
+distance(Point3 &p) : float
PointLight
+Point3 origin
+bool no_shadow
+PointLight()
+PointLight(RGB _color, Point3 _origin, float _intensity)
+~PointLight()
get_direction(Point3 &hit_point) : Vector3
is_shadow(vector<Object*> objects_p, Shader* sr) : bool
distance(Point3 &p) : float


Material
+Material()
+~Material()
+shade(Shader &sr) : RGB
+reflected_dir(Vector3 &ray_dir, Vector3 &normal) : Vector3
+refracted_dir(bool& exists, Vector3 ray_dir, Vector3 normal, float eta_mat, float eta_env) : Vector3
Glass
+Glass()
+~Glass()
+shade(Shader &sr) : RGB
Matte
+Matte()
+~Matte()
+shade(Shader &sr) : RGB
Mirror
+Mirror()
+~Mirror()
+shade(Shader &sr) : RGB
Phong
+Phong()
+~Phong()
+shade(Shader &sr) : RGB


Camera
-float zoom
-float d
-Point3 eye
-Point3 lookat
-ViewPlane view_plane
-Vector3 u
-Vector3 v
-Vector3 w
-paint_all_vp(World &world) : void
+Camera(Point3 _eye, Point3 _lookat, ViewPlane &_view_plane, float d, float _zoom)
+~Camera()
+show_image(World &world) : void
+output_image(World &world) : void
+write_image(World &world) : void
+write_image(World &world, string filename) : void


ViewPlane
-CImgDisplay display
-CImg img
+int height
+int width
+float px_size
+ViewPlane()
+ViewPlane(ViewPlane &vp)
+ViewPlane(unsigned int _width, unsigned int _height, float _px_size)
+~ViewPlane()
+paint_px(int x, int y, RGB color) : void
+show() : void
+wait_display() : void
+print_frame() : void
+write_to(string filename) : void


World
-RGB bg_color
-ViewPlane vp
+vector<Object*> objects_p
+AmbientLight ambient_light
+vector<Light*> lights
+int max_depth
-add_firefly(Point3 coords) : void
+World()
+~World()
+add_object(Object* obj_p) : void
+add_light(Light* _light) : void
+set_ambient_light(AmbientLight _ambient_light) : void
+hit(Ray &ray, int depth) : RGB
+build_static() : void
+render() : void
+make_video() : void



Capturas de pantalla