Start

End

Initialize

Generate original ray of light

Generate refraction light

Calculate intersection point

Calculate diffuse color

Reach maximum recursion depth?

Reflect?

Y

N

Y

Y

Refract?

Generate reflection light

Add to base color

N

Material

*Color getKd()*

*void setKd(Color kd)*

*Color getKs()*

*void setKs(Color ks)*

*Color getKt()*

*void setKt(Color kt)*

*float getN()*

*void setN(float n)*

*int getShiness()*

*void setShiness(int shiness)*

Material()

Material(Color kd, Color ks, Color kt)

Material(Color kd, Color ks, Color kt, float n)

Material(Color kd, Color ks, int shiness)

N

Scene

*int getPrimitiveCount()*

*int getLightCount()*

*Primitive getPrimitive(int index)*

*LightSource getLight(int index)*

*void loadScene()*

Scene()

Primitive

*Vector3 getNormal(Point p)*

*Intersection intersect(Ray ray)*

*AABB getAABB()*

*Type getType()*

*void setMaterial(Material mat)*

*Material getMaterial()*

Primitive()

Primitive(Material material)

A

Reflective surface

Camera Position

LightSource

*Color getLightColor()*

*Point getLightPosition()*

*Vector3 getLightDirection(Point p)*

*void setLightPosition(Point pos)*

LightSource()

Renderer

*void render()*

*void initRenderArea()*

*void initCoordinate()*

*void init()*

Renderer()

AABB

*Point getPosition()*

*Vector3 getSize()*

*bool intersect(AABB b)*

AABB()

AABB(Point pos, Vector3 size)

Canvas

*Pixel getPixel()*

*void setPixel(Pixel pixel, int x, int y)*

*int setHeight()*

*int setWidth()*

*void fillColor(Color color)*

*void setPixelColor(int x, int y, Color c)*

Canvas()

Canvas(int width, int height)

Camera

*Point getPosition()*

*void setPosition(Point pos)*

*Vector3 getForward()*

*void setForward(Vector3 forward)*

*Vector3 getUpward()*

*void setUpward(Vector3 upward)*

*float getWidth()*

*void setWidth(float width)*

*float getHeight()*

*void setHeight(float height)*

*float getDepth()*

*void setDepth(float depth)*

*void rotate(float angle, Vector3 axis)*

Camera()

Ray

*Point getOrigin()*

*void setOrigin(Point origin)*

*Vector3 getDirection()*

*void setDirection(Vector3 direction)*

*float getDistance(Point p)*

*Point getPoint(float f)*

Ray(Point origin, Vector3 direction)

Vecter3

*void set()*

*void normalize()*

*float length()*

*float sqrLength()*

*float dot()*

*Vector3 cross()*

Vector3()

Vector3(float x, float y, float z)

Primitive

Ray

Vector3

LightSource

PointLight

ParallelLight

RayTracingRenderer

Material

Box

Plane

Sphere

Scene

Canvas

Camera

Renderer

*render()*

*initRenderArea()*

*initCoordinate()*

*init()*