

# Raytracer Project

0.1

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	gameobj Namespace Reference . . . . .	9
5.2	gameui Namespace Reference . . . . .	9
5.3	raytracer Namespace Reference . . . . .	9
5.3.1	Function Documentation . . . . .	10
5.3.1.1	applyAngle() . . . . .	10
5.3.1.2	closerToOrigin() . . . . .	10
5.3.1.3	genRays() . . . . .	10
5.3.1.4	getAngles() . . . . .	11
5.3.1.5	operator*() [1/3] . . . . .	11
5.3.1.6	operator*() [2/3] . . . . .	11
5.3.1.7	operator*() [3/3] . . . . .	11
5.3.1.8	renderFrame() . . . . .	11
5.3.1.9	rotateX() . . . . .	12
5.3.1.10	rotateY() . . . . .	12
5.3.1.11	rotateZ() . . . . .	12
5.3.1.12	vectorFromAngles() . . . . .	12
5.3.1.13	vectorFromPoints() . . . . .	12
5.3.2	Variable Documentation . . . . .	12
5.3.2.1	pi . . . . .	12

<b>6</b>	<b>Class Documentation</b>	<b>13</b>
6.1	gameobj::AmbientLight Class Reference	13
6.1.1	Constructor & Destructor Documentation	13
6.1.1.1	AmbientLight()	14
6.1.2	Member Function Documentation	14
6.1.2.1	getBrightness()	14
6.1.2.2	getColor()	14
6.1.2.3	interact()	14
6.1.2.4	setBrightness()	14
6.1.2.5	setColor()	14
6.1.3	Member Data Documentation	15
6.1.3.1	brightness_	15
6.1.3.2	color_	15
6.2	raytracer::Camera Class Reference	15
6.2.1	Constructor & Destructor Documentation	15
6.2.1.1	Camera() [1/2]	15
6.2.1.2	Camera() [2/2]	16
6.2.2	Member Function Documentation	16
6.2.2.1	getAng()	16
6.2.2.2	getPos()	16
6.2.2.3	getViewAngle()	16
6.2.3	Member Data Documentation	16
6.2.3.1	angleVect_	16
6.2.3.2	pos_	16
6.2.3.3	viewAngle_	17
6.3	raytracer::Color Class Reference	17
6.3.1	Constructor & Destructor Documentation	17
6.3.1.1	Color() [1/2]	18
6.3.1.2	Color() [2/2]	18
6.3.2	Member Function Documentation	18

6.3.2.1	<a href="#">getB()</a>	18
6.3.2.2	<a href="#">getG()</a>	18
6.3.2.3	<a href="#">getR()</a>	18
6.3.2.4	<a href="#">normalize()</a>	18
6.3.2.5	<a href="#">operator*()</a> [1/2]	18
6.3.2.6	<a href="#">operator*()</a> [2/2]	19
6.3.2.7	<a href="#">operator+()</a>	19
6.3.2.8	<a href="#">operator+=()</a>	19
6.3.2.9	<a href="#">operator-()</a>	19
6.3.2.10	<a href="#">setB()</a>	19
6.3.2.11	<a href="#">setG()</a>	19
6.3.2.12	<a href="#">setR()</a>	19
6.3.3	<a href="#">Friends And Related Function Documentation</a>	20
6.3.3.1	<a href="#">operator*</a>	20
6.3.4	<a href="#">Member Data Documentation</a>	20
6.3.4.1	<a href="#">b_</a>	20
6.3.4.2	<a href="#">g_</a>	20
6.3.4.3	<a href="#">r_</a>	20
6.4	<a href="#">gameobj::Cube Class Reference</a>	20
6.4.1	<a href="#">Constructor &amp; Destructor Documentation</a>	21
6.4.1.1	<a href="#">Cube()</a>	21
6.4.2	<a href="#">Member Function Documentation</a>	21
6.4.2.1	<a href="#">getAngles()</a>	21
6.4.2.2	<a href="#">getCenter()</a>	21
6.4.2.3	<a href="#">getColor()</a>	22
6.4.2.4	<a href="#">getLandmark()</a>	22
6.4.2.5	<a href="#">getSide()</a>	22
6.4.2.6	<a href="#">intersecte()</a>	22
6.4.3	<a href="#">Member Data Documentation</a>	22
6.4.3.1	<a href="#">angles_</a>	22

6.4.3.2	center_ . . . . .	22
6.4.3.3	landmark_ . . . . .	23
6.4.3.4	side_ . . . . .	23
6.4.3.5	squares_ . . . . .	23
6.5	gameobj::DirectionalLight Class Reference . . . . .	23
6.5.1	Constructor & Destructor Documentation . . . . .	24
6.5.1.1	DirectionalLight() . . . . .	24
6.5.2	Member Function Documentation . . . . .	24
6.5.2.1	getBrightness() . . . . .	24
6.5.2.2	getColor() . . . . .	24
6.5.2.3	getDir() . . . . .	24
6.5.2.4	interact() . . . . .	24
6.5.2.5	setBrightness() . . . . .	25
6.5.2.6	setColor() . . . . .	25
6.5.2.7	setDir() . . . . .	25
6.5.3	Member Data Documentation . . . . .	25
6.5.3.1	brightness_ . . . . .	25
6.5.3.2	color_ . . . . .	25
6.5.3.3	dir_ . . . . .	25
6.6	gameobj::FlatShapable Class Reference . . . . .	26
6.6.1	Member Function Documentation . . . . .	26
6.6.1.1	getColor() . . . . .	26
6.6.1.2	intersecte() . . . . .	26
6.7	raytracer::Landmark Class Reference . . . . .	26
6.7.1	Constructor & Destructor Documentation . . . . .	27
6.7.1.1	Landmark() [1/2] . . . . .	27
6.7.1.2	Landmark() [2/2] . . . . .	27
6.7.2	Member Function Documentation . . . . .	27
6.7.2.1	getO() . . . . .	27
6.7.2.2	getX() . . . . .	28

6.7.2.3	<a href="#">getY()</a>	28
6.7.2.4	<a href="#">getZ()</a>	28
6.7.2.5	<a href="#">transposePoint()</a>	28
6.7.2.6	<a href="#">transposeVect()</a>	28
6.7.3	<a href="#">Member Data Documentation</a>	28
6.7.3.1	<a href="#">o_</a>	28
6.7.3.2	<a href="#">x_</a>	28
6.7.3.3	<a href="#">y_</a>	29
6.7.3.4	<a href="#">z_</a>	29
6.8	<a href="#">gameobj::Lightable Class Reference</a>	29
6.8.1	<a href="#">Member Function Documentation</a>	29
6.8.1.1	<a href="#">getBrightness()</a>	29
6.8.1.2	<a href="#">getColor()</a>	30
6.8.1.3	<a href="#">interact()</a>	30
6.8.1.4	<a href="#">setBrightness()</a>	30
6.8.1.5	<a href="#">setColor()</a>	30
6.9	<a href="#">gameobj::Plan Class Reference</a>	30
6.9.1	<a href="#">Constructor &amp; Destructor Documentation</a>	31
6.9.1.1	<a href="#">Plan() [1/3]</a>	31
6.9.1.2	<a href="#">Plan() [2/3]</a>	31
6.9.1.3	<a href="#">Plan() [3/3]</a>	31
6.9.2	<a href="#">Member Function Documentation</a>	31
6.9.2.1	<a href="#">getColor()</a>	32
6.9.2.2	<a href="#">getD()</a>	32
6.9.2.3	<a href="#">getNorm()</a>	32
6.9.2.4	<a href="#">intersecte()</a>	32
6.9.2.5	<a href="#">setColor()</a>	32
6.9.3	<a href="#">Member Data Documentation</a>	32
6.9.3.1	<a href="#">color_</a>	32
6.9.3.2	<a href="#">d_</a>	33

6.9.3.3	norm_ . . . . .	33
6.10	gameobj::PointLight Class Reference . . . . .	33
6.10.1	Constructor & Destructor Documentation . . . . .	33
6.10.1.1	PointLight() . . . . .	34
6.10.2	Member Function Documentation . . . . .	34
6.10.2.1	getBrightness() . . . . .	34
6.10.2.2	getColor() . . . . .	34
6.10.2.3	getPos() . . . . .	34
6.10.2.4	interact() . . . . .	34
6.10.2.5	setBrightness() . . . . .	34
6.10.2.6	setColor() . . . . .	35
6.10.2.7	setPos() . . . . .	35
6.10.3	Member Data Documentation . . . . .	35
6.10.3.1	brightness_ . . . . .	35
6.10.3.2	color_ . . . . .	35
6.10.3.3	pos_ . . . . .	35
6.11	raytracer::Ray Class Reference . . . . .	35
6.11.1	Constructor & Destructor Documentation . . . . .	36
6.11.1.1	Ray() . . . . .	36
6.11.2	Member Function Documentation . . . . .	36
6.11.2.1	colides() . . . . .	36
6.11.2.2	colidesBefore() . . . . .	36
6.11.2.3	getOrigin() . . . . .	36
6.11.2.4	getVect() . . . . .	37
6.11.3	Member Data Documentation . . . . .	37
6.11.3.1	origin_ . . . . .	37
6.11.3.2	vect_ . . . . .	37
6.12	raytracer::Screen Class Reference . . . . .	37
6.12.1	Constructor & Destructor Documentation . . . . .	38
6.12.1.1	Screen() . . . . .	38



6.12.2	Member Function Documentation	38
6.12.2.1	getCenter()	38
6.12.2.2	getHeight()	38
6.12.2.3	getI()	38
6.12.2.4	getJ()	38
6.12.2.5	getPixelSize()	38
6.12.2.6	getWidth()	39
6.12.3	Member Data Documentation	39
6.12.3.1	center_	39
6.12.3.2	height_	39
6.12.3.3	i_	39
6.12.3.4	j_	39
6.12.3.5	pixelSize_	39
6.12.3.6	width_	39
6.13	gameobj::Shapable Class Reference	40
6.13.1	Member Function Documentation	40
6.13.1.1	getColor()	40
6.13.1.2	intersecte()	40
6.14	gameobj::Square Class Reference	41
6.14.1	Constructor & Destructor Documentation	41
6.14.1.1	Square() [1/2]	41
6.14.1.2	Square() [2/2]	41
6.14.2	Member Function Documentation	42
6.14.2.1	getAB()	42
6.14.2.2	getAD()	42
6.14.2.3	getColor()	42
6.14.2.4	getPlan()	42
6.14.2.5	getPos()	42
6.14.2.6	intersecte()	42
6.14.3	Member Data Documentation	43

6.14.3.1	p_ . . . . .	43
6.14.3.2	pA_ . . . . .	43
6.14.3.3	vAB_ . . . . .	43
6.14.3.4	vAD_ . . . . .	43
6.15	gameui::UIScreen Class Reference . . . . .	43
6.15.1	Constructor & Destructor Documentation . . . . .	44
6.15.1.1	UIScreen() . . . . .	44
6.15.2	Member Data Documentation . . . . .	44
6.15.2.1	sceneRender_ . . . . .	44
6.16	raytracer::Vect2 Class Reference . . . . .	44
6.16.1	Constructor & Destructor Documentation . . . . .	45
6.16.1.1	Vect2() [1/3] . . . . .	45
6.16.1.2	Vect2() [2/3] . . . . .	45
6.16.1.3	Vect2() [3/3] . . . . .	45
6.16.2	Member Function Documentation . . . . .	45
6.16.2.1	dot() . . . . .	45
6.16.2.2	getNorm() . . . . .	45
6.16.2.3	getX() . . . . .	46
6.16.2.4	getY() . . . . .	46
6.16.2.5	isNullVect() . . . . .	46
6.16.2.6	normalize() . . . . .	46
6.16.2.7	operator*() . . . . .	46
6.16.2.8	operator+() . . . . .	46
6.16.2.9	operator-() [1/2] . . . . .	46
6.16.2.10	operator-() [2/2] . . . . .	47
6.16.2.11	setX() . . . . .	47
6.16.2.12	setY() . . . . .	47
6.16.3	Friends And Related Function Documentation . . . . .	47
6.16.3.1	operator* . . . . .	47
6.16.4	Member Data Documentation . . . . .	47

6.16.4.1	x_	47
6.16.4.2	y_	47
6.17	raytracer::Vect3 Class Reference	48
6.17.1	Constructor & Destructor Documentation	49
6.17.1.1	Vect3() [1/3]	49
6.17.1.2	Vect3() [2/3]	49
6.17.1.3	Vect3() [3/3]	49
6.17.2	Member Function Documentation	50
6.17.2.1	dot()	50
6.17.2.2	getNorm()	50
6.17.2.3	getX()	50
6.17.2.4	getY()	50
6.17.2.5	getZ()	51
6.17.2.6	isNullVect()	51
6.17.2.7	normalize()	51
6.17.2.8	operator*()	51
6.17.2.9	operator+()	52
6.17.2.10	operator-() [1/2]	52
6.17.2.11	operator-() [2/2]	52
6.17.2.12	setX()	53
6.17.2.13	setY()	53
6.17.2.14	setZ()	53
6.17.3	Friends And Related Function Documentation	53
6.17.3.1	operator*	53
6.17.4	Member Data Documentation	54
6.17.4.1	x_	54
6.17.4.2	y_	54
6.17.4.3	z_	54

<b>7 File Documentation</b>	<b>55</b>
7.1 includes/fwd.hh File Reference	55
7.2 includes/gameobj/ambientlight.hh File Reference	55
7.3 includes/gameobj/cube.hh File Reference	55
7.4 includes/gameobj/dirlight.hh File Reference	56
7.5 includes/gameobj/flatshapable.hh File Reference	56
7.6 includes/gameobj/lightable.hh File Reference	57
7.7 includes/gameobj/plan.hh File Reference	57
7.8 includes/gameobj/pointlight.hh File Reference	57
7.9 includes/gameobj/shapable.hh File Reference	58
7.10 includes/gameobj/square.hh File Reference	58
7.11 includes/raytracer/camera.hh File Reference	59
7.12 includes/raytracer/color.hh File Reference	59
7.13 includes/raytracer/landmark.hh File Reference	59
7.14 includes/raytracer/ray.hh File Reference	60
7.15 includes/raytracer/screen.hh File Reference	60
7.16 includes/raytracer/utils.hh File Reference	61
7.17 includes/raytracer/vect.hh File Reference	61
7.18 includes/ui/gameui.hh File Reference	62
7.19 src/gameobj/ambientlight.cc File Reference	62
7.20 src/gameobj/cube.cc File Reference	62
7.21 src/gameobj/dirlight.cc File Reference	62
7.22 src/gameobj/plan.cc File Reference	63
7.22.1 Function Documentation	63
7.22.1.1 computeD()	63
7.23 src/gameobj/pointlight.cc File Reference	63
7.24 src/gameobj/square.cc File Reference	63
7.25 src/main.cc File Reference	63
7.25.1 Function Documentation	64
7.25.1.1 main()	64
7.25.1.2 writeRender()	64
7.26 src/raytracer/camera.cc File Reference	64
7.27 src/raytracer/color.cc File Reference	64
7.28 src/raytracer/landmark.cc File Reference	65
7.29 src/raytracer/ray.cc File Reference	65
7.30 src/raytracer/screen.cc File Reference	65
7.31 src/raytracer/utils.cc File Reference	65
7.32 src/raytracer/vect.cc File Reference	65
7.33 src/ui/gameui.cc File Reference	65
<b>Index</b>	<b>67</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">gameobj</a>	.....	9
<a href="#">gameui</a>	.....	9
<a href="#">raytracer</a>	.....	9



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

raytracer::Camera . . . . .	15
raytracer::Color . . . . .	17
gameobj::FlatShapable . . . . .	26
gameobj::Plan . . . . .	30
gameobj::Square . . . . .	41
raytracer::Landmark . . . . .	26
gameobj::Lightable . . . . .	29
gameobj::AmbientLight . . . . .	13
gameobj::DirectionalLight . . . . .	23
gameobj::PointLight . . . . .	33
raytracer::Ray . . . . .	35
raytracer::Screen . . . . .	37
gameobj::Shapable . . . . .	40
gameobj::Cube . . . . .	20
gameobj::Plan . . . . .	30
gameobj::Square . . . . .	41
gameui::UiScreen . . . . .	43
raytracer::Vect2 . . . . .	44
raytracer::Vect3 . . . . .	48





## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">gameobj::AmbientLight</a>	13
<a href="#">raytracer::Camera</a>	15
<a href="#">raytracer::Color</a>	17
<a href="#">gameobj::Cube</a>	20
<a href="#">gameobj::DirectionalLight</a>	23
<a href="#">gameobj::FlatShapable</a>	26
<a href="#">raytracer::Landmark</a>	26
<a href="#">gameobj::Lightable</a>	29
<a href="#">gameobj::Plan</a>	30
<a href="#">gameobj::PointLight</a>	33
<a href="#">raytracer::Ray</a>	35
<a href="#">raytracer::Screen</a>	37
<a href="#">gameobj::Shapable</a>	40
<a href="#">gameobj::Square</a>	41
<a href="#">gameui::UIScreen</a>	43
<a href="#">raytracer::Vect2</a>	44
<a href="#">raytracer::Vect3</a>	48



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

includes/fwd.hh	55
includes/gameobj/ambientlight.hh	55
includes/gameobj/cube.hh	55
includes/gameobj/dirlight.hh	56
includes/gameobj/flatshapable.hh	56
includes/gameobj/lighttable.hh	57
includes/gameobj/plan.hh	57
includes/gameobj/pointlight.hh	57
includes/gameobj/shapable.hh	58
includes/gameobj/square.hh	58
includes/raytracer/camera.hh	59
includes/raytracer/color.hh	59
includes/raytracer/landmark.hh	59
includes/raytracer/ray.hh	60
includes/raytracer/screen.hh	60
includes/raytracer/utlis.hh	61
includes/raytracer/vect.hh	61
includes/ui/gameui.hh	62
src/main.cc	63
src/gameobj/ambientlight.cc	62
src/gameobj/cube.cc	62
src/gameobj/dirlight.cc	62
src/gameobj/plan.cc	63
src/gameobj/pointlight.cc	63
src/gameobj/square.cc	63
src/raytracer/camera.cc	64
src/raytracer/color.cc	64
src/raytracer/landmark.cc	65
src/raytracer/ray.cc	65
src/raytracer/screen.cc	65
src/raytracer/utlis.cc	65
src/raytracer/vect.cc	65
src/ui/gameui.cc	65



## Chapter 5

# Namespace Documentation

### 5.1 gameobj Namespace Reference

#### Classes

- class [AmbientLight](#)
- class [Cube](#)
- class [DirectionalLight](#)
- class [FlatShapable](#)
- class [Lightable](#)
- class [Plan](#)
- class [PointLight](#)
- class [Shapable](#)
- class [Square](#)

### 5.2 gameui Namespace Reference

#### Classes

- class [UiScreen](#)

### 5.3 raytracer Namespace Reference

#### Classes

- class [Camera](#)
- class [Color](#)
- class [Landmark](#)
- class [Ray](#)
- class [Screen](#)
- class [Vect2](#)
- class [Vect3](#)

## Functions

- **Color operator\*** (const float num, const **Color** &color)
- void **rotateX** (**Vect3** &vect, const float angle)
- void **rotateY** (**Vect3** &vect, const float angle)
- void **rotateZ** (**Vect3** &vect, const float angle)
- **Vect3 applyAngle** (**Vect3** &vect, const **Vect3** &angles)
- **Vect3 vectorFromAngles** (const **Vect3** &angles)
- **Vect3 getAngles** (const **Vect3** &v1, const **Vect3** &v2)
- **Vect3 vectorFromPoints** (const **Vect3** &v1, const **Vect3** &v2)
- std::vector< **Ray** > **genRays** (const **Camera** &cam, const **Screen** &screen)
- int **closerToOrigin** (**Vect3** origin, **Vect3** point1, **Vect3** point2)
- std::vector< **Color** > **renderFrame** (const std::vector< **gameobj::Shapable** \*> &objects, const std::vector< **Ray** > &rays, const std::vector< **gameobj::Lightable** \*> &lightList)
- **Vect3 operator\*** (const float num, const **Vect3** &vect)  
*Overload of the product multiplication between a float and a vector.*
- **Vect2 operator\*** (const float num, const **Vect2** &vect)

## Variables

- const float **pi** = std::acos(-1)

### 5.3.1 Function Documentation

#### 5.3.1.1 applyAngle()

```
raytracer::Vect3 raytracer::applyAngle (
    raytracer::Vect3 & vect,
    const Vect3 & angles )
```

#### 5.3.1.2 closerToOrigin()

```
int raytracer::closerToOrigin (
    raytracer::Vect3 origin,
    raytracer::Vect3 point1,
    raytracer::Vect3 point2 )
```

#### 5.3.1.3 genRays()

```
std::vector< raytracer::Ray > raytracer::genRays (
    const Camera & cam,
    const Screen & screen )
```

#### 5.3.1.4 getAngles()

```
raytracer::Vect3 raytracer::getAngles (
    const Vect3 & v1,
    const Vect3 & v2 )
```

#### 5.3.1.5 operator\*() [1/3]

```
raytracer::Color raytracer::operator* (
    const float num,
    const Color & color )
```

#### 5.3.1.6 operator\*() [2/3]

```
raytracer::Vect3 raytracer::operator* (
    const float num,
    const Vect3 & vect )
```

Overload of the product multiplication between a float and a vector.

##### Parameters

<i>num</i>	the float to multiply with the vector.
<i>vect</i>	the vector to multiply with the float.

##### Returns

A new vector.

#### 5.3.1.7 operator\*() [3/3]

```
raytracer::Vect2 raytracer::operator* (
    const float num,
    const Vect2 & vect )
```

#### 5.3.1.8 renderFrame()

```
std::vector< raytracer::Color > raytracer::renderFrame (
    const std::vector< gameobj::Shapable *> & objects,
    const std::vector< Ray > & rays,
    const std::vector< gameobj::Lightable *> & lightList )
```

#### 5.3.1.9 rotateX()

```
void raytracer::rotateX (
    raytracer::Vect3 & vect,
    const float angle )
```

#### 5.3.1.10 rotateY()

```
void raytracer::rotateY (
    raytracer::Vect3 & vect,
    const float angle )
```

#### 5.3.1.11 rotateZ()

```
void raytracer::rotateZ (
    raytracer::Vect3 & vect,
    const float angle )
```

#### 5.3.1.12 vectorFromAngles()

```
raytracer::Vect3 raytracer::vectorFromAngles (
    const Vect3 & angles )
```

#### 5.3.1.13 vectorFromPoints()

```
raytracer::Vect3 raytracer::vectorFromPoints (
    const Vect3 & v1,
    const Vect3 & v2 )
```

### 5.3.2 Variable Documentation

#### 5.3.2.1 pi

```
const float raytracer::pi = std::acos(-1)
```



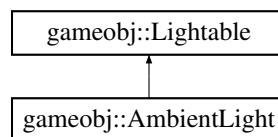
## Chapter 6

# Class Documentation

### 6.1 gameobj::AmbientLight Class Reference

```
#include <ambientlight.hh>
```

Inheritance diagram for gameobj::AmbientLight:



#### Public Member Functions

- [AmbientLight](#) (const [raytracer::Color](#) &color, const float brightness)
- [raytracer::Color getColor](#) () const
- float [getBrightness](#) () const
- void [setColor](#) (const [raytracer::Color](#) &color)
- void [setBrightness](#) (const float brightness)
- virtual [raytracer::Color interact](#) (const std::vector< [Shapable](#) \*> &obj, const [raytracer::Vect3](#) &point, const [FlatShapable](#) &) const override

#### Private Attributes

- [raytracer::Color color\\_](#)
- float [brightness\\_](#)

#### 6.1.1 Constructor & Destructor Documentation

#### 6.1.1.1 AmbientLight()

```
gameobj::AmbientLight::AmbientLight (
    const raytracer::Color & color,
    const float brightness )
```

### 6.1.2 Member Function Documentation

#### 6.1.2.1 getBrightness()

```
float gameobj::AmbientLight::getBrightness ( ) const
```

#### 6.1.2.2 getColor()

```
raytracer::Color gameobj::AmbientLight::getColor ( ) const
```

#### 6.1.2.3 interact()

```
raytracer::Color gameobj::AmbientLight::interact (
    const std::vector< Shapable *> & obj,
    const raytracer::Vect3 & point,
    const FlatShapable & obj ) const [override], [virtual]
```

Implements [gameobj::Lightable](#).

#### 6.1.2.4 setBrightness()

```
void gameobj::AmbientLight::setBrightness (
    const float brightness )
```

#### 6.1.2.5 setColor()

```
void gameobj::AmbientLight::setColor (
    const raytracer::Color & color )
```

### 6.1.3 Member Data Documentation

#### 6.1.3.1 brightness\_

```
float gameobj::AmbientLight::brightness_ [private]
```

#### 6.1.3.2 color\_

```
raytracer::Color gameobj::AmbientLight::color_ [private]
```

The documentation for this class was generated from the following files:

- includes/gameobj/[ambientlight.hh](#)
- src/gameobj/[ambientlight.cc](#)

## 6.2 raytracer::Camera Class Reference

```
#include <camera.hh>
```

### Public Member Functions

- [Camera](#) ()
- [Camera](#) ([Vect3](#) pos, [Vect3](#) angleVect, float viewAngle)
- [Vect3](#) [getPos](#) () const
- [Vect3](#) [getAng](#) () const
- float [getViewAngle](#) () const

### Private Attributes

- [Vect3](#) [pos\\_](#)
- [Vect3](#) [angleVect\\_](#)
- float [viewAngle\\_](#)

### 6.2.1 Constructor & Destructor Documentation

#### 6.2.1.1 [Camera\(\)](#) [1/2]

```
raytracer::Camera::Camera ( )
```

### 6.2.1.2 Camera() [2/2]

```
raytracer::Camera::Camera (
    raytracer::Vect3 pos,
    raytracer::Vect3 angleVect,
    float viewAngle = raytracer::pi / 2 )
```

## 6.2.2 Member Function Documentation

### 6.2.2.1 getAng()

```
raytracer::Vect3 raytracer::Camera::getAng ( ) const
```

### 6.2.2.2 getPos()

```
raytracer::Vect3 raytracer::Camera::getPos ( ) const
```

### 6.2.2.3 getViewAngle()

```
float raytracer::Camera::getViewAngle ( ) const
```

## 6.2.3 Member Data Documentation

### 6.2.3.1 angleVect\_

```
Vect3 raytracer::Camera::angleVect_ [private]
```

### 6.2.3.2 pos\_

```
Vect3 raytracer::Camera::pos_ [private]
```

### 6.2.3.3 viewAngle\_

```
float raytracer::Camera::viewAngle_ [private]
```

The documentation for this class was generated from the following files:

- includes/raytracer/camera.hh
- src/raytracer/camera.cc

## 6.3 raytracer::Color Class Reference

```
#include <color.hh>
```

### Public Member Functions

- [Color](#) ()
- [Color](#) (const float r, const float g, const float b)
- float [getR](#) () const
- float [getG](#) () const
- float [getB](#) () const
- void [setR](#) (const float v)
- void [setG](#) (const float v)
- void [setB](#) (const float v)
- void [normalize](#) ()
- [Color operator+](#) (const [Color](#) &other) const
- [Color operator+=](#) (const [Color](#) &other)
- [Color operator-](#) (const [Color](#) &other) const
- [Color operator\\*](#) (const float other) const
- [Color operator\\*](#) (const [Color](#) &other) const

### Private Attributes

- float [r\\_](#)
- float [g\\_](#)
- float [b\\_](#)

### Friends

- [Color operator\\*](#) (const float num, const [Color](#) &color)

### 6.3.1 Constructor & Destructor Documentation

#### 6.3.1.1 Color() [1/2]

```
raytracer::Color::Color ( )
```

#### 6.3.1.2 Color() [2/2]

```
raytracer::Color::Color (
    const float r,
    const float g,
    const float b )
```

### 6.3.2 Member Function Documentation

#### 6.3.2.1 getB()

```
float raytracer::Color::getB ( ) const
```

#### 6.3.2.2 getG()

```
float raytracer::Color::getG ( ) const
```

#### 6.3.2.3 getR()

```
float raytracer::Color::getR ( ) const
```

#### 6.3.2.4 normalize()

```
void raytracer::Color::normalize ( )
```

#### 6.3.2.5 operator\*() [1/2]

```
raytracer::Color raytracer::Color::operator* (
    const float other ) const
```

#### 6.3.2.6 operator\*() [2/2]

```
raytracer::Color raytracer::Color::operator* (
    const Color & other ) const
```

#### 6.3.2.7 operator+()

```
raytracer::Color raytracer::Color::operator+ (
    const Color & other ) const
```

#### 6.3.2.8 operator+=()

```
raytracer::Color raytracer::Color::operator+= (
    const Color & other )
```

#### 6.3.2.9 operator-()

```
raytracer::Color raytracer::Color::operator- (
    const Color & other ) const
```

#### 6.3.2.10 setB()

```
void raytracer::Color::setB (
    const float v )
```

#### 6.3.2.11 setG()

```
void raytracer::Color::setG (
    const float v )
```

#### 6.3.2.12 setR()

```
void raytracer::Color::setR (
    const float v )
```

### 6.3.3 Friends And Related Function Documentation

#### 6.3.3.1 operator\*

```
Color operator* (
    const float num,
    const Color & color ) [friend]
```

### 6.3.4 Member Data Documentation

#### 6.3.4.1 b\_

```
float raytracer::Color::b_ [private]
```

#### 6.3.4.2 g\_

```
float raytracer::Color::g_ [private]
```

#### 6.3.4.3 r\_

```
float raytracer::Color::r_ [private]
```

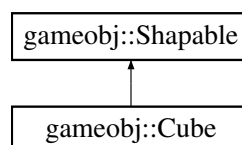
The documentation for this class was generated from the following files:

- [includes/raytracer/color.hh](#)
- [src/raytracer/color.cc](#)

## 6.4 gameobj::Cube Class Reference

```
#include <cube.hh>
```

Inheritance diagram for gameobj::Cube:





## Public Member Functions

- [Cube](#) (const [raytracer::Vect3](#) &center, const float side, const [raytracer::Vect3](#) &angles, const [raytracer::Color](#) &color)
- [raytracer::Vect3](#) [getCenter](#) () const
- float [getSide](#) () const
- [raytracer::Vect3](#) [getAngles](#) () const
- [raytracer::Landmark](#) [getLandmark](#) () const
- [raytracer::Color](#) [getColor](#) () const
- virtual std::optional< std::tuple< [raytracer::Vect3](#), [FlatShapable](#) \* > > [intersecte](#) (const [raytracer::Ray](#) &ray) override

## Private Attributes

- [raytracer::Vect3](#) center\_
- float side\_
- [raytracer::Vect3](#) angles\_
- [raytracer::Landmark](#) landmark\_
- std::vector< [Square](#) > squares\_

### 6.4.1 Constructor & Destructor Documentation

#### 6.4.1.1 Cube()

```
gameobj::Cube::Cube (
    const raytracer::Vect3 & center,
    const float side,
    const raytracer::Vect3 & angles,
    const raytracer::Color & color )
```

### 6.4.2 Member Function Documentation

#### 6.4.2.1 getAngles()

```
raytracer::Vect3 gameobj::Cube::getAngles ( ) const
```

#### 6.4.2.2 getCenter()

```
raytracer::Vect3 gameobj::Cube::getCenter ( ) const
```

#### 6.4.2.3 getColor()

```
raytracer::Color gameobj::Cube::getColor ( ) const [virtual]
```

Implements [gameobj::Shapable](#).

#### 6.4.2.4 getLandmark()

```
raytracer::Landmark gameobj::Cube::getLandmark ( ) const
```

#### 6.4.2.5 getSide()

```
float gameobj::Cube::getSide ( ) const
```

#### 6.4.2.6 intersecte()

```
std::optional< std::tuple< raytracer::Vect3, gameobj::FlatShapable * > > gameobj::Cube↵  
::intersecte (   
    const raytracer::Ray & ray ) [override], [virtual]
```

Implements [gameobj::Shapable](#).

### 6.4.3 Member Data Documentation

#### 6.4.3.1 angles\_

```
raytracer::Vect3 gameobj::Cube::angles_ [private]
```

#### 6.4.3.2 center\_

```
raytracer::Vect3 gameobj::Cube::center_ [private]
```

## 6.4.3.3 landmark\_

```
raytracer::Landmark gameobj::Cube::landmark_ [private]
```

## 6.4.3.4 side\_

```
float gameobj::Cube::side_ [private]
```

## 6.4.3.5 squares\_

```
std::vector<Square> gameobj::Cube::squares_ [private]
```

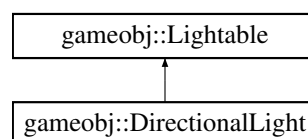
The documentation for this class was generated from the following files:

- includes/gameobj/cube.hh
- src/gameobj/cube.cc

## 6.5 gameobj::DirectionalLight Class Reference

```
#include <dirlight.hh>
```

Inheritance diagram for gameobj::DirectionalLight:



## Public Member Functions

- **DirectionalLight** (const raytracer::Vect3 &dir, const raytracer::Color &color, const float brightness)
- raytracer::Vect3 **getDir** () const
- raytracer::Color **getColor** () const
- float **getBrightness** () const
- void **setDir** (const raytracer::Vect3 &dir)
- void **setColor** (const raytracer::Color &color)
- void **setBrightness** (const float brightness)
- virtual raytracer::Color **interact** (const std::vector< Shapable \*> &obj, const raytracer::Vect3 &point, const FlatShapable &) const override

## Private Attributes

- [raytracer::Vect3](#) `dir_`
- [raytracer::Color](#) `color_`
- float `brightness_`

## 6.5.1 Constructor & Destructor Documentation

### 6.5.1.1 DirectionalLight()

```
gameobj::DirectionalLight::DirectionalLight (
    const raytracer::Vect3 & dir,
    const raytracer::Color & color,
    const float brightness )
```

## 6.5.2 Member Function Documentation

### 6.5.2.1 getBrightness()

```
float gameobj::DirectionalLight::getBrightness ( ) const
```

### 6.5.2.2 getColor()

```
raytracer::Color gameobj::DirectionalLight::getColor ( ) const
```

### 6.5.2.3 getDir()

```
raytracer::Vect3 gameobj::DirectionalLight::getDir ( ) const
```

### 6.5.2.4 interact()

```
raytracer::Color gameobj::DirectionalLight::interact (
    const std::vector< Shapable *> & obj,
    const raytracer::Vect3 & point,
    const FlatShapable & ) const [override], [virtual]
```

Implements [gameobj::Lightable](#).

#### 6.5.2.5 setBrightness()

```
void gameobj::DirectionalLight::setBrightness (
    const float brightness )
```

#### 6.5.2.6 setColor()

```
void gameobj::DirectionalLight::setColor (
    const raytracer::Color & color )
```

#### 6.5.2.7 setDir()

```
void gameobj::DirectionalLight::setDir (
    const raytracer::Vect3 & dir )
```

### 6.5.3 Member Data Documentation

#### 6.5.3.1 brightness\_

```
float gameobj::DirectionalLight::brightness_ [private]
```

#### 6.5.3.2 color\_

```
raytracer::Color gameobj::DirectionalLight::color_ [private]
```

#### 6.5.3.3 dir\_

```
raytracer::Vect3 gameobj::DirectionalLight::dir_ [private]
```

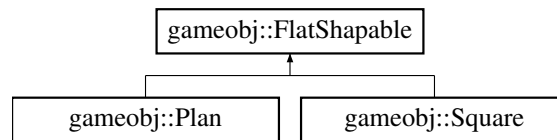
The documentation for this class was generated from the following files:

- includes/gameobj/[dirlight.hh](#)
- src/gameobj/[dirlight.cc](#)

## 6.6 gameobj::FlatShapable Class Reference

```
#include <flatshapable.hh>
```

Inheritance diagram for gameobj::FlatShapable:



### Public Member Functions

- virtual [raytracer::Color](#) [getColor](#) ( ) const =0
- virtual std::optional< std::tuple< [raytracer::Vect3](#), [FlatShapable](#) \* > > [intersecte](#) (const [raytracer::Ray](#) &ray)=0

#### 6.6.1 Member Function Documentation

##### 6.6.1.1 getColor()

```
virtual raytracer::Color gameobj::FlatShapable::getColor ( ) const [pure virtual]
```

Implemented in [gameobj::Square](#), and [gameobj::Plan](#).

##### 6.6.1.2 intersecte()

```
virtual std::optional<std::tuple<raytracer::Vect3, FlatShapable*> > gameobj::FlatShapable↵
::intersecte (
    const raytracer::Ray & ray ) [pure virtual]
```

Implemented in [gameobj::Square](#), and [gameobj::Plan](#).

The documentation for this class was generated from the following file:

- includes/gameobj/[flatshapable.hh](#)

## 6.7 raytracer::Landmark Class Reference

```
#include <landmark.hh>
```

## Public Member Functions

- [Landmark](#) ()
- [Landmark](#) (const [Vect3](#) &o, const [Vect3](#) &x, const [Vect3](#) &y, const [Vect3](#) &z)
- [Vect3](#) getO () const
- [Vect3](#) getX () const
- [Vect3](#) getY () const
- [Vect3](#) getZ () const
- [Vect3](#) transposePoint (const [Vect3](#) &point) const
- [Vect3](#) transposeVect (const [Vect3](#) &vect) const

## Private Attributes

- [Vect3](#) o\_
- [Vect3](#) x\_
- [Vect3](#) y\_
- [Vect3](#) z\_

## 6.7.1 Constructor & Destructor Documentation

### 6.7.1.1 [Landmark\(\)](#) [1/2]

```
raytracer::Landmark::Landmark ( )
```

### 6.7.1.2 [Landmark\(\)](#) [2/2]

```
raytracer::Landmark::Landmark (
    const Vect3 & o,
    const Vect3 & x,
    const Vect3 & y,
    const Vect3 & z )
```

## 6.7.2 Member Function Documentation

### 6.7.2.1 [getO\(\)](#)

```
raytracer::Vect3 raytracer::Landmark::getO ( ) const
```

#### 6.7.2.2 getX()

```
raytracer::Vect3 raytracer::Landmark::getX ( ) const
```

#### 6.7.2.3 getY()

```
raytracer::Vect3 raytracer::Landmark::getY ( ) const
```

#### 6.7.2.4 getZ()

```
raytracer::Vect3 raytracer::Landmark::getZ ( ) const
```

#### 6.7.2.5 transposePoint()

```
raytracer::Vect3 raytracer::Landmark::transposePoint (
    const Vect3 & point ) const
```

#### 6.7.2.6 transposeVect()

```
raytracer::Vect3 raytracer::Landmark::transposeVect (
    const Vect3 & vect ) const
```

### 6.7.3 Member Data Documentation

#### 6.7.3.1 o\_

```
Vect3 raytracer::Landmark::o_ [private]
```

#### 6.7.3.2 x\_

```
Vect3 raytracer::Landmark::x_ [private]
```



## 6.7.3.3 y\_

```
Vect3 raytracer::Landmark::y_ [private]
```

## 6.7.3.4 z\_

```
Vect3 raytracer::Landmark::z_ [private]
```

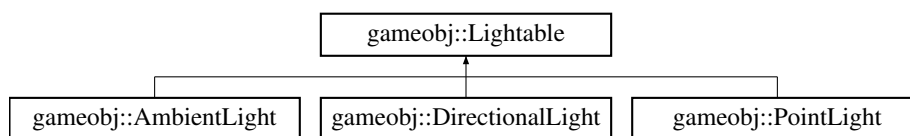
The documentation for this class was generated from the following files:

- includes/raytracer/[landmark.hh](#)
- src/raytracer/[landmark.cc](#)

## 6.8 gameobj::Lightable Class Reference

```
#include <lightable.hh>
```

Inheritance diagram for gameobj::Lightable:



### Public Member Functions

- [raytracer::Color getColor](#) () const
- float [getBrightness](#) () const
- void [setColor](#) (const [raytracer::Color](#) &color)
- void [setBrightness](#) (const float brightness)
- virtual [raytracer::Color interact](#) (const std::vector< [Shapable](#) \*> &obj, const [raytracer::Vect3](#) &point, const [FlatShapable](#) &) const =0

### 6.8.1 Member Function Documentation

#### 6.8.1.1 getBrightness()

```
float gameobj::Lightable::getBrightness ( ) const
```

### 6.8.1.2 getColor()

```
raytracer::Color gameobj::Lightable::getColor ( ) const
```

### 6.8.1.3 interact()

```
virtual raytracer::Color gameobj::Lightable::interact (
    const std::vector< Shapable *> & obj,
    const raytracer::Vect3 & point,
    const FlatShapable & ) const [pure virtual]
```

Implemented in [gameobj::DirectionalLight](#), [gameobj::PointLight](#), and [gameobj::AmbientLight](#).

### 6.8.1.4 setBrightness()

```
void gameobj::Lightable::setBrightness (
    const float brightness )
```

### 6.8.1.5 setColor()

```
void gameobj::Lightable::setColor (
    const raytracer::Color & color )
```

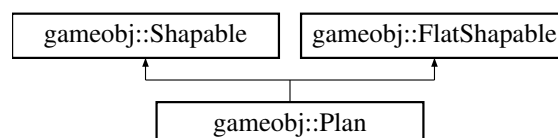
The documentation for this class was generated from the following file:

- [includes/gameobj/lightable.hh](#)

## 6.9 gameobj::Plan Class Reference

```
#include <plan.hh>
```

Inheritance diagram for gameobj::Plan:



## Public Member Functions

- [Plan](#) ()
- [Plan](#) (const [Plan](#) &p)
- [Plan](#) (const [raytracer::Vect3](#) &norm, const [raytracer::Vect3](#) &point, const [raytracer::Color](#) &color)
- [raytracer::Vect3](#) [getNorm](#) () const
- float [getD](#) () const
- [raytracer::Color](#) [getColor](#) () const override
- void [setColor](#) (const [raytracer::Color](#) &color)
- std::optional< std::tuple< [raytracer::Vect3](#), [FlatShapable](#) \* > > [intersecte](#) (const [raytracer::Ray](#) &ray) override

## Private Attributes

- float [d\\_](#)
- [raytracer::Vect3](#) [norm\\_](#)
- [raytracer::Color](#) [color\\_](#)

## 6.9.1 Constructor & Destructor Documentation

### 6.9.1.1 [Plan\(\)](#) [1/3]

```
gameobj::Plan::Plan ( )
```

### 6.9.1.2 [Plan\(\)](#) [2/3]

```
gameobj::Plan::Plan (
    const Plan & p )
```

### 6.9.1.3 [Plan\(\)](#) [3/3]

```
gameobj::Plan::Plan (
    const raytracer::Vect3 & norm,
    const raytracer::Vect3 & point,
    const raytracer::Color & color )
```

## 6.9.2 Member Function Documentation

#### 6.9.2.1 getColor()

```
raytracer::Color gameobj::Plan::getColor ( ) const [override], [virtual]
```

Implements [gameobj::FlatShapable](#).

#### 6.9.2.2 getD()

```
float gameobj::Plan::getD ( ) const
```

#### 6.9.2.3 getNorm()

```
raytracer::Vect3 gameobj::Plan::getNorm ( ) const
```

#### 6.9.2.4 intersecte()

```
std::optional< std::tuple< raytracer::Vect3, gameobj::FlatShapable * > > gameobj::Plan↵  
::intersecte (   
    const raytracer::Ray & ray ) [override], [virtual]
```

Implements [gameobj::FlatShapable](#).

#### 6.9.2.5 setColor()

```
void gameobj::Plan::setColor (   
    const raytracer::Color & color )
```

### 6.9.3 Member Data Documentation

#### 6.9.3.1 color\_

```
raytracer::Color gameobj::Plan::color_ [private]
```

## 6.9.3.2 d\_

```
float gameobj::Plan::d_ [private]
```

## 6.9.3.3 norm\_

```
raytracer::Vect3 gameobj::Plan::norm_ [private]
```

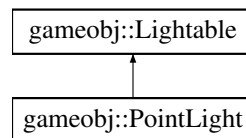
The documentation for this class was generated from the following files:

- includes/gameobj/[plan.hh](#)
- src/gameobj/[plan.cc](#)

## 6.10 gameobj::PointLight Class Reference

```
#include <pointlight.hh>
```

Inheritance diagram for gameobj::PointLight:



## Public Member Functions

- [PointLight](#) (const [raytracer::Vect3](#) &pos, const [raytracer::Color](#) &color, const float brightness)
- [raytracer::Vect3](#) [getPos](#) () const
- [raytracer::Color](#) [getColor](#) () const
- float [getBrightness](#) () const
- void [setPos](#) (const [raytracer::Vect3](#) &pos)
- void [setColor](#) (const [raytracer::Color](#) &color)
- void [setBrightness](#) (const float brightness)
- virtual [raytracer::Color](#) [interact](#) (const std::vector< [Shapable](#) \*> &obj, const [raytracer::Vect3](#) &point, const [FlatShapable](#) &) const override

## Private Attributes

- [raytracer::Vect3](#) [pos\\_](#)
- [raytracer::Color](#) [color\\_](#)
- float [brightness\\_](#)

## 6.10.1 Constructor &amp; Destructor Documentation

### 6.10.1.1 PointLight()

```
gameobj::PointLight::PointLight (
    const raytracer::Vect3 & pos,
    const raytracer::Color & color,
    const float brightness )
```

## 6.10.2 Member Function Documentation

### 6.10.2.1 getBrightness()

```
float gameobj::PointLight::getBrightness ( ) const
```

### 6.10.2.2 getColor()

```
raytracer::Color gameobj::PointLight::getColor ( ) const
```

### 6.10.2.3 getPos()

```
raytracer::Vect3 gameobj::PointLight::getPos ( ) const
```

### 6.10.2.4 interact()

```
raytracer::Color gameobj::PointLight::interact (
    const std::vector< Shapable *> & obj,
    const raytracer::Vect3 & point,
    const FlatShapable & obj ) const [override], [virtual]
```

Implements [gameobj::Lightable](#).

### 6.10.2.5 setBrightness()

```
void gameobj::PointLight::setBrightness (
    const float brightness )
```

#### 6.10.2.6 setColor()

```
void gameobj::PointLight::setColor (
    const raytracer::Color & color )
```

#### 6.10.2.7 setPos()

```
void gameobj::PointLight::setPos (
    const raytracer::Vect3 & pos )
```

### 6.10.3 Member Data Documentation

#### 6.10.3.1 brightness\_

```
float gameobj::PointLight::brightness_ [private]
```

#### 6.10.3.2 color\_

```
raytracer::Color gameobj::PointLight::color_ [private]
```

#### 6.10.3.3 pos\_

```
raytracer::Vect3 gameobj::PointLight::pos_ [private]
```

The documentation for this class was generated from the following files:

- includes/gameobj/[pointlight.hh](#)
- src/gameobj/[pointlight.cc](#)

## 6.11 raytracer::Ray Class Reference

```
#include <ray.hh>
```

## Public Member Functions

- [Ray](#) (const [Vect3](#) &origin, const [Vect3](#) &vect)
- [Vect3 getOrigin](#) () const
- [Vect3 getVect](#) () const
- bool [colidesBefore](#) (const std::vector< [gameobj::Shapable](#) \*> &objects, const float dist) const
- bool [colides](#) (const std::vector< [gameobj::Shapable](#) \*> &objects) const

## Private Attributes

- [Vect3 origin\\_](#)
- [Vect3 vect\\_](#)

## 6.11.1 Constructor & Destructor Documentation

### 6.11.1.1 Ray()

```
raytracer::Ray::Ray (  
    const Vect3 & origin,  
    const Vect3 & vect )
```

## 6.11.2 Member Function Documentation

### 6.11.2.1 colides()

```
bool raytracer::Ray::colides (  
    const std::vector< gameobj::Shapable *> & objects ) const
```

### 6.11.2.2 colidesBefore()

```
bool raytracer::Ray::colidesBefore (  
    const std::vector< gameobj::Shapable *> & objects,  
    const float dist ) const
```

### 6.11.2.3 getOrigin()

```
raytracer::Vect3 raytracer::Ray::getOrigin ( ) const
```



#### 6.11.2.4 getVect()

```
raytracer::Vect3 raytracer::Ray::getVect ( ) const
```

### 6.11.3 Member Data Documentation

#### 6.11.3.1 origin\_

```
Vect3 raytracer::Ray::origin_ [private]
```

#### 6.11.3.2 vect\_

```
Vect3 raytracer::Ray::vect_ [private]
```

The documentation for this class was generated from the following files:

- includes/raytracer/[ray.hh](#)
- src/raytracer/[ray.cc](#)

## 6.12 raytracer::Screen Class Reference

```
#include <screen.hh>
```

### Public Member Functions

- [Screen](#) (const [Camera](#) &camera, float width, float height)
- float [getWidth](#) () const
- float [getHeight](#) () const
- float [getPixelSize](#) () const
- [Vect3](#) [getCenter](#) () const
- [Vect3](#) [getI](#) () const
- [Vect3](#) [getJ](#) () const

### Private Attributes

- float [width\\_](#)
- float [height\\_](#)
- float [pixelSize\\_](#)
- [Vect3](#) [center\\_](#)
- [Vect3](#) [i\\_](#)
- [Vect3](#) [j\\_](#)

## 6.12.1 Constructor & Destructor Documentation

### 6.12.1.1 Screen()

```
raytracer::Screen::Screen (
    const Camera & camera,
    float width,
    float height )
```

## 6.12.2 Member Function Documentation

### 6.12.2.1 getCenter()

```
raytracer::Vect3 raytracer::Screen::getCenter ( ) const
```

### 6.12.2.2 getHeight()

```
float raytracer::Screen::getHeight ( ) const
```

### 6.12.2.3 getI()

```
raytracer::Vect3 raytracer::Screen::getI ( ) const
```

### 6.12.2.4 getJ()

```
raytracer::Vect3 raytracer::Screen::getJ ( ) const
```

### 6.12.2.5 getPixelSize()

```
float raytracer::Screen::getPixelSize ( ) const
```

#### 6.12.2.6 getWidth()

```
float raytracer::Screen::getWidth ( ) const
```

### 6.12.3 Member Data Documentation

#### 6.12.3.1 center\_

```
Vect3 raytracer::Screen::center_ [private]
```

#### 6.12.3.2 height\_

```
float raytracer::Screen::height_ [private]
```

#### 6.12.3.3 i\_

```
Vect3 raytracer::Screen::i_ [private]
```

#### 6.12.3.4 j\_

```
Vect3 raytracer::Screen::j_ [private]
```

#### 6.12.3.5 pixelSize\_

```
float raytracer::Screen::pixelSize_ [private]
```

#### 6.12.3.6 width\_

```
float raytracer::Screen::width_ [private]
```

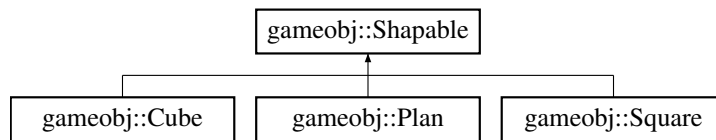
The documentation for this class was generated from the following files:

- includes/raytracer/[screen.hh](#)
- src/raytracer/[screen.cc](#)

## 6.13 gameobj::Shapable Class Reference

```
#include <shapable.hh>
```

Inheritance diagram for gameobj::Shapable:



### Public Member Functions

- virtual [raytracer::Color](#) [getColor](#) () const =0
- virtual std::optional< std::tuple< [raytracer::Vect3](#), [FlatShapable](#) \* > > [intersecte](#) (const [raytracer::Ray](#) &ray)=0

### 6.13.1 Member Function Documentation

#### 6.13.1.1 getColor()

```
virtual raytracer::Color gameobj::Shapable::getColor ( ) const [pure virtual]
```

Implemented in [gameobj::Cube](#), [gameobj::Square](#), and [gameobj::Plan](#).

#### 6.13.1.2 intersecte()

```
virtual std::optional<std::tuple<raytracer::Vect3, FlatShapable*> > gameobj::Shapable::intersecte
(
    const raytracer::Ray & ray ) [pure virtual]
```

Implemented in [gameobj::Cube](#), [gameobj::Square](#), and [gameobj::Plan](#).

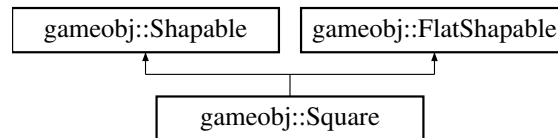
The documentation for this class was generated from the following file:

- includes/gameobj/[shapable.hh](#)

## 6.14 gameobj::Square Class Reference

```
#include <square.hh>
```

Inheritance diagram for gameobj::Square:



### Public Member Functions

- [Square](#) (const [Square](#) &s)
- [Square](#) (const [raytracer::Vect3](#) &A, const [raytracer::Vect3](#) &B, const [raytracer::Vect3](#) &D, const [raytracer::Vect3](#) &norm, const [raytracer::Color](#) &color)
- [raytracer::Vect3](#) [getPos](#) () const
- [raytracer::Vect3](#) [getAB](#) () const
- [raytracer::Vect3](#) [getAD](#) () const
- [gameobj::Plan](#) [getPlan](#) () const
- [raytracer::Color](#) [getColor](#) () const override
- virtual [std::optional< std::tuple< raytracer::Vect3, FlatShapable \\* > >](#) [intersecte](#) (const [raytracer::Ray](#) &ray) override

### Private Attributes

- [Plan](#) [p\\_](#)
- [raytracer::Vect3](#) [pA\\_](#)
- [raytracer::Vect3](#) [vAB\\_](#)
- [raytracer::Vect3](#) [vAD\\_](#)

### 6.14.1 Constructor & Destructor Documentation

#### 6.14.1.1 [Square\(\)](#) [1/2]

```
gameobj::Square::Square (
    const Square & s )
```

#### 6.14.1.2 [Square\(\)](#) [2/2]

```
gameobj::Square::Square (
    const raytracer::Vect3 & A,
    const raytracer::Vect3 & B,
    const raytracer::Vect3 & D,
    const raytracer::Vect3 & norm,
    const raytracer::Color & color )
```

## 6.14.2 Member Function Documentation

### 6.14.2.1 getAB()

```
raytracer::Vect3 gameobj::Square::getAB ( ) const
```

### 6.14.2.2 getAD()

```
raytracer::Vect3 gameobj::Square::getAD ( ) const
```

### 6.14.2.3 getColor()

```
raytracer::Color gameobj::Square::getColor ( ) const [override], [virtual]
```

Implements [gameobj::FlatShapable](#).

### 6.14.2.4 getPlan()

```
gameobj::Plan gameobj::Square::getPlan ( ) const
```

### 6.14.2.5 getPos()

```
raytracer::Vect3 gameobj::Square::getPos ( ) const
```

### 6.14.2.6 intersecte()

```
std::optional< std::tuple< raytracer::Vect3, gameobj::FlatShapable * > > gameobj::Square←  
::intersecte (   
    const raytracer::Ray & ray ) [override], [virtual]
```

Implements [gameobj::FlatShapable](#).

### 6.14.3 Member Data Documentation

#### 6.14.3.1 p\_

`Plan` gameobj::Square::p\_ [private]

#### 6.14.3.2 pA\_

`raytracer::Vect3` gameobj::Square::pA\_ [private]

#### 6.14.3.3 vAB\_

`raytracer::Vect3` gameobj::Square::vAB\_ [private]

#### 6.14.3.4 vAD\_

`raytracer::Vect3` gameobj::Square::vAD\_ [private]

The documentation for this class was generated from the following files:

- [includes/gameobj/square.hh](#)
- [src/gameobj/square.cc](#)

## 6.15 gameui::UIScreen Class Reference

```
#include <gameui.hh>
```

### Public Member Functions

- [UIScreen](#) ()

### Private Attributes

- `std::vector< raytracer::Color >` & `sceneRender_`

## 6.15.1 Constructor & Destructor Documentation

### 6.15.1.1 UIScreen()

```
gameui::UIScreen::UIScreen ( )
```

## 6.15.2 Member Data Documentation

### 6.15.2.1 sceneRender\_

```
std::vector<raytracer::Color>& gameui::UIScreen::sceneRender_ [private]
```

The documentation for this class was generated from the following file:

- [includes/ui/gameui.hh](#)

## 6.16 raytracer::Vect2 Class Reference

```
#include <vect.hh>
```

### Public Member Functions

- [Vect2](#) ()
- [Vect2](#) (float x, float y)
- [Vect2](#) (const [Vect2](#) &v)
- float [getX](#) () const
- float [getY](#) () const
- void [setX](#) (const float v)
- void [setY](#) (const float v)
- bool [isNullVect](#) () const
- float [getNorm](#) () const
- float [dot](#) (const [Vect2](#) &other) const
- void [normalize](#) ()
- [Vect2 operator-](#) () const
- [Vect2 operator+](#) (const [Vect2](#) &other) const
- [Vect2 operator-](#) (const [Vect2](#) &other) const
- [Vect2 operator\\*](#) (const float other) const

### Private Attributes

- float [x\\_](#)
- float [y\\_](#)



## Friends

- [Vect2 operator\\*](#) (const float num, const [Vect2](#) &vect)

## 6.16.1 Constructor & Destructor Documentation

### 6.16.1.1 Vect2() [1/3]

```
raytracer::Vect2::Vect2 ( )
```

### 6.16.1.2 Vect2() [2/3]

```
raytracer::Vect2::Vect2 (
    float x,
    float y )
```

### 6.16.1.3 Vect2() [3/3]

```
raytracer::Vect2::Vect2 (
    const Vect2 & v )
```

## 6.16.2 Member Function Documentation

### 6.16.2.1 dot()

```
float raytracer::Vect2::dot (
    const Vect2 & other ) const
```

### 6.16.2.2 getNorm()

```
float raytracer::Vect2::getNorm ( ) const
```

#### 6.16.2.3 getX()

```
float raytracer::Vect2::getX ( ) const
```

#### 6.16.2.4 getY()

```
float raytracer::Vect2::getY ( ) const
```

#### 6.16.2.5 isNullVect()

```
bool raytracer::Vect2::isNullVect ( ) const
```

#### 6.16.2.6 normalize()

```
void raytracer::Vect2::normalize ( )
```

#### 6.16.2.7 operator\*()

```
raytracer::Vect2 raytracer::Vect2::operator* (
    const float other ) const
```

#### 6.16.2.8 operator+()

```
raytracer::Vect2 raytracer::Vect2::operator+ (
    const Vect2 & other ) const
```

#### 6.16.2.9 operator-() <sup>[1/2]</sup>

```
raytracer::Vect2 raytracer::Vect2::operator- ( ) const
```

#### 6.16.2.10 operator-() [2/2]

```
raytracer::Vect2 raytracer::Vect2::operator- (
    const Vect2 & other ) const
```

#### 6.16.2.11 setX()

```
void raytracer::Vect2::setX (
    const float v )
```

#### 6.16.2.12 setY()

```
void raytracer::Vect2::setY (
    const float v )
```

### 6.16.3 Friends And Related Function Documentation

#### 6.16.3.1 operator\*

```
Vect2 operator* (
    const float num,
    const Vect2 & vect ) [friend]
```

### 6.16.4 Member Data Documentation

#### 6.16.4.1 x\_

```
float raytracer::Vect2::x_ [private]
```

#### 6.16.4.2 y\_

```
float raytracer::Vect2::y_ [private]
```

The documentation for this class was generated from the following files:

- includes/raytracer/[vect.hh](#)
- src/raytracer/[vect.cc](#)

## 6.17 raytracer::Vect3 Class Reference

```
#include <vect.hh>
```

### Public Member Functions

- [Vect3](#) ()  
*Default constructor.*
- [Vect3](#) (float x, float y, float z)  
*Intuitive constructor.*
- [Vect3](#) (const [Vect3](#) &v)  
*Copy constructor.*
- float [getX](#) () const  
*Getter of x value.*
- float [getY](#) () const  
*Getter of y value.*
- float [getZ](#) () const  
*Getter of z value.*
- void [setX](#) (const float v)  
*Setter of x value.*
- void [setY](#) (const float v)  
*Setter of y value.*
- void [setZ](#) (const float v)  
*Setter of z value.*
- bool [isNullVect](#) () const  
*Check if the vector is null.*
- float [getNorm](#) () const  
*Get the norm of the vector.*
- float [dot](#) (const [Vect3](#) &other) const  
*Get the dot product of this vect with an other.*
- void [normalize](#) ()  
*Normalize the vector.*
- [Vect3 operator-](#) () const  
*Overload of the opposite operator.*
- [Vect3 operator+](#) (const [Vect3](#) &other) const  
*Overload of the addition operator between two vectors.*
- [Vect3 operator-](#) (const [Vect3](#) &other) const  
*Overload of the subtraction operator between two vectors.*
- [Vect3 operator\\*](#) (const float other) const  
*Overload of the product multiplication between a vector and a float.*

### Private Attributes

- float [x\\_](#)
- float [y\\_](#)
- float [z\\_](#)

## Friends

- [Vect3 operator\\*](#) (const float num, const [Vect3](#) &vect)  
*Overload of the product multiplication between a float and a vector.*

## 6.17.1 Constructor & Destructor Documentation

### 6.17.1.1 Vect3() [1/3]

```
raytracer::Vect3::Vect3 ( )
```

Default constructor.

### 6.17.1.2 Vect3() [2/3]

```
raytracer::Vect3::Vect3 (
    float x,
    float y,
    float z )
```

Intuitive constructor.

#### Parameters

<i>x</i>	value of the x coordinate.
<i>y</i>	value of the y coordinate.
<i>z</i>	value of the z coordinate.

### 6.17.1.3 Vect3() [3/3]

```
raytracer::Vect3::Vect3 (
    const Vect3 & v )
```

Copy constructor.

#### Parameters

<i>v</i>	vect to copy.
----------	---------------

## 6.17.2 Member Function Documentation

### 6.17.2.1 dot()

```
float raytracer::Vect3::dot (
    const Vect3 & other ) const
```

Get the dot product of this vect with an other.

#### Returns

The dot product of the two vectors.

### 6.17.2.2 getNorm()

```
float raytracer::Vect3::getNorm ( ) const
```

Get the norm of the vector.

#### Returns

The norm of the vector.

### 6.17.2.3 getX()

```
float raytracer::Vect3::getX ( ) const
```

Getter of x value.

#### Returns

Value of x value.

### 6.17.2.4 getY()

```
float raytracer::Vect3::getY ( ) const
```

Getter of y value.

#### Returns

Value of y value.

#### 6.17.2.5 getZ()

```
float raytracer::Vect3::getZ ( ) const
```

Getter of z value.

##### Returns

Value of z value.

#### 6.17.2.6 isNullVect()

```
bool raytracer::Vect3::isNullVect ( ) const
```

Check if the vector is null.

##### Returns

True if the vector is null, False otherwise.

#### 6.17.2.7 normalize()

```
void raytracer::Vect3::normalize ( )
```

Normalize the vector.

#### 6.17.2.8 operator\*()

```
raytracer::Vect3 raytracer::Vect3::operator* (
    const float other ) const
```

Overload of the product multiplication between a vector and a float.

##### Parameters

<i>other</i>	the float to multiply with this vector.
--------------	---

##### Returns

A new vector.

#### 6.17.2.9 operator+()

```
raytracer::Vect3 raytracer::Vect3::operator+ (
    const Vect3 & other ) const
```

Overload of the addition operator between two vectors.

##### Parameters

<i>other</i>	the vector to add to this one.
--------------	--------------------------------

##### Returns

A new vector.

#### 6.17.2.10 operator-() [1/2]

```
raytracer::Vect3 raytracer::Vect3::operator- ( ) const
```

Overload of the opposite operator.

##### Returns

The opposite vector of this one.

#### 6.17.2.11 operator-() [2/2]

```
raytracer::Vect3 raytracer::Vect3::operator- (
    const Vect3 & other ) const
```

Overload of the subtraction operator between two vectors.

##### Parameters

<i>other</i>	the vector to subtract to this one.
--------------	-------------------------------------

##### Returns

A new vector.



#### 6.17.2.12 setX()

```
void raytracer::Vect3::setX (
    const float v )
```

Setter of x value.

##### Parameters

<i>v</i>	value to set x with.
----------	----------------------

#### 6.17.2.13 setY()

```
void raytracer::Vect3::setY (
    const float v )
```

Setter of y value.

##### Parameters

<i>v</i>	value to set y with.
----------	----------------------

#### 6.17.2.14 setZ()

```
void raytracer::Vect3::setZ (
    const float v )
```

Setter of z value.

##### Parameters

<i>v</i>	value to set z with
----------	---------------------

### 6.17.3 Friends And Related Function Documentation

#### 6.17.3.1 operator\*

```
Vect3 operator* (
    const float num,
    const Vect3 & vect ) [friend]
```

Overload of the product multiplication between a float and a vector.

**Parameters**

<i>num</i>	the float to multiply with the vector.
<i>vect</i>	the vector to multiply with the float.

**Returns**

A new vector.

**6.17.4 Member Data Documentation****6.17.4.1 x\_**

```
float raytracer::Vect3::x_ [private]
```

the x coordinate

**6.17.4.2 y\_**

```
float raytracer::Vect3::y_ [private]
```

the y coordinate

**6.17.4.3 z\_**

```
float raytracer::Vect3::z_ [private]
```

the z coordinate

The documentation for this class was generated from the following files:

- [includes/raytracer/vect.hh](#)
- [src/raytracer/vect.cc](#)

## Chapter 7

# File Documentation

### 7.1 includes/fwd.hh File Reference

#### Namespaces

- [raytracer](#)
- [gameobj](#)

### 7.2 includes/gameobj/ambientlight.hh File Reference

```
#include "fwd.hh"
#include <vector>
#include "color.hh"
#include "lightable.hh"
#include "shapable.hh"
#include "flatshapable.hh"
```

#### Classes

- class [gameobj::AmbientLight](#)

#### Namespaces

- [gameobj](#)

### 7.3 includes/gameobj/cube.hh File Reference

```
#include "fwd.hh"
#include <vector>
#include <tuple>
#include "ray.hh"
#include "vect.hh"
#include "square.hh"
#include "shapable.hh"
#include "landmark.hh"
#include "color.hh"
#include "flatshapable.hh"
```

## Classes

- class [gameobj::Cube](#)

## Namespaces

- [gameobj](#)

## 7.4 includes/gameobj/dirlight.hh File Reference

```
#include "fwd.hh"
#include <vector>
#include "vect.hh"
#include "color.hh"
#include "shapable.hh"
#include "lightable.hh"
#include "flatshapable.hh"
```

## Classes

- class [gameobj::DirectionalLight](#)

## Namespaces

- [gameobj](#)

## 7.5 includes/gameobj/flatshapable.hh File Reference

```
#include "fwd.hh"
#include <optional>
#include <tuple>
#include "vect.hh"
#include "ray.hh"
#include "color.hh"
```

## Classes

- class [gameobj::FlatShapable](#)

## Namespaces

- [gameobj](#)

## 7.6 includes/gameobj/lightable.hh File Reference

```
#include "fwd.hh"
#include "shapable.hh"
#include "flatshapable.hh"
```

### Classes

- class [gameobj::Lightable](#)

### Namespaces

- [gameobj](#)

## 7.7 includes/gameobj/plan.hh File Reference

```
#include "fwd.hh"
#include <optional>
#include <tuple>
#include "ray.hh"
#include "vect.hh"
#include "shapable.hh"
#include "color.hh"
#include "flatshapable.hh"
```

### Classes

- class [gameobj::Plan](#)

### Namespaces

- [gameobj](#)

## 7.8 includes/gameobj/pointlight.hh File Reference

```
#include "fwd.hh"
#include <vector>
#include "vect.hh"
#include "color.hh"
#include "shapable.hh"
#include "lightable.hh"
#include "flatshapable.hh"
```

## Classes

- class [gameobj::PointLight](#)

## Namespaces

- [gameobj](#)

## 7.9 includes/gameobj/shapable.hh File Reference

```
#include "fwd.hh"
#include <optional>
#include <tuple>
#include "vect.hh"
#include "ray.hh"
#include "color.hh"
#include "shapable.hh"
#include "flatshapable.hh"
```

## Classes

- class [gameobj::Shapable](#)

## Namespaces

- [gameobj](#)

## 7.10 includes/gameobj/square.hh File Reference

```
#include "fwd.hh"
#include <optional>
#include <tuple>
#include "ray.hh"
#include "vect.hh"
#include "plan.hh"
#include "shapable.hh"
#include "color.hh"
#include "flatshapable.hh"
```

## Classes

- class [gameobj::Square](#)

## Namespaces

- [gameobj](#)

## 7.11 includes/raytracer/camera.hh File Reference

```
#include "fwd.hh"
#include "vect.hh"
```

## Classes

- class [raytracer::Camera](#)

## Namespaces

- [raytracer](#)

## 7.12 includes/raytracer/color.hh File Reference

```
#include "fwd.hh"
```

## Classes

- class [raytracer::Color](#)

## Namespaces

- [raytracer](#)

## Functions

- Color [raytracer::operator\\*](#) (const float num, const Color &color)

## 7.13 includes/raytracer/landmark.hh File Reference

```
#include "fwd.hh"
#include "vect.hh"
```

## Classes

- class [raytracer::Landmark](#)

## Namespaces

- [raytracer](#)

## 7.14 includes/raytracer/ray.hh File Reference

```
#include "fwd.hh"
#include <vector>
#include "vect.hh"
#include "shapable.hh"
```

## Classes

- class [raytracer::Ray](#)

## Namespaces

- [raytracer](#)

## 7.15 includes/raytracer/screen.hh File Reference

```
#include "fwd.hh"
#include "vect.hh"
#include "camera.hh"
```

## Classes

- class [raytracer::Screen](#)

## Namespaces

- [raytracer](#)



## 7.16 includes/raytracer/utils.hh File Reference

```
#include "fwd.hh"
#include <cmath>
#include <vector>
#include "vect.hh"
#include "ray.hh"
#include "screen.hh"
#include "camera.hh"
#include "shapable.hh"
#include "square.hh"
#include "pointlight.hh"
#include "color.hh"
#include "lightable.hh"
```

### Namespaces

- [raytracer](#)

### Functions

- void [raytracer::rotateX](#) (Vect3 &vect, const float angle)
- void [raytracer::rotateY](#) (Vect3 &vect, const float angle)
- void [raytracer::rotateZ](#) (Vect3 &vect, const float angle)
- Vect3 [raytracer::applyAngle](#) (Vect3 &vect, const Vect3 &angles)
- Vect3 [raytracer::vectorFromAngles](#) (const Vect3 &angles)
- Vect3 [raytracer::getAngles](#) (const Vect3 &v1, const Vect3 &v2)
- Vect3 [raytracer::vectorFromPoints](#) (const Vect3 &v1, const Vect3 &v2)
- std::vector< Ray > [raytracer::genRays](#) (const Camera &cam, const Screen &screen)
- int [raytracer::closerToOrigin](#) (Vect3 origin, Vect3 point1, Vect3 point2)
- std::vector< Color > [raytracer::renderFrame](#) (const std::vector< [gameobj::Shapable](#) \*> &objects, const std::vector< Ray > &rays, const std::vector< [gameobj::Lightable](#) \*> &lightList)

### Variables

- const float [raytracer::pi](#) = std::acos(-1)

## 7.17 includes/raytracer/vect.hh File Reference

```
#include "fwd.hh"
```

### Classes

- class [raytracer::Vect3](#)
- class [raytracer::Vect2](#)

## Namespaces

- [raytracer](#)

## Functions

- Vect3 [raytracer::operator\\*](#) (const float num, const Vect3 &vect)  
*Overload of the product multiplication between a float and a vector.*
- Vect2 [raytracer::operator\\*](#) (const float num, const Vect2 &vect)

## 7.18 includes/ui/gameui.hh File Reference

```
#include "fwd.hh"  
#include <vector>
```

## Classes

- class [gameui::UIScreen](#)

## Namespaces

- [gameui](#)

## 7.19 src/gameobj/ambientlight.cc File Reference

```
#include "ambientlight.hh"
```

## 7.20 src/gameobj/cube.cc File Reference

```
#include "cube.hh"  
#include "utils.hh"
```

## 7.21 src/gameobj/dirlight.cc File Reference

```
#include "dirlight.hh"  
#include "ray.hh"
```

## 7.22 src/gameobj/plan.cc File Reference

```
#include "plan.hh"
```

### Functions

- float `computeD` (const `raytracer::Vect3` &p, const `raytracer::Vect3` &n)

### 7.22.1 Function Documentation

#### 7.22.1.1 `computeD()`

```
float computeD (  
    const raytracer::Vect3 & p,  
    const raytracer::Vect3 & n )
```

## 7.23 src/gameobj/pointlight.cc File Reference

```
#include "pointlight.hh"  
#include "utils.hh"  
#include "ray.hh"
```

## 7.24 src/gameobj/square.cc File Reference

```
#include "square.hh"  
#include "utils.hh"
```

## 7.25 src/main.cc File Reference

```
#include <iostream>  
#include <vector>  
#include <cmath>  
#include <string>  
#include <fstream>  
#include "ray.hh"  
#include "camera.hh"  
#include "vect.hh"  
#include "screen.hh"  
#include "square.hh"
```

```
#include "utils.hh"
#include "cube.hh"
#include "color.hh"
#include "pointlight.hh"
#include "shapable.hh"
#include "lightable.hh"
#include "ambientlight.hh"
#include "dirlight.hh"
```

## Functions

- void `writeRender` (std::string filename, std::vector< `raytracer::Color` > &pixels, unsigned width, unsigned height)
- int `main` ()

### 7.25.1 Function Documentation

#### 7.25.1.1 `main()`

```
int main ( )
```

#### 7.25.1.2 `writeRender()`

```
void writeRender (
    std::string filename,
    std::vector< raytracer::Color > & pixels,
    unsigned width,
    unsigned height )
```

### 7.26 `src/raytracer/camera.cc` File Reference

```
#include "camera.hh"
#include "utils.hh"
```

### 7.27 `src/raytracer/color.cc` File Reference

```
#include "color.hh"
```

## 7.28 src/raytracer/landmark.cc File Reference

```
#include "landmark.hh"
```

## 7.29 src/raytracer/ray.cc File Reference

```
#include "ray.hh"  
#include <tuple>  
#include <optional>  
#include "color.hh"  
#include "utils.hh"
```

## 7.30 src/raytracer/screen.cc File Reference

```
#include "screen.hh"  
#include <cmath>  
#include "utils.hh"
```

## 7.31 src/raytracer/utils.cc File Reference

```
#include "utils.hh"  
#include <optional>  
#include <tuple>  
#include "flatshapable.hh"
```

## 7.32 src/raytracer/vect.cc File Reference

```
#include "vect.hh"  
#include <cmath>
```

## 7.33 src/ui/gameui.cc File Reference



# Index

AmbientLight  
  gameobj::AmbientLight, [13](#)  
angleVect\_  
  raytracer::Camera, [16](#)  
angles\_  
  gameobj::Cube, [22](#)  
applyAngle  
  raytracer, [10](#)  
  
b\_  
  raytracer::Color, [20](#)  
brightness\_  
  gameobj::AmbientLight, [15](#)  
  gameobj::DirectionalLight, [25](#)  
  gameobj::PointLight, [35](#)  
  
Camera  
  raytracer::Camera, [15](#)  
center\_  
  gameobj::Cube, [22](#)  
  raytracer::Screen, [39](#)  
closerToOrigin  
  raytracer, [10](#)  
colides  
  raytracer::Ray, [36](#)  
colidesBefore  
  raytracer::Ray, [36](#)  
Color  
  raytracer::Color, [17](#), [18](#)  
color\_  
  gameobj::AmbientLight, [15](#)  
  gameobj::DirectionalLight, [25](#)  
  gameobj::Plan, [32](#)  
  gameobj::PointLight, [35](#)  
computeD  
  plan.cc, [63](#)  
Cube  
  gameobj::Cube, [21](#)  
  
d\_  
  gameobj::Plan, [32](#)  
dir\_  
  gameobj::DirectionalLight, [25](#)  
DirectionalLight  
  gameobj::DirectionalLight, [24](#)  
dot  
  raytracer::Vect2, [45](#)  
  raytracer::Vect3, [50](#)  
  
g\_  
  raytracer::Color, [20](#)

gameobj, [9](#)  
gameobj::AmbientLight, [13](#)  
  AmbientLight, [13](#)  
  brightness\_, [15](#)  
  color\_, [15](#)  
  getBrightness, [14](#)  
  getColor, [14](#)  
  interact, [14](#)  
  setBrightness, [14](#)  
  setColor, [14](#)  
gameobj::Cube, [20](#)  
  angles\_, [22](#)  
  center\_, [22](#)  
  Cube, [21](#)  
  getAngles, [21](#)  
  getCenter, [21](#)  
  getColor, [21](#)  
  getLandmark, [22](#)  
  getSide, [22](#)  
  intersecte, [22](#)  
  landmark\_, [22](#)  
  side\_, [23](#)  
  squares\_, [23](#)  
gameobj::DirectionalLight, [23](#)  
  brightness\_, [25](#)  
  color\_, [25](#)  
  dir\_, [25](#)  
  DirectionalLight, [24](#)  
  getBrightness, [24](#)  
  getColor, [24](#)  
  getDir, [24](#)  
  interact, [24](#)  
  setBrightness, [24](#)  
  setColor, [25](#)  
  setDir, [25](#)  
gameobj::FlatShapable, [26](#)  
  getColor, [26](#)  
  intersecte, [26](#)  
gameobj::Lightable, [29](#)  
  getBrightness, [29](#)  
  getColor, [29](#)  
  interact, [30](#)  
  setBrightness, [30](#)  
  setColor, [30](#)  
gameobj::Plan, [30](#)  
  color\_, [32](#)  
  d\_, [32](#)  
  getColor, [31](#)

- getNorm, 32
- getD, 32
- intersecte, 32
- norm\_, 33
- Plan, 31
- setColor, 32
- gameobj::PointLight, 33
  - brightness\_, 35
  - color\_, 35
  - getBrightness, 34
  - getColor, 34
  - getPos, 34
  - interact, 34
  - PointLight, 33
  - pos\_, 35
  - setBrightness, 34
  - setColor, 34
  - setPos, 35
- gameobj::Shapable, 40
  - getColor, 40
  - intersecte, 40
- gameobj::Square, 41
  - getAB, 42
  - getAD, 42
  - getColor, 42
  - getPlan, 42
  - getPos, 42
  - intersecte, 42
  - p\_, 43
  - pA\_, 43
  - Square, 41
  - vAB\_, 43
  - vAD\_, 43
- gameui, 9
- gameui::UIScreen, 43
  - sceneRender\_, 44
  - UIScreen, 44
- genRays
  - raytracer, 10
- getAB
  - gameobj::Square, 42
- getAD
  - gameobj::Square, 42
- getAng
  - raytracer::Camera, 16
- getAngles
  - gameobj::Cube, 21
  - raytracer, 10
- getBrightness
  - gameobj::AmbientLight, 14
  - gameobj::DirectionalLight, 24
  - gameobj::Lightable, 29
  - gameobj::PointLight, 34
- getCenter
  - gameobj::Cube, 21
  - raytracer::Screen, 38
- getColor
  - gameobj::AmbientLight, 14
  - gameobj::Cube, 21
  - gameobj::DirectionalLight, 24
  - gameobj::FlatShapable, 26
  - gameobj::Lightable, 29
  - gameobj::Plan, 31
  - gameobj::PointLight, 34
  - gameobj::Shapable, 40
  - gameobj::Square, 42
- getDir
  - gameobj::DirectionalLight, 24
- getHeight
  - raytracer::Screen, 38
- getLandmark
  - gameobj::Cube, 22
- getNorm
  - gameobj::Plan, 32
  - raytracer::Vect2, 45
  - raytracer::Vect3, 50
- getOrigin
  - raytracer::Ray, 36
- getPixelSize
  - raytracer::Screen, 38
- getPlan
  - gameobj::Square, 42
- getPos
  - gameobj::PointLight, 34
  - gameobj::Square, 42
  - raytracer::Camera, 16
- getSide
  - gameobj::Cube, 22
- getVect
  - raytracer::Ray, 36
- getViewAngle
  - raytracer::Camera, 16
- getWidth
  - raytracer::Screen, 38
- getB
  - raytracer::Color, 18
- getD
  - gameobj::Plan, 32
- getG
  - raytracer::Color, 18
- getI
  - raytracer::Screen, 38
- getJ
  - raytracer::Screen, 38
- getO
  - raytracer::Landmark, 27
- getR
  - raytracer::Color, 18
- getX
  - raytracer::Landmark, 27
  - raytracer::Vect2, 45
  - raytracer::Vect3, 50
- getY
  - raytracer::Landmark, 28
  - raytracer::Vect2, 46
  - raytracer::Vect3, 50



- getZ
  - raytracer::Landmark, 28
  - raytracer::Vect3, 50
- height\_
  - raytracer::Screen, 39
- i\_
  - raytracer::Screen, 39
- includes/fwd.hh, 55
- includes/gameobj/ambientlight.hh, 55
- includes/gameobj/cube.hh, 55
- includes/gameobj/dirlight.hh, 56
- includes/gameobj/flatshapable.hh, 56
- includes/gameobj/lighttable.hh, 57
- includes/gameobj/plan.hh, 57
- includes/gameobj/pointlight.hh, 57
- includes/gameobj/shapable.hh, 58
- includes/gameobj/square.hh, 58
- includes/raytracer/camera.hh, 59
- includes/raytracer/color.hh, 59
- includes/raytracer/landmark.hh, 59
- includes/raytracer/ray.hh, 60
- includes/raytracer/screen.hh, 60
- includes/raytracer/utils.hh, 61
- includes/raytracer/vect.hh, 61
- includes/ui/gameui.hh, 62
- interact
  - gameobj::AmbientLight, 14
  - gameobj::DirectionalLight, 24
  - gameobj::Lighttable, 30
  - gameobj::PointLight, 34
- intersecte
  - gameobj::Cube, 22
  - gameobj::FlatShapable, 26
  - gameobj::Plan, 32
  - gameobj::Shapable, 40
  - gameobj::Square, 42
- isNullVect
  - raytracer::Vect2, 46
  - raytracer::Vect3, 51
- j\_
  - raytracer::Screen, 39
- Landmark
  - raytracer::Landmark, 27
- landmark\_
  - gameobj::Cube, 22
- main
  - main.cc, 64
- main.cc
  - main, 64
  - writeRender, 64
- norm\_
  - gameobj::Plan, 33
- normalize
  - raytracer::Color, 18
  - raytracer::Vect2, 46
  - raytracer::Vect3, 51
- o\_
  - raytracer::Landmark, 28
- operator\*
  - raytracer, 11
  - raytracer::Color, 18, 20
  - raytracer::Vect2, 46, 47
  - raytracer::Vect3, 51, 53
- operator+
  - raytracer::Color, 19
  - raytracer::Vect2, 46
  - raytracer::Vect3, 51
- operator+=
  - raytracer::Color, 19
- operator-
  - raytracer::Color, 19
  - raytracer::Vect2, 46
  - raytracer::Vect3, 52
- origin\_
  - raytracer::Ray, 37
- p\_
  - gameobj::Square, 43
- pA\_
  - gameobj::Square, 43
- pi
  - raytracer, 12
- pixelSize\_
  - raytracer::Screen, 39
- Plan
  - gameobj::Plan, 31
- plan.cc
  - computeD, 63
- PointLight
  - gameobj::PointLight, 33
- pos\_
  - gameobj::PointLight, 35
  - raytracer::Camera, 16
- r\_
  - raytracer::Color, 20
- Ray
  - raytracer::Ray, 36
- raytracer, 9
  - applyAngle, 10
  - closerToOrigin, 10
  - genRays, 10
  - getAngles, 10
  - operator\*, 11
  - pi, 12
  - renderFrame, 11
  - rotateX, 11
  - rotateY, 12
  - rotateZ, 12
  - vectorFromAngles, 12
  - vectorFromPoints, 12
- raytracer::Camera, 15

- angleVect\_, 16
- Camera, 15
- getAng, 16
- getPos, 16
- getViewAngle, 16
- pos\_, 16
- viewAngle\_, 16
- raytracer::Color, 17
  - b\_, 20
  - Color, 17, 18
  - g\_, 20
  - getB, 18
  - getG, 18
  - getR, 18
  - normalize, 18
  - operator\*, 18, 20
  - operator+, 19
  - operator+=, 19
  - operator-, 19
  - r\_, 20
  - setB, 19
  - setG, 19
  - setR, 19
- raytracer::Landmark, 26
  - getO, 27
  - getX, 27
  - getY, 28
  - getZ, 28
  - Landmark, 27
  - o\_, 28
  - transposePoint, 28
  - transposeVect, 28
  - x\_, 28
  - y\_, 28
  - z\_, 29
- raytracer::Ray, 35
  - colides, 36
  - colidesBefore, 36
  - getOrigin, 36
  - getVect, 36
  - origin\_, 37
  - Ray, 36
  - vect\_, 37
- raytracer::Screen, 37
  - center\_, 39
  - getCenter, 38
  - getHeight, 38
  - getPixelSize, 38
  - getWidth, 38
  - getI, 38
  - getJ, 38
  - height\_, 39
  - i\_, 39
  - j\_, 39
  - pixelSize\_, 39
  - Screen, 38
  - width\_, 39
- raytracer::Vect2, 44
  - dot, 45
  - getNorm, 45
  - getX, 45
  - getY, 46
  - isNullVect, 46
  - normalize, 46
  - operator\*, 46, 47
  - operator+, 46
  - operator-, 46
  - setX, 47
  - setY, 47
  - Vect2, 45
  - x\_, 47
  - y\_, 47
- raytracer::Vect3, 48
  - dot, 50
  - getNorm, 50
  - getX, 50
  - getY, 50
  - getZ, 50
  - isNullVect, 51
  - normalize, 51
  - operator\*, 51, 53
  - operator+, 51
  - operator-, 52
  - setX, 52
  - setY, 53
  - setZ, 53
  - Vect3, 49
  - x\_, 54
  - y\_, 54
  - z\_, 54
- renderFrame
  - raytracer, 11
- rotateX
  - raytracer, 11
- rotateY
  - raytracer, 12
- rotateZ
  - raytracer, 12
- sceneRender\_
  - gameui::UiScreen, 44
- Screen
  - raytracer::Screen, 38
- setBrightness
  - gameobj::AmbientLight, 14
  - gameobj::DirectionalLight, 24
  - gameobj::Lightable, 30
  - gameobj::PointLight, 34
- setColor
  - gameobj::AmbientLight, 14
  - gameobj::DirectionalLight, 25
  - gameobj::Lightable, 30
  - gameobj::Plan, 32
  - gameobj::PointLight, 34
- setDir
  - gameobj::DirectionalLight, 25
- setPos

- gameobj::PointLight, [35](#)
- setB
  - raytracer::Color, [19](#)
- setG
  - raytracer::Color, [19](#)
- setR
  - raytracer::Color, [19](#)
- setX
  - raytracer::Vect2, [47](#)
  - raytracer::Vect3, [52](#)
- setY
  - raytracer::Vect2, [47](#)
  - raytracer::Vect3, [53](#)
- setZ
  - raytracer::Vect3, [53](#)
- side\_
  - gameobj::Cube, [23](#)
- Square
  - gameobj::Square, [41](#)
- squares\_
  - gameobj::Cube, [23](#)
- src/gameobj/ambientlight.cc, [62](#)
- src/gameobj/cube.cc, [62](#)
- src/gameobj/dirlight.cc, [62](#)
- src/gameobj/plan.cc, [63](#)
- src/gameobj/pointlight.cc, [63](#)
- src/gameobj/square.cc, [63](#)
- src/main.cc, [63](#)
- src/raytracer/camera.cc, [64](#)
- src/raytracer/color.cc, [64](#)
- src/raytracer/landmark.cc, [65](#)
- src/raytracer/ray.cc, [65](#)
- src/raytracer/screen.cc, [65](#)
- src/raytracer/utils.cc, [65](#)
- src/raytracer/vect.cc, [65](#)
- src/ui/gameui.cc, [65](#)
- transposePoint
  - raytracer::Landmark, [28](#)
- transposeVect
  - raytracer::Landmark, [28](#)
- UIScreen
  - gameui::UIScreen, [44](#)
- vAB\_
  - gameobj::Square, [43](#)
- vAD\_
  - gameobj::Square, [43](#)
- Vect2
  - raytracer::Vect2, [45](#)
- Vect3
  - raytracer::Vect3, [49](#)
- vect\_
  - raytracer::Ray, [37](#)
- vectorFromAngles
  - raytracer, [12](#)
- vectorFromPoints
  - raytracer, [12](#)
- viewAngle\_
  - raytracer::Camera, [16](#)
- width\_
  - raytracer::Screen, [39](#)
- writeRender
  - main.cc, [64](#)
- x\_
  - raytracer::Landmark, [28](#)
  - raytracer::Vect2, [47](#)
  - raytracer::Vect3, [54](#)
- y\_
  - raytracer::Landmark, [28](#)
  - raytracer::Vect2, [47](#)
  - raytracer::Vect3, [54](#)
- z\_
  - raytracer::Landmark, [29](#)
  - raytracer::Vect3, [54](#)