Raytracer Project

0.1

Generated by Doxygen 1.8.14

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

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

num	the float to multiply with the vector.
vect	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)
```

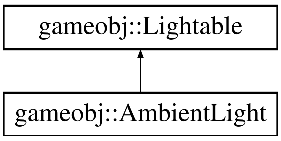
Chapter 6

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]
```

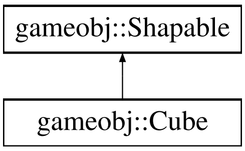
The documentation for this class was generated from the following files:

- 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]
```

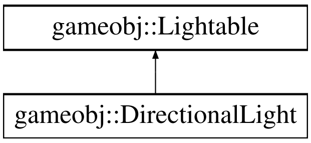
The documentation for this class was generated from the following files:

- 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]
```

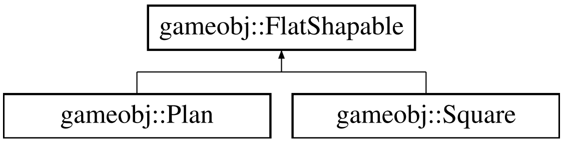
The documentation for this class was generated from the following files:

- 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]
```

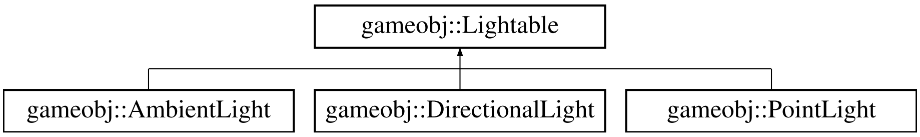
The documentation for this class was generated from the following files:

- 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 )
```

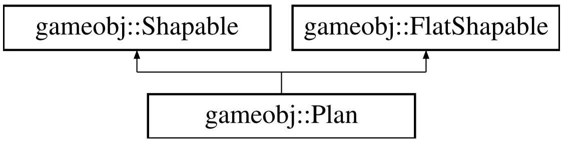
The documentation for this class was generated from the following file:

- 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]
```

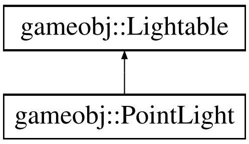
The documentation for this class was generated from the following files:

- 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 & 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]
```

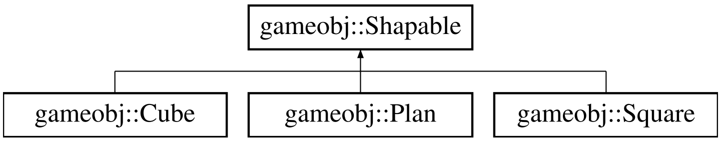
The documentation for this class was generated from the following files:

- 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.

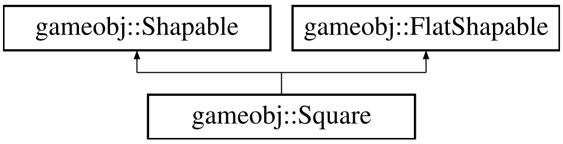
The documentation for this class was generated from the following file:

- 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-() [1/2]

```
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

x	value of the x coordinate.
y	value of the y coordinate.
z	value of the z coordinate.

6.17.1.3 Vect3() [3/3]

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

Copy constructor.

Parameters

v	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

other	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

other	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

other	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

v	value to set x with.

6.17.2.13 setY()

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

Setter of y value.

Parameters

v	value to set y with.

6.17.2.14 setZ()

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

Setter of z value.

Parameters

v	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

num	the float to multiply with the vector.
vect	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