# Fotorealistyczna Grafika Komputerowa

## Animacja, motion blur

Adam Błaszczyk 239636
Antonina Matuszek 239687

# Chapter 1

# Hierarchical Index

## 1.1 Class Hierarchy

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

# Chapter 2

# Class Index

## 2.1 Class List

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

# Chapter 3

# Class Documentation

## 3.1 hit.Hit Class Reference

Documentation for a class Hit.

### Public Member Functions

- def __init__ (self, point, distance, color, primitive)

  *Constructor.*

### Public Attributes

- **point**
- **distance**
- **color**
- **primitive**

### 3.1.1 Detailed Description

Documentation for a class Hit.

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

- hit.py

## 3.2 light_intensity.LightIntensity Class Reference

Documentation for a class LightIntensity.

### Public Member Functions

- def **__init__** (self, color=[0, 0, 0])
- def **__add__** (self, other)
- def **__truediv__** (self, other)

### Static Public Member Functions

- def clamp01 (value)

  *Clamps value between 0 and 1.*
- def clamp_0_255 (value)

  *Clamps value between 0 and 255.*
- def remap_0_255 (value)

  *Remapsvalue from 0-1 to 0-255.*
- def clamp_color (color)

  *Clamps color to 0-1.*

### Public Attributes

- **color**

#### 3.2.1 Detailed Description

Documentation for a class LightIntensity.

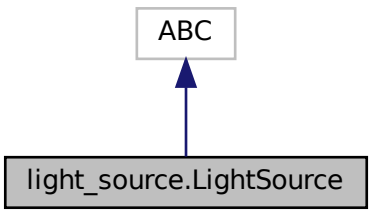
The documentation for this class was generated from the following file:

- light_intensity.py

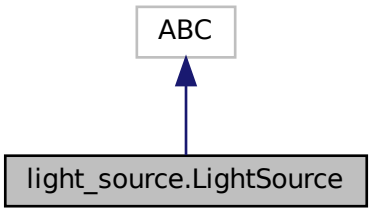
## 3.3 light_source.LightSource Class Reference

Documentation for a class LightSource.

Inheritance diagram for light_source.LightSource:



Collaboration diagram for light_source.LightSource:




## Public Member Functions

- def __init__ (self, position=Vec3(0, 0, 0), color=[1, 1, 1], intensity=1)
  *The constructor.*
- def __str__ (self)
  *Function returning object values in string format.*

## Public Attributes

- color
  *Colour of light source.*
- position
  *Position of light source.*
- intensity
  *Intensity of light.*

### 3.3.1 Detailed Description

Documentation for a class LightSource.

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 __init__()

```
def light_source.LightSource.__init__ (
              self,
              position = Vec3(0, 0, 0),
              color = [1, 1, 1],
              intensity = 1 )
```

The constructor.

Creates a LightSource with a specified Colour at a given Location.

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

- light_source.py

## 3.4 material.Material Class Reference

Documentation for a class Material.

### Public Member Functions

- def __init__ (self, ambientColour=(1, 1, 1), diffuseColour=(1, 1, 1), reflectColour=(1, 1, 1), specularColour=(0, 0, 0), specularExponent=1, mirror_reflection_coefficient=1, diffuse_reflection_coefficient=1, index_of_↩ refraction=1, material_type=MaterialType.Dull, texture=None)

  *The constructor.*
- def __str__ (self)

  *Function returning object values in string format.*

### Public Attributes

- ambientColour

  *Colour of Material under white ambient light.*
- diffuseColour

  *Colour of Material under direct white light.*
- reflectColour

  *Colour of reflected rays under direct white light.*
- specularColour

  *Colour of Material's specular highlights.*
- specularExponent

  *'Hardness' of Material's specular hightlights - high values give small, sharp highlights.*
- **mirror_reflection_coefficient**
- **diffuse_reflection_coefficient**
- **index_of_refraction**
- **material_type**
- **texture**

### 3.4.1 Detailed Description

Documentation for a class Material.

### 3.4.2 Member Data Documentation

#### 3.4.2.1 ambientColour

`material.Material.ambientColour`

Colour of Material under white ambient light.

Usually, but not always, the same as diffuseColour.

#### 3.4.2.2 diffuseColour

`material.Material.diffuseColour`

Colour of Material under direct white light.

Usually, but not always, the same as ambientColour.

#### 3.4.2.3 reflectColour

`material.Material.reflectColour`

Colour of reflected rays under direct white light.

If this is zero then there are no reflections.

#### 3.4.2.4 specularColour

`material.Material.specularColour`

Colour of Material's specular highlights.

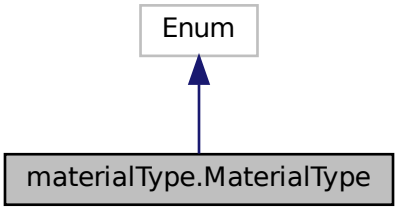
If this is zero then there are no highlights.

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

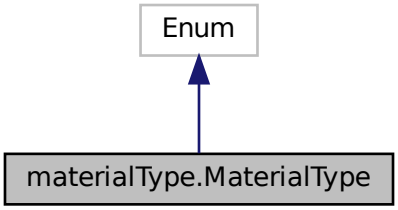
- material.py

## 3.5 materialType.MaterialType Class Reference

Inheritance diagram for materialType.MaterialType:



Collaboration diagram for materialType.MaterialType:




### Static Public Attributes

- int **Dull** = 1
- int **Reflective** = 2

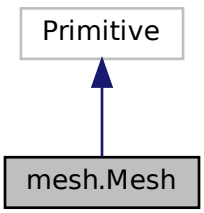
The documentation for this class was generated from the following file:

- materialType.py

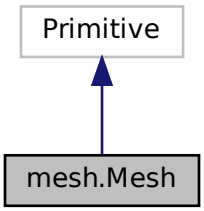
## 3.6 mesh.Mesh Class Reference

Documentation for a class Mesh.

Inheritance diagram for mesh.Mesh:



Collaboration diagram for mesh.Mesh:




### Public Member Functions

- def __init__ (self, obj_file, position=Vec3(), color=[1, 1, 1], material=None)

  *Constructor.*
- def get_detailed_intersections (self, ray)

  *Checks if ray intersects with mesh and returns list of hits.*
- def get_detailed_intersection (self, ray)

  *Checks if ray intersects with mesh and returns hit closest to ray origin.*
- def get_intersection (self, ray)

  *Function returning intersection point.*
- def get_normal (self, point)

  *Gets normal - does nothing for mesh, has to be invoked for triangle.*
- def get_texture_color (self, coords)

  *Gets texture color in given point.*

### Public Attributes

- **triangles**

### 3.6.1 Detailed Description

Documentation for a class Mesh.

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

- mesh.py

## 3.7 image.MyImage Class Reference

Documentation for a class MyImage.

### Public Member Functions

- def __init__ (self, width=500, height=500)

  *The constructor.*
- def len (self)

  *Function returning image length.*
- def clear_color (self, rgb_color)

  *Function setting background color.*
- def fancy_background (self)

  *Function setting background color.*
- def set_pixel (self, i, j, value)

  *Function changing pixel color.*
- def get_pixel_color (self, i, j)

  *Function getting pixel color.*
- def save_image (self, filename)

  *Function saving image to png format.*

### Public Attributes

- **width**
- **height**
- **image_matrix**

### 3.7.1 Detailed Description

Documentation for a class MyImage.

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

- image.py

## 3.8 orthogonal_camera.OrthogonalCamera Class Reference

Class for othogonal camera.

### Public Member Functions

- def __init__ (self, position=Vec3(0, 0, 0), view_direction=Vec3(0, 0, 1), width=512, height=512, pixel_size=(0.01, 0.01))

  *Constructor.*
- def render_scene (self, primitives)

  *Function rendering the scene.*

### Public Attributes

- position

  *Position of the camera.*
- view_direction

  *Direction camera is facing.*
- w

  *Width in pixels.*
- h

  *Height in pixels.*
- wh_ratio

  *Width-height raio.*
- hw_ratio

  *Height-width ratio.*
- arRay

  *Array of rays.*
- x_angle

  *Angle between view direction vector and X axis.*
- y_angle

  *Angle between view direction vector and Y axis.*
- z_angle

  *Angle between view direction vector and Z axis.*

#### 3.8.1 Detailed Description

Class for othogonal camera.

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

- orthogonal_camera.py

## 3.9 perspective_camera.PerspectiveCamera Class Reference

Class for othogonal camera.

### Public Member Functions

- def __init__ (self, position=Vec3(0, 0, 0), view_direction=Vec3(0, 0, 1), width=512, height=512, near=.1, far=1000, fov=60)

  *Constructor.*
- def render_part_of_scene (self, scene, shutter_exposure_timeframe, blur_ratio, antialiasing, start_row, start_col, rows, cols, lower_left_corner, pixel_horizontal, pixel_vertical, returned_img, process_nr)

  *Function rendering part of scene to run in parallel thread/process.*
- def render_scene (self, shutter_exposure_timeframe, blur_ratio=1, antialiasing=True, amount_of_↩ processes=1)

  *Function rendering the scene.*

### Static Public Member Functions

- def adaptive_antialiasing (ray, A, B, C, D, E, depth, max_depth, horizontal, vertical, background_color, scene, shutter_exposure_timeframe)

  *Function calculating color of pixel using adaptive antialiasing.*

### Public Attributes

- position

  *Position of the camera.*
- view_direction

  *Direction camera is facing.*
- width

  *Width in pixels.*
- height

  *Height in pixels.*
- near

  *Near clipping plane.*
- far

  *Far clipping plane.*
- fov

  *Field of View.*
- up

  *Vector direction aligned with the "up" direction of camera.*
- **last_percent**
- number_of_pixels

  *Coordinates viewPlane.*
- **current_px**

#### 3.9.1 Detailed Description

Class for othogonal camera.

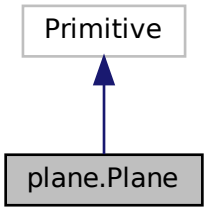
The documentation for this class was generated from the following file:

- perspective_camera.py

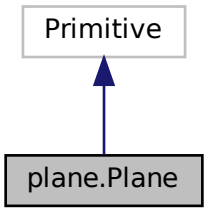
## 3.10 plane.Plane Class Reference

Documentation for a class Plane.

Inheritance diagram for plane.Plane:




Collaboration diagram for plane.Plane:




## Public Member Functions

- def __init__ (self, normal_vector, d, color=[1, 0, 1], material=None)

  *Constructor.*
- def __str__ (self)

  *Function returning object values in string format.*
- def get_detailed_intersections (self, ray)

  *Wrapper.*
- def get_detailed_intersection (self, ray)

  *Returns tuple with multiple data: point (None if no intersection), distance to point, color.*
- def get_intersection (self, ray)

  *Checks if plane and ray intersect witch each other and returns intersection point if they do, otherwise None.*
- def get_normal (self, point)

  *Returns normal for point.*
- def get_texture_color (self, coords)

  *Returns texture color for point.*

### Public Attributes

- normal_vector

  *Vector perpendicular to plane.*
- a

  *Represents A in 'Ax + By + Cz D = 0' equation.*
- b

  *Represents B in 'Ax + By + Cz D = 0' equation.*
- c

  *Represents C in 'Ax + By + Cz D = 0' equation.*
- d

  *Represents D in 'Ax + By + Cz D = 0' equation.*

### 3.10.1 Detailed Description

Documentation for a class Plane.

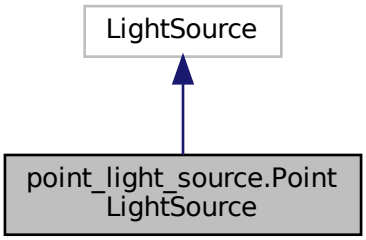
The documentation for this class was generated from the following file:

- plane.py

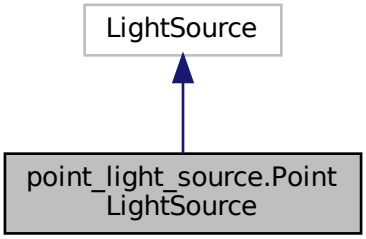
## 3.11 point_light_source.PointLightSource Class Reference

Documentation for a class PointLightSource.

Inheritance diagram for point_light_source.PointLightSource:



Collaboration diagram for point_light_source.PointLightSource:




### Public Member Functions

- def **__init__** (self, position=[0, 0, 0], color=[1, 1, 1], intensity=1)

### 3.11.1 Detailed Description

Documentation for a class PointLightSource.

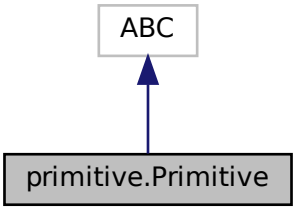
Light emitted from a Point.

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

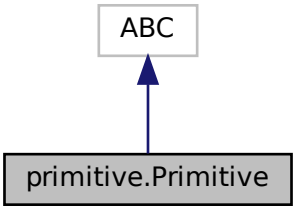
- point_light_source.py

## 3.12 primitive.Primitive Class Reference

Inheritance diagram for primitive.Primitive:



Collaboration diagram for primitive.Primitive:




### Public Member Functions

- def **__init__** (self, color, material=None)
- def **get_intersection** (self, ray)
- def **get_detailed_intersection** (self, ray)
- def **get_detailed_intersections** (self, ray)
- def **get_normal** (self, point)
- def **get_texture_color** (self, point)

### Public Attributes

- **color**
- **material**

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

- primitive.py

## 3.13 ray.Ray Class Reference

Documentation for a class Ray.

### Public Member Functions

- def __init__ (self, origin=Vec3(0, 0, 0), direction=None, target=None, length=math.inf, medium=None)

  *Constructor.*
- def __str__ (self)

  *Function returning object values in string format.*
- def is_point_on_ray (self, point)

  *Check if point is on ray, returns true if yes, false otherwise.*
- def set_direction (self, new_direction)

  *Sets new direction vector and converts it to normalized vector.*

- def set_target (self, new_target)

  *Sets new target and updates direction vector.*
- def get_plane_intersection (self, plane)

  *Plane.get_intersection(ray) wrapper.*
- def get_sphere_intersection (self, sphere)

  *Sphere.get_intersection(ray) wrapper.*
- def get_sphere_intersections (self, sphere)

  *Sphere.get_ray_intersections(ray) wrapper.*
- def get_medium_refraction_index (self)

  *Returns medium index of refraction.*
- def get_pixel_hit (self, primitives)

  *Iterates through list of primitives and returns the closest hit, raytracing step 2.*
- def get_pixel_color (self, scene=None, shutter_exposure_time=None, primitives=None, lights=None, recursion_number=0)

  *Iterates through list of primitives and lights and calculates pixel color.*
- def get_avg_pixel_color (self, scene=None, shutter_exposure_timeframe=None, amount_of_samples=3)

  *Gets average pixel color in given shutter exposure time and given amount of samples.*
- def check_intersection (self, primitives)

  *Checks if ray intersects with any given primitive.*
- def check_light_intersection (self, primitives)

  *Checks if ray intersects with any given primitive, ignores primitives with refractive material.*

## Public Attributes

- origin

  *Origin vector of a given ray.*
- direction

  *Direction vector of a given ray.*
- target

  *Target point of a given ray.*
- length

  *Length of a given ray.*
- medium

  *The medium in which the ray propagates.*

### 3.13.1 Detailed Description

Documentation for a class Ray.

### 3.13.2 Member Data Documentation

#### 3.13.2.1 direction

`ray.Ray.direction`

Direction vector of a given ray.

Cannot be (0, 0, 0).

#### 3.13.2.2 length

`ray.Ray.length`

Length of a given ray.

Default = Infinity

#### 3.13.2.3 origin

`ray.Ray.origin`

Origin vector of a given ray.

Default = (0, 0, 0)

#### 3.13.2.4 target

`ray.Ray.target`

Target point of a given ray.

Cannot be the same as origin.

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

- ray.py

## 3.14 Scene.Scene Class Reference

### Public Member Functions

- def **__init__** (self)
- def get_scene_for_time (self, i)

  *Returns scene state for given time from 0-1 range.*

### Public Attributes

- **primitives**
- **lights**

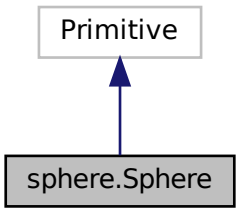
The documentation for this class was generated from the following file:

- Scene.py

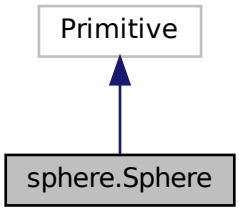
## 3.15 sphere.Sphere Class Reference

Documentation for a class Sphere.

Inheritance diagram for sphere.Sphere:




Collaboration diagram for sphere.Sphere:




### Public Member Functions

- def __init__ (self, position=Vec3(0, 0, 0), radius=1, color=[1, 0, 1], material=None)

  *Constructor.*
- def change_radius (self, new_radius)

  *Sets radius and recalculates area and volume.*
- def __str__ (self)

  *Function returning object values in string format.*
- def get_detailed_intersections (self, ray, invert_dir=False)

  *Checks if ray intersects with sphere and returns list of hits.*
- def get_detailed_intersection (self, ray)

  *Function returning hit.*
- def get_intersection (self, ray)

  *Checks if ray intersects with sphere and returns point closest to ray origin.*
- def get_normal (self, point)

  *Gets normal for given point.*
- def get_texture_color (self, coords)

  *Gets pixel color from material texture.*

### Public Attributes

- position
  *Centre of the sphere.*
- **color**
- radius
  *Radius of the sphere.*
- area
  *Area of the sphere.*
- volume
  *Volume of the sphere.*

### 3.15.1 Detailed Description

Documentation for a class Sphere.

### 3.15.2 Member Function Documentation

#### 3.15.2.1 get_texture_color()

```
def sphere.Sphere.get_texture_color (
              self,
              coords )
```

Gets pixel color from material texture.

If texture or material is None than return privmitive color

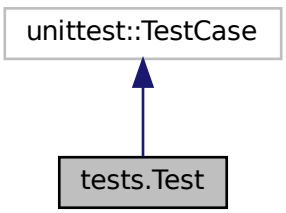
The documentation for this class was generated from the following file:

- sphere.py

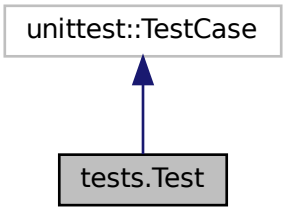
## 3.16 tests.Test Class Reference

Documentation for a class Test.

Inheritance diagram for tests.Test:



Collaboration diagram for tests.Test:




## Public Member Functions

- def **setUp** (self)
- def test_add (self)

  *Vector tests.*
- def test_sub (self)

  *Vector tests.*
- def test_pos (self)

  *Vector tests.*
- def test_neg (self)

  *Vector tests.*
- def test_length (self)

  *Vector tests.*
- def test___truediv__ (self)

  *Vector tests.*
- def test_mul (self)

  *Vector tests.*
- def test_cross (self)

  *Vector tests.*
- def test_point_on_line (self)

  *Ray tests.*
- def test_plane_intersection (self)

  *Plane tests.*
- def test_get_centre (self)

  *Sphere tests.*
- def test_get_radius (self)

  *Sphere tests.*
- def test_surface_area (self)

  *Sphere tests.*
- def test_get_volume (self)

  *Sphere tests.*
- def test_get_sphere_intersection (self)

  *Sphere tests.*
- def test_clamp_0_255 (self)

  *Light intensity tests.*

### Public Attributes

- **v1**
- **v2**
- **v3**
- **v4**
- **v5**
- **v6**
- **v7**
- **r1**
- **r2**
- **r3**
- **r4**
- **p1**
- **p2**
- **s1**
- **s2**
- **li1**
- **li2**
- **li3**

### 3.16.1 Detailed Description

Documentation for a class Test.

Unit tests.

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

- tests.py

## 3.17 texture.Texture Class Reference

Documentation for a class Texture.

### Public Member Functions

- None __init__ (self, file_name)

  *Constructor.*
- def rectangular_mapping (self, coords)

  *Returns pixel color for primitive and ray intersection represented by coords for rectangulars.*
- def spherical_mapping (self, coords, r)

  *Returns pixel color for primitive and ray intersection represented by coords for spheres.*
- str **__str__** (self)

### Public Attributes

- **file_name**
- **img**
- **height**
- **width**

### 3.17.1 Detailed Description

Documentation for a class Texture.

### 3.17.2 Member Function Documentation

#### 3.17.2.1 spherical_mapping()

```
def texture.Texture.spherical_mapping (
              self,
              coords,
              r )
```

Returns pixel color for primitive and ray intersection represented by coords for spheres.

Needs a spehere radius to scale properly

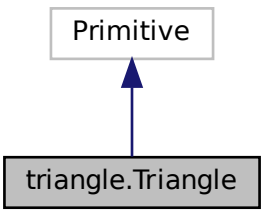
The documentation for this class was generated from the following file:

- texture.py

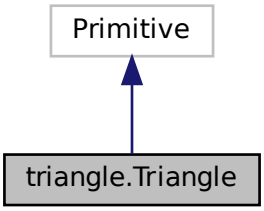
## 3.18 triangle.Triangle Class Reference

Documentation for a class Triangle.

Inheritance diagram for triangle.Triangle:



Collaboration diagram for triangle.Triangle:




## Public Member Functions

- def __init__ (self, v1=Vec3(0, 0, 0), v2=Vec3(0, 0, 0), v3=Vec3(0, 0, 0), color=[1, 0, 1], material=None)

  *Constructor.*
- def __str__ (self)

  *Function returning object values in string format.*
- def get_detailed_intersections (self, ray)

  *Checks if ray intersects with triangle and returns hit in form of a list.*
- def get_detailed_intersection (self, ray)

  *Checks if ray intercescts with triangle and returns hit.*
- def get_intersection (self, ray)

  *Checks if ray intersects with triangle and return intersection point.*
- def get_normal (self, point)

  *Returns normal for point.*
- def get_texture_color (self, coords)

  *Returns texture color for point.*

## Public Attributes

- v1

  *Triangle vertex.*
- v2

  *Triangle vertex.*
- v3

  *Triangle vertex.*
- color

  *Color of triangle.*
- normal_vector

  *Vector perpendicular to plane.*

### 3.18.1 Detailed Description

Documentation for a class Triangle.

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

- triangle.py

## 3.19 vector.Vec2 Class Reference

Documentation for a class Vec2.

### Public Member Functions

- def __init__ (self, x, y)

  *The constructor.*
- def __add__ (self, other)

  *Function returning sum of two vectors or sum of a vector and a scalar.*
- def __iadd__ (self, other)

  *Function returning sum (In-place addition) of two vectors or sum of a vector and a scalar.*
- def __sub__ (self, other)

  *Function returning difference of two vectors or difference of a vector and a scalar.*
- def __isub__ (self, other)

  *Function returning difference (In-place Subtraction) of two vectors or difference of a vector and a scalar.*
- def __eq__ (self, other)

  *Function "equal".*
- def __abs__ (self)

  *Function returning absolute value of a given vector.*
- def __ne__ (self, other)

  *Function "not equal".*
- def __neg__ (self)

  *Function negating vector coordinates.*
- def __pos__ (self)

  *Function for positive vector coordinates.*
- def __str__ (self)

  *Function returning object values in string format.*
- def length (self)

  *Function returning vector length.*
- def distance (self, other)

  *Function returning the length of the displacement vector (distance between two points).*
- def __truediv__ (self, other)

  *Function returning quotient of two vectors or quotient of a vector and a scalar.*
- def __itruediv__ (self, other)

  *Function returning quotient (In-place Division) of two vectors or quotient of a vector and a scalar.*
- def __mul__ (self, other)

  *Function returning dot product of two vectors or dot product of a vector and a scalar.*
- def __imul__ (self, other)

  *Function returning dot product (In-place multiplication) of two vectors or dot product of a vector and a scalar.*
- def __rmul__ (self, other)

  *Function returning cross product of two vectors.*

### Public Attributes

- x

  *A class variable.*
- y

  *A class variable.*

### 3.19.1 Detailed Description

Documentation for a class Vec2.

### 3.19.2 Member Data Documentation

#### 3.19.2.1 x

`vector.Vec2.x`

A class variable.

Coordinate x of a given vector.

#### 3.19.2.2 y

`vector.Vec2.y`

A class variable.

Coordinate y of a given vector.

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

- vector.py

## 3.20 vector.Vec3 Class Reference

Documentation for a class Vec3.

## Public Member Functions

- def __init__ (self, x=0, y=0, z=0)

  *The constructor.*
- def **x** (self)
- def **x** (self, inp)
- def **x** (self)
- def **y** (self)
- def **y** (self, inp)
- def **y** (self)
- def **z** (self)
- def **z** (self, inp)
- def **z** (self)
- def __add__ (self, other)

  *Function returning sum of two vectors or sum of a vector and a scalar.*
- def __iadd__ (self, other)

  *Function returning sum (In-place addition) of two vectors or sum of a vector and a scalar.*
- def __sub__ (self, other)

  *Function returning difference of two vectors or difference of a vector and a scalar.*
- def __isub__ (self, other)

  *Function returning difference (In-place Subtraction) of two vectors or difference of a vector and a scalar.*
- def __eq__ (self, other)

  *Function "equal".*
- def __abs__ (self)

  *Function returning absolute value of a given vector.*
- def __ne__ (self, other)

  *Function "not equal".*
- def __neg__ (self)

  *Function negating vector coordinates.*
- def __pos__ (self)

  *Function for positive vector coordinates.*
- def __str__ (self)

  *Function returning object values in string format.*
- def length (self)

  *Function returning vector length.*
- def distance (self, other)

  *Function returning the length of the displacement vector (distance between two points).*
- def is_point_on_ray (self, ray)

  *Is point on ray wrapper.*
- def **normalized** (self)
- def __truediv__ (self, other)

  *Function returning quotient of two vectors or quotient of a vector and a scalar.*
- def __itruediv__ (self, other)

  *Function returning quotient (In-place Division) of two vectors or quotient of a vector and a scalar.*
- def __mul__ (self, other)

  *Function returning dot product of two vectors or dot product of a vector and a scalar.*
- def __imul__ (self, other)

  *Function returning dot product (In-place multiplication) of two vectors or dot product of a vector and a scalar.*
- def __rmul__ (self, other)

  *Function returning dot product (Reverse multiplication).*
- def cross (self, other)

  *Function returning cross product of two vectors.*

## Public Attributes

- x
  *A class variable.*
- y
  *A class variable.*
- z
  *A class variable.*

## Static Public Attributes

- def r = x
  *Alias*
- def g = y
  *Alias.*
- def b = z
  *Alias.*

### 3.20.1 Detailed Description

Documentation for a class Vec3.

### 3.20.2 Member Data Documentation

#### 3.20.2.1 x

`vector.Vec3.x`

A class variable.

Coordinate x of a given vector.

#### 3.20.2.2 y

`vector.Vec3.y`

A class variable.

Coordinate y of a given vector.

#### 3.20.2.3 z

`vector.Vec3.z`

A class variable.

Coordinate z of a given vector.

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

- vector.py

# Index