

Fotorealistyczna Grafika Komputerowa

Kamera, akcja

Adam Błaszczyk 239636
Antonina Matuszek 239687

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 light_intensity.LightIntensity Class Reference	5
3.1.1 Detailed Description	5
3.2 image.MyImage Class Reference	5
3.2.1 Detailed Description	6
3.3 orthogonal_camera.OrthogonalCamera Class Reference	6
3.3.1 Detailed Description	7
3.4 perspective_camera.PerspectiveCamera Class Reference	7
3.4.1 Detailed Description	8
3.5 plane.Plane Class Reference	8
3.5.1 Detailed Description	9
3.6 primitive.Primitive Class Reference	9
3.7 ray.Ray Class Reference	10
3.7.1 Detailed Description	10
3.7.2 Member Data Documentation	10
3.7.2.1 direction	11
3.7.2.2 length	11
3.7.2.3 origin	11
3.7.2.4 target	11
3.8 sphere.Sphere Class Reference	11
3.8.1 Detailed Description	12
3.9 tests.Test Class Reference	12
3.9.1 Detailed Description	14
3.10 vector.Vec2 Class Reference	14
3.10.1 Detailed Description	15
3.10.2 Member Data Documentation	15
3.10.2.1 x	15
3.10.2.2 y	15
3.11 vector.Vec3 Class Reference	15
3.11.1 Detailed Description	17
3.11.2 Member Data Documentation	17
3.11.2.1 x	17
3.11.2.2 y	17
3.11.2.3 z	17
Index	19

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

ABC	
primitive.Primitive	9
image.MyImage	5
orthogonal_camera.OrthogonalCamera	6
perspective_camera.PerspectiveCamera	7
Primitive	
plane.Plane	8
sphere.Sphere	11
ray.Ray	10
TestCase	
tests.Test	12
vector.Vec2	14
vector.Vec3	15
light_intensity.LightIntensity	5

Chapter 2

Class Index

2.1 Class List

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

light_intensity.LightIntensity	
Documentation for a class LightIntensity	5
image.MyImage	
Documentation for a class MyImage	5
orthogonal_camera.OrthogonalCamera	
Class for othogonal camera	6
perspective_camera.PerspectiveCamera	
Class for othogonal camera	7
plane.Plane	
Documentation for a class Plane	8
primitive.Primitive	9
ray.Ray	
Documentation for a class Ray	10
sphere.Sphere	
Documentation for a class Sphere	11
tests.Test	
Documentation for a class Test	12
vector.Vec2	
Documentation for a class Vec2	14
vector.Vec3	
Documentation for a class Vec3	15

Chapter 3

Class Documentation

3.1 light_intensity.LightIntensity Class Reference

Documentation for a class [LightIntensity](#).

Inheritance diagram for light_intensity.LightIntensity:

Collaboration diagram for light_intensity.LightIntensity:

Public Member Functions

- def [clamp_0_1](#) (light)
Function clamping values to 0 - 1 range.
- def [clamp_0_255](#) (light)
Function clamping values to 0 - 255 range.

Additional Inherited Members

3.1.1 Detailed Description

Documentation for a class [LightIntensity](#).

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

- light_intensity.py

3.2 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)`
Function saving image to png format.

Public Attributes

- `width`
- `height`
- `image_matrix`

3.2.1 Detailed Description

Documentation for a class [MyImage](#).

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

- `image.py`

3.3 `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.

3.3.1 Detailed Description

Class for othogonal camera.

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

- `orthogonal_camera.py`

3.4 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_scene(self, primitives)`
Function rendering the scene.

Static Public Member Functions

- `def adaptive_antialiasing(ray, A, B, C, D, E, depth, max_depth, horizontal, vertical, background_color, primitives)`
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.

3.4.1 Detailed Description

Class for othogonal camera.

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

- `perspective_camera.py`

3.5 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])`
Constructor.
- `def __str__ (self)`
Function returning object values in string format.
- `def get_detailed_intersection (self, ray)`
Returns tuple with multiple data: point (None if no intersection), distance to point.
- `def get_intersection (self, ray)`
Checks if plane and ray intersect witch each other and returns intersection point if they do, otherwise None.

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.

3.5.1 Detailed Description

Documentation for a class [Plane](#).

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

- plane.py

3.6 primitive.Primitive Class Reference

Inheritance diagram for primitive.Primitive:

Collaboration diagram for primitive.Primitive:

Public Member Functions

- `def __init__(self, color)`
- `def get_intersection(self, ray)`
- `def get_detailed_intersection(self, ray)`

Public Attributes

- **color**

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

- primitive.py

3.7 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)`
- 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_pixel_color (self, primitives)`
- Iterates through list of primitives and returns color of the pixel.*

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.*

3.7.1 Detailed Description

Documentation for a class [Ray](#).

3.7.2 Member Data Documentation

3.7.2.1 direction

`ray.Ray.direction`

Direction vector of a given ray.

Cannot be (0, 0, 0).

3.7.2.2 length

`ray.Ray.length`

Length of a given ray.

Default = Infinity

3.7.2.3 origin

`ray.Ray.origin`

Origin vector of a given ray.

Default = (0, 0, 0)

3.7.2.4 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`

3.8 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, centre=Vec3(0, 0, 0), radius=1, color=[1, 0, 1])`
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_ray_intersections (self, ray)`
Checks if ray intersects with sphere and returns intersection points in form one- or two-element array.
- `def get_intersection (self, ray)`
Checks if ray intersects with sphere and returns point closest to ray origin.
- `def get_detailed_intersection (self, ray)`
Function returning intersection point and distance.

Public Attributes

- [centre](#)
Centre of the sphere.
- **color**
- [radius](#)
Radius of the sphere.
- [area](#)
Area of the sphere.
- [volume](#)
Volume of the sphere.

3.8.1 Detailed Description

Documentation for a class [Sphere](#).

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

- `sphere.py`

3.9 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**

3.9.1 Detailed Description

Documentation for a class [Test](#).

Unit tests.

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

- tests.py

3.10 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.

3.10.1 Detailed Description

Documentation for a class [Vec2](#).

3.10.2 Member Data Documentation

3.10.2.1 x

```
vector.Vec2.x
```

A class variable.

Coordinate x of a given vector.

3.10.2.2 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`

3.11 vector.Vec3 Class Reference

Documentation for a class [Vec3](#).

Inheritance diagram for vector.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 normalize (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.

3.11.1 Detailed Description

Documentation for a class `Vec3`.

3.11.2 Member Data Documentation

3.11.2.1 `x`

`vector.Vec3.x`

A class variable.

Coordinate x of a given vector.

3.11.2.2 `y`

`vector.Vec3.y`

A class variable.

Coordinate y of a given vector.

3.11.2.3 `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

- direction
 - ray.Ray, [10](#)
- image.MyImage, [5](#)
- length
 - ray.Ray, [11](#)
- light_intensity.LightIntensity, [5](#)
- origin
 - ray.Ray, [11](#)
- orthogonal_camera.OrthogonalCamera, [6](#)
- perspective_camera.PerspectiveCamera, [7](#)
- plane.Plane, [8](#)
- primitive.Primitive, [9](#)
- ray.Ray, [10](#)
 - direction, [10](#)
 - length, [11](#)
 - origin, [11](#)
 - target, [11](#)
- sphere.Sphere, [11](#)
- target
 - ray.Ray, [11](#)
- tests.Test, [12](#)
- vector.Vec2, [14](#)
 - x, [15](#)
 - y, [15](#)
- vector.Vec3, [15](#)
 - x, [17](#)
 - y, [17](#)
 - z, [17](#)
- x
 - vector.Vec2, [15](#)
 - vector.Vec3, [17](#)
- y
 - vector.Vec2, [15](#)
 - vector.Vec3, [17](#)
- z
 - vector.Vec3, [17](#)