

Geometry

0.1

Создано системой Doxygen 1.9.1

1	Алфавитный указатель классов	1
1.1	Классы	1
2	Список файлов	3
2.1	Файлы	3
3	Классы	5
3.1	Класс <code>mesh_object</code>	5
3.1.1	Подробное описание	5
3.1.2	Конструктор(ы)	6
3.1.2.1	<code>mesh_object()</code>	6
3.1.3	Методы	6
3.1.3.1	<code>get_mesh()</code>	6
3.1.3.2	<code>get_pos()</code>	6
3.1.3.3	<code>get_visible_mesh()</code>	6
3.1.3.4	<code>has_incedent_faces()</code>	7
3.1.3.5	<code>set_mesh()</code>	7
3.1.3.6	<code>set_mesh_part()</code>	7
3.1.3.7	<code>set_pos()</code>	8
3.2	Класс <code>point3f</code>	8
3.2.1	Подробное описание	9
3.2.2	Конструктор(ы)	9
3.2.2.1	<code>point3f()</code>	9
3.2.3	Методы	9
3.2.3.1	<code>cos()</code>	10
3.3	Класс <code>trface</code>	10
3.3.1	Подробное описание	11
3.3.2	Конструктор(ы)	11
3.3.2.1	<code>trface()</code>	11
3.4	Структура <code>trmesh</code>	11
3.4.1	Подробное описание	12
4	Файлы	13
4.1	Файл <code>geometry.hpp</code>	13
4.1.1	Подробное описание	13
4.2	Файл <code>geometry_util.hpp</code>	13
4.2.1	Подробное описание	14
4.2.2	Функции	14
4.2.2.1	<code>calculate_normal()</code>	14
4.2.2.2	<code>calculate_vertex_normal()</code>	14
4.2.2.3	<code>get_vector3f_length()</code>	15
4.2.2.4	<code>normalize_vector()</code>	15
4.3	Файл <code>mesh_object.cpp</code>	15
4.3.1	Подробное описание	16

4.4 Файл <code>mesh_object.hpp</code>	16
4.4.1 Подробное описание	16
4.5 Файл <code>point3f.hpp</code>	17
4.5.1 Подробное описание	17
4.6 Файл <code>trface.hpp</code>	17
4.6.1 Подробное описание	17
Предметный указатель	19

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

mesh_object	Mesh_object - Трехмерный объект	5
point3f	Point3f - класс точки	8
trface	Trface - класс треугольного полигона	10
trmesh	Trmesh - Полигональная сетка	11

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

geometry.hpp	13
geometry_util.hpp	13
mesh_object.cpp	15
mesh_object.hpp	16
point3f.hpp	17
trface.hpp	17

Глава 3

Классы

3.1 Класс mesh_object

`mesh_object` - Трехмерный объект

```
#include <mesh_object.hpp>
```

Открытые члены

- `mesh_object` (`point3f` _pos, int _type)
Конструктор
- void `set_mesh` (`trimesh` _mesh)
 - Функция установки полигональной сетки
- void `set_pos` (`point3f` _pos)
 - Функция установки позиции объекта
- void `set_mesh_part` (`trimesh` _mesh)
 - Функция установки части полигональной сетки
- `trimesh` `get_mesh` ()
 - Функция получения полигональной сетки
- `point3f` `get_pos` ()
 - Функция получения позиции объекта
- bool `has_incedent_faces` (int index)
 - Функция проверки наличия инцидентных граней
- `trimesh` * `get_visible_mesh` (`point3f` camera_pos)
 - Функция получения видимой полигональной сетки

Друзья

- `std::ostream` & `operator<<` (`std::ostream` &out, `mesh_object` mobj)

3.1.1 Подробное описание

`mesh_object` - Трехмерный объект

Состоит из треугольных полигонов. Каждый полигон и вершина сетки имеют свой индекс. Так-же у объект имеет позицию и тип(не используется в текущей версии)

3.1.2 Конструктор(ы)

3.1.2.1 mesh_object()

```
mesh_object::mesh_object (
    point3f _pos,
    int _type )
```

Конструктор

Аргументы

_pos	- Позиция объекта
_type	- Тип объекта(Не используется в текущей версии)

3.1.3 Методы

3.1.3.1 get_mesh()

```
trimesh mesh_object::get_mesh ( )
```

- Функция получения полигональной сетки

Возвращает

trimesh - Полигональная сетка

3.1.3.2 get_pos()

```
point3f mesh_object::get_pos ( )
```

- Функция получения позиции объекта

Возвращает

point3f - Позиция объекта

3.1.3.3 get_visible_mesh()

```
trimesh * mesh_object::get_visible_mesh (
    point3f camera_pos )
```

- Функция получения видимой полигональной сетки

Предполагается что камера направлена в центр объекта

Аргументы

camera_pos	- Позиция камеры
------------	------------------

Возвращает

*trmesh - Указатель на полигональную сетку

3.1.3.4 has_incident_faces()

```
bool mesh_object::has_incident_faces (
    int index )
```

- Функция проверки наличия инцидентных граней

Функция проверяет наличие инцидентных вершине, с указанным индексом, граней

Аргументы

index	- Индекс вершины
-------	------------------

Возвращает

bool

3.1.3.5 set_mesh()

```
void mesh_object::set_mesh (
    trmesh _mesh )
```

- Функция установки полигональной сетки

Аргументы

_mesh	- Полигональная сетка
-------	-----------------------

3.1.3.6 set_mesh_part()

```
void mesh_object::set_mesh_part (
```

```
trimesh _mesh )
```

- Функция установки части полигональной сетки

Функция изменяет полигональную сетку в соответствии индексами полигонов, полигоны имеющие хотя бы 2 вершины с одинаковыми индексами считаются удаленными

Аргументы

<code>_mesh</code>	- Полигональная сетка
--------------------	-----------------------

3.1.3.7 set_pos()

```
void mesh_object::set_pos (
    point3f _pos )
```

- Функция установки позиции объекта

Аргументы

<code>_pos</code>	- Позиция объекта
-------------------	-------------------

Объявления и описания членов классов находятся в файлах:

- [mesh_object.hpp](#)
- [mesh_object.cpp](#)

3.2 Класс point3f

[point3f](#) - класс точки

```
#include <point3f.hpp>
```

Открытые члены

- [point3f](#) (float _x=0, float _y=0, float _z=0)
Конструктор
- float [cos](#) ([point3f](#) vec)
cos - Функция косинуса
- [point3f](#) operator+ ([point3f](#) &point)
- [point3f](#) operator- ([point3f](#) &point)
- [point3f](#) operator/ (float number)
- [point3f](#) operator+= ([point3f](#) point)
- [point3f](#) operator-= ([point3f](#) point)
- [point3f](#) operator/= (float number)

Открытые атрибуты

- float `x`
x координата точки
- float `y`
y координата точки
- float `z`
z координата точки

Друзья

- `std::ostream & operator<< (std::ostream &out, const point3f point)`

3.2.1 Подробное описание

`point3f` - класс точки

координаты точки задаются вещественными(дробными) числами

3.2.2 Конструктор(ы)

3.2.2.1 point3f()

```
point3f::point3f (
    float _x = 0,
    float _y = 0,
    float _z = 0 )
```

Конструктор

Аргументы

↵ _↵ x	- x координата точки
↵ _↵ y	- y координата точки
↵ _↵ z	- z координата точки

3.2.3 Методы

3.2.3.1 `cos()`

```
float point3f::cos (
    point3f vec )
```

`cos` - Функция косинуса

Функция находит угол между переданным и текущим вектором

Аргументы

<code>vec</code>	- координаты вектора
------------------	----------------------

Возвращает

`float`

Объявления и описания членов классов находятся в файлах:

- [point3f.hpp](#)
- [point3f.cpp](#)

3.3 Класс `trface`

`trface` - класс треугольного полигона

```
#include <trface.hpp>
```

Открытые члены

- [trface](#) (int p1, int p2, int p3)
Конструктор

Открытые атрибуты

- int [point1](#)
Первая Точка
- int [point2](#)
Вторая Точка
- int [point3](#)
Третья Точка
- [point3f normal](#)
Координаты вектора нормали

Друзья

- `std::ostream & operator<< (std::ostream &out, const trface face)`

3.3.1 Подробное описание

trface - класс треугольного полигона

точки полигона задаются их индексами так же trface хранит координаты вектора нормали полигона

3.3.2 Конструктор(ы)

3.3.2.1 trface()

```
trface::trface (
    int p1,
    int p2,
    int p3 )
```

Конструктор

координаты вектора нормали рассчитываются специальной функцией позже при создании полигона устанавливаются как (0,0,0)

Аргументы

p1	- индекс первой точки
p2	- индекс второй точки
p3	- индекс третьей точки

Объявления и описания членов классов находятся в файлах:

- [trface.hpp](#)
- [trface.cpp](#)

3.4 Структура trmesh

trmesh - Полигональная сетка

```
#include <mesh_object.hpp>
```

Открытые атрибуты

- `map< int, point3f * > vertexes`
Набор вершин
- `map< int, trface * > faces`
Набор граней

3.4.1 Подробное описание

trmesh - Полигональная сетка

Состоит из треугольных полигонов. Каждый полигон и вершина сетки имеют свой индекс.

Объявления и описания членов структуры находятся в файле:

- [mesh_object.hpp](#)

Глава 4

Файлы

4.1 Файл geometry.hpp

```
#include "point3f.hpp"  
#include "trface.hpp"  
#include "mesh_object.hpp"  
#include "geometry_util.hpp"
```

4.1.1 Подробное описание

- это заголовочный файл библиотеки geometry

Эта библиотека предназначена для работы с трехмерной геометрией

Автор

Kostyunin Egor, kostyunin.eg@yandex.ru

Версия

0.1

Дата

2022

Авторство

MIT-License

4.2 Файл geometry_util.hpp

```
#include "mesh_object.hpp"
```

Функции

- void `calculate_normal` (`trmesh *mesh`)
- `point3f` `calculate_vertex_normal` (`trface *faces`, `int count`)
- float `get_vector3f_length` (`point3f vec`)
- void `normalize_vector` (`point3f *vec`)

4.2.1 Подробное описание

- этот файл содержит вспомогательные функции для работы с геометрией

Автор

Kostyunin Egor, kostyunin.eg@yandex.ru

Версия

0.1

Дата

2022

Авторство

MIT-License

4.2.2 Функции

4.2.2.1 `calculate_normal()`

```
void calculate_normal (
    trmesh * mesh )
```

@brief `calculate_normal` - функция расчета нормали

расчитывает нормали для полигональной сетки

Аргументы

mesh	- указатель на полигональную сетку
------	------------------------------------

4.2.2.2 `calculate_vertex_normal()`

```
point3f calculate_vertex_normal (
    trface * faces,
    int count )
```

@brief calculate_vertex_normal - функция расчета нормали

расчитывает нормали для вершины как среднее значение нормалей смежных полигонов

Аргументы

faces	- указатель на массив смежных полигонов
count	- кол-во полигонов

4.2.2.3 get_vector3f_length()

```
float get_vector3f_length (  
    point3f vec )
```

@brief get_vector3f_length - функция расчета длины вектора

Аргументы

vec	- координаты вектора
-----	----------------------

4.2.2.4 normalize_vector()

```
void normalize_vector (  
    point3f * vec )
```

@brief normalize_vector - функция нормализации вектора

Аргументы

vec	- указатель на координаты вектора
-----	-----------------------------------

4.3 Файл mesh_object.cpp

```
#include "geometry_util.hpp"
```

Функции

- std::ostream & operator<< (std::ostream &out, mesh_object mobj)

4.3.1 Подробное описание

- этот файл содержит реализацию класса трехмерного объекта [mesh_object](#)

Автор

Kostyunin Egor, kostyunin.eg@yandex.ru

Версия

0.1

Дата

2022

Авторство

MIT-License

4.4 Файл mesh_object.hpp

```
#include <iostream>
#include <map>
#include "point3f.hpp"
#include "trface.hpp"
```

Классы

- struct [trmesh](#)
trmesh - Полигональная сетка
- class [mesh_object](#)
[mesh_object](#) - Трехмерный объект

4.4.1 Подробное описание

- этот файл содержит класс трехмерного объекта [mesh_object](#) и определение структуры trmesh

Автор

Kostyunin Egor, kostyunin.eg@yandex.ru

Версия

0.1

Дата

2022

Авторство

MIT-License

4.5 Файл point3f.hpp

```
#include <iostream>
```

Классы

- class [point3f](#)
[point3f](#) - класс точки

4.5.1 Подробное описание

- этот файл содержит класс точки [point3f](#)

Автор

Kostyunin Egor, kostyunin.eg@yandex.ru

Версия

0.1

Дата

2022

Авторство

MIT-License

4.6 Файл trface.hpp

```
#include <iostream>  
#include "point3f.hpp"
```

Классы

- class [trface](#)
[trface](#) - класс треугольного полигона

4.6.1 Подробное описание

- этот файл содержит класс треугольного полигона [trface](#)

Автор

Kostyunin Egor, kostyunin.eg@yandex.ru

Версия

0.1

Дата

2022

Авторство

MIT-License

Предметный указатель

- calculate_normal
 - geometry_util.hpp, [14](#)
- calculate_vertex_normal
 - geometry_util.hpp, [14](#)
- cos
 - point3f, [9](#)
- geometry.hpp, [13](#)
- geometry_util.hpp, [13](#)
 - calculate_normal, [14](#)
 - calculate_vertex_normal, [14](#)
 - get_vector3f_length, [15](#)
 - normalize_vector, [15](#)
- get_mesh
 - mesh_object, [6](#)
- get_pos
 - mesh_object, [6](#)
- get_vector3f_length
 - geometry_util.hpp, [15](#)
- get_visible_mesh
 - mesh_object, [6](#)
- has_incident_faces
 - mesh_object, [7](#)
- mesh_object, [5](#)
 - get_mesh, [6](#)
 - get_pos, [6](#)
 - get_visible_mesh, [6](#)
 - has_incident_faces, [7](#)
 - mesh_object, [6](#)
 - set_mesh, [7](#)
 - set_mesh_part, [7](#)
 - set_pos, [8](#)
- mesh_object.cpp, [15](#)
- mesh_object.hpp, [16](#)
- normalize_vector
 - geometry_util.hpp, [15](#)
- point3f, [8](#)
 - cos, [9](#)
 - point3f, [9](#)
- point3f.hpp, [17](#)
- set_mesh
 - mesh_object, [7](#)
- set_mesh_part
 - mesh_object, [7](#)
- set_pos
 - mesh_object, [8](#)
- mesh_object, [8](#)
- trface, [10](#)
 - trface, [11](#)
- trface.hpp, [17](#)
- trmesh, [11](#)