

AppleCarving

Создано системой Doxygen 1.9.8



|  |    |
|--|----|
| 1 Иерархический список классов   | 1  |
| 1.1 Иерархия классов   | 1  |
| 2 Алфавитный указатель классов   | 3  |
| 2.1 Классы   | 3  |
| 3 Список файлов  | 5  |
| 3.1 Файлы  | 5  |
| 4 Классы   | 7  |
| 4.1 Класс Carving  | 7  |
| 4.1.1 Подробное описание   | 7  |
| 4.2 Класс Cities   | 8  |
| 4.2.1 Подробное описание   | 8  |
| 4.3 Структура Coordinate   | 9  |
| 4.3.1 Подробное описание   | 9  |
| 4.4 Шаблон класса My_vertex_base< Gt, Vb >   | 9  |
| 4.4.1 Подробное описание   | 10 |
| 4.5 Шаблон структуры My_vertex_base< Gt, Vb >::Rebind_TDS< TDS2 >  | 10 |
| 4.6 Структура Traping  | 10 |
| 4.6.1 Подробное описание   | 10 |
| 5 Файлы  | 11 |
| 5.1 Файл C:/Users/pozoy/Desktop/lib_apple/AppleCarving/project/carving/include/AppleCarving/AppleCarving.hpp | 11 |
| 5.1.1 Подробное описание   | 12 |
| 5.2 AppleCarving.hpp   | 12 |
| Предметный указатель   | 15 |



# Глава 1

## Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

|  |    |
|--|----|
| Carving . . . . .                                      | 7  |
| Cities . . . . .                                       | 8  |
| Coordinate . . . . .                                   | 9  |
| My_vertex_base< Gt, Vb >::Rebind_TDS< TDS2 > . . . . . | 10 |
| Traping . . . . .                                      | 10 |
| CGAL::Triangulation_vertex_base_2                      |    |
| My_vertex_base< Gt, Vb > . . . . .                     | 9  |



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

|  |  |    |
|--|--|----|
| <a href="#">Carving</a>  | Класс, реализующий алгоритм AppleCarving . . . . .   | 7  |
| <a href="#">Cities</a>   | Класс, содержащий информацию о городах . . . . .     | 8  |
| <a href="#">Coordinate</a>   | Структура координат точек триангуляции . . . . .     | 9  |
| <a href="#">My_vertex_base&lt; Gt, Vb &gt;</a>                           | Класс, реализующий построение триангуляции . . . . . | 9  |
| <a href="#">My_vertex_base&lt; Gt, Vb &gt;::Rebind_TDS&lt; TDS2 &gt;</a> |  | 10 |
| <a href="#">Traping</a>  | Структура содержащая оптимальный путь . . . . .      | 10 |





## Глава 3

# Список файлов

### 3.1 Файлы

Полный список документированных файлов.

C:/Users/pozoy/Desktop/lib\_apple/AppleCarving/project/carving/include/AppleCarving/[AppleCarving.hpp](#)  
Предоставляет классы и методы реализации алгоритма Apple [Carving](#) . . . . . 11



## Глава 4

# Классы

### 4.1 Класс Carving

Класс, реализующий алгоритм AppleCarving.

```
#include <AppleCarving.hpp>
```

Открытые члены

- Carving ()  
Конструктор класса
- ~Carving ()  
Деструктор класса
- void MergeFaces ([Cities](#) &cities)  
Вырезает полигоны из триангуляции
- void Generate\_path ([Cities](#) &cities, int begin\_trap, int end\_trap)  
Поиск оптимального пути

Открытые атрибуты

- std::vector< [Traping](#) > traping  
Вектор содержащий все варианты путей

#### 4.1.1 Подробное описание

Класс, реализующий алгоритм AppleCarving.

Объявления и описания членов класса находятся в файле:

- C:/Users/pozoy/Desktop/lib\_apple/AppleCarving/project/carving/include/AppleCarving/[AppleCarving.hpp](#)

## 4.2 Класс Cities

Класс, содержащий информацию о городах.

```
#include <AppleCarving.hpp>
```

Открытые члены

- Cities ()  
Конструктор класса
- ~Cities ()  
Деструктор класса
- void getCount (std::string [path](#))  
Установка количества городов
- void getCoordinate (std::string [path](#))  
Получение координат.
- void getWeight ()  
Установка расстояния между городами.
- void InitializationPoints (std::string [path](#), int argc, char \*argv[])  
Инициализация расстояний между городами
- void DrawBeforeCarving ()  
Отрисовывает триангуляцию перед применением алгоритма Apple [Carving](#).
- void DrawAfterCarving ()  
Отрисовывает триангуляцию после применения алгоритма Apple [Carving](#).

Открытые атрибуты

- int count  
Количество городов
- std::string path  
Путь к файлу
- std::vector< [Coordinate](#) > coordinate  
Координаты точек триангуляции
- std::vector< std::vector< int > > weight  
Расстояния между городами
- [Triangulation](#) TBeforeCarving  
Триангуляция до применения алгоритма
- [Triangulation](#) TAfterCarving  
Триангуляция после применения алгоритма
- std::vector< [Vertex\\_handle](#) > Vertex  
Вершины триангуляции

### 4.2.1 Подробное описание

Класс, содержащий информацию о городах.

Объявления и описания членов класса находятся в файле:

- C:/Users/pozoy/Desktop/lib\_apple/AppleCarving/project/carving/include/AppleCarving/[AppleCarving.hpp](#)

## 4.3 Структура Coordinate

Структура координат точек триангуляции

```
#include <AppleCarving.hpp>
```

Открытые атрибуты

- `int x`  
Координата x.
- `int y`  
Координата y.

### 4.3.1 Подробное описание

Структура координат точек триангуляции

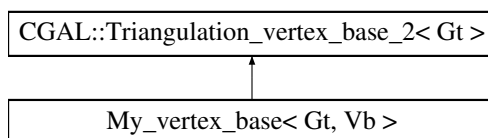
Объявления и описания членов структуры находятся в файле:

- `C:/Users/pozoy/Desktop/lib_apple/AppleCarving/project/carving/include/AppleCarving/AppleCarving.hpp`

## 4.4 Шаблон класса `My_vertex_base< Gt, Vb >`

Класс, реализующий построение триангуляции

Граф наследования: `My_vertex_base< Gt, Vb >`:



Классы

- `struct Rebind_TDS`

Открытые типы

- `typedef Vb::Vertex_handle Vertex_handle`  
Переопределенный тип `Vertex_handle`.
- `typedef Vb::Face_handle Face_handle`  
Переопределенный тип `Face_handle`.
- `typedef Vb::Point Point`  
Переопределенный тип `Point`.

## Открытые члены

- `My_vertex_base ()`  
Конструктор класса 1.
- `My_vertex_base (const Point &p)`  
Конструктор класса 2.
- `My_vertex_base (const Point &p, Face\_handle f)`  
Конструктор класса 3.
- `My_vertex_base (Face\_handle f)`  
Конструктор класса 4.
- `void set_associated_vertex (Vertex\_handle va)`  
Устанавливает связи между вершинами
- `Vertex\_handle get_associated_vertex ()`

## 4.4.1 Подробное описание

```
template<class Gt, class Vb = CGAL::Triangulation_vertex_base_2<Gt>>
class My_vertex_base< Gt, Vb >
```

Класс, реализующий построение триангуляции

Объявления и описания членов класса находятся в файле:

- `C:/Users/pozoy/Desktop/lib_apple/AppleCarving/project/carving/include/AppleCarving/AppleCarving.hpp`

4.5 Шаблон структуры `My_vertex_base< Gt, Vb >::Rebind_TDS< TDS2 >`

## Открытые типы

- `typedef Vb::template Rebind\_TDS< TDS2 >::Other Vb2`
- `typedef My\_vertex\_base< Gt, Vb2 > Other`

Объявления и описания членов структуры находятся в файле:

- `C:/Users/pozoy/Desktop/lib_apple/AppleCarving/project/carving/include/AppleCarving/AppleCarving.hpp`

4.6 Структура `Traping`

Структура содержащая оптимальный путь

```
#include <AppleCarving.hpp>
```

## Открытые атрибуты

- `int groups`  
Точки оптимального пути
- `int distance`  
Расстояние оптимального пути

## 4.6.1 Подробное описание

Структура содержащая оптимальный путь

Объявления и описания членов структуры находятся в файле:

- `C:/Users/pozoy/Desktop/lib_apple/AppleCarving/project/carving/include/AppleCarving/AppleCarving.hpp`

## Глава 5

### Файлы

#### 5.1 Файл C:/Users/pozoy/Desktop/lib\_apple/AppleCarving/project/carving/include/AppleCarving/AppleCarving.hpp

Предоставляет классы и методы реализации алгоритма Apple [Carving](#).

```
#include <CGAL/Exact_predicates_inexact_constructions_kernel.h>
#include <CGAL/Triangulation_2.h>
#include <CGAL/draw_triangulation_2.h>
#include <CGAL/boost/graph/Euler_operations.h>
#include <CGAL/Vector_2.h>
#include <cassert>
#include <string>
#include <vector>
#include <fstream>
#include <cmath>
```

#### Классы

- class [My\\_vertex\\_base< Gt, Vb >](#)  
Класс, реализующий построение триангуляции
- struct [My\\_vertex\\_base< Gt, Vb >::Rebind\\_TDS< TDS2 >](#)
- struct [Coordinate](#)  
Структура координат точек триангуляции
- struct [Traping](#)  
Структура содержащая оптимальный путь
- class [Cities](#)  
Класс, содержащий информацию о городах.
- class [Carving](#)  
Класс, реализующий алгоритм AppleCarving.

## Определения типов

- `typedef CGAL::Exact_predicates_inexact_constructions_kernel K`  
Переопределенный тип K.
- `typedef My_vertex_base< K > Vb`  
Переопределенный тип Vb.
- `typedef CGAL::Triangulation_data_structure_2< Vb > Tds`  
Переопределенный тип Tds.
- `typedef CGAL::Triangulation_2< K, Tds > Triangulation`  
Переопределенный тип Triangulation.
- `typedef Triangulation::Vertex_handle Vertex_handle`  
Переопределенный тип Vertex\_handle.
- `typedef Triangulation::Finite_faces_iterator Finite_faces_iterator`  
Переопределенный тип Finite\_faces\_iterator.
- `typedef Triangulation::Point Point`  
Переопределенный тип Point.

### 5.1.1 Подробное описание

Предоставляет классы и методы реализации алгоритма Apple [Carving](#).

•

## 5.2 AppleCarving.hpp

[См. документацию.](#)

```
00001 #include <CGAL/Exact_predicates_inexact_constructions_kernel.h>
00002 #include <CGAL/Triangulation_2.h>
00003 #include <CGAL/draw_triangulation_2.h>
00004 #include <CGAL/boost/graph/Euler_operations.h>
00005 #include <CGAL/Vector_2.h>
00006 #include <cassert>
00007 #include <string>
00008 #include <vector>
00009 #include <fstream>
00010 #include <cmath>
00011
00012 #ifndef APPLE_CARVING_HPP
00013 #define APPLE_CARVING_HPP
00014
00015 template <class Gt, class Vb = CGAL::Triangulation_vertex_base_2<Gt>
00016 class My_vertex_base
00017 : public Vb
00018 {
00019     typedef Vb Base;
00020 public:
00021     typedef typename Vb::Vertex_handle Vertex_handle;
00022
00023     typedef typename Vb::Face_handle Face_handle;
00024
00025     typedef typename Vb::Point Point;
00026
00027     template < typename TDS2 >
00028     struct Rebind_TDS {
00029         typedef typename Vb::template Rebind_TDS<TDS2>::Other Vb2;
00030         typedef My_vertex_base<Gt, Vb2> Other;
00031     };
00032
00033 private:
00034     Vertex_handle va_;
00035
00036 public:
00037     My_vertex_base() : Base() {}
00038     My_vertex_base(const Point& p) : Base(p) {}
```



```

00050     My_vertex_base(const Point& p, Face_handle f) : Base(f, p) {}
00052     My_vertex_base(Face_handle f) : Base(f) {}
00053
00055     void set_associated_vertex(Vertex_handle va) { va_ = va; }
00056     Vertex_handle get_associated_vertex() { return va_; }
00057 };
00058
00060 typedef CGAL::Exact_predicates_inexact_constructions_kernel K;
00061
00063 typedef My_vertex_base<K> Vb;
00064
00066 typedef CGAL::Triangulation_data_structure_2<Vb> Tds;
00067
00069 typedef CGAL::Triangulation_2<K, Tds> Triangulation;
00070
00072 typedef Triangulation::Vertex_handle Vertex_handle;
00073
00075 typedef Triangulation::Finite_faces_iterator Finite_faces_iterator;
00076
00078 typedef Triangulation::Point Point;
00079
00081 struct Coordinate
00082 {
00084     int x;
00085
00087     int y;
00088 };
00090 struct Traping
00091 {
00093     int groups;
00094
00096     int distance;
00097 };
00099
00102 class Cities
00103 {
00104 public:
00106     int count;
00107
00109     std::string path;
00110
00112     std::vector<Coordinate> coordinate;
00113
00115     std::vector<std::vector<int>> weight;
00116
00118     Triangulation TBeforeCarving;
00119
00121     Triangulation TAfterCarving;
00122
00124     std::vector<Vertex_handle> Vertex;
00125
00127     Cities();
00128
00130     ~Cities();
00131
00133     void getCount(std::string path);
00134
00136     void getCoordinate(std::string path);
00137
00139     void getWeight();
00140
00142     void InitializationPoints(std::string path, int argc, char* argv[]);
00143
00145     void DrawBeforeCarving();
00146
00148     void DrawAfterCarving();
00149
00150 };
00151
00154 class Carving {
00155
00156 public:
00158     std::vector<Traping> trapping;
00159
00161     Carving();
00162
00164     ~Carving();
00165
00167     void MergeFaces(Cities& cities);
00168
00170     void Generate_path(Cities& cities, int begin_trap, int end_trap);
00171
00172 };
00173 #endif

```



# Предметный указатель

C:/Users/pozoy/Desktop/lib\_apple/AppleCarving/project/carving/include/AppleCarving/AppleCarving.hpp,  
[11](#), [12](#)

Carving, [7](#)

Cities, [8](#)

Coordinate, [9](#)

My\_vertex\_base< Gt, Vb >, [9](#)

My\_vertex\_base< Gt, Vb >::Rebind\_TDS<  
TDS2 >, [10](#)

Traping, [10](#)