# PoissonsoftTestTask

Generated by Doxygen 1.11.0

# Chapter 1

# Class Index

## 1.1 Class List

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

# Chapter 2

# File Index

## 2.1 File List

Here is a list of all documented files with brief descriptions:

# Chapter 3

# Class Documentation

## 3.1 IntersectionChecker2D Class Reference

Provides methods to check the intersection of two triangles in 2D space.

`#include <IntersectionChecker2D.h>`

### Static Public Member Functions

- static bool checkIntersection (const Triangle &t1, const Triangle &t2)

  *Checks if two triangles intersect.*

### 3.1.1 Detailed Description

Provides methods to check the intersection of two triangles in 2D space.

### 3.1.2 Member Function Documentation

#### 3.1.2.1 checkIntersection()

```
bool IntersectionChecker2D::checkIntersection (
            const Triangle & t1,
            const Triangle & t2)  [static]
```

Checks if two triangles intersect.

**Parameters**

| | |
|---|---|
| *t1* | The first triangle. |
| *t2* | The second triangle. |

**Returns**

true if the triangles intersect, false otherwise.

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

- include/IntersectionChecker2D.h
- src/IntersectionChecker2D.cpp

## 3.2 Point2D Class Reference

Represents a point in 2D space.

`#include <Point2D.h>`

### Public Member Functions

- Point2D (float x=0, float y=0)
  *Constructs a Point2D object.*

### Public Attributes

- float **x**
- float y

### 3.2.1 Detailed Description

Represents a point in 2D space.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 Point2D()

```
Point2D::Point2D (
            float x = 0,
            float y = 0)  [inline], [explicit]
```

Constructs a Point2D object.

**Parameters**

| | |
|---|---|
| *x* | The x-coordinate of the point. |
| *y* | The y-coordinate of the point. |

### 3.2.3 Member Data Documentation

#### 3.2.3.1 y

```
float Point2D::y
```

The x and y coordinates of the point

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

- include/Point2D.h

## 3.3 TestRunner Class Reference

Runs tests to check the intersection of triangles using an IntersectionChecker2D.

`#include <TestRunner.h>`

### Public Member Functions

- TestRunner (std::unique_ptr< IntersectionChecker2D > checker)

  *Constructs a TestRunner object.*
- void runTests (const std::string &filename)

  *Runs tests from the specified file.*

### 3.3.1 Detailed Description

Runs tests to check the intersection of triangles using an IntersectionChecker2D.

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 TestRunner()

```
TestRunner::TestRunner (
            std::unique_ptr< IntersectionChecker2D > checker)  [inline], [explicit]
```

Constructs a TestRunner object.

**Parameters**

| | |
|---|---|
| *checker* | A unique pointer to an IntersectionChecker2D object. |

### 3.3.3 Member Function Documentation

#### 3.3.3.1 runTests()

```
void TestRunner::runTests (
            const std::string & filename)
```

Runs tests from the specified file.

**Parameters**

| | |
|---|---|
| *filename* | The name of the file containing test cases. |

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

- include/TestRunner.h
- src/TestRunner.cpp

## 3.4 Triangle Class Reference

Represents a triangle in 2D space.

`#include <Triangle.h>`

### Public Member Functions

- Triangle (const std::vector< Point2D > &vertices)
  *Constructs a Triangle object with given vertices.*

### Public Attributes

- std::vector< Point2D > vertices

### 3.4.1 Detailed Description

Represents a triangle in 2D space.

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 Triangle()

```
Triangle::Triangle (
            const std::vector< Point2D > & vertices)  [explicit]
```

Constructs a Triangle object with given vertices.

**Parameters**

| | |
|---|---|
| *vertices* | A vector of 3 points representing the vertices of the triangle. |

**Exceptions**

| | |
|---|---|
| *std::invalid_argument* | if the number of vertices is not equal to 3. |

### 3.4.3 Member Data Documentation

#### 3.4.3.1 vertices

```
std::vector<Point2D> Triangle::vertices
```

The vertices of the triangle

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

- include/Triangle.h
- src/Triangle.cpp

# Chapter 4

# File Documentation

## 4.1 IntersectionChecker2D.h

```
00001 #ifndef INTERSECTION_CHECKER_2D_H
00002 #define INTERSECTION_CHECKER_2D_H
00003
00004 #include "Triangle.h"
00005 #include "Orientation.h"
00006
00011 class IntersectionChecker2D {
00012 public:
00019     static bool checkIntersection(const Triangle& t1, const Triangle& t2) ;
00020
00021 private:
00022     static bool isLinesIntersect(const Point2D& p1, const Point2D& q1, const Point2D& p2, const
      Point2D& q2) ;
00023     static Orientation determineOrientation(const Point2D& p, const Point2D& q, const Point2D& r) ;
00024     static bool isOnSegment(const Point2D& p, const Point2D& q, const Point2D& r) ;
00025     static bool isPointInsideTriangle(const Point2D& p, const Triangle& t) ;
00026 };
00027
00028 #endif // INTERSECTION_CHECKER_2D_H
```

## 4.2 Orientation.h

```
00001 #ifndef ORIENTATION_H
00002 #define ORIENTATION_H
00003
00004 enum class Orientation{
00005     Collinear = 0,
00006     Clockwise = 1,
00007     Counterclockwise = 2
00008 };
00009
00010 #endif //ORIENTATION_H
```

## 4.3 Point2D.h

```
00001 #ifndef POINT2D_H
00002 #define POINT2D_H
00003
00008 class Point2D {
00009 public:
00015     explicit Point2D(float x = 0, float y = 0) : x(x), y(y) {}
00016     float x, y;
00017 };
00018
00019 #endif // POINT2D_H
```

## 4.4 TestRunner.h

```
00001 #ifndef TEST_RUNNER_H
00002 #define TEST_RUNNER_H
00003
00004 #include "IntersectionChecker2D.h"
00005 #include <vector>
00006 #include <string>
00007 #include <memory>
00008 #include <sstream>
00009
00014 class TestRunner {
00015 public:
00020     explicit TestRunner(std::unique_ptr<IntersectionChecker2D> checker) : checker(std::move(checker))
      {}
00021
00026     void runTests(const std::string& filename);
00027 private:
00028     std::unique_ptr<IntersectionChecker2D> checker;
00035     static std::vector<Point2D> parseVertices(std::stringstream& ss);
00036 };
00037
00038 #endif // TEST_RUNNER_H
```

## 4.5 Triangle.h

```
00001 #ifndef TRIANGLE_H
00002 #define TRIANGLE_H
00003
00004 #include <vector>
00005 #include "Point2D.h"
00006
00012 class Triangle {
00013 public:
00019     explicit Triangle(const std::vector<Point2D>& vertices);
00020
00021     std::vector<Point2D> vertices;
00022 };
00023
00024 #endif // TRIANGLE_H
```

## 4.6 Utilities.h

```
00001 #ifndef UTILITIES_H
00002 #define UTILITIES_H
00003
00004 #include <string>
00005 #include <iostream>
00006 #include <algorithm>
00007 #include <cctype>
00008
00009 inline std::string toLower(const std::string& str) {
00010     std::string result = str;
00011     std::transform(result.begin(), result.end(), result.begin(), [](unsigned char c) {
00012         return std::tolower(c);
00013     });
00014     return result;
00015 }
00016
00017 inline void printUsage(const std::string& programName) {
00018     std::cerr « "Usage: " « programName « " <input file>\n";
00019 }
00020
00021 #endif // UTILITIES_H
```