

A_star
1.0

Generated by Doxygen 1.8.6

Fri May 30 2014 03:56:37

Contents

1	A_star	1
1.1	Najważniejsze cechy	1
1.2	Opis algorytmu	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	benchmark Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	benchmark	8
4.1.3	Member Function Documentation	8
4.1.3.1	algorytm	8
4.1.3.2	czas_start	9
4.1.3.3	czas_stop	9
4.1.3.4	ile_czasu	9
4.1.4	Member Data Documentation	9
4.1.4.1	czas	9
4.1.4.2	czas_caly	9
4.1.4.3	elapsedTime	10
4.1.4.4	size	10
4.1.4.5	t1	10
4.1.4.6	t2	10
4.1.4.7	tmp	10
4.2	graf Class Reference	10
4.2.1	Detailed Description	12
4.2.2	Constructor & Destructor Documentation	12
4.2.2.1	graf	12

4.2.3	Member Function Documentation	12
4.2.3.1	A_star	12
4.2.3.2	F	12
4.2.3.3	F	12
4.2.3.4	G	13
4.2.3.5	H	13
4.2.3.6	H	13
4.2.3.7	Rozwiazanie	13
4.2.3.8	Stworz_Sciane	13
4.2.3.9	Ustaw_punkty	13
4.2.4	Friends And Related Function Documentation	14
4.2.4.1	operator<<	14
4.2.5	Member Data Documentation	14
4.2.5.1	koniec	14
4.2.5.2	poczatek	14
4.2.5.3	w	14
4.3	Punkt Struct Reference	14
4.3.1	Detailed Description	15
4.3.2	Member Function Documentation	15
4.3.2.1	operator!=	15
4.3.2.2	operator==	15
4.3.3	Member Data Documentation	15
4.3.3.1	x	15
4.3.3.2	y	15
4.4	Wierzcholek Struct Reference	16
4.4.1	Detailed Description	16
4.4.2	Member Data Documentation	16
4.4.2.1	f	16
4.4.2.2	g	16
4.4.2.3	h	17
4.4.2.4	punkt	17
4.4.2.5	rodzic	17
4.4.2.6	typ	17
5	File Documentation	19
5.1	/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/doc/pages/strona.dox File Reference	19
5.2	/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/benchmark.hh File Reference	19
5.2.1	Detailed Description	20
5.3	/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/graf.hh File Reference	20
5.3.1	Detailed Description	21

5.3.2	Enumeration Type Documentation	21
5.3.2.1	Wierzcholek_Typ	21
5.4	/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/benchmark.cpp File Reference . . .	22
5.4.1	Detailed Description	22
5.5	/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/graf.cpp File Reference	22
5.5.1	Detailed Description	23
5.6	/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/main.cpp File Reference	23
5.6.1	Detailed Description	24
5.6.2	Function Documentation	24
5.6.2.1	main	24
Index		25

Chapter 1

A_star

Author

Paweł Zurek

Date

29.05.2014

Version

1.0

Program umożliwia przeprowadzenie algorytmu A_star.

1.1 Najważniejsze cechy

Algorytm ten wyszukuje najlepszą ścieżkę pomiędzy dwoma punktami w grafie. W tym przypadku graf jest zilustrowany jako układ współrzędnych

1.2 Opis algorytmu

Algorytm wyszukuje miejsce, które odwiedzi jako następne na podstawie wyszukania punktu o najmniejszej wartości funkcji F. Funkcja F to suma arytmetyczna funkcji G i funkcji H. Funkcja H to funkcja Heurystyczna, przewidująca drogę z aktualnego punktu do końcowego. Funkcja G liczy drogę przebytą z punktu początkowego do aktualnego miejsca.

Screen wykonania programu na moim komputerze (1 - przeszkoda , 2 - trasa wyszukiwania) :

Chapter 2

Class Index

2.1 Class List

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

benchmark	Modeluje pojecie Benchmark	7
graf	Modeluje pojecie graf. Klasa sluzzy glownie do wykonania algorytmu a_star, czyli znalezienia najlepszej drogi miedzy dwoma punktami	10
Punkt	Struktura punktu Struktura ta ma zdefiniowane dwie zmienne x oraz y, ktore odpowiadaja za przechowywanie pozycji na siatce. Rowniez zdefiniowane sa operator przypisania oraz operator logiczny relacji	14
Wierzcholek	Struktura Wierzcholek . Opisuje wlasnosci wierzcholka	16

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/ benchmark.hh	
Definicje funkcji dla klasy benchmark	19
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/ graf.hh	
Definicje funkcji dla klasy graf	20
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/ benchmark.cpp	
Plik zawiera funkcje z klasy benchmark	22
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/ graf.cpp	
Plik zawiera funkcje z klasy graf	22
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/ main.cpp	
Plik zawiera funkcje main()	23

Chapter 4

Class Documentation

4.1 benchmark Class Reference

Modeluje pojecie Benchmark.

```
#include <benchmark.hh>
```

Public Member Functions

- `benchmark ()`
Konstruktor klasy Benchmark.
- `void czas_start ()`
Funkcja pomocnicza mierzenia czasu.
- `void czas_stop ()`
Funkcja pomocnicza mierzenia czasu.
- `double ile_czasu ()`
Funkcja obliczania czasu dzialania programu.
- `void algorytm ()`
Funkcja wykonujaca algorytm.

Private Attributes

- `double elapsedTime`
Pole typu double, bedzie uzywane do mierzenia czasu dzialania pojedynczego wypelniania.
- `double czas`
Pole typu double, bedzie uzywane do mierzenia calkowitego czasu dzialania programu.
- `double czas_caly`
Pole typu double, bedzie uzywane do mierzenia calkowitego czasu dzialania programu. !
- `timeval t1`
Pole typu timeval, pomoc do liczenia czasu dzialania operacji krotkich (tzn pojedynczego dzialania)
- `timeval t2`
- `string tmp`
- `int size`

4.1.1 Detailed Description

Modeluje pojecie Benchmark.

Klasa sluzy do przeprowadzenia Benchmarku programu, tzn :

-> przeliczenia czasu wykonywania algorytmu A_star, dla :

* 100 wiercholkow

* 400 wiercholkow

* 900 wiercholkow

* 1600 wiercholkow

* 2500 wiercholkow

Definition at line 35 of file benchmark.hh.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 benchmark::benchmark() [inline]

Konstruktor klasy Benchmark.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie skladowe klasy wartosciami zerowymi.

Definition at line 68 of file benchmark.hh.

4.1.3 Member Function Documentation

4.1.3.1 void benchmark::algorytm()

Funkcja wykonujaca algorytm.

Wykonanie algorytmu ma przebieg :

-> stworzenie grafow o roznym ilosciach wiercholkow:

-> zbadanie czasu wykonywania

-> zapisanie czasow do pliku

Definition at line 26 of file benchmark.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.3.2 void benchmark::czas_start ()

Funkcja pomocnicza mierzenia czasu.

Funkcja zaczyna liczyć czas od momentu wywołania tej metody Służy do liczenia czasu wykonywania pojedynczego wypełniania stosu/kolejki

Definition at line 13 of file benchmark.cpp.

4.1.3.3 void benchmark::czas_stop ()

Funkcja pomocnicza mierzenia czasu.

Funkcja kończy liczyć czas od momentu wywołania tej metody Służy do liczenia czasu wykonywania pojedynczego wypełniania stosu/kolejki

Definition at line 16 of file benchmark.cpp.

4.1.3.4 double benchmark::ile_czasu ()

Funkcja obliczania czasu działania programu.

Funkcja podaje czas wykonywania pojedynczego wypełniania stosu/kolejki

Returns

elapsedTime -> zmienna typu double (wynik obliczeń)

Definition at line 19 of file benchmark.cpp.

4.1.4 Member Data Documentation

4.1.4.1 double benchmark::czas [private]

Pole typu double, będzie używane do mierzenia całkowitego czasu działania programu.

Definition at line 45 of file benchmark.hh.

4.1.4.2 double benchmark::czas_caly [private]

Pole typu double, będzie używane do mierzenia całkowitego czasu działania programu. !

Definition at line 49 of file benchmark.hh.

4.1.4.3 `double benchmark::elapsedTime` [private]

Pole typu `double`, będzie używane do mierzenia czasu działania pojedynczego wypełniania.

Definition at line 41 of file `benchmark.hh`.

4.1.4.4 `int benchmark::size` [private]

Definition at line 57 of file `benchmark.hh`.

4.1.4.5 `timeval benchmark::t1` [private]

Pole typu `timeval`, pomoc do liczenia czasu działania operacji krótkich (tzn pojedynczego działania)

Definition at line 53 of file `benchmark.hh`.

4.1.4.6 `timeval benchmark::t2` [private]

Definition at line 53 of file `benchmark.hh`.

4.1.4.7 `string benchmark::tmp` [private]

Definition at line 55 of file `benchmark.hh`.

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

- /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/benchmark.hh
- /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/benchmark.cpp

4.2 graf Class Reference

Modeluje pojęcie graf. Klasa służy głównie do wykonania algorytmu `a_star`, czyli znalezienia najlepszej drogi między dwoma punktami.

```
#include <graf.hh>
```

Collaboration diagram for `graf`:



Public Member Functions

- [graf](#) (int rozmiar)
Konstruktor klasy graf.
- void [A_star](#) ()
Funkcja A_star. Funkcja implementująca algorytm przeszukiwania A_star. Wywoływana bezparametrycznie, ponieważ punkt początkowy jak i końcowy ustalane są w konstruktorze klasy graf.

Public Attributes

- vector< vector< [Wierzcholek](#) > > w
Pole typu vector<vector<>>, będzie używane do przechowywania informacji o wierzchołku.

Private Member Functions

- void [Stworz_Sciane](#) (int startX, int startY, int stopX, int stopY)
Funkcja tworząca sciane. Funkcja tworzy sciane, przez którą algorytm nie może przejść. Jest zmuszony do szukania drogi obok.
- void [Ustaw_punkty](#) (int startX, int startY, int stopX, int stopY)
Funkcja ustawiająca punkty do znalezienia. Funkcja ustala współrzędne punkty początkowego oraz końcowego. Następnie przydziela je do danego typu wierzchołka.
- void [H](#) ()
Funkcja heurystyczna. Funkcja oblicza przybliżoną drogę jaką algorytm musi przejść, aby odnaleźć cel.
- void [G](#) ()
Funkcja obliczająca poniesiony. Funkcja oblicza koszt poniesiony między wierzchołkiem początkowym a obecnym punktem.
- void [F](#) ()
Funkcja obliczająca współczynnik f. Współczynnik ten to suma arytmetyczna współczynnika g i h. Na podstawie tego algorytm A_star wybiera wierzchołek o najniższym współczynniku przez który przeszukuje graf.
- void [H](#) ([Punkt_w](#))
Funkcja heurystyczna obliczająca drogę o zdanym punkcie.
- void [F](#) ([Punkt_w](#))
Funkcja obliczająca współczynnik f o zdanym punkcie.
- void [Rozwiazanie](#) ()
Funkcja interpretująca rozwiązanie przeszukiwania przez algorytm.

Private Attributes

- [Punkt_poczatek](#)
Pole typu [Punkt](#), będzie używane do przechowywania informacji o punkcie początkowym.
- [Punkt_koniec](#)
Pole typu [Punkt](#), będzie używane do przechowywania informacji o punkcie końcowym.

Friends

- ostream & [operator<<](#) (ostream &wyjscie, [graf](#) &zmienna)
Zdefiniowany operator wyświetlania Operator wypisuje na wyjściu utworzone dane.

4.2.1 Detailed Description

Modeluje pojecie graf. Klasa sluzy glownie do wykonania algorytmu a_star, czyli znalezienia najlepszej drogi miedzy dwoma punktami.

Definition at line 93 of file graf.hh.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 graf::graf (int rozmiar)

Konstruktor klasy graf.

Konstruktor jest parametryczny. Parametrem tym jest rozmiar, który dedynuje rozmiar siatki i tym razem ilość wierzchołków (rozmiar^2). Następnie pozwala ustawić sciane w dowolnym miejscu oraz dowolnie wybrać punkty : początkowy i końcowy.

Parameters

<i>rozmiar</i>	
----------------	--

Definition at line 12 of file graf.cpp.

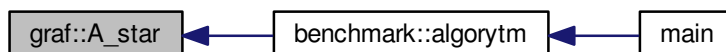
4.2.3 Member Function Documentation

4.2.3.1 void graf::A_star ()

Funkcja A_star. Funkcja implementująca algorytm przeszukiwania A_star. Wywoływana bezparametrycznie, ponieważ punkt początkowy jak i końcowy ustalane są w konstruktorze klasy graf.

Definition at line 125 of file graf.cpp.

Here is the caller graph for this function:



4.2.3.2 void graf::F () [private]

Funkcja obliczająca współczynnik f. Współczynnik ten to suma arytmetyczna współczynnika g i h. Na podstawie tego algorytm A_star wybiera wierzchołek o najniższym współczynniku przez który przeszukuje graf.

Definition at line 98 of file graf.cpp.

4.2.3.3 void graf::F (Punkt_w) [private]

Funkcja obliczająca współczynnik f o zadanym punkcie.

Parameters

<i>_w</i>	
-----------	--

Definition at line 112 of file graf.cpp.

4.2.3.4 void graf::G () [private]

Funkcja obliczająca poniesiony. Funkcja oblicza koszt poniesiony między wierzchołkiem początkowym a obecnym punktem.

Definition at line 88 of file graf.cpp.

4.2.3.5 void graf::H () [private]

Funkcja heurystyczna. Funkcja oblicza przybliżoną drogę jaką algorytm musi przejść, aby odnaleźć cel.

Definition at line 78 of file graf.cpp.

4.2.3.6 void graf::H (Punkt_w) [private]

Funkcja heurystyczna obliczająca drogę o zdanym punkcie.

Parameters

<i>_w</i>	
-----------	--

Definition at line 108 of file graf.cpp.

4.2.3.7 void graf::Rozwiazanie () [private]

Funkcja interpretująca rozwiązanie przeszukiwania przez algorytm.

Definition at line 116 of file graf.cpp.

4.2.3.8 void graf::Stworz_Sciane (int startX, int startY, int stopX, int stopY) [private]

Funkcja tworząca ściany. Funkcja tworzy ściany, przez którą algorytm nie może przejść. Jest zmuszony do szukania drogi obok.

Parameters

<i>startX</i>	
<i>startY</i>	
<i>stopX</i>	
<i>stopY</i>	

Definition at line 60 of file graf.cpp.

4.2.3.9 void graf::Ustaw_punkty (int startX, int startY, int stopX, int stopY) [private]

Funkcja ustawiająca punkty do znalezienia. Funkcja ustala współrzędne punkty początkowego oraz końcowego. Następnie przydziela je do danego typu wierzchołka.

Parameters

<i>startX</i>	
<i>startY</i>	
<i>stopX</i>	
<i>stopY</i>	

Definition at line 71 of file graf.cpp.

4.2.4 Friends And Related Function Documentation

4.2.4.1 ostream& operator<< (ostream & wyjście, graf & zmienna) [friend]

Zdefiniowany operator wyświetlania Operator wypisuje na wyjściu utworzone dane.

Returns

wyjście

Definition at line 175 of file graf.hh.

4.2.5 Member Data Documentation

4.2.5.1 Punkt graf::koniec [private]

Pole typu [Punkt](#), będzie używane do przechowywania informacji o punkcie końcowym.

Definition at line 101 of file graf.hh.

4.2.5.2 Punkt graf::poczatek [private]

Pole typu [Punkt](#), będzie używane do przechowywania informacji o punkcie początkowym.

Definition at line 97 of file graf.hh.

4.2.5.3 vector< vector<Wierzcholek> > graf::w

Pole typu vector<vector<>>, będzie używane do przechowywania informacji o wierzchołku.

Definition at line 158 of file graf.hh.

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

- /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/[graf.hh](#)
- /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/[graf.cpp](#)

4.3 Punkt Struct Reference

Struktura punktu Struktura ta ma zdefiniowane dwie zmienne x oraz y, które odpowiadają za przechowywanie pozycji na siatce. Również zdefiniowane są operator przypisania oraz operator logiczny relacji.

```
#include <graf.hh>
```

Public Member Functions

- bool `operator==` ([Punkt](#) wsp)

Zdefiniowany operator porownania. Jezeli zmienne x i y sa rowne wspolrzednym tych zmiennych to zwracana jest wartosc 1.

- bool `operator!=` ([Punkt](#) wsp)

Zdefiniowany operator relacji.

Public Attributes

- int `x`

Pole typu int, bedzie uzywane do przechowywania wspolrzednej x-owej.

- int `y`

Pole typu int, bedzie uzywane do przechowywania wspolrzednej y-kowej.

4.3.1 Detailed Description

Struktura punktu Struktura ta ma zdefiniowane dwie zmienne x oraz y, ktore odpowiadaja za przechowywanie pozycji na siatce. Rowniez zdefiniowane sa operator przypisania oraz operator logiczny relacji.

Definition at line 38 of file graf.hh.

4.3.2 Member Function Documentation

4.3.2.1 bool `Punkt::operator!=` (`Punkt` wsp) `[inline]`

Zdefiniowany operator relacji.

Definition at line 59 of file graf.hh.

4.3.2.2 bool `Punkt::operator==` (`Punkt` wsp) `[inline]`

Zdefiniowany operator porownania. Jezeli zmienne x i y sa rowne wspolrzednym tych zmiennych to zwracana jest wartosc 1.

Definition at line 52 of file graf.hh.

4.3.3 Member Data Documentation

4.3.3.1 int `Punkt::x`

Pole typu int, bedzie uzywane do przechowywania wspolrzednej x-owej.

Definition at line 43 of file graf.hh.

4.3.3.2 int `Punkt::y`

Pole typu int, bedzie uzywane do przechowywania wspolrzednej y-kowej.

Definition at line 47 of file graf.hh.

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

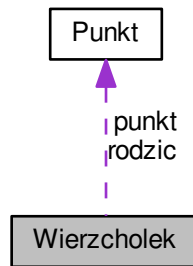
- /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/graf.hh

4.4 Wierzcholek Struct Reference

Struktura [Wierzcholek](#). Opisuje własności wierzchołka.

```
#include <graf.hh>
```

Collaboration diagram for Wierzcholek:



Public Attributes

- [Wierzcholek_Typ](#) typ
Pole typu [Wierzcholek_Typ](#), będzie używane do przechowywania informacji o typie wierzchołka.
- int [f](#)
*Pola typu int, będą używane do przechowywania wartości funkcji *f*, *g* oraz *h*.*
- int [g](#)
- int [h](#)
- [Punkt](#) punkt
Pole typu [Punkt](#), będzie używane do odnoszenia się do aktualnego punktu na układzie.
- [Punkt](#) rodzic
Pole typu [Punkt](#), będzie używane do odnoszenia się do poprzedniego punktu na układzie.

4.4.1 Detailed Description

Struktura [Wierzcholek](#). Opisuje własności wierzchołka.

Definition at line 68 of file graf.hh.

4.4.2 Member Data Documentation

4.4.2.1 int Wierzcholek::f

Pola typu int, będą używane do przechowywania wartości funkcji *f*, *g* oraz *h*.

Definition at line 76 of file graf.hh.

4.4.2.2 int Wierzcholek::g

Definition at line 76 of file graf.hh.

4.4.2.3 int Wierzcholek::h

Definition at line 76 of file graf.hh.

4.4.2.4 Punkt Wierzcholek::punkt

Pole typu [Punkt](#), bedzie uzywane do odnoszenia sie do aktualnego punktu na ukkladzie.

Definition at line 80 of file graf.hh.

4.4.2.5 Punkt Wierzcholek::rodzic

Pole typu [Punkt](#), bedzie uzywane do odnoszenia sie do poprzedniego punktu na ukkladzie.

Definition at line 84 of file graf.hh.

4.4.2.6 Wierzcholek_Typ Wierzcholek::typ

Pole typu Wierzcholek_Typ, bedzie uzywane do przechowywania informacji o typie wierzcholka.

Definition at line 72 of file graf.hh.

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

- [/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/graf.hh](#)

Chapter 5

File Documentation

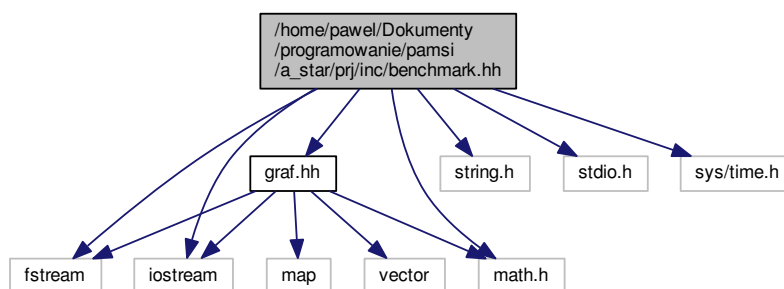
5.1 `/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/doc/pages/strona.dox` File Reference

5.2 `/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/benchmark.hh` File Reference

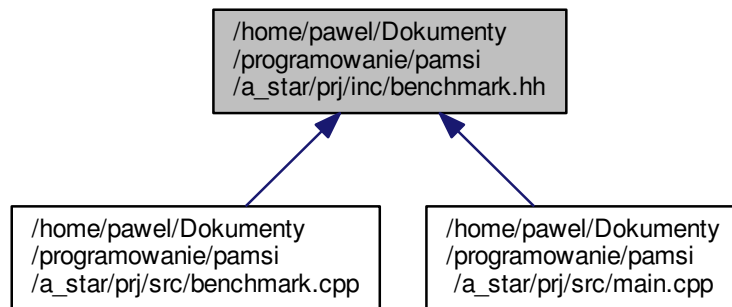
Definicje funkcji dla klasy benchmark.

```
#include <fstream>
#include <iostream>
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <sys/time.h>
#include "graf.hh"
```

Include dependency graph for benchmark.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class [benchmark](#)
Modeluje pojecie Benchmark.

5.2.1 Detailed Description

Definicje funkcji dla klasy `benchmark`.

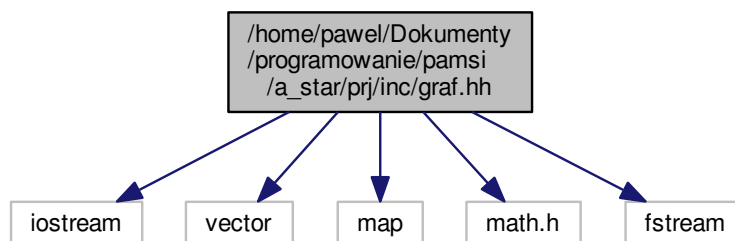
Definition in file [benchmark.hh](#).

5.3 `/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/graf.hh` File Reference

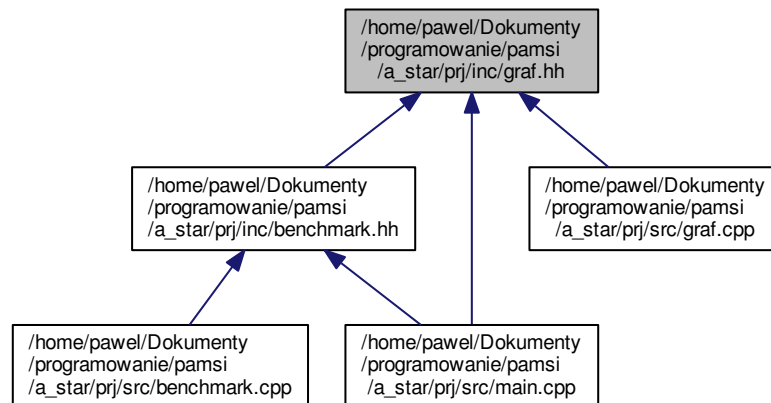
Definicje funkcji dla klasy `graf`.

```
#include <iostream>
#include <vector>
#include <map>
#include <math.h>
#include <fstream>
```

Include dependency graph for `graf.hh`:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Punkt](#)

Struktura punktu Struktura ta ma zdefiniowane dwie zmienne *x* oraz *y*, które odpowiadają za przechowywanie pozycji na siatce. Również zdefiniowane są operator przypisania oraz operator logiczny relacji.

- struct [Wierzcholek](#)

Struktura Wierzcholek. Opisuje własności wierzchołka.

- class [graf](#)

Modeluje pojęcie graf. Klasa służy głównie do wykonania algorytmu *a_star*, czyli znalezienia najlepszej drogi między dwoma punktami.

Enumerations

- enum [Wierzcholek_Typ](#) { [normalne](#), [sciana](#), [punkt](#) }

Rodzaje wierzchołków Zdefiniowanie 3 różnych rodzajów wierzchołka : -> *normalne* dla wierzchołków wolnych, -> *sciana* dla wierzchołków będących przeszkodą, -> *punkt* dla wierzchołków, które są punktami początkowymi i końcowymi.

5.3.1 Detailed Description

Definicje funkcji dla klasy [graf](#).

Definition in file [graf.hh](#).

5.3.2 Enumeration Type Documentation

5.3.2.1 enum [Wierzcholek_Typ](#)

Rodzaje wierzchołków Zdefiniowanie 3 różnych rodzajów wierzchołka : -> *normalne* dla wierzchołków wolnych, -> *sciana* dla wierzchołków będących przeszkodą, -> *punkt* dla wierzchołków, które są punktami początkowymi i końcowymi.

Enumerator

normalne

sciana

punkt

Definition at line 23 of file graf.hh.

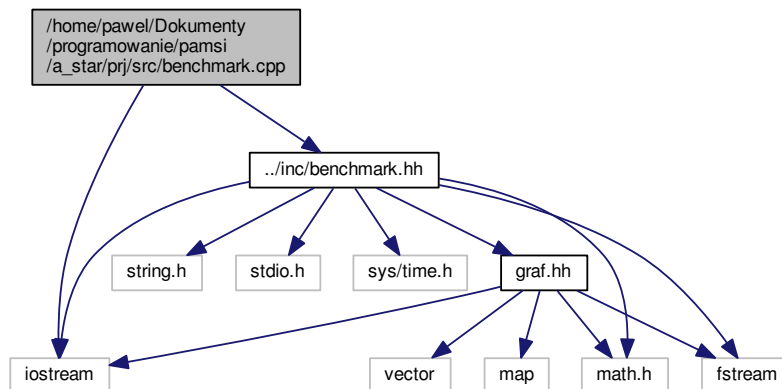
5.4 /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/benchmark.cpp File Reference

Plik zawiera funkcje z klasy benchmark.

```
#include "../inc/benchmark.hh"
```

```
#include <iostream>
```

Include dependency graph for benchmark.cpp:



5.4.1 Detailed Description

Plik zawiera funkcje z klasy benchmark.

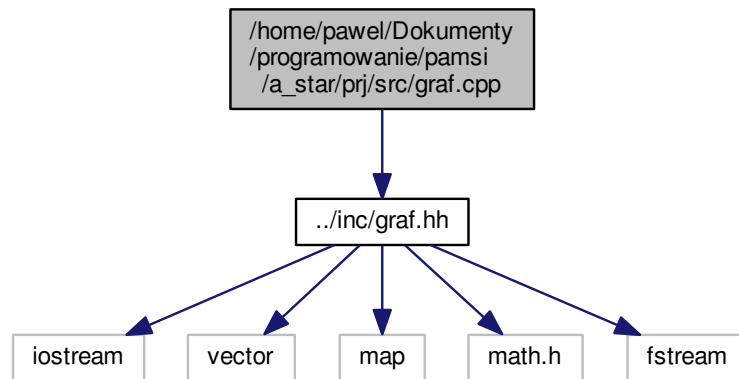
Definition in file [benchmark.cpp](#).

5.5 /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/graf.cpp File Reference

Plik zawiera funkcje z klasy graf.

```
#include "../inc/graf.hh"
```

Include dependency graph for graf.cpp:



5.5.1 Detailed Description

Plik zawiera funkcje z klasy graf.

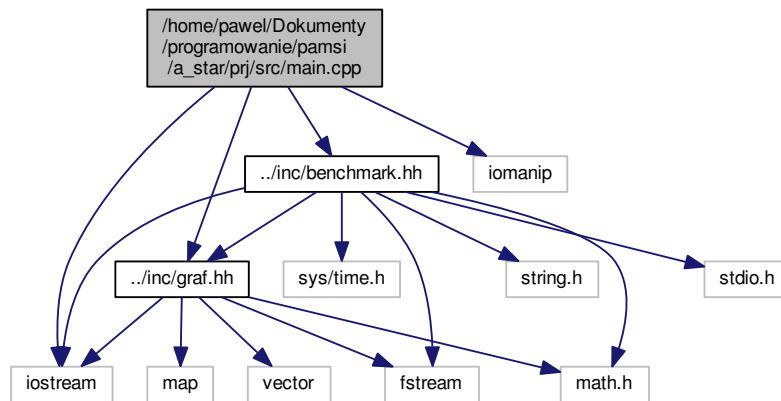
Definition in file [graf.cpp](#).

5.6 /home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/main.cpp File Reference

Plik zawiera funkcje [main\(\)](#)

```
#include <iostream>
#include <iomanip>
#include "../inc/graf.hh"
#include "../inc/benchmark.hh"
```

Include dependency graph for main.cpp:



Functions

- `int main ()`

5.6.1 Detailed Description

Plik zawiera funkcje `main()`

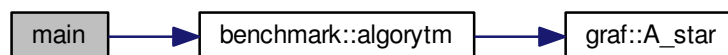
Definition in file `main.cpp`.

5.6.2 Function Documentation

5.6.2.1 `int main ()`

Definition at line 15 of file `main.cpp`.

Here is the call graph for this function:



Index

/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/doc/pages/strona.dox, [19](#)
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/benchmark.hh, [19](#)
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/inc/graf.hh, [20](#)
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/benchmark.cpp, [22](#)
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/graf.cpp, [22](#)
/home/pawel/Dokumenty/programowanie/pamsi/a_star/prj/src/main.cpp, [23](#)

A_star

graf, [12](#)

algorytm

benchmark, [8](#)

benchmark, [7](#)

algorytm, [8](#)
benchmark, [8](#)
czas, [9](#)
czas_caly, [9](#)
czas_start, [9](#)
czas_stop, [9](#)
elapsedTime, [9](#)
ile_czasu, [9](#)
size, [10](#)
t1, [10](#)
t2, [10](#)
tmp, [10](#)

czas

benchmark, [9](#)

czas_caly

benchmark, [9](#)

czas_start

benchmark, [9](#)

czas_stop

benchmark, [9](#)

elapsedTime

benchmark, [9](#)

F

graf, [12](#)

f

Wierzcholek, [16](#)

G

graf, [13](#)

g

Wierzcholek, [16](#)

graf, [10](#)

A_star, [12](#)

F, [12](#)

G, [13](#)

graf, [12](#)

H, [13](#)

koniec, [14](#)

operator<<, [14](#)

poczatek, [14](#)

Rozwiazanie, [13](#)

Stworz_Sciane, [13](#)

Ustaw_punkty, [13](#)

w, [14](#)

graf.hh

normalne, [22](#)

punkt, [22](#)

sciana, [22](#)

graf.hh

Wierzcholek_Typ, [21](#)

H

graf, [13](#)

h

Wierzcholek, [16](#)

ile_czasu

benchmark, [9](#)

koniec

graf, [14](#)

main

main.cpp, [24](#)

main.cpp

main, [24](#)

normalne

graf.hh, [22](#)

operator<<

graf, [14](#)

operator==

Punkt, [15](#)

poczatek

graf, [14](#)

Punkt, [14](#)

operator==, [15](#)

x, [15](#)

- y, [15](#)
- punkt
 - graf.hh, [22](#)
 - Wierzcholek, [17](#)
- rodzic
 - Wierzcholek, [17](#)
- Rozwiazanie
 - graf, [13](#)
- sciana
 - graf.hh, [22](#)
- size
 - benchmark, [10](#)
- Stworz_Sciane
 - graf, [13](#)
- t1
 - benchmark, [10](#)
- t2
 - benchmark, [10](#)
- tmp
 - benchmark, [10](#)
- typ
 - Wierzcholek, [17](#)
- Ustaw_punkty
 - graf, [13](#)
- w
 - graf, [14](#)
- Wierzcholek, [16](#)
 - f, [16](#)
 - g, [16](#)
 - h, [16](#)
 - punkt, [17](#)
 - rodzic, [17](#)
 - typ, [17](#)
- Wierzcholek_Typ
 - graf.hh, [21](#)
- x
 - Punkt, [15](#)
- y
 - Punkt, [15](#)