



WYDZIAŁ ELEKTRYCZNY POLITECHNIKI WARSZAWSKIEJ

JĘZYKI I METODY PROGRAMOWANIA II – PROJEKT 1

Siatkonator

Autor:
Barnaba TUREK
barnabaturek@gmail.com

6 marca 2012

Spis treści

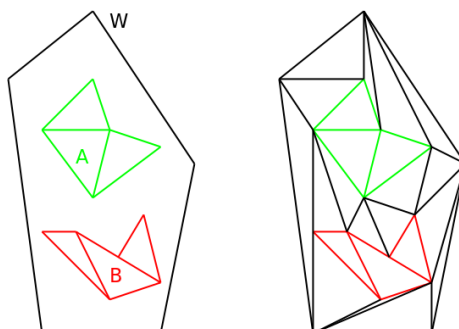
1	Specyfikacja funkcjonalna	1
1.1	Program	1
1.2	Wywołanie	1
1.3	Pliki	2
1.4	Dodatkowe uwagi	2
2	Specyfikacja implementacyjna	3
2.1	Struktura programu	3
2.1.1	Funkcje modułu <code>mesh_ops</code>	4
2.2	Algorytmy	4
2.3	struktury danych	5

1 Specyfikacja funkcjonalna

1.1 Program

Siatkonator to program sklejaący siatki trójkątne. Program przyjmuje jedną lub więcej siatek trójkątnych oraz jeden wielokąt.

Wynikiem działania programu jest siatka trójkątna opisująca zadany wielokąt, która zawiera podane określone siatki.



Rysunek 1: przykład sklejania zadanego wielokątu W i siatek A i B

1.2 Wywołanie

Program nie prowadzi dialogu z użytkownikiem.

Listing 1: Przykładowe wywołanie

```
1 $ siatkonator -e siatka1.ele -e siatka2.ele polygon.poly
```

Ostatni argument programu określa nazwę pliku (wraz ze ścieżką) w którym opisany jest wielokąt. Nie podanie tego argumentu spowoduje błąd wykonania programu.

Wcześniejsze argumenty określają opcje wykonania programu i nie są wymagane. Kolejność argumentów nie ma znaczenia.

Dostępne są następujące opcje:

- e <name> dodaje siatkę, która zostanie “sklejona” z wielokątem. Program zakłada, że istnieje także plik o takiej samej nazwie bazowej z rozszerzeniem `node` opisujący wierzchołki.
- o <name> podaje nazwę pliku wyjściowego. W przypadku braku tego parametru wynik zostanie wypisany na standardowe wejście (najpierw plik `ele`, potem `node`).
- a <area> określa maksymalną powierzchnię trójkąta (nie zmienia zadanych siatek!).
- q <degrees> określa minimalną miarę kąta w trójkącie w wygenerowanej siatce trójkątnej (nie zmienia zadanych siatek!),

W przypadku powodzenia program zwraca zero. W przeciwnym wypadku program zwraca jeden z kodów błędów:

`kod 1` Błąd otwarcia pliku (spowodowany np. brakiem uprawnień lub pliku).

`kod 2` Błąd wczytania pliku (spowodowany błędnym formatem któregoś z plików źródłowych).

`kod 3` Błędny format argumentów linii poleceń.

Ponadto program w czasie działania wypisuje informacje o swoim aktualnym stanie na wyjście `STDERR`.

1.3 Pliki

Siatkonator korzysta z plików w formatach `poly`, `ele` i `node`.

poly format pliku opisującego wielokąt

ele format pliku opisującego z których wierzchołków składają się trójkąty siatki

poly format pliku opisującego wierzchołki siatki

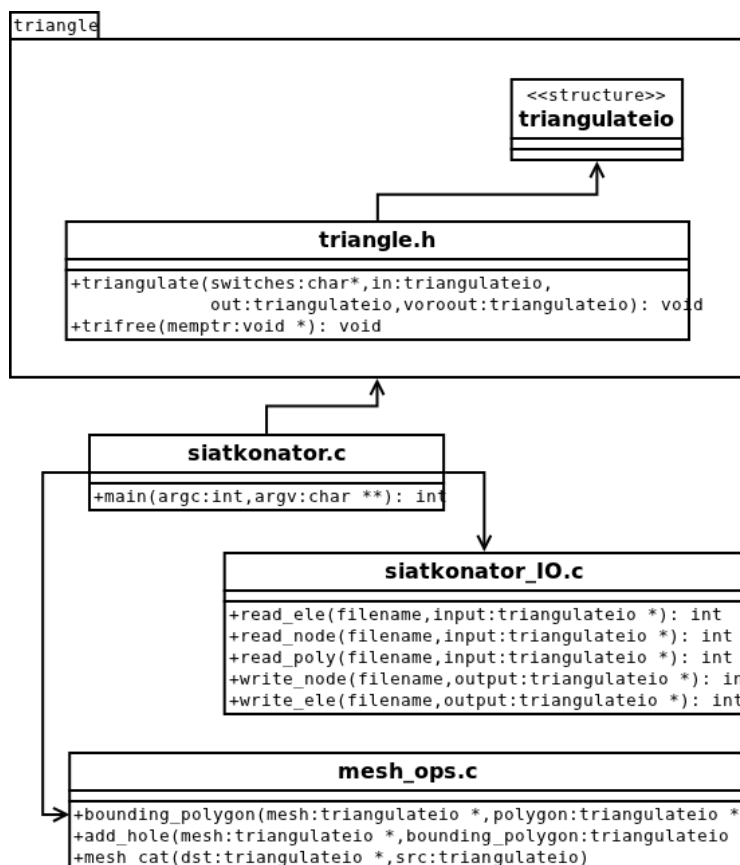
Wszystkie wykorzystywane formaty są zgodne z formatami wykorzystywanymi przez program `triangle`.

1.4 Dodatkowe uwagi

Odradzam użytkownikowi podawanie programowi siatek, które przecinają się ze sobą, bądź przecinają się z podanym wielokątem. Jeżeli użytkownik koniecznie chce podać bezsensowne dane, to otrzyma bezsensowne wyniki. Taka sytuacja nie jest rozumiana jako błąd programu i nie jest do niej przypisany żaden kod błędu.

2 Specyfikacja implementacyjna

2.1 Struktura programu



Rysunek 2: Diagram klas

Program podzielony jest na trzy pliki:

siatkonator.c Plik zawierający główną funkcję odpowiedzialną za wczytanie argumentów i zależnie od nich wykonanie algorytmu sklejania siatek.

siatkonator_IO.c Plik zawierający funkcje związane z czytaniem i zapisywaniem plików *node*, *ele* i *poly* do struktur **triangulateio**.

mesh_ops.c Plik zawierający funkcje obsługujące logikę sklejania siatek (znajdowanie wielokąta otaczającego siatkę, wycinanie dziur w siatkach i łączenie siatek).

common.c Plik zawierający funkcje nie związane z głównym celem istnienia programu (takie jak wypisywanie wiadomości).

2.1.1 Funkcje modułu mesh_ops

bounding_polygon funkcja przyjmująca wskaźnik na wczytaną gotową siatkę i znajdującą otaczający ją wielokąt (który należy wyciąć z otoczki przed przeprowadzeniem triangulacji).

add_hole Funkcja przyjmująca wskaźnik na wczytaną otoczkę (być może już z dziurami) i na wielokąt otaczający jedną ze sklejanых siatek. Funkcja dodaje do otoczki dziurę w miejscu, w którym zostanie w nią wklejona siatka.

mesh_cat Funkcja przyjmująca dwa wskaźniki na siatki i dodająca elementy drugiej siatki do elementów pierwszej siatki.

2.2 Algorytmy

Listing 2: Pseudokod algorytmu głównej funkcji programu sklejającej siatki

```
1  wczytaj dane z pliku poly do struktury p
2  dla kazdego pliku .ele:
3      wczytaj dane z plikow ele i nodes do struktury E
4      znajdz w strukturze E krawedzie, ktore wystepuja tylko raz
5      usun te krawedzie ze struktury E
6      dodaj te krawedzie do struktury P jako sekcje
7      dodaj dziure do struktury P wewnatrz kazdego trojkata ze
        struktury E
8  wykonaj triangulacje na strukturze p do siatki S
9  dla kazdego pliku .ele:
10     wstaw siatke z pliku do dziur w siatce S
11  zapisz wynik triangulacji do plikow ele i node
```

Listing 3: Pseudokod funkcji szukającej wielokąta otaczającego daną siatkę: **bounding_polygon**

```
1  struct element {
2      double lower_node_id
3      double higher_node_id
4      boolean more_than_once
5  }
6
7  T = element[]
8
9  Dla kazdego elementu e:
10     pos = znajdz w T element
11     if pos:
12         T[pos]->more_than_once = true
13         continue
14     else:
15         T[pos]->lower_node_id = e->lower_node_id
16         T[pos]->higher_node_id = e->higher_node_id
17         T[pos]->more_than_once = false
18  Dla kazdego elementu e w T:
19     Jezeli e->more_than_once == true:
20         dodaj e jako punkt w danych wyjsciowych
21     barycenter = srodek ciiezkości pierwszego trojkata w T
22     dodaj dziure do danych wyjsciowych w punkcie barycenter
```

2.3 struktury danych

Struktura danych, której będę używał do komunikacji między funkcjami to struktura `triangulateio` dostarczona wraz z programem `triangle`. Struktura ta pozwala opisać siatkę na każdym etapie jej budowania (także siatki wyjściowe).

Jej wadą jest fakt, że nie jest specjalizowana do konkretnych zadań i wiele pól nie będzie miało zastosowania. Mimo to jest już napisana, udokumentowana i mogę być pewien, że zawiera wszystkie potrzebne pola.

Zastosowanie innej struktury, jako struktury pośredniej wymagałoby tłumaczenia jej do `triangulateio` przed wykonaniem funkcji `triangulate()`.