

Triangulacja metodą QuadTree

Dokumentacja użytkownika

Tomasz Cudek
Marcin Paśko

Ściągnięcie, kompilacja i uruchomienie

Kody źródłowe wizualizatora oraz triangulatora są dostępne na *Githubie* pod adresem:

<http://github.com/mpasko/geometry.git>

Aby pobrać kody źródłowe przydatny jest klient Gita np wbudowany w powłokę Linuksa lub *TortoiseGit* pod Windowsem:

<http://code.google.com/p/tortoisegit/>

Aby pobrać projekt należy

Kody źródłowe triangulatora zostały napisane w języku C++, znajdują się w podkatalogu *./Triangulacja/*

Były rozwijane i kompilowane przy użyciu *NetBeans* IDE:

<http://netbeans.org/>

Współpracuje on z zestawem kompilatorów dostarczonych przez *Cygwina* lub *MinGW*:

<http://www.cygwin.com/>

<http://sourceforge.net/projects/mingw/files/>

Wizualizator został napisany w języku Java. Kody źródłowe znajdują się w katalogu *./Wizualizator/projekt_wizualizatora/src/*

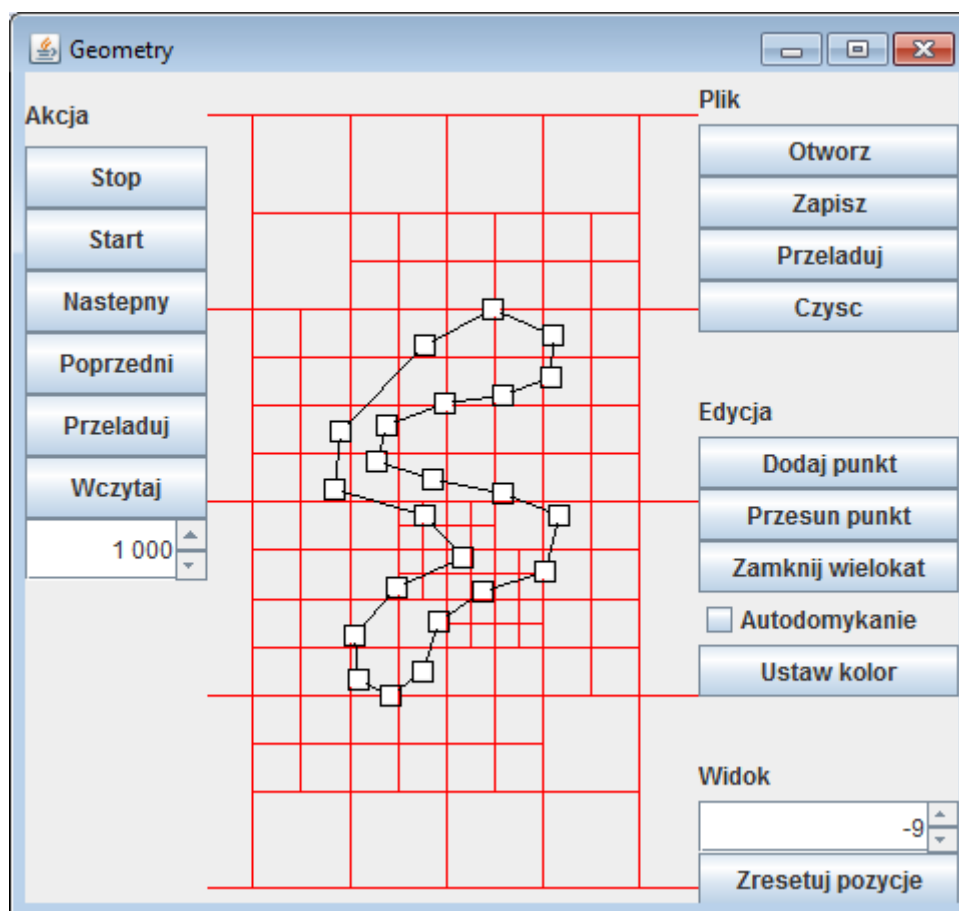
Kompilację i uruchomienie może umożliwić dowolne IDE języka Java np wspomniany wyżej *NetBeans* lub *Eclipse*:

<http://www.eclipse.org/>

Skompilowany projekt gotowy do uruchomienia pod każdą platformą znajduje się w katalogu: *./Wizualizator/geometria.jar*

Komunikacja pomiędzy triangulatorem a wizualizatorem odbywa się poprzez plik, którego format i sposób używania zostanie przedstawiony w kolejnych punktach dokumentacji.

Wizualizator



Plik

- Otwórz -umożliwia wczytywanie końcowego efektu pracy algorytmu (bez animacji)
- Zapisz -umożliwia zapis aktualnego stanu punktów do pliku (bez animacji)
- Przeglądaj -wczytuje ponownie ostatnio otwarty plik
- Czyść -usuwa wszystkie punkty wraz z wielokątami z programu

Edycja

- Dodaj punkt -umożliwia dodawanie nowego punktu poprzez kliknięcie na obszar graficzny
- Przesuń punkt -umożliwia zmianę położenia punktu
- Zamknij wielokąt -tworzy nowy wielokąt z ostatnio dodanych punktów, które nie wchodzi jeszcze w skład żadnego wielokąta
- Autodomykanie -sprawia że każdy nowo dodany punkt staje się samodzielnym elementem graficznym i nie może być włączony do żadnego wielokąta
- Ustaw kolor -ustawia kolor dodawanych punktów i wielokątów

Widok

- Zoom -kontrolka umożliwia zmianę powiększenia obrazka na obszarze graficznym
- Zresetuj pozycję -ustawia wyjściową pozycję i powiększenie obrazka

Akcja

- Stop -zatrzymuje animację kroków
- Start -wznawia animację kroków algorytmu
- Następny -przechodzi do następnego kroku
- Poprzedni -przechodzi do poprzedniego kroku
- Przeładuj -wczytuje symulację ponownie
- Wczytaj -wczytuje symulację algorytmu z pliku
- Interwał -kontrolka umożliwia zmianę interwału czasowego animacji

Format danych

Format ten umożliwia wymianę danych pomiędzy wizualizatorem a programem implementującym algorytm jako kolejne kroki symulacji.

Jest on generowany automatycznie przez program, specyfikacja:

```
-100.0, -100.0, 00ff00, A, 1  
-100.0, 100.0, 00ff00, B, 1  
step
```

Dane:

- [współrzędna x],
- [współrzędna y],
- [kolor rgb szesnastkowo],
- [etykieta punktu],
- [numer wielokąta, do którego należy punkt]

Komendy:

- step -oznacza kolejny krok algorytmu

Program implementujący algorytm

```
$>quad_tree.exe polygon plik [plik_wyjsciowy]
```

- Wczytuje punkty z pliku o nazwie "*plik*" i traktuje jako punkty wielokąta do triangulacji

```
$>quad_tree.exe points plik [plik_wyjsciowy]
```

- Wczytuje punkty z pliku o nazwie "*plik*" i traktuje je jako niepołączone ze sobą punkty do triangulacji

```
$>quad_tree.exe random rectangle a b liczba_punktów  
[plik_wyjsciowy]
```

- Generuje losowy zestaw punktów o liczebności "*liczba_punktów*" wewnątrz prostokąta o bokach długości *a* oraz *b*

```
$>quad_tree.exe random circle r liczba_punktów  
[plik_wyjsciowy]
```

- Generuje losowy zestaw punktów o liczebności "*liczba_punktów*" wewnątrz koła o promieniu *r*
- *plik_wyjsciowy* – plik do którego zapisany zostanie wynik. Jeżeli nie zostanie podany, wynik zostanie zapisany do pliku "*sim_out.txt*"