AKADEMIA GÓRNICZO-HUTNICZA

IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI



DOKUMENTACJA UŻYTKOWNIKA

**NARZĘDZIE DO WIZUALIZACJI SIATEK TRÓJWYMIAROWYCH**

WOJCIECH DYMEK

KATARZYNA GŁĄB

KATARZYNA KONIECZNA

EWA MARCZEWSKA

OPIEKUN:

Dr inż. Tomasz Jurczyk

Spis treści

[Podstawowe informacje 3](#_Toc467067437)

[Wymagania aplikacji 3](#_Toc467067438)

[Konfiguracja środowiska 3](#_Toc467067439)

[Kompilacja 3](#_Toc467067440)

[Uruchomienie aplikacji 3](#_Toc467067441)

[Plik konfiguracyjny *user.config.xml* 4](#_Toc467067442)

[Struktura pliku 4](#_Toc467067443)

[Interfejs programistyczny użytkownika API 5](#_Toc467067444)

[Podstawowe użycie narzędzia 5](#_Toc467067445)

[Dodatkowa funkcjonalność 5](#_Toc467067446)

[Wykorzystanie API w różnych językach programowania 6](#_Toc467067447)

[Java 6](#_Toc467067448)

[Python 6](#_Toc467067449)

[C++ 6](#_Toc467067450)

[Smeshalist Manager 7](#_Toc467067451)

[Statistics 7](#_Toc467067452)

[Options 8](#_Toc467067453)

[Filters 9](#_Toc467067454)

[Types 9](#_Toc467067455)

[Groups 10](#_Toc467067456)

[Quality 11](#_Toc467067457)

[Coordinates 12](#_Toc467067458)

[Spis ilustracji 13](#_Toc467067459)

# Podstawowe informacje

Smeshalist to aplikacja służąca do wizualizacji siatek trójwymiarowych oraz ich elementów. Jej celem jest ułatwienie m. in. tworzenia algorytmów z zakresu geometrii obliczeniowej poprzez wyświetlanie kolejnych kroków rozwiązania, zaimplementowanego w języku Java, C++ lub Python. Program może być uruchamiany na systemie operacyjnym Linuks lub Windows.

## Wymagania aplikacji

Do poprawnego uruchomienia aplikacji zainstalować należy:

1. Java JRE 1.8
2. GLUT – OpenGL Utility Toolkit

# Konfiguracja środowiska

## Kompilacja

Znajdując się w folderze głównym aplikacji uruchomić polecenie *make all.* Zostanie wygenerowany plik wykonywalny *SmeshalistCore*, który odpowiada za komunikację między poszczególnymi częściami systemu oraz wizualizację.

## Uruchomienie aplikacji

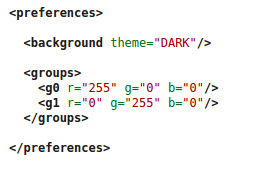
Kroki potrzebne do poprawnego uruchomienia aplikacji:

1. W razie konieczności zmiana pliku konfiguracyjnego *user.config.xml*.
2. Uruchomienie *SmeshalistCore*.

# Plik konfiguracyjny *user.config.xml*

Plik konfiguracyjny powinien znajdować się w tej samej lokalizacji co plik wykonywalny *SmeshalistCore*.

## Struktura pliku



Rysunek 1 Przykładowa struktura pliku konfiguracyjnego

Konfiguracja znajduje się w węźle *<preferences>*. Dostępne opcje to:

1. *<background > -* konfiguracja koloru tła obszaru roboczego. W przypadku nie podania węzła, bądź nie uzupełnienia atrybutu *theme* domyślnie kolor jest biały. W celu ustawienia ciemnego tła należy dodać powyższy węzeł wraz z atrybutem *theme=”DARK”.*
2. *<groups>* - konfiguracja kolorów zdefiniowanych dla poszczególnych grup. Aby ustawić kolor wybranej grupie należy dodać potomka w postaci węzła *<gX>* gdzie *X* oznacza numer grupy wraz z atrybutami *r, g, b.*

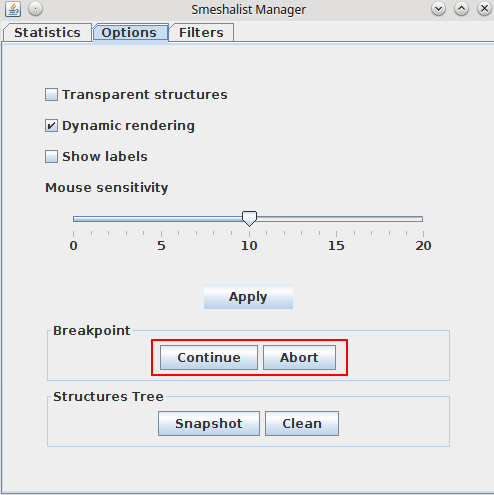
# Interfejs programistyczny użytkownika API

## Podstawowe użycie narzędzia

1. Wywołanie metody *getInstance* z możliwością podania portu, na którym odbywać się będzie komunikacja z narzędziem.
2. Stworzenie struktur geometrycznych (*Vertex, Edge, TriangleFace, Block).*
3. Dodanie geometrii do przesłania z wykorzystaniem metody *addGeometry.*
4. Przesłanie struktur do wizualizacji metodą *flushBuffer.*

## Dodatkowa funkcjonalność

1. *Breakpoint ­*– funkcja umożliwiająca wstrzymanie wykonania algorytmu. Jego kontynuacja jest możliwa poprzez wybranie opcji *continue* w zakładce *Options* narzędzia *Smeshalist Manager.*



Rysunek 2 Smeshalist Manager zakładka Options

1. *Render –* w przypadku wyłączonej opcji *Dynamic rendering* wymusza wyświetlenie przesłanych struktur w obszarze roboczym.
2. *Clean –* wymusza usunięcie struktur z aktywnego drzewa.

## Wykorzystanie API w różnych językach programowania

### Java

Aby wykorzystać narzędzie w języku Java należy dołączyć do projektu archiwuj jar znajdujące się w katalogu *JavaAPI*.

### Python

Aby wykorzystać narzędzie w języku Python należy zaimportować do skryptu moduły *Smeshalist* oraz *geometry* znajdujące się w katalogu *PythonAPI.*

### C++

Aby wykorzystać narzędzie w języku C++ należy zlinkować projekt z biblioteką *Smeshalist.a* znajdującym się w katalogu *CppAPI.*

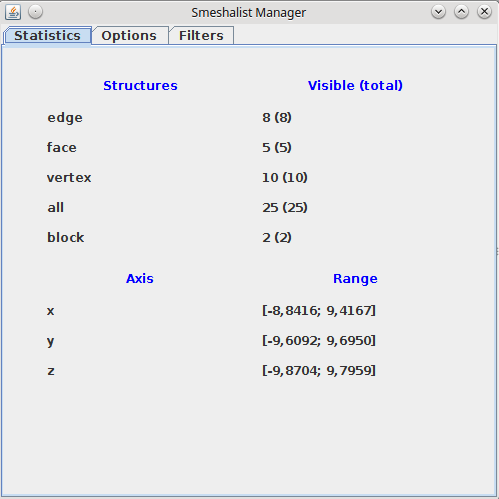
# Smeshalist Manager

Smeshalist Manager to okienko umożliwiające kontrolę nad strukturami wyświetlanymi w obszarze roboczym oraz wyświetlanie informacji o tych strukturach.

## Statistics

W zakładce statystyk wyświetlane są:

* ilość przesłanych i zwizualizowanych struktur danego typu
* współrzędne prostopadłościanu ograniczającego

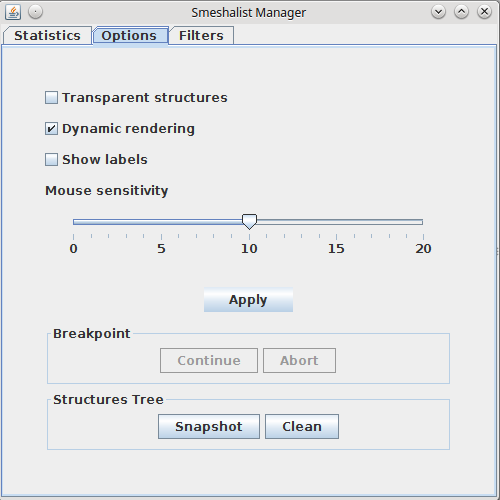


Rysunek 3 Smeshalist Manager - Statystyki

## Options

W zakładce opcji użytkownik ma możliwość:

* włączyć/wyłączyć dynamiczne renderowanie (*Dynamic rendering*)
* włączyć/wyłączyć wyświetlanie przezroczystych struktur (*Transparent structures*)
* włączyć/wyłączyć wyświetlanie etykiet struktur (*Show labels*)
* ustawić czułość myszki (*Mouse sensitivity*)
* zareagować na wywołanie metody *Breakpoint* (przyciski *Continue* i *Abort*)
* zrobić nowy zrzut drzewa struktur (przycisk *Snapshot*)
* wyczyścić aktywne drzewo struktur (przycisk *Clean*)



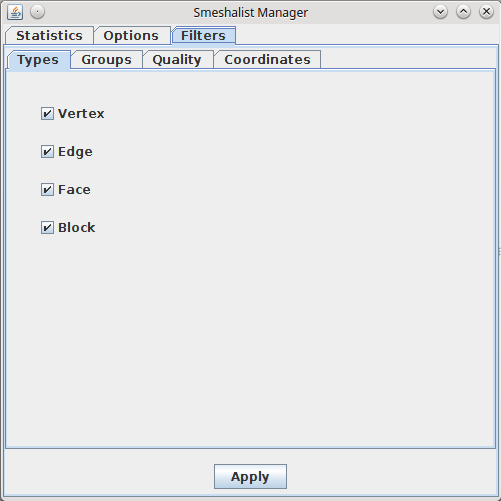
Rysunek 4 Smeshalist Manager - Opcje

## Filters

Zakładka filtrów oferuje użytkownikowi możliwość filtrowania wyświetlonych struktur z wykorzystaniem czterech rodzajów filtrów. Zmiany w filtrach należy zatwierdzić przyciskiem *Apply*, co spowoduje uruchomienie filtrowania.

### Types

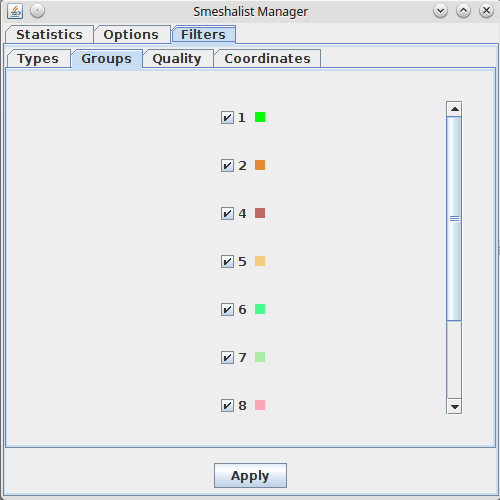
Umożliwia filtrowanie po typie struktury (Vertex, Edge, Face, Block).



Rysunek 5 Smeshalist Manager - Filtr Typy

### Groups

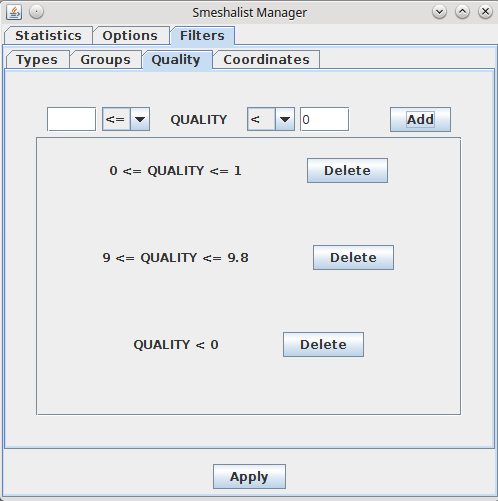
Umożliwia filtrowanie po ID grupy.



Rysunek 6 Smeshalist Manager - Filtr Grupy

### Quality

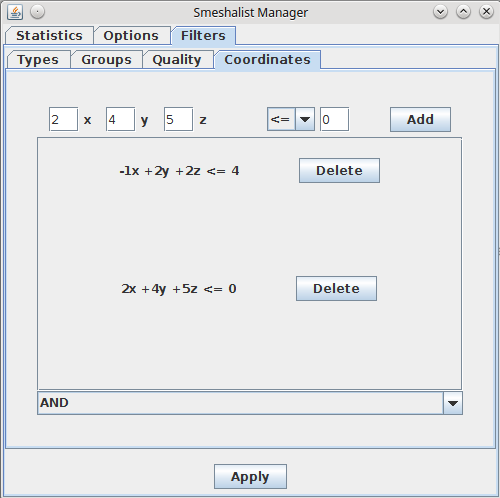
Umożliwia filtrowanie po właściwości Quality każdej struktury. Użytkownik dodaje warunek przyciskiem *Add*. Wszystkie warunki są łączone spójnikiem LUB. Aby usunąć odpowiedni warunek należy użyć odpowiadającemu mu przycisku *Delete*.



Rysunek 7 Smeshalist Manager - Filtr Jakość

### Coordinates

Umożliwia filtrowanie po współrzędnych struktur. Użytkownik dodaje warunek przyciskiem *Add*. Aby usunąć odpowiedni warunek należy użyć odpowiadającemu mu przycisku *Delete.* Warunki mogą być łączone spójnikiem I albo LUB. Filtr działa tylko na struktury, które w całości spełniają warunek (a nie np. tylko jeden z wierzchołków ściany).



Rysunek Smeshalist Manager - Filtr Współrzędne

# Spis ilustracji

[Rysunek 1 Przykładowa struktura pliku konfiguracyjnego 4](#_Toc467067417)

[Rysunek 2 Smeshalist Manager zakładka Options 5](#_Toc467067418)

[Rysunek 3 Smeshalist Manager - Statystyki 7](#_Toc467067419)

[Rysunek 4 Smeshalist Manager - Opcje 8](#_Toc467067420)

[Rysunek 5 Smeshalist Manager - Filtr Typy 9](#_Toc467067421)

[Rysunek 6 Smeshalist Manager - Filtr Grupy 10](#_Toc467067422)

[Rysunek 7 Smeshalist Manager - Filtr Jakość 11](#_Toc467067423)

[Rysunek 8 Smeshalist Manager - Filtr Współrzędne 12](#_Toc467067424)