

Środowisko do analizy systemów modelowanych jako hipergrafy lub hipersieci

Wygenerowano przez Doxygen 1.8.11

Spis treści

| | |
|--|----------|
| 1 Indeks przestrzeni nazw | 2 |
| 1.1 Pakiety | 2 |
| 2 Indeks hierarchiczny | 3 |
| 2.1 Hierarchia klas | 3 |
| 3 Indeks klas | 3 |
| 3.1 Lista klas | 3 |
| 4 Indeks plików | 4 |
| 4.1 Lista plików | 4 |
| 5 Dokumentacja przestrzeni nazw | 4 |
| 5.1 Dokumentacja przestrzeni nazw application-main | 4 |
| 5.1.1 Dokumentacja zmiennych | 5 |
| 5.2 Dokumentacja przestrzeni nazw application_main | 5 |
| 5.3 Dokumentacja przestrzeni nazw GuiWindow | 5 |
| 5.3.1 Dokumentacja funkcji | 5 |
| 5.4 Dokumentacja przestrzeni nazw HgMatrixAnalyzer | 6 |
| 5.5 Dokumentacja przestrzeni nazw HyperGraph | 6 |
| 5.6 Dokumentacja przestrzeni nazw OCL | 6 |
| 5.6.1 Dokumentacja zmiennych | 7 |
| 5.7 Dokumentacja przestrzeni nazw Utils | 7 |
| 5.8 Dokumentacja przestrzeni nazw Vert | 7 |
| 5.9 Dokumentacja przestrzeni nazw VertArranger | 7 |

| | |
|---|----------|
| 6 Dokumentacja klas | 8 |
| 6.1 Dokumentacja klasy EditorDialog | 8 |
| 6.1.1 Opis szczegółowy | 10 |
| 6.1.2 Dokumentacja konstruktora i destruktora | 10 |
| 6.1.3 Dokumentacja funkcji składowych | 10 |
| 6.1.4 Dokumentacja atrybutów składowych | 11 |
| 6.2 Dokumentacja klasy EntryDialog | 12 |
| 6.2.1 Opis szczegółowy | 13 |
| 6.2.2 Dokumentacja konstruktora i destruktora | 13 |
| 6.2.3 Dokumentacja funkcji składowych | 13 |
| 6.2.4 Dokumentacja atrybutów składowych | 14 |
| 6.3 Dokumentacja klasy GuiWindow | 14 |
| 6.3.1 Opis szczegółowy | 18 |
| 6.3.2 Dokumentacja konstruktora i destruktora | 18 |
| 6.3.3 Dokumentacja funkcji składowych | 18 |
| 6.3.4 Dokumentacja atrybutów składowych | 40 |
| 6.4 Dokumentacja klasy TextResultDialog | 43 |
| 6.4.1 Opis szczegółowy | 44 |
| 6.4.2 Dokumentacja konstruktora i destruktora | 44 |
| 6.5 Dokumentacja klasy TreeViewResultDialog | 45 |
| 6.5.1 Opis szczegółowy | 46 |
| 6.5.2 Dokumentacja konstruktora i destruktora | 46 |
| 6.6 Dokumentacja klasy HgMatrixAnalyzer | 46 |
| 6.6.1 Opis szczegółowy | 47 |
| 6.6.2 Dokumentacja konstruktora i destruktora | 47 |
| 6.6.3 Dokumentacja funkcji składowych | 47 |
| 6.7 Dokumentacja klasy HyperGraph | 48 |
| 6.7.1 Opis szczegółowy | 52 |
| 6.7.2 Dokumentacja konstruktora i destruktora | 53 |
| 6.7.3 Dokumentacja funkcji składowych | 53 |

| | | |
|--------|---|-----|
| 6.7.4 | Dokumentacja atrybutów składowych | 95 |
| 6.8 | Dokumentacja klasy object | 96 |
| 6.9 | Dokumentacja klasy OCL | 97 |
| 6.9.1 | Opis szczegółowy | 98 |
| 6.9.2 | Dokumentacja funkcji składowych | 98 |
| 6.9.3 | Dokumentacja atrybutów składowych | 99 |
| 6.10 | Dokumentacja klasy Utils | 99 |
| 6.10.1 | Opis szczegółowy | 100 |
| 6.10.2 | Dokumentacja konstruktora i destruktora | 100 |
| 6.10.3 | Dokumentacja funkcji składowych | 100 |
| 6.11 | Dokumentacja klasy HBNODE | 103 |
| 6.11.1 | Opis szczegółowy | 106 |
| 6.11.2 | Dokumentacja konstruktora i destruktora | 106 |
| 6.11.3 | Dokumentacja funkcji składowych | 106 |
| 6.11.4 | Dokumentacja atrybutów składowych | 106 |
| 6.12 | Dokumentacja klasy Node | 107 |
| 6.12.1 | Opis szczegółowy | 109 |
| 6.12.2 | Dokumentacja konstruktora i destruktora | 110 |
| 6.12.3 | Dokumentacja atrybutów składowych | 110 |
| 6.13 | Dokumentacja klasy Vert | 110 |
| 6.13.1 | Opis szczegółowy | 115 |
| 6.13.2 | Dokumentacja konstruktora i destruktora | 115 |
| 6.13.3 | Dokumentacja funkcji składowych | 115 |
| 6.13.4 | Dokumentacja atrybutów składowych | 130 |
| 6.14 | Dokumentacja klasy VertArranger | 133 |
| 6.14.1 | Opis szczegółowy | 134 |
| 6.14.2 | Dokumentacja funkcji składowych | 134 |

| | |
|---|------------|
| 7 Dokumentacja plików | 136 |
| 7.1 Dokumentacja pliku app-python3-gtk3-cairo/application-main.py | 136 |
| 7.2 Dokumentacja pliku app-python3-gtk3-cairo/GuiWindow.py | 136 |
| 7.3 Dokumentacja pliku app-python3-gtk3-cairo/HgMatrixAnalyzer.py | 137 |
| 7.4 Dokumentacja pliku app-python3-gtk3-cairo/HyperGraph.py | 137 |
| 7.5 Dokumentacja pliku app-python3-gtk3-cairo/OCL.py | 137 |
| 7.6 Dokumentacja pliku app-python3-gtk3-cairo/OpenCIColors.cl | 138 |
| 7.6.1 Opis szczegółowy | 138 |
| 7.6.2 Dokumentacja funkcji | 138 |
| 7.7 Dokumentacja pliku app-python3-gtk3-cairo/Utils.py | 139 |
| 7.8 Dokumentacja pliku app-python3-gtk3-cairo/Vert.py | 140 |
| 7.9 Dokumentacja pliku app-python3-gtk3-cairo/VertArranger.py | 140 |
| Indeks | 141 |

1 Indeks przestrzeni nazw

1.1 Pakiety

Oto lista pakietów wraz z krótkim opisem (o ile jest dostępny):

| | |
|-------------------------|-----------|
| application-main | ?? |
| application_main | ?? |
| GuiWindow | ?? |
| HgMatrixAnalyzer | ?? |
| HyperGraph | ?? |
| OCL | ?? |
| Utils | ?? |
| Vert | ?? |
| VertArranger | ?? |

2 Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

| | |
|-----------------------------|------------|
| Dialog | |
| EditorDialog | 8 |
| EntryDialog | 12 |
| TextResultDialog | 43 |
| TreeViewResultDialog | 45 |
| Window | |
| GuiWindow | 14 |
| HgMatrixAnalyzer | 46 |
| object | 96 |
| HyperGraph | 48 |
| Vert | 110 |
| HBNode | 103 |
| Node | 107 |
| OCL | 97 |
| Utils | 99 |
| VertArranger | 133 |

3 Indeks klas

3.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

| | |
|--|-----------|
| EditorDialog | |
| Klasa edytora skryptów | ?? |
| EntryDialog | |
| Klasa EntryDialog | ?? |
| GuiWindow | |
| Klasa GuiWindow | ?? |
| TextResultDialog | |
| Klasa TextResultDialog Pozwala na wyświetlanie wyników w formie tekstu | ?? |
| TreeViewResultDialog | |
| Klasa TreeViewResultDialog Pozwala na wyświetlanie wyników w formie listy wierszy | ?? |

| | |
|--------------------------------|----|
| HgMatrixAnalyzer | |
| Klasa HgMatrixAnalyzer | ?? |
| HyperGraph | |
| Klasa HyperGraph | ?? |
| object | ?? |
| OCL | |
| Klasa narzędzi OpenCL | ?? |
| Utils | |
| Klasa Utils | ?? |
| HBNode | |
| Klasa HBNode | ?? |
| Node | |
| Klasa Node | ?? |
| Vert | |
| Klasa abstrakcyjna Vert | ?? |
| VertArranger | |
| Klasa VertArranger | ?? |

4 Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

| | |
|---|----|
| app-python3-gtk3-cairo/application-main.py | ?? |
| app-python3-gtk3-cairo/GuiWindow.py | ?? |
| app-python3-gtk3-cairo/HgMatrixAnalyzer.py | ?? |
| app-python3-gtk3-cairo/HyperGraph.py | ?? |
| app-python3-gtk3-cairo/OCL.py | ?? |
| app-python3-gtk3-cairo/OpenCIColors.cl | ?? |
| app-python3-gtk3-cairo/Utils.py | ?? |
| app-python3-gtk3-cairo/Vert.py | ?? |
| app-python3-gtk3-cairo/VertArranger.py | ?? |

5 Dokumentacja przestrzeni nazw

5.1 Dokumentacja przestrzeni nazw application-main

Zmienne

- **pa** = **GuiWindow()**
Część właściwa programu głównego.

5.1.1 Dokumentacja zmiennych

5.1.1.1 `pa = GuiWindow()`

Część właściwa programu głównego.

Nie jest wykonywana, w przypadku importu tego pliku do innego pliku. Obiekt okna aplikacji.

Definicja w linii 23 pliku `application-main.py`.

5.2 Dokumentacja przestrzeni nazw `application_main`5.3 Dokumentacja przestrzeni nazw `GuiWindow`

Komponenty

- class `EditorDialog`
Klasa edytora skryptów.
- class `EntryDialog`
Klasa `EntryDialog`.
- class `GuiWindow`
Klasa `GuiWindow`.
- class `TextResultDialog`
Klasa `TextResultDialog` Pozwala na wyświetlanie wyników w formie tekstu.
- class `TreeViewResultDialog`
Klasa `TreeViewResultDialog` Pozwala na wyświetlanie wyników w formie listy wierszy.

Funkcje

- def `redraw_function_decorator` (method)
Dekorator na funkcje, po których należy odświeżyć ekran.
- def `undo_redo_function_decorator` (method)
Dekorator na funkcje, które modyfikują hipergraf.

5.3.1 Dokumentacja funkcji

5.3.1.1 `def GuiWindow.redraw_function_decorator (method)`

Dekorator na funkcje, po których należy odświeżyć ekran.

Pozwala obudować funkcje, aby wykonać akcje przed i po wykonaniu funkcji.

Parametry

| | |
|---------------|------------------------------------|
| <i>method</i> | Metoda, która ma zostać obudowana. |
|---------------|------------------------------------|

Zwraca

Zwraca dalej funkcję, która obudowuje daną funkcję.

Definicja w linii 58 pliku GuiWindow.py.

5.3.1.2 def GuiWindow.undo_redo_function_decorator (method)

Dekorator na funkcje, które modyfikują hipergraf.

Pozwala obudować funkcje, aby wykonać akcje przed i po wykonaniu funkcji.

Parametry

| | |
|---------------|------------------------------------|
| <i>method</i> | Metoda, która ma zostać obudowana. |
|---------------|------------------------------------|

Zwraca

Zwraca dalej funkcję, która obudowuje daną funkcję.

Definicja w linii 36 pliku GuiWindow.py.

5.4 Dokumentacja przestrzeni nazw HgMatrixAnalyzer**Komponenty**

- class [HgMatrixAnalyzer](#)
Klasa [HgMatrixAnalyzer](#).

5.5 Dokumentacja przestrzeni nazw HyperGraph**Komponenty**

- class [HyperGraph](#)
Klasa [HyperGraph](#).

5.6 Dokumentacja przestrzeni nazw OCL**Komponenty**

- class [OCL](#)
Klasa narzędzi OpenCL.

Zmienne

- bool [DEFAULT_ENABLE_OPENCL](#) = False
Zmienna ustawiająca, czy program powinien korzystać z OpenCL.

5.6.1 Dokumentacja zmiennych

5.6.1.1 bool DEFAULT_ENABLE_OPENCL = False

Zmienna ustawiająca, czy program powinien korzystać z OpenCL.

DEFAULT_ENABLE_OPENCL = True

Definicja w linii 10 pliku OCL.py.

5.7 Dokumentacja przestrzeni nazw Utils

Komponenty

- class [Utils](#)
Klasa [Utils](#).

5.8 Dokumentacja przestrzeni nazw Vert

Komponenty

- class [HBNode](#)
Klasa [HBNode](#).
- class [Node](#)
Klasa [Node](#).
- class [Vert](#)
Klasa abstrakcyjna [Vert](#).

5.9 Dokumentacja przestrzeni nazw VertArranger

Komponenty

- class [VertArranger](#)
Klasa [VertArranger](#).

6 Dokumentacja klas

6.1 Dokumentacja klasy EditorDialog

Klasa edytora skryptów.

Diagram dziedziczenia dla EditorDialog

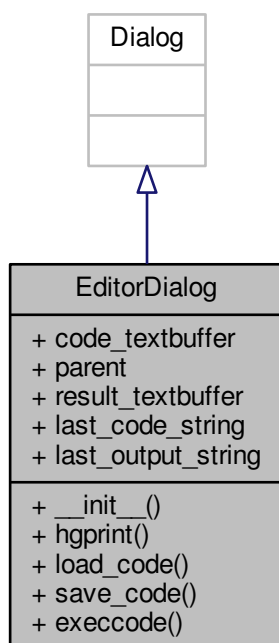
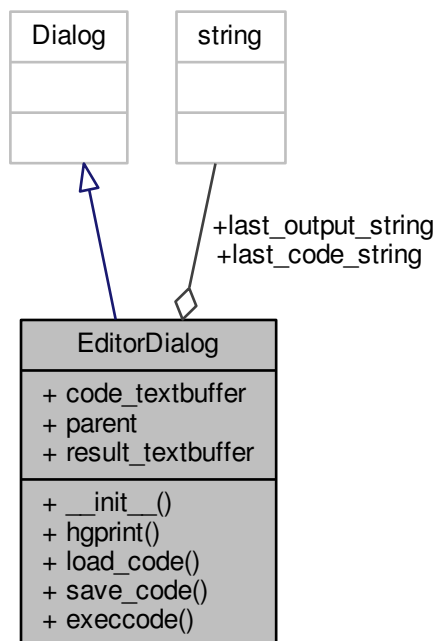


Diagram współpracy dla EditorDialog:



Metody publiczne

- `def __init__ (self, parent, dialog_name="Edytor")`
Konstruktor.
- `def hgprint (self, inputstr)`
Metoda pozwalająca na wyświetlanie tekstu w polu wynikowym edytora.
- `def load_code (self, butt=None)`
Metoda pozwalająca odczytać kod z pliku i wczytać go do okna edytora.
- `def save_code (self, butt=None)`
Metoda pozwalająca zapisać kod z edytora do pliku.

Statyczne metody publiczne

- `def execcode (win, editor, button=None)`
Metoda uruchamiająca napisany przez użytkownika kod.

Atrybuty publiczne

- `code_textbuffer`
Bufor zawierający tekst pola tekstowego kodu.
- `parent`
Obiekt klasy, która wywołała okno edytora.
- `result_textbuffer`
Bufor zawierający tekst pola tekstowego wyniku.

Statyczne atrybuty publiczne

- string `last_code_string` = `'hgprint('\Hello World')\nhgprint(hg)'`
- string `last_output_string`

6.1.1 Opis szczegółowy

Klasa edytora skryptów.

Pozwala na napisanie, wczytanie, zapisanie, uruchomienie skryptu w celu analizy hipergrafu. Składa się z pola edytora kodu oraz pola wyników.

Definicja w linii 222 pliku `GuiWindow.py`.

6.1.2 Dokumentacja konstruktora i destruktora

6.1.2.1 `def __init__(self, parent, dialog_name = "Edytor")`

Konstruktor.

Definicja w linii 247 pliku `GuiWindow.py`.

6.1.3 Dokumentacja funkcji składowych

6.1.3.1 `def execcode(win, editor, button = None) [static]`

Metoda uruchamiająca napisany przez użytkownika kod.

Definicja w linii 404 pliku `GuiWindow.py`.

6.1.3.2 `def hgprint(self, inputstr)`

Metoda pozwalająca na wyświetlanie tekstu w polu wynikowym edytora.

Definicja w linii 397 pliku `GuiWindow.py`.

6.1.3.3 `def load_code(self, butt = None)`

Metoda pozwalająca odczytać kod z pliku i wczytać go do okna edytora.

Definicja w linii 362 pliku `GuiWindow.py`.

6.1.3.4 `def save_code(self, butt = None)`

Metoda pozwalająca zapisać kod z edytora do pliku.

Definicja w linii 328 pliku `GuiWindow.py`.

6.1.4 Dokumentacja atrybutów składowych

6.1.4.1 code_textbuffer

Bufor zawierający tekst pola tekstowego kodu.

Definicja w linii 276 pliku GuiWindow.py.

6.1.4.2 string last_code_string = 'hgprint('\Hello World')\nhgprint(hg)' [static]

Definicja w linii 223 pliku GuiWindow.py.

6.1.4.3 string last_output_string [static]

Wartość początkowa:

```
1 = """Wynik pojawi się tutaj.  
2  
3 Dostępne zmienne  
4 \thg - obiekt hipergrafu  
5 \twin - obiekt okna aplikacji  
6 \tUtils - klasa narzędzi pomocniczych  
7 \tMa - klasa zawierająca algorytmy  
8 \tOCL - klasa narzędzi OpenCL  
9  
10 Do wyświetlania tekstu służy funkcja hgprint()  
11  
12 Należy zwrócić szczególną uwagę, że do zmiennej \thg\  
13 przypisywana jest wartość z \win.active_hg\  
14 co oznacza, że w wyniku wykonywania komend  
15 cofnięcia, powtórzenia, zapisania  
16 lub wczytania stanu ewolucji hipergrafu  
17 wartość w zmiennej \thg\  
18 od tej w \win.active_hg\  
19 Należy ją wtedy przypisać ponownie  
20 lub w całym skrypcie korzystać  
21 jedynie z \win.active_hg\."""
```

Definicja w linii 224 pliku GuiWindow.py.

6.1.4.4 parent

Obiekt klasy, która wywołała okno edytora.

Definicja w linii 252 pliku GuiWindow.py.

6.1.4.5 result_textbuffer

Bufor zawierający tekst pola tekstowego wyniku.

Definicja w linii 290 pliku GuiWindow.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `app-python3-gtk3-cairo/GuiWindow.py`

6.2 Dokumentacja klasy EntryDialog

Klasa [EntryDialog](#).

Diagram dziedziczenia dla EntryDialog

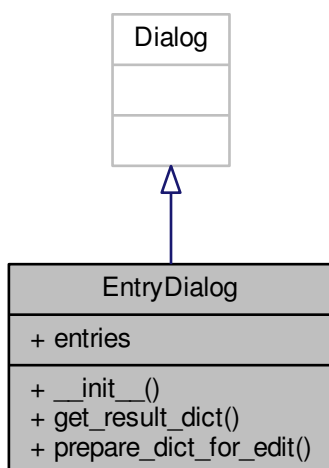
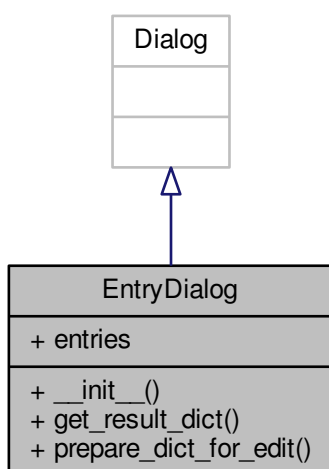


Diagram współpracy dla EntryDialog:



Metody publiczne

- `def __init__ (self, parent, dialog_name, labels_entries_dict)`
Konstruktor.
- `def get_result_dict (self)`
Metoda zwracająca słownik wyjściowy.

Statyczne metody publiczne

- `def prepare_dict_for_edit (dict_for_edit)`
Przekształca wartości elementów słownika na postać tekstową, z której będzie dało się odzyskać oryginalny typ.

Atrybuty publiczne

- `entries`
Słownik na pola wprowadzania tekstu.

6.2.1 Opis szczegółowy

Klasa `EntryDialog`.

Pozwala wyświetlać pola z podpisami, w celu ich edycji. Następnie pobiera się z jej obiektu słownik z wynikowymi wartościami.

Definicja w linii 78 pliku `GuiWindow.py`.

6.2.2 Dokumentacja konstruktora i destruktora

6.2.2.1 `def __init__ (self, parent, dialog_name, labels_entries_dict)`

Konstruktor.

Definicja w linii 80 pliku `GuiWindow.py`.

6.2.3 Dokumentacja funkcji składowych

6.2.3.1 `def get_result_dict (self)`

Metoda zwracająca słownik wyjściowy.

Wartości słownika zostały przekształcone z wartości tekstowej na oryginalny typ tej wartości.

Zwraca

Słownik wyjściowy z przywróconymi typami wartości.

Definicja w linii 115 pliku `GuiWindow.py`.

6.2.3.2 `def prepare_dict_for_edit (dict_for_edit) [static]`

Przekształca wartości elementów słownika na postać tekstową, z której będzie dało się odzyskać oryginalny typ.

Parametry

| | |
|----------------------------|---------------------------|
| <code>dict_for_edit</code> | Słownik do przygotowania. |
|----------------------------|---------------------------|

Definicja w linii 105 pliku GuiWindow.py.

6.2.4 Dokumentacja atrybutów składowych

6.2.4.1 entries

Słownik na pola wprowadzania tekstu.

Definicja w linii 90 pliku GuiWindow.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `app-python3-gtk3-cairo/GuiWindow.py`

6.3 Dokumentacja klasy GuiWindow

Klasa `GuiWindow`.

Diagram dziedziczenia dla GuiWindow

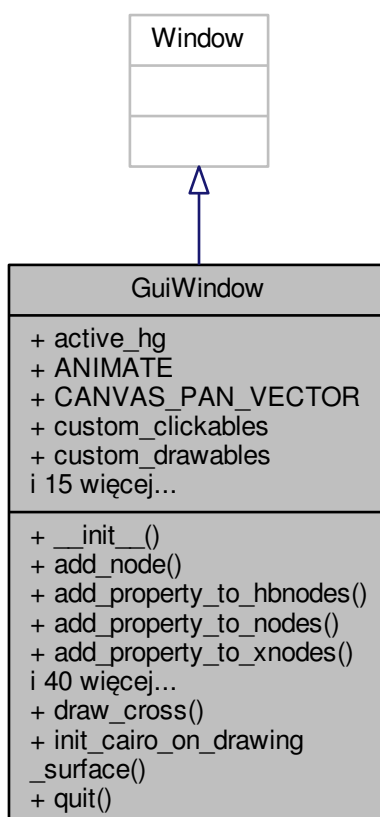
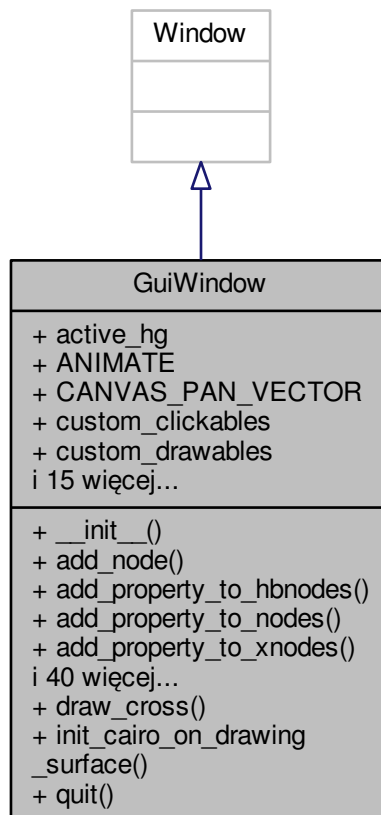


Diagram współpracy dla GuiWindow:



Metody publiczne

- `def __init__(self)`
Konstruktor klasy okna głównego.
- `def add_node(self, pos)`
Metoda pozwalająca utworzyć wierzchołek w danym miejscu.
- `def add_property_to_hbnodes(self, button=None)`
Metoda pozwalająca dodać nową właściwość do hipergałęzi hipergrafu.
- `def add_property_to_nodes(self, button=None)`
Metoda pozwalająca dodać nową właściwość do wierzchołków hipergrafu.
- `def add_property_to_xnodes(self, button=None)`
Metoda pozwalająca dodać nową właściwość do wierzchołków i hipergałęzi hipergrafu.
- `def after_modify_action_handler(self)`
Metoda wykonywana po akcji edytującej w jakiś sposób stan hipergrafu.
- `def before_modify_action_handler(self)`
Metoda wykonywana przed akcją edytującą w jakiś sposób stan hipergrafu.
- `def deactivate_all(self, button=None)`
Metoda usuwająca wyróżnienie ze wszystkich elementów hipergrafu.

- def `delete_selected_xnode` (self, button=None)
Metoda usuwająca zaznaczone elementy hipergrafu.
- def `deselect_xnode` (self, xid)
Metoda pozwalająca odznaczyć element hipergrafu po jego id.
- def `documenting_dummy_fun` (self)
- def `draw_canvas_cross` (self, cr)
Metoda rysująca krzyż środka płótna rysowania Cairo.
- def `draw_canvas_grid` (self, cr)
Metoda rysująca siatkę tła okna wyświetlania.
- def `draw_center_cross` (self, cr)
Metoda rysująca krzyż środka ekranu.
- def `edit_all_selected_xnodes` (self, button=None)
Metoda pozwalająca edytować wszystkie zaznaczone elementy hipergrafu.
- def `edit_selected_xnode` (self, button=None)
Metoda pozwalająca edytować ostatni zaznaczony element hipergrafu.
- def `expose` (self, da, cr)
Metoda powodująca rysowanie hipergrafu.
- def `init_button_panel` (self)
Metoda inicjująca pasek przycisków.
- def `init_drawing_area` (self)
Metoda inicjująca okno rysowania.
- def `init_drawing_frame` (self)
Metoda inicjująca ramkę rysowania.
- def `init_drawing_surface` (self, frame, drawing_area)
Metoda inicjująca powierzchnię rysowania.
- def `init_events_callbacks` (self)
Metoda przypisująca metody obsługi zdarzeń odpowiednim zdarzeniom na obiektach .
- def `init_evo_scale` (self)
Metoda inicjująca suwak historii ewolucji.
- def `init_toolbar` (self)
Metoda inicjująca pasek narzędzi.
- def `key_pressed` (self, widget, event)
Metoda pozwalająca reagować na wcisnięte klawisze na klawiaturze.
- def `load_hypergraph_from_file` (self, button=None, path=None)
Metoda ładująca zserializowany plik hipergrafu do projektu.
- def `make_hyperarc` (self, button=None)
Metoda pozwalająca utworzyć nowy hiperłuk.
- def `make_hyperedge` (self, button=None)
Metoda pozwalająca utworzyć nową hiperkrawędź.
- def `make_hyperloop` (self, button=None)
Metoda pozwalająca utworzyć nową hiperpętlę.
- def `make_node` (self, button=None)
Metoda pozwalająca utworzyć nowy wierzchołek.
- def `mouse_button_clicked` (self, widget, event)
Metoda pozwalająca reagować na kliknięcie przyciskiem myszy.
- def `mouse_moved_while_clicked` (self, widget, event)
Metoda pozwalająca reagować na przeciągnięcie myszy podczas kliknięcia.
- def `mouse_wheel_scrolled` (self, widget, event)
Metoda pozwalająca reagować na obrót rolki myszy.
- def `redo_action` (self, button=None)
Metoda pozwalająca powtórzyć cofniętą akcję.

- `def reset_hypergraph (self, button=None, hg=None)`
Metoda czyszcząca całkowicie wszystkie elementy hipergrafu.
- `def reset_view_pan_and_scale (self, button=None)`
Metoda resetująca przesunięcie i powiększenie widoku.
- `def save_hypergraph_to_file (self, button=None, path=None)`
Metoda zapisująca zserializowany plik hipergrafu do pliku.
- `def scale_value_changed (self, event)`
Metoda pozwalająca reagować na zmiany położenia suwaka historii ewolucji hipergrafu.
- `def select_toggle_colliding_xnode (self, xid)`
Metoda pozwalająca zaznaczyć lub odznaczyć kliknięty element hipergrafu.
- `def select_xnode (self, xid)`
Metoda pozwalająca zaznaczyć element hipergrafu po jego id.
- `def set_active_hypergraph (self, hg)`
Metoda ustawiająca dany hipergraf jako aktywny.
- `def show_matrices (self, button=None)`
Metoda wyświetlająca okno z listą macierzy hipergrafu.
- `def toggle_animate (self, button=None)`
Metoda pozwalająca włączyć lub wyłączyć animację elementów hipergrafu.
- `def toggle_select_all (self, button=None)`
Metoda pozwalająca zaznaczyć lub odznaczyć element hipergrafu po jego id.
- `def undo_action (self, button=None)`
Metoda pozwalająca cofnąć cofniętą wykonaną akcję.

Statyczne metody publiczne

- `def draw_cross (cr, p, r_cnv)`
Metoda rysująca krzyż w danym punkcie o danej wielkości.
- `def init_cairo_on_drawing_surface (surface)`
Inicjalizacja Cairo na danej powierzchni rysowania.
- `def quit (widget=None)`
Metoda realizująca zamknięcie aplikacji.

Atrybuty publiczne

- `active_hg`
Obiekt aktualnie używanego hipergrafu.
- `ANIMATE`
Zmienna ustalająca czy hipergraf ma być animowany.
- `CANVAS_PAN_VECTOR`
Wektor przesunięcia punktu środka płaszczyzny rysowania elementów w stosunku do lewego górnego rogu ekranu.
- `custom_clickables`
Lista dodatkowych elementów zdefiniowanych przez użytkownika, które mają reagować na kliknięcia.
- `custom_drawables`
Lista dodatkowych elementów zdefiniowanych przez użytkownika, które mają być wyświetlone na płótnie.
- `DEBUG_MODE`
Zmienna ustalająca tryb testowania.
- `dialog`
- `DRAWING_AREA_CENTER`
Zmienna przechowująca aktualny punkt środka okna rysowania jako tablica NumPy.
- `DRAWING_AREA_SIZE`

- Zmienna przechowująca aktualną wielkość okna rysowania jako tablica NumPy.*
- [frame](#)
Ramka na pole do rysowania.
- [MAXZOOM](#)
Maksymalna wartość powiększenia płótna.
- [MINZOOM](#)
Minimalna wartość powiększenia płótna.
- [old_mouse_button](#)
Zmienna przechowująca ostatnio używany przycisk myszy.
- [old_mouse_pos](#)
Zmienna przechowująca poprzednią pozycję myszy przy kliknięciu.
- [redo_states](#)
Zmienna przechowująca stany chronologicznie nowsze, niż aktualnie cofnięty.
- [START_TIME_SEC](#)
Zmienna czasu startu aplikacji.
- [surface](#)
Powierzchnia, na której Cairo będzie rysował.
- [TIME_DELTA_SEC](#)
Stała czasowa, która wyraża ilość czasu jaki musi minąć pomiędzy dwoma kolejnymi odświeżeniami widoku.
- [undo_states](#)
Zmienna przechowująca poprzednie stany hipergrafu, aby móc przywrócić stan przy cofnięciu akcji.
- [ZOOM](#)
Współczynnik powiększenia płaszczyzny rysowania elementów w stosunku do bazy współrzędnych ekranu.

6.3.1 Opis szczegółowy

Klasa [GuiWindow](#).

Główna klasa aplikacji, tworzy okno edycji hipergrafu.

Definicja w linii 427 pliku GuiWindow.py.

6.3.2 Dokumentacja konstruktora i destruktora

6.3.2.1 `def __init__(self)`

Konstruktor klasy okna głównego.

Definicja w linii 429 pliku GuiWindow.py.

6.3.3 Dokumentacja funkcji składowych

6.3.3.1 `def add_node(self, pos)`

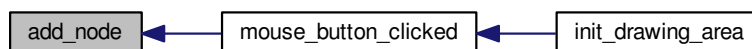
Metoda pozwalająca utworzyć wierzchołek w danym miejscu.

Parametry

| | |
|------------|---------------------------------|
| <i>pos</i> | Miejsce utworzenia wierzchołka. |
|------------|---------------------------------|

Definicja w linii 1135 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:

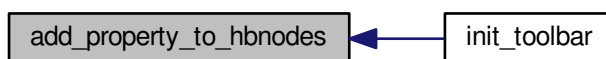


6.3.3.2 `def add_property_to_hbnodes (self, button = None)`

Metoda pozwalająca dodać nową właściwość do hipergałęzi hipergrafu.

Definicja w linii 1587 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:

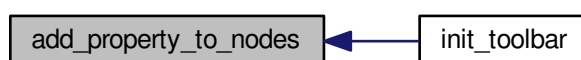


6.3.3.3 `def add_property_to_nodes (self, button = None)`

Metoda pozwalająca dodać nową właściwość do wierzchołków hipergrafu.

Definicja w linii 1558 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:

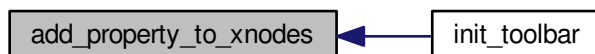


6.3.3.4 `def add_property_to_xnodes (self, button = None)`

Metoda pozwalająca dodać nową właściwość do wierzchołków i hipergałęzi hipergrafu.

Definicja w linii 1615 pliku `GuiWindow.py`.

Oto graf wywołań tej funkcji:



6.3.3.5 `def after_modify_action_handler (self)`

Metoda wykonywana po akcji edytującej w jakiś sposób stan hipergrafu.

Definicja w linii 1058 pliku `GuiWindow.py`.

6.3.3.6 `def before_modify_action_handler (self)`

Metoda wykonywana przed akcją edytującą w jakiś sposób stan hipergrafu.

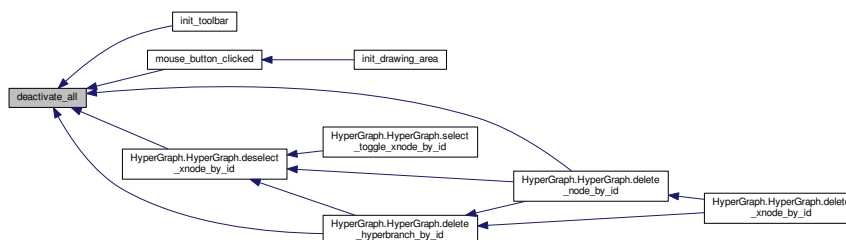
Definicja w linii 1051 pliku `GuiWindow.py`.

6.3.3.7 `def deactivate_all (self, button = None)`

Metoda usuwająca wyróżnienie ze wszystkich elementów hipergrafu.

Definicja w linii 1348 pliku `GuiWindow.py`.

Oto graf wywołań tej funkcji:

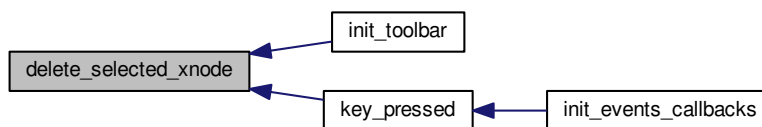


6.3.3.8 `def delete_selected_xnode (self, button = None)`

Metoda usuwająca zaznaczone elementy hipergrafu.

Definicja w linii 1550 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.9 `def deselect_xnode (self, xid)`

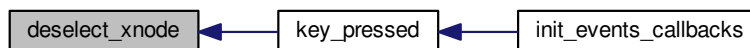
Metoda pozwalająca odznaczyć element hipergrafu po jego id.

Parametry

| | |
|------------------|-------------------------|
| <code>xid</code> | Id elementu hipergrafu. |
|------------------|-------------------------|

Definicja w linii 1363 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.10 `def documenting_dummy_fun (self)`

Definicja w linii 1819 pliku GuiWindow.py.

6.3.3.11 `def draw_canvas_cross (self, cr)`

Metoda rysująca krzyż środka płótna rysowania Cairo.

Parametry

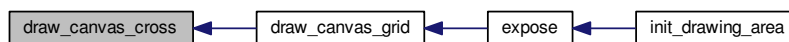
| | |
|-----------------|--------------------------|
| <code>cr</code> | Obiekt biblioteki Cairo. |
|-----------------|--------------------------|

Definicja w linii 1690 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.12 `def draw_canvas_grid(self, cr)`

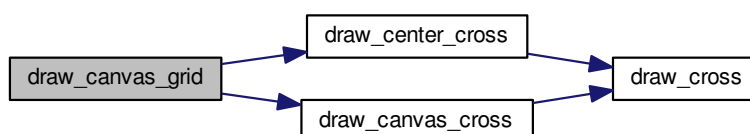
Metoda rysująca siatkę tła okna wyświetlania.

Parametry

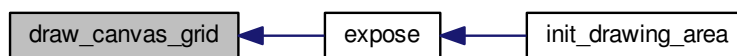
| | |
|-----------|--------------------------|
| <i>cr</i> | Obiekt biblioteki Cairo. |
|-----------|--------------------------|

Definicja w linii 1697 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.13 `def draw_center_cross (self, cr)`

Metoda rysująca krzyż środka ekranu.

Parametry

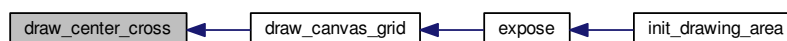
| | |
|-----------|--------------------------|
| <i>cr</i> | Obiekt biblioteki Cairo. |
|-----------|--------------------------|

Definicja w linii 1685 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.14 `def draw_cross (cr, p, r_canv) [static]`

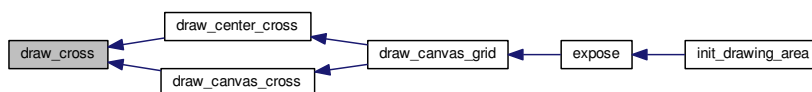
Metoda rysująca krzyż w danym punkcie o danej wielkości.

Parametry

| | |
|---------------|--------------------------|
| <i>cr</i> | Obiekt biblioteki Cairo. |
| <i>p</i> | Punkt środka. |
| <i>r_canv</i> | Promień na płótnie. |

Definicja w linii 1657 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



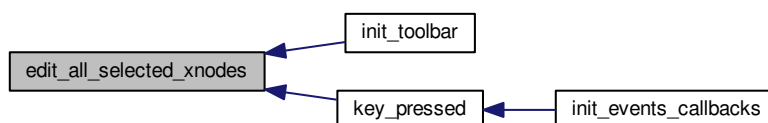
6.3.3.15 `def edit_all_selected_xnodes (self, button=None)`

Metoda pozwalająca edytować wszystkie zaznaczone elementy hipergrafu.

Wyświetlany jest dialog z polami, które można edytować.

Definicja w linii 1501 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.16 `def edit_selected_xnode (self, button=None)`

Metoda pozwalająca edytować ostatni zaznaczony element hipergrafu.

Wyświetlany jest dialog z polami, które można edytować.

Definicja w linii 1468 pliku GuiWindow.py.

6.3.3.17 `def expose (self, da, cr)`

Metoda powodująca rysowanie hipergrafu.

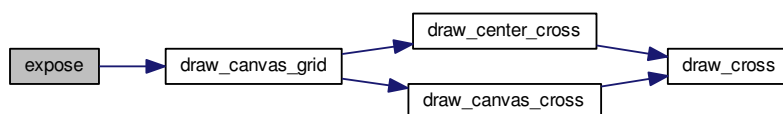
Jest wywoływana przez obiekt klasy `DrawingArea`.

Parametry

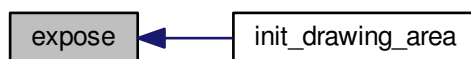
| | |
|-----------|--|
| <i>da</i> | Obiekt klasy <code>DrawingArea</code> z biblioteki <code>Gtk3</code> . |
| <i>cr</i> | Obiekt biblioteki <code>Cairo</code> . |

Definicja w linii 1746 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.18 def init_button_panel (self)

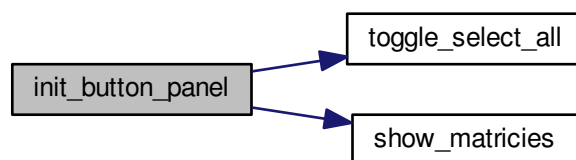
Metoda inicjująca pasek przycisków.

Zwraca

Obiekt paska przycisków.

Definicja w linii 610 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



6.3.3.19 def init_cairo_on_drawing_surface (surface) [static]

Inicjalizacja Cairo na danej powierzchni rysowania.

Parametry

| | |
|----------------------|---|
| <code>surface</code> | Powierzchnia rysowania na której rysować będzie Cairo . |
|----------------------|---|

Zwraca

Obiekt kontekstu Cairo.

Definicja w linii 914 pliku GuiWindow.py.

6.3.3.20 `def init_drawing_area (self)`

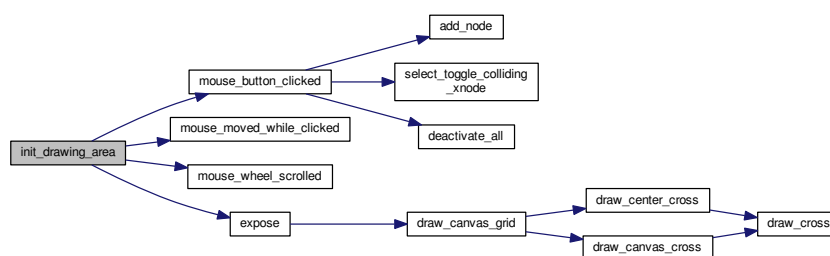
Metoda inicjująca okno rysowania.

Zwraca

Obiekt okna rysowania.

Definicja w linii 878 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



6.3.3.21 `def init_drawing_frame (self)`

Metoda inicjująca ramkę rysowania.

Zwraca

Obiekt ramki rysowania.

Definicja w linii 855 pliku GuiWindow.py.

6.3.3.22 `def init_drawing_surface (self, frame, drawing_area)`

Metoda inicjująca powierzchnię rysowania.

Parametry

| | |
|---------------------|------------------|
| <i>frame</i> | Ramka rysowania. |
| <i>drawing_area</i> | Okno rysowania. |

Zwraca

Obiekt powierzchni rysowania dla Cairo.

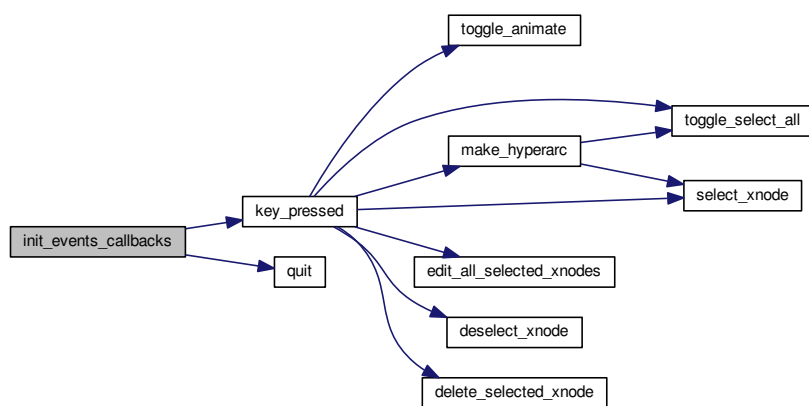
Definicja w linii 900 pliku GuiWindow.py.

6.3.3.23 `def init_events_callbacks (self)`

Metoda przypisująca metody obsługi zdarzeń odpowiednim zdarzeniom na obiektach .

Definicja w linii 921 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:

6.3.3.24 `def init_evo_scale (self)`

Metoda inicjująca suwak historii ewolucji.

Zwraca

Obiekty kontrolera suwaka i samego suwaka.

Definicja w linii 864 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



6.3.3.25 `def init_toolbar (self)`

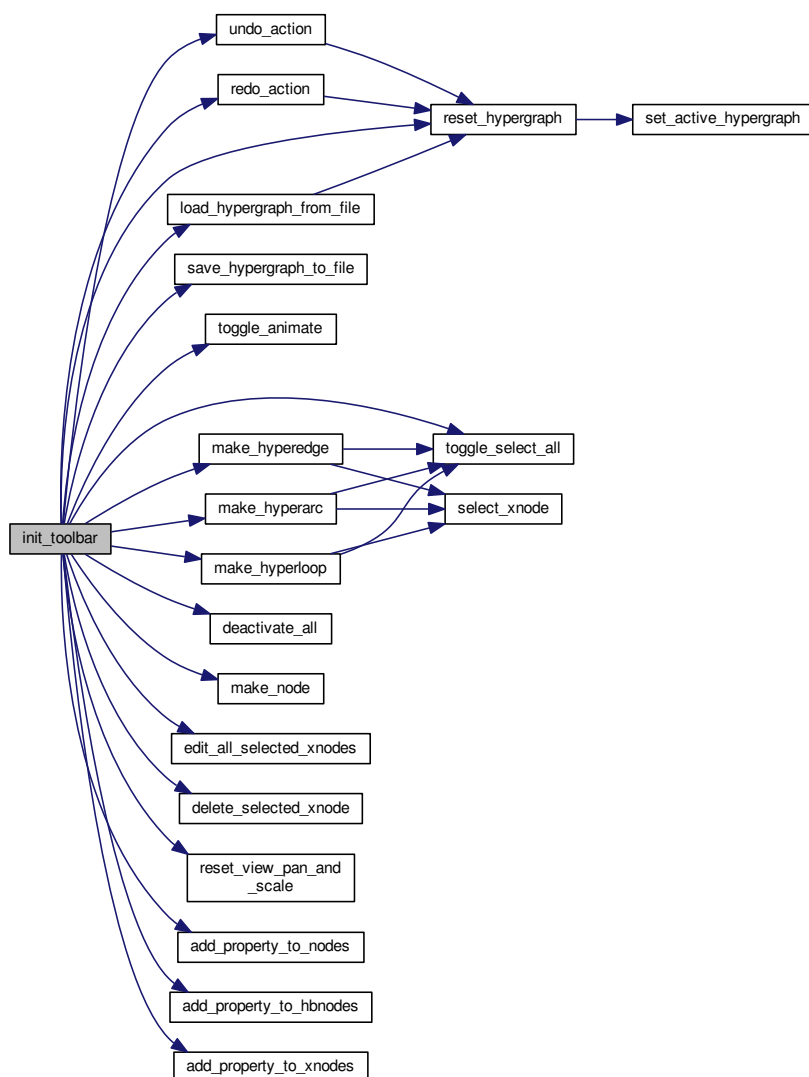
Metoda inicjująca pasek narzędzi.

Zwraca

Obiekt paska narzędzi.

Definicja w linii 538 pliku `GuiWindow.py`.

Oto graf wywołań dla tej funkcji:



6.3.3.26 `def key_pressed (self, widget, event)`

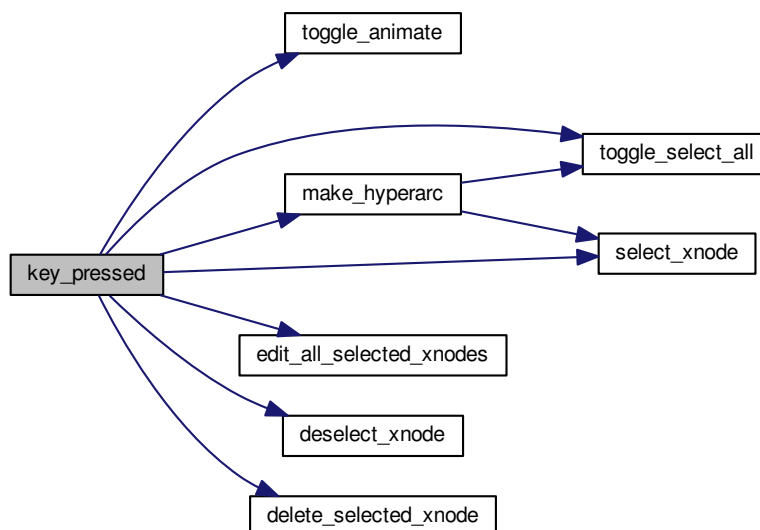
Metoda pozwalająca reagować na wciśnięte klawisze na klawiaturze.

Parametry

| | |
|--------------|---|
| <i>event</i> | Obiekt zdarzenia, które zawiera informacje m.in. o kodzie wciśniętego klawisza. |
|--------------|---|

Definicja w linii 1200 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.27 `def load_hypergraph_from_file (self, button = None, path = None)`

Metoda ładująca zserializowany plik hipergrafu do projektu.

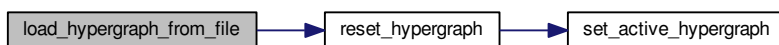
Wyświetlane jest okno wyboru pliku.

Parametry

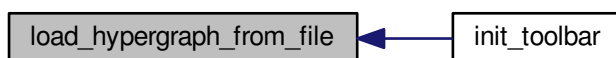
| | |
|---------------|--|
| <i>button</i> | Opcjonalna referencja do przycisku, w przypadku, gdy metoda jest wywoływana jako callback po wciśnięciu przycisku. |
| <i>path</i> | Opcjonalna ścieżka. Jeśli jest ustawiona, hipergraf ładowany jest z danego pliku bez okna wyboru. |

Definicja w linii 986 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



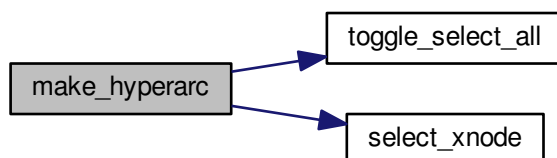
6.3.3.28 `def make_hyperarc(self, button=None)`

Metoda pozwalająca utworzyć nowy hiperłuk.

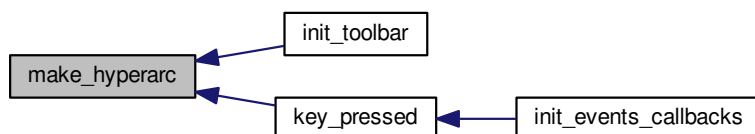
Wyświetlany jest dialog z polami, które można edytować.

Definicja w linii 1441 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



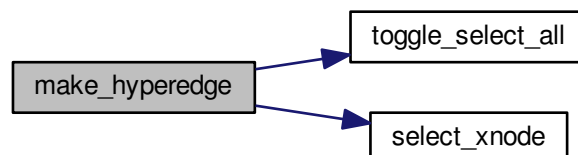
6.3.3.29 `def make_hyperedge (self, button = None)`

Metoda pozwalająca utworzyć nową hiperkrawędź.

Wyświetlany jest dialog z polami, które można edytować.

Definicja w linii 1376 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



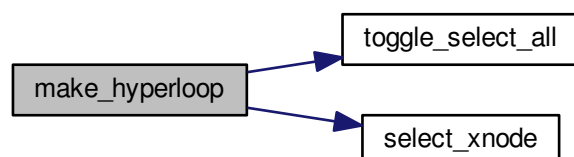
6.3.3.30 `def make_hyperloop (self, button = None)`

Metoda pozwalająca utworzyć nową hiperpętlę.

Wyświetlany jest dialog z polami, które można edytować.

Definicja w linii 1406 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.31 `def make_node (self, button =None)`

Metoda pozwalająca utworzyć nowy wierzchołek.

Definicja w linii 1368 pliku `GuiWindow.py`.

Oto graf wywoływań tej funkcji:



6.3.3.32 `def mouse_button_clicked (self, widget, event)`

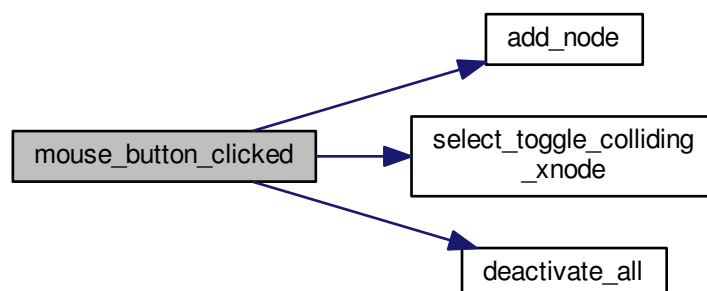
Metoda pozwalająca reagować na kliknięcie przyciskiem myszy.

Parametry

| | |
|--------------|--|
| <i>event</i> | Obiekt zdarzenia, które zawiera informacje m.in. o numerze przycisku i krotności kliknięcia. |
|--------------|--|

Definicja w linii 1071 pliku `GuiWindow.py`.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.33 `def mouse_moved_while_clicked (self, widget, event)`

Metoda pozwalająca reagować na przeciągnięcie myszy podczas kliknięcia.

Parametry

| | |
|--------------------|---|
| <code>event</code> | Obiekt zdarzenia, które zawiera informacje m.in. o wektorze przesunięcia. |
|--------------------|---|

Definicja w linii 1152 pliku `GuiWindow.py`.

Oto graf wywoływań tej funkcji:



6.3.3.34 `def mouse_wheel_scrolled (self, widget, event)`

Metoda pozwalająca reagować na obrót rolki myszy.

Parametry

| | |
|--------------|---|
| <i>event</i> | Obiekt zdarzenia, które zawiera informacje m.in. o kierunku obrotu. |
|--------------|---|

Definicja w linii 1182 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:

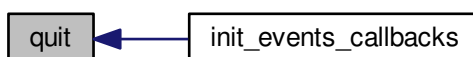


6.3.3.35 `def quit (widget=None) [static]`

Metoda realizująca zamknięcie aplikacji.

Definicja w linii 1815 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.36 `def redo_action (self, button=None)`

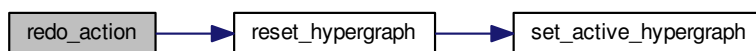
Metoda pozwalająca powtórzyć cofniętą akcję.

Parametry

| | |
|---------------|---|
| <i>button</i> | Obiekt przycisku, z którego ewentualnie wywoływana jest metoda. |
|---------------|---|

Definicja w linii 1322 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.37 `def reset_hypergraph (self, button = None, hg = None)`

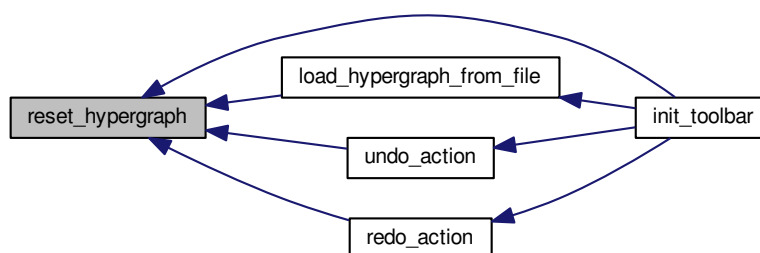
Metoda czyszcząca całkowicie wszystkie elementy hipergrafu.

Definicja w linii 1040 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.3.38 `def reset_view_pan_and_scale (self, button = None)`

Metoda resetująca przesunięcie i powiększenie widoku.

Definicja w linii 1034 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.39 `def save_hypergraph_to_file (self, button = None, path = None)`

Metoda zapisująca zserializowany plik hipergrafu do pliku.

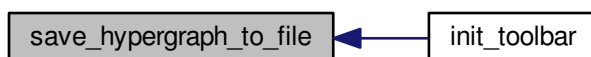
Wyświetlane jest okno wyboru folderu i nazwy pliku.

Parametry

| | |
|---------------|--|
| <i>button</i> | Opcjonalna referencja do przycisku, w przypadku, gdy metoda jest wywoływana jako callback po wciśnięciu przycisku. |
| <i>path</i> | Opcjonalna ścieżka. Jeśli jest ustawiona, hipergraf zapisywany jest do danego pliku bez okna zapisu. |

Definicja w linii 946 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.40 `def scale_value_changed (self, event)`

Metoda pozwalająca reagować na zmiany położenia suwaka historii ewolucji hipergrafu.

Definicja w linii 1287 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.41 `def select_toggle_colliding_xnode (self, xid)`

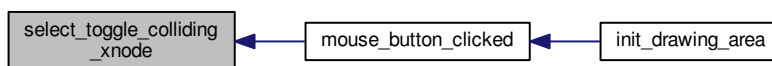
Metoda pozwalająca zaznaczyć lub odznaczyć kliknięty element hipergrafu.

Parametry

| | |
|------------|-------------------------|
| <i>xid</i> | Id elementu hipergrafu. |
|------------|-------------------------|

Definicja w linii 1144 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.42 `def select_xnode (self, xid)`

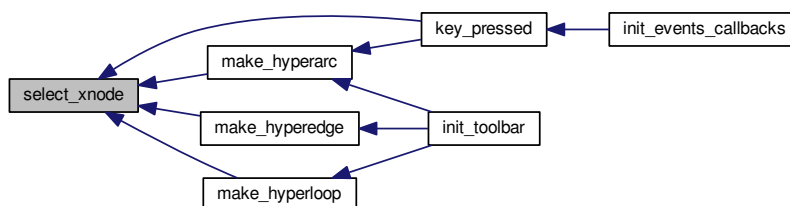
Metoda pozwalająca zaznaczyć element hipergrafu po jego id.

Parametry

| | |
|------------|-------------------------|
| <i>xid</i> | Id elementu hipergrafu. |
|------------|-------------------------|

Definicja w linii 1358 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:



6.3.3.43 `def set_active_hypergraph (self, hg)`

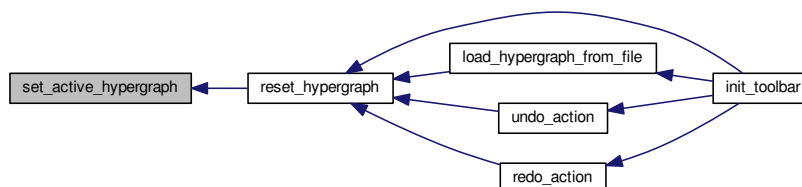
Metoda ustawiająca dany hipergraf jako aktywny.

Parametry

| | |
|-----------|--|
| <i>hg</i> | Obiekt hipergrafu, który ma zostać ustawiony jako aktywny. |
|-----------|--|

Definicja w linii 936 pliku `GuiWindow.py`.

Oto graf wywoływań tej funkcji:



6.3.3.44 `def show_matrices (self, button = None)`

Metoda wyświetlająca okno z listą macierzy hipergrafu.

Definicja w linii 1642 pliku `GuiWindow.py`.

Oto graf wywoływań tej funkcji:



6.3.3.45 `def toggle_animate (self, button = None)`

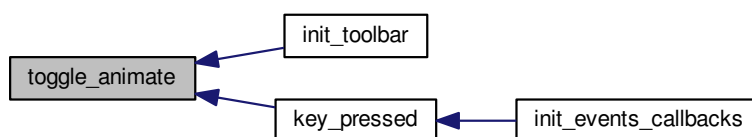
Metoda pozwalająca włączyć lub wyłączyć animację elementów hipergrafu.

Parametry

| | |
|---------------------|---|
| <code>button</code> | Obiekt przycisku, z którego ewentualnie wywoływana jest metoda. |
|---------------------|---|

Definicja w linii 1297 pliku GuiWindow.py.

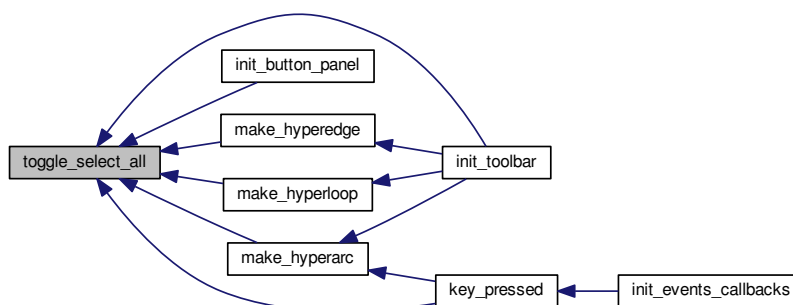
Oto graf wywoływań tej funkcji:

6.3.3.46 `def toggle_select_all (self, button = None)`

Metoda pozwalająca zaznaczyć lub odznaczyć element hipergrafu po jego id.

Definicja w linii 1333 pliku GuiWindow.py.

Oto graf wywoływań tej funkcji:

6.3.3.47 `def undo_action (self, button = None)`

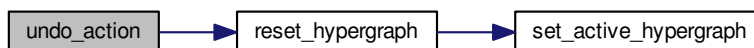
Metoda pozwalająca cofnąć cofniętą wykonaną akcję.

Parametry

| | |
|---------------|---|
| <i>button</i> | Obiekt przycisku, z którego ewentualnie wywoływana jest metoda. |
|---------------|---|

Definicja w linii 1302 pliku GuiWindow.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.3.4 Dokumentacja atrybutów składowych

6.3.4.1 active_hg

Obiekt aktualnie używanego hipergrafu.

Definicja w linii 480 pliku GuiWindow.py.

6.3.4.2 ANIMATE

Zmienna ustalająca czy hipergraf ma być animowany.

Definicja w linii 464 pliku GuiWindow.py.

6.3.4.3 CANVAS_PAN_VECTOR

Wektor przesunięcia punktu środka płaszczyzny rysowania elementów w stosunku do lewego górnego rogu ekranu.

Wyrażony w bazie płaszczyzny rysowania elementów (nie ekranu!)

Definicja w linii 460 pliku GuiWindow.py.

6.3.4.4 custom_clickables

Lista dodatkowych elementów zdefiniowanych przez użytkownika, które mają reagować na kliknięcia.

Definicja w linii 486 pliku GuiWindow.py.

6.3.4.5 custom_drawables

Lista dodatkowych elementów zdefiniowanych przez użytkownika, które mają być wyświetlone na płótnie.

Definicja w linii 483 pliku GuiWindow.py.

6.3.4.6 DEBUG_MODE

Zmienna ustalająca tryb testowania.

Definicja w linii 446 pliku GuiWindow.py.

6.3.4.7 dialog

Definicja w linii 1820 pliku GuiWindow.py.

6.3.4.8 DRAWING_AREA_CENTER

Zmienna przechowująca aktualny punkt środka okna rysowania jako tablica NumPy.

Jest wyrażony w układzie współrzędnych ekranu.

Definicja w linii 443 pliku GuiWindow.py.

6.3.4.9 DRAWING_AREA_SIZE

Zmienna przechowująca aktualną wielkość okna rysowania jako tablica NumPy.

Jest wyrażony w układzie współrzędnych ekranu.

Definicja w linii 439 pliku GuiWindow.py.

6.3.4.10 frame

Ramka na pole do rysowania.

Definicja w linii 857 pliku GuiWindow.py.

6.3.4.11 MAXZOOM

Maksymalna wartość powiększenia płótna.

Definicja w linii 456 pliku GuiWindow.py.

6.3.4.12 MINZOOM

Minimalna wartość powiększenia płótna.

Definicja w linii 453 pliku GuiWindow.py.

6.3.4.13 old_mouse_button

Zmienna przechowująca ostatnio używany przycisk myszy.

Definicja w linii 473 pliku GuiWindow.py.

6.3.4.14 old_mouse_pos

Zmienna przechowująca poprzednią pozycję myszy przy kliknięciu.

Definicja w linii 470 pliku GuiWindow.py.

6.3.4.15 redo_states

Zmienna przechowująca stany chronologicznie nowsze, niż aktualnie cofnięty.

Definicja w linii 492 pliku GuiWindow.py.

6.3.4.16 START_TIME_SEC

Zmienna czasu startu aplikacji.

Definicja w linii 476 pliku GuiWindow.py.

6.3.4.17 surface

Powierzchnia, na której Cairo będzie rysował.

Definicja w linii 902 pliku GuiWindow.py.

6.3.4.18 TIME_DELTA_SEC

Stała czasowa, która wyraża ilość czasu jaki musi minąć pomiędzy dwoma kolejnymi odświeżeniami widoku.

Definicja w linii 467 pliku GuiWindow.py.

6.3.4.19 undo_states

Zmienna przechowująca poprzednie stany hipergrafu, aby móc przywrócić stan przy cofnięciu akcji.

Definicja w linii 489 pliku GuiWindow.py.

6.3.4.20 ZOOM

Współczynnik powiększenia płaszczyzny rysowania elementów w stosunku do bazy współrzędnych ekranu.

Im większy tym większe rysowane są elementy na ekranie.

Definicja w linii 450 pliku GuiWindow.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/GuiWindow.py](#)

6.4 Dokumentacja klasy TextResultDialog

Klasa [TextResultDialog](#) Pozwala na wyświetlanie wyników w formie tekstu.

Diagram dziedziczenia dla TextResultDialog

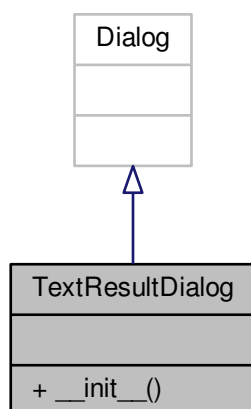
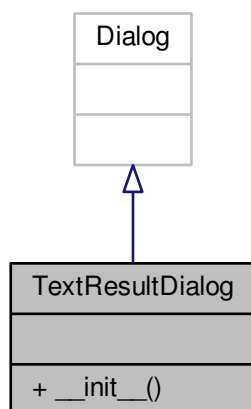


Diagram współpracy dla `TextResultDialog`:



Metody publiczne

- `def __init__(self, parent, dialog_name="Result", result_text="result text")`
Konstruktor.

6.4.1 Opis szczegółowy

Klasa `TextResultDialog` Pozwala na wyświetlanie wyników w formie tekstu.

Definicja w linii 192 pliku `GuiWindow.py`.

6.4.2 Dokumentacja konstruktora i destruktora

6.4.2.1 `def __init__(self, parent, dialog_name = "Result", result_text = "result text")`

Konstruktor.

Definicja w linii 194 pliku `GuiWindow.py`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `app-python3-gtk3-cairo/GuiWindow.py`

6.5 Dokumentacja klasy `TreeViewResultDialog`

Klasa `TreeViewResultDialog` Pozwala na wyświetlanie wyników w formie listy wierszy.

Diagram dziedziczenia dla `TreeViewResultDialog`

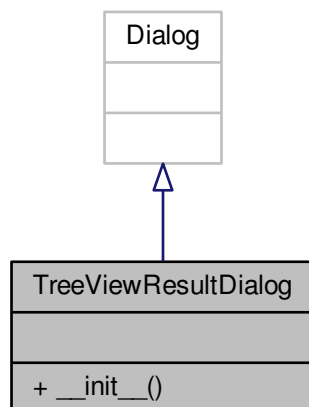
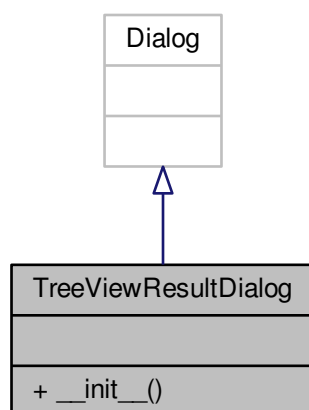


Diagram współpracy dla `TreeViewResultDialog`:



Metody publiczne

- `def __init__(self, parent, dialog_name="Result", result_matrix_dict=None)`
Konstruktor.

6.5.1 Opis szczegółowy

Klasa [TreeViewResultDialog](#) Pozwala na wyświetlanie wyników w formie listy wierszy.

Definicja w linii 128 pliku GuiWindow.py.

6.5.2 Dokumentacja konstruktora i destruktora

6.5.2.1 `def __init__(self, parent, dialog_name = "Result", result_matrix_dict = None)`

Konstruktor.

Definicja w linii 130 pliku GuiWindow.py.

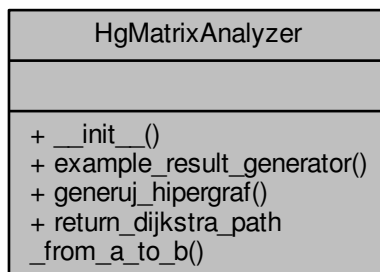
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/GuiWindow.py](#)

6.6 Dokumentacja klasy HgMatrixAnalyzer

Klasa [HgMatrixAnalyzer](#).

Diagram współpracy dla HgMatrixAnalyzer:



Metody publiczne

- `def __init__(self)`

Statyczne metody publiczne

- `def example_result_generator(hg_obj)`
Metoda generująca przykładowy wynik.
- `def generuj_hipergraf`
Metoda generująca hipergraf z danych wprowadzonych przez użytkownika.
- `def return_dijkstra_path_from_a_to_b`
Metoda wykonująca algorytm Dijkstry wyszukiwania najkrótszej drogi między dwoma punktami.

6.6.1 Opis szczegółowy

Klasa [HgMatrixAnalyzer](#).

Klasa odpowiadająca za analizę danego hipergrafu. Jej metodami są m.in. algorytmy.

Definicja w linii 18 pliku HgMatrixAnalyzer.py.

6.6.2 Dokumentacja konstruktora i destruktora

6.6.2.1 `def __init__(self)`

Definicja w linii 19 pliku HgMatrixAnalyzer.py.

6.6.3 Dokumentacja funkcji składowych

6.6.3.1 `def example_result_generator (hg_obj) [static]`

Metoda generująca przykładowy wynik.

Parametry

| | |
|---------------|---|
| <i>hg_obj</i> | Obiekt hipergrafu na którym wykonywany jest algorytm. |
|---------------|---|

Definicja w linii 25 pliku HgMatrixAnalyzer.py.

6.6.3.2 `def generuj_hipergraf (hgobj) [static]`

Metoda generująca hipergraf z danych wprowadzonych przez użytkownika.

Parametry

| | |
|--------------|---|
| <i>hgobj</i> | Obiekt hipergrafu na którym wykonywany jest algorytm. |
| <i>nn</i> | Ilość wierzchołków do wygenerowania. |
| <i>nh</i> | Ilość hipergałęzi do wygenerowania. |
| <i>evo</i> | Zmienna sterująca, czy kolejne dodania elementów do hipergrafu mają być traktowane jako kolejny krok jego ewolucji. |

Definicja w linii 222 pliku HgMatrixAnalyzer.py.

6.6.3.3 `def return_dijkstra_path_from_a_to_b (hgobj) [static]`

Metoda wykonująca algorytm Dijkstry wyszukiwania najkrótszej drogi między dwoma punktami.

Parametry

| | |
|--------------|---|
| <i>hgobj</i> | Obiekt hipergrafu na którym wykonywany jest algorytm. |
|--------------|---|

Parametry

| | |
|------------------------|------------------------------|
| $nid \leftarrow$ _a | Id wierzchołka początkowego. |
| $nid \leftarrow$ _b | Id wierzchołka końcowego. |

Definicja w linii 71 pliku HgMatrixAnalyzer.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/HgMatrixAnalyzer.py](#)

6.7 Dokumentacja klasy HyperGraph

Klasa [HyperGraph](#).

Diagram dziedziczenia dla HyperGraph

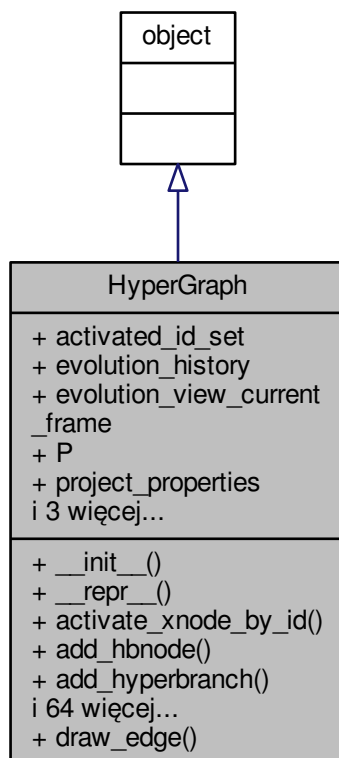
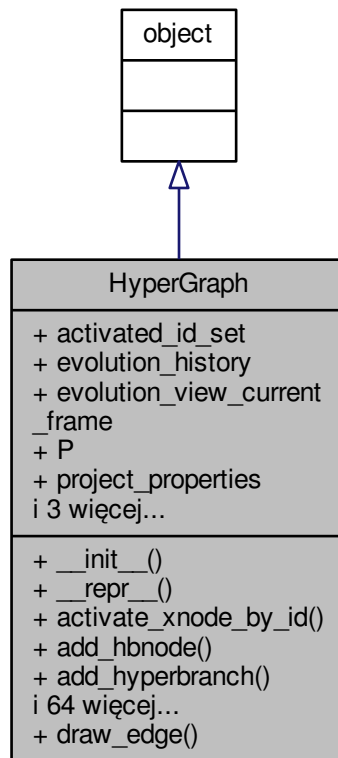


Diagram współpracy dla HyperGraph:



Metody publiczne

- `def __init__(self)`
Konstruktor obiektu hipergrafu.
- `def __repr__(self)`
Metoda zwracająca wartość tekstową hipergrafu.
- `def activate_xnode_by_id(self, xid)`
Metoda pozwalająca na wyróżnienie elementu hipergrafu po jego id.
- `def add_hbnode(self, name, hbtype, position, prop_dict, normalize_pos=True)`
Metoda tworząca nowy obiekt reprezentujący wizualnie hipergałąź.
- `def add_hyperbranch(self, name, hbtype, nodes_id_list, prop_dict, normalize_pos=True)`
Metoda tworząca nową hipergałąź.
- `def add_node(self, name, position, prop_dict, normalize_pos=True)`
Metoda tworząca nowy obiekt reprezentujący wizualnie wierzchołek.
- `def bin_macierz_incydencji(self, xid_list=None)`
Metoda zwracająca binarną macierz incydencji Ab.
- `def bin_macierz_przejsc(self, nodes_id=None)`
Metoda zwracająca binarną macierz przejść Pb.
- `def bin_macierz_przyleglosci_galezi(self, hyperbranches_id=None)`
Metoda zwracająca binarną macierz przyległości gałęzi Bb.

- def `bin_macierz_przyleglosci_wierzchołkow` (self, nodes_id=None)
Metoda zwracająca binarną macierz przyległości wierzchołków Rb.
- def `deactivate_all` (self)
Metoda usuwająca wyróżnienie ze wszystkich elementów.
- def `deactivate_xnode_by_id` (self, xid)
Metoda pozwalająca na usunięcia wyróżnienia z elementu hipergrafu po jego id.
- def `delete_hyperbranch_by_id` (self, hid)
Metoda usuwająca hipergałęź po id.
- def `delete_node_by_id` (self, nid)
Metoda usuwająca wierzchołek po id.
- def `delete_xnode_by_id` (self, xid)
Metoda usuwająca wierzchołek lub hipergałęź po id.
- def `deselect_xnode_by_id` (self, xid)
Metoda pozwalająca na odznaczenie elementu hipergrafu po jego id.
- def `documenting_dummy_fun` (self)
- def `draw` (self, cro, pan_vec, center, zoom)
Metoda rysująca hipergraf na płótnie rysowania.
- def `dump_hg_as_dict` (self, elems_to_dump=None)
Metoda zwracająca hipergraf jako słownik wybranych elementów.
- def `dump_hg_as_tuple` (self)
Metoda zwracająca hipergraf jako krotkę jego elementów.
- def `get_activated_xnodes_id` (self)
Metoda zwracająca zbiór wyróżnionych elementów hipergrafu.
- def `get_all_connected_nodes_id` (self)
Metoda generująca zbiór elementów elementów, które należą do przynajmniej jednej hipergałęzi.
- def `get_all_connected_nodes_id_by_node_id` (self, nid)
Metoda zwracająca wszystkie id wierzchołków połączonych do danego wierzchołka.
- def `get_all_connected_xnodes_id_by_xnode_id` (self, xid)
Metoda zwracająca wszystkie id elementów połączonych do danego elementu.
- def `get_all_hyperbranches_id` (self)
Metoda zwracająca id wszystkich hipergałęzi hipergrafu.
- def `get_all_hyperbranches_id_by_node_id` (self, nid, hbtype=None)
Metoda zwracająca wszystkie id hipergałęzi do których należy dany wierzchołek.
- def `get_all_nodes_id` (self)
Metoda zwracająca id wszystkich wierzchołków hipergrafu.
- def `get_all_nodes_id_by_hyperbranch_id` (self, hid)
Metoda zwracająca wszystkie id wierzchołków należących do danej hipergałęzi.
- def `get_all_nodes_id_where_node_id_is_reachable` (self, nid)
Metoda zwracająca wszystkie id wierzchołków z których da się przejść do danego wierzchołka.
- def `get_all_reachable_nodes_id_from_node_id` (self, nid)
Metoda zwracająca wszystkie id wierzchołków do których da się przejść z danego wierzchołka.
- def `get_all_shared_nodes_id_in_hyperbranches_list` (self, hbid_list)
Metoda zwracająca zbiór wierzchołków wspólny dla danych hipergałęzi.
- def `get_all_unconnected_nodes_id` (self)
Metoda generująca zbiór elementów elementów, które nie należą do żadnej hipergałęzi.
- def `get_all_xnodes_id` (self)
Metoda zwracająca id wszystkich elementów hipergrafu.
- def `get_closest_xnode_id_by_xnode_id` (self, xid, xid_list=None)
Metoda zwracająca id najbliższego elementu hipergrafu do id danego elementu.
- def `get_colliding_xnode_id_by_position` (self, pos, rmul=4.0)
Metoda zwracająca id elementu który jest na danej pozycji.

- def [get_connections_between_nodes_by_nodes_id](#) (self, nid1, nid2, directed=True)
Metoda zwracająca zbiór id hipergałęzi łączących daną parę wierzchołków.
- def [get_hbnode_by_id](#) (self, hid)
Metoda zwracająca obiekt reprezentujący wizualnie hipergałąź po id hipergałęzi.
- def [get_hyperbranch_by_id](#) (self, hid)
Metoda zwracająca zestaw (listę, krotkę, zbiór) id wierzchołków w danej hipergałęzi.
- def [get_max_xnode_index](#) (self)
Metoda zwracająca maksymalny id elementu hipergrafu.
- def [get_node_by_id](#) (self, nid)
Metoda zwracająca obiekt wizualnie reprezentujący wierzchołek po jego id.
- def [get_node_degree_by_node_id](#) (self, nid)
Metoda zwracająca stopień wierzchołka (liczbę hiperkrawędzi do niego połączonych).
- def [get_selected_hbnodes_id](#) (self)
Metoda zwracająca listę zaznaczonych hipergałęzi hipergrafu.
- def [get_selected_nodes_id](#) (self)
Metoda zwracająca listę zaznaczonych wierzchołków hipergrafu.
- def [get_selected_xnodes_id](#) (self)
Metoda zwracająca listę zaznaczonych elementów hipergrafu.
- def [get_xnode_by_id](#) (self, xid)
Metoda zwracająca obiekt wizualnie reprezentujący element hipergrafu po jego id.
- def [get_xnode_degree_by_xnode_id](#) (self, xid, hbtype=None)
Metoda zwracająca stopień elementu jako liczbę elementów z nim połączonych.
- def [get_xnodes_center_of_mass_by_xnodes_id_list](#) (self, xid_list)
Metoda obliczająca środek masy danych elementów.
- def [load_evolutionary_state](#) (self, frame)
Ładuje dany stan ewolucji hipergrafu do podglądu.
- def [load_hg_from_dict](#) (self, hgdict)
Metoda wypełniająca hipergraf za pomocą słownika wybranych elementów.
- def [load_hg_from_tuple](#) (self, hgtuple)
Metoda wypełniająca hipergraf za pomocą krotki jego elementów.
- def [load_next_evolutionary_state](#) (self)
Ładuje następny stan ewolucji hipergrafu do podglądu.
- def [load_previous_evolutionary_state](#) (self)
Ładuje poprzedni stan ewolucji hipergrafu do podglądu.
- def [macierz_incydencji](#) (self, xid_list=None)
Metoda zwracająca macierz incydencji A w formie numerycznej.
- def [macierz_incydencji_u21](#) (self, xid_list=None)
Metoda zwracająca macierz incydencji A w formie stringa.
- def [macierz_osiagalnosci](#) (self, nodes_id=None)
Metoda zwracająca macierz osiągalności D.
- def [macierz_przejsc](#) (self, nodes_id=None)
Metoda zwracająca macierz przejść P.
- def [macierz_przyleglosci_galezi](#) (self, hyperbranches_id=None)
Metoda zwracająca macierz przyległości gałęzi B.
- def [macierz_przyleglosci_wierzchoolkow](#) (self, nodes_id=None)
Metoda zwracająca macierz przyległości wierzchołków R.
- def [macierz_skalarow_odleglosci](#) (self, xid_list=None, cl=False)
Metoda zwracająca macierz skalarów odległości pomiędzy elementami hipergrafu.
- def [macierz_spojnosci](#) (self, nodes_id=None)
Metoda zwracająca macierz spójności S.
- def [macierz_wekt_kierunkowych](#) (self, xid_list=None, cl=False)

- Metoda zwracająca macierz wektorów kierunku pomiędzy elementami hipergrafu.
- def `macierz_wekt_odleglosci` (self, xid_list=None, cl=False)
 - Metoda zwracająca macierz wektorów odległości pomiędzy elementami hipergrafu.
- def `normalize_xnode_position` (self, xnode, rmul=1.5)
 - Metoda, która generuje nową pozycję elementu hipergrafu jeśli aktualna pozycja jest zajęta przez inny element.
- def `save_evolutionary_state` (self, incrementcounter=True)
 - Zapisuje aktualny stan hipergrafu jako stan jego ewolucji.
- def `select_toggle_xnode_by_id` (self, xid)
 - Metoda pozwalająca na zaznaczenie lub odznaczenie elementu hipergrafu po jego id.
- def `select_xnode_by_id` (self, xid)
 - Metoda pozwalająca na zaznaczenie elementu hipergrafu po jego id.
- def `update` (self, dt)
 - Metoda odświeżająca hipergraf.
- def `update_all_xnodes` (self, dt)
 - Metoda odświeżająca wszystkie wierzchołki i hipergałęzie hipergrafu.
- def `update_xnode_by_id` (self, xid)
 - Metoda odświeżająca dany wierzchołek lub hipergałęz hipergrafu.

Statyczne metody publiczne

- def `draw_edge` (v1pos_zoomed, v2pos_zoomed, cro, zoom, color=None)
 - Metoda rysująca połączenie pomiędzy wierzchołkiem a hiperkrawędzią.

Atrybuty publiczne

- `activated_id_set`
 - Zmienna zbioru nieuporządkowanego przechowująca identyfikatory wyróżnionych elementów hipergrafu.
- `evolution_history`
 - Zmienna przechowująca historię kroków ewolucji hipergrafu.
- `evolution_view_current_frame`
 - Zmienna informująca o numerze aktualnie przeglądane kroku ewolucji hipergrafu.
- `P`
 - Zmienna słownikowa przechowująca relacje wierzchołków do hipergałęzi w hipergrafie.
- `project_properties`
 - Zmienna zawierająca ustawienia aktualnego projektu.
- `selected_id_list`
 - Zmienna typu lista przechowująca w odpowiedniej kolejności id klikniętych wierzchołków.
- `U`
 - Zmienna słownikowa przechowująca hipergałęzie hipergrafu.
- `X`
 - Zmienna słownikowa przechowująca wierzchołki hipergrafu.

6.7.1 Opis szczegółowy

Klasa `HyperGraph`.

Główna klasa projektu. Tu znajdują się wszystkie najważniejsze elementy. Zawiera zmienne słownikowe X, U, P oraz zmienną zbioru `activated_id_set`. Są to główne elementy modelu hipergrafu. Definiują one postać hipergrafu. Dodatkowo zawiera zmienne projektu `project_properties`, `selected_id_list`. Model hipergrafu może ewoluować, na przykład na podstawie jakiegoś algorytmu.

Definicja w linii 31 pliku `HyperGraph.py`.

6.7.2 Dokumentacja konstruktora i destruktora

6.7.2.1 `def __init__(self)`

Konstruktor obiektu hipergrafu.

Definicja w linii 36 pliku HyperGraph.py.

6.7.3 Dokumentacja funkcji składowych

6.7.3.1 `def __repr__(self)`

Metoda zwracająca wartość tekstową hipergrafu.

Zwraca

Wartość tekstowa hipergrafu zawierająca słowniki X, U, P oraz listę zaznaczonych elementów.

Definicja w linii 869 pliku HyperGraph.py.

6.7.3.2 `def activate_xnode_by_id(self, xid)`

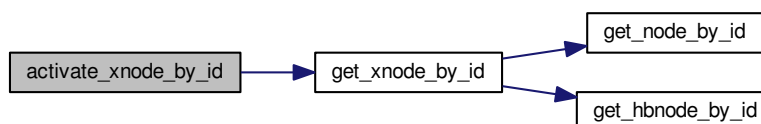
Metoda pozwalająca na wyróżnienie elementu hipergrafu po jego id.

Parametry

| | |
|------------|-----------------------------|
| <i>xid</i> | Id elementu do wyróżnienia. |
|------------|-----------------------------|

Definicja w linii 708 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:

6.7.3.3 `def add_hbnode(self, name, hbtype, position, prop_dict, normalize_pos = True)`

Metoda tworząca nowy obiekt reprezentujący wizualnie hipergałąź.

Parametry

| | |
|-------------|--------------------|
| <i>name</i> | Nazwa hipergałęzi. |
|-------------|--------------------|

Parametry

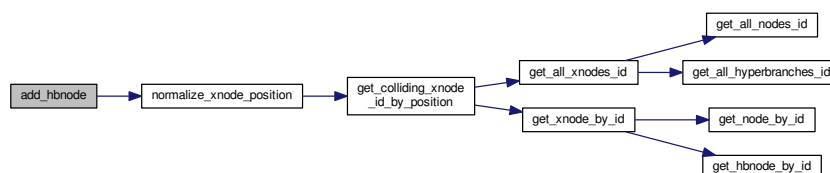
| | |
|------------------|---|
| <i>type</i> | Typ hipergałęzi (hiperkrawędź, hiperłuk, hiperpętla). |
| <i>position</i> | Współrzędne utworzenia obiektu. |
| <i>prop_dict</i> | Słownik właściwości do przypisania hipergałęzi. |

Zwraca

Id nowoutworzonego obiektu.

Definicja w linii 105 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



```
6.7.3.4 def add_hyperbranch ( self, name, hbtype, nodes_id_list, prop_dict, normalize_pos = True )
```

Metoda tworząca nową hipergałąź.

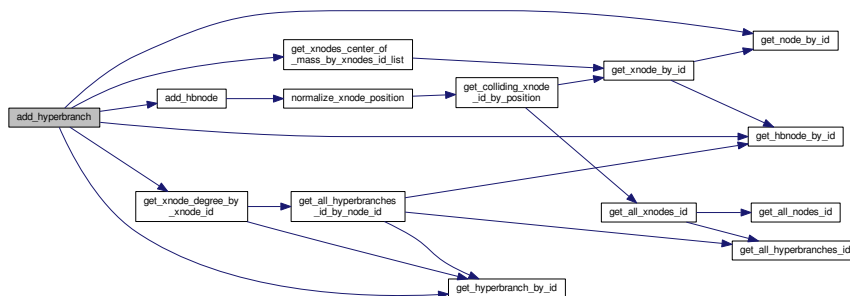
Obejmuje to stworzenie nowej relacji wierzchołek-hipergałąź oraz stworzenie nowego obiektu reprezentującego wizualnie hipergałąź.

Parametry

| | |
|----------------------|---|
| <i>name</i> | Nazwa hipergałęzi. |
| <i>type</i> | Typ hipergałęzi (hiperkrawędź, hiperłuk, hiperpętla). |
| <i>nodes_id_list</i> | Lista id wierzchołków do przypisania hiperkrawędzi. |
| <i>prop_dict</i> | Słownik właściwości do przypisania hipergałęzi. |

Definicja w linii 125 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.5 `def add_node(self, name, position, prop_dict, normalize_pos = True)`

Metoda tworząca nowy obiekt reprezentujący wizualnie wierzchołek.

Parametry

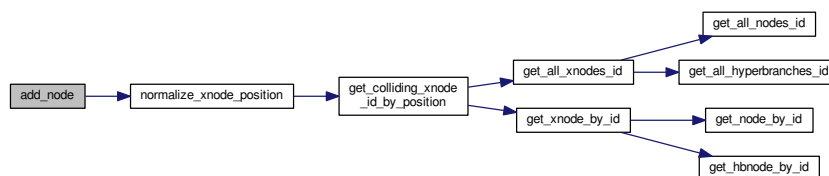
| | |
|------------------|---|
| <i>name</i> | Nazwa wierzchołka. |
| <i>position</i> | Współrzędne utworzenia obiektu. |
| <i>prop_dict</i> | Słownik właściwości do przypisania wierzchołkowi. |

Zwraca

Id nowoutworzonego obiektu.

Definicja w linii 82 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.6 `def bin_macierz_incydencji(self, xid_list = None)`

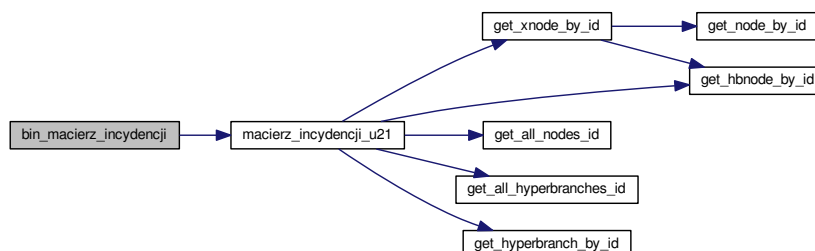
Metoda zwracająca binarną macierz incydencji Ab.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1104 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.7 def bin_macierz_przejsc (self, nodes_id=None)

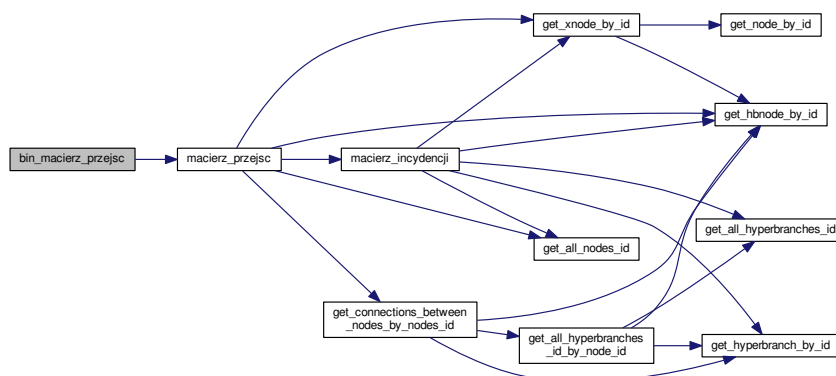
Metoda zwracająca binarną macierz przejść Pb.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1269 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.8 `def bin_macierz_przyleglosci_galezi (self, hyperbranches_id = None)`

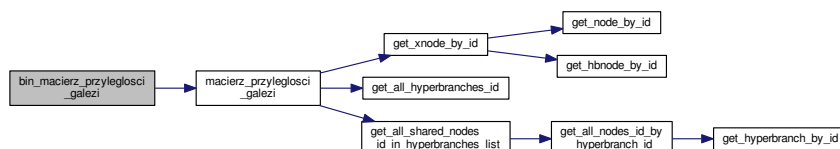
Metoda zwracająca binarną macierz przyległości gałęzi Bb.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1188 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.9 `def bin_macierz_przyleglosci_wierzchoolkow (self, nodes_id = None)`

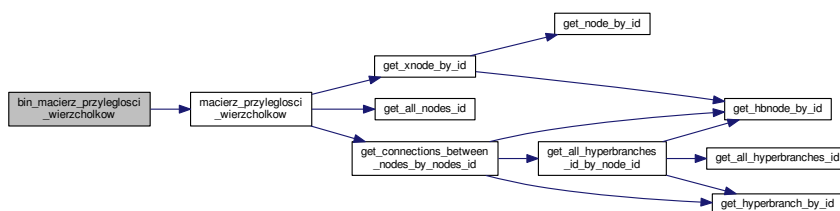
Metoda zwracająca binarną macierz przyległości wierzchołków Rb.

Zwraca

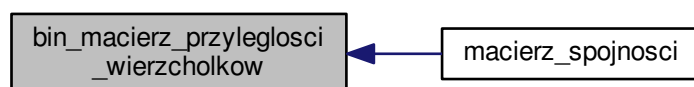
Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1146 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

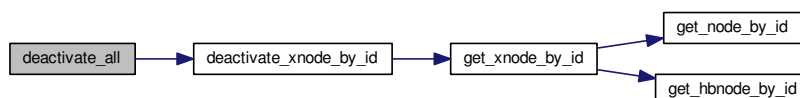


6.7.3.10 def deactivate_all (self)

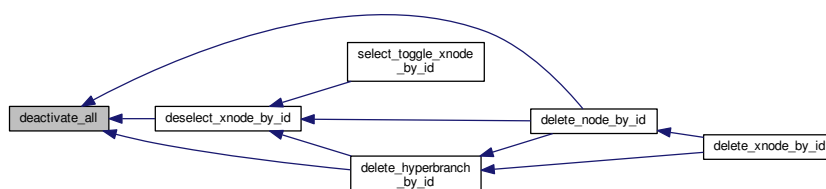
Metoda usuwająca wyróżnienie ze wszystkich elementów.

Definicja w linii 719 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.11 def deactivate_xnode_by_id (self, xid)

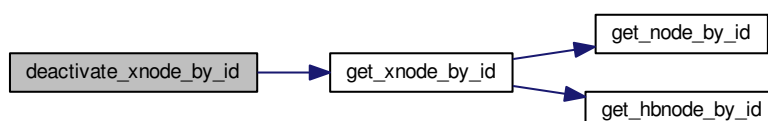
Metoda pozwalająca na usunięcia wyróżnienia z elementu hipergrafu po jego id.

Parametry

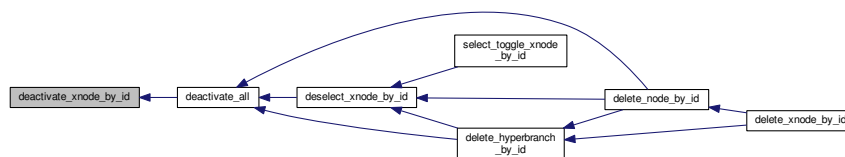
| | |
|------------|---------------------------------------|
| <i>xid</i> | Id elementu do usunięcia wyróżnienia. |
|------------|---------------------------------------|

Definicja w linii 714 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.7.3.12 def delete_hyperbranch_by_id (self, hid)

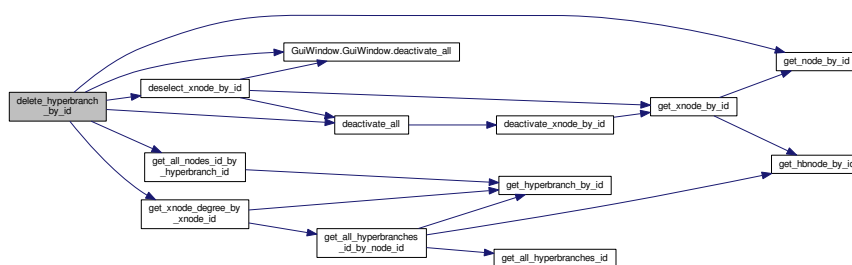
Metoda usuwająca hipergałąż po id.

Parametry

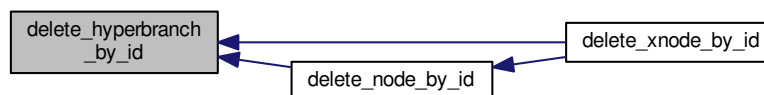
| | |
|------------|-----------------|
| <i>hid</i> | Id hipergałązi. |
|------------|-----------------|

Definicja w linii 853 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.7.3.13 def delete_node_by_id (self, nid)

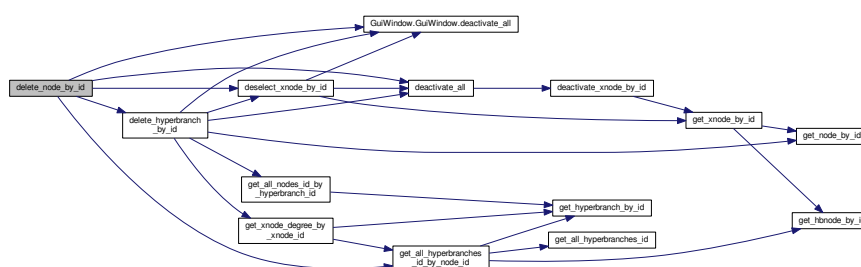
Metoda usuwająca wierzchołek po id.

Parametry

| | |
|------------|-----------------|
| <i>nid</i> | Id wierzchołka. |
|------------|-----------------|

Definicja w linii 841 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.14 def delete_xnode_by_id (self, xid)

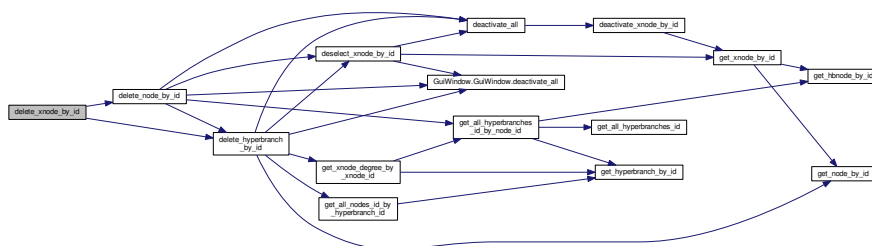
Metoda usuwająca wierzchołek lub hipergałąż po id.

Parametry

| | |
|------------|---------------------------------|
| <i>xid</i> | Id wierzchołka lub hipergałąży. |
|------------|---------------------------------|

Definicja w linii 832 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.15 def deselect_xnode_by_id (self, xid)

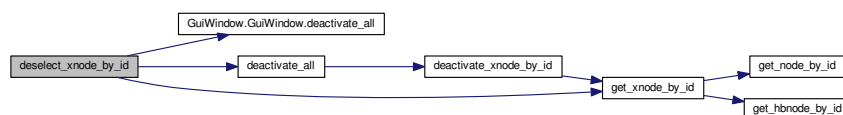
Metoda pozwalająca na odznaczenie elementu hipergrafu po jego id.

Parametry

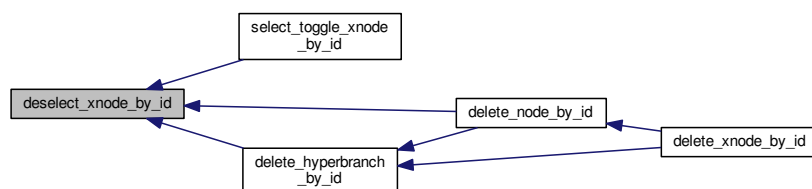
| | |
|------------|-----------------------------|
| <i>xid</i> | Id elementu do odznaczenia. |
|------------|-----------------------------|

Definicja w linii 696 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.16 def documenting_dummy_fun (self)

Definicja w linii 1496 pliku HyperGraph.py.

6.7.3.17 `def draw (self, cro, pan_vec, center, zoom)`

Metoda rysująca hipergraf na płótnie rysowania.

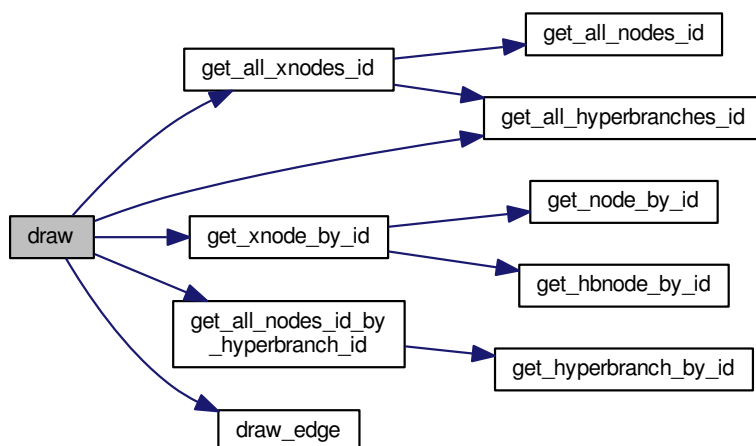
Zajmuje ok 15% czasu CPU. Rysuje wszystkie połączenia, wierzchołki i hipergałęzie a także dodatkowe informacje, na przykład osiągalność z ostatnio zaznaczonego wierzchołka lub pogrubienie połączenia, jeśli zaznaczony element do niego należy.

Parametry

| | |
|----------------|--|
| <i>cro</i> | Kontekst rysowania biblioteki Cairo. |
| <i>dt</i> | Stałą czasowa, odstęp pomiędzy rysowaniem. |
| <i>pan_vec</i> | Wektor przesunięcia punktu środka płaszczyzny rysowania elementów w stosunku do lewego górnego rogu ekranu. Wyrażony w bazie płaszczyzny rysowania elementów (nie ekranu!) |
| <i>center</i> | Punkt środka ekranu w bazie współrzędnych ekranu. |
| <i>zoom</i> | Współczynnik powiększenia płaszczyzny rysowania elementów w stosunku do bazy współrzędnych ekranu. Im większy tym większe rysowane są elementy na ekranie. |

Definicja w linii 583 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.18 `def draw_edge (v1pos_zoomed, v2pos_zoomed, cro, zoom, color=None) [static]`

Metoda rysująca połączenie pomiędzy wierzchołkiem a hiperkrawędzią.

Zajmuje ok 5% czasu CPU.

Parametry

| | |
|-------------|---|
| <i>pos1</i> | Współrzędna punktu początku połączenia w bazie współrzędnych płaszczyzny rysowania elementów. |
|-------------|---|

Parametry

| | |
|----------------|--|
| <i>pos2</i> | Współrzędna punktu końca połączenia w bazie współrzędnych płaszczyzny rysowania elementów. |
| <i>cro</i> | Kontekst rysowania biblioteki Cairo. |
| <i>pan_vec</i> | Wektor przesunięcia punktu środka płaszczyzny rysowania elementów w stosunku do lewego górnego rogu ekranu. |
| <i>center</i> | Punkt środka ekranu w bazie współrzędnych ekranu. |
| <i>zoom</i> | Współczynnik powiększenia płaszczyzny rysowania elementów w stosunku do bazy współrzędnych ekranu. Im większy tym większe rysowane są elementy na ekranie. |
| <i>color</i> | Kolor połączenia. |

Definicja w linii 559 pliku HyperGraph.py.

Oto graf wywoływań tej funkcji:



6.7.3.19 `def dump_hg_as_dict (self, elems_to_dump=None, dict)`

Metoda zwracająca hipergraf jako słownik wybranych elementów.

Parametry

| | |
|----------------------|---|
| <i>elems_to_dump</i> | Lista symboli elementów, które mają być zawarte w słowniku. |
|----------------------|---|

Zwraca

Słownik zawierający elementy hipergrafu.

Definicja w linii 895 pliku HyperGraph.py.

Oto graf wywoływań tej funkcji:



6.7.3.20 `def dump_hg_as_tuple (self)`

Metoda zwracająca hipergraf jako krotkę jego elementów.

Zwraca

Krotka zawierająca elementy hipergrafu.

Definicja w linii 874 pliku HyperGraph.py.

6.7.3.21 `def get_activated_xnodes_id (self)`

Metoda zwracająca zbiór wyróżnionych elementów hipergrafu.

Zwraca

Zbiór elementów wyróżnionych .

Definicja w linii 493 pliku HyperGraph.py.

6.7.3.22 `def get_all_connected_nodes_id (self)`

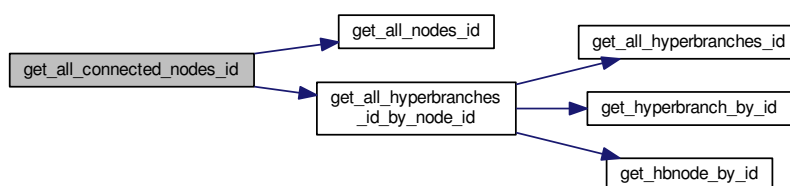
Metoda generująca zbiór elementów elementów, które należą do przynajmniej jednej hipergałęzi.

Zwraca

Zbiór elementów elementów, które należą do przynajmniej jednej hipergałęzi.

Definicja w linii 538 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.23 `def get_all_connected_nodes_id_by_node_id (self, nid)`

Metoda zwracająca wszystkie id wierzchołków połączonych do danego wierzchołka.

Nie jest tutaj ważny kierunek działania hipergałęzi wspólnych między danymi wierzchołkami, liczy się po prostu dowolne połączenie.

Parametry

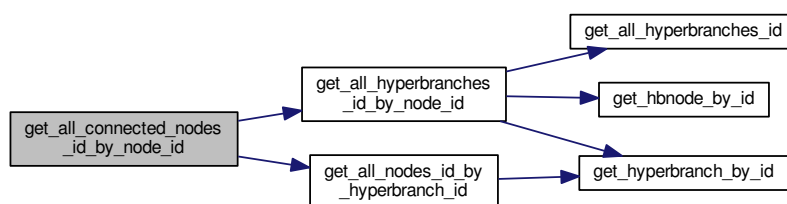
| | |
|------------|-----------------|
| <i>nid</i> | Id wierzchołka. |
|------------|-----------------|

Zwraca

Zbiór id wierzchołków, które są połączone z danym wierzchołkiem.

Definicja w linii 358 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:

6.7.3.24 `def get_all_connected_xnodes_id_by_xnode_id (self, xid)`

Metoda zwracająca wszystkie id elementów połączonych do danego elementu.

Nie jest istotny typ elementu, może to być wierzchołek lub hipergałąż. Nie jest tutaj ważny kierunek połączenia, liczy się po prostu dowolne połączenie.

Parametry

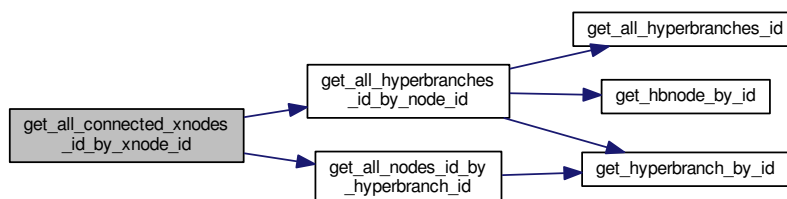
| | |
|------------|--------------|
| <i>xid</i> | Id elementu. |
|------------|--------------|

Zwraca

Zbiór id elementów, które są połączone z danym elementem w dowolny sposób

Definicja w linii 345 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.25 def get_all_hyperbranches_id (self)

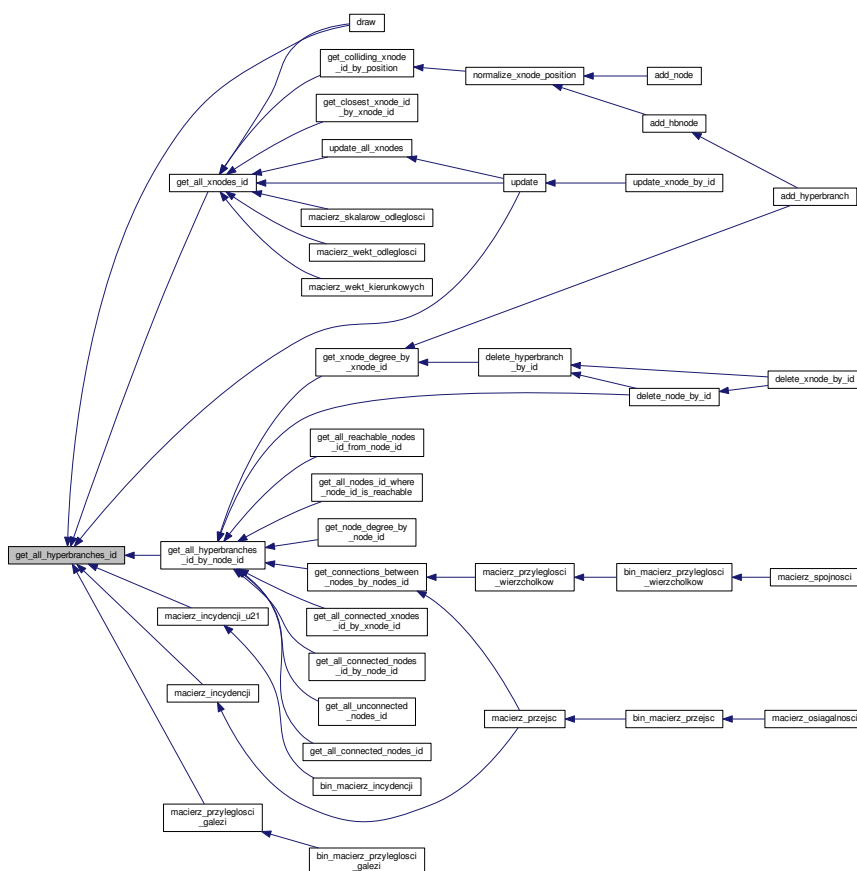
Metoda zwracająca id wszystkich hipergałęzi hipergrafu.

Zwraca

Lista id wszystkich hipergałęzi w hipergrafie.

Definicja w linii 337 pliku HyperGraph.py.

Oto graf wywoływań tej funkcji:



6.7.3.26 def get_all_hyperbranches_id_by_node_id (self, nid, hbtype=None)

Metoda zwracająca wszystkie id hipergałęzi do których należy dany wierzchołek.

Parametry

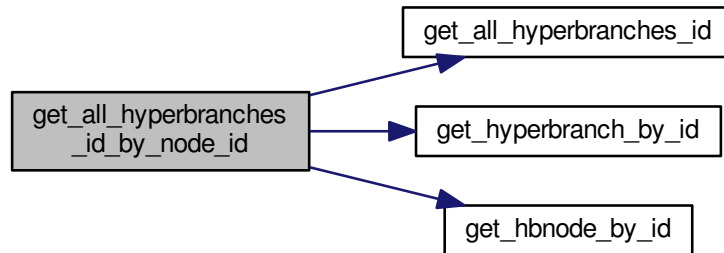
| | |
|---------------|---|
| <i>nid</i> | Id wierzchołka. |
| <i>hbtype</i> | Opcjonalny typ hipergałęzi, do filtrowania. |

Zwraca

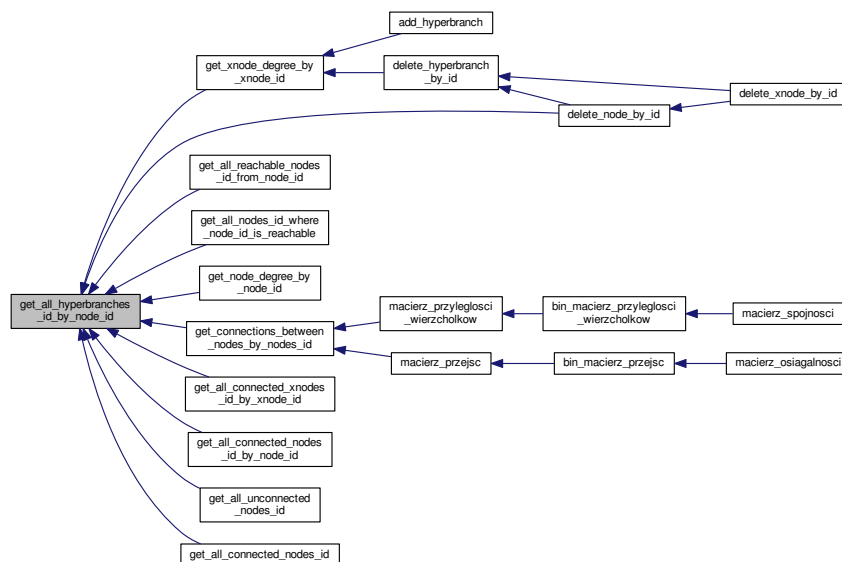
Lista id hipergałęzi, do których wierzchołek należy.

Definicja w linii 380 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**6.7.3.27 def get_all_nodes_id (self)**

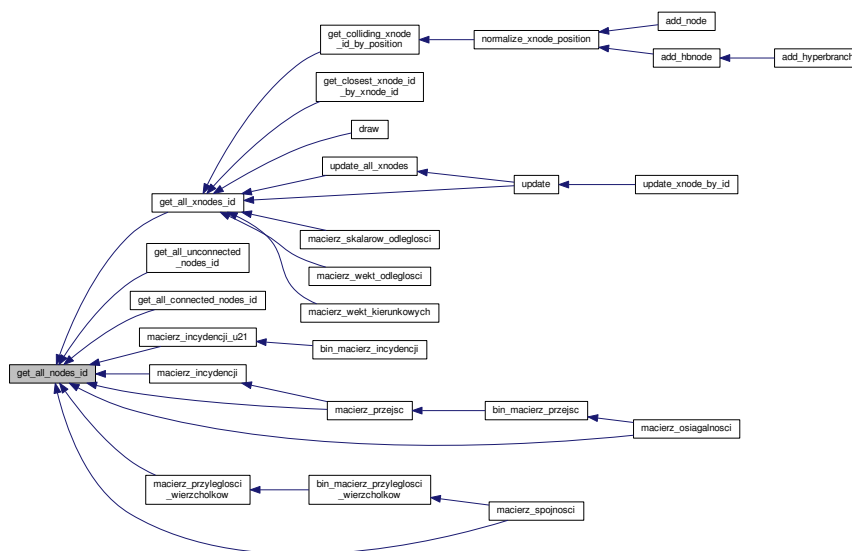
Metoda zwracająca id wszystkich wierzchołków hipergrafu.

Zwraca

Lista id wszystkich wierzchołków w hipergrafie.

Definicja w linii 332 pliku HyperGraph.py.

Oto graf wywołań tej funkcji:



6.7.3.28 `def get_all_nodes_id_by_hyperbranch_id (self, hid)`

Metoda zwracająca wszystkie id wierzchołków należących do danej hipergałęzi.

Parametry

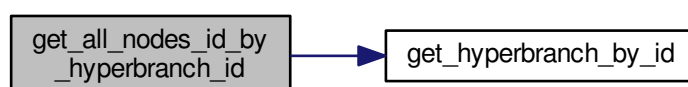
| | |
|------------|-----------------|
| <i>hid</i> | Id hipergałęzi. |
|------------|-----------------|

Zwraca

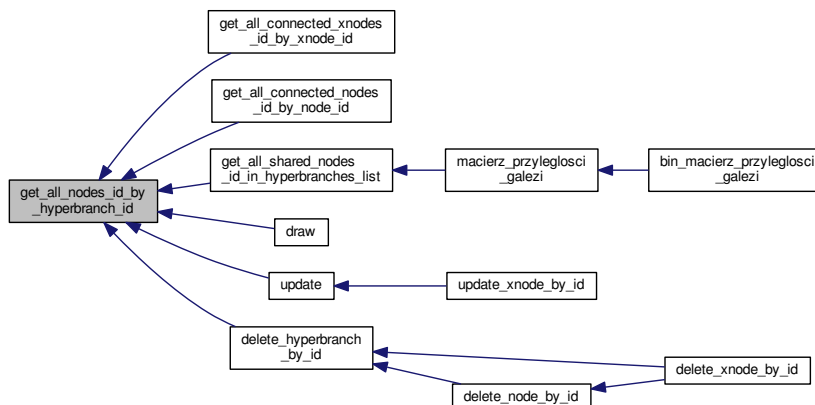
Lista id wierzchołków, które należą do danej hipergałęzi..

Definicja w linii 373 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.7.3.29 def get_all_nodes_id_where_node_id_is_reachable (self, nid)

Metoda zwracająca wszystkie id wierzchołków z których da się przejść do danego wierzchołka.

Parametry

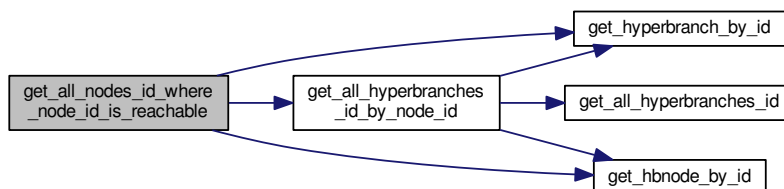
| | |
|------------|-----------------|
| <i>nid</i> | Id wierzchołka. |
|------------|-----------------|

Zwraca

Zbiór id wierzchołków z których da się przejść do danego wierzchołka.

Definicja w linii 235 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.30 def get_all_reachable_nodes_id_from_node_id (self, nid)

Metoda zwracająca wszystkie id wierzchołków do których da się przejść z danego wierzchołka.

Parametry

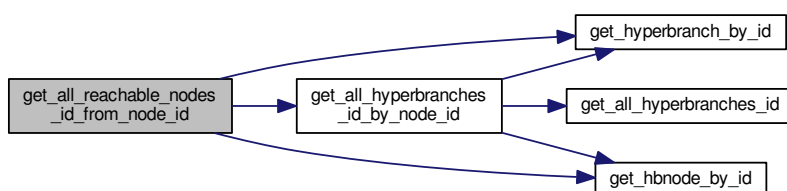
| | |
|------------|-----------------|
| <i>nid</i> | Id wierzchołka. |
|------------|-----------------|

Zwraca

Zbiór id wierzchołków do których da się przejść z danego wierzchołka.

Definicja w linii 205 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.331 `def get_all_shared_nodes_id_in_hyperbranches_list (self, hbid_list)`

Metoda zwracająca zbiór wierzchołków wspólny dla danych hipergałęzi.

Parametry

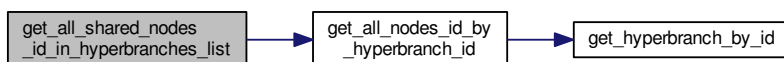
| | |
|------------------|--------------------------------------|
| <i>hbid_list</i> | Lista id hipergałęzi do sprawdzenia. |
|------------------|--------------------------------------|

Zwraca

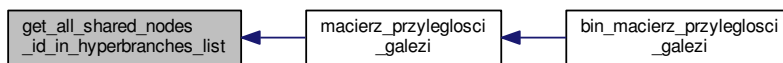
Zbiór id wierzchołków należących jednocześnie do wszystkich hipergałęzi z listy.

Definicja w linii 394 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.32 def get_all_unconnected_nodes_id (self)

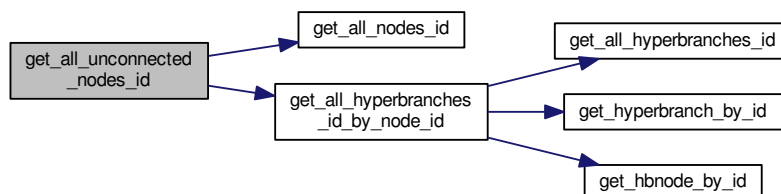
Metoda generująca zbiór elementów elementów, które nie należą do żadnej hipergałęzi.

Zwraca

Zbiór elementów elementów, które nie należą do żadnej hipergałęzi.

Definicja w linii 526 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.33 def get_all_xnodes_id (self)

Metoda zwracająca id wszystkich elementów hipergrafu.

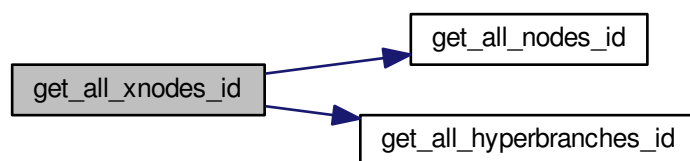
Nie jest istotny typ elementu, może to być wierzchołek lub hipergałąż.

Zwraca

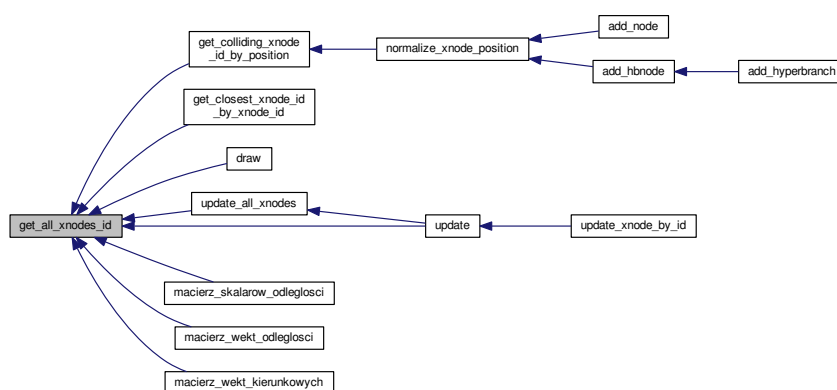
Lista id wszystkich elementów w hipergrafie.

Definicja w linii 327 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.34 `def get_closest_xnode_id_by_xnode_id (self, xid, xid_list=None)`

Metoda zwracająca id najbliższego elementu hipergrafu do id danego elementu.

Parametry

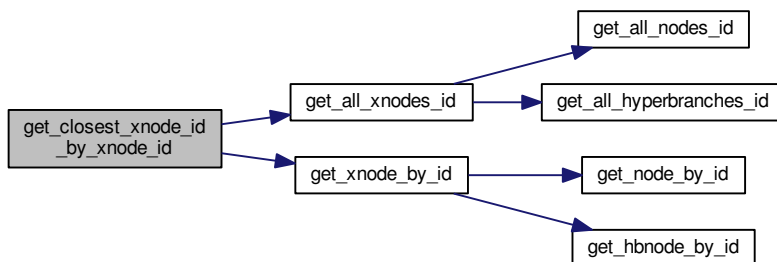
| | |
|-----------------|--|
| <i>xid</i> | Id elementu, do którego należy wyszukać najbliższy element. |
| <i>xid_list</i> | Opcjonalna lista id elementów, z których należy korzystać. Jeśli nie jest ustawiona, przeszukiwane są wszystkie wierzchołki. |

Zwraca

Id najbliższego elementu.

Definicja w linii 463 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.35 `def get_colliding_xnode_id_by_position (self, pos, rmul = 4 . 0)`

Metoda zwracająca Id elementu który jest na danej pozycji.

Parametry

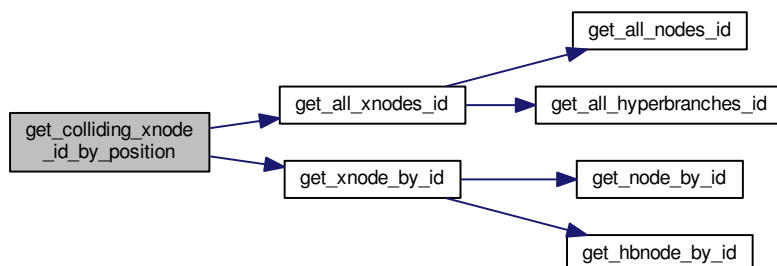
| | |
|-------------|---|
| <i>pos</i> | Pozycja, która ma zostać sprawdzona w poszukiwaniu elementów. |
| <i>rmul</i> | Mnożnik promienia, w którym należy szukać. |

Zwraca

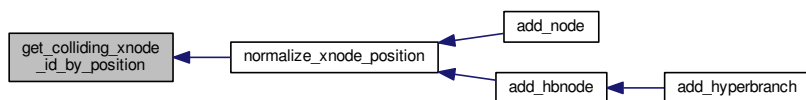
Id elementu znajdującego się na danej pozycji.

Definicja w linii 415 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.36 `def get_connections_between_nodes_by_nodes_id (self, nid1, nid2, directed = True)`

Metoda zwracająca zbiór id hipergałęzi łączących daną parę wierzchołków.

Parametry

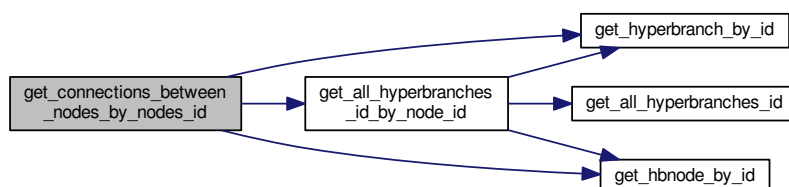
| | |
|-----------------|--|
| <i>nid1</i> | Id wierzchołka początkowego. |
| <i>nid2</i> | Id wierzchołka końcowego. |
| <i>directed</i> | Opcjonalny argument sterujący, czy kierunek ma być wzięty pod uwagę czy zignorowany. |

Zwraca

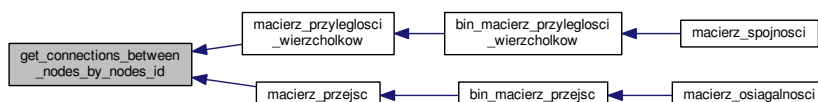
Zbiór id hipergałęzi łączących daną parę wierzchołków.

Definicja w linii 273 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.37 `def get_hbnode_by_id (self, hid)`

Metoda zwracająca obiekt reprezentujący wizualnie hipergałąź po id hipergałęzi.

Parametry

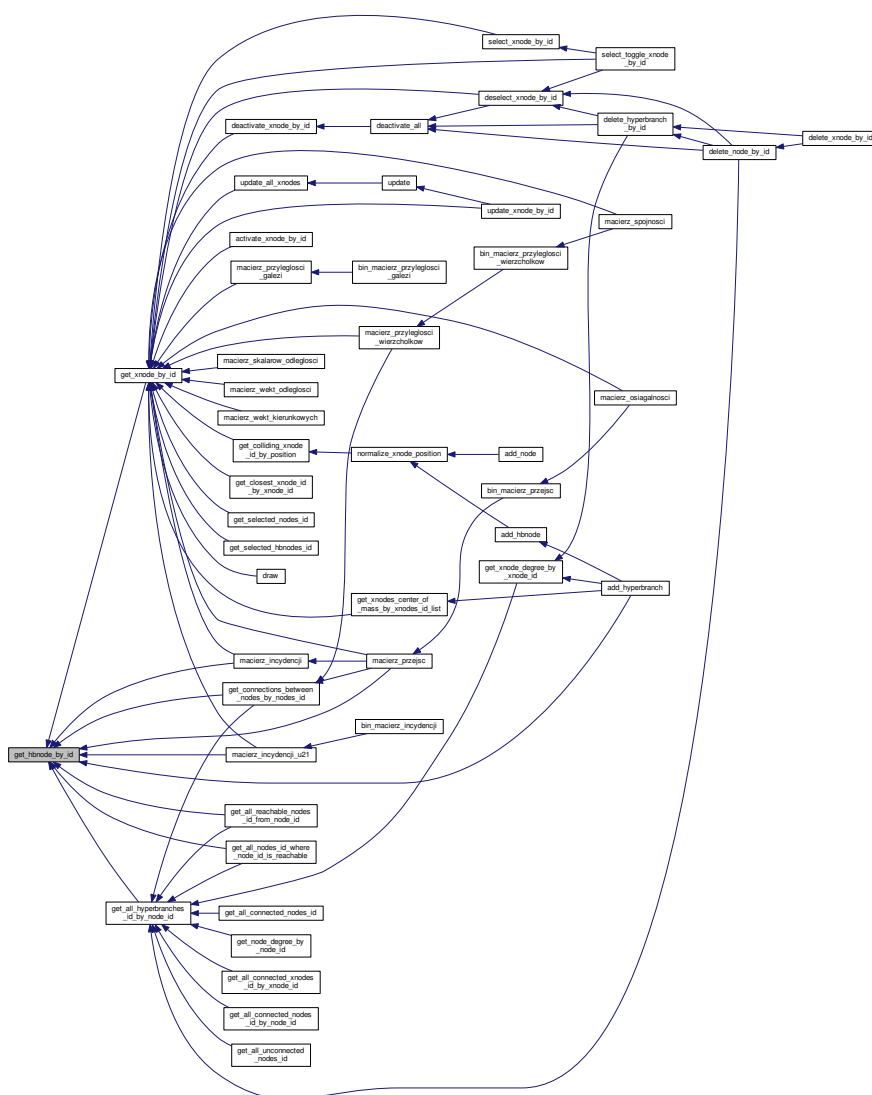
| | |
|------------|-----------------|
| <i>hid</i> | Id hipergałęzi. |
|------------|-----------------|

Zwraca

Obiekt reprezentujący wizualnie hipergałąż.

Definicja w linii 315 pliku HyperGraph.py.

Oto graf wywoływań tej funkcji:



```
6.7.3.38 def get_hyperbranch_by_id ( self, hid )
```

Metoda zwracająca zestaw (listę, krotkę, zbiór) id wierzchołków w danej hipergalezi.

Parametry

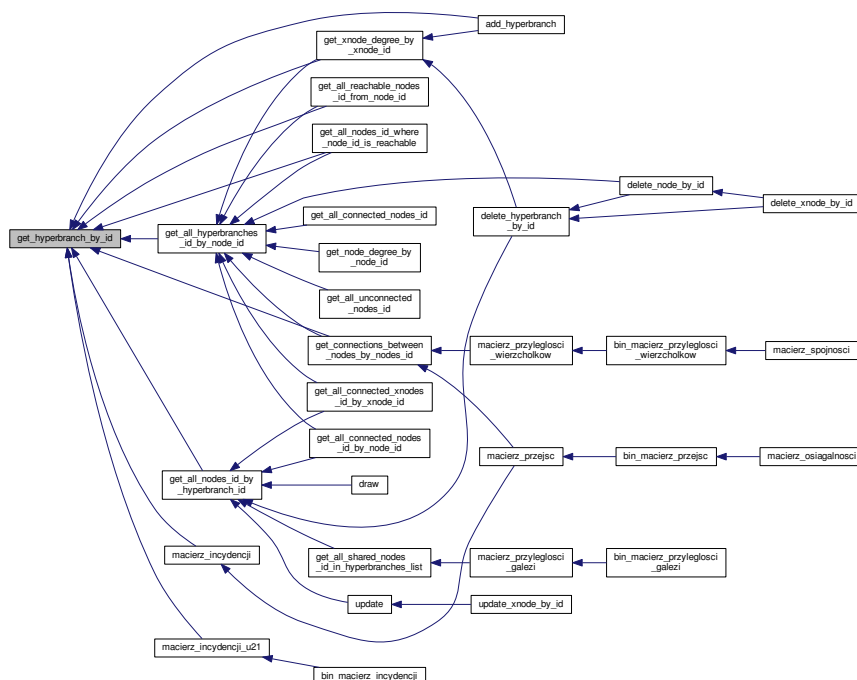
| | |
|------------|-----------------|
| <i>hid</i> | Id hipergałęzi. |
|------------|-----------------|

Zwraca

Zestaw id wierzchołków w formie listy, zbioru lub krotki.

Definicja w linii 321 pliku HyperGraph.py.

Oto graf wywoływań tej funkcji:



```
6.7.3.39 def get_max_xnode_index ( self )
```

Metoda zwracająca maksymalny id elementu hipergrafu.

Zwraca

Wartość liczbowa maksymalnego id elementu w hipergrafie.

Definicja w linii 173 pliku HyperGraph.py.

```
6.7.3.40 def get_node_by_id ( self, nid )
```

Metoda zwracająca obiekt wizualnie reprezentujący wierzchołek po jego id.

Parametry

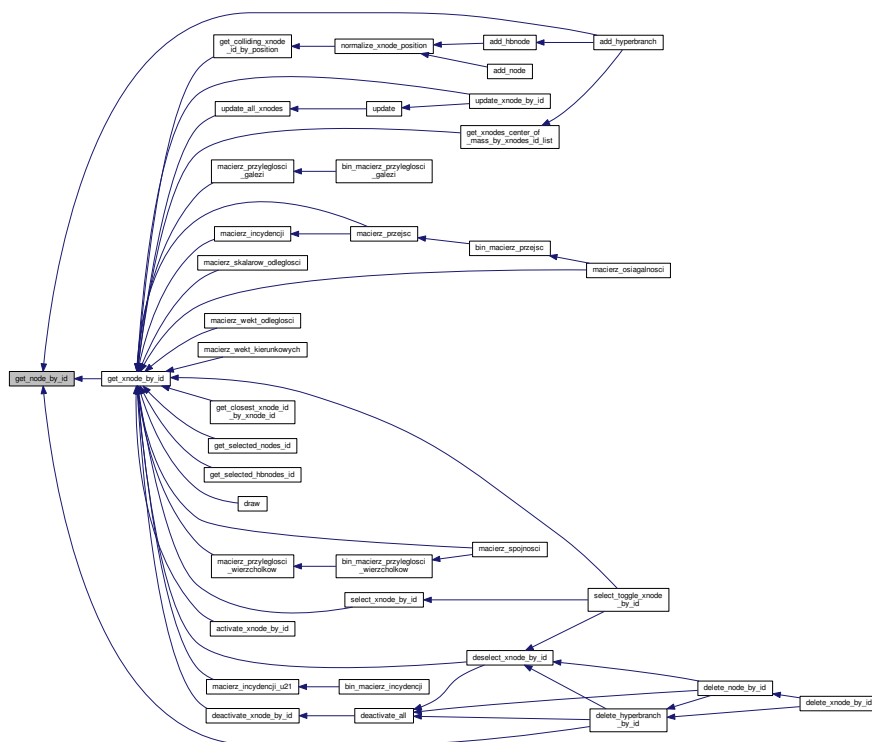
| | |
|------------|-----------------|
| <i>nid</i> | Id wierzchołka. |
|------------|-----------------|

Zwraca

Obiekt wizualnie reprezentujący wierzchołek.

Definicja w linii 199 pliku HyperGraph.py.

Oto graf wywoływań tej funkcji:



6.7.3.41 def get_node_degree_by_node_id (self, nid)

Metoda zwracająca stopień wierzchołka (liczbę hiperkrawędzi do niego połączonych).

Parametry

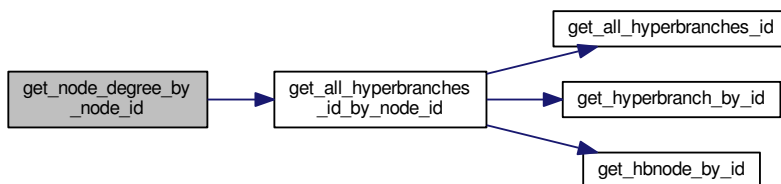
| | |
|------------|-----------------|
| <i>nid</i> | Id wierzchołka. |
|------------|-----------------|

Zwraca

Wartość liczbową stopnia wierzchołka.

Definicja w linii 265 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.42 `def get_selected_hbnodes_id (self)`

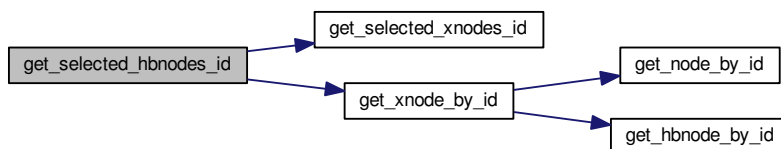
Metoda zwracająca listę zaznaczonych hipergałęzi hipergrafu.

Zwraca

Lista hipergałęzi zaznaczonych w kolejności zaznaczania (ostatni element zaznaczony jako ostatni).

Definicja w linii 503 pliku `HyperGraph.py`.

Oto graf wywołań dla tej funkcji:



6.7.3.43 `def get_selected_nodes_id (self)`

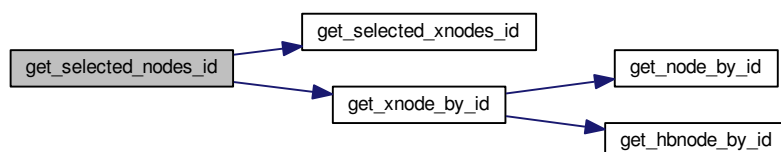
Metoda zwracająca listę zaznaczonych wierzchołków hipergrafu.

Zwraca

Lista wierzchołków zaznaczonych w kolejności zaznaczania (ostatni element zaznaczony jako ostatni).

Definicja w linii 498 pliku `HyperGraph.py`.

Oto graf wywołań dla tej funkcji:



6.7.3.44 def get_selected_xnodes_id (self)

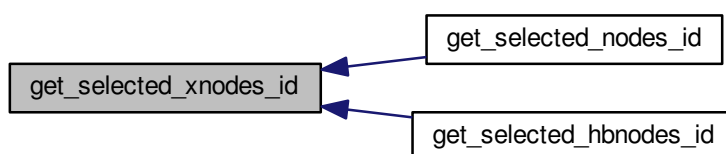
Metoda zwracająca listę zaznaczonych elementów hipergrafu.

Zwraca

Lista elementów zaznaczonych w kolejności zaznaczania (ostatni element zaznaczony jako ostatni).

Definicja w linii 488 pliku HyperGraph.py.

Oto graf wywołań tej funkcji:



6.7.3.45 def get_xnode_by_id (self, xid)

Metoda zwracająca obiekt wizualnie reprezentujący element hipergrafu po jego id.

Może być to wierzchołek lub hipergałąź.

Parametry

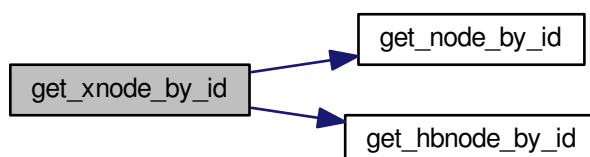
| | |
|------------|--------------|
| <i>xid</i> | Id elementu. |
|------------|--------------|

Zwraca

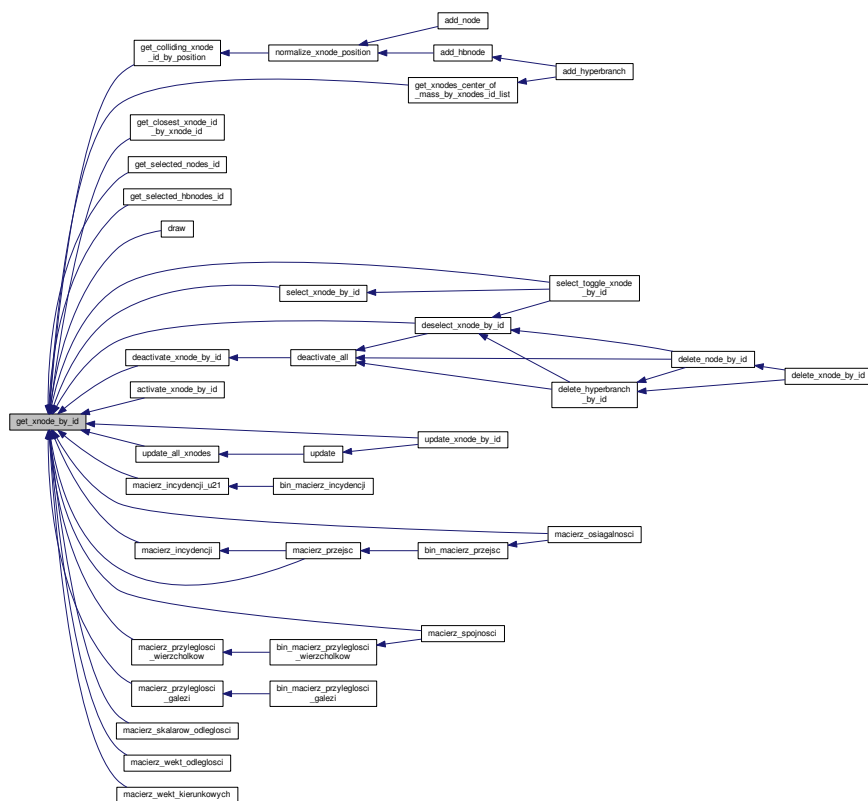
Obiekt wizualnie reprezentujący element.

Definicja w linii 162 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.7.3.46 def get_xnode_degree_by_xnode_id (self, xid, hbtpe = None)

Metoda zwracająca stopień elementu jako liczbę elementów z nim połączonych.

W przypadku wierzchołka jest to liczba hiperkrawędzi, które go zawierają. W przypadku hiperkrawędzi jest to liczba wierzchołków do niej należących.

Parametry

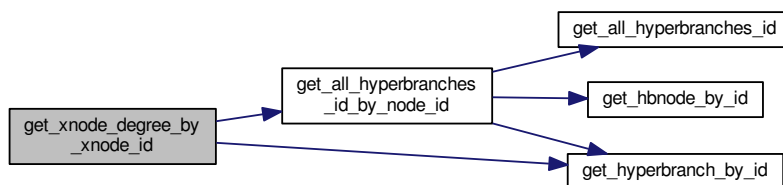
| | |
|--------------|-----------------------------|
| <i>xid</i> | Id elementu. |
| <i>hbtpe</i> | Opcjonalny typ hipergałęzi. |

Zwraca

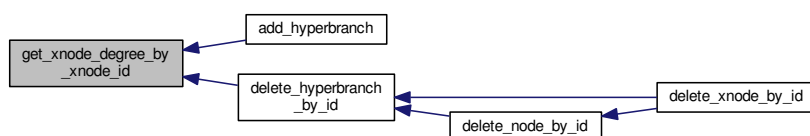
Wartość liczbową stopnia elementu.

Definicja w linii 187 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.47 `def get_xnodes_center_of_mass_by_xnodes_id_list (self, xid_list)`

Metoda obliczająca środek masy danych elementów.

Parametry

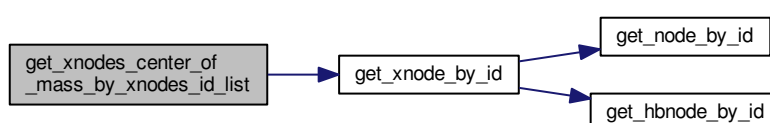
| | |
|-----------------------|--|
| <code>xid_list</code> | Lista id elementów, z której należy wyznaczyć środek masy. |
|-----------------------|--|

Zwraca

Punkt środka masy jako tablica biblioteki NumPy.

Definicja w linii 509 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.48 `def load_evolutionary_state (self, frame)`

Ładuje dany stan ewolucji hipergrafu do podglądu.

Parametry

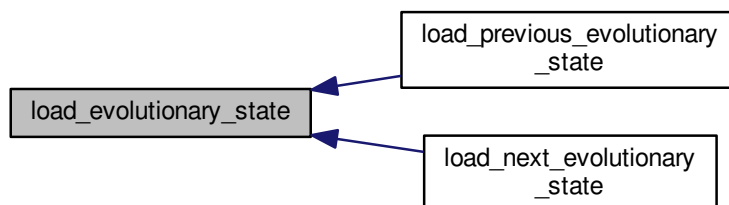
| | |
|--------------|--|
| <i>frame</i> | Zmienna informująca, który stan załadować. |
|--------------|--|

Definicja w linii 972 pliku `HyperGraph.py`.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.49 `def load_hg_from_dict (self, hgdict)`

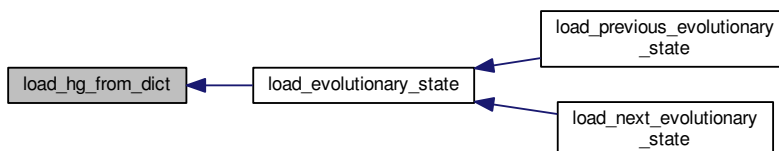
Metoda wypełniająca hipergraf za pomocą słownika wybranych elementów.

Parametry

| | |
|---------------|---|
| <i>hgdict</i> | Słownik zawierający elementy hipergrafu |
|---------------|---|

Definicja w linii 927 pliku HyperGraph.py.

Oto graf wywołań tej funkcji:

6.7.3.50 `def load_hg_from_tuple (self, hgtuple)`

Metoda wypełniająca hipergraf za pomocą krotki jego elementów.

Parametry

| | |
|----------------|---|
| <i>hgtuple</i> | Krotka zawierająca elementy hipergrafu. |
|----------------|---|

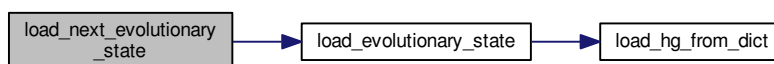
Definicja w linii 879 pliku HyperGraph.py.

6.7.3.51 `def load_next_evolutionary_state (self)`

Ładuje następny stan ewolucji hipergrafu do podglądu.

Definicja w linii 993 pliku HyperGraph.py.

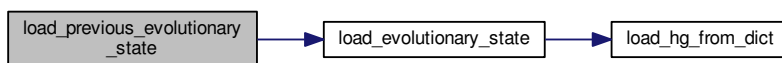
Oto graf wywołań dla tej funkcji:

6.7.3.52 `def load_previous_evolutionary_state (self)`

Ładuje poprzedni stan ewolucji hipergrafu do podglądu.

Definicja w linii 984 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.53 `def macierz_incydencji (self, xid_list = None)`

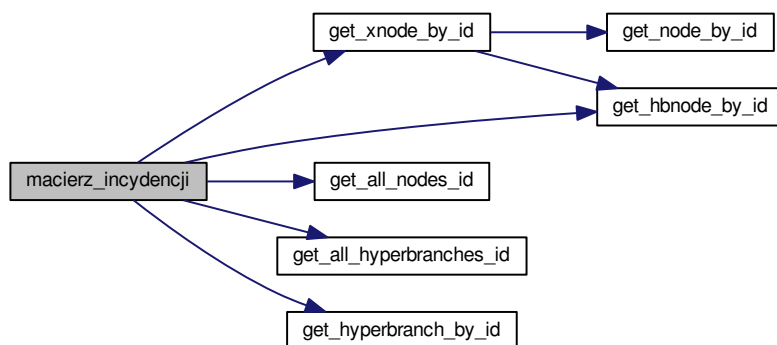
Metoda zwracająca macierz incydencji A w formie numerycznej.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1053 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.54 `def macierz_incydencji_u21 (self, xid_list = None)`

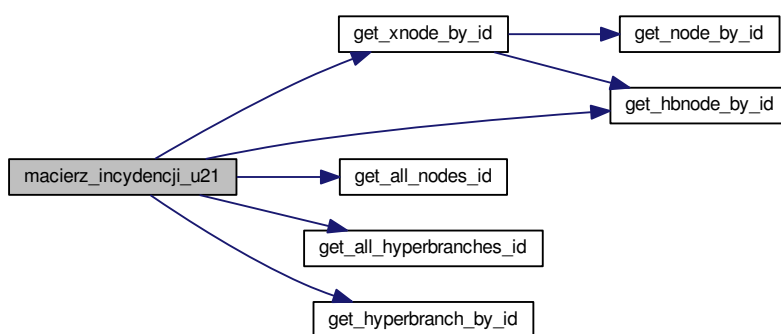
Metoda zwracająca macierz incydencji A w formie stringa.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1006 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.55 `def macierz_osiagalnosci (self, nodes_id = None)`

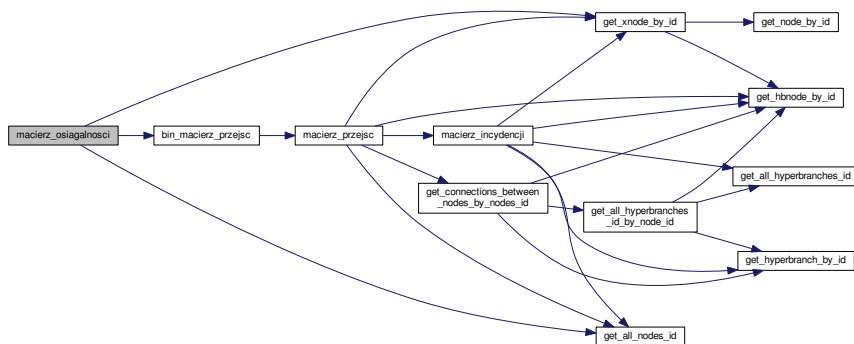
Metoda zwracająca macierz osiągalności D.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1282 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.56 `def macierz_przejsc (self, nodes_id = None)`

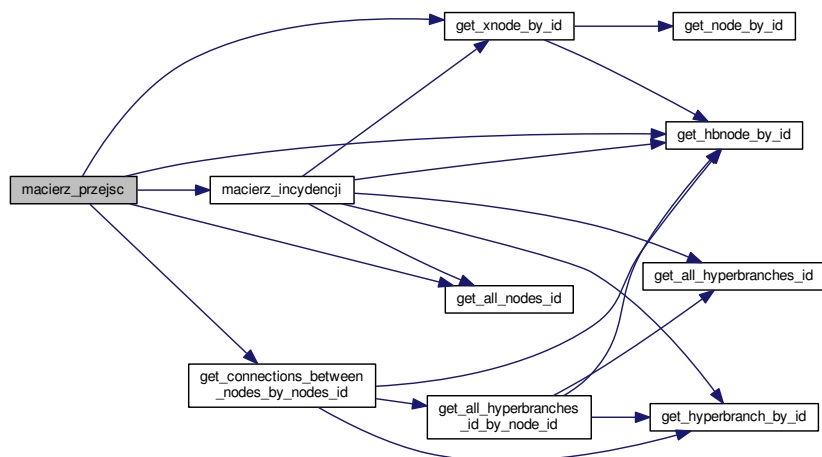
Metoda zwracająca macierz przejść P.

Zwraca

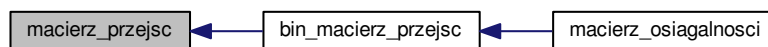
Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1201 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.57 `def macierz_przyleglosci_galezi (self, hyperbranches_id = None)`

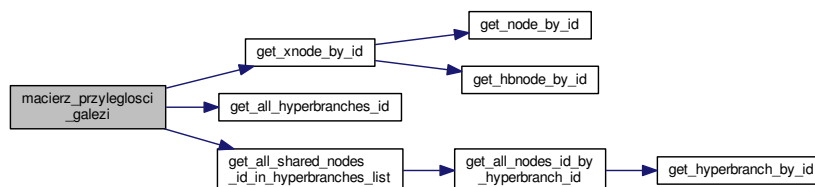
Metoda zwracająca macierz przyległości gałęzi B.

Zwraca

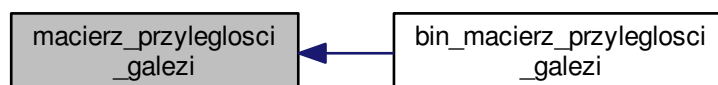
Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1159 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.58 `def macierz_przyleglosci_wierzchołkow (self, nodes_id = None)`

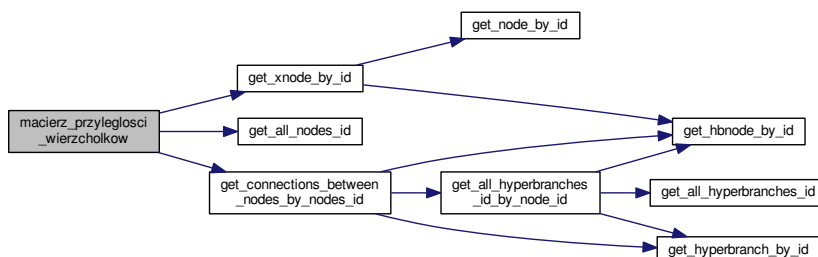
Metoda zwracająca macierz przyległości wierzchołków R.

Zwraca

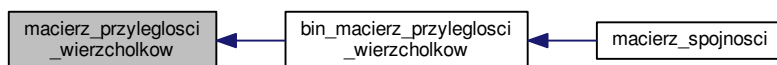
Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1117 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.7.3.59 `def macierz_skalarow_odleglosci (self, xid_list = None, cl = False)`

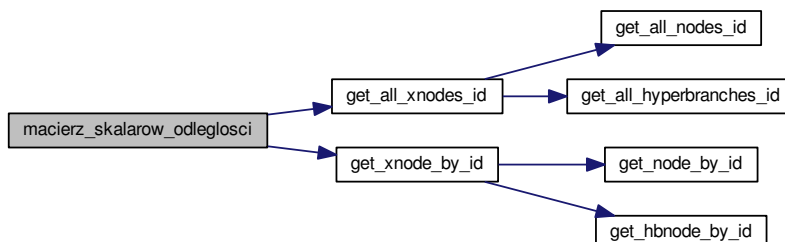
Metoda zwracająca macierz skalarów odległości pomiędzy elementami hipergrafu.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1354 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.60 `def macierz_spojnosci (self, nodes_id = None)`

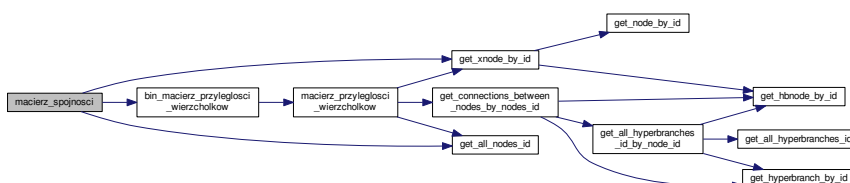
Metoda zwracająca macierz spójności S.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1318 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.61 `def macierz_wekt_kierunkowych (self, xid_list = None, cl = False)`

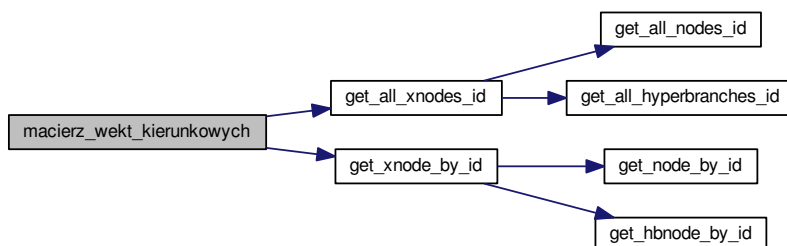
Metoda zwracająca macierz wektorów kierunku pomiędzy elementami hipergrafu.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1445 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.62 `def macierz_wekt_odleglosci (self, xid_list=None, cl=False)`

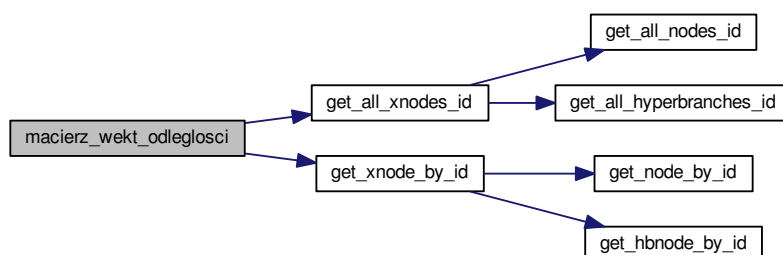
Metoda zwracająca macierz wektorów odległości pomiędzy elementami hipergrafu.

Zwraca

Słownik z macierzą wraz z opisem wierszy i kolumn.

Definicja w linii 1403 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.63 `def normalize_xnode_position (self, xnode, rmul=1.5)`

Metoda, która generuje nową pozycję elementu hipergrafu jeśli aktualna pozycja jest zajęta przez inny element.

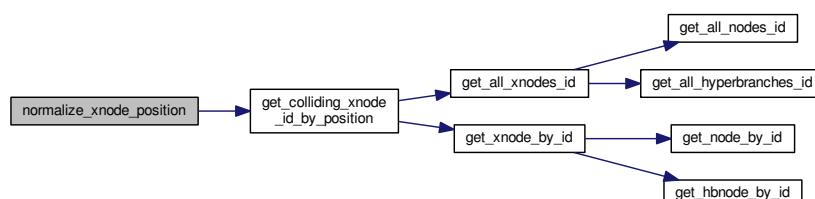
Element jest przesuwany w pętli o losowy wektor. Jeśli nowowygenerowana pozycja także jest zajęta, generowane są następne, aż do skutku.

Parametry

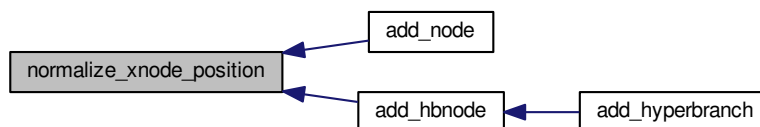
| | |
|--------------------|--|
| <code>xnode</code> | Element hipergrafu, którego pozycja ma być znormalizowana. |
|--------------------|--|

Definicja w linii 737 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



6.7.3.64 `def save_evolutionary_state (self, incrementcounter = True)`

Zapisuje aktualny stan hipergrafu jako stan jego ewolucji.

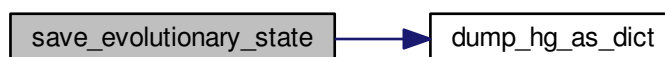
Jeśli licznik aktualnie aktywnego stanu nie wskazuje na ostatni stan, to wszystkie stany następujące po wskazywanym zostają usunięte, a następnie do powstałej w ten sposób listy dopisywany jest aktualny stan.

Parametry

| | |
|-------------------------|--|
| <i>incrementcounter</i> | Zmienna informująca, czy zwiększyć licznik zapisanych stanów ewolucji. |
|-------------------------|--|

Definicja w linii 954 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.65 `def select_toggle_xnode_by_id (self, xid)`

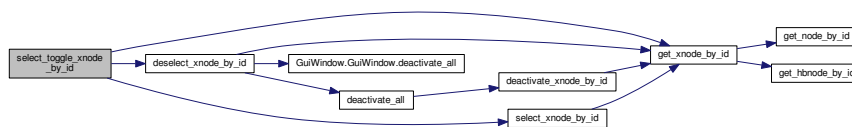
Metoda pozwalająca na zaznaczenie lub odznaczenie elementu hipergrafu po jego id.

Parametry

| | |
|------------|---|
| <i>xid</i> | Id elementu do zaznaczenia lub odznaczenia. |
|------------|---|

Definicja w linii 726 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.3.66 def select_xnode_by_id (self, xid)

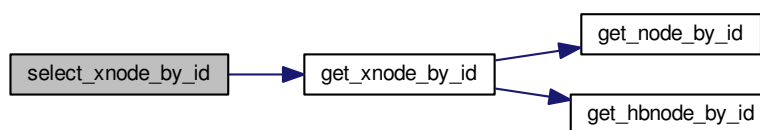
Metoda pozwalająca na zaznaczenie elementu hipergrafu po jego id.

Parametry

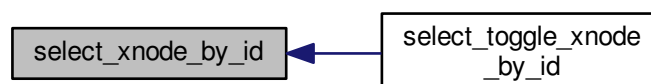
| | |
|------------|-----------------------------|
| <i>xid</i> | Id elementu do zaznaczenia. |
|------------|-----------------------------|

Definicja w linii 688 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.67 def update (self, dt)

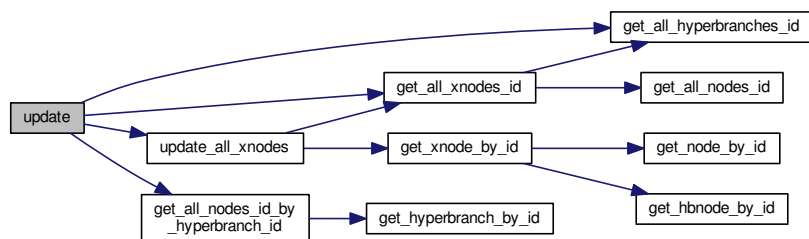
Metoda odświeżająca hipergraf.

Parametry

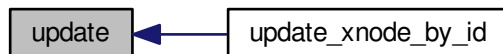
| | |
|-----------|--|
| <i>dt</i> | Czas pomiędzy kolejnymi odświeżeniami. |
|-----------|--|

Definicja w linii 779 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

6.7.3.68 `def update_all_xnodes (self, dt)`

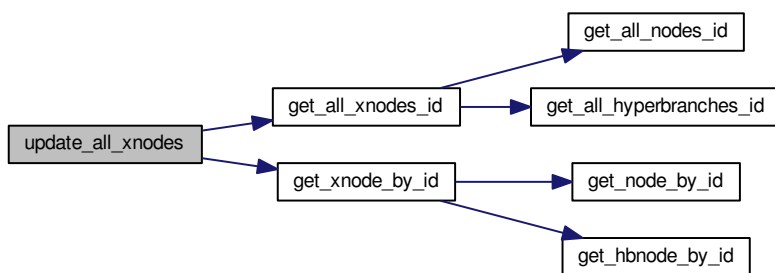
Metoda odświeżająca wszystkie wierzchołki i hipergałęzie hipergrafu.

Parametry

| | |
|-----------|--|
| <i>dt</i> | Czas pomiędzy kolejnymi odświeżeniami. |
|-----------|--|

Definicja w linii 751 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.7.3.69 `def update_xnode_by_id (self, xid)`

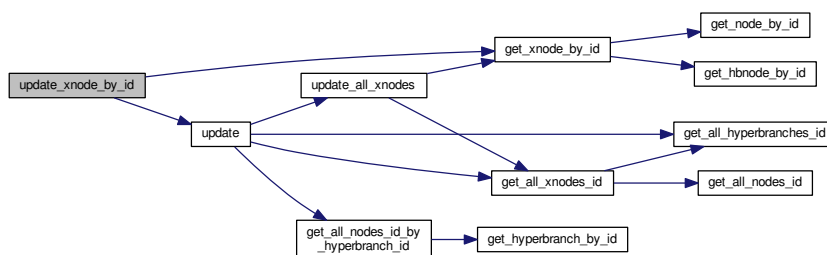
Metoda odświeżająca dany wierzchołek lub hipergałąź hipergrafu.

Parametry

| | |
|------------|---------------------------------------|
| <i>xid</i> | Id danego wierzchołka lub hipergrafu. |
|------------|---------------------------------------|

Definicja w linii 746 pliku HyperGraph.py.

Oto graf wywołań dla tej funkcji:



6.7.4 Dokumentacja atrybutów składowych

6.7.4.1 `activated_id_set`

Zmienna zbioru nieuporządkowanego przechowująca identyfikatory wyróżnionych elementów hipergrafu.

Definicja w linii 55 pliku HyperGraph.py.

6.7.4.2 `evolution_history`

Zmienna przechowująca historię kroków ewolucji hipergrafu.

Definicja w linii 58 pliku HyperGraph.py.

6.7.4.3 `evolution_view_current_frame`

Zmienna informująca o numerze aktualnie przeglądanej ewolucji hipergrafu.

Definicja w linii 73 pliku HyperGraph.py.

6.7.4.4 `P`

Zmienna słownikowa przechowująca relacje wierzchołków do hipergałęzi w hipergrafie.

Przykładowa zawartość: {16: (6, 7), 17: (7, 8), 18: (8, 9), 19: (3, 8), 10: (0, 1), 11: (1, 2), 12: (2, 3), 13: (3, 4), 14: (4, 5), 15: (5, 6)}

Definicja w linii 52 pliku HyperGraph.py.

6.7.4.5 `project_properties`

Zmienna zawierająca ustawienia aktualnego projektu.

Definicja w linii 66 pliku HyperGraph.py.

6.7.4.6 `selected_id_list`

Zmienna typu lista przechowująca w odpowiedniej kolejności id klikniętych wierzchołków.

Przykładowa zawartość: [0, 9]

Definicja w linii 63 pliku HyperGraph.py.

6.7.4.7 `U`

Zmienna słownikowa przechowująca hipergałęzie hipergrafu.

Przykładowa zawartość: {16: 16:ha1, 17: 17:ha2, 18: 18:ha3, 19: 19:ha4, 10: 10:ha5, 11: 11:ha6, 12: 12:ha7, 13: 13:ha8, 14: 14:ha9, 15: 15:ha10}

Definicja w linii 47 pliku HyperGraph.py.

6.7.4.8 X

Zmienna słownikowa przechowująca wierzchołki hipergrafu.

Przykładowa zawartość: {0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9}.

Definicja w linii 42 pliku HyperGraph.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/HyperGraph.py](#)

6.8 Dokumentacja klasy object

Diagram dziedziczenia dla object

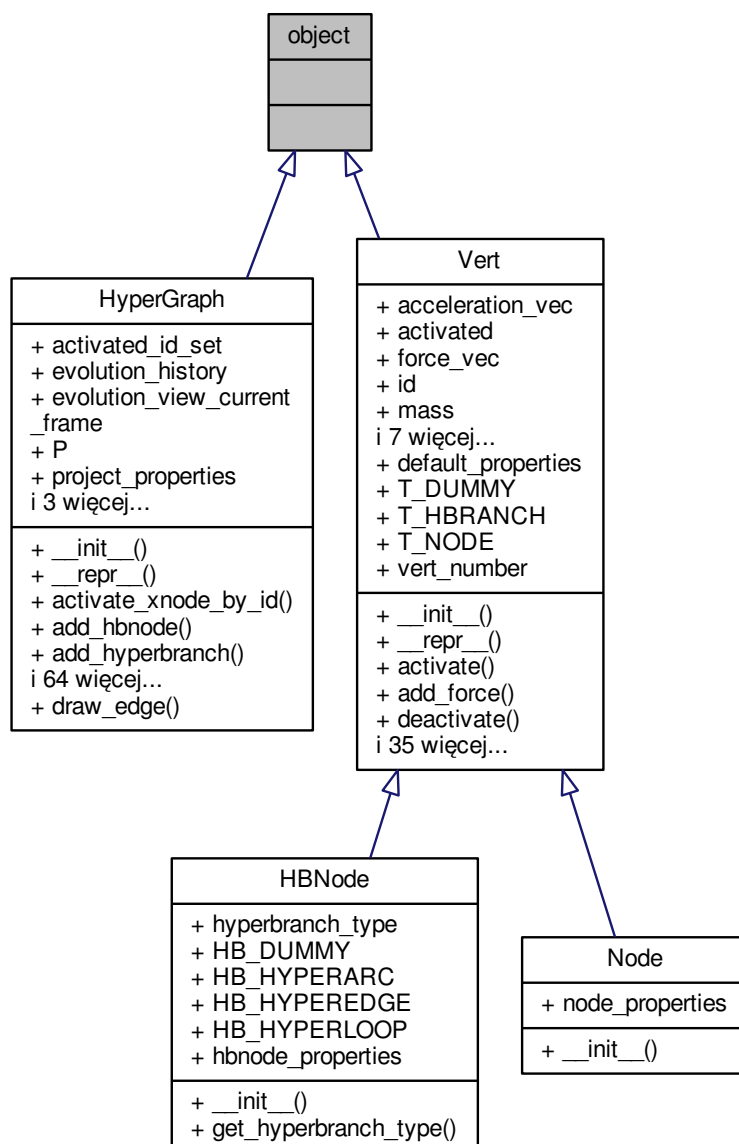
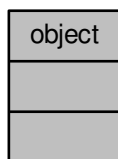


Diagram współpracy dla object:



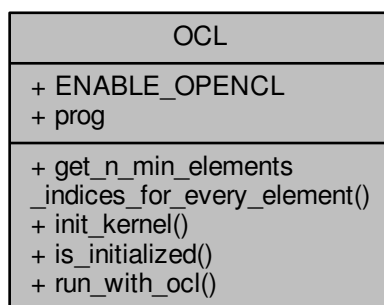
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/Vert.py](#)

6.9 Dokumentacja klasy OCL

Klasa narzędzi OpenCL.

Diagram współpracy dla OCL:



Statyczne metody publiczne

- def [get_n_min_elements_indices_for_every_element](#) (nparr, n)
Metoda, która dla każdego wiersza macierzy zwraca jego n najmniejszych elementów z pominięciem przekątnej.
- def [init_kernel](#) ()
Inicjalizacja kerneli OpenCL.
- def [is_initialized](#) ()
Metoda informująca, czy kerneli OpenCL zostały zainicjowane.
- def [run_with_ocl](#) (np_in_list, np_out_list, shape, oclfun, preset_outbuf=False, rw_outbuf=False)
Metoda wspomagająca pisanie innych funkcji przy pomocy PyOpenCL.

Statyczne atrybuty publiczne

- `ENABLE_OPENCL = DEFAULT_ENABLE_OPENCL`
- `prog = None`

6.9.1 Opis szczegółowy

Klasa narzędzi OpenCL.

Korzysta z biblioteki PyOpenCL. Aktualnie wąskim gardłem użytkowania OpenCL na GPU jest ilość transferów z/do GPU, dlatego też małe zestawy danych liczone są na procesorze głównym. Można zoptymalizować użycie GPU (zmniejszyć ilość zapytań na jednostkę czasu) poprzez wysłanie jednego zapytania, w którym znajdą się informacje co wykonać i dla jakich zestawów wierzchołków. Wymaga to napisania jednego, głównego kernela OpenCL, który zarządzał będzie pracą GPU w taki sposób, aby z danych zestawów wierzchołków przekazanych do GPU, ich pozycji itd generował odpowiedź i zwracał ją tylko raz na każde odświeżenie hipergrafu.

Definicja w linii 36 pliku OCL.py.

6.9.2 Dokumentacja funkcji składowych

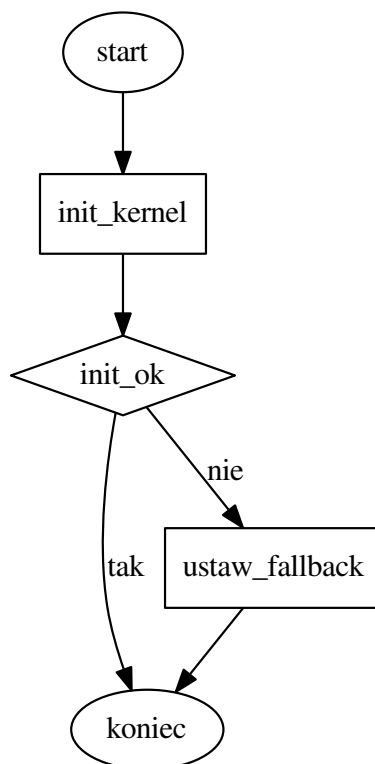
6.9.2.1 `def get_n_min_elements_indices_for_every_element(nparr, n) [static]`

Metoda, która dla każdego wiersza macierzy zwraca jego n najmniejszych elementów z pominięciem przekątnej.

Definicja w linii 43 pliku OCL.py.

6.9.2.2 `def init_kernel() [static]`

Inicjalizacja kerneli OpenCL.



Definicja w linii 79 pliku OCL.py.

6.9.2.3 `def is_initialized () [static]`

Metoda informująca, czy kernele OpenCL zostały zainicjowane.

Definicja w linii 196 pliku OCL.py.

6.9.2.4 `def run_with_ocl (np_in_list, np_out_list, shape, oclfun, preset_outbuf=False, rw_outbuf=False) [static]`

Metoda wspomagająca pisanie innych funkcji przy pomocy PyOpenCL.

Definicja w linii 116 pliku OCL.py.

6.9.3 Dokumentacja atrybutów składowych

6.9.3.1 `ENABLE_OPENCL = DEFAULT_ENABLE_OPENCL [static]`

Definicja w linii 37 pliku OCL.py.

6.9.3.2 `prog = None [static]`

Definicja w linii 39 pliku OCL.py.

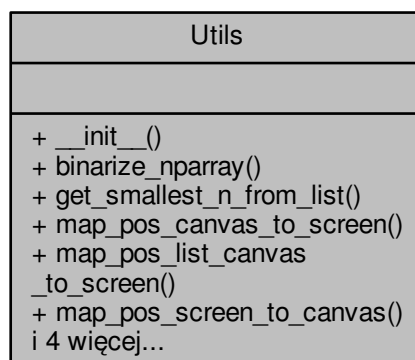
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/OCL.py](#)

6.10 Dokumentacja klasy Utils

Klasa [Utils](#).

Diagram współpracy dla Utils:



Metody publiczne

- `def __init__ (self)`

Statyczne metody publiczne

- `def binarize_ndarray (ndarray)`
Metoda binaryzująca macierz tekstową lub liczbową.
- `def get_smallest_n_from_list (list_to_check, n)`
Metoda pomocnicza zwracająca wybraną ilość najmniejszych elementów z listy.
- `def map_pos_canvas_to_screen (posc, center, zoom, pan_vec)`
Metoda mapująca punkt z przestrzeni płótna na punkt na przestrzeni ekranu.
- `def map_pos_list_canvas_to_screen (posc_list, center, zoom, pan_vec)`
Metoda mapująca listę punktów z przestrzeni płótna na listę punktów na przestrzeni ekranu.
- `def map_pos_screen_to_canvas (posc, center, zoom, pan_vec)`
Metoda mapująca punkt z przestrzeni ekranu na punkt na przestrzeni płótna.
- `def map_vec_canvas_to_screen (vec, zoom)`
Metoda mapująca wektor z przestrzeni płótna na przestrzeń ekranu .
- `def map_vec_screen_to_canvas (vec, zoom)`
Metoda mapująca wektor z przestrzeni ekranu na przestrzeń płótna.
- `def matrix_result_dict_as_string (M_dict)`
Metoda przekształcająca wynikowy słownik macierzy hipergrafu na tekst.
- `def suma_szeregu_geometrycznego_ndarray (ndarray, n)`
Metoda wykonująca sumę na macierzy $A: A^1 + A^2 + \dots$

6.10.1 Opis szczegółowy

Klasa `Utils`.

Klasa zawierająca przydatne narzędzia, wspomagające pracę aplikacji.

Definicja w linii 17 pliku `Utils.py`.

6.10.2 Dokumentacja konstruktora i destruktora

6.10.2.1 `def __init__ (self)`

Definicja w linii 18 pliku `Utils.py`.

6.10.3 Dokumentacja funkcji składowych

6.10.3.1 `def binarize_ndarray (ndarray) [static]`

Metoda binaryzująca macierz tekstową lub liczbową.

Parametry

| | |
|----------------------|----------------------------------|
| <code>ndarray</code> | Macierz NumPy do zbinaryzowania. |
|----------------------|----------------------------------|

Zwraca

Macierz zbiniaryzowana.

Definicja w linii 121 pliku Utils.py.

```
6.10.3.2 def get_smallest_n_from_list ( list_to_check, n ) [static]
```

Metoda pomocnicza zwracająca wybraną ilość najmniejszych elementów z listy.

Parametry

| | |
|----------------------|--|
| <i>list_to_check</i> | Lista elementów do przeskanowania. |
| <i>n</i> | Maksymalna ilość elementów do zwrócenia. |

Zwraca

Gotowa lista n (lub mniej) najmniejszych elementów.

Definicja w linii 82 pliku Utils.py.

```
6.10.3.3 def map_pos_canvas_to_screen ( posc, center, zoom, pan_vec ) [static]
```

Metoda mapująca punkt z przestrzeni płótna na punkt na przestrzeni ekranu.

Zajmuje 20%-30% czasu CPU.

Parametry

| | |
|----------------|--|
| <i>posc</i> | Punkt do zmapowania. |
| <i>center</i> | Punkt środka ekranu w bazie współrzędnych ekranu. |
| <i>zoom</i> | Współczynnik powiększenia płaszczyzny rysowania elementów w stosunku do bazy współrzędnych ekranu. Im większy tym większe rysowane są elementy na ekranie. |
| <i>pan_vec</i> | Wektor przesunięcia punktu środka płaszczyzny rysowania elementów w stosunku do lewego górnego rogu ekranu. Wyrażony w bazie płaszczyzny rysowania elementów (nie ekranu!) |

Zwraca

Zmapowany punkt na przestrzeni ekranu.

Definicja w linii 32 pliku Utils.py.

```
6.10.3.4 def map_pos_list_canvas_to_screen ( posc_list, center, zoom, pan_vec ) [static]
```

Metoda mapująca listę punktów z przestrzeni płótna na listę punktów na przestrzeni ekranu.

Definicja w linii 37 pliku Utils.py.

```
6.10.3.5 def map_pos_screen_to_canvas ( poss, center, zoom, pan_vec ) [static]
```

Metoda mapująca punkt z przestrzeni ekranu na punkt na przestrzeni płótna.

Parametry

| | |
|----------------|--|
| <i>poss</i> | Punkt do zmapowania. |
| <i>center</i> | Punkt środka ekranu w bazie współrzędnych ekranu. |
| <i>zoom</i> | Współczynnik powiększenia płaszczyzny rysowania elementów w stosunku do bazy współrzędnych ekranu. Im większy tym większe rysowane są elementy na ekranie. |
| <i>pan_vec</i> | Wektor przesunięcia punktu środka płaszczyzny rysowania elementów w stosunku do lewego górnego rogu ekranu. Wyrażony w bazie płaszczyzny rysowania elementów (nie ekranu!) |

Zwraca

Zmapowany punkt na przestrzeni płótna.

Definicja w linii 56 pliku Utils.py.

```
6.10.3.6 def map_vec_canvas_to_screen ( vec, zoom ) [static]
```

Metoda mapująca wektor z przestrzeni płótna na przestrzeń ekranu .

Parametry

| | |
|-------------|---|
| <i>vec</i> | Wektor do zmapowania. |
| <i>zoom</i> | Powiększenie płótna w stosunku do płaszczyzny ekranu. |

Zwraca

Zmapowany wektor na przestrzeni ekranu.

Definicja w linii 65 pliku Utils.py.

```
6.10.3.7 def map_vec_screen_to_canvas ( vec, zoom ) [static]
```

Metoda mapująca wektor z przestrzeni ekranu na przestrzeń płótna.

Parametry

| | |
|-------------|---|
| <i>vec</i> | Wektor do zmapowania. |
| <i>zoom</i> | Powiększenie płótna w stosunku do płaszczyzny ekranu. |

Zwraca

Zmapowany wektor na przestrzeni płótna.

Definicja w linii 74 pliku Utils.py.

```
6.10.3.8 def matrix_result_dict_as_string ( M_dict ) [static]
```

Metoda przekształcająca wynikowy słownik macierzy hipergrafu na tekst.

Parametry

| | |
|---------------|---|
| <i>M_dict</i> | Słownik macierzy hipergrafu zawierający macierz, opisy wierszy i opisy kolumn macierzy. |
|---------------|---|

Zwraca

Wynikowa postać tekstowa macierzy.

Definicja w linii 99 pliku Utils.py.

6.10.3.9 `def suma_szeregu_geometrycznego_nparray (nparray, n) [static]`

Metoda wykonująca sumę na macierzy A: $A^1 + A^2 + \dots$

Parametry

| | |
|----------------|--|
| <i>nparray</i> | Macierz NumPy, na której pracuje metoda. |
| <i>n</i> | Maksymalna wartość potęgi. |

Zwraca

Macierz wynikowa NumPy.

Definicja w linii 157 pliku Utils.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/Utils.py](#)

6.11 Dokumentacja klasy HBNode

Klasa [HBNode](#).

Diagram dziedziczenia dla HBNode

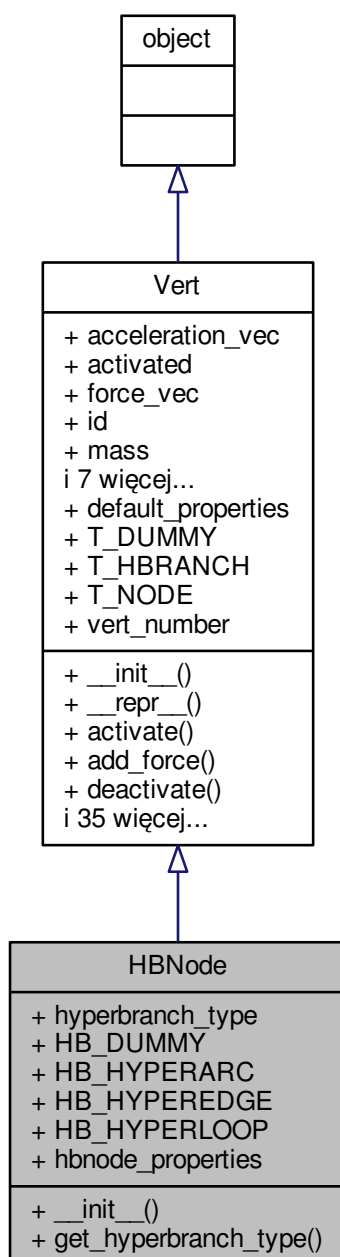
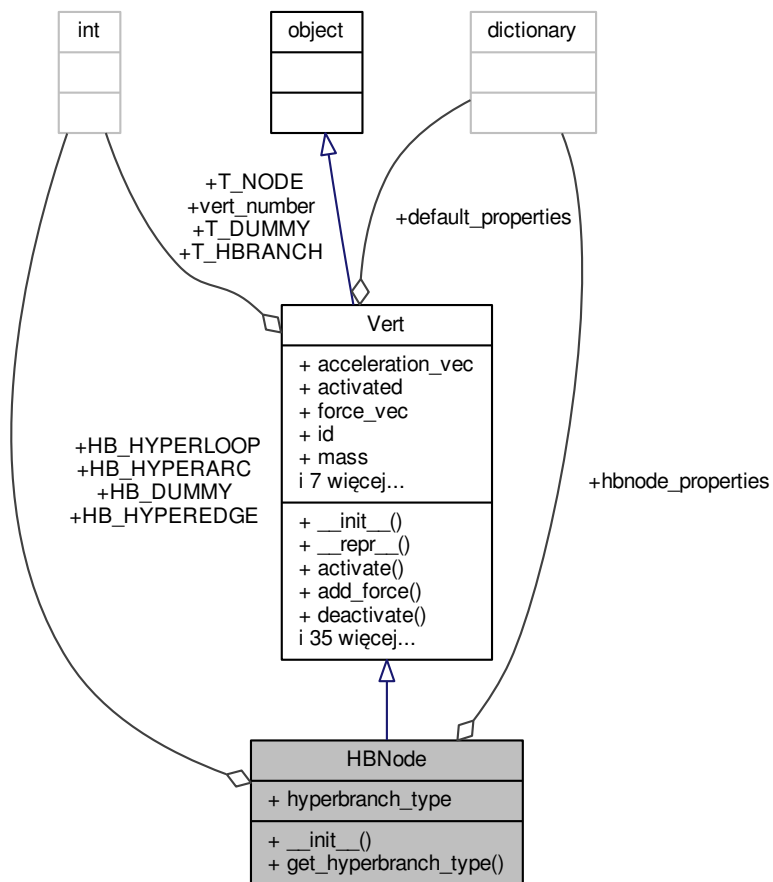


Diagram współpracy dla HBNode:



Metody publiczne

- `def __init__(self, name, hbtype, pos, prop_dict)`
Konstruktor.
- `def get_hyperbranch_type(self)`
Metoda zwracająca typ hipergałęzi.

Atrybuty publiczne

- `hyperbranch_type`
Zmienna oznaczająca typ hipergałęzi (hiperkrawędź, hiperłuk, hiperpętla).

Statyczne atrybuty publiczne

- `int HB_DUMMY = 3`
Wartość enum oznaczająca typ pusty, wierzchołek nie jest tworzony w celu dodania go do hipergrafu.
- `int HB_HYPERARC = 1`

- *Wartość enum oznaczająca hiperłuk.*
• int `HB_HYPEREDGE` = 0
- *Wartość enum oznaczająca hiperkrawędź.*
• int `HB_HYPERLOOP` = 2
- *Wartość enum oznaczająca hiperpętlę.*
• dictionary `hbnode_properties`
• *Domyślne właściwości hipergałęzi.*

6.11.1 Opis szczegółowy

Klasa `HBNode`.

Klasa elementu wizualizacji hipergrafu. Opisuje wizualnie hipergałęzie hipergrafu.

Definicja w linii 455 pliku `Vert.py`.

6.11.2 Dokumentacja konstruktora i destruktora

6.11.2.1 `def __init__(self, name, hbtype, pos, prop_dict)`

Konstruktor.

Definicja w linii 476 pliku `Vert.py`.

6.11.3 Dokumentacja funkcji składowych

6.11.3.1 `def get_hyperbranch_type(self)`

Metoda zwracająca typ hipergałęzi.

Zwraca

Typ hipergałęzi. Może to być hiperkrawędź, hiperłuk lub hiperpętla.

Definicja w linii 501 pliku `Vert.py`.

6.11.4 Dokumentacja atrybutów składowych

6.11.4.1 `int HB_DUMMY = 3 [static]`

Wartość enum oznaczająca typ pusty, wierzchołek nie jest tworzony w celu dodania go do hipergrafu.

Definicja w linii 468 pliku `Vert.py`.

6.11.4.2 `int HB_HYPERARC = 1 [static]`

Wartość enum oznaczająca hiperłuk.

Definicja w linii 462 pliku `Vert.py`.

6.11.4.3 `int HB_HYPEREDGE = 0` `[static]`

Wartość enum oznaczająca hiperkrawędź.

Definicja w linii 459 pliku Vert.py.

6.11.4.4 `int HB_HYPERLOOP = 2` `[static]`

Wartość enum oznaczająca hiperpętlę.

Definicja w linii 465 pliku Vert.py.

6.11.4.5 `dictionary hbnnode_properties` `[static]`

Wartość początkowa:

```
1 = {  
2     "elements": "None"  
3 }
```

Domyślne właściwości hipergałęzi.

Definicja w linii 471 pliku Vert.py.

6.11.4.6 `hyperbranch_type`

Zmienna oznaczająca typ hipergałęzi (hiperkrawędź, hiperluk, hiperpętla).

Definicja w linii 483 pliku Vert.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/Vert.py](#)

6.12 Dokumentacja klasy Node

Klasa [Node](#).

Diagram dziedziczenia dla Node

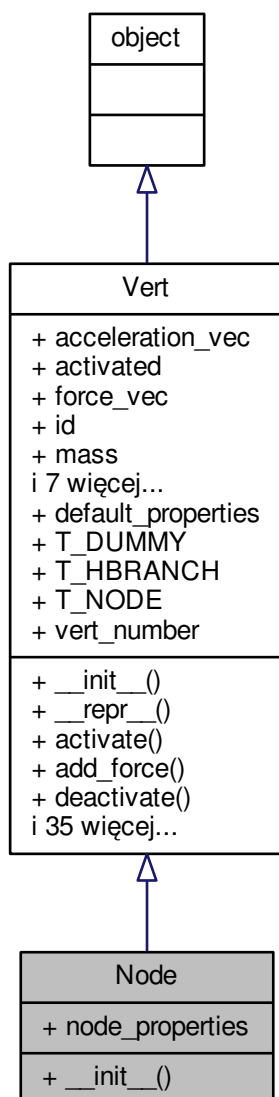
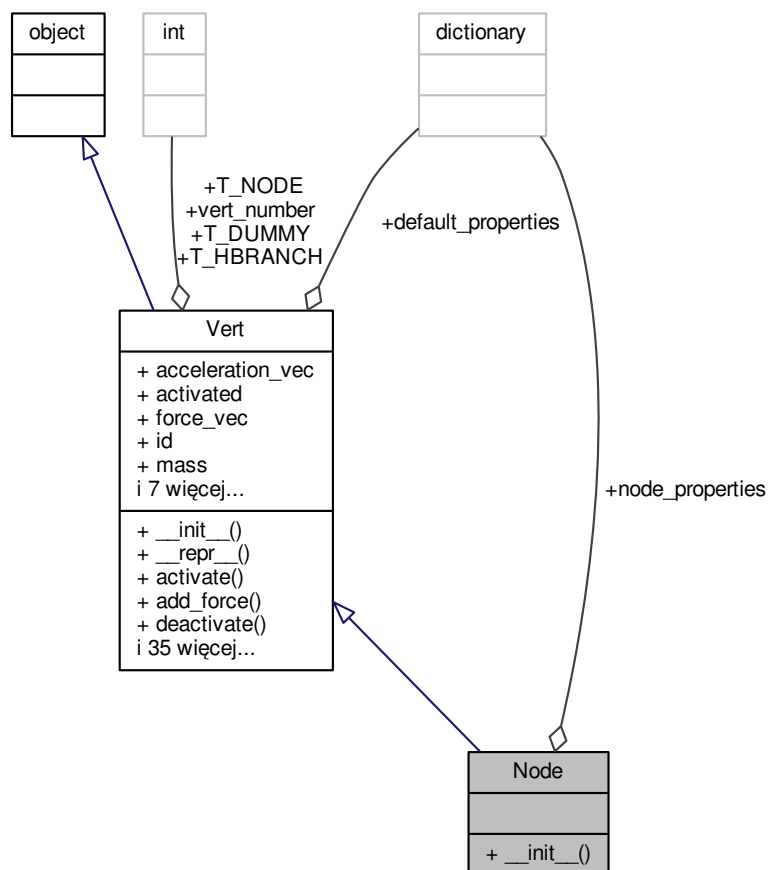


Diagram współpracy dla Node:



Metody publiczne

- `def __init__(self, name, pos, prop_dict)`
Konstruktor.

Statyczne atrybuty publiczne

- `dictionary node_properties`
Właściwości domyślne wierzchołka.

Dodatkowe Dziedziczone Składowe

6.12.1 Opis szczegółowy

Klasa `Node`.

Klasa elementu wizualizacji hipergrafu. Opisuje wizualnie wierzchołki hipergrafu.

Definicja w linii 438 pliku `Vert.py`.

6.12.2 Dokumentacja konstruktora i destruktora

6.12.2.1 `def __init__(self, name, pos, prop_dict)`

Konstruktor.

Definicja w linii 445 pliku `Vert.py`.

6.12.3 Dokumentacja atrybutów składowych

6.12.3.1 `dictionary node_properties [static]`

Wartość początkowa:

```
1 = {  
2 }
```

Właściwości domyślne wierzchołka.

Definicja w linii 441 pliku `Vert.py`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `app-python3-gtk3-cairo/Vert.py`

6.13 Dokumentacja klasy `Vert`

Klasa abstrakcyjna `Vert`.

Diagram dziedziczenia dla Vert

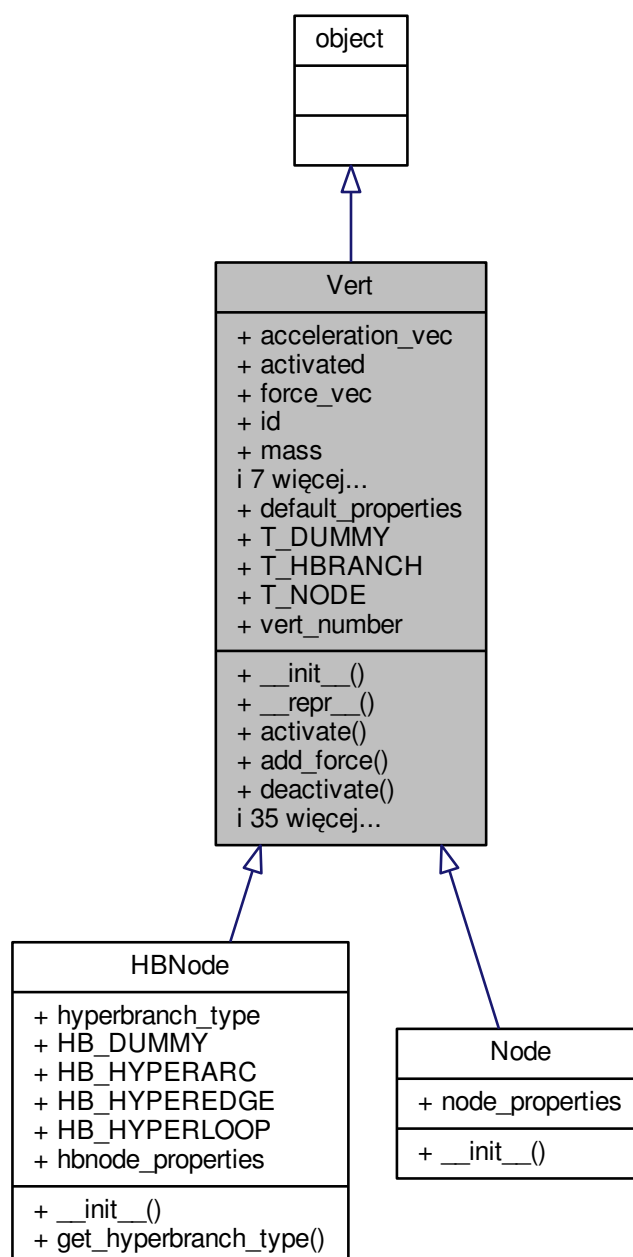
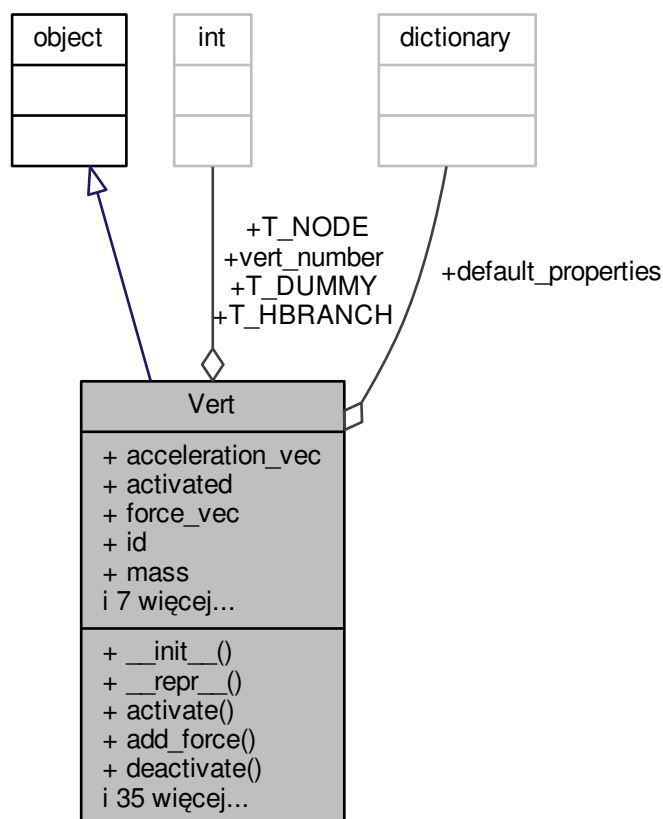


Diagram współpracy dla Vert:



Metody publiczne

- `def __init__(self, name, pos, prop_dict, vtype=T_NODE)`
Konstruktor.
- `def __repr__(self)`
Metoda zwracająca wartość tekstową hipergrafu.
- `def activate(self)`
Metoda wyróżniającą element.
- `def add_force(self, F)`
Metoda dodająca siłę do sił elementu.
- `def deactivate(self)`
Metoda zmieniającą stan elementu z wyróżnionego na normalny.
- `def deselect(self)`
Metoda odznaczającą element.
- `def distance_norm_from(self, other)`
Metoda obliczająca skalar odległości pomiędzy tym a innym elementem.
- `def distance_vect_from(self, other)`
Metoda obliczająca wektor przesunięcia między tym a innym elementem.

- def `draw` (self, cro, pos_zoomed, zoom)
Metoda rysująca dany element Zajmuje ok 10% czasu CPU.
- def `get_acceleration` (self)
Metoda zwracająca aktualne przyspieszenie elementu.
- def `get_id` (self)
Metoda zwracająca id obiektu.
- def `get_mass` (self)
Metoda zwracająca masę obiektu.
- def `get_name` (self)
Metoda zwracająca nazwę obiektu.
- def `get_position` (self)
Metoda zwracająca pozycję elementu w układzie płótna.
- def `get_properties_dict` (self)
Metoda zwracająca wszystkie właściwości obiektu.
- def `get_property_value` (self, pname)
Metoda zwracająca wartość danej właściwości.
- def `get_radius` (self)
Metoda zwracająca promień elementu hipergrafu.
- def `get_velocity` (self)
Metoda zwracająca prędkość elementu.
- def `get_vert_type` (self)
Metoda zwracająca typ elementu (wierzchołek lub hipergałąź)
- def `is_activated` (self)
Metoda zwracająca, czy dany element jest wyróżniony.
- def `is_colliding` (self, pos, rmul=1.0)
Metoda licząca, czy obiekt znajduje się na danej pozycji.
- def `is_colliding_with_another` (self, other_vert)
Metoda sprawdzająca czy ten element koliduje z innym elementem.
- def `is_of_vert_type` (self, vtype)
Metoda sprawdzająca, czy ten element jest danego typu.
- def `is_selected` (self)
Metoda zwracająca, czy dany element jest zaznaczony.
- def `select` (self)
Metoda zaznaczająca element.
- def `select_toggle` (self)
Metoda pozwalająca zaznaczyć lub odznaczyć element.
- def `set_acceleration` (self, acc)
Metoda ustawiająca przyspieszenie elementu.
- def `set_mass` (self, mass)
Metoda zwracająca masę obiektu.
- def `set_mass_from_degree` (self, deg)
Metoda ustawiająca masę elementu na podstawie jego stopnia w sensie wizualizacji.
- def `set_name` (self, name)
Metoda ustawiająca nazwę obiektu.
- def `set_position` (self, pos)
Metoda ustawiająca pozycję elementu.
- def `set_property_value` (self, pname, pval)
Metoda ustawiająca właściwość na daną wartość.
- def `set_rad_and_mass_from_degree` (self, deg)
Metoda ustawiająca parametry fizyczne elementu na podstawie jego stopnia w sensie wizualizacji.
- def `set_radius` (self, rad)

- Metoda ustawiająca wartość promienia obiektu.
- def `set_radius_from_degree` (self, deg)
 - Metoda przeliczająca stopień elementu (w sensie wizualizacji) na promień, a następnie przypisująca do obiektu ten promień.
- def `set_velocity` (self, vel)
 - Metoda ustawiająca prędkość elementu.
- def `stop_movement` (self)
 - Metoda zatrzymująca element.
- def `translate_by_vec` (self, vec)
 - Metoda pozwalająca przesuwać element za pomocą wektora.
- def `update` (self, dt)
 - Metoda uaktualniająca element.
- def `update_properties` (self, prop_dict)
 - Metoda aktualizująca właściwości obiektu.

Atrybuty publiczne

- `acceleration_vec`
 - Wektor przyspieszenia elementu.
- `activated`
 - Zmienna informująca, czy element został wyróżniony.
- `force_vec`
 - Wektor siły przypisany do elementu.
- `id`
 - ID aktualnego elementu (numer obiektu klasy `Vert`).
- `mass`
 - Zmienna przechowująca masę elementu.
- `name`
- `position_vec`
 - Współrzędne elementu.
- `properties_dict`
 - Zmienna słownikowa zawierająca właściwości elementu.
- `radius`
 - Zmienna przechowująca promień elementu.
- `selected`
 - Zmienna informująca, czy element jest zaznaczony.
- `velocity_vec`
 - Wektor prędkości elementu.
- `vert_type`
 - Zmienna enum zawierająca informację o typie elementu (wierzchołek lub hipergałąź).

Statyczne atrybuty publiczne

- dictionary `default_properties`
 - Domyślne właściwości elementu hipergrafu.
- int `T_DUMMY` = 2
 - Wartość enum oznaczająca pusty typ – obiekt nie będzie elementem hipergrafu.
- int `T_HBRANCH` = 1
 - Wartość enum oznaczająca hipergałąź.
- int `T_NODE` = 0
 - Wartość enum oznaczająca wierzchołek.
- int `vert_number` = 0
 - Zmienna klasy inkrementowana przy utworzeniu nowego elementu hipergrafu.

6.13.1 Opis szczegółowy

Klasa abstrakcyjna [Vert](#).

Klasa elementu wizualizacji hipergrafu. Opisuje wizualnie zarówno wierzchołki jak i hipergałęzie.

Definicja w linii 17 pliku Vert.py.

6.13.2 Dokumentacja konstruktora i destruktora

6.13.2.1 `def __init__(self, name, pos, prop_dict, vtype = T_NODE)`

Konstruktor.

Definicja w linii 55 pliku Vert.py.

6.13.3 Dokumentacja funkcji składowych

6.13.3.1 `def __repr__(self)`

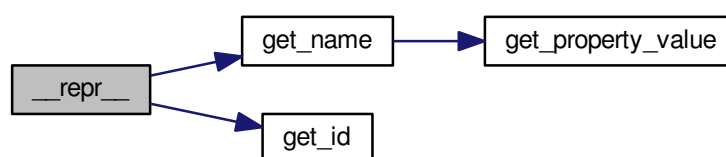
Metoda zwracająca wartość tekstową hipergrafu.

Zwraca

Wartość tekstowa elementu zwracająca id oraz nazwę, jeśli jest dostępna.

Definicja w linii 348 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



6.13.3.2 `def activate(self)`

Metoda wyróżniająca element.

Definicja w linii 372 pliku Vert.py.

6.13.3.3 `def add_force(self, F)`

Metoda dodająca siłę do sił elementu.

Parametry

| | |
|----------|------------------|
| <i>F</i> | Siła do dodania. |
|----------|------------------|

Definicja w linii 381 pliku Vert.py.

6.13.3.4 def deactivate (self)

Metoda zmieniająca stan elementu z wyróżnionego na normalny.

Definicja w linii 376 pliku Vert.py.

6.13.3.5 def deselect (self)

Metoda odznaczająca element.

Definicja w linii 368 pliku Vert.py.

Oto graf wywołań tej funkcji:

**6.13.3.6 def distance_norm_from (self, other)**

Metoda obliczająca skalar odległości pomiędzy tym a innym elementem.

Parametry

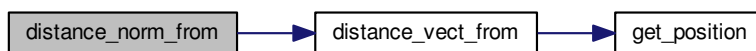
| | |
|--------------|---|
| <i>other</i> | Drugi element do obliczenia odległości. |
|--------------|---|

Zwraca

Skalar odległości.

Definicja w linii 229 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



6.13.3.7 `def distance_vect_from (self, other)`

Metoda obliczająca wektor przesunięcia między tym a innym elementem.

Parametry

| | |
|--------------|--------------------------------------|
| <i>other</i> | Drugi element do obliczenia wektora. |
|--------------|--------------------------------------|

Zwraca

Wektor przesunięcia.

Definicja w linii 223 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.13.3.8 `def draw (self, cro, pos_zoomed, zoom)`

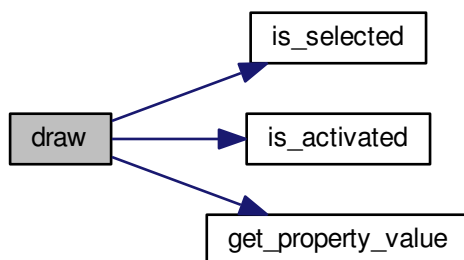
Metoda rysująca dany element Zajmuje ok 10% czasu CPU.

Parametry

| | |
|----------------|--|
| <i>cro</i> | Kontekst rysowania biblioteki Cairo. |
| <i>pan_vec</i> | Wektor przesunięcia punktu środka płaszczyzny rysowania elementów w stosunku do lewego górnego rogu ekranu. Wyrażony w bazie płaszczyzny rysowania elementów (nie ekranu!) |
| <i>center</i> | Punkt środka ekranu w bazie współrzędnych ekranu. |
| <i>zoom</i> | Współczynnik powiększenia płaszczyzny rysowania elementów w stosunku do bazy współrzędnych ekranu. Im większy tym większe rysowane są elementy na ekranie. |

Definicja w linii 284 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



6.13.3.9 def get_acceleration (self)

Metoda zwracająca aktualne przyspieszenie elementu.

Zwraca

Przyspieszenie elementu.

Definicja w linii 202 pliku Vert.py.

Oto graf wywoływań tej funkcji:



6.13.3.10 `def get_id (self)`

Metoda zwracająca id obiektu.

Zwraca

Id obiektu.

Definicja w linii 106 pliku Vert.py.

Oto graf wywołań tej funkcji:



6.13.3.11 `def get_mass (self)`

Metoda zwracająca masę obiektu.

Zwraca

Masa obiektu.

Definicja w linii 166 pliku Vert.py.

Oto graf wywołań tej funkcji:



6.13.3.12 `def get_name (self)`

Metoda zwracająca nazwę obiektu.

Zwraca

Nazwa obiektu.

Definicja w linii 111 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.13.3.13 `def get_position (self)`

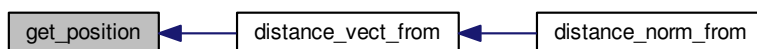
Metoda zwracająca pozycję elementu w układzie płótna.

Zwraca

Pozycja elementu.

Definicja w linii 217 pliku Vert.py.

Oto graf wywoływań tej funkcji:



6.13.3.14 `def get_properties_dict (self)`

Metoda zwracająca wszystkie właściwości obiektu.

Zwraca

Słownik właściwości obiektu.

Definicja w linii 130 pliku Vert.py.

6.13.3.15 `def get_property_value (self, pname)`

Metoda zwracająca wartość danej właściwości.

Parametry

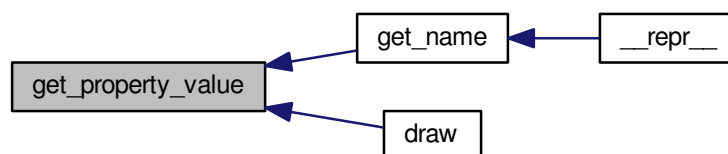
| | |
|--------------|-------------------------------|
| <i>pname</i> | Nazwa parametru do uzyskania. |
|--------------|-------------------------------|

Zwraca

Wartość uzyskanego parametru.

Definicja w linii 122 pliku Vert.py.

Oto graf wywoływań tej funkcji:



6.13.3.16 `def get_radius (self)`

Metoda zwracająca promień elementu hipergrafu.

Zwraca

Wartość liczbowa promienia elementu.

Definicja w linii 146 pliku Vert.py.

Oto graf wywoływań tej funkcji:



6.13.3.17 `def get_velocity (self)`

Metoda zwracająca prędkość elementu.

Zwraca

Prędkość elementu w jednostkach płótna.

Definicja w linii 192 pliku `Vert.py`.

Oto graf wywoływań tej funkcji:



6.13.3.18 `def get_vert_type (self)`

Metoda zwracająca typ elementu (wierzchołek lub hipergałąż)

Zwraca

Typ elementu.

Definicja w linii 212 pliku `Vert.py`.

Oto graf wywoływań tej funkcji:



6.13.3.19 `def is_activated (self)`

Metoda zwracająca, czy dany element jest wyróżniony.

Zwraca

True/False

Definicja w linii 257 pliku Vert.py.

Oto graf wywołań tej funkcji:



6.13.3.20 `def is_colliding (self, pos, rmul = 1.0)`

Metoda licząca, czy obiekt znajduje się na danej pozycji.

Parametry

| | |
|-------------|---|
| <i>pos</i> | Pozycja do sprawdzenia. |
| <i>rmul</i> | Mnożnik promienia, w którym szukana jest kolizja. |

Zwraca

True/False

Definicja w linii 238 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



6.13.3.21 `def is_colliding_with_another (self, other_vert)`

Metoda sprawdzająca czy ten element koliduje z innym elementem.

Parametry

| | |
|-------------------|----------------|
| <i>other_vert</i> | Drugi element. |
|-------------------|----------------|

Zwraca

True/False

Definicja w linii 245 pliku Vert.py.

6.13.3.22 `def is_of_vert_type (self, vtype)`

Metoda sprawdzająca, czy ten element jest danego typu.

Parametry

| | |
|--------------|--|
| <i>vtype</i> | |
|--------------|--|

Zwraca

True/False

Definicja w linii 263 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



6.13.3.23 `def is_selected (self)`

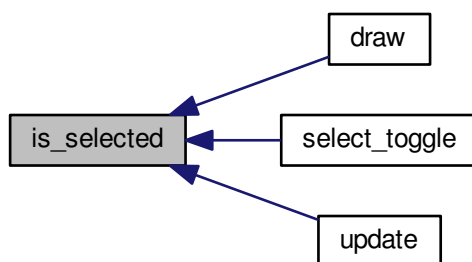
Metoda zwracająca, czy dany element jest zaznaczony.

Zwraca

True/False

Definicja w linii 252 pliku Vert.py.

Oto graf wywoływań tej funkcji:



6.13.3.24 `def select (self)`

Metoda zaznaczająca element.

Definicja w linii 364 pliku `Vert.py`.

Oto graf wywoływań tej funkcji:

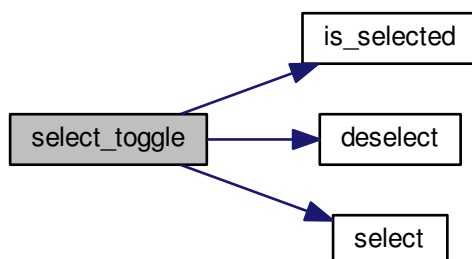


6.13.3.25 `def select_toggle (self)`

Metoda pozwalająca zaznaczyć lub odznaczyć element.

Definicja w linii 357 pliku `Vert.py`.

Oto graf wywołań dla tej funkcji:



6.13.3.26 `def set_acceleration (self, acc)`

Metoda ustawiająca przyspieszenie elementu.

Parametry

| | |
|------------|--------------------------|
| <i>acc</i> | Przyspieszenie elementu. |
|------------|--------------------------|

Definicja w linii 207 pliku Vert.py.

Oto graf wywoływań tej funkcji:



6.13.3.27 `def set_mass (self, mass)`

Metoda zwracająca masę obiektu.

Zwraca

Masa obiektu.

Definicja w linii 172 pliku Vert.py.

6.13.3.28 `def set_mass_from_degree (self, deg)`

Metoda ustawiająca masę elementu na podstawie jego stopnia w sensie wizualizacji.

Parametry

| | |
|------------|-------------------|
| <i>deg</i> | Stopień elementu. |
|------------|-------------------|

Definicja w linii 178 pliku Vert.py.

Oto graf wywoływań tej funkcji:



6.13.3.29 def set_name (self, name)

Metoda ustawiająca nazwę obiektu.

Parametry

| | |
|-------------|--------------------------------|
| <i>name</i> | Nazwa obiektu do przypisania.. |
|-------------|--------------------------------|

Definicja w linii 116 pliku Vert.py.

6.13.3.30 def set_position (self, pos)

Metoda ustawiająca pozycję elementu.

Parametry

| | |
|------------|-----------------------|
| <i>pos</i> | Pozycja do ustawienia |
|------------|-----------------------|

Definicja w linii 389 pliku Vert.py.

6.13.3.31 def set_property_value (self, pname, pval)

Metoda ustawiająca właściwość na daną wartość.

Parametry

| | |
|--------------|--------------------|
| <i>pname</i> | Nazwa właściwości. |
| <i>pval</i> | Wartość ustawiana. |

Definicja w linii 136 pliku Vert.py.

6.13.3.32 def set_rad_and_mass_from_degree (self, deg)

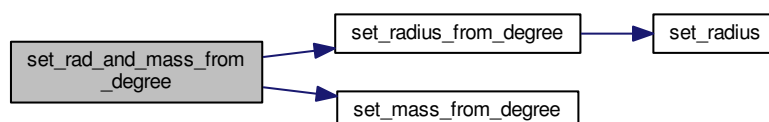
Metoda ustawiająca parametry fizyczne elementu na podstawie jego stopnia w sensie wizualizacji.

Parametry

| | |
|------------|-------------------|
| <i>deg</i> | Stopień elementu. |
|------------|-------------------|

Definicja w linii 186 pliku Vert.py.

Oto graf wywołań dla tej funkcji:

6.13.3.33 `def set_radius (self, rad)`

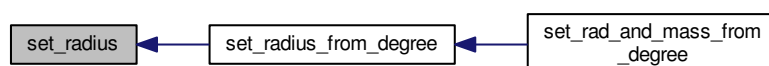
Metoda ustawiająca wartość promienia obiektu.

Parametry

| | |
|------------|------------------|
| <i>rad</i> | Promień obiektu. |
|------------|------------------|

Definicja w linii 151 pliku Vert.py.

Oto graf wywoływań tej funkcji:

6.13.3.34 `def set_radius_from_degree (self, deg)`

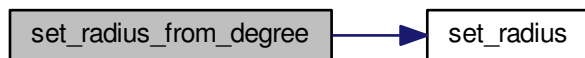
Metoda przeliczająca stopień elementu (w sensie wizualizacji) na promień, a następnie przypisująca do obiektu ten promień.

Parametry

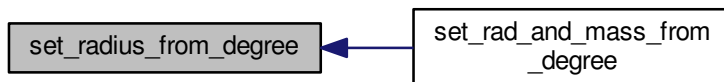
| | |
|------------|-------------------|
| <i>deg</i> | Stopień elementu. |
|------------|-------------------|

Definicja w linii 156 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.13.3.35 `def set_velocity (self, vel)`

Metoda ustawiająca prędkość elementu.

Parametry

| | |
|------------|--------------------------------|
| <i>vel</i> | Prędkość w jednostkach płótna. |
|------------|--------------------------------|

Definicja w linii 197 pliku Vert.py.

6.13.3.36 `def stop_movement (self)`

Metoda zatrzymująca element.

Definicja w linii 393 pliku Vert.py.

6.13.3.37 `def translate_by_vec (self, vec)`

Metoda pozwalająca przesuwać element za pomocą wektora.

Parametry

| | |
|------------|---|
| <i>vec</i> | Wektor, o który przesunięty zostanie element. |
|------------|---|

Definicja w linii 432 pliku Vert.py.

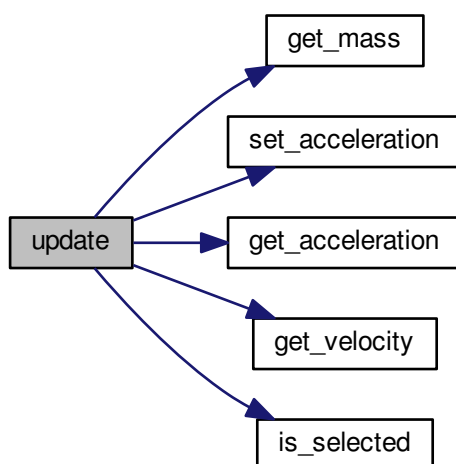
6.13.3.38 `def update (self, dt)`

Metoda uaktualniająca element.

Liczone są tu m.in. nowe właściwości dynamiczne na podstawie sił.

Definicja w linii 399 pliku Vert.py.

Oto graf wywołań dla tej funkcji:



6.13.3.39 `def update_properties (self, prop_dict)`

Metoda aktualizująca właściwości obiektu.

Parametry

| | |
|------------------------|--------------------------------------|
| <code>prop_dict</code> | Słownik nowych wartości właściwości. |
|------------------------|--------------------------------------|

Definicja w linii 141 pliku Vert.py.

6.13.4 Dokumentacja atrybutów składowych

6.13.4.1 `acceleration_vec`

Wektor przyspieszenia elementu.

Definicja w linii 86 pliku Vert.py.

6.13.4.2 activated

Zmienna informująca, czy element został wyróżniony.

Definicja w linii 98 pliku Vert.py.

6.13.4.3 dictionary default_properties [static]

Wartość początkowa:

```
1 = {
2     "id": "",
3     "name": "a",
4     "value": 1.0,
5
6     "ring_color": (1.0, 1.0, 1.0),
7     "selected_ring_color": (0.0, 0.0, 0.0),
8     "activated_ring_color": (1.0, 0.0, 0.0),
9
10    "bg_color": (0.3, 0.7, 0.3),
11    "selected_bg_color": (1.0, 0.2, 0.2),
12    "activated_bg_color": (1.0, 0.7, 0.35),
13
14    "text_color": (0.2, 0.2, 0.2),
15    "selected_text_color": (0.0, 0.0, 0.0),
16    "activated_text_color": (0.0, 0.0, 0.0)
17 }
```

Domyślne właściwości elementu hipergrafu.

Są one kopiowane przez każdą instancję dowolnej klasy dziedziczącej.

Definicja w linii 31 pliku Vert.py.

6.13.4.4 force_vec

Wektor siły przypisany do elementu.

Definicja w linii 83 pliku Vert.py.

6.13.4.5 id

ID aktualnego elementu (numer obiektu klasy [Vert](#)).

Definicja w linii 58 pliku Vert.py.

6.13.4.6 mass

Zmienna przechowująca masę elementu.

Definicja w linii 77 pliku Vert.py.

6.13.4.7 name

Definicja w linii 117 pliku Vert.py.

6.13.4.8 position_vec

Współrzędne elementu.

Definicja w linii 92 pliku Vert.py.

6.13.4.9 properties_dict

Zmienna słownikowa zawierająca właściwości elementu.

Definicja w linii 63 pliku Vert.py.

6.13.4.10 radius

Zmienna przechowująca promień elementu.

Definicja w linii 73 pliku Vert.py.

6.13.4.11 selected

Zmienna informująca, czy element jest zaznaczony.

Definicja w linii 95 pliku Vert.py.

6.13.4.12 int T_DUMMY = 2 [static]

Wartość enum oznaczająca pusty typ – obiekt nie będzie elementem hipergrafu.

Definicja w linii 27 pliku Vert.py.

6.13.4.13 int T_HBRANCH = 1 [static]

Wartość enum oznaczająca hipergałąź.

Definicja w linii 24 pliku Vert.py.

6.13.4.14 int T_NODE = 0 [static]

Wartość enum oznaczająca wierzchołek.

Definicja w linii 21 pliku Vert.py.

6.13.4.15 velocity_vec

Wektor prędkości elementu.

Definicja w linii 89 pliku Vert.py.

6.13.4.16 int vert_number = 0 [static]

Zmienna klasy inkrementowana przy utworzeniu nowego elementu hipergrafu.

Definicja w linii 50 pliku Vert.py.

6.13.4.17 vert_type

Zmienna enum zawierająca informację o typie elementu (wierzchołek lub hipergałąż).

Definicja w linii 69 pliku Vert.py.

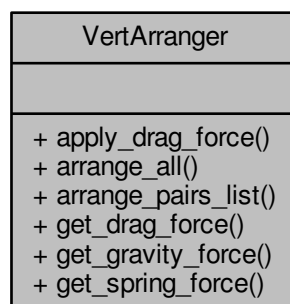
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/Vert.py](#)

6.14 Dokumentacja klasy VertArranger

Klasa [VertArranger](#).

Diagram współpracy dla VertArranger:



Statyczne metody publiczne

- def [apply_drag_force](#) (hgobj, xid_list, drag)
Metoda działająca siłą oporu zależną od prędkości na każdy element danej listy.
- def [arrange_all](#) (hgobj, xid_list, u_mul=1.0, k=0.0, grav=0.0)
Metoda licząca siły występujące pomiędzy podzbiorami elementów na danej liście id, oraz przypisująca je do nich.
- def [arrange_pairs_list](#) (hgobj, list1, list2, u_mul=1.0, k=0.0, grav=0.0)
Metoda licząca siły występujące pomiędzy parami elementów na danej liście par, oraz przypisująca je do nich.
- def [get_drag_force](#) (o, drag_coef)
Zwraca siłę oporu.
- def [get_gravity_force](#) (m1, m2, d, dir_vec, grav_const)
Zwraca siłę grawitacyjną.
- def [get_spring_force](#) (d, dir_vec, u, k)
Zwraca siłę sprężynową.

6.14.1 Opis szczegółowy

Klasa [VertArranger](#).

Klasa wspomagająca obliczanie nowych współrzędnych wierzchołków na płaszczyźnie płótna, na podstawie zależności fizycznych (grawitacja, sprężystość, opór). Jest dość wymagająca obliczeniowo. Należy ją zoptymalizować.

Definicja w linii 21 pliku VertArranger.py.

6.14.2 Dokumentacja funkcji składowych

6.14.2.1 `def apply_drag_force (hgobj, xid_list, drag) [static]`

Metoda działająca siłą oporu zależną od prędkości na każdy element danej listy.

Parametry

| | |
|-----------------|--|
| <i>hgobj</i> | Obiekt hipergrafu, którego elementy są rozpatrywane. |
| <i>xid_list</i> | Lista elementów do zastosowania oporu. |
| <i>drag</i> | Współczynnik oporu do zastosowania. |

Definicja w linii 293 pliku VertArranger.py.

6.14.2.2 `def arrange_all (hgobj, xid_list, u_mul=1.0, k=0.0, grav=0.0) [static]`

Metoda licząca siły występujące pomiędzy podzbiorami elementów na danej liście id, oraz przypisująca je do nich.

Parametry

| | |
|-----------------|--|
| <i>hgobj</i> | Obiekt hipergrafu. |
| <i>xid_list</i> | Lista id elementów. |
| <i>u_mul</i> | Mnożnik wartości zadanej odległości (opcjonalny). |
| <i>k</i> | Stała sprężystości oddziaływań sprężystych pomiędzy elementami (opcjonalna). |
| <i>grav</i> | Stała grawitacyjna (opcjonalna). |
| <i>nclosest</i> | Ilość innych elementów do obliczenia na każdy element (pomniejszona o jeden licznosc podzbioru). |

Definicja w linii 179 pliku VertArranger.py.

6.14.2.3 `def arrange_pairs_list (hgobj, list1, list2, u_mul=1.0, k=0.0, grav=0.0) [static]`

Metoda licząca siły występujące pomiędzy parami elementów na danej liście par, oraz przypisująca je do nich.

Parametry

| | |
|-----------------------------|---|
| <i>hgobj</i> | Obiekt hipergrafu. |
| <i>xnodes_id_pairs_list</i> | Lista par id elementów. |
| <i>u_mul</i> | Mnożnik wartości zadanej odległości (opcjonalny). |
| <i>k</i> | Stała sprężystości (opcjonalna). |
| <i>grav</i> | Stała grawitacyjna (opcjonalna). |

Definicja w linii 71 pliku VertArranger.py.

6.14.2.4 `def get_drag_force (o, drag_coef) [static]`

Zwraca siłę oporu.

Parametry

| | |
|------------------|---|
| <i>o</i> | Obiekt elementu, który jest rozpatrywany. |
| <i>drag_coef</i> | Współczynnik oporu. |

Zwraca

Siła oporu wprost proporcjonalna do prędkości elementu i jwspółczynnika oporu.

Definicja w linii 60 pliku VertArranger.py.

6.14.2.5 `def get_gravity_force (m1, m2, d, dir_vec, grav_const) [static]`

Zwraca siłę grawitacyjną.

Parametry

| | |
|-------------------|--------------------------------|
| <i>m1</i> | Masa elementu. |
| <i>m2</i> | Masa drugiego elementu. |
| <i>r1</i> | Promień elementu. |
| <i>r2</i> | Promień drugiego elementu. |
| <i>d</i> | Odległość pomiędzy elementami. |
| <i>dir_vec</i> | Wektor kierunku. |
| <i>grav_const</i> | Stała grawitacyjna. |

Zwraca

Siła grawitacyjna.

Definicja w linii 50 pliku VertArranger.py.

6.14.2.6 `def get_spring_force (d, dir_vec, u, k) [static]`

Zwraca siłę sprężynową.

Parametry

| | |
|----------------|---|
| <i>d</i> | Rzeczywista skalarna odległość między dwoma elementami. |
| <i>dir_vec</i> | Wektor jednostkowy kierunku. |
| <i>u</i> | Długość sprężyny (wartość zadana). |
| <i>k</i> | Stała sprężystości sprężyny. |

Zwraca

Siła obliczona zgodna z kierunkiem.

Definicja w linii 30 pliku VertArranger.py.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [app-python3-gtk3-cairo/VertArranger.py](#)

7 Dokumentacja plików

7.1 Dokumentacja pliku app-python3-gtk3-cairo/application-main.py

Przestrzenie nazw

- [application-main](#)
- [application_main](#)

Zmienne

- [pa](#) = `GuiWindow()`
Część właściwa programu głównego.

7.2 Dokumentacja pliku app-python3-gtk3-cairo/GuiWindow.py

Komponenty

- class [EditorDialog](#)
Klasa edytora skryptów.
- class [EntryDialog](#)
Klasa [EntryDialog](#).
- class [GuiWindow](#)
Klasa [GuiWindow](#).
- class [TextResultDialog](#)
Klasa [TextResultDialog](#) Pozwala na wyświetlanie wyników w formie tekstu.
- class [TreeViewResultDialog](#)
Klasa [TreeViewResultDialog](#) Pozwala na wyświetlanie wyników w formie listy wierszy.

Przestrzenie nazw

- [GuiWindow](#)

Funkcje

- def [redraw_function_decorator](#) (method)
Dekorator na funkcje, po których należy odświeżyć ekran.
- def [undo_redo_function_decorator](#) (method)
Dekorator na funkcje, które modyfikują hipergraf.

7.3 Dokumentacja pliku app-python3-gtk3-cairo/HgMatrixAnalyzer.py

Komponenty

- class [HgMatrixAnalyzer](#)
Klasa [HgMatrixAnalyzer](#).

Przestrzenie nazw

- [HgMatrixAnalyzer](#)

7.4 Dokumentacja pliku app-python3-gtk3-cairo/HyperGraph.py

Komponenty

- class [HyperGraph](#)
Klasa [HyperGraph](#).

Przestrzenie nazw

- [HyperGraph](#)

7.5 Dokumentacja pliku app-python3-gtk3-cairo/OCL.py

Komponenty

- class [OCL](#)
Klasa narzędzi [OpenCL](#).

Przestrzenie nazw

- [OCL](#)

Zmienne

- bool [DEFAULT_ENABLE_OPENCL](#) = False
Zmienna ustawiająca, czy program powinien korzystać z [OpenCL](#).

7.6 Dokumentacja pliku app-python3-gtk3-cairo/OpenCIKernels.cl

Funkcje

- float `dist_s` (float *ax*, float *ay*, float *bx*, float *by*)
- __kernel void `openc1_kernel_a_to_p` (__global const int **A_nm*, __global const int **hbtype_m*, __global int **kmax*, __global int **P_nn*)
- __kernel void `openc1_kernel_mat_line_sum_to_vec` (__global const float **mat_n_x_n_x_2*, __global float **vec_n_x_2*)
- __kernel void `openc1_kernel_n_closest_elements_array` (__global const float **shape_out*, __global const float **read_buffer*, __global float **write_buffer*)
- __kernel void `openc1_kernel_scalar_dist` (__global const float **read_buffer*, __global float **write_buffer*)
- __kernel void `openc1_kernel_vector_dir` (__global const float **read_buffer*, __global float **write_buffer*)
- __kernel void `openc1_kernel_vector_dist` (__global const float **read_buffer*, __global float **write_buffer*)
- __kernel void `openc1_kernel_vector_force` (__global const float **pos_vec_n_x_2*, __global const float **mass_vec_n_x_2*, __global const float **rad_vec_n_x_2*, __global const float **coeffs_vec_3_x_1*, __global float **force_mat_n_x_n_x_2*)
- __kernel void `openc1_kernel_vector_vector_force` (__global const float **pos_vec_n_x_2*, __global const float **mass_vec_n_x_2*, __global const float **rad_vec_n_x_2*, __global const float **coeffs_vec_3_x_1*, __global float **vec_n_x_2*)

7.6.1 Opis szczegółowy

OpenCIKernels

7.6.2 Dokumentacja funkcji

7.6.2.1 float `dist_s` (float *ax*, float *ay*, float *bx*, float *by*)

Funkcja obliczająca skalar odległości pomiędzy dwoma punktami.

Definicja w linii 10 pliku OpenCIKernels.cl.

7.6.2.2 __kernel void `openc1_kernel_a_to_p` (__global const int * *A_nm*, __global const int * *hbtype_m*, __global int * *kmax*, __global int * *P_nn*)

Funkcja obliczająca macierz *P* z macierzy *A*.

Definicja w linii 249 pliku OpenCIKernels.cl.

7.6.2.3 __kernel void `openc1_kernel_mat_line_sum_to_vec` (__global const float * *mat_n_x_n_x_2*, __global float * *vec_n_x_2*)

Funkcja obliczająca sumę dla każdego wiersza macierzy.

Definicja w linii 221 pliku OpenCIKernels.cl.

7.6.2.4 __kernel void `openc1_kernel_n_closest_elements_array` (__global const float * *shape_out*, __global const float * *read_buffer*, __global float * *write_buffer*)

Funkcja obliczająca indeksy *n* najmniejszych elementów dla każdego wiersza macierzy.

Definicja w linii 97 pliku OpenCIKernels.cl.

7.6.2.5 `__kernel void openc1_kernel_scalar_dist (__global const float * read_buffer, __global float * write_buffer)`

Funkcja obliczająca macierz skalarów odległości pomiędzy danymi punktami.

Definicja w linii 19 pliku OpenCIKernels.cl.

7.6.2.6 `__kernel void openc1_kernel_vector_dir (__global const float * read_buffer, __global float * write_buffer)`

Funkcja obliczająca macierz wektorów kierunkowych pomiędzy danymi punktami.

Definicja w linii 66 pliku OpenCIKernels.cl.

7.6.2.7 `__kernel void openc1_kernel_vector_dist (__global const float * read_buffer, __global float * write_buffer)`

Funkcja obliczająca macierz wektorów odległości pomiędzy danymi punktami.

Definicja w linii 40 pliku OpenCIKernels.cl.

7.6.2.8 `__kernel void openc1_kernel_vector_force (__global const float * pos_vec_n_x_2, __global const float * mass_vec_n_x_2, __global const float * rad_vec_n_x_2, __global const float * coeffs_vec_3_x_1, __global float * force_mat_n_x_n_x_2)`

Funkcja obliczająca macierz sił pomiędzy danymi punktami.

Definicja w linii 147 pliku OpenCIKernels.cl.

7.6.2.9 `__kernel void openc1_kernel_vector_vector_force (__global const float * pos_vec_n_x_2, __global const float * mass_vec_n_x_2, __global const float * rad_vec_n_x_2, __global const float * coeffs_vec_3_x_1, __global float * vec_n_x_2)`

Funkcja obliczająca wektor siły dla każdego punktu.

Definicja w linii 328 pliku OpenCIKernels.cl.

7.7 Dokumentacja pliku app-python3-gtk3-cairo/Utils.py

Komponenty

- class [Utils](#)
Klasa [Utils](#).

Przestrzenie nazw

- [Utils](#)

7.8 Dokumentacja pliku app-python3-gtk3-cairo/Vert.py

Komponenty

- class [HBNode](#)
Klasa [HBNode](#).
- class [Node](#)
Klasa [Node](#).
- class [Vert](#)
Klasa abstrakcyjna [Vert](#).

Przestrzenie nazw

- [Vert](#)

7.9 Dokumentacja pliku app-python3-gtk3-cairo/VertArranger.py

Komponenty

- class [VertArranger](#)
Klasa [VertArranger](#).

Przestrzenie nazw

- [VertArranger](#)

Skorowidz

- `__init__`
 - `GuiWindow::EditorDialog`, 10
 - `GuiWindow::EntryDialog`, 13
 - `GuiWindow::GuiWindow`, 18
 - `GuiWindow::TextResultDialog`, 44
 - `GuiWindow::TreeViewResultDialog`, 46
 - `HgMatrixAnalyzer::HgMatrixAnalyzer`, 47
 - `HyperGraph::HyperGraph`, 53
 - `Utils::Utils`, 100
 - `Vert::HBNode`, 106
 - `Vert::Node`, 110
 - `Vert::Vert`, 115
- `__repr__`
 - `HyperGraph::HyperGraph`, 53
 - `Vert::Vert`, 115
- ANIMATE
 - `GuiWindow::GuiWindow`, 40
- `acceleration_vec`
 - `Vert::Vert`, 130
- `activate`
 - `Vert::Vert`, 115
- `activate_xnode_by_id`
 - `HyperGraph::HyperGraph`, 53
- `activated`
 - `Vert::Vert`, 130
- `activated_id_set`
 - `HyperGraph::HyperGraph`, 95
- `active_hg`
 - `GuiWindow::GuiWindow`, 40
- `add_force`
 - `Vert::Vert`, 115
- `add_hbnode`
 - `HyperGraph::HyperGraph`, 53
- `add_hyperbranch`
 - `HyperGraph::HyperGraph`, 54
- `add_node`
 - `GuiWindow::GuiWindow`, 18
 - `HyperGraph::HyperGraph`, 55
- `add_property_to_hbnodes`
 - `GuiWindow::GuiWindow`, 19
- `add_property_to_nodes`
 - `GuiWindow::GuiWindow`, 19
- `add_property_to_xnodes`
 - `GuiWindow::GuiWindow`, 19
- `after_modify_action_handler`
 - `GuiWindow::GuiWindow`, 20
- `app-python3-gtk3-cairo/GuiWindow.py`, 136
- `app-python3-gtk3-cairo/HgMatrixAnalyzer.py`, 137
- `app-python3-gtk3-cairo/HyperGraph.py`, 137
- `app-python3-gtk3-cairo/OCL.py`, 137
- `app-python3-gtk3-cairo/OpenCikernels.cl`, 138
- `app-python3-gtk3-cairo/Utils.py`, 139
- `app-python3-gtk3-cairo/Vert.py`, 140
- `app-python3-gtk3-cairo/VertArranger.py`, 140
- `app-python3-gtk3-cairo/application-main.py`, 136
- `application-main`, 4
 - `pa`, 5
- `application_main`, 5
- `apply_drag_force`
 - `VertArranger::VertArranger`, 134
- `arrange_all`
 - `VertArranger::VertArranger`, 134
- `arrange_pairs_list`
 - `VertArranger::VertArranger`, 134
- `before_modify_action_handler`
 - `GuiWindow::GuiWindow`, 20
- `bin_macierz_incydencji`
 - `HyperGraph::HyperGraph`, 55
- `bin_macierz_przejsc`
 - `HyperGraph::HyperGraph`, 56
- `bin_macierz_przyleglosci_galezi`
 - `HyperGraph::HyperGraph`, 57
- `bin_macierz_przyleglosci_wierzcholekow`
 - `HyperGraph::HyperGraph`, 57
- `binarize_narray`
 - `Utils::Utils`, 100
- CANVAS_PAN_VECTOR
 - `GuiWindow::GuiWindow`, 40
- `code_textbuffer`
 - `GuiWindow::EditorDialog`, 11
- `custom_clickables`
 - `GuiWindow::GuiWindow`, 40
- `custom_drawables`
 - `GuiWindow::GuiWindow`, 41
- DEBUG_MODE
 - `GuiWindow::GuiWindow`, 41
- DEFAULT_ENABLE_OPENCL
 - OCL, 7
- DRAWING_AREA_CENTER
 - `GuiWindow::GuiWindow`, 41
- DRAWING_AREA_SIZE
 - `GuiWindow::GuiWindow`, 41
- `deactivate`
 - `Vert::Vert`, 116
- `deactivate_all`
 - `GuiWindow::GuiWindow`, 20
 - `HyperGraph::HyperGraph`, 57
- `deactivate_xnode_by_id`
 - `HyperGraph::HyperGraph`, 58
- `default_properties`
 - `Vert::Vert`, 131
- `delete_hyperbranch_by_id`
 - `HyperGraph::HyperGraph`, 59
- `delete_node_by_id`
 - `HyperGraph::HyperGraph`, 59
- `delete_selected_xnode`
 - `GuiWindow::GuiWindow`, 20

delete_xnode_by_id
 HyperGraph::HyperGraph, 60
 deselect
 Vert::Vert, 116
 deselect_xnode
 GuiWindow::GuiWindow, 21
 deselect_xnode_by_id
 HyperGraph::HyperGraph, 61
 dialog
 GuiWindow::GuiWindow, 41
 dist_s
 OpenCLKernels.cl, 138
 distance_norm_from
 Vert::Vert, 116
 distance_vect_from
 Vert::Vert, 117
 documenting_dummy_fun
 GuiWindow::GuiWindow, 21
 HyperGraph::HyperGraph, 61
 draw
 HyperGraph::HyperGraph, 61
 Vert::Vert, 117
 draw_canvas_cross
 GuiWindow::GuiWindow, 21
 draw_canvas_grid
 GuiWindow::GuiWindow, 22
 draw_center_cross
 GuiWindow::GuiWindow, 23
 draw_cross
 GuiWindow::GuiWindow, 23
 draw_edge
 HyperGraph::HyperGraph, 62
 dump_hg_as_dict
 HyperGraph::HyperGraph, 63
 dump_hg_as_tuple
 HyperGraph::HyperGraph, 63

 ENABLE_OPENCL
 OCL::OCL, 99
 edit_all_selected_xnodes
 GuiWindow::GuiWindow, 24
 edit_selected_xnode
 GuiWindow::GuiWindow, 24
 EditorDialog, 8
 entries
 GuiWindow::EntryDialog, 14
 EntryDialog, 12
 evolution_history
 HyperGraph::HyperGraph, 95
 evolution_view_current_frame
 HyperGraph::HyperGraph, 95
 example_result_generator
 HgMatrixAnalyzer::HgMatrixAnalyzer, 47
 execcode
 GuiWindow::EditorDialog, 10
 expose
 GuiWindow::GuiWindow, 24

 force_vec

 Vert::Vert, 131
 frame
 GuiWindow::GuiWindow, 41

 generuj_hipergraf
 HgMatrixAnalyzer::HgMatrixAnalyzer, 47
 get_acceleration
 Vert::Vert, 118
 get_activated_xnodes_id
 HyperGraph::HyperGraph, 64
 get_all_connected_nodes_id
 HyperGraph::HyperGraph, 64
 get_all_connected_nodes_id_by_node_id
 HyperGraph::HyperGraph, 64
 get_all_connected_xnodes_id_by_xnode_id
 HyperGraph::HyperGraph, 65
 get_all_hyperbranches_id
 HyperGraph::HyperGraph, 66
 get_all_hyperbranches_id_by_node_id
 HyperGraph::HyperGraph, 66
 get_all_nodes_id
 HyperGraph::HyperGraph, 67
 get_all_nodes_id_by_hyperbranch_id
 HyperGraph::HyperGraph, 68
 get_all_nodes_id_where_node_id_is_reachable
 HyperGraph::HyperGraph, 69
 get_all_reachable_nodes_id_from_node_id
 HyperGraph::HyperGraph, 69
 get_all_shared_nodes_id_in_hyperbranches_list
 HyperGraph::HyperGraph, 70
 get_all_unconnected_nodes_id
 HyperGraph::HyperGraph, 71
 get_all_xnodes_id
 HyperGraph::HyperGraph, 71
 get_closest_xnode_id_by_xnode_id
 HyperGraph::HyperGraph, 72
 get_colliding_xnode_id_by_position
 HyperGraph::HyperGraph, 73
 get_connections_between_nodes_by_nodes_id
 HyperGraph::HyperGraph, 74
 get_drag_force
 VertArranger::VertArranger, 135
 get_gravity_force
 VertArranger::VertArranger, 135
 get_hbnode_by_id
 HyperGraph::HyperGraph, 74
 get_hyperbranch_by_id
 HyperGraph::HyperGraph, 75
 get_hyperbranch_type
 Vert::HBNode, 106
 get_id
 Vert::Vert, 118
 get_mass
 Vert::Vert, 119
 get_max_xnode_index
 HyperGraph::HyperGraph, 76
 get_n_min_elements_indices_for_every_element
 OCL::OCL, 98
 get_name

- Vert::Vert, 119
- get_node_by_id
 - HyperGraph::HyperGraph, 76
- get_node_degree_by_node_id
 - HyperGraph::HyperGraph, 77
- get_position
 - Vert::Vert, 120
- get_properties_dict
 - Vert::Vert, 120
- get_property_value
 - Vert::Vert, 121
- get_radius
 - Vert::Vert, 121
- get_result_dict
 - GuiWindow::EntryDialog, 13
- get_selected_hbnodes_id
 - HyperGraph::HyperGraph, 78
- get_selected_nodes_id
 - HyperGraph::HyperGraph, 78
- get_selected_xnodes_id
 - HyperGraph::HyperGraph, 78
- get_smallest_n_from_list
 - Utils::Utils, 101
- get_spring_force
 - VertArranger::VertArranger, 135
- get_velocity
 - Vert::Vert, 122
- get_vert_type
 - Vert::Vert, 122
- get_xnode_by_id
 - HyperGraph::HyperGraph, 79
- get_xnode_degree_by_xnode_id
 - HyperGraph::HyperGraph, 80
- get_xnodes_center_of_mass_by_xnodes_id_list
 - HyperGraph::HyperGraph, 81
- GuiWindow, 5, 14
 - redraw_function_decorator, 5
 - undo_redo_function_decorator, 6
- GuiWindow::EditorDialog
 - __init__, 10
 - code_textbuffer, 11
 - execcode, 10
 - hgprint, 10
 - last_code_string, 11
 - last_output_string, 11
 - load_code, 10
 - parent, 11
 - result_textbuffer, 11
 - save_code, 10
- GuiWindow::EntryDialog
 - __init__, 13
 - entries, 14
 - get_result_dict, 13
 - prepare_dict_for_edit, 13
- GuiWindow::GuiWindow
 - __init__, 18
 - ANIMATE, 40
 - active_hg, 40
 - add_node, 18
 - add_property_to_hbnodes, 19
 - add_property_to_nodes, 19
 - add_property_to_xnodes, 19
 - after_modify_action_handler, 20
 - before_modify_action_handler, 20
 - CANVAS_PAN_VECTOR, 40
 - custom_clickables, 40
 - custom_drawables, 41
 - DEBUG_MODE, 41
 - DRAWING_AREA_CENTER, 41
 - DRAWING_AREA_SIZE, 41
 - deactivate_all, 20
 - delete_selected_xnode, 20
 - deselect_xnode, 21
 - dialog, 41
 - documenting_dummy_fun, 21
 - draw_canvas_cross, 21
 - draw_canvas_grid, 22
 - draw_center_cross, 23
 - draw_cross, 23
 - edit_all_selected_xnodes, 24
 - edit_selected_xnode, 24
 - expose, 24
 - frame, 41
 - init_button_panel, 25
 - init_cairo_on_drawing_surface, 25
 - init_drawing_area, 26
 - init_drawing_frame, 26
 - init_drawing_surface, 26
 - init_events_callbacks, 27
 - init_evo_scale, 27
 - init_toolbar, 27
 - key_pressed, 28
 - load_hypergraph_from_file, 29
 - MAXZOOM, 41
 - MINZOOM, 41
 - make_hyperarc, 30
 - make_hyperedge, 31
 - make_hyperloop, 31
 - make_node, 32
 - mouse_button_clicked, 32
 - mouse_moved_while_clicked, 33
 - mouse_wheel_scrolled, 33
 - old_mouse_button, 42
 - old_mouse_pos, 42
 - quit, 34
 - redo_action, 34
 - redo_states, 42
 - reset_hypergraph, 35
 - reset_view_pan_and_scale, 35
 - START_TIME_SEC, 42
 - save_hypergraph_to_file, 36
 - scale_value_changed, 36
 - select_toggle_colliding_xnode, 37
 - select_xnode, 37
 - set_active_hypergraph, 38
 - show_matrices, 38

surface, 42
 TIME_DELTA_SEC, 42
 toggle_animate, 38
 toggle_select_all, 39
 undo_action, 39
 undo_states, 42
 ZOOM, 42
 GuiWindow::TextResultDialog
 __init__, 44
 GuiWindow::TreeViewResultDialog
 __init__, 46
 HB_DUMMY
 Vert::HBNode, 106
 HB_HYPERARC
 Vert::HBNode, 106
 HB_HYPEREDGE
 Vert::HBNode, 106
 HB_HYPERLOOP
 Vert::HBNode, 107
 HBNode, 103
 hbnnode_properties
 Vert::HBNode, 107
 HgMatrixAnalyzer, 6, 46
 HgMatrixAnalyzer::HgMatrixAnalyzer
 __init__, 47
 example_result_generator, 47
 generuj_hipergraf, 47
 return_dijkstra_path_from_a_to_b, 47
 hgprint
 GuiWindow::EditorDialog, 10
 HyperGraph, 6, 48
 HyperGraph::HyperGraph
 __init__, 53
 __repr__, 53
 activate_xnode_by_id, 53
 activated_id_set, 95
 add_hbnnode, 53
 add_hyperbranch, 54
 add_node, 55
 bin_macierz_incydencji, 55
 bin_macierz_przejsc, 56
 bin_macierz_przyleglosci_galezi, 57
 bin_macierz_przyleglosci_wierzchołkow, 57
 deactivate_all, 57
 deactivate_xnode_by_id, 58
 delete_hyperbranch_by_id, 59
 delete_node_by_id, 59
 delete_xnode_by_id, 60
 deselect_xnode_by_id, 61
 documenting_dummy_fun, 61
 draw, 61
 draw_edge, 62
 dump_hg_as_dict, 63
 dump_hg_as_tuple, 63
 evolution_history, 95
 evolution_view_current_frame, 95
 get_activated_xnodes_id, 64
 get_all_connected_nodes_id, 64
 get_all_connected_nodes_id_by_node_id, 64
 get_all_connected_xnodes_id_by_xnode_id, 65
 get_all_hyperbranches_id, 66
 get_all_hyperbranches_id_by_node_id, 66
 get_all_nodes_id, 67
 get_all_nodes_id_by_hyperbranch_id, 68
 get_all_nodes_id_where_node_id_is_reachable, 69
 get_all_reachable_nodes_id_from_node_id, 69
 get_all_shared_nodes_id_in_hyperbranches_list, 70
 get_all_unconnected_nodes_id, 71
 get_all_xnodes_id, 71
 get_closest_xnode_id_by_xnode_id, 72
 get_colliding_xnode_id_by_position, 73
 get_connections_between_nodes_by_nodes_id, 74
 get_hbnnode_by_id, 74
 get_hyperbranch_by_id, 75
 get_max_xnode_index, 76
 get_node_by_id, 76
 get_node_degree_by_node_id, 77
 get_selected_hbnodes_id, 78
 get_selected_nodes_id, 78
 get_selected_xnodes_id, 78
 get_xnode_by_id, 79
 get_xnode_degree_by_xnode_id, 80
 get_xnodes_center_of_mass_by_xnodes_id_list, 81
 load_evolutionary_state, 82
 load_hg_from_dict, 82
 load_hg_from_tuple, 83
 load_next_evolutionary_state, 83
 load_previous_evolutionary_state, 83
 macierz_incydencji, 84
 macierz_incydencji_u21, 84
 macierz_osiagalnosci, 85
 macierz_przejsc, 86
 macierz_przyleglosci_galezi, 87
 macierz_przyleglosci_wierzchołkow, 87
 macierz_skalarow_odleglosci, 88
 macierz_spojnosci, 88
 macierz_wekt_kierunkowych, 89
 macierz_wekt_odleglosci, 89
 normalize_xnode_position, 90
 P, 95
 project_properties, 95
 save_evolutionary_state, 91
 select_toggle_xnode_by_id, 91
 select_xnode_by_id, 92
 selected_id_list, 95
 U, 95
 update, 92
 update_all_xnodes, 93
 update_xnode_by_id, 94
 X, 95
 hyperbranch_type
 Vert::HBNode, 107

id
 Vert::Vert, 131
 init_button_panel
 GuiWindow::GuiWindow, 25
 init_cairo_on_drawing_surface
 GuiWindow::GuiWindow, 25
 init_drawing_area
 GuiWindow::GuiWindow, 26
 init_drawing_frame
 GuiWindow::GuiWindow, 26
 init_drawing_surface
 GuiWindow::GuiWindow, 26
 init_events_callbacks
 GuiWindow::GuiWindow, 27
 init_evo_scale
 GuiWindow::GuiWindow, 27
 init_kernel
 OCL::OCL, 98
 init_toolbar
 GuiWindow::GuiWindow, 27
 is_activated
 Vert::Vert, 122
 is_colliding
 Vert::Vert, 123
 is_colliding_with_another
 Vert::Vert, 123
 is_initialized
 OCL::OCL, 99
 is_of_vert_type
 Vert::Vert, 124
 is_selected
 Vert::Vert, 124

 key_pressed
 GuiWindow::GuiWindow, 28

 last_code_string
 GuiWindow::EditorDialog, 11
 last_output_string
 GuiWindow::EditorDialog, 11
 load_code
 GuiWindow::EditorDialog, 10
 load_evolutionary_state
 HyperGraph::HyperGraph, 82
 load_hg_from_dict
 HyperGraph::HyperGraph, 82
 load_hg_from_tuple
 HyperGraph::HyperGraph, 83
 load_hypergraph_from_file
 GuiWindow::GuiWindow, 29
 load_next_evolutionary_state
 HyperGraph::HyperGraph, 83
 load_previous_evolutionary_state
 HyperGraph::HyperGraph, 83

 MAXZOOM
 GuiWindow::GuiWindow, 41
 MINZOOM
 GuiWindow::GuiWindow, 41

 macierz_incydencji
 HyperGraph::HyperGraph, 84
 macierz_incydencji_u21
 HyperGraph::HyperGraph, 84
 macierz_osiagalnosci
 HyperGraph::HyperGraph, 85
 macierz_przejsc
 HyperGraph::HyperGraph, 86
 macierz_przyleglosci_galezi
 HyperGraph::HyperGraph, 87
 macierz_przyleglosci_wierzchołkow
 HyperGraph::HyperGraph, 87
 macierz_skalarow_odleglosci
 HyperGraph::HyperGraph, 88
 macierz_spojnosci
 HyperGraph::HyperGraph, 88
 macierz_wekt_kierunkowych
 HyperGraph::HyperGraph, 89
 macierz_wekt_odleglosci
 HyperGraph::HyperGraph, 89
 make_hyperarc
 GuiWindow::GuiWindow, 30
 make_hyperedge
 GuiWindow::GuiWindow, 31
 make_hyperloop
 GuiWindow::GuiWindow, 31
 make_node
 GuiWindow::GuiWindow, 32
 map_pos_canvas_to_screen
 Utils::Utils, 101
 map_pos_list_canvas_to_screen
 Utils::Utils, 101
 map_pos_screen_to_canvas
 Utils::Utils, 101
 map_vec_canvas_to_screen
 Utils::Utils, 102
 map_vec_screen_to_canvas
 Utils::Utils, 102
 mass
 Vert::Vert, 131
 matrix_result_dict_as_string
 Utils::Utils, 102
 mouse_button_clicked
 GuiWindow::GuiWindow, 32
 mouse_moved_while_clicked
 GuiWindow::GuiWindow, 33
 mouse_wheel_scrolled
 GuiWindow::GuiWindow, 33

 name
 Vert::Vert, 131
 Node, 107
 node_properties
 Vert::Node, 110
 normalize_xnode_position
 HyperGraph::HyperGraph, 90

 OCL::OCL
 ENABLE_OPENCL, 99

- get_n_min_elements_indices_for_every_element, 98
- init_kernel, 98
- is_initialized, 99
- prog, 99
- run_with_ocl, 99
- OCL, 6, 97
 - DEFAULT_ENABLE_OPENCL, 7
- object, 96
- old_mouse_button
 - GuiWindow::GuiWindow, 42
- old_mouse_pos
 - GuiWindow::GuiWindow, 42
- OpenCLKernels.cl
 - dist_s, 138
 - openccl_kernel_a_to_p, 138
 - openccl_kernel_mat_line_sum_to_vec, 138
 - openccl_kernel_n_closest_elements_array, 138
 - openccl_kernel_scalar_dist, 138
 - openccl_kernel_vector_dir, 139
 - openccl_kernel_vector_dist, 139
 - openccl_kernel_vector_force, 139
 - openccl_kernel_vector_vector_force, 139
- openccl_kernel_a_to_p
 - OpenCLKernels.cl, 138
- openccl_kernel_mat_line_sum_to_vec
 - OpenCLKernels.cl, 138
- openccl_kernel_n_closest_elements_array
 - OpenCLKernels.cl, 138
- openccl_kernel_scalar_dist
 - OpenCLKernels.cl, 138
- openccl_kernel_vector_dir
 - OpenCLKernels.cl, 139
- openccl_kernel_vector_dist
 - OpenCLKernels.cl, 139
- openccl_kernel_vector_force
 - OpenCLKernels.cl, 139
- openccl_kernel_vector_vector_force
 - OpenCLKernels.cl, 139
- P
 - HyperGraph::HyperGraph, 95
- pa
 - application-main, 5
- parent
 - GuiWindow::EditorDialog, 11
- position_vec
 - Vert::Vert, 131
- prepare_dict_for_edit
 - GuiWindow::EntryDialog, 13
- prog
 - OCL::OCL, 99
- project_properties
 - HyperGraph::HyperGraph, 95
- properties_dict
 - Vert::Vert, 132
- quit
 - GuiWindow::GuiWindow, 34
- radius
 - Vert::Vert, 132
- redo_action
 - GuiWindow::GuiWindow, 34
- redo_states
 - GuiWindow::GuiWindow, 42
- redraw_function_decorator
 - GuiWindow, 5
- reset_hypergraph
 - GuiWindow::GuiWindow, 35
- reset_view_pan_and_scale
 - GuiWindow::GuiWindow, 35
- result_textbuffer
 - GuiWindow::EditorDialog, 11
- return_dijkstra_path_from_a_to_b
 - HgMatrixAnalyzer::HgMatrixAnalyzer, 47
- run_with_ocl
 - OCL::OCL, 99
- START_TIME_SEC
 - GuiWindow::GuiWindow, 42
- save_code
 - GuiWindow::EditorDialog, 10
- save_evolutionary_state
 - HyperGraph::HyperGraph, 91
- save_hypergraph_to_file
 - GuiWindow::GuiWindow, 36
- scale_value_changed
 - GuiWindow::GuiWindow, 36
- select
 - Vert::Vert, 125
- select_toggle
 - Vert::Vert, 125
- select_toggle_colliding_xnode
 - GuiWindow::GuiWindow, 37
- select_toggle_xnode_by_id
 - HyperGraph::HyperGraph, 91
- select_xnode
 - GuiWindow::GuiWindow, 37
- select_xnode_by_id
 - HyperGraph::HyperGraph, 92
- selected
 - Vert::Vert, 132
- selected_id_list
 - HyperGraph::HyperGraph, 95
- set_acceleration
 - Vert::Vert, 126
- set_active_hypergraph
 - GuiWindow::GuiWindow, 38
- set_mass
 - Vert::Vert, 126
- set_mass_from_degree
 - Vert::Vert, 126
- set_name
 - Vert::Vert, 127
- set_position
 - Vert::Vert, 127
- set_property_value
 - Vert::Vert, 127

set_rad_and_mass_from_degree
 Vert::Vert, 127
 set_radius
 Vert::Vert, 128
 set_radius_from_degree
 Vert::Vert, 128
 set_velocity
 Vert::Vert, 129
 show_matrices
 GuiWindow::GuiWindow, 38
 stop_movement
 Vert::Vert, 129
 suma_szeregu_geometrycznego_npararray
 Utils::Utils, 103
 surface
 GuiWindow::GuiWindow, 42

 T_DUMMY
 Vert::Vert, 132
 T_HBRANCH
 Vert::Vert, 132
 T_NODE
 Vert::Vert, 132
 TIME_DELTA_SEC
 GuiWindow::GuiWindow, 42
 TextResultDialog, 43
 toggle_animate
 GuiWindow::GuiWindow, 38
 toggle_select_all
 GuiWindow::GuiWindow, 39
 translate_by_vec
 Vert::Vert, 129
 TreeViewResultDialog, 45

 U
 HyperGraph::HyperGraph, 95
 undo_action
 GuiWindow::GuiWindow, 39
 undo_redo_function_decorator
 GuiWindow, 6
 undo_states
 GuiWindow::GuiWindow, 42
 update
 HyperGraph::HyperGraph, 92
 Vert::Vert, 129
 update_all_xnodes
 HyperGraph::HyperGraph, 93
 update_properties
 Vert::Vert, 130
 update_xnode_by_id
 HyperGraph::HyperGraph, 94
 Utils, 7, 99
 Utils::Utils
 __init__, 100
 binarize_npararray, 100
 get_smallest_n_from_list, 101
 map_pos_canvas_to_screen, 101
 map_pos_list_canvas_to_screen, 101
 map_pos_screen_to_canvas, 101
 map_vec_canvas_to_screen, 102
 map_vec_screen_to_canvas, 102
 matrix_result_dict_as_string, 102
 suma_szeregu_geometrycznego_npararray, 103

 velocity_vec
 Vert::Vert, 132
 Vert, 7, 110
 Vert::HBNode
 __init__, 106
 get_hyperbranch_type, 106
 HB_DUMMY, 106
 HB_HYPERARC, 106
 HB_HYPEREDGE, 106
 HB_HYPERLOOP, 107
 hbnnode_properties, 107
 hyperbranch_type, 107
 Vert::Node
 __init__, 110
 node_properties, 110
 Vert::Vert
 __init__, 115
 __repr__, 115
 acceleration_vec, 130
 activate, 115
 activated, 130
 add_force, 115
 deactivate, 116
 default_properties, 131
 deselect, 116
 distance_norm_from, 116
 distance_vect_from, 117
 draw, 117
 force_vec, 131
 get_acceleration, 118
 get_id, 118
 get_mass, 119
 get_name, 119
 get_position, 120
 get_properties_dict, 120
 get_property_value, 121
 get_radius, 121
 get_velocity, 122
 get_vert_type, 122
 id, 131
 is_activated, 122
 is_colliding, 123
 is_colliding_with_another, 123
 is_of_vert_type, 124
 is_selected, 124
 mass, 131
 name, 131
 position_vec, 131
 properties_dict, 132
 radius, 132
 select, 125
 select_toggle, 125
 selected, 132
 set_acceleration, 126

- set_mass, [126](#)
- set_mass_from_degree, [126](#)
- set_name, [127](#)
- set_position, [127](#)
- set_property_value, [127](#)
- set_rad_and_mass_from_degree, [127](#)
- set_radius, [128](#)
- set_radius_from_degree, [128](#)
- set_velocity, [129](#)
- stop_movement, [129](#)
- T_DUMMY, [132](#)
- T_HBRANCH, [132](#)
- T_NODE, [132](#)
- translate_by_vec, [129](#)
- update, [129](#)
- update_properties, [130](#)
- velocity_vec, [132](#)
- vert_number, [132](#)
- vert_type, [132](#)
- vert_number
 - Vert::Vert, [132](#)
- vert_type
 - Vert::Vert, [132](#)
- VertArranger, [7](#), [133](#)
- VertArranger::VertArranger
 - apply_drag_force, [134](#)
 - arrange_all, [134](#)
 - arrange_pairs_list, [134](#)
 - get_drag_force, [135](#)
 - get_gravity_force, [135](#)
 - get_spring_force, [135](#)
- X
 - HyperGraph::HyperGraph, [95](#)
- ZOOM
 - GuiWindow::GuiWindow, [42](#)