

Zwierzaki Podręcznik
2.0

Wygenerowano przez Doxygen 1.5.2

Tue Jan 22 12:58:54 2008

Spis treści

1 Zwierzaki Struktura katalogów	1
1.1 Zwierzaki Katalogi	1
2 Zwierzaki Indeks przestrzeni nazw	3
2.1 Zwierzaki Lista przestrzeni nazw	3
3 Zwierzaki Indeks hierarchiczny	5
3.1 Zwierzaki Hierarchia klas	5
4 Zwierzaki Indeks klas	7
4.1 Zwierzaki Lista klas	7
5 Zwierzaki Indeks plików	9
5.1 Zwierzaki Lista plików	9
6 Zwierzaki Dokumentacja katalogów	11
6.1 Dokumentacja katalogu inc/	11
6.2 Dokumentacja katalogu src/	13
7 Zwierzaki Dokumentacja przestrzeni nazw	15
7.1 Dokumentacja przestrzeni nazw std	15
8 Zwierzaki Dokumentacja klas	17
8.1 Dokumentacja klasy Animal	17
8.2 Dokumentacja klasy Animal_	21
8.3 Dokumentacja klasy AnimalParam	28
8.4 Dokumentacja klasy B_AvoidObstacles	32
8.5 Dokumentacja klasy B_BasicFlee	34
8.6 Dokumentacja klasy B_Cohesion	36
8.7 Dokumentacja klasy B_Encirclement	38
8.8 Dokumentacja klasy B_EncirclementLux	40

8.9	Dokumentacja klasy B_FlightMulti	42
8.10	Dokumentacja klasy B_Separation	44
8.11	Dokumentacja klasy B_Tommy	46
8.12	Dokumentacja klasy B_TommyLux	48
8.13	Dokumentacja klasy B_Wander	50
8.14	Dokumentacja klasy Behavior	52
8.15	Dokumentacja struktury Event	55
8.16	Dokumentacja klasy GUI_	56
8.17	Dokumentacja klasy GUI_Animal	57
8.18	Dokumentacja klasy GUI_Authors	59
8.19	Dokumentacja klasy GUI_Editor	60
8.20	Dokumentacja klasy GUI_Game	63
8.21	Dokumentacja klasy GUI_Help	64
8.22	Dokumentacja klasy GUI_Obstacle	65
8.23	Dokumentacja klasy GUI_Scene	66
8.24	Dokumentacja klasy History	70
8.25	Dokumentacja klasy History_	73
8.26	Dokumentacja klasy HistoryStruct	75
8.27	Dokumentacja klasy MainWindow	76
8.28	Dokumentacja klasy noGUI	78
8.29	Dokumentacja klasy Obstacle	79
8.30	Dokumentacja klasy Obstacle_Circle	82
8.31	Dokumentacja klasy Obstacles	84
8.32	Dokumentacja klasy SceneConfig	85
8.33	Dokumentacja klasy SceneManager	92
8.34	Dokumentacja klasy Vector2D	95
9	Zwierzaki Dokumentacja plików	99
9.1	Dokumentacja pliku inc/animal.hpp	99
9.2	Dokumentacja pliku inc/animal_.hpp	101
9.3	Dokumentacja pliku inc/animalParam.hpp	102
9.4	Dokumentacja pliku inc/b_AvoidObstacles.hpp	103
9.5	Dokumentacja pliku inc/b_BasicFlee.hpp	104
9.6	Dokumentacja pliku inc/b_Cohesion.hpp	105
9.7	Dokumentacja pliku inc/b_Encirclement.hpp	106
9.8	Dokumentacja pliku inc/b_EncirclementLux.hpp	107
9.9	Dokumentacja pliku inc/b_FlightMulti.hpp	108

9.10	Dokumentacja pliku inc/b_Separation.hpp	109
9.11	Dokumentacja pliku inc/b_Tommy.hpp	110
9.12	Dokumentacja pliku inc/b_TommyLux.hpp	111
9.13	Dokumentacja pliku inc/b_Wander.hpp	112
9.14	Dokumentacja pliku inc/behavior.hpp	113
9.15	Dokumentacja pliku inc/event.hpp	114
9.16	Dokumentacja pliku inc/gui_.hpp	115
9.17	Dokumentacja pliku inc/gui_animal.hpp	116
9.18	Dokumentacja pliku inc/gui_authors.hpp	117
9.19	Dokumentacja pliku inc/gui_editor.hpp	118
9.20	Dokumentacja pliku inc/gui_game.hpp	119
9.21	Dokumentacja pliku inc/gui_help.hpp	120
9.22	Dokumentacja pliku inc/gui_mainWindow.hpp	121
9.23	Dokumentacja pliku inc/gui_obstacle.hpp	122
9.24	Dokumentacja pliku inc/gui_scene.hpp	123
9.25	Dokumentacja pliku inc/history.hpp	125
9.26	Dokumentacja pliku inc/history_.hpp	126
9.27	Dokumentacja pliku inc/noGUI.hpp	127
9.28	Dokumentacja pliku inc/obs_circle.hpp	128
9.29	Dokumentacja pliku inc/obstacle.hpp	129
9.30	Dokumentacja pliku inc/obstacles.hpp	130
9.31	Dokumentacja pliku inc/sceneConfig.hpp	131
9.32	Dokumentacja pliku inc/sceneManager.hpp	132
9.33	Dokumentacja pliku inc/vector2D.hpp	133
9.34	Dokumentacja pliku src/animal.cpp	134
9.35	Dokumentacja pliku src/animal_.cpp	135
9.36	Dokumentacja pliku src/animalParam.cpp	136
9.37	Dokumentacja pliku src/b_AvoidObstacles.cpp	137
9.38	Dokumentacja pliku src/b_BasicFlee.cpp	138
9.39	Dokumentacja pliku src/b_Cohesion.cpp	139
9.40	Dokumentacja pliku src/b_Encirclement.cpp	140
9.41	Dokumentacja pliku src/b_EncirclementLux.cpp	141
9.42	Dokumentacja pliku src/b_FlightMulti.cpp	142
9.43	Dokumentacja pliku src/b_Separation.cpp	143
9.44	Dokumentacja pliku src/b_Tommy.cpp	144
9.45	Dokumentacja pliku src/b_TommyLux.cpp	145

9.46	Dokumentacja pliku src/b_Wander.cpp	146
9.47	Dokumentacja pliku src/behavior.cpp	147
9.48	Dokumentacja pliku src/gui_.cpp	148
9.49	Dokumentacja pliku src/gui_animal.cpp	149
9.50	Dokumentacja pliku src/gui_authors.cpp	150
9.51	Dokumentacja pliku src/gui_editor.cpp	151
9.52	Dokumentacja pliku src/gui_game.cpp	152
9.53	Dokumentacja pliku src/gui_help.cpp	153
9.54	Dokumentacja pliku src/gui_mainWindow.cpp	154
9.55	Dokumentacja pliku src/gui_obstacle.cpp	155
9.56	Dokumentacja pliku src/gui_scene.cpp	156
9.57	Dokumentacja pliku src/history.cpp	157
9.58	Dokumentacja pliku src/history_.cpp	158
9.59	Dokumentacja pliku src/main.cpp	159
9.60	Dokumentacja pliku src/noGUI.cpp	160
9.61	Dokumentacja pliku src/obs_circle.cpp	161
9.62	Dokumentacja pliku src/obstacle.cpp	162
9.63	Dokumentacja pliku src/obstacles.cpp	163
9.64	Dokumentacja pliku src/sceneConfig.cpp	164
9.65	Dokumentacja pliku src/sceneManager.cpp	165
9.66	Dokumentacja pliku src/vector2D.cpp	166

Rozdział 1

Zwierzaki Struktura katalogów

1.1 Zwierzaki Katalogi

Ta struktura katalogów jest posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

inc	11
src	13

Rozdział 2

Zwierzaki Indeks przestrzeni nazw

2.1 Zwierzaki Lista przestrzeni nazw

Tutaj znajdują się wszystkie przestrzenie nazw wraz z ich krótkimi opisami:

<code>std</code>	15
------------------------	----

Rozdział 3

Zwierzaki Indeks hierarchiczny

3.1 Zwierzaki Hierarchia klas

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

Animal_	21
Animal	17
AnimalParam	28
Behavior	52
B_AvoidObstacles	32
B_BasicFlee	34
B_Cohesion	36
B_Encirclement	38
B_EncirclementLux	40
B_FlightMulti	42
B_Separation	44
B_Tommy	46
B_TommyLux	48
B_Wander	50
Event	55
GUI_	56
GUI_Authors	59
GUI_Editor	60
GUI_Game	63
GUI_Help	64
GUI_Animal	57
GUI_Obstacle	65
GUI_Scene	66
History	70
History_	73
HistoryStruct	75
MainWindow	76
noGUI	78
Obstacle	79
Obstacle_Circle	82
Obstacles	84
SceneConfig	85

SceneManager	92
Vector2D	95

Rozdział 4

Zwierzaki Indeks klas

4.1 Zwierzaki Lista klas

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

Animal	17
Animal_	21
AnimalParam	28
B_AvoidObstacles	32
B_BasicFlee	34
B_Cohesion	36
B_Encirclement	38
B_EncirclementLux	40
B_FlightMulti	42
B_Separation	44
B_Tommy	46
B_TommyLux	48
B_Wander	50
Behavior	52
Event	55
GUI	56
GUI_Animal	57
GUI_Authors	59
GUI_Editor	60
GUI_Game	63
GUI_Help	64
GUI_Obstacle	65
GUI_Scene	66
History	70
History_	73
HistoryStruct	75
MainWindow	76
noGUI	78
Obstacle	79
Obstacle_Circle	82
Obstacles	84
SceneConfig	85
SceneManager	92

Vector2D	95
----------------	----

Rozdział 5

Zwierzaki Indeks plików

5.1 Zwierzaki Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

inc/animal.hpp	99
inc/animal_.hpp	101
inc/animalParam.hpp	102
inc/b_AvoidObstacles.hpp	103
inc/b_BasicFlee.hpp	104
inc/b_Cohesion.hpp	105
inc/b_Encirclement.hpp	106
inc/b_EncirclementLux.hpp	107
inc/b_FlightMulti.hpp	108
inc/b_Separation.hpp	109
inc/b_Tommy.hpp	110
inc/b_TommyLux.hpp	111
inc/b_Wander.hpp	112
inc/behavior.hpp	113
inc/event.hpp	114
inc/gui_.hpp	115
inc/gui_animal.hpp	116
inc/gui_authors.hpp	117
inc/gui_editor.hpp	118
inc/gui_game.hpp	119
inc/gui_help.hpp	120
inc/gui_mainWindow.hpp	121
inc/gui_obstacle.hpp	122
inc/gui_scene.hpp	123
inc/history.hpp	125
inc/history_.hpp	126
inc/noGUI.hpp	127
inc/obs_circle.hpp	128
inc/obstacle.hpp	129
inc/obstacles.hpp	130
inc/sceneConfig.hpp	131
inc/sceneManager.hpp	132
inc/vector2D.hpp	133

src/animal.cpp	134
src/animal_.cpp	135
src/animalParam.cpp	136
src/b_AvoidObstacles.cpp	137
src/b_BasicFlee.cpp	138
src/b_Cohesion.cpp	139
src/b_Encirclement.cpp	140
src/b_EncirclementLux.cpp	141
src/b_FlightMulti.cpp	142
src/b_Separation.cpp	143
src/b_Tommy.cpp	144
src/b_TommyLux.cpp	145
src/b_Wander.cpp	146
src/behavior.cpp	147
src/gui_.cpp	148
src/gui_animal.cpp	149
src/gui_authors.cpp	150
src/gui_editor.cpp	151
src/gui_game.cpp	152
src/gui_help.cpp	153
src/gui_mainWindow.cpp	154
src/gui_obstacle.cpp	155
src/gui_scene.cpp	156
src/history.cpp	157
src/history_.cpp	158
src/main.cpp	159
src/noGUI.cpp	160
src/obs_circle.cpp	161
src/obstacle.cpp	162
src/obstacles.cpp	163
src/sceneConfig.cpp	164
src/sceneManager.cpp	165
src/vector2D.cpp	166

Rozdział 6

Zwierzaki Dokumentacja katalogów

6.1 Dokumentacja katalogu inc/

Pliki

- plik `animal.hpp`
- plik `animal_.hpp`
- plik `animalParam.hpp`
- plik `b_AvoidObstacles.hpp`
- plik `b_BasicFlee.hpp`
- plik `b_Cohesion.hpp`
- plik `b_Encirclement.hpp`
- plik `b_EncirclementLux.hpp`
- plik `b_FlightMulti.hpp`
- plik `b_Separation.hpp`
- plik `b_Tommy.hpp`
- plik `b_TommyLux.hpp`
- plik `b_Wander.hpp`
- plik `behavior.hpp`
- plik `event.hpp`
- plik `gui_.hpp`
- plik `gui_animal.hpp`
- plik `gui_authors.hpp`
- plik `gui_editor.hpp`
- plik `gui_game.hpp`
- plik `gui_help.hpp`
- plik `gui_mainWindow.hpp`
- plik `gui_obstacle.hpp`
- plik `gui_scene.hpp`
- plik `history.hpp`
- plik `history_.hpp`
- plik `noGUI.hpp`
- plik `obs_circle.hpp`
- plik `obstacle.hpp`
- plik `obstacles.hpp`

- plik `sceneConfig.hpp`
- plik `sceneManager.hpp`
- plik `vector2D.hpp`

6.2 Dokumentacja katalogu src/

Pliki

- plik **animal.cpp**
- plik **animal_.cpp**
- plik **animalParam.cpp**
- plik **b_AvoidObstacles.cpp**
- plik **b_BasicFlee.cpp**
- plik **b_Cohesion.cpp**
- plik **b_Encirclement.cpp**
- plik **b_EncirclementLux.cpp**
- plik **b_FlightMulti.cpp**
- plik **b_Separation.cpp**
- plik **b_Tommy.cpp**
- plik **b_TommyLux.cpp**
- plik **b_Wander.cpp**
- plik **behavior.cpp**
- plik **gui_.cpp**
- plik **gui_animal.cpp**
- plik **gui_authors.cpp**
- plik **gui_editor.cpp**
- plik **gui_game.cpp**
- plik **gui_help.cpp**
- plik **gui_mainWindow.cpp**
- plik **gui_obstacle.cpp**
- plik **gui_scene.cpp**
- plik **history.cpp**
- plik **history_.cpp**
- plik **main.cpp**
- plik **noGUI.cpp**
- plik **obs_circle.cpp**
- plik **obstacle.cpp**
- plik **obstacles.cpp**
- plik **sceneConfig.cpp**
- plik **sceneManager.cpp**
- plik **vector2D.cpp**

Rozdział 7

Zwierzaki Dokumentacja przestrzeni nazw

7.1 Dokumentacja przestrzeni nazw std

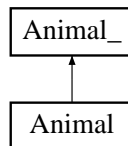
Rozdział 8

Zwierzaki Dokumentacja klas

8.1 Dokumentacja klasy Animal

```
#include <animal.hpp>
```

Diagram dziedziczenia dla Animal:



Metody publiczne

- **Animal** (unsigned int ID)
- unsigned int **getID** () const
- unsigned int **getLiderID** () const
- **Vector2D** **go** (int t)
- void **reaction** (list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- void **setBehavior** (**Behavior** *beh)
- void **setID** (unsigned int ID)
- void **setLiderID** (unsigned int ID)

Atrybuty chronione

- **Behavior** * **_behavior**
- unsigned int **_ID**
- unsigned int **_LiderID**
- int **_Order**
- unsigned int **_OrderParam**
- unsigned int **_Target**

Przyjaciele

- ostream & **operator**<< (ostream &Strm, const **Animal** &anim)
- istream & **operator**>> (istream &Strm, **Animal** &anim)

8.1.1 Dokumentacja konstruktora i destruktora

8.1.1.1 **Animal::Animal** (unsigned int *ID*)

Konstruktor. Ustawia ID (

Zobacz również:

Animal _ (str. 21)). Ustawia stan na 'zywy' (_state). Ustawia **_behavior** (str. 20) na 0.

Parametry:

ID - unikalny identyfikator zwierzenia

8.1.2 Dokumentacja funkcji składowych

8.1.2.1 unsigned int **Animal::getID** () const

Zwraca ID.

Zwraca:

ID zwierzaka

Zobacz również:

_ID (str. 20)

8.1.2.2 unsigned int **Animal::getLiderID** () const

Pobiera ID Lidera

Zwraca:

ID Lidera

Zobacz również:

_LiderID (str. 20)

8.1.2.3 **Vector2D Animal::go** (int *t*)

Przesuwa zwierzaka wg biezacych parametrow w zadanym czasie t. Dbaj, aby nowy stan nie przekroczył wartosci granicznych.

Zwierze moze sie ruszyc, tylko gdy jego energia jest wieksza od 0. Energia zwierzenia jest pomniejsza wg wzoru: $E_k = 0.5 * m * V^2$, gdzie E_k - wartosc, o ktora zostaje pomniejszona energia (w zadanym czasie t) m - masa zwierzenia V - biezaca predkosc zwierzenia

Parametry:

t - czas przesunięcia [ms]

Zwraca:

Nowe współrzędne.

ponieważ `E_k` jest tutaj bez jednostek, to możemy sobie pozwolić na potęgowanie już zeskalowanego `V`

8.1.2.4 void Animal::reaction (list< Animal_ > & family, list< Animal_ > & opponent, Obstacles & obs)

Wykonuje zachowanie (`Behavior` (str. 52)) i ustala bieżące parametry zwierzęcia (np. orientację, prędkość).

Parametry:

lista swoich

lista obcych

lista przeszkód

8.1.2.5 void Animal::setBehavior (Behavior * beh)

Ustawia wskaźnik zachowania.

Parametry:

zachowanie

8.1.2.6 void Animal::setID (unsigned int ID)

Ustala ID

8.1.2.7 void Animal::setLiderID (unsigned int ID)

Ustala ID lidera

Parametry:

IDLidera

Zobacz również:

`_LiderID` (str. 20)

8.1.3 Dokumentacja przyjaciół i funkcji związanych

8.1.3.1 `ostream& operator<< (ostream & Strm, const Animal & anim)` [friend]

8.1.3.2 `istream& operator>> (istream & Strm, Animal & anim)` [friend]

8.1.4 Dokumentacja atrybutów składowych

8.1.4.1 `Behavior* Animal::_behavior` [protected]

8.1.4.2 `unsigned int Animal::_ID` [protected]

8.1.4.3 `unsigned int Animal::_LiderID` [protected]

8.1.4.4 `int Animal::_Order` [protected]

8.1.4.5 `unsigned int Animal::_OrderParam` [protected]

8.1.4.6 `unsigned int Animal::_Target` [protected]

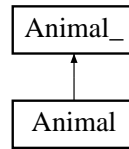
Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/animal.hpp`
- `src/animal.cpp`

8.2 Dokumentacja klasy Animal_

```
#include <animal_.hpp>
```

Diagram dziedziczenia dla Animal_:



8.2.1 Opis szczegółowy

Klasa definiująca zwierzaka (bez uwzględniania strategii).

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

Metody publiczne

- **Animal_** ()
- double **get_a_max** () const
- double **get_da_max** () const
- double **get_dE_max** () const
- double **get_dphi_max** () const
- double **get_dV_max** () const
- double **get_E_max** () const
- double **get_m** () const
- double **get_phi_max** () const
- unsigned int **get_r** () const
- int **get_t** () const
- double **get_V_max** () const
- **Animal_ getAnimal_** () const
- int **getType** () const
- void **set_a_max** (double a_max, double da_max=3)
- void **set_da_max** (double da_max=3)
- void **set_dE_max** (double dE_max=3)
- void **set_dphi_max** (double dphi_max=3)
- void **set_dV_max** (double dV_max=3)
- void **set_E_max** (double E_max, double dE_max=3)
- void **set_m** (double m)

- void **set_phi_max** (double phi_max, double dphi_max=3)
- void **set_r** (unsigned int r)
- void **set_t** (unsigned int t)
- void **set_V_max** (double V_max, double dV_max=3)
- int **setType** (int typ)

Atrybuty publiczne

- **AnimalParam _param**

Atrybuty chronione

- double **_a_max**
- double **_da_max**
- double **_dE_max**
- double **_dphi_max**
- double **_dV_max**
- double **_E_max**
- double **_m**
- string **_name**
- double **_phi_max**
- unsigned int **_r**
- unsigned int **_t**
- int **_type**
- double **_V_max**

8.2.2 Dokumentacja konstruktora i destruktora

8.2.2.1 **Animal_::Animal_ ()**

Konstruktor. Ustawia stan na 'zywy' (

Zobacz również:

AnimalParam (str. 28)).

8.2.3 Dokumentacja funkcji składowych

8.2.3.1 **double Animal_::get_a_max () const**

Pobiera przyspieszenie maksymalne.

Zwraca:

przyspieszenie maksymalne

8.2.3.2 `double Animal_::get_da_max () const`

Pobiera tolerancje przyspieszenia.

Zwraca:

tolerancja przyspieszenia

8.2.3.3 `double Animal_::get_dE_max () const`

Pobiera tolerancje energii.

Zwraca:

tolerancja energii

8.2.3.4 `double Animal_::get_dphi_max () const`

Pobiera tolerancje zwrotnosci.

Zwraca:

tolerancja zwrotnosci

8.2.3.5 `double Animal_::get_dV_max () const`

Pobiera tolerancje predkosci.

Zwraca:

tolerancja predkosci

8.2.3.6 `double Animal_::get_E_max () const`

Pobiera energie maksymalna.

Zwraca:

energia maksymalna

8.2.3.7 `double Animal_::get_m () const`

Pobiera mase

Zwraca:

masa [kg]

8.2.3.8 double Animal_::get_phi_max () const

Pobiera zwrotnosc maksymalna.

Zwraca:

zwrotnosc maksymalna

8.2.3.9 unsigned int Animal_::get_r () const

Pobiera gabaryty zwierzecia

Zwraca:

r - promien [m]

8.2.3.10 int Animal_::get_t () const

Pobiera czas reakcji

Zwraca:

t - czas reakcji [ms]

8.2.3.11 double Animal_::get_V_max () const

Pobiera predkosc maksymalna.

Zwraca:

predkosc maksymalna zwierzecia

8.2.3.12 Animal_ Animal_::getAnimal_ () const

Tworzy obiekt **Animal_** (str. 21) z biezacyimi parametrami

8.2.3.13 int Animal_::getType () const

Pobiera typ zwierzecia

8.2.3.14 void Animal_::set_a_max (double a_max, double da_max = 3)

Ustawia przyspieszenie maksymalne i tolerancje.

Parametry:

a_max - przyspieszenie maksymalne [m²/s]

da_max - tolerancja [%], domyslnie 3

8.2.3.15 `void Animal_::set_da_max (double da_max = 3)`

Ustala tolerancje przyspieszenia

Parametry:

dV_max - tolerancja [%], domyslnie 3

8.2.3.16 `void Animal_::set_dE_max (double dE_max = 3)`

Ustala tolerancje energii

Parametry:

dEmax - tolerancja [%], domyslnie 3

8.2.3.17 `void Animal_::set_dphi_max (double dphi_max = 3)`

Ustala tolerancje zwrotnosci

Parametry:

dphi_max - tolerancja [%], domyslnie 3

8.2.3.18 `void Animal_::set_dV_max (double dV_max = 3)`

Ustala tolerancje predkosci

Parametry:

dV_max - tolerancja [%], domyslnie 3

8.2.3.19 `void Animal_::set_E_max (double E_max, double dE_max = 3)`

Ustawia energie maksymalna i tolerancje.

Parametry:

E_max - energia maksymalna

dE_max - tolerancja [%], domyslnie 3

8.2.3.20 `void Animal_::set_m (double m)`

Ustala mase.

Parametry:

m - masa [kg]

8.2.3.21 void Animal_::set_phi_max (double *phi_max*, double *dphi_max* = 3)

Ustawia zwrotnosc maksymalna i tolerancje.

Parametry:

phi_max - zwrotnosc maksymalna

dphi_max - tolerancja [%], domyslnie 3

8.2.3.22 void Animal_::set_r (unsigned int *r*)

Ustala gabaryty zwierzecia zamodelowany jako okrag o promieniu *r* (

Zobacz również:

_r (str.27))

Parametry:

r - promien [m]

8.2.3.23 void Animal_::set_t (unsigned int *t*)

Ustala czas reakcji

Parametry:

t - czas reakcji [ms]

8.2.3.24 void Animal_::set_V_max (double *V_max*, double *dV_max* = 3)

Ustawia predkosc maksymalna i tolerancje.

Parametry:

V_max - predkosc maksymalna [m/s]

dV_max - tolerancja [%], domyslnie 3

8.2.3.25 int Animal_::setType (int *typ*)

Ustawia typ zwierzecia.

Parametry:

typ, *ANIMAL_TYPE_PREDATOR*-drapieznik, *ANIMAL_TYPE_ANIMAL*-ofiara
inny - ofiara

Zwraca:

przyznany typ

8.2.4 Dokumentacja atrybutów składowych

8.2.4.1 `double Animal_::_a_max` [protected]

8.2.4.2 `double Animal_::_da_max` [protected]

8.2.4.3 `double Animal_::_dE_max` [protected]

8.2.4.4 `double Animal_::_dphi_max` [protected]

8.2.4.5 `double Animal_::_dV_max` [protected]

8.2.4.6 `double Animal_::_E_max` [protected]

8.2.4.7 `double Animal_::_m` [protected]

8.2.4.8 `string Animal_::_name` [protected]

8.2.4.9 `AnimalParam Animal_::_param`

8.2.4.10 `double Animal_::_phi_max` [protected]

8.2.4.11 `unsigned int Animal_::_r` [protected]

8.2.4.12 `unsigned int Animal_::_t` [protected]

8.2.4.13 `int Animal_::_type` [protected]

8.2.4.14 `double Animal_::_V_max` [protected]

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/animal_.hpp`
- `src/animal_.cpp`

8.3 Dokumentacja klasy AnimalParam

```
#include <animalParam.hpp>
```

8.3.1 Opis szczegółowy

Klasa definiująca bieżące parametry zwierzaka (stan).

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

Metody publiczne

- **AnimalParam** ()
- double **geta** () const
- double **getDelta** () const
- double **getE** () const
- **Vector2D** **getPos** () const
- int **getState** () const
- double **getV** () const
- double **seta** (double a)
- void **setDelta** (double delta)
- void **setE** (double E)
- void **setPos** (**Vector2D** &vec)
- void **setState** (int state)
- void **setV** (double V)

Atrybuty chronione

- double **_a**
- double **_delta**
- double **_E**
- **Vector2D** **_pos**
- int **_state**
- double **_V**

8.3.2 Dokumentacja konstruktora i destruktora

8.3.2.1 AnimalParam::AnimalParam ()

Konstruktor. Ustawia wszystkie param na 0. Status jest ustawiany na ANIMAL_STATE_LIVE
(

Zobacz również:

`_state` (str. 31)).

8.3.3 Dokumentacja funkcji składowych

8.3.3.1 double AnimalParam::geta () const

Zwraca bieżące przyspieszenie.

Zwraca:

Zwraca bieżące przyspieszenie.

8.3.3.2 double AnimalParam::getDelta () const

Zwraca bieżącą orientację.

Zwraca:

Bieżąca orientacja.

8.3.3.3 double AnimalParam::getE () const

Zwraca bieżącą energię.

Zwraca:

Zwraca bieżącą energię.

8.3.3.4 Vector2D AnimalParam::getPos () const

Zwraca bieżące współrzędne.

Zwraca:

Zwraca bieżące współrzędne.

8.3.3.5 int AnimalParam::getState () const

Pobranie stanu.

Parametry:

stan wg ANIMAL_STATE

8.3.3.6 double AnimalParam::getV () const

Zwraca biezaca predkosc.

Zwraca:

Zwraca biezaca predkosc.

8.3.3.7 double AnimalParam::seta (double *a*)

Ustawia biezace przyspieszenie

Parametry:

a - nowe przyspieszenie

8.3.3.8 void AnimalParam::setDelta (double *delta*)

Ustawia biezaca orientacje.

Parametry:

delta - nowa orientacja

8.3.3.9 void AnimalParam::setE (double *E*)

Ustawia biezaca energie.

Parametry:

E - nowa energia

8.3.3.10 void AnimalParam::setPos (Vector2D & *vec*)

Ustawia biezace wspolrzedne.

Parametry:

vec - nowe wspolrzedne

8.3.3.11 void AnimalParam::setState (int *state*)

Ustawienie stanu.

Parametry:

nowy stan wg ANIMAL_STATE

8.3.3.12 `void AnimalParam::setV (double V)`

Ustawia biezaca predkosc.

Parametry:

`V` - nowa predkosc

8.3.4 Dokumentacja atrybutów składowych

8.3.4.1 `double AnimalParam::_a` [protected]

8.3.4.2 `double AnimalParam::_delta` [protected]

8.3.4.3 `double AnimalParam::_E` [protected]

8.3.4.4 `Vector2D AnimalParam::_pos` [protected]

8.3.4.5 `int AnimalParam::_state` [protected]

8.3.4.6 `double AnimalParam::_V` [protected]

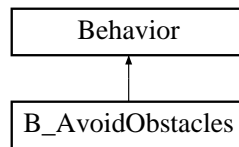
Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/animalParam.hpp`
- `src/animalParam.cpp`

8.4 Dokumentacja klasy B_AvoidObstacles

```
#include <b_AvoidObstacles.hpp>
```

Diagram dziedziczenia dla B_AvoidObstacles:



8.4.1 Opis szczegółowy

Separacja od stada.

Autor:

Arkadiusz Lisiecki (ark@data.pl)

Data:

2007-12-15

Wersja:

1.0

Metody publiczne

- **AnimalParam action** (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_AvoidObstacles** ()

8.4.2 Dokumentacja konstruktora i destruktora

8.4.2.1 B_AvoidObstacles::B_AvoidObstacles ()

Konstruktor.

8.4.3 Dokumentacja funkcji składowych

8.4.3.1 AnimalParam B_AvoidObstacles::action (**Animal_** & *anim*, list< **Animal_** > & *family*, list< **Animal_** > & *opponent*, **Obstacles** & *obs*) [virtual]

Separacja od stada. Separacja następuje uwzględniając te zwierzeta, ktore mieszczą się w ustalonym w konfiguracji promieniu.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

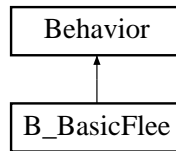
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/b_AvoidObstacles.hpp
- src/b_AvoidObstacles.cpp

8.5 Dokumentacja klasy B_BasicFlee

```
#include <b_BasicFlee.hpp>
```

Diagram dziedziczenia dla B_BasicFlee:



8.5.1 Opis szczegółowy

Prosta ucieczka. Działa tak jak Separation, tyle, że dotyczy nieprzyjaciół.

Autor:

Arkadiusz Lisiecki (ark@data.pl)

Data:

2007-12-15

Wersja:

1.0

Metody publiczne

- **AnimalParam** action (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_BasicFlee** ()

8.5.2 Dokumentacja konstruktora i destruktora

8.5.2.1 B_BasicFlee::B_BasicFlee ()

Konstruktor.

8.5.3 Dokumentacja funkcji składowych

8.5.3.1 AnimalParam B_BasicFlee::action (**Animal_** & anim, list< **Animal_** > & family, list< **Animal_** > & opponent, **Obstacles** & obs) [virtual]

Prosta ucieczka. Działa tak jak Separation, tyle, że dotyczy nieprzyjaciół.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciół (zwykle osobników tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

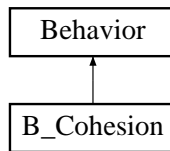
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/b_BasicFlee.hpp
- src/b_BasicFlee.cpp

8.6 Dokumentacja klasy B_Cohesion

```
#include <b_Cohesion.hpp>
```

Diagram dziedziczenia dla B_Cohesion:



8.6.1 Opis szczegółowy

Grupowanie stada. Grupowanie następuje uwzględniając te zwierzęta, które mieszczą się w ustalonym w konfiguracji promieniu.

Autor:

Arkadiusz Lisiecki (ark@data.pl)

Data:

2007-12-15

Wersja:

1.0

Metody publiczne

- **AnimalParam action** (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_Cohesion** ()

8.6.2 Dokumentacja konstruktora i destruktora

8.6.2.1 B_Cohesion::B_Cohesion ()

Konstruktor.

8.6.3 Dokumentacja funkcji składowych

8.6.3.1 AnimalParam B_Cohesion::action (**Animal_** & anim, list< **Animal_** > & family, list< **Animal_** > & opponent, **Obstacles** & obs) [virtual]

Grupowanie stada. Grupowanie następuje uwzględniając te zwierzęta, które mieszczą się w ustalonym w konfiguracji promieniu.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

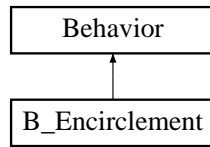
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/b_Cohesion.hpp
- src/b_Cohesion.cpp

8.7 Dokumentacja klasy B_Encirclement

```
#include <b_Encirclement.hpp>
```

Diagram dziedziczenia dla B_Encirclement:



8.7.1 Opis szczegółowy

Pogon za pierwsza ofiara (lista: wrogowie) Pr. max jest od razu osiagana. Strategie Encirclement charakteryzuje to, ze drapieznik porusza sie w kierunku polozenia ofiary.

Autor:

Arkadiusz Lisiecki (ark@data.pl)
Tomasz Kurowski (pomyslodawca metody)

Data:

2007-06-18

Wersja:

1.0

Metody publiczne

- **AnimalParam** action (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_Encirclement** ()

8.7.2 Dokumentacja konstruktora i destruktora

8.7.2.1 B_Encirclement::B_Encirclement ()

Konstruktor.

8.7.3 Dokumentacja funkcji składowych

8.7.3.1 AnimalParam B_Encirclement::action (**Animal_** & anim, list< **Animal_** > & family, list< **Animal_** > & opponent, **Obstacles** & obs) [virtual]

Pogon za pierwsza ofiara (lista: opponent) Pr. max jest od razu osiagana. Strategie Encirclement charakteryzuje to, ze drapieznik porusza sie w kierunku polozenia ofiary.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

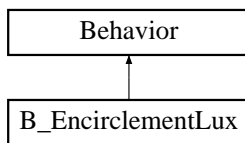
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/b_Encirclement.hpp
- src/b_Encirclement.cpp

8.8 Dokumentacja klasy B_EncirclementLux

```
#include <b_EncirclementLux.hpp>
```

Diagram dziedziczenia dla B_EncirclementLux:



8.8.1 Opis szczegółowy

Pogon za pierwsza ofiara (lista: wrogowie) Pr. max jest od razu osiagana. Strategie Encirclement charakteryzuje to, ze drapieznik porusza sie w kierunku polozenia ofiary. Wersja zmodyfikowana o dodanie do wyliczonej orientacji stalej 0.35 rad (ze znakiem +, gdy ofiara jest po lewej stronie drapiezczy ze znakiem -, gdy ofiara jest poprawej)

Autor:

Arkadiusz Lisiecki (ark@data.pl)
Tomasz Kurowski (pomyslodawca metody)

Data:

2007-06-18

Wersja:

1.0

Metody publiczne

- **AnimalParam** action (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_EncirclementLux** ()

8.8.2 Dokumentacja konstruktora i destruktora

8.8.2.1 B_EncirclementLux::B_EncirclementLux ()

Konstruktor.

8.8.3 Dokumentacja funkcji składowych

8.8.3.1 AnimalParam B_EncirclementLux::action (**Animal_** & anim, list< **Animal_** > & family, list< **Animal_** > & opponent, **Obstacles** & obs) [virtual]

Pogon za pierwsza ofiara (lista: opponent) Pr. max jest od razu osiagana. Strategie Encirclement charakteryzuje to, ze drapieznik porusza sie w kierunku polozenia ofiary. Wersja zmodyfikowana

o dodanie do wyliczonej orientacji stałej 0.35 rad (ze znakiem $+$, gdy ofiara jest po lewej stronie drapieżcy, ze znakiem $-$, gdy ofiara jest po prawej)

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciół (zwykle osobników tego samego gatunku)

lista wrogów (zwykle osobników przeciwnego gatunku)

lista przeszkód

konfiguracja sceny

Zwraca:

nowe parametry bieżące

Reimplementowana z **Behavior** (str. 53).

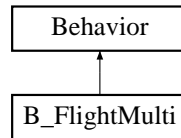
Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/b_EncirclementLux.hpp`
- `src/b_EncirclementLux.cpp`

8.9 Dokumentacja klasy B_FlightMulti

```
#include <b_FlightMulti.hpp>
```

Diagram dziedziczenia dla B_FlightMulti:



8.9.1 Opis szczegółowy

Ucieczka jak najdalej od drapieżników z kryterium hiperbolicznym. Pr. max jest od razu osiągnana.

Autor:

Arkadiusz Lisiecki (ark@data.pl)
Tomasz Kurowski (pomyslodawca metody)

Data:

2007-06-18

Wersja:

1.0

Metody publiczne

- **AnimalParam action** (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_FlightMulti** ()

8.9.2 Dokumentacja konstruktora i destruktora

8.9.2.1 B_FlightMulti::B_FlightMulti ()

Konstruktor.

8.9.3 Dokumentacja funkcji składowych

8.9.3.1 AnimalParam B_FlightMulti::action (**Animal_** & *anim*, list< **Animal_** > & *family*, list< **Animal_** > & *opponent*, **Obstacles** & *obs*) [virtual]

Ucieczka jak najdalej od drapieżników z kryterium hiperbolicznym. Pr. max jest od razu osiągnana.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

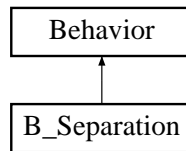
Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/b_FlightMulti.hpp`
- `src/b_FlightMulti.cpp`

8.10 Dokumentacja klasy B_Separation

```
#include <b_Separation.hpp>
```

Diagram dziedziczenia dla B_Separation:



8.10.1 Opis szczegółowy

Separacja od stada.

Autor:

Arkadiusz Lisiecki (ark@data.pl)

Data:

2007-12-15

Wersja:

1.0

Metody publiczne

- **AnimalParam action** (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_Separation** ()

8.10.2 Dokumentacja konstruktora i destruktora

8.10.2.1 B_Separation::B_Separation ()

Konstruktor.

8.10.3 Dokumentacja funkcji składowych

8.10.3.1 AnimalParam B_Separation::action (**Animal_** & *anim*, list< **Animal_** > & *family*, list< **Animal_** > & *opponent*, **Obstacles** & *obs*) [virtual]

Separacja od stada. Separacja następuje uwzględniając te zwierzęta, które mieszczą się w ustalonym w konfiguracji promieniu.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przeciwnego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

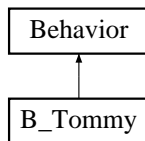
Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/b_Separation.hpp`
- `src/b_Separation.cpp`

8.11 Dokumentacja klasy B_Tommy

```
#include <b_Tommy.hpp>
```

Diagram dziedziczenia dla B_Tommy:



8.11.1 Opis szczegółowy

Pogon za pierwsza ofiara (lista: wrogowie) Pr. max jest od razu osiagana. Strategie Tommy charakteryzuje to, ze drapieznik porusza sie w kierunku punku przeciecia z ofiara.

Autor:

Arkadiusz Lisiecki (ark@data.pl)
Tomasz Kurowski (autor metody)

Data:

2007-06-18

Wersja:

1.0

Metody publiczne

- **AnimalParam** action (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_Tommy** ()

8.11.2 Dokumentacja konstruktora i destruktora

8.11.2.1 B_Tommy::B_Tommy ()

Konstruktor.

8.11.3 Dokumentacja funkcji składowych

8.11.3.1 AnimalParam B_Tommy::action (**Animal_** & anim, list< **Animal_** > & family, list< **Animal_** > & opponent, **Obstacles** & obs) [virtual]

Pogon za pierwsza ofiara (lista: opponent) Pr. max jest od razu osiagana. Strategie Tommy charakteryzuje to, ze drapieznik porusza sie w kierunku punku przeciecia z ofiara.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

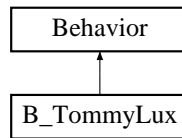
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/b_Tommy.hpp
- src/b_Tommy.cpp

8.12 Dokumentacja klasy B_TommyLux

```
#include <b_TommyLux.hpp>
```

Diagram dziedziczenia dla B_TommyLux:



8.12.1 Opis szczegółowy

Pogon za pierwsza ofiara (lista: wrogowie) Pr. max jest od razu osiagana. Strategie Tommy charakteryzuje to, ze drapieznik porusza sie w kierunku przeciecia z ofiara. Wersja zmodyfikowana o dodanie do wyliczonej orientacji stalej 0.35 (ze znakiem +, gdy ofiara jest po lewej stronie drapiezczy, ze znakiem -, gdy ofiara jest poprawej)

Autor:

Arkadiusz Lisiecki (ark@data.pl)
Tomasz Kurowski (pomyslodawca metody)

Data:

2007-06-18

Wersja:

1.0

Metody publiczne

- **AnimalParam** action (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_TommyLux** ()

8.12.2 Dokumentacja konstruktora i destruktora

8.12.2.1 B_TommyLux::B_TommyLux ()

Konstruktor.

8.12.3 Dokumentacja funkcji skladowych

8.12.3.1 AnimalParam B_TommyLux::action (**Animal_** & anim, list< **Animal_** > & family, list< **Animal_** > & opponent, **Obstacles** & obs) [virtual]

Pogon za pierwsza ofiara (lista: wrogowie) Pr. max jest od razu osiagana. Strategie Tommy charakteryzuje to, ze drapieznik porusza sie w kierunkua przeciecia z ofiara. Wersja zmodyfikowana

o dodanie do wyliczonej orientacji stałej 0.35 (ze znakiem +, gdy ofiara jest po lewej stronie drapieżcy, ze znakiem -, gdy ofiara jest po prawej)

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciół (zwykle osobników tego samego gatunku)

lista wrogów (zwykle osobników przeciwnego gatunku)

lista przeszkód

konfiguracja sceny

Zwraca:

nowe parametry bieżące

Reimplementowana z **Behavior** (str. 53).

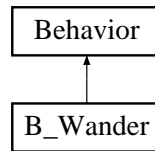
Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/b_TommyLux.hpp`
- `src/b_TommyLux.cpp`

8.13 Dokumentacja klasy B_Wander

```
#include <b_Wander.hpp>
```

Diagram dziedziczenia dla B_Wander:



8.13.1 Opis szczegółowy

Chaotyczne ruchy

Autor:

Arkadiusz Lisiecki (ark@data.pl)

Data:

2007-12-15

Wersja:

1.0

Metody publiczne

- **AnimalParam** action (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)
- **B_Wander** ()

8.13.2 Dokumentacja konstruktora i destruktora

8.13.2.1 B_Wander::B_Wander ()

Konstruktor.

8.13.3 Dokumentacja funkcji składowych

8.13.3.1 AnimalParam B_Wander::action (**Animal_** & anim, list< **Animal_** > & family, list< **Animal_** > & opponent, **Obstacles** & obs) [virtual]

Wyliczany jest losowy kierunek.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana z **Behavior** (str. 53).

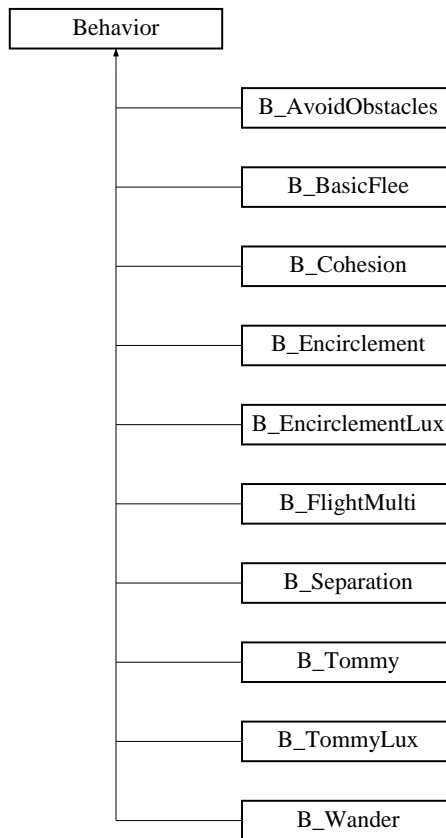
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/b_Wander.hpp
- src/b_Wander.cpp

8.14 Dokumentacja klasy Behavior

```
#include <behavior.hpp>
```

Diagram dziedziczenia dla Behavior:



8.14.1 Opis szczegółowy

Klasa definiująca zachowanie zwierzeczka, zachowanie może się odbyć przy użyciu różnych strategii zachowania w zależności od stanu zwierzeczka.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

Metody publiczne

- virtual **AnimalParam** action (**Animal_** &anim, list< **Animal_** > &family, list< **Animal_** > &opponent, **Obstacles** &obs)

- Behavior ()
- void setConfig (SceneConfig &cfg)

Atrybuty chronione

- SceneConfig _cfg
- string _description
- string _name

8.14.2 Dokumentacja konstruktora i destruktora

8.14.2.1 Behavior::Behavior ()

Konstruktor, nic nie wykonuje

8.14.3 Dokumentacja funkcji składowych

8.14.3.1 AnimalParam Behavior::action (Animal_ & anim, list< Animal_ > & family, list< Animal_ > & opponent, Obstacles & obs) [virtual]

Wykonuje akcje w zaleznosci od przyjetych strategii.

Parametry:

zwierzak,ktorego dotyczy zachowanie

lista przyjaciol (zwykle osobnikow tego samego gatunku)

lista wrogow (zwykle osobnikow przecinego gatunku)

lista przeszkod

konfiguracja sceny

Zwraca:

nowe parametry biezace

Reimplementowana w **B_AvoidObstacles** (str. 32), **B_BasicFlee** (str. 34), **B_Cohesion** (str. 36), **B_Encirclement** (str. 38), **B_EncirclementLux** (str. 40), **B_FlightMulti** (str. 42), **B_Separation** (str. 44), **B_Tommy** (str. 46), **B_TommyLux** (str. 48) i **B_Wander** (str. 50).

8.14.3.2 void Behavior::setConfig (SceneConfig & cfg)

Pobiera konfiguracje, parametry symulacji

8.14.4 Dokumentacja atrybutów składowych

8.14.4.1 `SceneConfig Behavior::_cfg` [protected]

8.14.4.2 `string Behavior::_description` [protected]

8.14.4.3 `string Behavior::_name` [protected]

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/behavior.hpp`
- `src/behavior.cpp`

8.15 Dokumentacja struktury Event

```
#include <event.hpp>
```

8.15.1 Opis szczegółowy

Struktura opisująca zdarzenie.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

Atrybuty publiczne

- `int event`
- `unsigned int ID1`
- `unsigned int ID2`
- `Vector2D pos1`
- `Vector2D pos2`
- `unsigned int t`

8.15.2 Dokumentacja atrybutów składowych

8.15.2.1 `int Event::event`

8.15.2.2 `unsigned int Event::ID1`

8.15.2.3 `unsigned int Event::ID2`

8.15.2.4 `Vector2D Event::pos1`

8.15.2.5 `Vector2D Event::pos2`

8.15.2.6 `unsigned int Event::t`

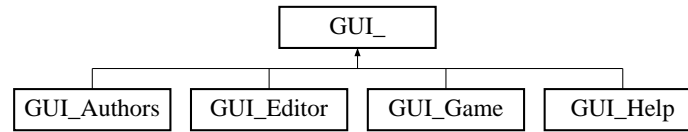
Dokumentacja dla tej struktury została wygenerowana z pliku:

- `inc/event.hpp`

8.16 Dokumentacja klasy GUI_

```
#include <gui_.hpp>
```

Diagram dziedziczenia dla GUI_:



Metody publiczne

- `GUI_ (QWidget *parent=0, Qt::WindowFlags flags=0)`

8.16.1 Dokumentacja konstruktora i destruktora

8.16.1.1 `GUI_::GUI_ (QWidget * parent = 0, Qt::WindowFlags flags = 0)`

Wymaga, by istniał `parent`, który posiada metodę `slot_closeChild(int ID)`

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/gui_.hpp`
- `src/gui_.cpp`

8.17 Dokumentacja klasy GUI_Animal

```
#include <gui_animal.hpp>
```

Metody publiczne

- **QRectF boundingRect** () const
- **Animal * getAnimal** ()
- **Animal * getAnimalConst** () const
- **Animal getStatAnimal** () const
- **GUI_Animal** (GUI_Scene *scene, Vector2D point)
- void **newPos** (Vector2D &pos)
- void **paint** (QPainter *, const QStyleOptionGraphicsItem *, QWidget *)

Atrybuty prywatne

- **Animal * _animal**
- qreal radius
- bool selected

Przyjaciele

- ostream & **operator<<** (ostream &Strm, const GUI_Animal &anim)
- istream & **operator>>** (istream &Strm, const GUI_Animal &anim)

8.17.1 Dokumentacja konstruktora i destruktora

8.17.1.1 GUI_Animal::GUI_Animal (GUI_Scene * scene, Vector2D point)

Konstruktor zwierzaka, umieszczony na scenie w danym punkcie

8.17.2 Dokumentacja funkcji składowych

8.17.2.1 QRectF GUI_Animal::boundingRect () const

Wymagane przez QGraphicsItem (metoda wirtualna)

8.17.2.2 Animal * GUI_Animal::getAnimal ()

8.17.2.3 Animal * GUI_Animal::getAnimalConst () const

8.17.2.4 Animal GUI_Animal::getStatAnimal () const

8.17.2.5 void GUI_Animal::newPos (Vector2D & pos)

Zmienia pozycje zwierzaka

8.17.2.6 void GUI_Animal::paint (QPainter *, const QStyleOptionGraphicsItem *, QWidget *)

Wymagane przez QGraphicsItem (metoda wirtualna)

8.17.3 Dokumentacja przyjaciół i funkcji związanych

8.17.3.1 ostream& operator<< (ostream & *Strm*, const GUI_Animal & *anim*)
[friend]

8.17.3.2 istream& operator>> (istream & *Strm*, const GUI_Animal & *anim*)
[friend]

8.17.4 Dokumentacja atrybutów składowych

8.17.4.1 Animal* GUI_Animal::_animal [private]

8.17.4.2 qreal GUI_Animal::radius [private]

8.17.4.3 bool GUI_Animal::selected [private]

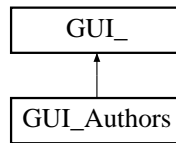
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/gui_animal.hpp
- src/gui_animal.cpp

8.18 Dokumentacja klasy GUI_Authors

```
#include <gui_authors.hpp>
```

Diagram dziedziczenia dla GUI_Authors:



Metody publiczne

- `GUI_Authors` (`QWidget *parent=0`, `Qt::WindowFlags flags=0`)

8.18.1 Dokumentacja konstruktora i destruktora

8.18.1.1 `GUI_Authors::GUI_Authors` (`QWidget *parent = 0`, `Qt::WindowFlags flags = 0`)

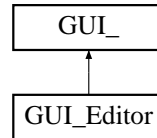
Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/gui_authors.hpp`
- `src/gui_authors.cpp`

8.19 Dokumentacja klasy GUI_Editor

```
#include <gui_editor.hpp>
```

Diagram dziedziczenia dla GUI_Editor:



Metody publiczne

- **GUI_Editor** (QWidget *parent=0, Qt::WindowFlags flags=0)

Sloty prywatne

- void **edAddAnimal** ()
- void **edAddObstacle** ()
- void **edRem** ()
- void **edSelect** ()
- void **initScene** (int W, int H)
- void **newScene** ()
- void **openScene** ()
- void **saveScene** ()
- void **setParamBox** ()

Metody prywatne

- void **setupActions** ()

Atrybuty prywatne

- QDockWidget * **_dockParamWidget**
- QAction * **addObAct**
- QAction * **addShAct**
- QToolBar * **editToolBar**
- QToolBar * **fileToolBar**
- GUI_Scene * **gScene**
- QAction * **newAct**
- QAction * **openAct**
- QAction * **remAct**
- QAction * **saveAct**
- QAction * **selAct**

8.19.1 Dokumentacja konstruktora i destruktor

8.19.1.1 GUI_Editor::GUI_Editor (QWidget * *parent* = 0, Qt::WindowFlags *flags* = 0)

8.19.2 Dokumentacja funkcji składowych

8.19.2.1 void GUI_Editor::edAddAnimal () [private, slot]

8.19.2.2 void GUI_Editor::edAddObstacle () [private, slot]

8.19.2.3 void GUI_Editor::edRem () [private, slot]

8.19.2.4 void GUI_Editor::edSelect () [private, slot]

8.19.2.5 void GUI_Editor::initScene (int *W*, int *H*) [private, slot]

Tworzy scene o podanych wymiarach

Parametry:

W - szerokosc sceny [m]

H - wysokosc sceny [m]

8.19.2.6 void GUI_Editor::newScene () [private, slot]

8.19.2.7 void GUI_Editor::openScene () [private, slot]

8.19.2.8 void GUI_Editor::saveScene () [private, slot]

8.19.2.9 void GUI_Editor::setParamBox () [private, slot]

Ustawia okienko parametrow na podstawie danych ze sceny

Zobacz również:

gScene (str. 62)

8.19.2.10 void GUI_Editor::setupActions () [private]

8.19.3 Dokumentacja atrybutów składowych

8.19.3.1 QDockWidget* GUI_Editor::_dockParamWidget [private]

8.19.3.2 QAction* GUI_Editor::addObAct [private]

8.19.3.3 QAction* GUI_Editor::addShAct [private]

8.19.3.4 QToolBar* GUI_Editor::editToolBar [private]

8.19.3.5 QToolBar* GUI_Editor::fileToolBar [private]

8.19.3.6 GUI_Scene* GUI_Editor::gScene [private]

8.19.3.7 QAction* GUI_Editor::newAct [private]

8.19.3.8 QAction* GUI_Editor::openAct [private]

8.19.3.9 QAction* GUI_Editor::remAct [private]

8.19.3.10 QAction* GUI_Editor::saveAct [private]

8.19.3.11 QAction* GUI_Editor::selAct [private]

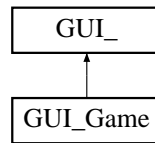
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/gui_editor.hpp
- src/gui_editor.cpp

8.20 Dokumentacja klasy GUI_Game

```
#include <gui_game.hpp>
```

Diagram dziedziczenia dla GUI_Game:



Metody publiczne

- **GUI_Game** (QWidget *parent=0, Qt::WindowFlags flags=0)

8.20.1 Dokumentacja konstruktora i destruktora

8.20.1.1 **GUI_Game::GUI_Game** (QWidget * *parent* = 0, Qt::WindowFlags *flags* = 0)

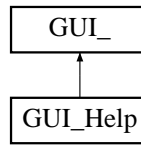
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/**gui_game.hpp**
- src/**gui_game.cpp**

8.21 Dokumentacja klasy GUI_Help

```
#include <gui_help.hpp>
```

Diagram dziedziczenia dla GUI_Help:



Metody publiczne

- GUI_Help (QWidget *parent=0, Qt::WindowFlags flags=0)

8.21.1 Dokumentacja konstruktora i destruktora

8.21.1.1 GUI_Help::GUI_Help (QWidget * *parent* = 0, Qt::WindowFlags *flags* = 0)

Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/gui_help.hpp
- src/gui_help.cpp

8.22 Dokumentacja klasy GUI_Obstacle

```
#include <gui_obstacle.hpp>
```

Metody publiczne

- GUI_Obstacle (QWidget *parent=0, Qt::WindowFlags flags=0)

8.22.1 Dokumentacja konstruktora i destruktora

8.22.1.1 GUI_Obstacle::GUI_Obstacle (QWidget * *parent* = 0, Qt::WindowFlags *flags* = 0)

Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/gui_obstacle.hpp
- src/gui_obstacle.cpp

8.23 Dokumentacja klasy GUI_Scene

```
#include <gui_scene.hpp>
```

Typy publiczne

- enum `mode` { `NONE`, `ADDOBSTACLE`, `ADDANIMAL`, `REMOVE` }

Sygnaly

- void `changed` ()

Metody publiczne

- void `addAnimal` (const `Vector2D` &origin=`Vector2D`(0, 0))
- void `addToFile` (ostream &F)
- vector< `GUI_Animal` * > `getAnimals` () const
- `GUI_Animal` * `getCurrAnimal` ()
- `QWidget` * `getParamBox` ()
- `GUI_Scene` (`QWidget` *parent=0, qreal w=500, qreal h=500)
- void `setEditMode` (`mode`)
- ~`GUI_Scene` ()

Metody chronione

- void `clearPolygon` ()
- void `createPotentialMap` ()
- void `drawBackground` (`QPainter` *, const `QRectF` &)
- bool `isSelectedAnimal` (`QPoint` point)
- virtual void `keyPressEvent` (`QKeyEvent` *)
- virtual void `mouseDoubleClickEvent` (`QMouseEvent` *)
- virtual void `mouseMoveEvent` (`QMouseEvent` *)
- virtual void `mousePressEvent` (`QMouseEvent` *)
- virtual void `mouseReleaseEvent` (`QMouseEvent` *)
- bool `selectSheep` (`QPoint`)

Atrybuty chronione

- vector< `GUI_Animal` * > `_animals`
- `mode` `_currMode`
- qreal `_height`
- vector< `Obstacle` > `_obstacle`
- qreal `_scale`
- qreal `_width`

Metody prywatne

- void `getParamBoxAnimal` (`QWidget` *widget)

Atrybuty prywatne

- GUI_Animal * _currAnimal
- int _currObstacle
- QWidget * _paramBox
- bool antiAliased
- QImage image
- QGraphicsPolygonItem * newObstacle
- bool potentialMap

Przyjaciele

- ostream & operator<< (ostream &Strm, const vector< GUI_Animal * > &anim)
- ostream & operator<< (ostream &Strm, const GUI_Scene &scene)
- istream & operator>> (istream &Strm, GUI_Scene &scene)

8.23.1 Dokumentacja składowych wyliczanych

8.23.1.1 enum GUI_Scene::mode

Wartości wyliczeń:

NONE

ADDOBSTACLE

ADDANIMAL

REMOVE

8.23.2 Dokumentacja konstruktora i destruktora

8.23.2.1 GUI_Scene::GUI_Scene (QWidget * parent = 0, qreal w = 500, qreal h = 500)

8.23.2.2 GUI_Scene::~~GUI_Scene ()

8.23.3 Dokumentacja funkcji składowych

8.23.3.1 void GUI_Scene::addAnimal (const Vector2D & origin = Vector2D(0, 0))

Dodaje zwierzaka na podane wspolrzene

8.23.3.2 void GUI_Scene::addToFile (ostream & *F*)

8.23.3.3 void GUI_Scene::changed () [signal]

8.23.3.4 void GUI_Scene::clearPolygon () [protected]

8.23.3.5 void GUI_Scene::createPotentialMap () [protected]

8.23.3.6 void GUI_Scene::drawBackground (QPainter *, const QRectF &)
[protected]

8.23.3.7 vector< GUI_Animal * > GUI_Scene::getAnimals () const

8.23.3.8 GUI_Animal * GUI_Scene::getCurrAnimal ()

8.23.3.9 QWidget * GUI_Scene::getParamBox ()

Zwraca okienko parametrow dla biezacego elementu

8.23.3.10 void GUI_Scene::getParamBoxAnimal (QWidget * *widget*) [private]

Wyswietla okienko parametrow dla biezacego zwierzaka

Zobacz również:

`_currAnimal` (str. 69)

8.23.3.11 bool GUI_Scene::isSelectedAnimal (QPoint *point*) [protected]

Czy zostal wybrany zwierzak

8.23.3.12 void GUI_Scene::keyPressEvent (QKeyEvent *) [protected, virtual]

8.23.3.13 void GUI_Scene::mouseDoubleClickEvent (QMouseEvent *)
[protected, virtual]

8.23.3.14 void GUI_Scene::mouseMoveEvent (QMouseEvent *) [protected,
virtual]

8.23.3.15 void GUI_Scene::mousePressEvent (QMouseEvent *) [protected,
virtual]

Reakcja na wcisnieciu przycisku myszy

8.23.3.16 void GUI_Scene::mouseReleaseEvent (QMouseEvent *) [protected, virtual]

8.23.3.17 bool GUI_Scene::selectSheep (QPoint) [protected]

8.23.3.18 void GUI_Scene::setEditMode (mode)

8.23.4 Dokumentacja przyjaciół i funkcji związanych

8.23.4.1 ostream& operator<< (ostream & *Strm*, const vector< GUI_Animal * > & *anim*) [friend]

8.23.4.2 ostream& operator<< (ostream & *Strm*, const GUI_Scene & *scene*) [friend]

8.23.4.3 istream& operator>> (istream & *Strm*, GUI_Scene & *scene*) [friend]

8.23.5 Dokumentacja atrybutów składowych

8.23.5.1 vector<GUI_Animal*> GUI_Scene::_animals [protected]

8.23.5.2 GUI_Animal* GUI_Scene::_currAnimal [private]

8.23.5.3 mode GUI_Scene::_currMode [protected]

8.23.5.4 int GUI_Scene::_currObstacle [private]

8.23.5.5 qreal GUI_Scene::_height [protected]

8.23.5.6 vector<Obstacle> GUI_Scene::_obstacle [protected]

8.23.5.7 QWidget* GUI_Scene::_paramBox [private]

8.23.5.8 qreal GUI_Scene::_scale [protected]

8.23.5.9 qreal GUI_Scene::_width [protected]

8.23.5.10 bool GUI_Scene::antiAliased [private]

8.23.5.11 QImage GUI_Scene::image [private]

8.23.5.12 QGraphicsPolygonItem* GUI_Scene::newObstacle [private]

8.23.5.13 bool GUI_Scene::potentialMap [private]

Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/gui_scene.hpp
- src/gui_scene.cpp

8.24 Dokumentacja klasy History

```
#include <history.hpp>
```

8.24.1 Opis szczegółowy

Klasa przechowująca historie zwierząt i zdarzeń.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-29

Sloty publiczne

- void `saveToFile` (string `fname`)

Metody publiczne

- void `add` (unsigned int `ID`, `AnimalParam` &`param`, unsigned int `t`)
- void `addEvent` (`Event` &`ev`)
- `AnimalParam` `get` (unsigned int `ID`, unsigned int `t`)
- void `getEvent` (unsigned int `t`, list< `Event` > &`ret`)
- `History` ()

Atrybuty chronione

- map< unsigned int, `History_` > `_animalHist`
- list< `Event` > `_eventHist`

8.24.2 Dokumentacja konstruktora i destruktora

8.24.2.1 `History::History` ()

Konstruktor.

8.24.3 Dokumentacja funkcji składowych

8.24.3.1 void `History::add` (unsigned int `ID`, `AnimalParam` & `param`, unsigned int `t`)

Dodaj moment historyczny

Parametry:

identyfikator zwierzecia
parametry w danym momencie
moment [ms]

8.24.3.2 void History::addEvent (Event & ev)

Dodaje zdarzenie

Parametry:

zdarzenie

8.24.3.3 AnimalParam History::get (unsigned int ID, unsigned int t)

Pobierz moment historyczny

Parametry:

identyfikator zwierzecia
moment [ms]

Zwraca:

zwracane sa parametry bedace najblizej momentu, jesli identyfikator nie istnieje, zwracane sa 'zerowe' parametry

8.24.3.4 void History::getEvent (unsigned int t, list< Event > & ret)

Pobiera zdarzenie

Parametry:

moment [ms]
poprzez zmienna zwracana jest lista zdarzen, ktore wystapily w danym momencie

8.24.3.5 void History::saveToFile (string fname) [slot]

Wykonuje rzut historii w formacie m-file.

Parametry:

fname

```
iterHist->second.saveToFile(mfile, (*iterHist).first);
```

8.24.4 Dokumentacja atrybutów składowych

8.24.4.1 `map<unsigned int, History_> History::_animalHist` [protected]

8.24.4.2 `list<Event> History::_eventHist` [protected]

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/history.hpp`
- `src/history.cpp`

8.25 Dokumentacja klasy History_

```
#include <history_.hpp>
```

Metody publiczne

- void **add** (**AnimalParam** ¶m, unsigned int t)
- **AnimalParam** **get** (unsigned int t)
- **History_** ()
- void **saveToFile** (fstream &plik, unsigned int i)

Atrybuty chronione

- list< **HistoryStruct** > **_story**

Atrybuty prywatne

- list< **HistoryStruct** >::iterator **_iterStory**
- **HistoryStruct** **_tmpHist**

8.25.1 Dokumentacja konstruktora i destruktor

8.25.1.1 **History_::History_** ()

Konstruktor.

8.25.2 Dokumentacja funkcji składowych

8.25.2.1 void **History_::add** (**AnimalParam** & *param*, unsigned int *t*)

Dodaj moment historyczny

Parametry:

parametry w danym momencie

moment [ms]

8.25.2.2 **AnimalParam** **History_::get** (unsigned int *t*)

Pobierz moment historyczny

Parametry:

moment [ms]

Zwraca:

zwracane sa parametry bedace najblizej momentu

8.25.2.3 void History_::saveToFile (fstream & *plik*, unsigned int *i*)

Wykonuje rzut historii do pliku w formacie m-file (tworzy tabele).o nazwie: [prefix][num]

Parametry:

adres do pliku

i - id przyjęty do zapisu

Zwraca:

wynik zapisywany jest do pliku, indeks tabeli wyznacza ID zwierzaka brak rozroznienia na ofiary/drapieżców

8.25.3 Dokumentacja atrybutów składowych

8.25.3.1 list<HistoryStruct>::iterator History_::_iterStory [private]

8.25.3.2 list<HistoryStruct> History_::_story [protected]

8.25.3.3 HistoryStruct History_::_tmpHist [private]

Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/history_.hpp
- src/history_.cpp

8.26 Dokumentacja klasy HistoryStruct

```
#include <history_.hpp>
```

8.26.1 Opis szczegółowy

Klasa przechowująca historie jednego zwierzęcia.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-29

Atrybuty publiczne

- **AnimalParam param**
- **unsigned int t**

8.26.2 Dokumentacja atrybutów składowych

8.26.2.1 AnimalParam HistoryStruct::param

8.26.2.2 unsigned int HistoryStruct::t

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `inc/history_.hpp`

8.27 Dokumentacja klasy MainWindow

```
#include <gui_mainWindow.hpp>
```

Metody publiczne

- **MainWindow** (QWidget *parent=0, Qt::WindowFlags flags=0)
- void **setupMain** ()

Sloty prywatne

- void **slot _closeChild** ()
- void **slot _closeMain** ()
- void **slot _showAuthors** ()
- void **slot _showEditor** ()
- void **slot _showGame** ()
- void **slot _showHelp** ()
- void **slot _showMain** ()

Atrybuty prywatne

- QAction * **act _OpenAuthors**
- GUI_ * **subGUI**

8.27.1 Dokumentacja konstruktora i destruktora

8.27.1.1 MainWindow::MainWindow (QWidget * *parent* = 0, Qt::WindowFlags *flags* = 0)

Klasa glownego okienka programu

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2008-01-15

8.27.2 Dokumentacja funkcji składowych

8.27.2.1 void MainWindow::setupMain ()

8.27.2.2 void MainWindow::slot _closeChild () [private, slot]

Informuje o zamknięciu okienka dziecka, o podanym ID

8.27.2.3 void MainWindow::slot_closeMain () [private, slot]

Odpowiada za zamknięcie programu, okienka głównego i dzieci

8.27.2.4 void MainWindow::slot_showAuthors () [private, slot]

8.27.2.5 void MainWindow::slot_showEditor () [private, slot]

8.27.2.6 void MainWindow::slot_showGame () [private, slot]

8.27.2.7 void MainWindow::slot_showHelp () [private, slot]

8.27.2.8 void MainWindow::slot_showMain () [private, slot]

8.27.3 Dokumentacja atrybutów składowych

8.27.3.1 QAction* MainWindow::act_OpenAuthors [private]

8.27.3.2 GUI_* MainWindow::subGUI [private]

Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/gui_mainWindow.hpp
- src/gui_mainWindow.cpp

8.28 Dokumentacja klasy noGUI

```
#include <noGUI.hpp>
```

Sloty publiczne

- void **saveHist** (QApplication &app, char *nazMat)
- void **saveHist_** (char *nazMat)

Metody publiczne

- **noGUI** (QApplication &app, char *nazCfg, char *nazMat, int t)

Atrybuty prywatne

- list< **Animal** > **a**
- **Animal** * **aTmp**
- list< **Animal** > **b**
- **SceneConfig** * **cfg**
- unsigned int **maxID**
- **SceneManager** * **sm**

8.28.1 Dokumentacja konstruktora i destruktora

8.28.1.1 **noGUI::noGUI** (QApplication & *app*, char * *nazCfg*, char * *nazMat*, int *t*)

8.28.2 Dokumentacja funkcji składowych

8.28.2.1 void **noGUI::saveHist** (QApplication & *app*, char * *nazMat*) [slot]

8.28.2.2 void **noGUI::saveHist_** (char * *nazMat*) [slot]

8.28.3 Dokumentacja atrybutów składowych

8.28.3.1 list<Animal> **noGUI::a** [private]

8.28.3.2 **Animal*** **noGUI::aTmp** [private]

8.28.3.3 list<Animal> **noGUI::b** [private]

8.28.3.4 **SceneConfig*** **noGUI::cfg** [private]

8.28.3.5 unsigned int **noGUI::maxID** [private]

8.28.3.6 **SceneManager*** **noGUI::sm** [private]

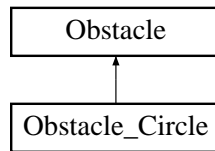
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/**noGUI.hpp**
- src/**noGUI.cpp**

8.29 Dokumentacja klasy Obstacle

```
#include <obstacle.hpp>
```

Diagram dziedziczenia dla Obstacle:



8.29.1 Opis szczegółowy

Klasa definiująca przeszkode.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

Metody publiczne

- virtual double **getAngle** (**Vector2D** &vec, double orient)
- virtual **Vector2D** **getCross** (**Vector2D** &vec, double orient)
- std::string **getType** ()
- **Obstacle** ()
- void **setConfig** (**SceneConfig** &cfg)

Atrybuty chronione

- **Vector2D** **_bufRetPoint**
- double **_bufSrcOrient**
- **Vector2D** **_bufSrcPoint**
- **SceneConfig** **_cfg**
- std::string **_type**

8.29.2 Dokumentacja konstruktora i destruktora

8.29.2.1 Obstacle::Obstacle ()

Konstruktor.

8.29.3 Dokumentacja funkcji składowych

8.29.3.1 `double Obstacle::getAngle (Vector2D & vec, double orient)` [virtual]

Zwraca 'kat natarcia' do najbliższej krawedzi przedzkiody

Parametry:

punkt odniesienia

kierunek [rad]

Zwraca:

kat [rad] w przedziale $-\pi/2$ do $\pi/2$, albo π w przypadku, gdy nie mozna okreslic katu

Reimplementowana w `Obstacle_Circle` (str. 83).

8.29.3.2 `Vector2D Obstacle::getCross (Vector2D & vec, double orient)` [virtual]

Zwraca punkt przeciecia najbliższej krawedzi przeszkody.

Parametry:

punkt odniesienia

kierunek [rad]

Zwraca:

punkt przeciecia, gdy zwrocony punkt przeciecia pokrywa sie z *vec*, to znaczy, ze przeciecie nie zachodzi

Reimplementowana w `Obstacle_Circle` (str. 83).

8.29.3.3 `std::string Obstacle::getType ()`

Zwraca typ przeszkody

Zwraca:

typ przeszkody

8.29.3.4 `void Obstacle::setConfig (SceneConfig & cfg)`

Pobiera konfiguracje, parametry symulacji

8.29.4 Dokumentacja atrybutów składowych

8.29.4.1 Vector2D Obstacle::_bufRetPoint [protected]

8.29.4.2 double Obstacle::_bufSrcOrient [protected]

8.29.4.3 Vector2D Obstacle::_bufSrcPoint [protected]

8.29.4.4 SceneConfig Obstacle::_cfg [protected]

8.29.4.5 std::string Obstacle::_type [protected]

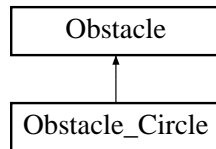
Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/**obstacle.hpp**
- src/**obstacle.cpp**

8.30 Dokumentacja klasy Obstacle_Circle

```
#include <obs_circle.hpp>
```

Diagram dziedziczenia dla Obstacle_Circle:



8.30.1 Opis szczegółowy

Klasa definiująca przeszkode (okrag).

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

Metody publiczne

- double **getAngle** (**Vector2D** &vec, double orient)
- **Vector2D** **getCross** (**Vector2D** &vec, double orient)
- **Obstacle_Circle** (**Vector2D** center, double r)

Atrybuty prywatne

- **Vector2D** **_center**
- double **_r**

8.30.2 Dokumentacja konstruktora i destruktora

8.30.2.1 Obstacle_Circle::Obstacle_Circle (**Vector2D** center, double r)

Konstruktor. Ustala typ przodka na 'Circle'

Parametry:

srodek okregu

promien okregu

8.30.3 Dokumentacja funkcji składowych

8.30.3.1 `double Obstacle_Circle::getAngle (Vector2D & vec, double orient)` [virtual]

Zwraca 'kat natarcia' do najbliższej krawedzi przedzkiody

Parametry:

punkt odniesienia

kierunek [rad]

Zwraca:

kat [rad] w przedziale $-\pi/2$ do $\pi/2$, albo π w przypadku, gdy nie mozna okreslic katu

Reimplementowana z `Obstacle` (str. 80).

8.30.3.2 `Vector2D Obstacle_Circle::getCross (Vector2D & vec, double orient)` [virtual]

Zwraca punkt przeciecia najbliższej krawedzi przeszkody.

Parametry:

punkt odniesienia

kierunek [rad]

Zwraca:

odleglosc [m] albo -1, gdy polprosta nie przecina krawedzi przeszkody

Reimplementowana z `Obstacle` (str. 80).

8.30.4 Dokumentacja atrybutów składowych

8.30.4.1 `Vector2D Obstacle_Circle::_center` [private]

8.30.4.2 `double Obstacle_Circle::_r` [private]

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/obs_circle.hpp`
- `src/obs_circle.cpp`

8.31 Dokumentacja klasy Obstacles

```
#include <obstacles.hpp>
```

8.31.1 Opis szczegółowy

Klasa obsługująca przeszkody. Przede wszystkim ma 'odpowiadać na pytanie', jak daleko w danym kierunku znajduje się najbliższa przeszkoda.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

Metody publiczne

- `double getDistance (Vector2D &vec, double delta)`
- `Obstacles ()`

8.31.2 Dokumentacja konstruktora i destruktora

8.31.2.1 Obstacles::Obstacles ()

Konstruktor.

8.31.3 Dokumentacja funkcji składowych

8.31.3.1 double Obstacles::getDistance (Vector2D & vec, double delta)

Zwraca informacje o tym, jak daleko w danym kierunku znajduje się najbliższa przeszkoda.

Parametry:

punkt odniesienia

orientacja [rad]

Zwraca:

odległość od najbliższej przeszkody [m]

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/obstacles.hpp`
- `src/obstacles.cpp`

8.32 Dokumentacja klasy SceneConfig

```
#include <sceneConfig.hpp>
```

8.32.1 Opis szczegółowy

Klasa przechowuje informacje o konfiguracji sceny.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-12-04

Metody publiczne

- double **getELen** ()
- double **getEPhi** ()
- double **getLevelAvoidObstacles** ()
- double **getLevelBasicFlee** ()
- double **getLevelCohesion** ()
- double **getLevelSeparation** ()
- unsigned int **getMinRM** ()
- int **getMinRT** ()
- double **getRadiusAvoidObstacles** ()
- double **getRadiusBasicFlee** ()
- double **getRadiusCohesion** ()
- double **getRadiusSeparation** ()
- int **getStepTime** ()
- double **getWarningDistance** ()
- **SceneConfig** (int stepTime=5, unsigned int minRM=10)
- void **setLevelAvoidObstacles** (double lc)
- void **setLevelBasicFlee** (double lbf)
- void **setLevelCohesion** (double lc)
- void **setLevelSeparation** (double ls)
- void **setMinRM** (unsigned int minRM)
- void **setRadiusAvoidObstacles** (double rs)
- void **setRadiusBasicFlee** (double rbf)
- void **setRadiusCohesion** (double rc)
- void **setRadiusSeparation** (double rs)
- void **setStepTime** (int stepTime)
- void **setWarningDistance** (double wd)

Atrybuty chronione

- double `_e_len`
- double `_e_phi`
- double `_levelAvoidObstacles`
- double `_levelBasicFlee`
- double `_levelCohesion`
- double `_levelSeparation`
- unsigned int `_minReactionMultiplicity`
- int `_minReactionTime`
- double `_radiusAvoidObstacles`
- double `_radiusBasicFlee`
- double `_radiusCohesion`
- double `_radiusSeparation`
- int `_stepTime`
- double `_warningDistance`

8.32.2 Dokumentacja konstruktora i destruktora

8.32.2.1 SceneConfig::SceneConfig (int *stepTime* = 5, unsigned int *minRM* = 10)

Konstruktor. Ustala elementarny czas kroku symulacji. Dokladnosc katowa (

Zobacz również:

`_e_phi` (str. 91)) ustawiana jest na 0.1 Dokladnosc odlegosci (
`_e_len` (str. 91)) ustawiana jest na 1.

Parametry:

czas kroku symulacji

krotnosc kroku symulacji dla min. czasu reakcji

8.32.3 Dokumentacja funkcji składowych

8.32.3.1 double SceneConfig::getELen ()

Pobiera dokladnosc odlegosci [m]

Zwraca:

dokladnosc odlegosci

8.32.3.2 double SceneConfig::getEPhi ()

Pobiera dokladnosc katowa [rad]

Zwraca:

dokladnosc kata [rad]

8.32.3.3 double SceneConfig::getLevelAvoidObstacles ()

Pobiera mnoznik przeszkod

Zwraca:

mnoznik przeszkod

8.32.3.4 double SceneConfig::getLevelBasicFlee ()

Pobiera mnoznik algorytmu BasicFlee

Zwraca:

mnoznik algorytmu BasicFlee

8.32.3.5 double SceneConfig::getLevelCohesion ()

Pobiera mnoznik algorytmu Cohesion

Zwraca:

mnoznik algorytmu Cohesion

8.32.3.6 double SceneConfig::getLevelSeparation ()

Pobiera mnoznik algorytmu Separation

Zwraca:

mnoznik algorytmu Separation

8.32.3.7 unsigned int SceneConfig::getMinRM ()

Pobiera krotnosc kroku symulacji dla min. czasu reakcji

Zwraca:

minRM

8.32.3.8 int SceneConfig::getMinRT ()

Pobiera min. czas reakcji zwierzecia

Zwraca:

min. czas reakcji zwierzecia [ms]

8.32.3.9 double SceneConfig::getRadiusAvoidObstacles ()

Pobiera promień krytycznej odl. od przeszkody

Zwraca:

promień krytycznej odl. od przeszkody

8.32.3.10 double SceneConfig::getRadiusBasicFlee ()

Pobiera promień algorytmu BasicFlee

Zwraca:

promień algorytmu BasicFlee

8.32.3.11 double SceneConfig::getRadiusCohesion ()

Pobiera promień algorytmu Cohesion

Zwraca:

promień algorytmu Cohesion

8.32.3.12 double SceneConfig::getRadiusSeparation ()

Pobiera promień algorytmu Separation

Zwraca:

promień algorytmu Separation

8.32.3.13 int SceneConfig::getStepTime ()

Pobiera czas kroku symulacji

Zwraca:

czas kroku symulacji [ms]

8.32.3.14 double SceneConfig::getWarningDistance ()

Pobiera parametr minimalnej odległości od przeszkody

Zwraca:

min. odległość od przeszkody, wywołująca funkcję wpływu przeszkody na kierunek ruchu zwierzęcia

8.32.3.15 void SceneConfig::setLevelAvoidObstacles (double *lc*)

Ustala mnoznik przeszkod

Parametry:

mnoznik przeszkod

8.32.3.16 void SceneConfig::setLevelBasicFlee (double *lbf*)

Ustala mnoznik algorytmu BasicFlee

Parametry:

mnoznik algorytmu BasicFlee

8.32.3.17 void SceneConfig::setLevelCohesion (double *lc*)

Ustala mnoznik algorytmu Cohesion

Parametry:

mnoznik algorytmu Cohesion

8.32.3.18 void SceneConfig::setLevelSeparation (double *ls*)

Ustala mnoznik algorytmu Separation

Parametry:

mnoznik algorytmu Separation

8.32.3.19 void SceneConfig::setMinRM (unsigned int *minRM*)

Zapisuje krotnosc kroku symulacji dla min. czasu reakcji

Parametry:

krotnosc kroku symulacji

8.32.3.20 void SceneConfig::setRadiusAvoidObstacles (double *rs*)

Ustala promien krytycznej odl. od przeszkody

Parametry:

promien krytycznej odl. od przeszkody

8.32.3.21 void SceneConfig::setRadiusBasicFlee (double *rbf*)

Ustala promien algorytmu BasicFlee

Parametry:

promien algorytmu BasicFlee

8.32.3.22 void SceneConfig::setRadiusCohesion (double *rc*)

Ustala promien algorytmu Cohesion

Parametry:

promien algorytmu Cohesion

8.32.3.23 void SceneConfig::setRadiusSeparation (double *rs*)

Ustala promien algorytmu Separation

Parametry:

promien algorytmu Separation

8.32.3.24 void SceneConfig::setStepTime (int *stepTime*)

Zapisuje czas kroku symulacji.

Parametry:

czas kroku symulacji [ms]

8.32.3.25 void SceneConfig::setWarningDistance (double *wd*)

Ustala min. odleglosc od przeszkody.

Parametry:

min. odleglosc od przedzkody, wyzwalajaca funkcje wplywu przeszkody na kierunek ruchu zwierzaka

8.32.4 Dokumentacja atrybutów składowych

8.32.4.1 `double SceneConfig::_e_len` [protected]

8.32.4.2 `double SceneConfig::_e_phi` [protected]

8.32.4.3 `double SceneConfig::_levelAvoidObstacles` [protected]

8.32.4.4 `double SceneConfig::_levelBasicFlee` [protected]

8.32.4.5 `double SceneConfig::_levelCohesion` [protected]

8.32.4.6 `double SceneConfig::_levelSeparation` [protected]

8.32.4.7 `unsigned int SceneConfig::_minReactionMultiplicity` [protected]

8.32.4.8 `int SceneConfig::_minReactionTime` [protected]

8.32.4.9 `double SceneConfig::_radiusAvoidObstacles` [protected]

8.32.4.10 `double SceneConfig::_radiusBasicFlee` [protected]

8.32.4.11 `double SceneConfig::_radiusCohesion` [protected]

8.32.4.12 `double SceneConfig::_radiusSeparation` [protected]

8.32.4.13 `int SceneConfig::_stepTime` [protected]

8.32.4.14 `double SceneConfig::_warningDistance` [protected]

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/sceneConfig.hpp`
- `src/sceneConfig.cpp`

8.33 Dokumentacja klasy SceneManager

```
#include <SceneManager.hpp>
```

8.33.1 Opis szczegółowy

Klasa definiująca zarządcę sceny.

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-12-04

Sloty publiczne

- void **pause** ()
- void **play** ()

Metody publiczne

- void **AnimalToAnimal_** (list< **Animal** > &anim, list< **Animal_** > &anim_)
- **History** * **getHistory** () const
- **SceneManager** (**SceneConfig** *cfg, list< **Animal** > *animals, list< **Animal** > *raptors, **Obstacles** *obs=0, **History** *hist=0)

Sloty chronione

- void **step** ()

Metody chronione

- void **step_** (list< **Animal** > &animals, bool isRaptor=false)

Atrybuty chronione

- list< **Animal** > * **_animals**
- list< **Animal_** > **_animals_**
- **SceneConfig** * **_cfg**
- **History** * **_hist**
- list< **Animal** >::iterator **_iterAnim**
- list< **Animal_** >::iterator **_iterAnim_**
- **Obstacles** * **_obs**
- list< **Animal** > * **_raptors**

- list< **Animal** _ > _raptors _
- unsigned int _simTime
- QTimer * _timer

8.33.2 Dokumentacja konstruktora i destruktora

8.33.2.1 SceneManager::SceneManager (SceneConfig * *cfg*, list< **Animal** > * *animals*, list< **Animal** > * *raptors*, Obstacles * *obs* = 0, History * *hist* = 0)

Konstruktor. Tworzy timer na podstawie konfiguracji.

Parametry:

konfiguracja ceny

wskaznik na liste zwierzakow (ofiary)

wskaznik na liste zwierzakow (drapieżcy)

wskaznik na przeszkody

wskaznik na historie

8.33.3 Dokumentacja funkcji składowych

8.33.3.1 void SceneManager::AnimalToAnimal_ (list< **Animal** > & *anim*, list< **Animal** _ > & *anim_*)

Konwertuje liste **Animal** (str. 17) na liste **Animal**_ (str. 21).

Parametry:

referencja na liste **Animal** (str. 17)

referencja na liste **Animal**_ (str. 21), lista ta jest czyszczona przed operacja konwersji

8.33.3.2 History * SceneManager::getHistory () const

Zwraca wskaznik na liste symulacji

Zwraca:

wskaznik na liste symulacji

8.33.3.3 void SceneManager::pause () [slot]

Zatrzymanie symulacji.

8.33.3.4 void SceneManager::play () [slot]

Uruchomienie/wznowienie symulacji wg parametrow

Zobacz również:

_cfg (str. 94)

8.33.3.5 void SceneManager::step () [protected, slot]

Wykonuje krok symulacji. Uaktualnia interwał zegara symulacji.

8.33.3.6 void SceneManager::step_ (list< Animal > & animals, bool isRaptor = false) [protected]

Wykonuje krok symulacji dla listy zwierząt.

Parametry:

lista zwierząt

true jeśli lista zwierząt to drapieżniki, *false* w przeciwnym wypadku

8.33.4 Dokumentacja atrybutów składowych

8.33.4.1 list<Animal>* SceneManager::_animals [protected]

8.33.4.2 list<Animal_> SceneManager::_animals_ [protected]

8.33.4.3 SceneConfig* SceneManager::_cfg [protected]

8.33.4.4 History* SceneManager::_hist [protected]

8.33.4.5 list<Animal>::iterator SceneManager::_iterAnim [protected]

8.33.4.6 list<Animal>::iterator SceneManager::_iterAnim_ [protected]

8.33.4.7 Obstacles* SceneManager::_obs [protected]

8.33.4.8 list<Animal>* SceneManager::_raptors [protected]

8.33.4.9 list<Animal_> SceneManager::_raptors_ [protected]

8.33.4.10 unsigned int SceneManager::_simTime [protected]

8.33.4.11 QTimer* SceneManager::_timer [protected]

Dokumentacja dla tej klasy została wygenerowana z plików:

- inc/sceneManager.hpp
- src/sceneManager.cpp

8.34 Dokumentacja klasy Vector2D

```
#include <vector2D.hpp>
```

8.34.1 Opis szczegółowy

Implementuje wektor 2D, rozumiany również jako punkt w przestrzeni 2D.

Autor:

Arkadiusz Lisiecki

Data:

26.11.2007

Wersja:

1.0

Metody publiczne

- double **argument** ()
- double **distance** (Vector2D vec)
- double **module** ()
- Vector2D **move** (double alpha, double dist)
- Vector2D & **operator** * (const double value)
- Vector2D & **operator** * (const int value)
- bool **operator**!= (const Vector2D &vec) const
- Vector2D & **operator**+ (const Vector2D &vec)
- Vector2D & **operator**- (const Vector2D &vec)
- Vector2D & **operator**/ (const double value)
- Vector2D & **operator**/ (const int value)
- Vector2D & **operator**= (const Vector2D &vec)
- bool **operator**== (const Vector2D &vec1) const
- double **operator**[] (unsigned int idx) const
- double & **operator**[] (unsigned int idx)
- Vector2D (double a_, double b_)
- Vector2D ()

Atrybuty chronione

- double **a**
- double **b**

8.34.2 Dokumentacja konstruktora i destruktora

8.34.2.1 Vector2D::Vector2D ()

Konstruktor, ustala współczynniki na 0

8.34.2.2 `Vector2D::Vector2D (double a_, double b_)`

Konstruktor, z inicjalizacja współczynników

8.34.3 Dokumentacja funkcji składowych

8.34.3.1 `double Vector2D::argument ()`

Wyznacza argument.

Zwraca:

wyznaczony argument wg punktów 2D

8.34.3.2 `double Vector2D::distance (Vector2D vec)`

Zwraca odległość między dwoma punktami (współrzędne bieżącego wektora i współrzędne wektora podanego jako param).

Parametry:

vec wektor

8.34.3.3 `double Vector2D::module ()`

Wyznacza modul

Zwraca:

wyznaczony modul 2D

8.34.3.4 `Vector2D Vector2D::move (double alpha, double dist)`

Przesuwa punkt o zadane odległość i kąt

Parametry:

alpha - kąt [rad]

dist - odległość

Zwraca:

nowe współrzędne

- 8.34.3.5 `Vector2D & Vector2D::operator * (const double value)`
- 8.34.3.6 `Vector2D & Vector2D::operator * (const int value)`
- 8.34.3.7 `bool Vector2D::operator!= (const Vector2D & vec) const`
- 8.34.3.8 `Vector2D & Vector2D::operator+ (const Vector2D & vec)`
- 8.34.3.9 `Vector2D & Vector2D::operator- (const Vector2D & vec)`
- 8.34.3.10 `Vector2D & Vector2D::operator/ (const double value)`
- 8.34.3.11 `Vector2D & Vector2D::operator/ (const int value)`
- 8.34.3.12 `Vector2D & Vector2D::operator= (const Vector2D & vec)`
- 8.34.3.13 `bool Vector2D::operator== (const Vector2D & vec1) const`
- 8.34.3.14 `double Vector2D::operator[] (unsigned int idx) const`
- 8.34.3.15 `double & Vector2D::operator[] (unsigned int idx)`

8.34.4 Dokumentacja atrybutów składowych

- 8.34.4.1 `double Vector2D::a [protected]`
- 8.34.4.2 `double Vector2D::b [protected]`

Dokumentacja dla tej klasy została wygenerowana z plików:

- `inc/vector2D.hpp`
- `src/vector2D.cpp`

Rozdział 9

Zwierzaki Dokumentacja plików

9.1 Dokumentacja pliku inc/animal.hpp

```
#include <list>
#include "animal_.hpp"
#include "behavior.hpp"
#include "obstacles.hpp"
#include "vector2D.hpp"
```

Komponenty

- class **Animal**

Definicje

- #define **ORDER_DISP** 3
- #define **ORDER_FORMGROUP** 2
- #define **ORDER_NOTHING** 0
- #define **ORDER_RUN** 1

Funkcje

- ostream & **operator**<< (ostream &Strm, const **Animal** &anim)
- istream & **operator**>> (istream &Strm, **Animal** &anim)

9.1.1 Dokumentacja definicji

9.1.1.1 `#define ORDER_DISP 3`

9.1.1.2 `#define ORDER_FORMGROUP 2`

9.1.1.3 `#define ORDER_NOTHING 0`

Klasa definiująca zwierzaka (z uwzględnianiem strategii).

Autor:

Arkadiusz Lisiecki

Wersja:

1.0

Data:

2007-11-26

9.1.1.4 `#define ORDER_RUN 1`

9.1.2 Dokumentacja funkcji

9.1.2.1 `ostream& operator<< (ostream & Strm, const Animal & anim)`

9.1.2.2 `istream& operator>> (istream & Strm, Animal & anim)`

9.2 Dokumentacja pliku inc/animal_.hpp

```
#include "animalParam.hpp"  
#include "vector2D.hpp"
```

Przestrzenie nazw

- namespace `std`

Komponenty

- class `Animal_`

Definicje

- `#define ANIMAL_TYPE_ANIMAL 2`
- `#define ANIMAL_TYPE_PREDATOR 1`

9.2.1 Dokumentacja definicji

9.2.1.1 `#define ANIMAL_TYPE_ANIMAL 2`

9.2.1.2 `#define ANIMAL_TYPE_PREDATOR 1`

9.3 Dokumentacja pliku inc/animalParam.hpp

```
#include "vector2D.hpp"
```

Komponenty

- class `AnimalParam`

Definicje

- `#define ANIMAL_STATE_DEAD 0`
- `#define ANIMAL_STATE_LIVE 1`

9.3.1 Dokumentacja definicji

9.3.1.1 `#define ANIMAL_STATE_DEAD 0`

9.3.1.2 `#define ANIMAL_STATE_LIVE 1`

9.4 Dokumentacja pliku inc/b_AvoidObstacles.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B_AvoidObstacles

9.5 Dokumentacja pliku inc/b__BasicFlee.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B__BasicFlee

9.6 Dokumentacja pliku inc/b_Cohesion.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B_Cohesion

9.7 Dokumentacja pliku inc/b_Encirclement.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B_Encirclement

9.8 Dokumentacja pliku inc/b_EncirclementLux.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B_EncirclementLux

9.9 Dokumentacja pliku inc/b_FlightMulti.hpp

```
#include "behavior.hpp"
```

Komponenty

- class `B_FlightMulti`

9.10 Dokumentacja pliku inc/b_Separation.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B_Separation

9.11 Dokumentacja pliku inc/b_Tommy.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B_Tommy

9.12 Dokumentacja pliku inc/b_TommyLux.hpp

```
#include "behavior.hpp"
```

Komponenty

- class `B_TommyLux`

9.13 Dokumentacja pliku inc/b_Wander.hpp

```
#include "behavior.hpp"
```

Komponenty

- class B_Wander

9.14 Dokumentacja pliku inc/behavior.hpp

```
#include <list>
#include "obstacles.hpp"
#include "animal_.hpp"
#include "animalParam.hpp"
#include "sceneConfig.hpp"
```

Komponenty

- class **Behavior**

9.15 Dokumentacja pliku inc/event.hpp

```
#include "vector2D.hpp"
```

Komponenty

- struct **Event**

Definicje

- #define **EVENT_CRASH** 3
- #define **EVENT_FRIEND_CRASH** 2
- #define **EVENT_HUNTED** 1
- #define **EVENT_KILLED** 4

9.15.1 Dokumentacja definicji

9.15.1.1 #define **EVENT_CRASH** 3

9.15.1.2 #define **EVENT_FRIEND_CRASH** 2

9.15.1.3 #define **EVENT_HUNTED** 1

9.15.1.4 #define **EVENT_KILLED** 4

9.16 Dokumentacja pliku inc/gui_ .hpp

```
#include <QMainWindow>
#include <QWidget>
#include <QDialog>
```

Komponenty

- class GUI_

9.17 Dokumentacja pliku inc/gui_animal.hpp

```
#include <QWidget>
#include <QDialog>
#include <QToolBar>
#include <QGraphicsItem>
#include <QPainter>
#include <QTextStream>
#include "vector2D.hpp"
#include "animal.hpp"
```

Komponenty

- class `GUI_Animal`

Funkcje

- ostream & **operator**<< (ostream &Strm, const `GUI_Animal` &anim)
- istream & **operator**>> (istream &Strm, const `GUI_Animal` &anim)

9.17.1 Dokumentacja funkcji

9.17.1.1 ostream& **operator**<< (ostream & *Strm*, const `GUI_Animal` & *anim*)

9.17.1.2 istream& **operator**>> (istream & *Strm*, const `GUI_Animal` & *anim*)

9.18 Dokumentacja pliku inc/gui_authors.hpp

```
#include <QWidget>
#include <QDialog>
#include <QToolBar>
#include "gui_.hpp"
```

Komponenty

- class GUI_Authors

9.19 Dokumentacja pliku inc/gui_editor.hpp

```
#include <iostream>
#include <fstream>
#include <QWidget>
#include <QDialog>
#include <QToolBar>
#include <QLabel>
#include <QGridLayout>
#include <QAction>
#include <QIcon>
#include <QSpinBox>
#include <QPushButton>
#include <QFileDialog>
#include <QFile>
#include <QMessageBox>
#include <QDockWidget>
#include "gui_.hpp"
#include "gui_scene.hpp"
```

Komponenty

- class GUI_Editor

9.20 Dokumentacja pliku inc/gui_game.hpp

```
#include <QWidget>
#include <QDialog>
#include <QToolBar>
#include "gui_.hpp"
```

Komponenty

- class `GUI_Game`

9.21 Dokumentacja pliku inc/gui_help.hpp

```
#include <QWidget>
#include <QDialog>
#include <QToolBar>
#include "gui_.hpp"
```

Komponenty

- class `GUI_Help`

9.22 Dokumentacja pliku inc/gui_mainWindow.hpp

```
#include <QMainWindow>
#include <QToolBar>
#include <QPushButton>
#include "gui_.hpp"
#include "gui_authors.hpp"
#include "gui_help.hpp"
#include "gui_editor.hpp"
#include "gui_game.hpp"
```

Komponenty

- class **MainWindow**

9.23 Dokumentacja pliku inc/gui_obstacle.hpp

```
#include <QWidget>
#include <QDialog>
#include <QToolBar>
#include "vector2D.hpp"
```

Komponenty

- class GUI_Obstacle

9.24 Dokumentacja pliku inc/gui_scene.hpp

```
#include <iostream>
#include <fstream>
#include <QWidget>
#include <QPolygonF>
#include <QRectF>
#include <QVector>
#include <QImage>
#include <QTextStream>
#include <QGraphicsView>
#include <vector>
#include <QGridLayout>
#include <QSpinBox>
#include <QPushButton>
#include <QLabel>
#include <QDialog>
#include <QMouseEvent>
#include <QPoint>
#include <QComboBox>
#include "gui_animal.hpp"
#include "obstacle.hpp"
#include "vector2D.hpp"
```

Komponenty

- class **GUI_Scene**

Funkcje

- ostream & **operator<<** (ostream &Strm, const vector< **GUI_Animal** * > &anim)
- ostream & **operator<<** (ostream &Strm, const **GUI_Scene** &scene)
- istream & **operator>>** (istream &Strm, **GUI_Scene** &scene)

9.24.1 Dokumentacja funkcji

9.24.1.1 ostream& operator<< (ostream & *Strm*, const vector< GUI_Animal * > & *anim*)

9.24.1.2 ostream& operator<< (ostream & *Strm*, const GUI_Scene & *scene*)

9.24.1.3 istream& operator>> (istream & *Strm*, GUI_Scene & *scene*)

9.25 Dokumentacja pliku inc/history.hpp

```
#include <QObject>
#include <iostream>
#include <fstream>
#include <map>
#include <list>
#include "history_.hpp"
#include "event.hpp"
```

Komponenty

- class **History**

9.26 Dokumentacja pliku inc/history_.hpp

```
#include <iostream>
#include <fstream>
#include <list>
#include "animalParam.hpp"
```

Komponenty

- class **History_**
- class **HistoryStruct**

9.27 Dokumentacja pliku inc/noGUI.hpp

```
#include <QApplication>
#include <QTimer>
#include <list>
#include "gui_mainWindow.hpp"
#include "sceneConfig.hpp"
#include "sceneManager.hpp"
#include "animal.hpp"
#include "history.hpp"
#include "b_Encirclement.hpp"
#include "b_FlightMulti.hpp"
```

Komponenty

- class **noGUI**

9.28 Dokumentacja pliku inc/obs_circle.hpp

```
#include "obstacle.hpp"  
#include "vector2D.hpp"
```

Komponenty

- class `Obstacle_Circle`

9.29 Dokumentacja pliku inc/obstacle.hpp

```
#include <string>
#include <iostream>
#include "vector2D.hpp"
#include "sceneConfig.hpp"
```

Komponenty

- class **Obstacle**

9.30 Dokumentacja pliku inc/obstacles.hpp

```
#include "vector2D.hpp"
```

Komponenty

- class `Obstacles`

9.31 Dokumentacja pliku inc/sceneConfig.hpp

Komponenty

- class SceneConfig

9.32 Dokumentacja pliku inc/sceneManager.hpp

```
#include <QObject>
#include <QTimer>
#include <list>
#include "animal.hpp"
#include "obstacles.hpp"
#include "history.hpp"
#include "sceneConfig.hpp"
```

Komponenty

- class `SceneManager`

9.33 Dokumentacja pliku inc/vector2D.hpp

```
#include <iostream>
```

```
#include <cmath>
```

Komponenty

- class **Vector2D**

9.34 Dokumentacja pliku src/animal.cpp

```
#include <list>
#include "obstacles.hpp"
#include "animal_.hpp"
#include "animalParam.hpp"
#include "animal.hpp"
```

Funkcje

- ostream & **operator**<< (ostream &Strm, const **Animal** &anim)
- istream & **operator**>> (istream &Strm, **Animal** &anim)

9.34.1 Dokumentacja funkcji

9.34.1.1 ostream& **operator**<< (ostream & *Strm*, const **Animal** & *anim*)

9.34.1.2 istream& **operator**>> (istream & *Strm*, **Animal** & *anim*)

9.35 Dokumentacja pliku src/animal_.cpp

```
#include "animal_.hpp"
```

9.36 Dokumentacja pliku src/animalParam.cpp

```
#include "animalParam.hpp"
```

9.37 Dokumentacja pliku src/b_AvoidObstacles.cpp

```
#include <cmath>
#include "b_AvoidObstacles.hpp"
```

9.38 Dokumentacja pliku src/b_BasicFlee.cpp

```
#include <cmath>
#include "b_BasicFlee.hpp"
```


9.39 Dokumentacja pliku src/b_Cohesion.cpp

```
#include <cmath>
#include "b_Cohesion.hpp"
```

9.40 Dokumentacja pliku src/b_Encirclement.cpp

```
#include <cmath>
#include "b_Encirclement.hpp"
```

9.41 Dokumentacja pliku src/b_EncirclementLux.cpp

```
#include <cmath>
#include "b_EncirclementLux.hpp"
```

9.42 Dokumentacja pliku src/b_FlightMulti.cpp

```
#include <cmath>
#include "b_FlightMulti.hpp"
```

9.43 Dokumentacja pliku src/b_Separation.cpp

```
#include <cmath>
#include "b_Separation.hpp"
```

9.44 Dokumentacja pliku src/b_Tommy.cpp

```
#include <cmath>
#include "b_Tommy.hpp"
```

9.45 Dokumentacja pliku src/b_TommyLux.cpp

```
#include <cmath>
#include "b_TommyLux.hpp"
```

9.46 Dokumentacja pliku src/b_Wander.cpp

```
#include <QtGlobal>
#include <cmath>
#include "b_Wander.hpp"
```


9.47 Dokumentacja pliku src/behavior.cpp

```
#include "behavior.hpp"
```

9.48 Dokumentacja pliku src/gui_ .cpp

```
#include "gui_ .hpp"
```

9.49 Dokumentacja pliku src/gui_animal.cpp

```
#include "gui_animal.hpp"
```

Funkcje

- ostream & **operator**<< (ostream &Strm, const **GUI_Animal** &anim)
- istream & **operator**>> (istream &Strm, const **GUI_Animal** &anim)

9.49.1 Dokumentacja funkcji

9.49.1.1 ostream& **operator**<< (ostream & *Strm*, const **GUI_Animal** & *anim*)

9.49.1.2 istream& **operator**>> (istream & *Strm*, const **GUI_Animal** & *anim*)

9.50 Dokumentacja pliku src/gui_authors.cpp

```
#include "gui_authors.hpp"
```

9.51 Dokumentacja pliku src/gui_editor.cpp

```
#include "gui_editor.hpp"
```

9.52 Dokumentacja pliku src/gui_game.cpp

```
#include "gui_game.hpp"
```

9.53 Dokumentacja pliku src/gui_help.cpp

```
#include "gui_help.hpp"
```

9.54 Dokumentacja pliku src/gui_mainWindow.cpp

```
#include <QtGui>
#include <QFile>
#include <QTextStream>
#include <QFileDialog>
#include "gui_mainWindow.hpp"
#include <iostream>
```


9.55 Dokumentacja pliku src/gui_obstacle.cpp

```
#include "gui_obstacle.hpp"
```

9.56 Dokumentacja pliku src/gui_scene.cpp

```
#include "gui_scene.hpp"
```

Funkcje

- ostream & **operator**<< (ostream &Strm, const **GUI_Scene** &scene)
- ostream & **operator**<< (ostream &Strm, const vector< **GUI_Animal** * > &anim)
- istream & **operator**>> (istream &Strm, **GUI_Scene** &scene)

9.56.1 Dokumentacja funkcji

9.56.1.1 ostream& **operator**<< (ostream & *Strm*, const **GUI_Scene** & *scene*)

9.56.1.2 ostream& **operator**<< (ostream & *Strm*, const vector< **GUI_Animal** * > & *anim*)

9.56.1.3 istream& **operator**>> (istream & *Strm*, **GUI_Scene** & *scene*)

9.57 Dokumentacja pliku src/history.cpp

```
#include "history.hpp"
```

9.58 Dokumentacja pliku src/history_.cpp

```
#include "history_.hpp"
```

9.59 Dokumentacja pliku src/main.cpp

```
#include <QApplication>
#include <QTimer>
#include <list>
#include "gui_mainWindow.hpp"
#include "sceneConfig.hpp"
#include "sceneManager.hpp"
#include "animal.hpp"
#include "history.hpp"
#include "noGUI.hpp"
#include "b_Encirclement.hpp"
#include "b_FlightMulti.hpp"
```

Funkcje

- int **main** (int argc, char *argv[])
- bool **pomoc** ()

9.59.1 Dokumentacja funkcji

9.59.1.1 int **main** (int *argc*, char * *argv*[])

9.59.1.2 bool **pomoc** ()

Pomoc

9.60 Dokumentacja pliku src/noGUI.cpp

```
#include "noGUI.hpp"
```

9.61 Dokumentacja pliku src/obs_circle.cpp

```
#include "obs_circle.hpp"
```

9.62 Dokumentacja pliku src/obstacle.cpp

```
#include "obstacle.hpp"
```


9.63 Dokumentacja pliku src/obstacles.cpp

```
#include "obstacles.hpp"
```

9.64 Dokumentacja pliku src/sceneConfig.cpp

```
#include "sceneConfig.hpp"
```

9.65 Dokumentacja pliku src/sceneManager.cpp

```
#include "sceneManager.hpp"
```

9.66 Dokumentacja pliku src/vector2D.cpp

```
#include "vector2D.hpp"
```


- Animal_, 27
- _minReactionMultiplicity
 - SceneConfig, 91
- _minReactionTime
 - SceneConfig, 91
- _name
 - Animal_, 27
 - Behavior, 54
- _obs
 - SceneManager, 94
- _obstacle
 - GUI_Scene, 69
- _param
 - Animal_, 27
- _paramBox
 - GUI_Scene, 69
- _phi_max
 - Animal_, 27
- _pos
 - AnimalParam, 31
- _r
 - Animal_, 27
 - Obstacle_Circle, 83
- _radiusAvoidObstacles
 - SceneConfig, 91
- _radiusBasicFlee
 - SceneConfig, 91
- _radiusCohesion
 - SceneConfig, 91
- _radiusSeparation
 - SceneConfig, 91
- _raptors
 - SceneManager, 94
- _raptors_
 - SceneManager, 94
- _scale
 - GUI_Scene, 69
- _simTime
 - SceneManager, 94
- _state
 - AnimalParam, 31
- _stepTime
 - SceneConfig, 91
- _story
 - History_, 74
- _t
 - Animal_, 27
- _timer
 - SceneManager, 94
- _tmpHist
 - History_, 74
- _type
 - Animal_, 27
 - Obstacle, 81
- _warningDistance
 - SceneConfig, 91
- _width
 - GUI_Scene, 69
- ~GUI_Scene
 - GUI_Scene, 67
- a
 - noGUI, 78
 - Vector2D, 97
- act_OpenAuthors
 - MainWindow, 77
- action
 - B_AvoidObstacles, 32
 - B_BasicFlee, 34
 - B_Cohesion, 36
 - B_Encirclement, 38
 - B_EncirclementLux, 40
 - B_FlightMulti, 42
 - B_Separation, 44
 - B_Tommy, 46
 - B_TommyLux, 48
 - B_Wander, 50
 - Behavior, 53
- add
 - History, 70
 - History_, 73
- ADDANIMAL
 - GUI_Scene, 67
- addAnimal
 - GUI_Scene, 67
- addEvent
 - History, 71
- addObAct
 - GUI_Editor, 62
- ADDOBSTACLE
 - GUI_Scene, 67
- addShAct
 - GUI_Editor, 62
- addToFile
 - GUI_Scene, 67
- Animal, 17
 - _ID, 20
 - _LiderID, 20
 - _Order, 20
 - _OrderParam, 20
 - _Target, 20
 - _behavior, 20
 - Animal, 18
 - getID, 18
 - getLiderID, 18
 - go, 18
 - operator<<, 20
 - operator>>, 20

- reaction, 19
- setBehavior, 19
- setID, 19
- setLiderID, 19
- animal.cpp
 - operator<<, 134
 - operator>>, 134
- animal.hpp
 - operator<<, 100
 - operator>>, 100
 - ORDER_DISP, 100
 - ORDER_FORMGROUP, 100
 - ORDER_NOTHING, 100
 - ORDER_RUN, 100
- Animal_, 21
 - _E_max, 27
 - _V_max, 27
 - _a_max, 27
 - _dE_max, 27
 - _dV_max, 27
 - _da_max, 27
 - _dphi_max, 27
 - _m, 27
 - _name, 27
 - _param, 27
 - _phi_max, 27
 - _r, 27
 - _t, 27
 - _type, 27
- Animal_, 22
 - get_a_max, 22
 - get_da_max, 22
 - get_dE_max, 23
 - get_dphi_max, 23
 - get_dV_max, 23
 - get_E_max, 23
 - get_m, 23
 - get_phi_max, 23
 - get_r, 24
 - get_t, 24
 - get_V_max, 24
 - getAnimal_, 24
 - getType, 24
 - set_a_max, 24
 - set_da_max, 24
 - set_dE_max, 25
 - set_dphi_max, 25
 - set_dV_max, 25
 - set_E_max, 25
 - set_m, 25
 - set_phi_max, 25
 - set_r, 26
 - set_t, 26
 - set_V_max, 26
- setType, 26
- animal_.hpp
 - ANIMAL_TYPE_ANIMAL, 101
 - ANIMAL_TYPE_PREDATOR, 101
- ANIMAL_STATE_DEAD
 - animalParam.hpp, 102
- ANIMAL_STATE_LIVE
 - animalParam.hpp, 102
- ANIMAL_TYPE_ANIMAL
 - animal_.hpp, 101
- ANIMAL_TYPE_PREDATOR
 - animal_.hpp, 101
- AnimalParam, 28
 - _E, 31
 - _V, 31
 - _a, 31
 - _delta, 31
 - _pos, 31
 - _state, 31
- AnimalParam, 29
 - geta, 29
 - getDelta, 29
 - getE, 29
 - getPos, 29
 - getState, 29
 - getV, 29
 - seta, 30
 - setDelta, 30
 - setE, 30
 - setPos, 30
 - setState, 30
 - setV, 30
- animalParam.hpp
 - ANIMAL_STATE_DEAD, 102
 - ANIMAL_STATE_LIVE, 102
- AnimalToAnimal_
 - SceneManager, 93
- antiAliased
 - GUI_Scene, 69
- argument
 - Vector2D, 96
- aTmp
 - noGUI, 78
- b
 - noGUI, 78
 - Vector2D, 97
- B_AvoidObstacles, 32
 - action, 32
 - B_AvoidObstacles, 32
- B_BasicFlee, 34
 - action, 34
 - B_BasicFlee, 34
- B_Cohesion, 36

- action, 36
 - B_Cohesion, 36
- B_Encirclement, 38
 - action, 38
 - B_Encirclement, 38
- B_EncirclementLux, 40
 - action, 40
 - B_EncirclementLux, 40
- B_FlightMulti, 42
 - action, 42
 - B_FlightMulti, 42
- B_Separation, 44
 - action, 44
 - B_Separation, 44
- B_Tommy, 46
 - action, 46
 - B_Tommy, 46
- B_TommyLux, 48
 - action, 48
 - B_TommyLux, 48
- B_Wander, 50
 - action, 50
 - B_Wander, 50
- Behavior, 52
 - _cfg, 54
 - _description, 54
 - _name, 54
 - action, 53
 - Behavior, 53
 - setConfig, 53
- boundingRect
 - GUI_Animal, 57
- cfg
 - noGUI, 78
- changed
 - GUI_Scene, 68
- clearPolygon
 - GUI_Scene, 68
- createPotentialMap
 - GUI_Scene, 68
- distance
 - Vector2D, 96
- Dokumentacja katalogu inc/, 11
- Dokumentacja katalogu src/, 13
- drawBackground
 - GUI_Scene, 68
- edAddAnimal
 - GUI_Editor, 61
- edAddObstacle
 - GUI_Editor, 61
- editToolBar
 - GUI_Editor, 62
- edRem
 - GUI_Editor, 61
- edSelect
 - GUI_Editor, 61
- Event, 55
 - event, 55
 - ID1, 55
 - ID2, 55
 - pos1, 55
 - pos2, 55
 - t, 55
- event
 - Event, 55
- event.hpp
 - EVENT_CRASH, 114
 - EVENT_FRIEND_CRASH, 114
 - EVENT_HUNTED, 114
 - EVENT_KILLED, 114
- EVENT_CRASH
 - event.hpp, 114
- EVENT_FRIEND_CRASH
 - event.hpp, 114
- EVENT_HUNTED
 - event.hpp, 114
- EVENT_KILLED
 - event.hpp, 114
- fileToolBar
 - GUI_Editor, 62
- get
 - History, 71
 - History_, 73
- get_a_max
 - Animal_, 22
- get_da_max
 - Animal_, 22
- get_dE_max
 - Animal_, 23
- get_dphi_max
 - Animal_, 23
- get_dV_max
 - Animal_, 23
- get_E_max
 - Animal_, 23
- get_m
 - Animal_, 23
- get_phi_max
 - Animal_, 23
- get_r
 - Animal_, 24
- get_t
 - Animal_, 24

- get_V_max
 - Animal_, 24
- geta
 - AnimalParam, 29
- getAngle
 - Obstacle, 80
 - Obstacle_Circle, 83
- getAnimal
 - GUI_Animal, 57
- getAnimal_
 - Animal_, 24
- getAnimalConst
 - GUI_Animal, 57
- getAnimals
 - GUI_Scene, 68
- getCross
 - Obstacle, 80
 - Obstacle_Circle, 83
- getCurrAnimal
 - GUI_Scene, 68
- getDelta
 - AnimalParam, 29
- getDistance
 - Obstacles, 84
- getE
 - AnimalParam, 29
- getELen
 - SceneConfig, 86
- getEPhi
 - SceneConfig, 86
- getEvent
 - History, 71
- getHistory
 - SceneManager, 93
- getID
 - Animal, 18
- getLevelAvoidObstacles
 - SceneConfig, 86
- getLevelBasicFlee
 - SceneConfig, 87
- getLevelCohesion
 - SceneConfig, 87
- getLevelSeparation
 - SceneConfig, 87
- getLiderID
 - Animal, 18
- getMinRM
 - SceneConfig, 87
- getMinRT
 - SceneConfig, 87
- getParamBox
 - GUI_Scene, 68
- getParamBoxAnimal
 - GUI_Scene, 68
- getPos
 - AnimalParam, 29
- getRadiusAvoidObstacles
 - SceneConfig, 87
- getRadiusBasicFlee
 - SceneConfig, 88
- getRadiusCohesion
 - SceneConfig, 88
- getRadiusSeparation
 - SceneConfig, 88
- getStatAnimal
 - GUI_Animal, 57
- getState
 - AnimalParam, 29
- getStepTime
 - SceneConfig, 88
- getType
 - Animal_, 24
 - Obstacle, 80
- getV
 - AnimalParam, 29
- getWarningDistance
 - SceneConfig, 88
- go
 - Animal, 18
- gScene
 - GUI_Editor, 62
- GUI_, 56
 - GUI_, 56
- GUI_Animal, 57
 - _animal, 58
 - boundingRect, 57
 - getAnimal, 57
 - getAnimalConst, 57
 - getStatAnimal, 57
 - GUI_Animal, 57
 - newPos, 57
 - operator<<, 58
 - operator>>, 58
 - paint, 57
 - radius, 58
 - selected, 58
- gui_animal.cpp
 - operator<<, 149
 - operator>>, 149
- gui_animal.hpp
 - operator<<, 116
 - operator>>, 116
- GUI_Authors, 59
 - GUI_Authors, 59
- GUI_Editor, 60
 - _dockParamWidget, 62
 - addObAct, 62
 - addShAct, 62

- edAddAnimal, 61
- edAddObstacle, 61
- editToolBar, 62
- edRem, 61
- edSelect, 61
- fileToolBar, 62
- gScene, 62
- GUI_Editor, 61
- initScene, 61
- newAct, 62
- newScene, 61
- openAct, 62
- openScene, 61
- remAct, 62
- saveAct, 62
- saveScene, 61
- selAct, 62
- setParamBox, 61
- setupActions, 61
- GUI_Game, 63
 - GUI_Game, 63
- GUI_Help, 64
 - GUI_Help, 64
- GUI_Obstacle, 65
 - GUI_Obstacle, 65
- GUI_Scene, 66
 - _animals, 69
 - _currAnimal, 69
 - _currMode, 69
 - _currObstacle, 69
 - _height, 69
 - _obstacle, 69
 - _paramBox, 69
 - _scale, 69
 - _width, 69
 - ~GUI_Scene, 67
- ADDANIMAL, 67
- addAnimal, 67
- ADDOBSTACLE, 67
- addToFile, 67
- antiAliased, 69
- changed, 68
- clearPolygon, 68
- createPotentialMap, 68
- drawBackground, 68
- getAnimals, 68
- getCurrAnimal, 68
- getParamBox, 68
- getParamBoxAnimal, 68
- GUI_Scene, 67
- image, 69
- isSelectedAnimal, 68
- keyPressEvent, 68
- mode, 67
- mouseDoubleClickEvent, 68
- mouseMoveEvent, 68
- mousePressEvent, 68
- mouseReleaseEvent, 68
- newObstacle, 69
- NONE, 67
- operator<<, 69
- operator>>, 69
- potentialMap, 69
- REMOVE, 67
- selectSheep, 69
- setEditMode, 69
- gui_scene.cpp
 - operator<<, 156
 - operator>>, 156
- gui_scene.hpp
 - operator<<, 124
 - operator>>, 124
- History, 70
 - _animalHist, 72
 - _eventHist, 72
 - add, 70
 - addEvent, 71
 - get, 71
 - getEvent, 71
 - History, 70
 - saveToFile, 71
- History_, 73
 - _iterStory, 74
 - _story, 74
 - _tmpHist, 74
 - add, 73
 - get, 73
 - History_, 73
 - saveToFile, 73
- HistoryStruct, 75
 - param, 75
 - t, 75
- ID1
 - Event, 55
- ID2
 - Event, 55
- image
 - GUI_Scene, 69
- inc/animal.hpp, 99
- inc/animal_.hpp, 101
- inc/animalParam.hpp, 102
- inc/b_AvoidObstacles.hpp, 103
- inc/b_BasicFlee.hpp, 104
- inc/b_Cohesion.hpp, 105
- inc/b_Encirclement.hpp, 106
- inc/b_EncirclementLux.hpp, 107

inc/b_FlightMulti.hpp, 108
 inc/b_Separation.hpp, 109
 inc/b_Tommy.hpp, 110
 inc/b_TommyLux.hpp, 111
 inc/b_Wander.hpp, 112
 inc/behavior.hpp, 113
 inc/event.hpp, 114
 inc/gui_.hpp, 115
 inc/gui_animal.hpp, 116
 inc/gui_authors.hpp, 117
 inc/gui_editor.hpp, 118
 inc/gui_game.hpp, 119
 inc/gui_help.hpp, 120
 inc/gui_mainWindow.hpp, 121
 inc/gui_obstacle.hpp, 122
 inc/gui_scene.hpp, 123
 inc/history.hpp, 125
 inc/history_.hpp, 126
 inc/noGUI.hpp, 127
 inc/obs_circle.hpp, 128
 inc/obstacle.hpp, 129
 inc/obstacles.hpp, 130
 inc/sceneConfig.hpp, 131
 inc/sceneManager.hpp, 132
 inc/vector2D.hpp, 133
 initScene
 GUI_Editor, 61
 isSelectedAnimal
 GUI_Scene, 68

 keyPressEvent
 GUI_Scene, 68

 main
 main.cpp, 159
 main.cpp
 main, 159
 pomoc, 159
 MainWindow, 76
 act_OpenAuthors, 77
 MainWindow, 76
 setupMain, 76
 slot_closeChild, 76
 slot_closeMain, 76
 slot_showAuthors, 77
 slot_showEditor, 77
 slot_showGame, 77
 slot_showHelp, 77
 slot_showMain, 77
 subGUI, 77
 maxID
 noGUI, 78
 mode
 GUI_Scene, 67

 module
 Vector2D, 96
 mouseDoubleClickEvent
 GUI_Scene, 68
 mouseMoveEvent
 GUI_Scene, 68
 mousePressEvent
 GUI_Scene, 68
 mouseReleaseEvent
 GUI_Scene, 68
 move
 Vector2D, 96

 newAct
 GUI_Editor, 62
 newObstacle
 GUI_Scene, 69
 newPos
 GUI_Animal, 57
 newScene
 GUI_Editor, 61
 noGUI, 78
 a, 78
 aTmp, 78
 b, 78
 cfg, 78
 maxID, 78
 noGUI, 78
 saveHist, 78
 saveHist_, 78
 sm, 78
 NONE
 GUI_Scene, 67

 Obstacle, 79
 _bufRetPoint, 81
 _bufSrcOrient, 81
 _bufSrcPoint, 81
 _cfg, 81
 _type, 81
 getAngle, 80
 getCross, 80
 getType, 80
 Obstacle, 79
 setConfig, 80
 Obstacle_Circle, 82
 _center, 83
 _r, 83
 getAngle, 83
 getCross, 83
 Obstacle_Circle, 82
 Obstacles, 84
 getDistance, 84
 Obstacles, 84

openAct
 GUI_Editor, 62
 openScene
 GUI_Editor, 61
 operator *
 Vector2D, 96, 97
 operator !=
 Vector2D, 97
 operator +
 Vector2D, 97
 operator -
 Vector2D, 97
 operator /
 Vector2D, 97
 operator <<
 Animal, 20
 animal.cpp, 134
 animal.hpp, 100
 GUI_Animal, 58
 gui_animal.cpp, 149
 gui_animal.hpp, 116
 GUI_Scene, 69
 gui_scene.cpp, 156
 gui_scene.hpp, 124
 operator =
 Vector2D, 97
 operator ==
 Vector2D, 97
 operator >>
 Animal, 20
 animal.cpp, 134
 animal.hpp, 100
 GUI_Animal, 58
 gui_animal.cpp, 149
 gui_animal.hpp, 116
 GUI_Scene, 69
 gui_scene.cpp, 156
 gui_scene.hpp, 124
 operator []
 Vector2D, 97
 ORDER_DISP
 animal.hpp, 100
 ORDER_FORMGROUP
 animal.hpp, 100
 ORDER_NOTHING
 animal.hpp, 100
 ORDER_RUN
 animal.hpp, 100

 paint
 GUI_Animal, 57
 param
 HistoryStruct, 75
 pause
 SceneManager, 93
 play
 SceneManager, 93
 pomoc
 main.cpp, 159
 pos1
 Event, 55
 pos2
 Event, 55
 potentialMap
 GUI_Scene, 69

 radius
 GUI_Animal, 58
 reaction
 Animal, 19
 remAct
 GUI_Editor, 62
 REMOVE
 GUI_Scene, 67

 saveAct
 GUI_Editor, 62
 saveHist
 noGUI, 78
 saveHist_
 noGUI, 78
 saveScene
 GUI_Editor, 61
 saveToFile
 History, 71
 History_, 73
 SceneConfig, 85
 _e_len, 91
 _e_phi, 91
 _levelAvoidObstacles, 91
 _levelBasicFlee, 91
 _levelCohesion, 91
 _levelSeparation, 91
 _minReactionMultiplicity, 91
 _minReactionTime, 91
 _radiusAvoidObstacles, 91
 _radiusBasicFlee, 91
 _radiusCohesion, 91
 _radiusSeparation, 91
 _stepTime, 91
 _warningDistance, 91
 getELen, 86
 getEPhi, 86
 getLevelAvoidObstacles, 86
 getLevelBasicFlee, 87
 getLevelCohesion, 87
 getLevelSeparation, 87
 getMinRM, 87

- getMinRT, 87
- getRadiusAvoidObstacles, 87
- getRadiusBasicFlee, 88
- getRadiusCohesion, 88
- getRadiusSeparation, 88
- getStepTime, 88
- getWarningDistance, 88
- SceneConfig, 86
- setLevelAvoidObstacles, 88
- setLevelBasicFlee, 89
- setLevelCohesion, 89
- setLevelSeparation, 89
- setMinRM, 89
- setRadiusAvoidObstacles, 89
- setRadiusBasicFlee, 89
- setRadiusCohesion, 90
- setRadiusSeparation, 90
- setStepTime, 90
- setWarningDistance, 90
- SceneManager, 92
 - _animals, 94
 - _animals_, 94
 - _cfg, 94
 - _hist, 94
 - _iterAnim, 94
 - _iterAnim_, 94
 - _obs, 94
 - _raptors, 94
 - _raptors_, 94
 - _simTime, 94
 - _timer, 94
 - AnimalToAnimal_, 93
 - getHistory, 93
 - pause, 93
 - play, 93
 - SceneManager, 93
 - step, 93
 - step_, 94
- selAct
 - GUI_Editor, 62
- selected
 - GUI_Animal, 58
- selectSheep
 - GUI_Scene, 69
- set_a_max
 - Animal_, 24
- set_da_max
 - Animal_, 24
- set_dE_max
 - Animal_, 25
- set_dphi_max
 - Animal_, 25
- set_dV_max
 - Animal_, 25
- set_E_max
 - Animal_, 25
- set_m
 - Animal_, 25
- set_phi_max
 - Animal_, 25
- set_r
 - Animal_, 26
- set_t
 - Animal_, 26
- set_V_max
 - Animal_, 26
- seta
 - AnimalParam, 30
- setBehavior
 - Animal, 19
- setConfig
 - Behavior, 53
 - Obstacle, 80
- setDelta
 - AnimalParam, 30
- setE
 - AnimalParam, 30
- setEditMode
 - GUI_Scene, 69
- setID
 - Animal, 19
- setLevelAvoidObstacles
 - SceneConfig, 88
- setLevelBasicFlee
 - SceneConfig, 89
- setLevelCohesion
 - SceneConfig, 89
- setLevelSeparation
 - SceneConfig, 89
- setLiderID
 - Animal, 19
- setMinRM
 - SceneConfig, 89
- setParamBox
 - GUI_Editor, 61
- setPos
 - AnimalParam, 30
- setRadiusAvoidObstacles
 - SceneConfig, 89
- setRadiusBasicFlee
 - SceneConfig, 89
- setRadiusCohesion
 - SceneConfig, 90
- setRadiusSeparation
 - SceneConfig, 90
- setState
 - AnimalParam, 30
- setStepTime

- SceneConfig, 90
- setType
 - Animal_, 26
- setupActions
 - GUI_Editor, 61
- setupMain
 - MainWindow, 76
- setV
 - AnimalParam, 30
- setWarningDistance
 - SceneConfig, 90
- slot_closeChild
 - MainWindow, 76
- slot_closeMain
 - MainWindow, 76
- slot_showAuthors
 - MainWindow, 77
- slot_showEditor
 - MainWindow, 77
- slot_showGame
 - MainWindow, 77
- slot_showHelp
 - MainWindow, 77
- slot_showMain
 - MainWindow, 77
- sm
 - noGUI, 78
- src/animal.cpp, 134
- src/animal_.cpp, 135
- src/animalParam.cpp, 136
- src/b_AvoidObstacles.cpp, 137
- src/b_BasicFlee.cpp, 138
- src/b_Cohesion.cpp, 139
- src/b_Encirclement.cpp, 140
- src/b_EncirclementLux.cpp, 141
- src/b_FlightMulti.cpp, 142
- src/b_Separation.cpp, 143
- src/b_Tommy.cpp, 144
- src/b_TommyLux.cpp, 145
- src/b_Wander.cpp, 146
- src/behavior.cpp, 147
- src/gui_.cpp, 148
- src/gui_animal.cpp, 149
- src/gui_authors.cpp, 150
- src/gui_editor.cpp, 151
- src/gui_game.cpp, 152
- src/gui_help.cpp, 153
- src/gui_mainWindow.cpp, 154
- src/gui_obstacle.cpp, 155
- src/gui_scene.cpp, 156
- src/history.cpp, 157
- src/history_.cpp, 158
- src/main.cpp, 159
- src/noGUI.cpp, 160
- src/obs_circle.cpp, 161
- src/obstacle.cpp, 162
- src/obstacles.cpp, 163
- src/sceneConfig.cpp, 164
- src/sceneManager.cpp, 165
- src/vector2D.cpp, 166
- std, 15
- step
 - SceneManager, 93
- step_
 - SceneManager, 94
- subGUI
 - MainWindow, 77
- t
 - Event, 55
 - HistoryStruct, 75
- Vector2D, 95
 - a, 97
 - argument, 96
 - b, 97
 - distance, 96
 - module, 96
 - move, 96
 - operator *, 96, 97
 - operator!=, 97
 - operator+, 97
 - operator-, 97
 - operator/, 97
 - operator=, 97
 - operator==, 97
 - operator[], 97
 - Vector2D, 95