Arcade

Generated by Doxygen 1.8.17

1 Arcade **1**

1.1 Introduction . . . 1

1.2 Games . . . 1

1.3 Graphics Libs . . . 1

1.4 Commands and keys . . . 1

1.4.1 Game -> Play . . . 1

1.4.2 Game -> Other actions . . . 2

1.4.3 Menu -> Commands . . . 2

1.5 Interface Sharing . . . 2

1.5.1 Groups . . . 2

1.6 Technical Documentation . . . 2

2 Hierarchical Index **3**

2.1 Class Hierarchy . . . 3

3 Class Index **5**

3.1 Class List . . . 5

4 File Index **7**

4.1 File List . . . 7

5 Class Documentation **9**

5.1 Bullet Class Reference . . . 9

5.1.1 Constructor & Destructor Documentation . . . 9

5.1.1.1 Bullet() . . . 9

5.1.1.2 ~Bullet() . . . 9

5.1.2 Member Data Documentation . . . 10

5.1.2.1 _x . . . 10

5.1.2.2 _y . . . 10

5.2 Centipede Class Reference . . . 10

5.2.1 Detailed Description . . . 11

5.2.2 Constructor & Destructor Documentation . . . 12

5.2.2.1 Centipede() . . . 12

5.2.2.2 ~Centipede() . . . 12

5.2.3 Member Function Documentation . . . 12

5.2.3.1 changeDir() . . . 12

5.2.3.2 checkCollision() . . . 13

5.2.3.3 checkInputs() . . . 14

5.2.3.4 createCentipede() . . . 14

5.2.3.5 getBulletX() . . . 15

5.2.3.6 getBulletY() . . . 15

5.2.3.7 getFoodX() . . . 15

5.2.3.8 getFoodY() . . . 15

5.2.3.9 getScore() . . . 16

5.2.3.10 getSnake() 16
5.2.3.11 getSnakeXpos() 16
5.2.3.12 getSnakeYpos() 16
5.2.3.13 getXpos() 16
5.2.3.14 getYpos() 17
5.2.3.15 hitWall() 17
5.2.3.16 relaunch() 17
5.2.3.17 setLock() 18
5.2.4 Member Data Documentation 18
5.2.4.1 bullet_x 18
5.2.4.2 bullet_y 18
5.2.4.3 food_x 18
5.2.4.4 food_y 18
5.2.4.5 hit_counter 18
5.2.4.6 lock 18
5.2.4.7 snakes 19
5.2.4.8 wall 19
5.2.4.9 x 19
5.2.4.10 xcoords 19
5.2.4.11 y 19
5.2.4.12 ycoords 19
5.3 Core Class Reference 20
5.3.1 Detailed Description 22
5.3.2 Constructor & Destructor Documentation 22
5.3.2.1 Core() 22
5.3.2.2 ∼Core() 22
5.3.3 Member Function Documentation 22
5.3.3.1 addPlayerName() 23
5.3.3.2 gameLoop() 23
5.3.3.3 gamePrint() 24
5.3.3.4 getGame() 25
5.3.3.5 getGameName() 26
5.3.3.6 getGraphic() 26
5.3.3.7 getGraphName() 26
5.3.3.8 getInput() 27
5.3.3.9 getPlay() 28
5.3.3.10 inGamesName() 28
5.3.3.11 inGraphicsName() 29
5.3.3.12 lib() 29
5.3.3.13 menu() 30
5.3.3.14 pActual() 31
5.3.3.15 pCommand() 32

5.3.3.16 pNotHandle() . . . 32
5.3.3.17 printLib() . . . 33
5.3.3.18 setFirstGame() . . . 33
5.3.3.19 setFirstGraph() . . . 34
5.3.3.20 setGame() . . . 35
5.3.3.21 setGraphic() . . . 36
5.3.3.22 setNextGame() . . . 36
5.3.3.23 setNextGraphic() . . . 37
5.3.3.24 setPosGame() . . . 37
5.3.3.25 setPosGraph() . . . 38
5.3.3.26 setPrevGame() . . . 38
5.3.3.27 setPrevGraphic() . . . 39
5.3.3.28 startOrExit() . . . 40
5.3.4 Member Data Documentation . . . 41
5.3.4.1 _actualGame . . . 41
5.3.4.2 _actualLib . . . 41
5.3.4.3 _command . . . 41
5.3.4.4 _game . . . 41
5.3.4.5 _gamesName . . . 41
5.3.4.6 _graphic . . . 42
5.3.4.7 _graphName . . . 42
5.3.4.8 _maxGame . . . 42
5.3.4.9 _maxGraph . . . 42
5.3.4.10 _play . . . 42
5.3.4.11 _playerName . . . 42
5.3.4.12 _posGame . . . 42
5.3.4.13 _posGraph . . . 43
5.4 DLLoader Class Reference . . . 43
5.4.1 Detailed Description . . . 44
5.4.2 Constructor & Destructor Documentation . . . 44
5.4.2.1 DLLoader() . . . 44
5.4.2.2 ∼DLLoader() . . . 44
5.4.3 Member Function Documentation . . . 44
5.4.3.1 closeAllHandles() . . . 44
5.4.3.2 getGameInstance() . . . 44
5.4.3.3 getGameInstanceName() . . . 45
5.4.3.4 getGamesNb() . . . 45
5.4.3.5 getGraphicsNb() . . . 46
5.4.3.6 getGraphInstance() . . . 46
5.4.3.7 getGraphInstanceName() . . . 46
5.4.3.8 getInstanceOfGame() . . . 46
5.4.3.9 getInstanceOfGraph() . . . 47

5.4.3.10 insideOfGames() 48
5.4.4 Member Data Documentation 48
5.4.4.1 _games 48
5.4.4.2 _gamesName 48
5.4.4.3 _gamesNb 48
5.4.4.4 _graphics 48
5.4.4.5 _graphicsName 49
5.4.4.6 _graphicsNb 49
5.4.4.7 _handles 49
5.5 IGame Class Reference 49
5.5.1 Detailed Description 50
5.5.2 Constructor & Destructor Documentation 50
5.5.2.1 ∼IGame() 50
5.5.3 Member Function Documentation 50
5.5.3.1 checkInputs() 50
5.5.3.2 getBulletX() 51
5.5.3.3 getBulletY() 51
5.5.3.4 getFoodX() 51
5.5.3.5 getFoodY() 51
5.5.3.6 getScore() 51
5.5.3.7 getSnakeXpos() 52
5.5.3.8 getSnakeYpos() 52
5.5.3.9 getXpos() 52
5.5.3.10 getYpos() 52
5.5.3.11 relaunch() 52
5.5.3.12 setLock() 53
5.6 IGraphic Class Reference 53
5.6.1 Detailed Description 54
5.6.2 Constructor & Destructor Documentation 54
5.6.2.1 ∼IGraphic() 54
5.6.3 Member Function Documentation 54
5.6.3.1 drawobj() 54
5.6.3.2 getInput() 54
5.6.3.3 launch() 55
5.6.3.4 settingInput() 55
5.6.3.5 stop() 55
5.7 MyNcurses Class Reference 56
5.7.1 Detailed Description 57
5.7.2 Constructor & Destructor Documentation 57
5.7.2.1 MyNcurses() 57
5.7.2.2 ∼MyNcurses() 57
5.7.3 Member Function Documentation 57

5.7.3.1 drawobj() 57
5.7.3.2 getInput() 58
5.7.3.3 init() 58
5.7.3.4 launch() 59
5.7.3.5 print_map() 59
5.7.3.6 settingInput() 59
5.7.3.7 stop() 59
5.7.4 Member Data Documentation 60
5.7.4.1 _input 60
5.7.4.2 _screen 60
5.7.4.3 apple 60
5.7.4.4 bullet 60
5.7.4.5 rectangle 60
5.8 MySDL2 Class Reference 61
5.8.1 Detailed Description 62
5.8.2 Constructor & Destructor Documentation 62
5.8.2.1 MySDL2() 62
5.8.2.2 ∼MySDL2() 62
5.8.3 Member Function Documentation 62
5.8.3.1 drawApple() 62
5.8.3.2 drawBullet() 63
5.8.3.3 drawobj() 64
5.8.3.4 drawSnake() 64
5.8.3.5 getInput() 65
5.8.3.6 init() 65
5.8.3.7 launch() 65
5.8.3.8 setInput() 66
5.8.3.9 settingInput() 66
5.8.3.10 stop() 66
5.8.4 Member Data Documentation 66
5.8.4.1 _event 67
5.8.4.2 _input 67
5.8.4.3 _render 67
5.8.4.4 _screen 67
5.9 MySfml Class Reference 67
5.9.1 Detailed Description 68
5.9.2 Constructor & Destructor Documentation 69
5.9.2.1 MySfml() 69
5.9.2.2 ∼MySfml() 69
5.9.3 Member Function Documentation 69
5.9.3.1 drawobj() 69
5.9.3.2 getInput() 70

5.9.3.3 init() 70
5.9.3.4 launch() 70
5.9.3.5 setInput() 70
5.9.3.6 settingInput() 71
5.9.3.7 stop() 71
5.9.4 Member Data Documentation 71
5.9.4.1 _input 71
5.9.4.2 apple 72
5.9.4.3 bullet 72
5.9.4.4 rectangle 72
5.9.4.5 screen 72
5.10 Nibbler Class Reference 72
5.10.1 Detailed Description 74
5.10.2 Constructor & Destructor Documentation 74
5.10.2.1 Nibbler() 74
5.10.2.2 ∼Nibbler() 74
5.10.3 Member Function Documentation 74
5.10.3.1 changeDir() 74
5.10.3.2 changeFoodCoord() 75
5.10.3.3 checkInputs() 76
5.10.3.4 checkSnake() 77
5.10.3.5 createSnake() 77
5.10.3.6 getBulletX() 78
5.10.3.7 getBulletY() 78
5.10.3.8 getFoodX() 78
5.10.3.9 getFoodY() 78
5.10.3.10 getScore() 79
5.10.3.11 getSnake() 79
5.10.3.12 getSnakeXpos() 79
5.10.3.13 getSnakeYpos() 79
5.10.3.14 getXpos() 79
5.10.3.15 getYpos() 80
5.10.3.16 relaunch() 80
5.10.3.17 setLock() 80
5.10.4 Member Data Documentation 80
5.10.4.1 food_x 80
5.10.4.2 food_y 81
5.10.4.3 lock 81
5.10.4.4 score 81
5.10.4.5 snakes 81
5.10.4.6 x 81
5.10.4.7 xcoords 81

5.10.4.8 y 81
5.10.4.9 ycoords 81

6 File Documentation 83
6.1 core_f/core.cpp File Reference 83
6.1.1 Detailed Description 83
6.2 core_f/Core.hpp File Reference 84
6.2.1 Function Documentation 85
6.2.1.1 gameLoop() 85
6.2.1.2 get_lib() 85
6.3 core_f/DLLoader.cpp File Reference 86
6.3.1 Detailed Description 86
6.4 core_f/DLLoader.hpp File Reference 87
6.5 core_f/libfinder.cpp File Reference 88
6.5.1 Detailed Description 88
6.5.2 Function Documentation 88
6.5.2.1 find_in_game() 88
6.5.2.2 find_in_gra() 89
6.5.2.3 get_lib() 89
6.5.2.4 niddleyearsold() 90
6.6 core_f/main.cpp File Reference 91
6.6.1 Function Documentation 91
6.6.1.1 main() 92
6.7 game/centipede/centipede.cpp File Reference 92
6.7.1 Detailed Description 93
6.7.2 Function Documentation 93
6.7.2.1 __attribute__() [1/2] 93
6.7.2.2 __attribute__() [2/2] 94
6.7.2.3 calledFirst() 94
6.7.2.4 calledLast() 94
6.7.2.5 entryPoint() 94
6.8 game/centipede/centipede.hpp File Reference 94
6.9 game/igame.hpp File Reference 95
6.10 game/nibbler/nibbler.cpp File Reference 96
6.10.1 Detailed Description 97
6.10.2 Function Documentation 97
6.10.2.1 __attribute__() [1/2] 97
6.10.2.2 __attribute__() [2/2] 98
6.10.2.3 calledFirst() 98
6.10.2.4 calledLast() 98
6.10.2.5 entryPoint() 98
6.11 game/nibbler/nibbler.hpp File Reference 98

6.12 graphic/igraphic.hpp File Reference 99
6.13 graphic/ncurses/ncurses.cpp File Reference 100
6.13.1 Detailed Description 101
6.13.2 Function Documentation 101
6.13.2.1 __attribute__() [1/2] 101
6.13.2.2 __attribute__() [2/2] 101
6.13.2.3 calledFirst() 101
6.13.2.4 calledLast() 101
6.13.2.5 entryPoint() 102
6.14 graphic/ncurses/ncurses.hpp File Reference 102
6.15 graphic/sdl_deux/MySDL2.cpp File Reference 103
6.15.1 Detailed Description 103
6.15.2 Function Documentation 103
6.15.2.1 __attribute__() [1/2] 104
6.15.2.2 __attribute__() [2/2] 104
6.15.2.3 calledFirst() 104
6.15.2.4 calledLast() 104
6.15.2.5 entryPoint() 104
6.16 graphic/sdl_deux/MySDL2.hpp File Reference 105
6.17 graphic/sfml/sfmldraw.cpp File Reference 105
6.17.1 Detailed Description 106
6.17.2 Function Documentation 106
6.17.2.1 __attribute__() [1/2] 107
6.17.2.2 __attribute__() [2/2] 107
6.17.2.3 calledFirst() 107
6.17.2.4 calledLastgra() 107
6.17.2.5 entryPoint() 107
6.18 graphic/sfml/sfmldraw.hpp File Reference 108

Index **109**

Chapter 1

Arcade

1.1 Introduction

Create your own clone of famous games. Your project should comply with a structure that separates the heart of your game launcher and its graphic dependencies.

1.2 Games

Two Games done :

- Nibbler (No wall)
- Centipede (No obstacle)

1.3 Graphics Libs

Three Graphics Libs done :

- SFML (MySfml)
- SDL2 (MySDL2)
- NCurses (MyNcurses)

1.4 Commands and keys

1.4.1 Game -> Play

Keys :

- z -> move up
- q -> move left
- s -> move down
- d -> move right
- space -> shoot bullet (Centipede)

1.4.2 Game -> Other actions

Keys :

- i -> next game
- k -> prev game
- l -> next graphic lib
- j -> previous graphic lib
- a -> quit game and go to menu

1.4.3 Menu -> Commands

Commands :

- start -> launch actual game with actual graphic lib
- exit -> quit menu, close program
- reload -> reload avaible games and graphics lib (in /lib directory)
- lib + "wanted lib" -> set actual lib to wanted lib
- game + "wanted game" -> set actual game to wanted game

1.5 Interface Sharing

Interface Sharing with 1 group

1.5.1 Groups

Julien Delphine (`julien.delphine@epitech.eu`) and Jules Vitrac (`jules.vitrac@epitech.eu`) (Nothing testable)

1.6 Technical Documentation

Technical Documentation in bonus directory

- latex -> refman.pdf for pdf version
- html -> index.html for web version

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Bullet 9
Core 20
DLLoader 43
IGame 49
  Centipede 10
  Nibbler 72
IGraphic 53
  MyNcurses 56
  MySDL2 61
  MySfml 67

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Bullet 9
Centipede
Centipede class for game Centipede use IGame for Interface 10
Core
Core create bridge between game and graphic also do main menu 20
DLLoader
Load .so and create Instance of Interface 43
IGame
Interface for game 49
IGraphic
Interface for graph class 53
MyNcurses
Fucntion for draw game in ncurses use IGraphic interface 56
MySDL2
SDL2 function for draw game and handle input 61
MySfml
SFML function for draw game use IGraphic interface 67
Nibbler
Game use IGame as Interface 72

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

core_f/core.cpp
File for core Function : change Game or Lib . . . 83
core_f/Core.hpp . . . 84
core_f/DLLoader.cpp
DLLoader function for create Instance of IGame or IGraphic and manage it . . . 86
core_f/DLLoader.hpp . . . 87
core_f/libfinder.cpp
Function for add new lib in actual lib vector . . . 88
core_f/main.cpp . . . 91
game/igame.hpp . . . 95
game/centipede/centipede.cpp
Function for Centipede Game . . . 92
game/centipede/centipede.hpp . . . 94
game/nibbler/nibbler.cpp
Function for game nibbler . . . 96
game/nibbler/nibbler.hpp . . . 98
graphic/igraphic.hpp . . . 99
graphic/ncurses/ncurses.cpp
All function for ncurses print of game infos . . . 100
graphic/ncurses/ncurses.hpp . . . 102
graphic/sdl_deux/MySDL2.cpp
SDL2 Function . . . 103
graphic/sdl_deux/MySDL2.hpp . . . 105
graphic/sfml/sfmldraw.cpp
Function for draw with SFML . . . 105
graphic/sfml/sfmldraw.hpp . . . 108

Chapter 5

Class Documentation

5.1 Bullet Class Reference

```
#include <centipede.hpp>
```

Public Member Functions

- Bullet ()
- ∼Bullet ()

Private Attributes

- int _x
- int _y

5.1.1 Constructor & Destructor Documentation

5.1.1.1 Bullet()

```
Bullet::Bullet ( )
```

5.1.1.2 ∼Bullet()

```
Bullet::∼Bullet ( )
```

5.1.2 Member Data Documentation

5.1.2.1 _x

`int Bullet::_x [private]`

5.1.2.2 _y

`int Bullet::_y [private]`

The documentation for this class was generated from the following file:

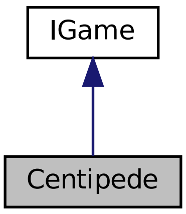
- game/centipede/centipede.hpp

5.2 Centipede Class Reference

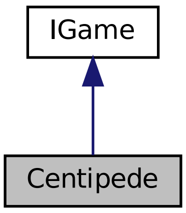
Centipede class for game Centipede use IGame for Interface.

`#include <centipede.hpp>`

Inheritance diagram for Centipede:



Collaboration diagram for Centipede:



Public Member Functions

- Centipede (int a, int b)
- ∼Centipede ()
- int getXpos ()
- int getYpos ()
- std::vector< int > getSnakeXpos ()
- std::vector< int > getSnakeYpos ()
- int getFoodY ()
- int getFoodX ()
- int getBulletY ()
- int getBulletX ()
- std::list< Centipede > getSnake ()
- int checkInputs (int dir)

    reset dir of centipede when touched
- void setLock ()

    set lock to false for good restart of centipede
- void relaunch ()

    relaunch correctly game
- int getScore ()

    return score

Private Member Functions

- void createCentipede ()

    create Centipede
- int changeDir (Centipede logic, int dir)

    change Direction of centipede
- void hitWall ()

    action if wall hited
- void checkCollision (int xx)

    checkCollision bullet / centipede

Private Attributes

- int x
- int y
- std::list< Centipede > snakes
- std::vector< int > xcoords
- std::vector< int > ycoords
- int food_x
- int food_y
- bool lock
- int wall
- int bullet_x
- int bullet_y
- int hit_counter

5.2.1 Detailed Description

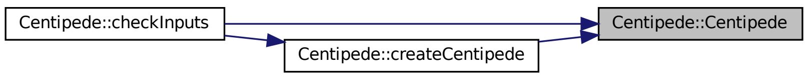
Centipede class for game Centipede use IGame for Interface.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Centipede()

```
Centipede::Centipede (
             int a,
             int b )  [inline]
```

Here is the caller graph for this function:



5.2.2.2 ∼Centipede()

```
Centipede::∼Centipede ( )  [inline]
```

5.2.3 Member Function Documentation

5.2.3.1 changeDir()

```
int Centipede::changeDir (
             Centipede logic,
             int dir )  [private]
```

change Direction of centipede

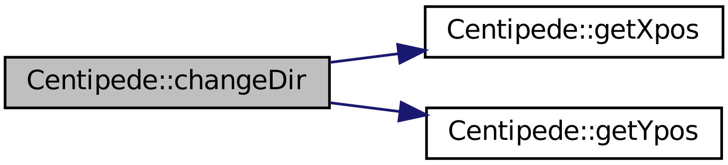
Parameters

in	*logic*	for position of centipede
in	*direction*	of centipde

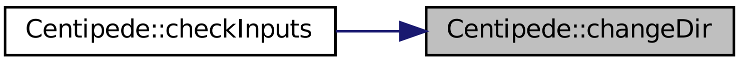
Returns

1 or 0

Here is the call graph for this function:



Here is the caller graph for this function:



5.2.3.2 checkCollision()

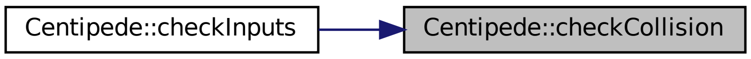
```
void Centipede::checkCollision (
             int xx )  [private]
```

checkCollision bullet / centipede

Parameters

in	*xx*	

Here is the caller graph for this function:



5.2.3.3 checkInputs()

```
int Centipede::checkInputs (
             int dir )  [virtual]
```

reset dir of centipede when touched

Parameters

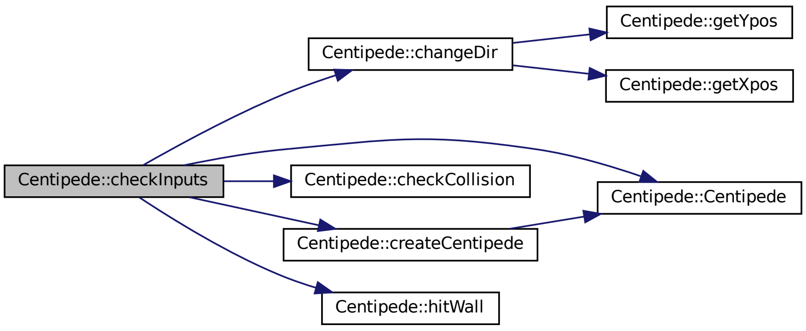
in	*dir*	-> direction of centipede

Returns

0 or 2 for good exit

Implements IGame.

Here is the call graph for this function:

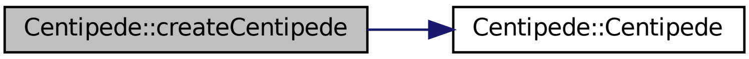


5.2.3.4 createCentipede()

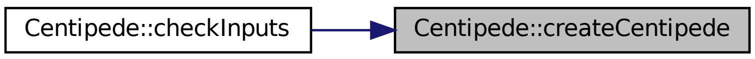
```
void Centipede::createCentipede ( )  [private]
```

create Centipede

Here is the call graph for this function:



Here is the caller graph for this function:



5.2.3.5 getBulletX()

```
int Centipede::getBulletX ( )  [inline], [virtual]
```

Implements IGame.

5.2.3.6 getBulletY()

```
int Centipede::getBulletY ( )  [inline], [virtual]
```

Implements IGame.

5.2.3.7 getFoodX()

```
int Centipede::getFoodX ( )  [inline], [virtual]
```

Implements IGame.

5.2.3.8 getFoodY()

```
int Centipede::getFoodY ( )  [inline], [virtual]
```

Implements IGame.

5.2.3.9 getScore()

```
int Centipede::getScore ( )  [virtual]
```

return score

Returns

hit_counter -> score of centipede

Implements IGame.

5.2.3.10 getSnake()

```
std::list<Centipede> Centipede::getSnake ( )  [inline]
```

5.2.3.11 getSnakeXpos()

```
std::vector<int> Centipede::getSnakeXpos ( )  [inline], [virtual]
```

Implements IGame.

5.2.3.12 getSnakeYpos()

```
std::vector<int> Centipede::getSnakeYpos ( )  [inline], [virtual]
```

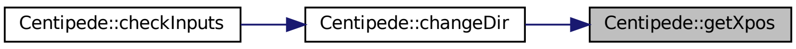
Implements IGame.

5.2.3.13 getXpos()

```
int Centipede::getXpos ( )  [inline], [virtual]
```

Implements IGame.

Here is the caller graph for this function:

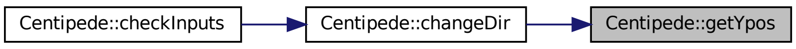


5.2.3.14 getYpos()

```
int Centipede::getYpos ( )  [inline], [virtual]
```

Implements IGame.

Here is the caller graph for this function:

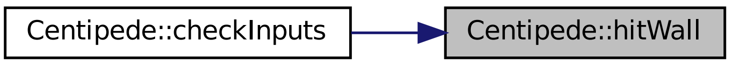


5.2.3.15 hitWall()

```
void Centipede::hitWall ( )  [private]
```

action if wall hited

Here is the caller graph for this function:



5.2.3.16 relaunch()

```
void Centipede::relaunch ( )  [virtual]
```

relaunch correctly game

Implements IGame.

5.2.3.17 setLock()

```
void Centipede::setLock ( )  [virtual]
```

set lock to false for good restart of centipede

Implements IGame.

5.2.4 Member Data Documentation

5.2.4.1 bullet_x

```
int Centipede::bullet_x  [private]
```

5.2.4.2 bullet_y

```
int Centipede::bullet_y  [private]
```

5.2.4.3 food_x

```
int Centipede::food_x  [private]
```

5.2.4.4 food_y

```
int Centipede::food_y  [private]
```

5.2.4.5 hit_counter

```
int Centipede::hit_counter  [private]
```

5.2.4.6 lock

```
bool Centipede::lock  [private]
```

5.2.4.7 snakes

```
std::list<Centipede> Centipede::snakes  [private]
```

5.2.4.8 wall

```
int Centipede::wall  [private]
```

5.2.4.9 x

```
int Centipede::x  [private]
```

5.2.4.10 xcoords

```
std::vector<int> Centipede::xcoords  [private]
```

5.2.4.11 y

```
int Centipede::y  [private]
```

5.2.4.12 ycoords

```
std::vector<int> Centipede::ycoords  [private]
```

The documentation for this class was generated from the following files:

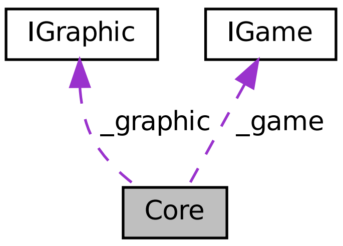
- game/centipede/centipede.hpp
- game/centipede/centipede.cpp

5.3 Core Class Reference

Core create bridge between game and graphic also do main menu.

```
#include <Core.hpp>
```

Collaboration diagram for Core:



Public Member Functions

- Core ()
  Core constructor -> Get Lib of lib/ folder.
- ∼Core ()
  Core Deconstructor.
- void pActual ()
- void pCommand ()
  Print of avaible command in menu.
- void pNotHandle (std::string error)
  Print for NotHandle menu print.
- bool getPlay ()
  Return _play.
- int startOrExit ()
  Handle start and exit command of menu.
- int lib (DLLoader loader)
  Handle change of lib in menu.
- int gamePrint (DLLoader loader)
  Handle change of game in menu.
- void getInput (int input, DLLoader loader)
  Test input for changing lib of game.
- void printLib (std::vector< std::string > libs)
  Print avaible graphics lib in menu.
- void setGame (IGame ∗game)
  set _game with a new IGame instance
- void setGraphic (IGraphic ∗graphic)
  set _graphic with a new Igraph instance
- void setNextGame (DLLoader loader)

set _game to next _gamesName (vector containing name of all games lib we use). If actual _game is last of vector, first one is set

- void setNextGraphic (DLLoader loader)

  set _graphic to next _graphName (vector containing name of all graphics lib we use). If actual _graphic is last of vector, first one is set
- void setPrevGame (DLLoader loader)

  set _game to previous _gamesName (vector containing name of all games lib we use). If actual _game is first of vector, last one is set
- void setPrevGraphic (DLLoader loader)

  set _graphic to previous _graphName (vector containing name of all graphics lib we use). If actual _graphic is first of vector, last one is set
- void setPosGraph (int i)

  set _posGraph to i value
- void setPosGame (int i)

  set _posGame to i value
- std::string getGraphName (int nb)

  return name of graphic vector of given nb element
- std::string getGameName (int nb)

  return name of game vector of given nb element
- IGame ∗ getGame ()

  Return _game.
- IGraphic ∗ getGraphic ()

  return _graphic
- int gameLoop (DLLoader loader)

  all function for restart game and do game loop
- int inGamesName (std::string name)

  check if given name is in vector _gamesName
- int inGraphicsName (std::string name)

  check if given name is in vector _graphName
- void setFirstGame (DLLoader loader)

  set _game to first game found in /lib dir
- void setFirstGraph (DLLoader loader, std::string name)

  set _game to first graph found in /lib dir
- int menu (DLLoader loader)

  menu loop
- void addPlayerName (void)

  get player name

Public Attributes

- std::string _actualGame
- std::string _actualLib

Private Attributes

- IGame ∗ _game
- IGraphic ∗ _graphic
- std::vector< std::string > _gamesName
- std::vector< std::string > _graphName
- bool _play
- int _posGame
- int _posGraph
- int _maxGraph
- int _maxGame
- std::string _command
- std::string _playerName

5.3.1 Detailed Description

Core create bridge between game and graphic also do main menu.

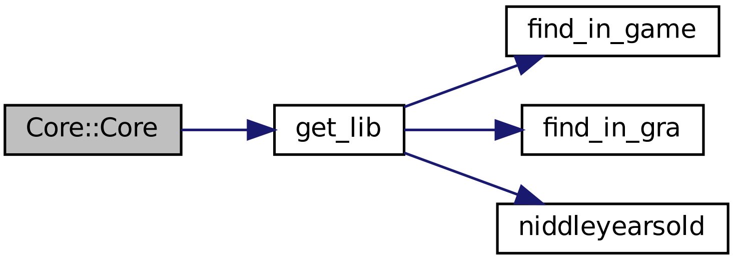
5.3.2 Constructor & Destructor Documentation

5.3.2.1 Core()

```
Core::Core ( )
```

Core constructor -> Get Lib of lib/ folder.

Here is the call graph for this function:



5.3.2.2 ∼Core()

```
Core::∼Core ( )
```

Core Deconstructor.

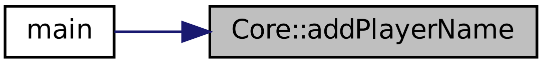
5.3.3 Member Function Documentation

5.3.3.1 addPlayerName()

```
void Core::addPlayerName (
             void  )
```

get player name

Here is the caller graph for this function:



5.3.3.2 gameLoop()

```
int Core::gameLoop (
             DLLoader loader )
```

all function for restart game and do game loop

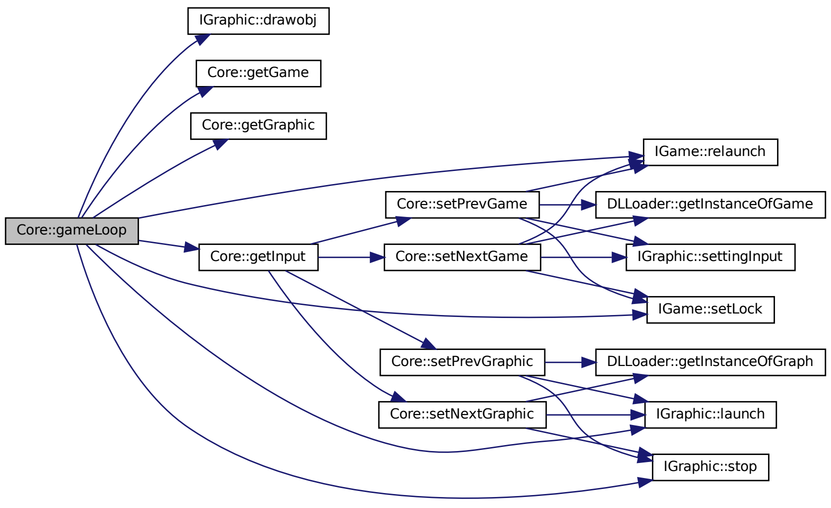
Parameters

in	*loader*	-> get Input -> change lib and game during game

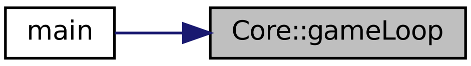
Returns

1 if gq,e quit

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.3 gamePrint()

```
int Core::gamePrint (
             DLLoader loader )
```

Handle change of game in menu.

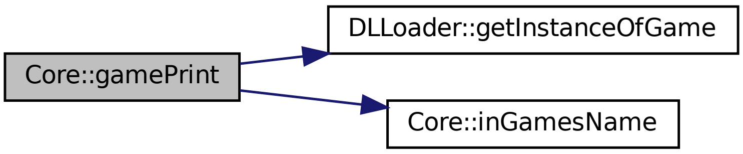
Parameters

in	*loader*	(DLLoader) need for getInstance of Graph or Game

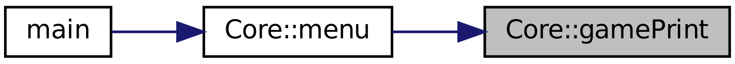
Returns

1 or 0

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.4 getGame()

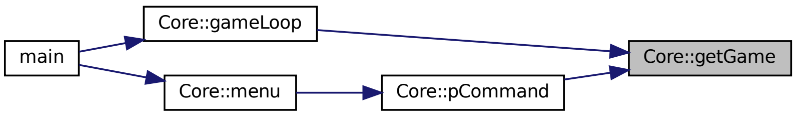
```
IGame * Core::getGame ( )
```

Return _game.

Returns

_game -> Igame of Core

Here is the caller graph for this function:



5.3.3.5 getGameName()

```
std::string Core::getGameName (
              int nb )
```

return name of game vector of given nb element

Parameters

in	*nb*	-> element of vector we want

Returns

_gamesName[nb] -> nb element of vector of games names

5.3.3.6 getGraphic()

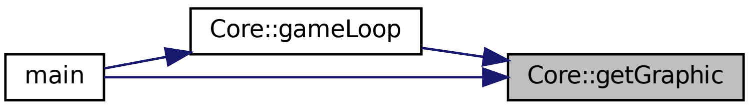
```
IGraphic * Core::getGraphic ( )
```

return _graphic

Returns

_graphic -> IGraph of Core

Here is the caller graph for this function:



5.3.3.7 getGraphName()

```
std::string Core::getGraphName (
              int nb )
```

return name of graphic vector of given nb element

Parameters

in	*nb*	-> element of vector we want

Returns

_graphName[nb] -> nb element of vector of graph names

5.3.3.8 getInput()

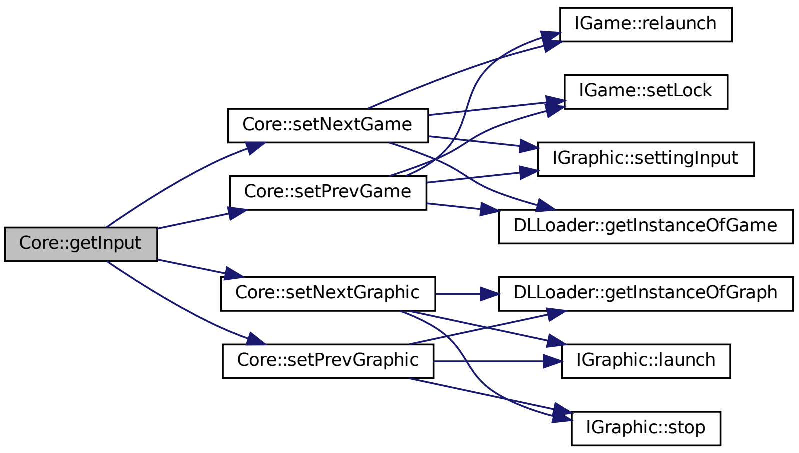
```
void Core::getInput (
             int input,
             DLLoader loader )
```

Test input for changing lib of game.

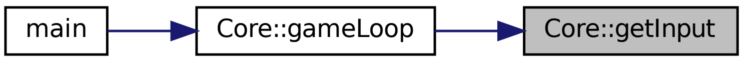
Parameters

in	*input*	-> value of input return by _graphic
in	*loader*	-> Need for get new Instance

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.9 getPlay()

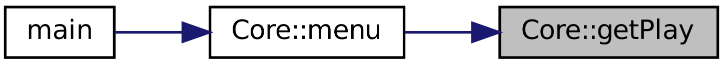
```
bool Core::getPlay ( )
```

Return _play.

Returns

_play

Here is the caller graph for this function:



5.3.3.10 inGamesName()

```
int Core::inGamesName (
            std::string name )
```

check if given name is in vector _gamesName

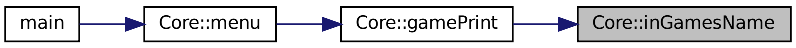
Parameters

in	*name*	-> given name

Returns

-1 if not found position of name in vector if found

Here is the caller graph for this function:



5.3.3.11 inGraphicsName()

```
int Core::inGraphicsName (
            std::string name )
```

check if given name is in vector _graphName

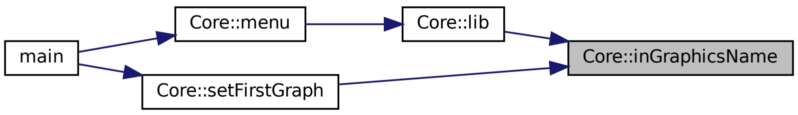
Parameters

in	*name*	-> given name

Returns

-1 if not found position of name in vector if found

Here is the caller graph for this function:



5.3.3.12 lib()

```
int Core::lib (
            DLLoader loader )
```

Handle change of lib in menu.

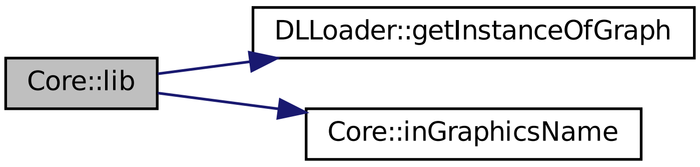
Parameters

`in`	*loader*	(DLLoader) need for getInstance of Graph or Game

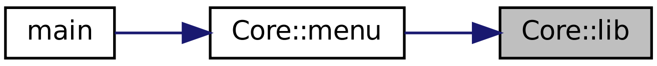
Returns

1 or 0

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.13 menu()

```
int Core::menu (
             DLLoader loader )
```

menu loop

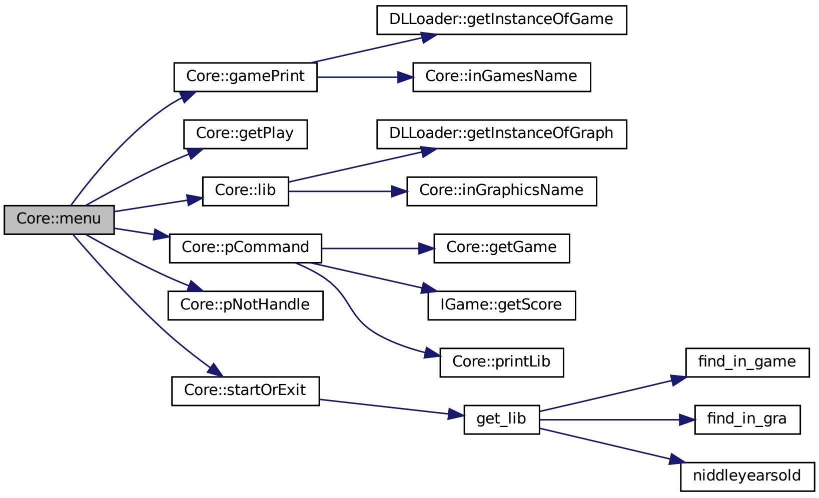
Parameters

`in`	*loader*	-> get new Instance of game or graph

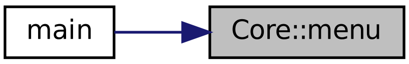
Returns

0 or 1

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.14 pActual()

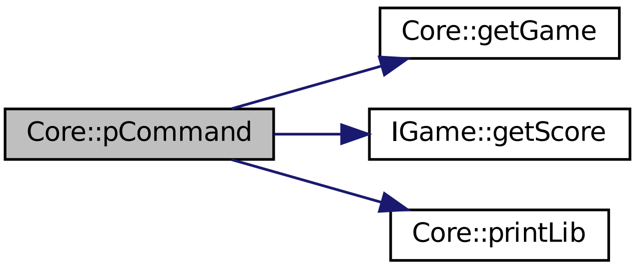
```
void Core::pActual ( )
```

5.3.3.15 pCommand()

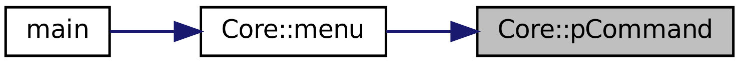
```
void Core::pCommand ( )
```

Print of avaible command in menu.

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.16 pNotHandle()

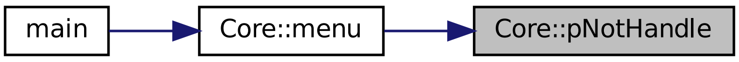
```
void Core::pNotHandle (
             std::string error )
```

Print for NotHandle menu print.

Parameters

in	*String*	for type of not handle type

Here is the caller graph for this function:



5.3.3.17 printLib()

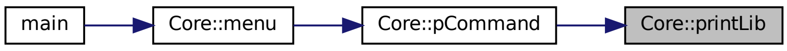
```
void Core::printLib (
             std::vector< std::string > libs )
```

Print avaible graphics lib in menu.

Parameters

in	*vector*	of libs name

Here is the caller graph for this function:



5.3.3.18 setFirstGame()

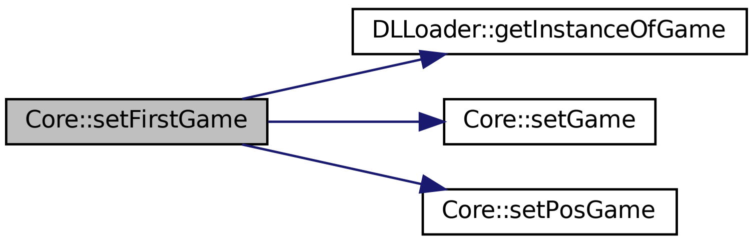
```
void Core::setFirstGame (
             DLLoader loader )
```

set _game to first game found in /lib dir

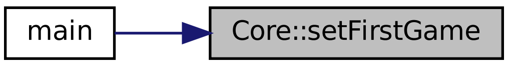
Parameters

in	*loader*	-> use to get instance of game

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.19 setFirstGraph()

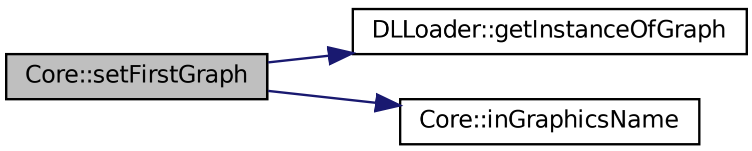
```
void Core::setFirstGraph (
             DLLoader loader,
             std::string name )
```

set _game to first graph found in /lib dir

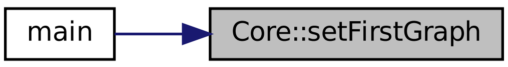
Parameters

in	*loader*	-> use to get instance of graph

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.20 setGame()

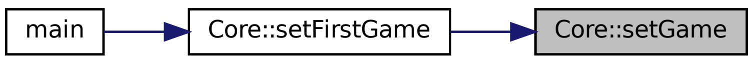
```
void Core::setGame (
              IGame * game )
```

set _game with a new IGame instance

Parameters

in	*game*	-> new Instance of IGame

Here is the caller graph for this function:



5.3.3.21 setGraphic()

```
void Core::setGraphic (
             IGraphic * graphic )
```

set _graphic with a new Igraph instance

Parameters

in	*graphic*	-> new Instance of Igraph

5.3.3.22 setNextGame()

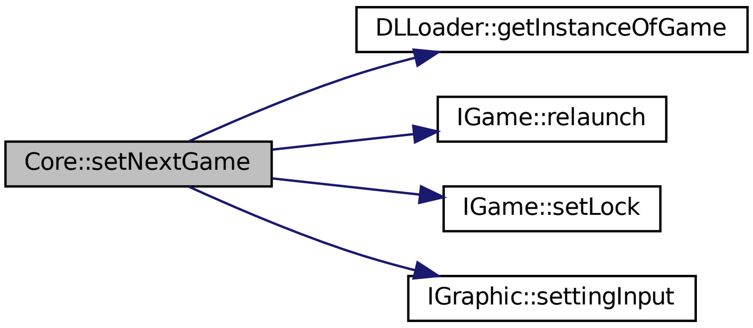
```
void Core::setNextGame (
             DLLoader loader )
```

set _game to next _gamesName (vector containing name of all games lib we use). If actual _game is last of vector, first one is set

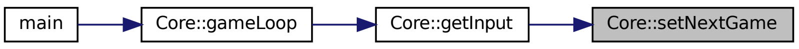
Parameters

in	*loader*	-> need to create new Instance of lib

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.23 setNextGraphic()

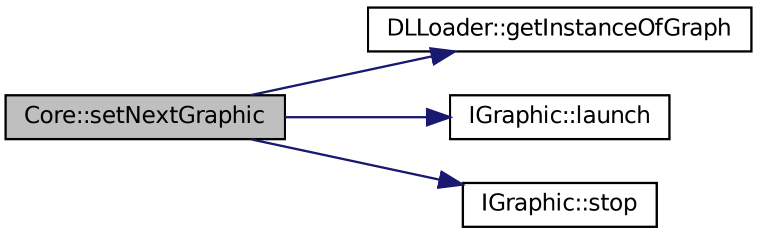
```
void Core::setNextGraphic (
            DLLoader loader )
```

set _graphic to next _graphName (vector containing name of all graphics lib we use). If actual _graphic is last of vector, first one is set

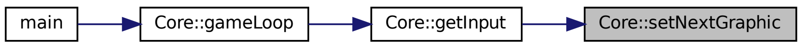
Parameters

in	*loader*	-> need to create new Instance of lib

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.24 setPosGame()

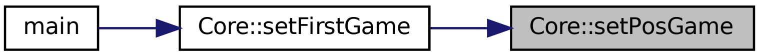
```
void Core::setPosGame (
            int i )
```

set _posGame to i value

Parameters

in	*i*	-> new value we want to set

Here is the caller graph for this function:



5.3.3.25 setPosGraph()

```
void Core::setPosGraph (
             int i )
```

set _posGraph to i value

Parameters

in	*i*	-> new value we want to set

5.3.3.26 setPrevGame()

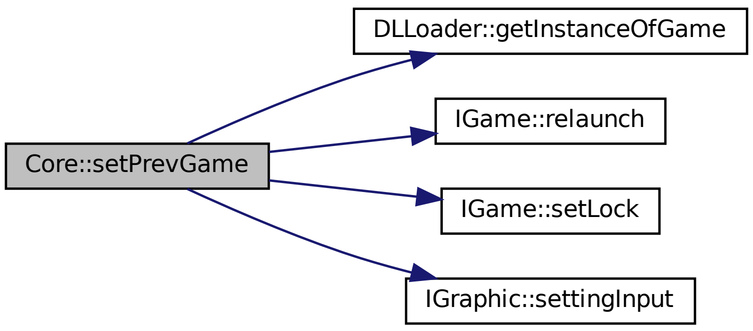
```
void Core::setPrevGame (
             DLLoader loader )
```

set _game to previous _gamesName (vector containing name of all games lib we use). If actual _game is first of vector, last one is set

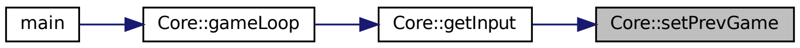
Parameters

in	*loader*	-> need to create new Instance of lib

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.27 setPrevGraphic()

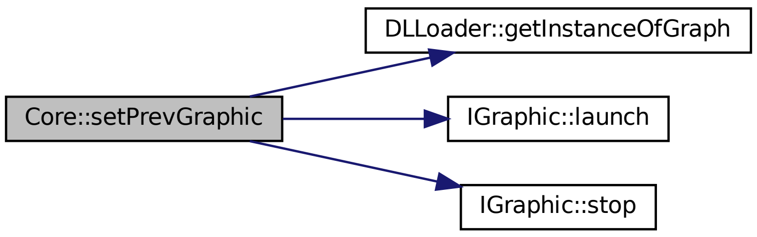
```
void Core::setPrevGraphic (
            DLLoader loader )
```

set _graphic to previous _graphName (vector containing name of all graphics lib we use). If actual _graphic is first of vector, last one is set

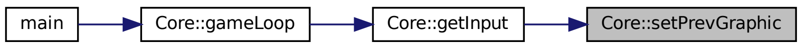
Parameters

in	*loader*	-> need to create new Instance of lib

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.3.28 startOrExit()

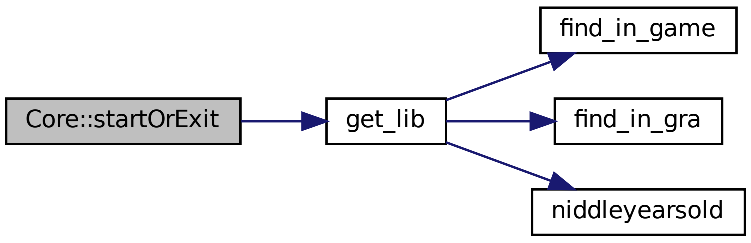
```
int Core::startOrExit ( )
```

Handle start and exit command of menu.

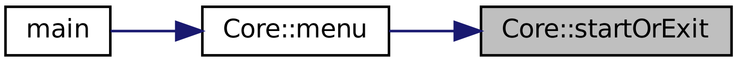
Returns

-1 if exit 2 if reload and 0 for other case

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.4 Member Data Documentation

5.3.4.1 _actualGame

```
std::string Core::_actualGame
```

5.3.4.2 _actualLib

```
std::string Core::_actualLib
```

5.3.4.3 _command

```
std::string Core::_command  [private]
```

5.3.4.4 _game

```
IGame* Core::_game  [private]
```

5.3.4.5 _gamesName

```
std::vector<std::string> Core::_gamesName  [private]
```

5.3.4.6 _graphic

```
IGraphic* Core::_graphic  [private]
```

5.3.4.7 _graphName

```
std::vector<std::string> Core::_graphName  [private]
```

5.3.4.8 _maxGame

```
int Core::_maxGame  [private]
```

5.3.4.9 _maxGraph

```
int Core::_maxGraph  [private]
```

5.3.4.10 _play

```
bool Core::_play  [private]
```

5.3.4.11 _playerName

```
std::string Core::_playerName  [private]
```

5.3.4.12 _posGame

```
int Core::_posGame  [private]
```

5.3.4.13 _posGraph

```
int Core::_posGraph  [private]
```

The documentation for this class was generated from the following files:

- core_f/Core.hpp
- core_f/core.cpp

5.4 DLLoader Class Reference

Load .so and create Instance of Interface.

```
#include <DLLoader.hpp>
```

Public Member Functions

- DLLoader ()

  Constructor of DLLoader.
- ∼DLLoader ()

  Deconstructor of DLLoader.
- IGame ∗ getInstanceOfGame (std::string name)

  use dlopen() and dlsym() for create a new Instance of IGame, put new Instance in vector
- IGraphic ∗ getInstanceOfGraph (std::string name)

  use dlopen() and dlsym() for create a new Instance of IGraph, put new Instance in vector
- void closeAllHandles ()

  close all handles create by dlopen
- std::string getGraphInstanceName (int nb)

  return name of nb element of _graphicsName
- std::string getGameInstanceName (int nb)

  return name of nb element of _gamesName
- IGraphic ∗ getGraphInstance (int nb)

  return IGame instance of nb element of _graphics
- IGame ∗ getGameInstance (int nb)

  return IGame instance of nb element of _games
- int getGamesNb ()

  return _gamesNb
- int getGraphicsNb ()

  return _graphicsNb
- void insideOfGames ()

  print all name of games

Protected Attributes

- std::vector< IGraphic ∗ > _graphics
- std::vector< std::string > _graphicsName
- int _graphicsNb
- std::vector< IGame ∗ > _games
- std::vector< std::string > _gamesName
- int _gamesNb
- std::vector< void ∗ > _handles

5.4.1 Detailed Description

Load .so and create Instance of Interface.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 DLLoader()

```
DLLoader::DLLoader ( )
```

Constructor of DLLoader.

5.4.2.2 ∼DLLoader()

```
DLLoader::∼DLLoader ( )
```

Deconstructor of DLLoader.

5.4.3 Member Function Documentation

5.4.3.1 closeAllHandles()

```
void DLLoader::closeAllHandles ( )
```

close all handles create by dlopen

5.4.3.2 getGameInstance()

```
IGame * DLLoader::getGameInstance (
              int nb )
```

return IGame instance of nb element of _games

Parameters

in	*nb*	-> nth element we wanted

Returns

_games[nb] -> Instance IGame of vector _games

5.4.3.3 getGameInstanceName()

```
std::string DLLoader::getGameInstanceName (
             int nb )
```

return name of nb element of _gamesName

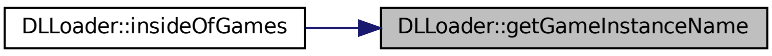
Parameters

in	*nb*	-> nth element we wanted

Returns

_gamesName[nb] -> nb element of vector of name

Here is the caller graph for this function:



5.4.3.4 getGamesNb()

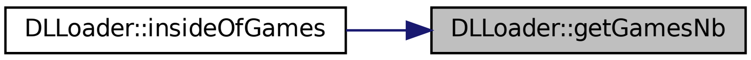
```
int DLLoader::getGamesNb ( )
```

return _gamesNb

Returns

_gamesNb -> number of game load

Here is the caller graph for this function:



5.4.3.5 getGraphicsNb()

```
int DLLoader::getGraphicsNb ( )
```

return _graphicsNb

Returns

_graphicsNb -> number of graphic load

5.4.3.6 getGraphInstance()

```
IGraphic * DLLoader::getGraphInstance (
            int nb )
```

return IGame instance of nb element of _graphics

Parameters

in	*nb*	-> nth element we wanted

Returns

_graphics[nb] -> Instance IGame of vector _graphics

5.4.3.7 getGraphInstanceName()

```
std::string DLLoader::getGraphInstanceName (
            int nb )
```

return name of nb element of _graphicsName

Returns

_graphicsName[nb] -> nb element of vector of name

5.4.3.8 getInstanceOfGame()

```
IGame * DLLoader::getInstanceOfGame (
            std::string name )
```

use dlopen() and dlsym() for create a new Instance of IGame, put new Instance in vector

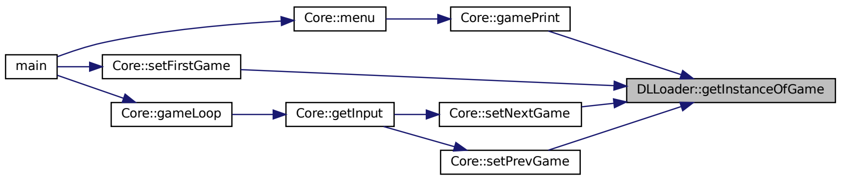
Parameters

in	*name*	-> path of wanted lib
in	*gname*	-> name for represents new Instance

Returns

return of entry -> link to function EntryPoint of wanted class

Here is the caller graph for this function:



5.4.3.9 getInstanceOfGraph()

```
IGraphic * DLLoader::getInstanceOfGraph (
            std::string name )
```

use dlopen() and dlsym() for create a new Instance of IGraph, put new Instance in vector

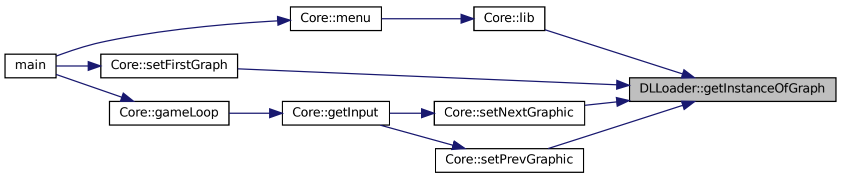
Parameters

in	*name*	-> path of wanted lib
in	*gname*	-> name for represents new Instance

Returns

return of entry -> link to function EntryPoint of wanted class

Here is the caller graph for this function:

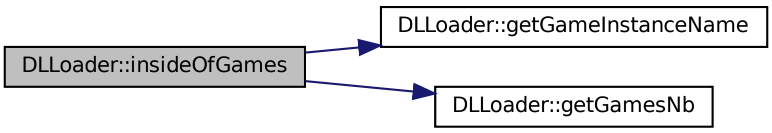


5.4.3.10 insideOfGames()

```
void DLLoader::insideOfGames ( )
```

print all name of games

Here is the call graph for this function:



5.4.4 Member Data Documentation

5.4.4.1 _games

```
std::vector<IGame *> DLLoader::_games [protected]
```

5.4.4.2 _gamesName

```
std::vector<std::string> DLLoader::_gamesName [protected]
```

5.4.4.3 _gamesNb

```
int DLLoader::_gamesNb [protected]
```

5.4.4.4 _graphics

```
std::vector<IGraphic *> DLLoader::_graphics [protected]
```

5.4.4.5 _graphicsName

```
std::vector<std::string> DLLoader::_graphicsName  [protected]
```

5.4.4.6 _graphicsNb

```
int DLLoader::_graphicsNb  [protected]
```

5.4.4.7 _handles

```
std::vector<void *> DLLoader::_handles  [protected]
```

The documentation for this class was generated from the following files:

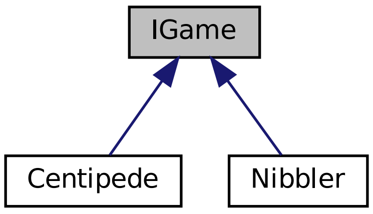
- core_f/DLLoader.hpp
- core_f/DLLoader.cpp

5.5 IGame Class Reference

Interface for game.

```
#include <igame.hpp>
```

Inheritance diagram for IGame:



Public Member Functions

- virtual ∼IGame ()=default
- virtual int checkInputs (int dir)=0
- virtual int getXpos ()=0
- virtual int getYpos ()=0
- virtual std::vector< int > getSnakeXpos ()=0
- virtual std::vector< int > getSnakeYpos ()=0
- virtual int getFoodY ()=0
- virtual int getFoodX ()=0
- virtual int getBulletX ()=0
- virtual int getBulletY ()=0
- virtual void setLock ()=0
- virtual void relaunch ()=0
- virtual int getScore ()=0

5.5.1 Detailed Description

Interface for game.

Interface for Game.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 ∼IGame()

```
virtual IGame::∼IGame ( )  [virtual], [default]
```

5.5.3 Member Function Documentation

5.5.3.1 checkInputs()

```
virtual int IGame::checkInputs (
             int dir )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

5.5.3.2 getBulletX()

```
virtual int IGame::getBulletX ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

5.5.3.3 getBulletY()

```
virtual int IGame::getBulletY ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

5.5.3.4 getFoodX()

```
virtual int IGame::getFoodX ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

5.5.3.5 getFoodY()

```
virtual int IGame::getFoodY ( )  [pure virtual]
```

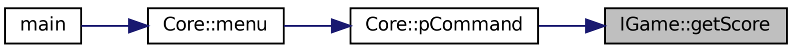
Implemented in Centipede, and Nibbler.

5.5.3.6 getScore()

```
virtual int IGame::getScore ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

Here is the caller graph for this function:



5.5.3.7 getSnakeXpos()

```
virtual std::vector<int> IGame::getSnakeXpos ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

5.5.3.8 getSnakeYpos()

```
virtual std::vector<int> IGame::getSnakeYpos ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

5.5.3.9 getXpos()

```
virtual int IGame::getXpos ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

5.5.3.10 getYpos()

```
virtual int IGame::getYpos ( )  [pure virtual]
```

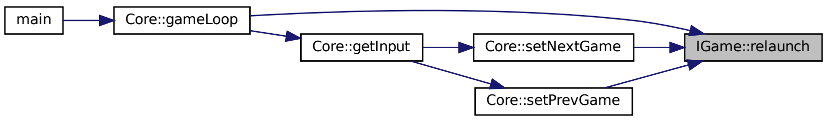
Implemented in Centipede, and Nibbler.

5.5.3.11 relaunch()

```
virtual void IGame::relaunch ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

Here is the caller graph for this function:

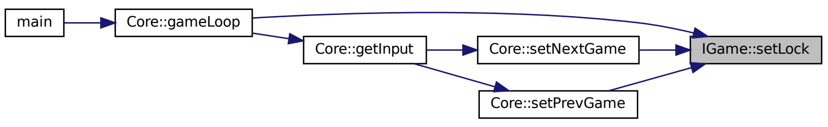


5.5.3.12 setLock()

```
virtual void IGame::setLock ( )  [pure virtual]
```

Implemented in Centipede, and Nibbler.

Here is the caller graph for this function:



The documentation for this class was generated from the following file:

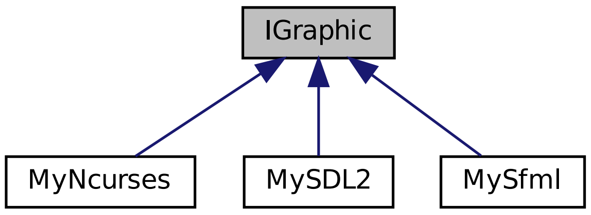
- game/igame.hpp

5.6 IGraphic Class Reference

Interface for graph class.

```
#include <igraphic.hpp>
```

Inheritance diagram for IGraphic:



Public Member Functions

- virtual ~IGraphic ()=default
- virtual void drawobj (std::vector< int > xcoords, std::vector< int > ycoords, int food_x, int food_y, int bullet_x, int bullet_y)=0
- virtual int getInput ()=0
- virtual void launch ()=0
- virtual void stop ()=0
- virtual void settingInput (int)=0

5.6.1 Detailed Description

Interface for graph class.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 ∼IGraphic()

```
virtual IGraphic::∼IGraphic ( )  [virtual], [default]
```

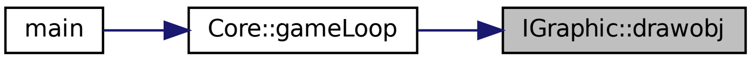
5.6.3 Member Function Documentation

5.6.3.1 drawobj()

```
virtual void IGraphic::drawobj (
             std::vector< int > xcoords,
             std::vector< int > ycoords,
             int food_x,
             int food_y,
             int bullet_x,
             int bullet_y )  [pure virtual]
```

Implemented in MySDL2, MySfml, and MyNcurses.

Here is the caller graph for this function:



5.6.3.2 getInput()

```
virtual int IGraphic::getInput ( )  [pure virtual]
```

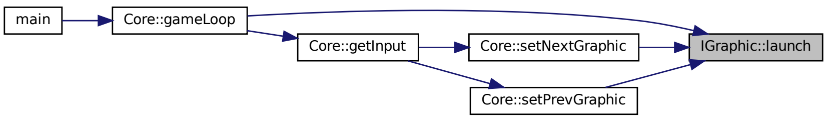
Implemented in MySDL2, MySfml, and MyNcurses.

5.6.3.3 launch()

```
virtual void IGraphic::launch ( )  [pure virtual]
```

Implemented in MySDL2, MyNcurses, and MySfml.

Here is the caller graph for this function:

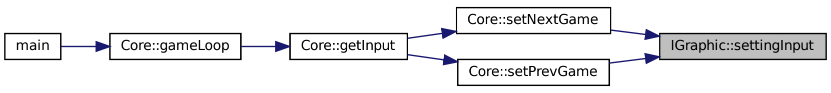


5.6.3.4 settingInput()

```
virtual void IGraphic::settingInput (
            int  )  [pure virtual]
```

Implemented in MySDL2, MySfml, and MyNcurses.

Here is the caller graph for this function:

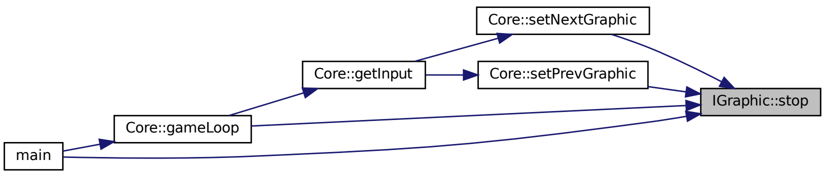


5.6.3.5 stop()

```
virtual void IGraphic::stop ( )  [pure virtual]
```

Implemented in MySDL2, MyNcurses, and MySfml.

Here is the caller graph for this function:



The documentation for this class was generated from the following file:

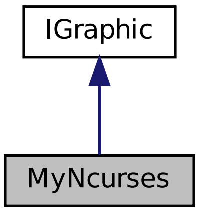
- graphic/igraphic.hpp

5.7 MyNcurses Class Reference

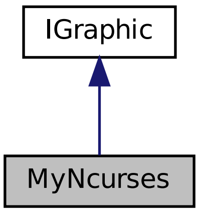
fucntion for draw game in ncurses use IGraphic interface

```
#include <ncurses.hpp>
```

Inheritance diagram for MyNcurses:



Collaboration diagram for MyNcurses:



Public Member Functions

- MyNcurses ()

  constructor of MyNcurses
- ∼MyNcurses ()

  deconstructor
- void init ()
- void stop ()

  strop sceen
- void launch ()

  Set symbol for print and init screen.
- int getInput ()

  get input and set _input
- void print_map ()

print ext map

- void settingInput (int input)

    set _input to given input

- void drawobj (std::vector< int > xcoords, std::vector< int > ycoords, int food_x, int food_y, int bullet_x, int bullet_y)

    draw snake, apple, bullet

Private Attributes

- int _input
- char rectangle
- char apple
- char bullet
- WINDOW ∗ _screen

5.7.1 Detailed Description

fucntion for draw game in ncurses use IGraphic interface

5.7.2 Constructor & Destructor Documentation

5.7.2.1 MyNcurses()

```
MyNcurses::MyNcurses ( )
```

constructor of MyNcurses

5.7.2.2 ∼MyNcurses()

```
MyNcurses::∼MyNcurses ( )
```

deconstructor

5.7.3 Member Function Documentation

5.7.3.1 drawobj()

```
void MyNcurses::drawobj (
            std::vector< int > xcoords,
            std::vector< int > ycoords,
            int food_x,
            int food_y,
            int bullet_x,
            int bullet_y ) [virtual]
```

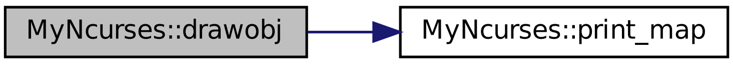
draw snake, apple, bullet

Parameters

in	*xcoord*	-> vector of x coor d of snake
in	*ycoords*	-> vector ycoords for snake
in	*food_x*	-> coord x of apple
in	*food_y*	-> coord y of apple
in	*bullet←_x*	-> coord x of bullet
in	*bullet←_y*	-> coord y of bullet

Implements IGraphic.

Here is the call graph for this function:



5.7.3.2 getInput()

```
int MyNcurses::getInput ( )  [virtual]
```

get input and set _input

Returns

_input

Implements IGraphic.

5.7.3.3 init()

```
void MyNcurses::init ( )
```

5.7.3.4 launch()

```
void MyNcurses::launch ( )  [virtual]
```

Set symbol for print and init screen.

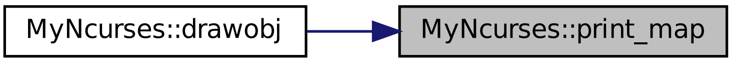
Implements IGraphic.

5.7.3.5 print_map()

```
void MyNcurses::print_map ( )
```

print ext map

Here is the caller graph for this function:



5.7.3.6 settingInput()

```
void MyNcurses::settingInput (
            int input )  [virtual]
```

set _input to given input

Parameters

in	*input*	-> new input

Implements IGraphic.

5.7.3.7 stop()

```
void MyNcurses::stop ( )  [virtual]
```

strop sceen

Implements IGraphic.

5.7.4 Member Data Documentation

5.7.4.1 _input

```
int MyNcurses::_input  [private]
```

5.7.4.2 _screen

```
WINDOW* MyNcurses::_screen  [private]
```

5.7.4.3 apple

```
char MyNcurses::apple  [private]
```

5.7.4.4 bullet

```
char MyNcurses::bullet  [private]
```

5.7.4.5 rectangle

```
char MyNcurses::rectangle  [private]
```

The documentation for this class was generated from the following files:

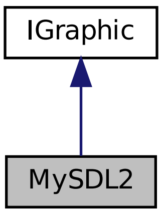
- graphic/ncurses/ncurses.hpp
- graphic/ncurses/ncurses.cpp

5.8 MySDL2 Class Reference

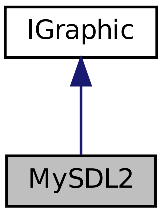
SDL2 function for draw game and handle input.

```
#include <MySDL2.hpp>
```

Inheritance diagram for MySDL2:



Collaboration diagram for MySDL2:



Public Member Functions

- MySDL2 ()

    constructor
- ∼MySDL2 ()

    deconstructor
- void init ()
- void stop ()

    destroy window and quit sdl
- void launch ()

    set window and render for launch window
- int getInput ()

    return _input
- void setInput ()

set _input to value set by key press
- void settingInput (int input)

  set _input to given value
- void drawSnake (std::vector< int > xcoords, std::vector< int > ycoords)

  draw snake at given coord
- void drawApple (int, int)

  draw apple
- void drawBullet (int, int)

  draw bullet
- void drawobj (std::vector< int > xcoords, std::vector< int > ycoords, int food_x, int food_y, int bullet_x, int bullet_y)

  call all function for draw game

Private Attributes

- int _input
- SDL_Event ∗ _event
- SDL_Window ∗ _screen
- SDL_Renderer ∗ _render

5.8.1 Detailed Description

SDL2 function for draw game and handle input.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 MySDL2()

```
MySDL2::MySDL2 ( )
```

constructor

5.8.2.2 ∼MySDL2()

```
MySDL2::∼MySDL2 ( )
```

deconstructor

5.8.3 Member Function Documentation

5.8.3.1 drawApple()

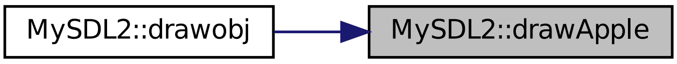
```
void MySDL2::drawApple (
             int food_x,
             int food_y )
```

draw apple

Parameters

in	*food←_x*	-> coord x of apple
in	*food←_y*	-> coord y of apple

Here is the caller graph for this function:



5.8.3.2 drawBullet()

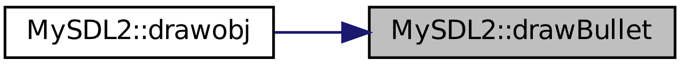
```
void MySDL2::drawBullet (
             int bullet_x,
             int bullet_y )
```

draw bullet

Parameters

in	*bullet←_x*	-> coord x of bullet
in	*bullet←_y*	-> coord y of bullet

Here is the caller graph for this function:



5.8.3.3 drawobj()

```
void MySDL2::drawobj (
             std::vector< int > xcoords,
             std::vector< int > ycoords,
             int food_x,
             int food_y,
             int bullet_x,
             int bullet_y )  [virtual]
```

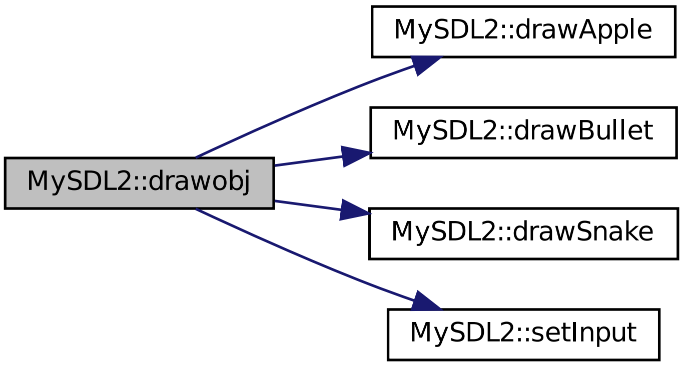
call all function for draw game

Parameters

in	*xcoord*	-> vector of x coor d of snake
in	*ycoords*	-> vector ycoords for snake
in	*food_x*	-> coord x of apple
in	*food_y*	-> coord y of apple
in	*bullet↩ _x*	-> coord x of bullet
in	*bullet↩ _y*	-> coord y of bullet

Implements IGraphic.

Here is the call graph for this function:



5.8.3.4 drawSnake()

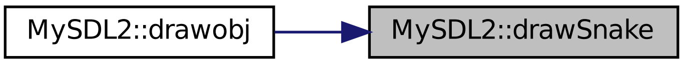
```
void MySDL2::drawSnake (
             std::vector< int > xcoords,
             std::vector< int > ycoords )
```

draw snake at given coord

Parameters

`in`	*xcoords*	-> vector x of coord of snake]
`in`	*ycoords*	-> vector of coord y of snake

Here is the caller graph for this function:



5.8.3.5 getInput()

```
int MySDL2::getInput ( )  [virtual]
```

return _input

Returns

_input

Implements IGraphic.

5.8.3.6 init()

```
void MySDL2::init ( )
```

5.8.3.7 launch()

```
void MySDL2::launch ( )  [virtual]
```

set window and render for launch window

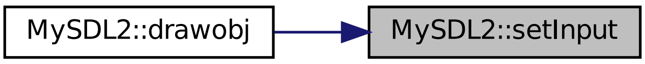
Implements IGraphic.

5.8.3.8 setInput()

```
void MySDL2::setInput ( )
```

set _input to value set by key press

Here is the caller graph for this function:



5.8.3.9 settingInput()

```
void MySDL2::settingInput (
             int input )  [virtual]
```

set _input to given value

Parameters

in	*input->*	new value to set

Implements IGraphic.

5.8.3.10 stop()

```
void MySDL2::stop ( )  [virtual]
```

destroy window and quit sdl

Implements IGraphic.

5.8.4 Member Data Documentation

5.8.4.1 _event

```
SDL_Event* MySDL2::_event  [private]
```

5.8.4.2 _input

```
int MySDL2::_input  [private]
```

5.8.4.3 _render

```
SDL_Renderer* MySDL2::_render  [private]
```

5.8.4.4 _screen

```
SDL_Window* MySDL2::_screen  [private]
```

The documentation for this class was generated from the following files:

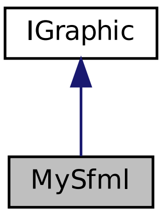
- graphic/sdl_deux/MySDL2.hpp
- graphic/sdl_deux/MySDL2.cpp

5.9 MySfml Class Reference

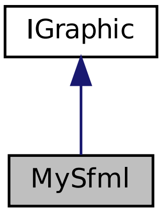
SFML function for draw game use IGraphic interface.

```
#include <sfmldraw.hpp>
```

Inheritance diagram for MySfml:



Collaboration diagram for MySfml:



Public Member Functions

- MySfml ()

  constructor
- ∼MySfml ()

  deconstructor
- void init ()
- void stop ()

  close window
- void launch ()

  set screen and element to draw
- int getInput ()

  return _input
- void setInput (sf::Event event)

  set _input with correspondant key pressed
- void settingInput (int input)

  set _input to given input
- void drawobj (std::vector< int > xcoords, std::vector< int > ycoords, int food_x, int food_y, int bullet_x, int bullet_y)

  draw and set position of obj of game

Private Attributes

- int _input
- sf::RenderWindow ∗ screen
- sf::RectangleShape ∗ rectangle
- sf::CircleShape ∗ apple
- sf::CircleShape ∗ bullet

5.9.1 Detailed Description

SFML function for draw game use IGraphic interface.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 MySfml()

```
MySfml::MySfml ( )
```

constructor

5.9.2.2 ∼MySfml()

```
MySfml::∼MySfml ( )
```

deconstructor

5.9.3 Member Function Documentation

5.9.3.1 drawobj()

```
void MySfml::drawobj (
            std::vector< int > xcoords,
            std::vector< int > ycoords,
            int food_x,
            int food_y,
            int bullet_x,
            int bullet_y ) [virtual]
```

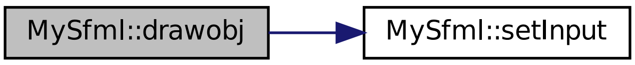
draw and set position of obj of game

Parameters

in	*xcoord*	-> vector of x coor d of snake
in	*ycoords*	-> vector ycoords for snake
in	*food_x*	-> coord x of apple
in	*food_y*	-> coord y of apple
in	*bullet←↩ _x*	-> coord x of bullet
in	*bullet←↩ _y*	-> coord y of bullet

Implements IGraphic.

Here is the call graph for this function:



5.9.3.2 getInput()

```
int MySfml::getInput ( )  [virtual]
```

return _input

Returns

_input

Implements IGraphic.

5.9.3.3 init()

```
void MySfml::init ( )
```

5.9.3.4 launch()

```
void MySfml::launch ( )  [virtual]
```

set screen and element to draw

Implements IGraphic.

5.9.3.5 setInput()

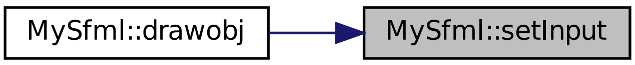
```
void MySfml::setInput (
             sf::Event event )
```

set _input with correspondant key pressed

Parameters

in	*event*	-> event of key press

Here is the caller graph for this function:



5.9.3.6 settingInput()

```
void MySfml::settingInput (
             int input )  [virtual]
```

set _input to given input

Parameters

in	*input*	-> new input

Implements IGraphic.

5.9.3.7 stop()

```
void MySfml::stop ( )  [virtual]
```

close window

Implements IGraphic.

5.9.4 Member Data Documentation

5.9.4.1 _input

```
int MySfml::_input  [private]
```

5.9.4.2 apple

```
sf::CircleShape* MySfml::apple  [private]
```

5.9.4.3 bullet

```
sf::CircleShape* MySfml::bullet  [private]
```

5.9.4.4 rectangle

```
sf::RectangleShape* MySfml::rectangle  [private]
```

5.9.4.5 screen

```
sf::RenderWindow* MySfml::screen  [private]
```

The documentation for this class was generated from the following files:

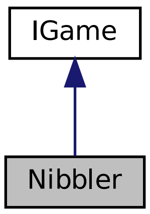
- graphic/sfml/sfmldraw.hpp
- graphic/sfml/sfmldraw.cpp

5.10 Nibbler Class Reference

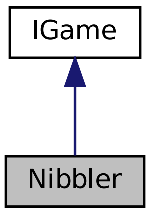
Game use IGame as Interface.

```
#include <nibbler.hpp>
```

Inheritance diagram for Nibbler:



Collaboration diagram for Nibbler:



Public Member Functions

- Nibbler (int a, int b)
- ∼Nibbler ()
- int getXpos ()
- int getYpos ()
- std::vector< int > getSnakeXpos ()
- std::vector< int > getSnakeYpos ()
- int getFoodY ()
- int getFoodX ()
- int getBulletY ()
- int getBulletX ()
- std::list< Nibbler > getSnake ()
- int checkInputs (int dir)

  check inputs
- void setLock ()

  set lock to false for good restart of game
- void relaunch ()

  relaunch correctly game
- int getScore ()

  return score

Private Member Functions

- void createSnake ()

  createSnake set all var
- void changeFoodCoord (int x, int y)

  change coord of apple by rand
- int changeDir (Nibbler logic, int dir)

  change direction of snake
- int checkSnake (int dir, int xx)

  check snake and change food coord

Private Attributes

- int x
- int y
- std::list< Nibbler > snakes
- std::vector< int > xcoords
- std::vector< int > ycoords
- int food_x
- int food_y
- bool lock
- int score

5.10.1 Detailed Description

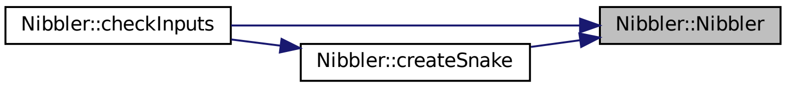
Game use IGame as Interface.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 Nibbler()

```
Nibbler::Nibbler (
            int a,
            int b )  [inline]
```

Here is the caller graph for this function:



5.10.2.2 ∼Nibbler()

```
Nibbler::∼Nibbler ( )  [inline]
```

5.10.3 Member Function Documentation

5.10.3.1 changeDir()

```
int Nibbler::changeDir (
            Nibbler logic,
            int dir )  [private]
```

change direction of snake

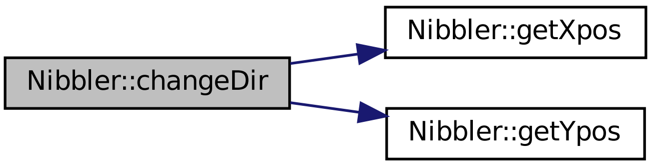
Parameters

in	*logic*	
in	*dir*	-> direction

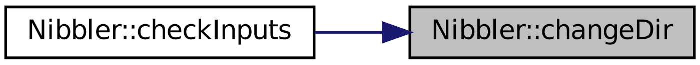
Returns

0 or 1

Here is the call graph for this function:



Here is the caller graph for this function:



5.10.3.2 changeFoodCoord()

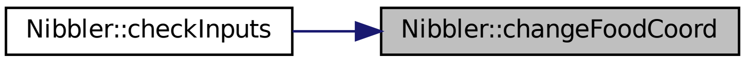
```
void Nibbler::changeFoodCoord (
             int x,
             int y )  [private]
```

change coord of apple by rand

Parameters

in	*x*	
in	*y*	

Here is the caller graph for this function:



5.10.3.3 checkInputs()

```
int Nibbler::checkInputs (
            int dir )  [virtual]
```

check inputs

Parameters

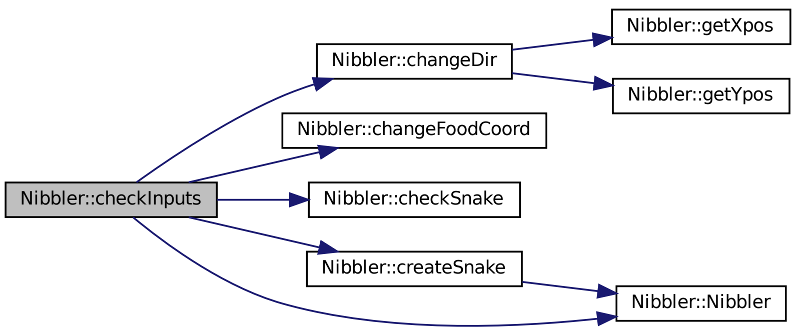
in	*direction*	

Returns

0 or 2

Implements IGame.

Here is the call graph for this function:



5.10.3.4 checkSnake()

```
int Nibbler::checkSnake (
             int dir,
             int xx )  [private]
```

check snake and change food coord

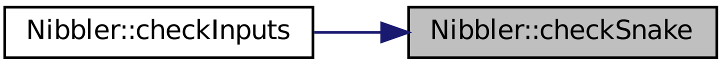
Parameters

in	*dir*	for direction of snake
in	*xx*	??

Returns

0 or 2

Here is the caller graph for this function:

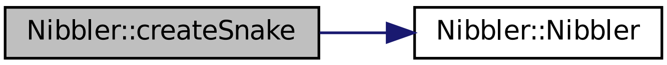

$$\begin{tikzcd}\text{Nibbler::checkInputs} \arrow[r] & \boxed{\text{Nibbler::checkSnake}}\end{tikzcd}$$

5.10.3.5 createSnake()

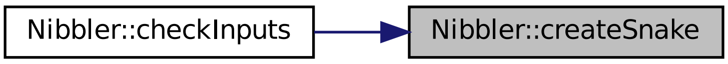
```
void Nibbler::createSnake ( )  [private]
```

createSnake set all var

Here is the call graph for this function:


$$\begin{tikzcd}\boxed{\text{Nibbler::createSnake}} \arrow[r] & \text{Nibbler::Nibbler}\end{tikzcd}$$

Here is the caller graph for this function:



5.10.3.6 getBulletX()

```
int Nibbler::getBulletX ( )  [inline], [virtual]
```

Implements IGame.

5.10.3.7 getBulletY()

```
int Nibbler::getBulletY ( )  [inline], [virtual]
```

Implements IGame.

5.10.3.8 getFoodX()

```
int Nibbler::getFoodX ( )  [inline], [virtual]
```

Implements IGame.

5.10.3.9 getFoodY()

```
int Nibbler::getFoodY ( )  [inline], [virtual]
```

Implements IGame.

5.10.3.10 getScore()

```
int Nibbler::getScore ( ) [virtual]
```

return score

Returns

score

Implements IGame.

5.10.3.11 getSnake()

```
std::list<Nibbler> Nibbler::getSnake ( ) [inline]
```

5.10.3.12 getSnakeXpos()

```
std::vector<int> Nibbler::getSnakeXpos ( ) [inline], [virtual]
```

Implements IGame.

5.10.3.13 getSnakeYpos()

```
std::vector<int> Nibbler::getSnakeYpos ( ) [inline], [virtual]
```

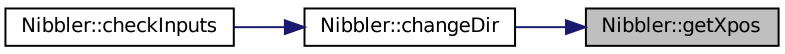
Implements IGame.

5.10.3.14 getXpos()

```
int Nibbler::getXpos ( ) [inline], [virtual]
```

Implements IGame.

Here is the caller graph for this function:

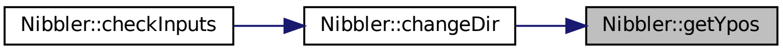


5.10.3.15 getYpos()

```
int Nibbler::getYpos ( ) [inline], [virtual]
```

Implements IGame.

Here is the caller graph for this function:



5.10.3.16 relaunch()

```
void Nibbler::relaunch ( ) [virtual]
```

relaunch correctly game

Implements IGame.

5.10.3.17 setLock()

```
void Nibbler::setLock ( ) [virtual]
```

set lock to false for good restart of game

Implements IGame.

5.10.4 Member Data Documentation

5.10.4.1 food_x

```
int Nibbler::food_x [private]
```

5.10.4.2 food_y

```
int Nibbler::food_y  [private]
```

5.10.4.3 lock

```
bool Nibbler::lock  [private]
```

5.10.4.4 score

```
int Nibbler::score  [private]
```

5.10.4.5 snakes

```
std::list<Nibbler> Nibbler::snakes  [private]
```

5.10.4.6 x

```
int Nibbler::x  [private]
```

5.10.4.7 xcoords

```
std::vector<int> Nibbler::xcoords  [private]
```

5.10.4.8 y

```
int Nibbler::y  [private]
```

5.10.4.9 ycoords

```
std::vector<int> Nibbler::ycoords  [private]
```

The documentation for this class was generated from the following files:

- game/nibbler/nibbler.hpp
- game/nibbler/nibbler.cpp

Chapter 6

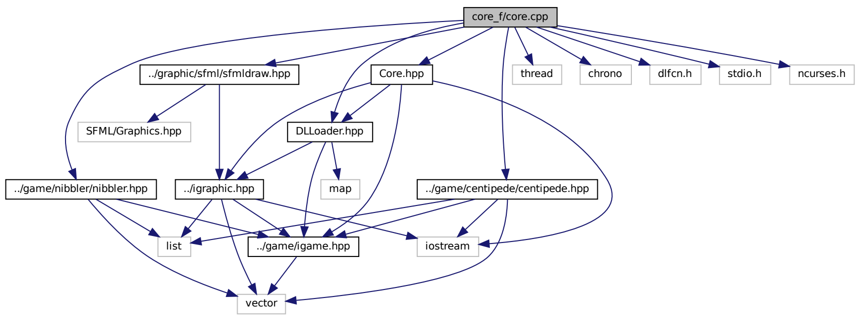
File Documentation

6.1 core_f/core.cpp File Reference

File for core Function : change Game or Lib.

```
#include "../graphic/sfml/sfmldraw.hpp"
#include "../game/nibbler/nibbler.hpp"
#include "../game/centipede/centipede.hpp"
#include "DLLoader.hpp"
#include "Core.hpp"
#include <thread>
#include <chrono>
#include <dlfcn.h>
#include <stdio.h>
#include <ncurses.h>
```

Include dependency graph for core.cpp:



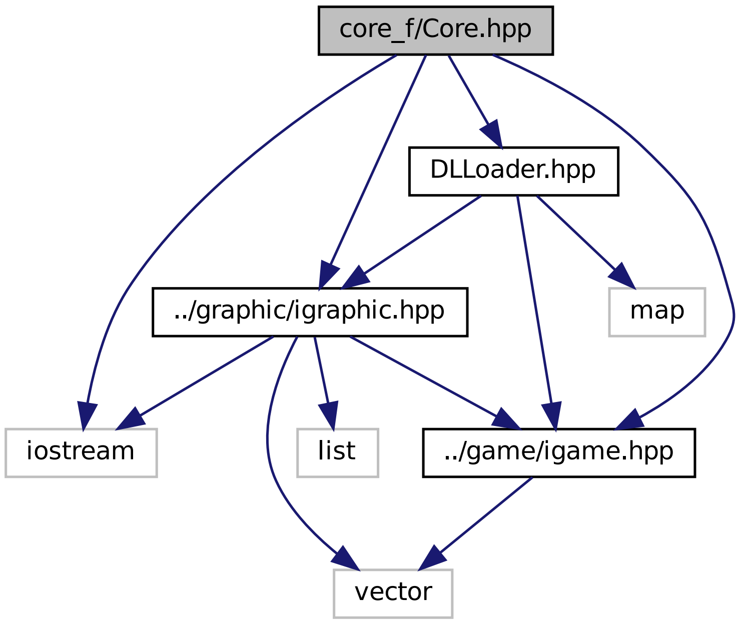
6.1.1 Detailed Description

File for core Function : change Game or Lib.

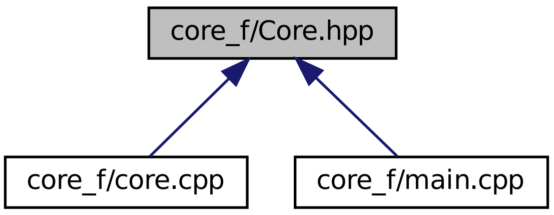
6.2 core_f/Core.hpp File Reference

```
#include <iostream>
#include "../game/igame.hpp"
#include "../graphic/igraphic.hpp"
#include "DLLoader.hpp"
```

Include dependency graph for Core.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class Core

  Core create bridge between game and graphic also do main menu.

Functions

- std::vector< std::string > get_lib (int i)

    find in directory new lib
- int gameLoop (DLLoader loader, Core core)

6.2.1 Function Documentation

6.2.1.1 gameLoop()

```
int gameLoop (
            DLLoader loader,
            Core core )
```

6.2.1.2 get_lib()

```
std::vector<std::string> get_lib (
            int i )
```

find in directory new lib

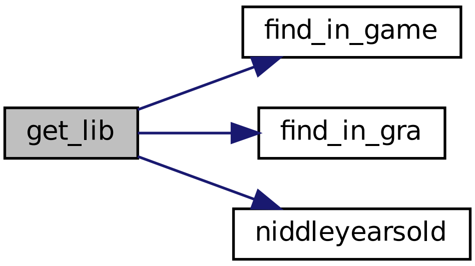
Parameters

in	*i*	-> 1 for game 2 for graph

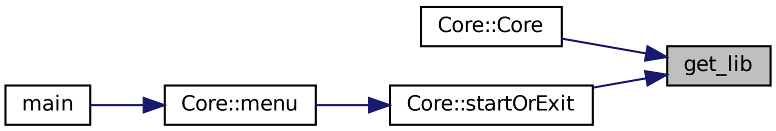
Returns

files -> name of new lib

Here is the call graph for this function:



Here is the caller graph for this function:

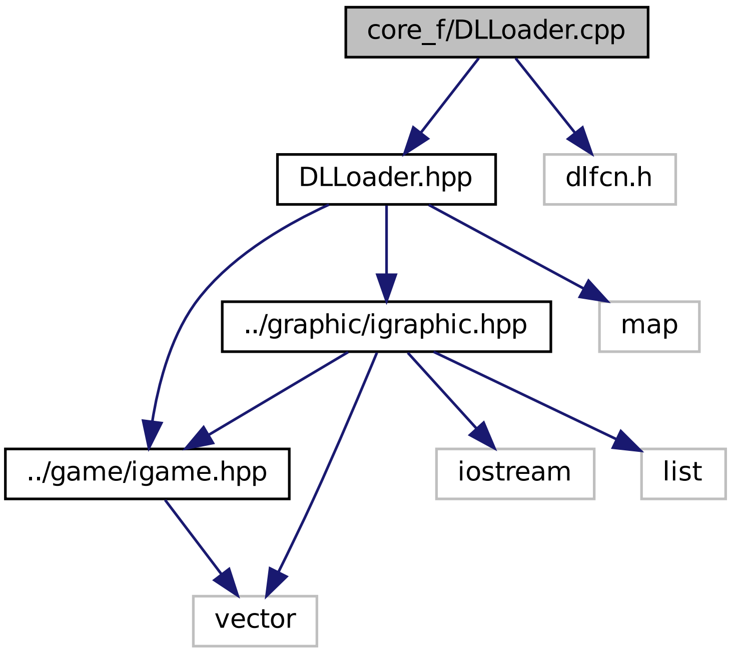


6.3 core_f/DLLoader.cpp File Reference

DLLoader function for create Instance of IGame or IGraphic and manage it.

```
#include "DLLoader.hpp"
#include <dlfcn.h>
```

Include dependency graph for DLLoader.cpp:



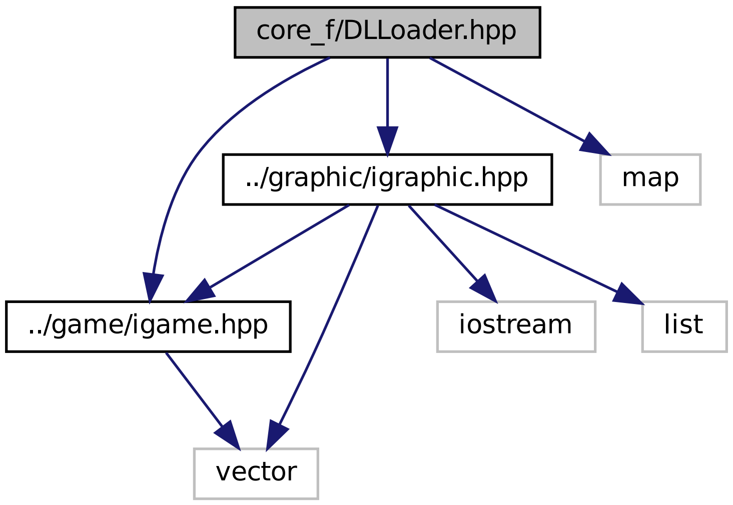
6.3.1 Detailed Description

DLLoader function for create Instance of IGame or IGraphic and manage it.

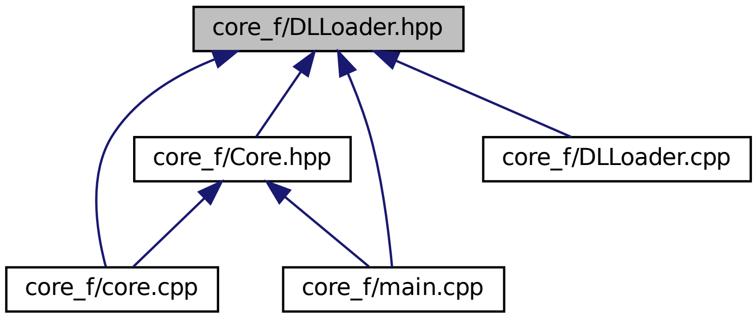
6.4 core_f/DLLoader.hpp File Reference

```
#include "../game/igame.hpp"
#include "../graphic/igraphic.hpp"
#include <map>
```

Include dependency graph for DLLoader.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class DLLoader

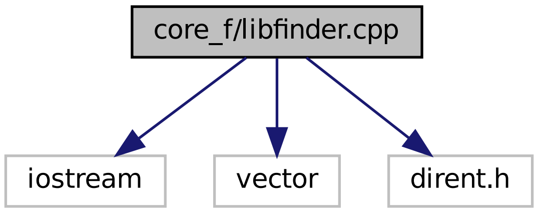
  Load .so and create Instance of Interface.

6.5 core_f/libfinder.cpp File Reference

Function for add new lib in actual lib vector.

```
#include <iostream>
#include <vector>
#include <dirent.h>
```

Include dependency graph for libfinder.cpp:



Functions

- int niddleyearsold (std::string name)

  need help to known what function do
- int find_in_gra (std::string name)

  check if given name is one of graphics libs we wanted
- int find_in_game (std::string name)

  check if given name is one of graphics games we wanted
- std::vector< std::string > get_lib (int i)

  find in directory new lib

6.5.1 Detailed Description

Function for add new lib in actual lib vector.

6.5.2 Function Documentation

6.5.2.1 find_in_game()

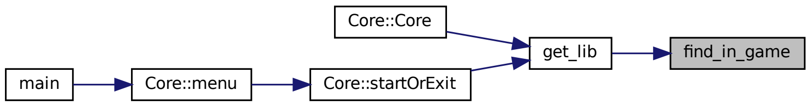
```
int find_in_game (
            std::string name )
```

check if given name is one of graphics games we wanted

Parameters

in	*name*	-> name of game

Here is the caller graph for this function:



6.5.2.2 find_in_gra()

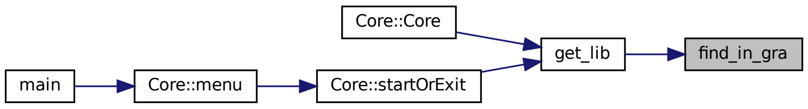
```
int find_in_gra (
            std::string name )
```

check if given name is one of graphics libs we wanted

Parameters

in	*name*	-> name of lib

Here is the caller graph for this function:



6.5.2.3 get_lib()

```
std::vector<std::string> get_lib (
            int i )
```

find in directory new lib

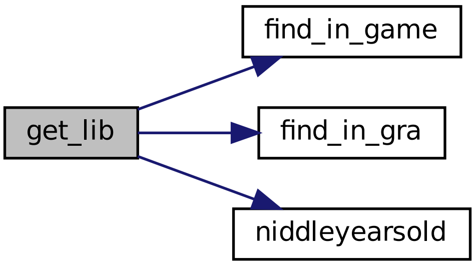
Parameters

in	*i*	-> 1 for game 2 for graph

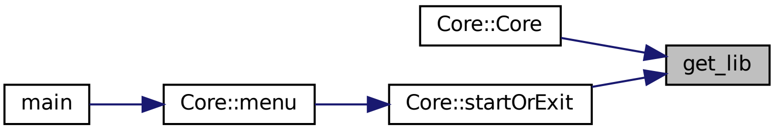
Returns

files -> name of new lib

Here is the call graph for this function:



Here is the caller graph for this function:



6.5.2.4 niddleyearsold()

```
int niddleyearsold (
            std::string name )
```

need help to known what function do

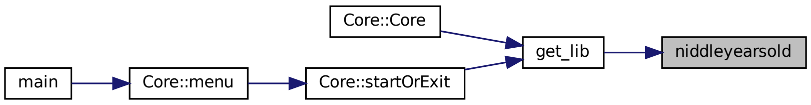
Parameters

in	*name*	

Returns

0 is not handle lib 2 if contains .so and arcade

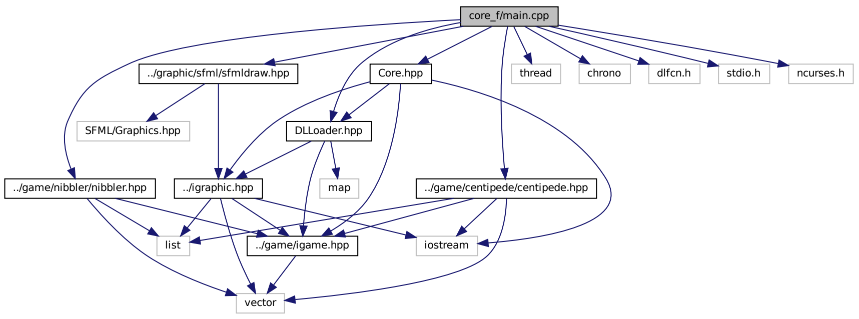
Here is the caller graph for this function:



6.6 core_f/main.cpp File Reference

```
#include "../graphic/sfml/sfmldraw.hpp"
#include "../game/nibbler/nibbler.hpp"
#include "../game/centipede/centipede.hpp"
#include "DLLoader.hpp"
#include "Core.hpp"
#include <thread>
#include <chrono>
#include <dlfcn.h>
#include <stdio.h>
#include <ncurses.h>
```

Include dependency graph for main.cpp:



Functions

- int main (int ac, char ∗∗av)

  main

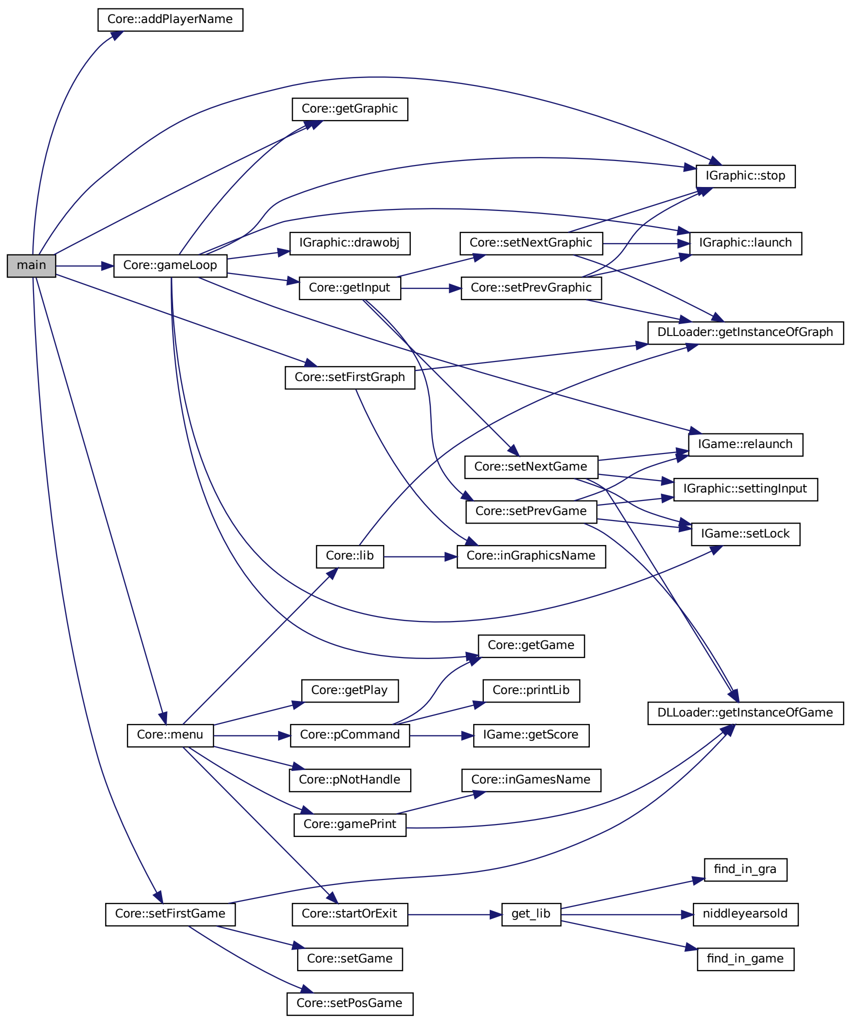
6.6.1 Function Documentation

6.6.1.1 main()

```
int main (
            int ac,
            char ** av )
```

main

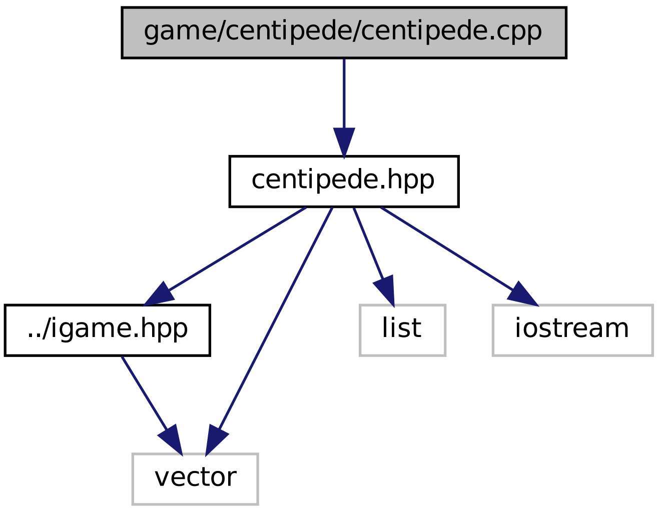
Here is the call graph for this function:



6.7 game/centipede/centipede.cpp File Reference

Function for Centipede Game.

```
#include "centipede.hpp"
```
Include dependency graph for centipede.cpp:



Functions

- void __attribute__ ((constructor)) calledFirst()
- void __attribute__ ((destructor)) calledLast()
- IGame ∗ entryPoint ()

  entryPoint -> for dynamic lib
- void calledFirst ()

  function call during construction
- void calledLast ()

  function call during desctruction

6.7.1 Detailed Description

Function for Centipede Game.

6.7.2 Function Documentation

6.7.2.1 __attribute__() [1/2]

```
void __attribute__ (
            (constructor)  )
```

6.7.2.2 __attribute__() [2/2]

```
void __attribute__ (
             (destructor)  )
```

6.7.2.3 calledFirst()

```
void calledFirst ( )
```

function call during construction

6.7.2.4 calledLast()

```
void calledLast ( )
```

function call during desctruction

6.7.2.5 entryPoint()

```
IGame* entryPoint ( )
```

entryPoint -> for dynamic lib

Returns

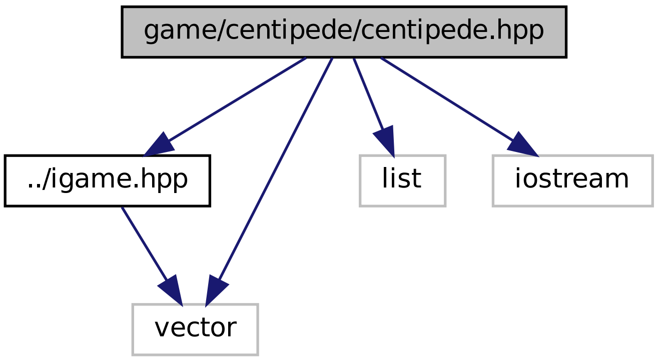
nib -> Instance of IGame who is Centipede

6.8 game/centipede/centipede.hpp File Reference

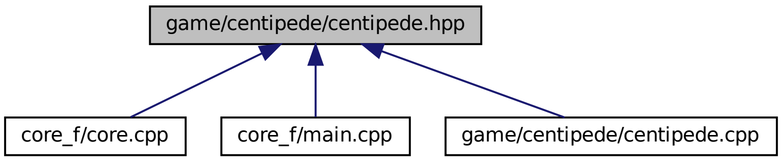
```
#include "../igame.hpp"
#include <list>
#include <vector>
```

```
#include <iostream>
```

Include dependency graph for centipede.hpp:



This graph shows which files directly or indirectly include this file:



Classes

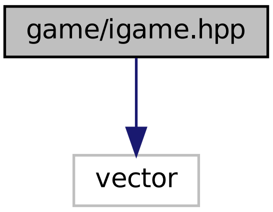
- class Centipede

  Centipede class for game Centipede use IGame for Interface.
- class Bullet

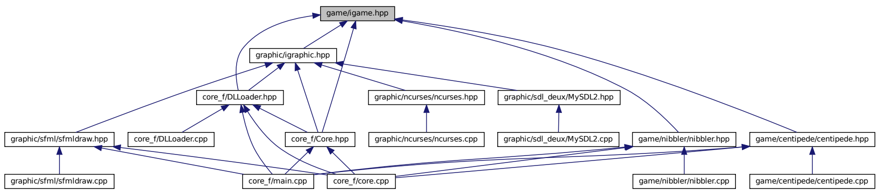
6.9 game/igame.hpp File Reference

```
#include <vector>
```

Include dependency graph for igame.hpp:



This graph shows which files directly or indirectly include this file:


game/igame.hpp
graphic/igraphic.hpp
core_f/DLLoader.hpp
graphic/ncurses/ncurses.hpp
graphic/sdl_deux/MySDL2.hpp
graphic/sfml/sfmldraw.hpp
core_f/DLLoader.cpp
core_f/Core.hpp
graphic/ncurses/ncurses.cpp
graphic/sdl_deux/MySDL2.cpp
game/nibbler/nibbler.hpp
game/centipede/centipede.hpp
graphic/sfml/sfmldraw.cpp
core_f/main.cpp
core_f/core.cpp
game/nibbler/nibbler.cpp
game/centipede/centipede.cpp

Classes

- class IGame

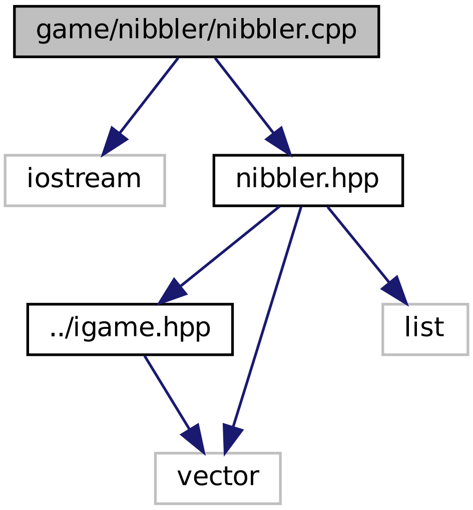
    Interface for game.

6.10 game/nibbler/nibbler.cpp File Reference

Function for game nibbler.

```
#include <iostream>
#include "nibbler.hpp"
```

Include dependency graph for nibbler.cpp:



Functions

- void __attribute__ ((constructor)) calledFirst()
- void __attribute__ ((destructor)) calledLast()
- IGame ∗ entryPoint ()

  entryPoint -> for dynamic lib
- void calledFirst ()

  function call during construction
- void calledLast ()

  function call during deconstruction

6.10.1 Detailed Description

Function for game nibbler.

6.10.2 Function Documentation

6.10.2.1 __attribute__() [1/2]

```
void __attribute__ (
                (constructor) )
```

6.10.2.2 __attribute__() [2/2]

```
void __attribute__ (
             (destructor)  )
```

6.10.2.3 calledFirst()

```
void calledFirst ( )
```

function call during construction

6.10.2.4 calledLast()

```
void calledLast ( )
```

function call during deconstruction

6.10.2.5 entryPoint()

```
IGame* entryPoint ( )
```

entryPoint -> for dynamic lib

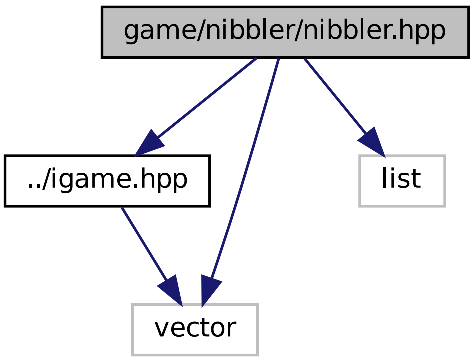
Returns

nib -> Instance of IGame who is Nibble

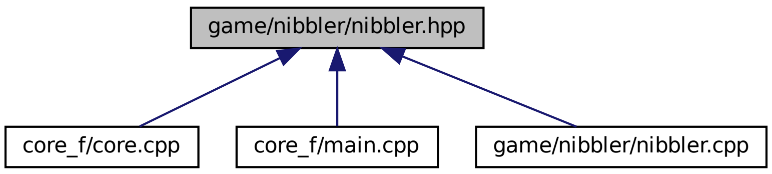
6.11 game/nibbler/nibbler.hpp File Reference

```
#include "../igame.hpp"
#include <list>
#include <vector>
```

Include dependency graph for nibbler.hpp:



This graph shows which files directly or indirectly include this file:



Classes

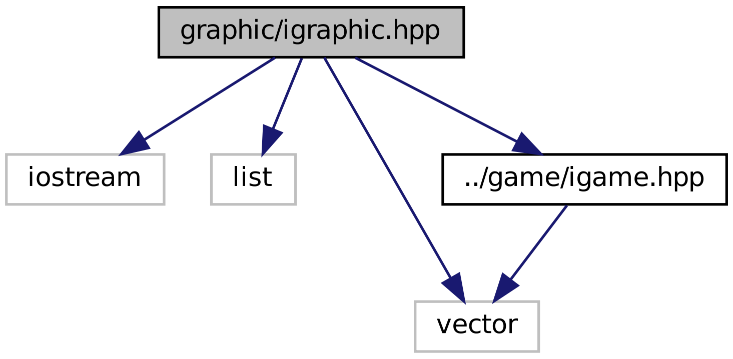
- class Nibbler

  Game use IGame as Interface.

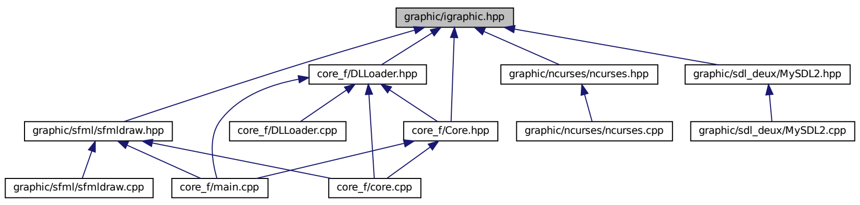
6.12 graphic/igraphic.hpp File Reference

```
#include <iostream>
#include <list>
#include <vector>
#include "../game/igame.hpp"
```

Include dependency graph for igraphic.hpp:



This graph shows which files directly or indirectly include this file:


graphic/igraphic.hpp
core_f/DLLoader.hpp
graphic/ncurses/ncurses.hpp
graphic/sdl_deux/MySDL2.hpp
graphic/sfml/sfmldraw.hpp
core_f/DLLoader.cpp
core_f/Core.hpp
graphic/ncurses/ncurses.cpp
graphic/sdl_deux/MySDL2.cpp
graphic/sfml/sfmldraw.cpp
core_f/main.cpp
core_f/core.cpp

Classes

- class IGraphic

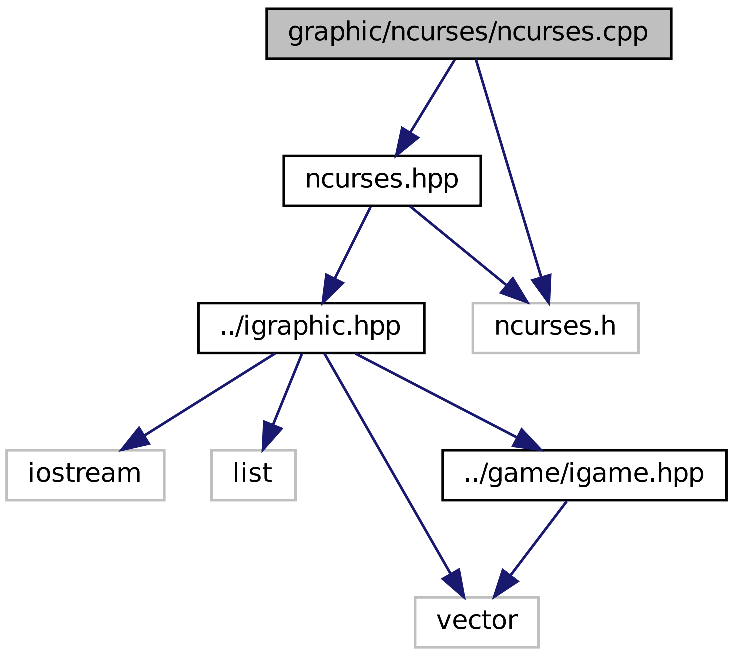
  Interface for graph class.

6.13 graphic/ncurses/ncurses.cpp File Reference

all function for ncurses print of game infos

```
#include "ncurses.hpp"
#include <ncurses.h>
```

Include dependency graph for ncurses.cpp:



Functions

- void __attribute__ ((constructor)) calledFirst()
- void __attribute__ ((destructor)) calledLast()
- IGraphic ∗ entryPoint ()

  entryPoint
- void calledFirst ()

  function call during constructor
- void calledLast ()

  function call during desconstructor

6.13.1 Detailed Description

all function for ncurses print of game infos

6.13.2 Function Documentation

6.13.2.1 __attribute__() `[1/2]`

```
void __attribute__ (
             (constructor)  )
```

6.13.2.2 __attribute__() `[2/2]`

```
void __attribute__ (
             (destructor)  )
```

6.13.2.3 calledFirst()

```
void calledFirst ( )
```

function call during constructor

6.13.2.4 calledLast()

```
void calledLast ( )
```

function call during desconstructor

6.13.2.5 entryPoint()

```
IGraphic* entryPoint ( )
```

entryPoint

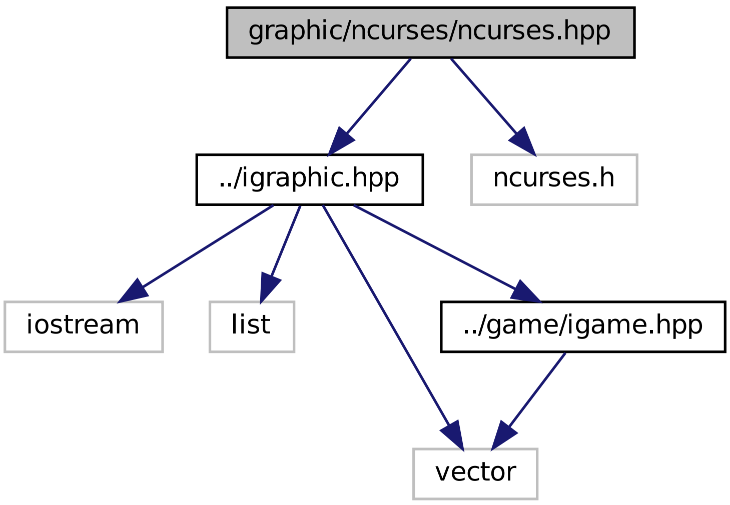
Returns

instance of IGraphic

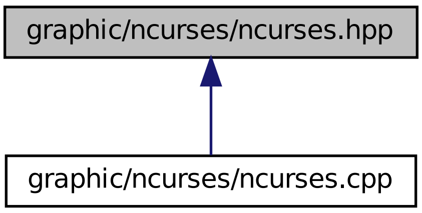
6.14 graphic/ncurses/ncurses.hpp File Reference

```
#include "../igraphic.hpp"
#include <ncurses.h>
```

Include dependency graph for ncurses.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class MyNcurses

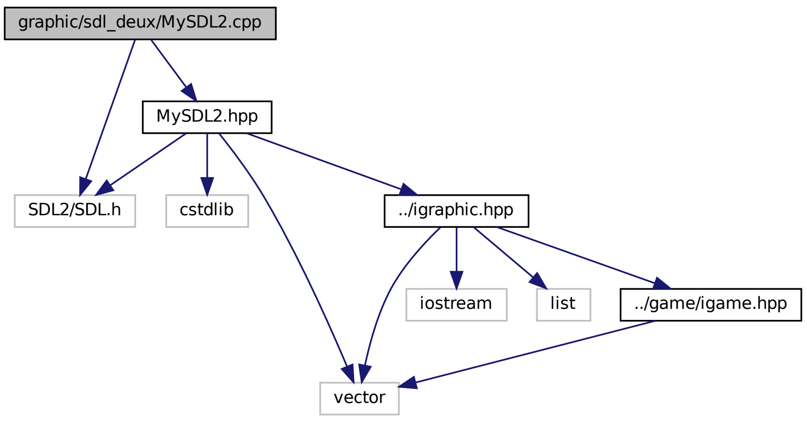
  fucntion for draw game in ncurses use IGraphic interface

6.15 graphic/sdl_deux/MySDL2.cpp File Reference

SDL2 Function.

```
#include <SDL2/SDL.h>
#include "MySDL2.hpp"
```

Include dependency graph for MySDL2.cpp:



Functions

- void __attribute__ ((constructor)) calledFirst()
- void __attribute__ ((destructor)) calledLast()
- IGraphic $*$ entryPoint ()

  entryPoint
- void calledFirst ()

  function call during constructor
- void calledLast ()

  function call during desconstructor

6.15.1 Detailed Description

SDL2 Function.

6.15.2 Function Documentation

6.15.2.1 __attribute__() [1/2]

```
void __attribute__ (
             (constructor) )
```

6.15.2.2 __attribute__() [2/2]

```
void __attribute__ (
             (destructor) )
```

6.15.2.3 calledFirst()

```
void calledFirst ( )
```

function call during constructor

6.15.2.4 calledLast()

```
void calledLast ( )
```

function call during desconstructor

6.15.2.5 entryPoint()

```
IGraphic* entryPoint ( )
```

entryPoint

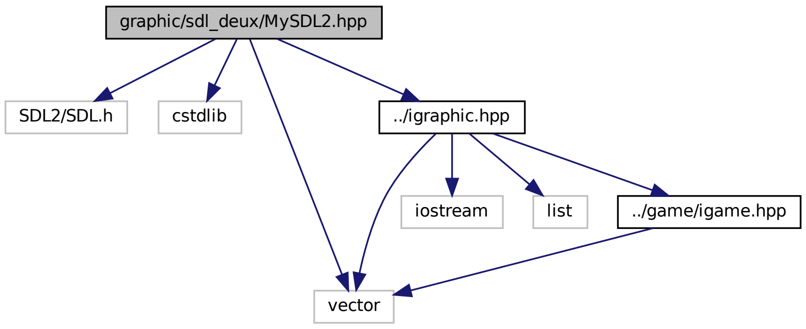
Returns

instance of IGraphic

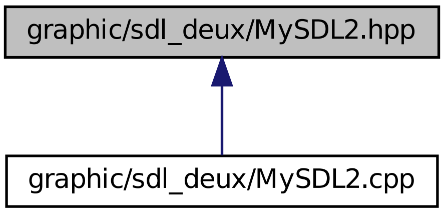
6.16 graphic/sdl_deux/MySDL2.hpp File Reference

```
#include <SDL2/SDL.h>
#include <cstdlib>
#include <vector>
#include "../igraphic.hpp"
```

Include dependency graph for MySDL2.hpp:



This graph shows which files directly or indirectly include this file:



Classes

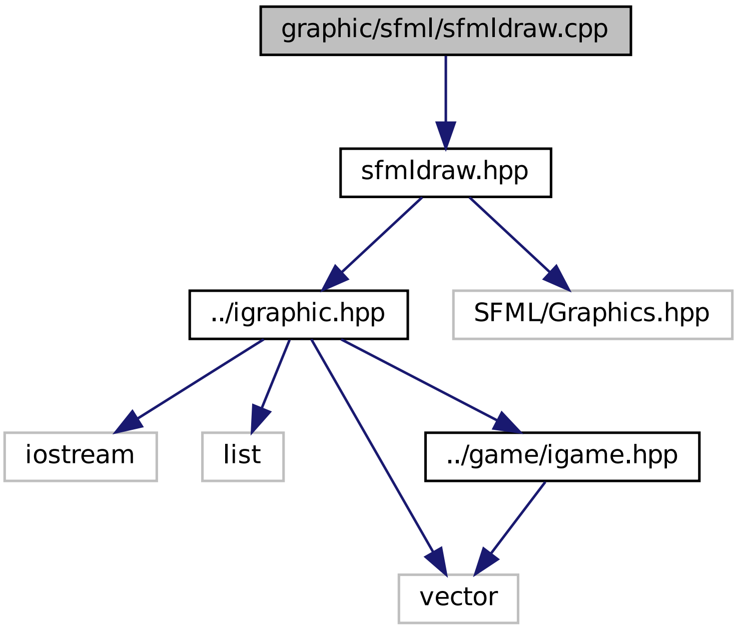
- class MySDL2

  SDL2 function for draw game and handle input.

6.17 graphic/sfml/sfmldraw.cpp File Reference

Function for draw with SFML.

```
#include "sfmldraw.hpp"
```
Include dependency graph for sfmldraw.cpp:



Functions

- void __attribute__ ((constructor)) calledFirst()
- void __attribute__ ((destructor)) calledLast()
- IGraphic * entryPoint ()

  entryPoint
- void calledFirst ()

  function call during constructor
- void calledLastgra ()

  function call during desconstructor

6.17.1 Detailed Description

Function for draw with SFML.

6.17.2 Function Documentation

6.17.2.1 __attribute__() [1/2]

```
void __attribute__ (
             (constructor)  )
```

6.17.2.2 __attribute__() [2/2]

```
void __attribute__ (
             (destructor)  )
```

6.17.2.3 calledFirst()

```
void calledFirst ( )
```

function call during constructor

6.17.2.4 calledLastgra()

```
void calledLastgra ( )
```

function call during desconstructor

6.17.2.5 entryPoint()

```
IGraphic* entryPoint ( )
```

entryPoint

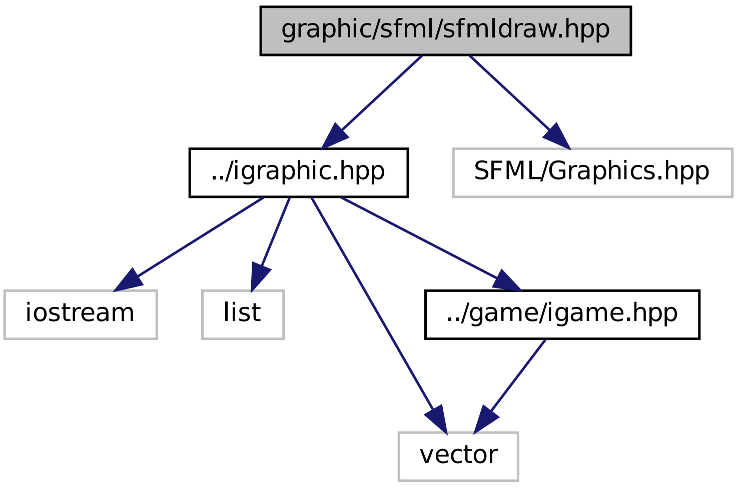
Returns

instance of IGraphic

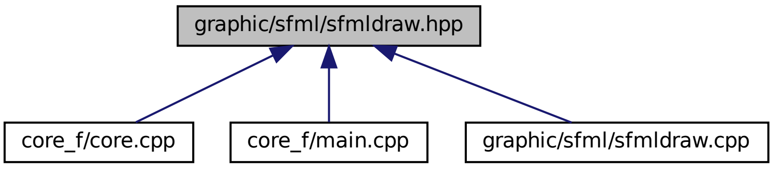
6.18 graphic/sfml/sfmldraw.hpp File Reference

```
#include "../igraphic.hpp"
#include <SFML/Graphics.hpp>
```

Include dependency graph for sfmldraw.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class MySfml

  SFML function for draw game use IGraphic interface.

Index

__attribute__
    centipede.cpp, 93
    MySDL2.cpp, 103, 104
    ncurses.cpp, 101
    nibbler.cpp, 97
    sfmldraw.cpp, 106, 107
_actualGame
    Core, 41
_actualLib
    Core, 41
_command
    Core, 41
_event
    MySDL2, 66
_game
    Core, 41
_games
    DLLoader, 48
_gamesName
    Core, 41
    DLLoader, 48
_gamesNb
    DLLoader, 48
_graphName
    Core, 42
_graphic
    Core, 41
_graphics
    DLLoader, 48
_graphicsName
    DLLoader, 48
_graphicsNb
    DLLoader, 49
_handles
    DLLoader, 49
_input
    MyNcurses, 60
    MySDL2, 67
    MySfml, 71
_maxGame
    Core, 42
_maxGraph
    Core, 42
_play
    Core, 42
_playerName
    Core, 42
_posGame
    Core, 42
_posGraph
    Core, 42
_render
    MySDL2, 67
_screen
    MyNcurses, 60
    MySDL2, 67
_x
    Bullet, 10
_y
    Bullet, 10
∼Bullet
    Bullet, 9
∼Centipede
    Centipede, 12
∼Core
    Core, 22
∼DLLoader
    DLLoader, 44
∼IGame
    IGame, 50
∼IGraphic
    IGraphic, 54
∼MyNcurses
    MyNcurses, 57
∼MySDL2
    MySDL2, 62
∼MySfml
    MySfml, 69
∼Nibbler
    Nibbler, 74

addPlayerName
    Core, 22
apple
    MyNcurses, 60
    MySfml, 71

Bullet, 9
    _x, 10
    _y, 10
    ∼Bullet, 9
    Bullet, 9
bullet
    MyNcurses, 60
    MySfml, 72
bullet_x
    Centipede, 18
bullet_y
    Centipede, 18

calledFirst
  centipede.cpp, 94
  MySDL2.cpp, 104
  ncurses.cpp, 101
  nibbler.cpp, 98
  sfmldraw.cpp, 107
calledLast
  centipede.cpp, 94
  MySDL2.cpp, 104
  ncurses.cpp, 101
  nibbler.cpp, 98
calledLastgra
  sfmldraw.cpp, 107
Centipede, 10
  ∼Centipede, 12
  bullet_x, 18
  bullet_y, 18
  Centipede, 12
  changeDir, 12
  checkCollision, 13
  checkInputs, 13
  createCentipede, 14
  food_x, 18
  food_y, 18
  getBulletX, 15
  getBulletY, 15
  getFoodX, 15
  getFoodY, 15
  getScore, 15
  getSnake, 16
  getSnakeXpos, 16
  getSnakeYpos, 16
  getXpos, 16
  getYpos, 16
  hit_counter, 18
  hitWall, 17
  lock, 18
  relaunch, 17
  setLock, 17
  snakes, 18
  wall, 19
  x, 19
  xcoords, 19
  y, 19
  ycoords, 19
centipede.cpp
  __attribute__, 93
  calledFirst, 94
  calledLast, 94
  entryPoint, 94
changeDir
  Centipede, 12
  Nibbler, 74
changeFoodCoord
  Nibbler, 75
checkCollision
  Centipede, 13
checkInputs
  Centipede, 13
  IGame, 50
  Nibbler, 76
checkSnake
  Nibbler, 76
closeAllHandles
  DLLoader, 44
Core, 20
  _actualGame, 41
  _actualLib, 41
  _command, 41
  _game, 41
  _gamesName, 41
  _graphName, 42
  _graphic, 41
  _maxGame, 42
  _maxGraph, 42
  _play, 42
  _playerName, 42
  _posGame, 42
  _posGraph, 42
  ∼Core, 22
  addPlayerName, 22
  Core, 22
  gameLoop, 23
  gamePrint, 24
  getGame, 25
  getGameName, 25
  getGraphic, 26
  getGraphName, 26
  getInput, 27
  getPlay, 28
  inGamesName, 28
  inGraphicsName, 29
  lib, 29
  menu, 30
  pActual, 31
  pCommand, 31
  pNotHandle, 32
  printLib, 33
  setFirstGame, 33
  setFirstGraph, 34
  setGame, 35
  setGraphic, 35
  setNextGame, 36
  setNextGraphic, 37
  setPosGame, 37
  setPosGraph, 38
  setPrevGame, 38
  setPrevGraphic, 39
  startOrExit, 40
Core.hpp
  gameLoop, 85
  get_lib, 85
core_f/core.cpp, 83
core_f/Core.hpp, 84
core_f/DLLoader.cpp, 86
core_f/DLLoader.hpp, 87

core_f/libfinder.cpp, 88
core_f/main.cpp, 91
createCentipede
    Centipede, 14
createSnake
    Nibbler, 77

DLLoader, 43
    _games, 48
    _gamesName, 48
    _gamesNb, 48
    _graphics, 48
    _graphicsName, 48
    _graphicsNb, 49
    _handles, 49
    ∼DLLoader, 44
    closeAllHandles, 44
    DLLoader, 44
    getGameInstance, 44
    getGameInstanceName, 45
    getGamesNb, 45
    getGraphicsNb, 45
    getGraphInstance, 46
    getGraphInstanceName, 46
    getInstanceOfGame, 46
    getInstanceOfGraph, 47
    insideOfGames, 47
drawApple
    MySDL2, 62
drawBullet
    MySDL2, 63
drawobj
    IGraphic, 54
    MyNcurses, 57
    MySDL2, 63
    MySfml, 69
drawSnake
    MySDL2, 64

entryPoint
    centipede.cpp, 94
    MySDL2.cpp, 104
    ncurses.cpp, 101
    nibbler.cpp, 98
    sfmldraw.cpp, 107

find_in_game
    libfinder.cpp, 88
find_in_gra
    libfinder.cpp, 89
food_x
    Centipede, 18
    Nibbler, 80
food_y
    Centipede, 18
    Nibbler, 80

game/centipede/centipede.cpp, 92
game/centipede/centipede.hpp, 94
game/igame.hpp, 95
game/nibbler/nibbler.cpp, 96
game/nibbler/nibbler.hpp, 98
gameLoop
    Core, 23
    Core.hpp, 85
gamePrint
    Core, 24
get_lib
    Core.hpp, 85
    libfinder.cpp, 89
getBulletX
    Centipede, 15
    IGame, 50
    Nibbler, 78
getBulletY
    Centipede, 15
    IGame, 51
    Nibbler, 78
getFoodX
    Centipede, 15
    IGame, 51
    Nibbler, 78
getFoodY
    Centipede, 15
    IGame, 51
    Nibbler, 78
getGame
    Core, 25
getGameInstance
    DLLoader, 44
getGameInstanceName
    DLLoader, 45
getGameName
    Core, 25
getGamesNb
    DLLoader, 45
getGraphic
    Core, 26
getGraphicsNb
    DLLoader, 45
getGraphInstance
    DLLoader, 46
getGraphInstanceName
    DLLoader, 46
getGraphName
    Core, 26
getInput
    Core, 27
    IGraphic, 54
    MyNcurses, 58
    MySDL2, 65
    MySfml, 70
getInstanceOfGame
    DLLoader, 46
getInstanceOfGraph
    DLLoader, 47
getPlay

Core, 28
getScore
Centipede, 15
IGame, 51
Nibbler, 78
getSnake
Centipede, 16
Nibbler, 79
getSnakeXpos
Centipede, 16
IGame, 51
Nibbler, 79
getSnakeYpos
Centipede, 16
IGame, 52
Nibbler, 79
getXpos
Centipede, 16
IGame, 52
Nibbler, 79
getYpos
Centipede, 16
IGame, 52
Nibbler, 79
graphic/igraphic.hpp, 99
graphic/ncurses/ncurses.cpp, 100
graphic/ncurses/ncurses.hpp, 102
graphic/sdl_deux/MySDL2.cpp, 103
graphic/sdl_deux/MySDL2.hpp, 105
graphic/sfml/sfmldraw.cpp, 105
graphic/sfml/sfmldraw.hpp, 108

hit_counter
Centipede, 18
hitWall
Centipede, 17

IGame, 49
~IGame, 50
checkInputs, 50
getBulletX, 50
getBulletY, 51
getFoodX, 51
getFoodY, 51
getScore, 51
getSnakeXpos, 51
getSnakeYpos, 52
getXpos, 52
getYpos, 52
relaunch, 52
setLock, 52
IGraphic, 53
~IGraphic, 54
drawobj, 54
getInput, 54
launch, 54
settingInput, 55
stop, 55
inGamesName
Core, 28
inGraphicsName
Core, 29
init
MyNcurses, 58
MySDL2, 65
MySfml, 70
insideOfGames
DLLoader, 47

launch
IGraphic, 54
MyNcurses, 58
MySDL2, 65
MySfml, 70
lib
Core, 29
libfinder.cpp
find_in_game, 88
find_in_gra, 89
get_lib, 89
niddleyearsold, 90
lock
Centipede, 18
Nibbler, 81

main
main.cpp, 91
main.cpp
main, 91
menu
Core, 30
MyNcurses, 56
_input, 60
_screen, 60
~MyNcurses, 57
apple, 60
bullet, 60
drawobj, 57
getInput, 58
init, 58
launch, 58
MyNcurses, 57
print_map, 59
rectangle, 60
settingInput, 59
stop, 59
MySDL2, 61
_event, 66
_input, 67
_render, 67
_screen, 67
~MySDL2, 62
drawApple, 62
drawBullet, 63
drawobj, 63
drawSnake, 64
getInput, 65
init, 65

launch, 65
MySDL2, 62
setInput, 65
settingInput, 66
stop, 66
MySDL2.cpp
__attribute__, 103, 104
calledFirst, 104
calledLast, 104
entryPoint, 104
MySfml, 67
_input, 71
∼MySfml, 69
apple, 71
bullet, 72
drawobj, 69
getInput, 70
init, 70
launch, 70
MySfml, 69
rectangle, 72
screen, 72
setInput, 70
settingInput, 71
stop, 71

ncurses.cpp
__attribute__, 101
calledFirst, 101
calledLast, 101
entryPoint, 101
Nibbler, 72
∼Nibbler, 74
changeDir, 74
changeFoodCoord, 75
checkInputs, 76
checkSnake, 76
createSnake, 77
food_x, 80
food_y, 80
getBulletX, 78
getBulletY, 78
getFoodX, 78
getFoodY, 78
getScore, 78
getSnake, 79
getSnakeXpos, 79
getSnakeYpos, 79
getXpos, 79
getYpos, 79
lock, 81
Nibbler, 74
relaunch, 80
score, 81
setLock, 80
snakes, 81
x, 81
xcoords, 81
y, 81
ycoords, 81
nibbler.cpp
__attribute__, 97
calledFirst, 98
calledLast, 98
entryPoint, 98
niddleyearsold
libfinder.cpp, 90

pActual
Core, 31
pCommand
Core, 31
pNotHandle
Core, 32
print_map
MyNcurses, 59
printLib
Core, 33

rectangle
MyNcurses, 60
MySfml, 72
relaunch
Centipede, 17
IGame, 52
Nibbler, 80

score
Nibbler, 81
screen
MySfml, 72
setFirstGame
Core, 33
setFirstGraph
Core, 34
setGame
Core, 35
setGraphic
Core, 35
setInput
MySDL2, 65
MySfml, 70
setLock
Centipede, 17
IGame, 52
Nibbler, 80
setNextGame
Core, 36
setNextGraphic
Core, 37
setPosGame
Core, 37
setPosGraph
Core, 38
setPrevGame
Core, 38
setPrevGraphic
Core, 39

settingInput
IGraphic, 55
MyNcurses, 59
MySDL2, 66
MySfml, 71
sfmldraw.cpp
__attribute__, 106, 107
calledFirst, 107
calledLastgra, 107
entryPoint, 107
snakes
Centipede, 18
Nibbler, 81
startOrExit
Core, 40
stop
IGraphic, 55
MyNcurses, 59
MySDL2, 66
MySfml, 71

wall
Centipede, 19

x
Centipede, 19
Nibbler, 81
xcoords
Centipede, 19
Nibbler, 81

y
Centipede, 19
Nibbler, 81
ycoords
Centipede, 19
Nibbler, 81