

Arcade

Generated by Doxygen 1.9.1

1 Namespace Index	1
1.1 Namespace List	1
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 Namespace Documentation	9
5.1 arcade Namespace Reference	9
5.1.1 Enumeration Type Documentation	11
5.1.1.1 BlockType	11
5.1.1.2 changes	11
5.1.1.3 displayLibs	12
5.1.1.4 EventType	12
5.1.1.5 gameLibs	12
5.1.1.6 libChange	13
5.1.1.7 Sound	13
5.1.2 Function Documentation	13
5.1.2.1 changePos()	14
5.1.2.2 chargeAllLibs()	14
5.1.2.3 checkLibChange()	14
5.1.2.4 closeAllLibs()	15
5.1.2.5 core()	15
5.1.2.6 createDisplayModule()	16
5.1.2.7 createGameModule()	16
5.1.2.8 deleteAllModules()	17
5.1.2.9 displayMenu()	17
5.1.2.10 fillDisplays()	18
5.1.2.11 fillGames()	18
5.1.2.12 findDisplay()	19
5.1.2.13 findDisplayLib()	19
5.1.2.14 findGame()	19
5.1.2.15 gameLoop()	20
5.1.2.16 getDisplayLibName()	20
5.1.2.17 getGameLibName()	21
5.1.2.18 Menu()	21
5.1.2.19 nextDisplay()	22
5.1.2.20 nextGame()	22
5.1.2.21 nextIndex()	22

5.1.2.22 openLib()	23
5.1.2.23 previousIndex()	23
6 Class Documentation	25
6.1 arcade::ADisplayNcurses Class Reference	25
6.1.1 Constructor & Destructor Documentation	26
6.1.1.1 ADisplayNcurses()	26
6.1.1.2 ~ADisplayNcurses()	26
6.1.2 Member Function Documentation	26
6.1.2.1 clear()	26
6.1.2.2 displayBlock()	26
6.1.2.3 displayText()	27
6.1.2.4 getBlockSize()	27
6.1.2.5 getWindowSize()	27
6.1.2.6 playSound()	28
6.1.2.7 pollEvent()	28
6.1.2.8 refresh()	28
6.1.2.9 setWindowSize()	28
6.1.2.10 setWindowTitle()	29
6.1.3 Member Data Documentation	29
6.1.3.1 _blockSize	29
6.1.3.2 _windowSize	29
6.1.3.3 _windowTitle	30
6.2 arcade::ADisplaySdl Class Reference	30
6.2.1 Constructor & Destructor Documentation	31
6.2.1.1 ADisplaySdl()	31
6.2.1.2 ~ADisplaySdl()	31
6.2.2 Member Function Documentation	31
6.2.2.1 clear()	32
6.2.2.2 close()	32
6.2.2.3 displayBlock()	32
6.2.2.4 displayText()	32
6.2.2.5 getBlockSize()	33
6.2.2.6 getWindowSize()	33
6.2.2.7 playSound()	33
6.2.2.8 pollEvent()	34
6.2.2.9 refresh()	34
6.2.2.10 setWindowSize()	34
6.2.2.11 setWindowTitle()	35
6.2.3 Member Data Documentation	35
6.2.3.1 _blockSize	35
6.2.3.2 _CoinBuffer	35

6.2.3.3 _font	35
6.2.3.4 _gameOverBuffer	36
6.2.3.5 _hurtBuffer	36
6.2.3.6 _renderer	36
6.2.3.7 _spriteSheet	36
6.2.3.8 _spriteSheetRect	36
6.2.3.9 _startBuffer	36
6.2.3.10 _superBuffer	36
6.2.3.11 _window	36
6.2.3.12 _windowSize	37
6.2.3.13 _windowTitle	37
6.3 arcade::ADisplaySfml Class Reference	37
6.3.1 Constructor & Destructor Documentation	38
6.3.1.1 ADisplaySfml()	38
6.3.1.2 ~ADisplaySfml()	38
6.3.2 Member Function Documentation	38
6.3.2.1 clear()	39
6.3.2.2 displayBlock()	39
6.3.2.3 displayText()	39
6.3.2.4 getBlockSize()	40
6.3.2.5 getWindowSize()	40
6.3.2.6 playSound()	40
6.3.2.7 pollEvent()	41
6.3.2.8 refresh()	41
6.3.2.9 setWindowSize()	41
6.3.2.10 setWindowTitle()	42
6.3.3 Member Data Documentation	42
6.3.3.1 _blockSize	42
6.3.3.2 _CoinBuffer	42
6.3.3.3 _font	42
6.3.3.4 _gameOverBuffer	42
6.3.3.5 _hurtBuffer	43
6.3.3.6 _startBuffer	43
6.3.3.7 _superBuffer	43
6.3.3.8 _window	43
6.3.3.9 _windowSize	43
6.4 cannotCloseLibrary Class Reference	43
6.4.1 Member Function Documentation	44
6.4.1.1 what()	44
6.5 cannotCloseModule Class Reference	44
6.5.1 Member Function Documentation	44
6.5.1.1 what()	44

6.6 cannotCreateModule Class Reference	45
6.6.1 Member Function Documentation	45
6.6.1.1 what()	45
6.7 cannotOpenLibrary Class Reference	45
6.7.1 Member Function Documentation	46
6.7.1.1 what()	46
6.8 arcade::Core Class Reference	46
6.8.1 Constructor & Destructor Documentation	46
6.8.1.1 Core()	47
6.8.1.2 ~Core()	47
6.8.2 Member Function Documentation	47
6.8.2.1 getName()	47
6.8.2.2 setName()	47
6.8.3 Member Data Documentation	47
6.8.3.1 _displayLibs	47
6.8.3.2 _gameLibs	48
6.8.3.3 _name	48
6.9 arcade::Event Struct Reference	48
6.9.1 Detailed Description	48
6.9.2 Member Data Documentation	48
6.9.2.1 type	48
6.10 arcade::ICore Class Reference	49
6.10.1 Member Function Documentation	49
6.10.1.1 getName()	49
6.10.1.2 setName()	49
6.11 arcade::IDisplayModule Class Reference	50
6.11.1 Detailed Description	50
6.11.2 Constructor & Destructor Documentation	50
6.11.2.1 ~IDisplayModule()	50
6.11.3 Member Function Documentation	50
6.11.3.1 clear()	51
6.11.3.2 displayBlock()	51
6.11.3.3 displayText()	51
6.11.3.4 getWindowSize()	51
6.11.3.5 playSound()	51
6.11.3.6 pollEvent()	52
6.11.3.7 refresh()	52
6.11.3.8 setWindowSize()	52
6.11.3.9 setWindowTitle()	52
6.12 arcade::IGameModule Class Reference	53
6.12.1 Detailed Description	53
6.12.2 Constructor & Destructor Documentation	53

6.12.2.1 ~IGameModule()	53
6.12.3 Member Function Documentation	53
6.12.3.1 getName()	53
6.12.3.2 start()	54
6.12.3.3 tick()	54
6.13 arcade::Nibbler Class Reference	54
6.13.1 Constructor & Destructor Documentation	55
6.13.1.1 Nibbler()	56
6.13.1.2 ~Nibbler()	56
6.13.2 Member Function Documentation	56
6.13.2.1 applyEvent()	56
6.13.2.2 changeDirection()	56
6.13.2.3 deathCondition()	56
6.13.2.4 end()	57
6.13.2.5 gamePause()	57
6.13.2.6 getName()	57
6.13.2.7 headSnakeOnSnake()	57
6.13.2.8 headSnakeOnWall()	57
6.13.2.9 isFruitOnSnake()	57
6.13.2.10 moveCorps()	58
6.13.2.11 moveSnake()	58
6.13.2.12 placeFruits()	58
6.13.2.13 printFruit()	58
6.13.2.14 printHighScore()	58
6.13.2.15 printMap()	59
6.13.2.16 printScoreAndName()	59
6.13.2.17 printSnake()	59
6.13.2.18 setAllFruits()	59
6.13.2.19 setMap()	59
6.13.2.20 setNewFruit()	60
6.13.2.21 snakeEatFruit()	60
6.13.2.22 start()	60
6.13.2.23 tick()	60
6.13.2.24 verifDown()	60
6.13.2.25 verifLeft()	61
6.13.2.26 verifRight()	61
6.13.2.27 verifUp()	61
6.13.3 Member Data Documentation	61
6.13.3.1 _changeDirectionList	61
6.13.3.2 _direction	61
6.13.3.3 _display	61
6.13.3.4 _events	62

6.13.3.5 _fruit	62
6.13.3.6 _highScore	62
6.13.3.7 _Map	62
6.13.3.8 _name	62
6.13.3.9 _score	62
6.13.3.10 _snakeCara	62
6.13.3.11 _speed	62
6.13.3.12 _waitDirection	63
6.13.3.13 _wall	63
6.14 arcade::Position Struct Reference	63
6.14.1 Detailed Description	63
6.14.2 Member Data Documentation	63
6.14.2.1 x	63
6.14.2.2 y	63
6.15 arcade::Snake Class Reference	64
6.15.1 Constructor & Destructor Documentation	65
6.15.1.1 Snake()	65
6.15.1.2 ~Snake()	65
6.15.2 Member Function Documentation	65
6.15.2.1 applyEvent()	66
6.15.2.2 changeDirection()	66
6.15.2.3 deathCondition()	66
6.15.2.4 end()	66
6.15.2.5 gamePause()	66
6.15.2.6 getName()	67
6.15.2.7 moveCorps()	67
6.15.2.8 moveSnake()	67
6.15.2.9 printFruit()	67
6.15.2.10 printHighScore()	68
6.15.2.11 printMap()	68
6.15.2.12 printScoreAndName()	68
6.15.2.13 printSnake()	68
6.15.2.14 setNewFruit()	68
6.15.2.15 snakeEatFruit()	69
6.15.2.16 start()	69
6.15.2.17 tick()	69
6.15.3 Member Data Documentation	69
6.15.3.1 _changeDirectionList	70
6.15.3.2 _direction	70
6.15.3.3 _display	70
6.15.3.4 _events	70
6.15.3.5 _fruit	70

6.15.3.6 _highScore	70
6.15.3.7 _Map	70
6.15.3.8 _name	70
6.15.3.9 _score	71
6.15.3.10 _snakeCara	71
6.15.3.11 _speed	71
6.15.3.12 _waitDirection	71
7 File Documentation	73
7.1 src/core/chargeLib.cpp File Reference	73
7.2 src/core/closeLib.cpp File Reference	73
7.3 src/core/core.cpp File Reference	74
7.3.1 Function Documentation	74
7.3.1.1 help()	74
7.3.1.2 main()	75
7.4 src/core/core.hpp File Reference	75
7.5 src/core/findLib.cpp File Reference	77
7.6 src/core/initLib.cpp File Reference	77
7.7 src/core/menu.cpp File Reference	77
7.8 src/errors/errors.hpp File Reference	78
7.9 src/interfaces/Event.hpp File Reference	78
7.10 src/interfaces/ICore.hpp File Reference	79
7.11 src/interfaces/IDisplayModule.hpp File Reference	79
7.11.1 Function Documentation	80
7.11.1.1 createDisplayModule()	80
7.12 src/interfaces/IGameModule.hpp File Reference	80
7.12.1 Function Documentation	80
7.12.1.1 createGameModule()	80
7.13 src/Ldisplay/ncurses/ADisplayNcurses.cpp File Reference	81
7.13.1 Function Documentation	81
7.13.1.1 createDisplayModule()	81
7.14 src/Ldisplay/ncurses/ADisplayNcurses.hpp File Reference	81
7.15 src/Ldisplay/sdl/ADisplaySdl.cpp File Reference	81
7.15.1 Function Documentation	82
7.15.1.1 createDisplayModule()	82
7.16 src/Ldisplay/sdl/ADisplaySdl.hpp File Reference	82
7.17 src/Ldisplay/sfml/ADisplaySfml.cpp File Reference	82
7.17.1 Function Documentation	83
7.17.1.1 createDisplayModule()	83
7.18 src/Ldisplay/sfml/ADisplaySfml.hpp File Reference	83
7.19 src/Lgame/nibbler/nibbler.cpp File Reference	83
7.19.1 Function Documentation	83

7.19.1.1 addVector()	84
7.19.1.2 createGameModule()	84
7.20 src/Lgame/nibbler/nibbler.hpp File Reference	84
7.21 src/Lgame/snake/snake.cpp File Reference	84
7.21.1 Function Documentation	85
7.21.1.1 createGameModule()	85
7.21.1.2 headOnKillBloc()	85
7.21.1.3 isFruitOnSnake()	85
7.22 src/Lgame/snake/snake.hpp File Reference	85
Index	87

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

arcade	9
------------------------	-------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

arcade::Event	48
std::exception	
cannotCloseLibrary	43
cannotCloseModule	44
cannotCreateModule	45
cannotOpenLibrary	45
arcade::ICore	49
arcade::Core	46
arcade::IDisplayModule	50
arcade::ADisplayNcurses	25
arcade::ADisplaySdl	30
arcade::ADisplaySfml	37
arcade::IGameModule	53
arcade::Nibbler	54
arcade::Snake	64
arcade::Position	63

Chapter 3

Class Index

3.1 Class List

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

arcade::ADisplayNcurses	25
arcade::ADisplaySdl	30
arcade::ADisplaySfml	37
cannotCloseLibrary	43
cannotCloseModule	44
cannotCreateModule	45
cannotOpenLibrary	45
arcade::Core	46
arcade::Event	48
arcade::ICore	49
arcade::IDisplayModule	50
arcade::IGameModule	53
arcade::Nibbler	54
arcade::Position	63
arcade::Snake	64

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/core/ chargeLib.cpp	73
src/core/ closeLib.cpp	73
src/core/ core.cpp	74
src/core/ core.hpp	75
src/core/ findLib.cpp	77
src/core/ initLib.cpp	77
src/core/ menu.cpp	77
src/errors/ errors.hpp	78
src/interfaces/ Event.hpp	78
src/interfaces/ ICore.hpp	79
src/interfaces/ IDisplayModule.hpp	79
src/interfaces/ IGameModule.hpp	80
src/Ldisplay/ncurses/ ADisplayNcurses.cpp	81
src/Ldisplay/ncurses/ ADisplayNcurses.hpp	81
src/Ldisplay/sdl/ ADisplaySdl.cpp	81
src/Ldisplay/sdl/ ADisplaySdl.hpp	82
src/Ldisplay/sfml/ ADisplaySfml.cpp	82
src/Ldisplay/sfml/ ADisplaySfml.hpp	83
src/Lgame/nibbler/ nibbler.cpp	83
src/Lgame/nibbler/ nibbler.hpp	84
src/Lgame/snake/ snake.cpp	84
src/Lgame/snake/ snake.hpp	85

Chapter 5

Namespace Documentation

5.1 arcade Namespace Reference

Classes

- class [Core](#)
- struct [Event](#)
- class [ICore](#)
- struct [Position](#)
- class [IDisplayModule](#)
- class [IGameModule](#)
- class [ADisplayNcurses](#)
- class [ADisplaySdl](#)
- class [ADisplaySfml](#)
- class [Nibbler](#)
- class [Snake](#)

Enumerations

- enum [gameLibs](#) {
 [Snake](#) , [Nibbler](#) , [Pacman](#) , [Centipede](#) ,
 [Qix](#) , [Solarfox](#) }
- enum [displayLibs](#) {
 [Ncurses](#) , [SFML](#) , [SDL2](#) , [Aalib](#) ,
 [Caca](#) , [OpenGL](#) , [Allegro](#) , [Gtk](#) ,
 [Xlib](#) , [Irrlicht](#) , [Qt5](#) , [Vulkain](#) ,
 [Ndk](#) }
- enum class [libChange](#) {
 [NextGame](#) , [PreviousGame](#) , [NextLibrary](#) , [PreviousLibrary](#) ,
 [ReturnToSelectionMenu](#) , [Restart](#) , [Quit](#) , [None](#) }
- enum [changes](#) {
 [Left](#) , [Right](#) , [Up](#) , [Down](#) ,
 [None](#) , [None](#) }
- enum class [EventType](#) {
 [Up](#) , [Down](#) , [Left](#) , [Right](#) ,
 [Action](#) , [Pause](#) , [Quit](#) , [SwitchToNextGame](#) ,
 [SwitchToPreviousGame](#) , [ReturnToSelectionMenu](#) , [NextLibrary](#) , [PreviousLibrary](#) ,
 [RestartGame](#) }

- enum class [BlockType](#) {
[BrickWall](#) , [Grass](#) , [Fruit](#) , [Coin](#) ,
[Character](#) , [Enemy](#) , [EnemyVulnerable](#) , [EnemyDead](#) }
- enum class [Sound](#) {
[Coin](#) , [Hurt](#) , [GameOver](#) , [Start](#) ,
[Super](#) }

Functions

- [IDisplayModule](#) * [createDisplayModule](#) (void *lib)
Create a Display Module object.
- [IGameModule](#) * [createGameModule](#) (void *lib)
Create a Game Module object.
- std::string [nextIndex](#) (std::vector< std::string > lib, std::string curent)
return the next index of a vector and the first one if it's the last
- std::string [previousIndex](#) (std::vector< std::string > lib, std::string curent)
return the previous index of a vector and the last one if it's the first
- void [deleteAllModules](#) ([IDisplayModule](#) *displayModule, [IGameModule](#) *gameModule)
delete a display and a game module
- void [closeAllLibs](#) ([Core](#) *core)
Close all libraries.
- enum [libChange](#) [checkLibChange](#) (const std::list< [Event](#) > &events)
check if the lib has to change
- int [gameLoop](#) ([Core](#) *core, [IDisplayModule](#) *displayModule, [IGameModule](#) *gameModule, std::vector< std::string > games, std::vector< std::string > displays, std::string gameName, std::string displayName)
game loop
- int [core](#) (char *displayModule)
Core of the program were display and game modules are created,linked and deleted.
- [IGameModule](#) * [nextGame](#) ([Core](#) *core, [IGameModule](#) *gameModule, std::vector< std::string > &[gameLibs](#))
- [IDisplayModule](#) * [nextDisplay](#) ([Core](#) *core, [IDisplayModule](#) *displayModule, std::vector< std::string > &[displayLibs](#))
- [Core](#) * [chargeAllLibs](#) (void)
Get the All Libs object and fill core->_displayLibs and core->_gameLibs.
- void * [openLib](#) (const char *libName)
Open a library.
- std::string [findDisplayLib](#) ([Core](#) *core, char *file)
Find a display library.
- std::string [findDisplay](#) ([Core](#) *core, enum [displayLibs](#) libs)
Find a display library.
- std::string [findGame](#) ([Core](#) *core, enum [gameLibs](#) libs)
Find a game library.
- int [Menu](#) ([IDisplayModule](#) *display, [Core](#) *core, std::string firstDisplay)
Print the menu and handle the events. THIS FUNCTION CALL THE GAME LOOP.
- std::string [getGameLibName](#) (enum [gameLibs](#) gameLib)
Get the Game Lib Name object.
- std::string [getDisplayLibName](#) (enum [displayLibs](#) displayLib)
Get the Display Lib Name object.
- std::pair< int, int > [changePos](#) (const std::list< [Event](#) > &events, std::pair< int, int > pos, std::vector< std::string > games, std::vector< std::string > displays)
highlight the selected game or library

- `std::pair< int, int > displayMenu` (`std::pair< int, int > pos`, `std::list< Event > events`, `IDisplayModule *displayModule`, `std::vector< std::string > games`, `std::vector< std::string > displays`, `std::string name`)
display the menu and highlight the selected game or library
- `std::vector< std::string > fillGames` (`Core *core`)
fill the vector with the games names
- `std::vector< std::string > fillDisplays` (`Core *core`)
fill the vector with the displays names

5.1.1 Enumeration Type Documentation

5.1.1.1 BlockType

```
enum arcade::BlockType [strong]
```

A type of block to be rendered

Enumerator

BrickWall	
Grass	
Fruit	
Coin	
Character	
Enemy	
EnemyVulnerable	
EnemyDead	

5.1.1.2 changes

```
enum arcade::changes
```

Enumerator

Left	
Right	
Up	
Down	
None	
None	

5.1.1.3 displayLibs

```
enum arcade::displayLibs
```

Enumerator

Ncurses	
SFML	
SDL2	
Aalib	
Caca	
OpenGL	
Allegro	
Gtk	
Xlib	
Irrlicht	
Qt5	
Vulkain	
Ndk	

5.1.1.4 EventType

```
enum arcade::EventType [strong]
```

The type of a keyboard or mouse event

Enumerator

Up	Go UP (arrow key)
Down	Go DOWN (arrow key)
Left	Go LEFT (arrow key)
Right	Go RIGHT (arrow key)
Action	Activate (space bar)
Pause	Toggle Pause (escape key)
Quit	Exit program (backspace key or close the window)
SwitchToNextGame	Quit current game and load the next one (+)
SwitchToPreviousGame	Quit current game and load the previous one (-)
ReturnToSelectionMenu	Quit current game and return to the game selection menu (enter key)
NextLibrary	Switch to the next graphical library (page up or p key)
PreviousLibrary	Switch to the previous graphical library (page down or o key)
RestartGame	Restart the current game (r key)

5.1.1.5 gameLibs

```
enum arcade::gameLibs
```

Enumerator

Snake	
Nibbler	
Pacman	
Centipede	
Qix	
Solarfox	

5.1.1.6 libChange

```
enum arcade::libChange [strong]
```

Enumerator

NextGame	
PreviousGame	
NextLibrary	
PreviousLibrary	
ReturnToSelectionMenu	
Restart	
Quit	
None	

5.1.1.7 Sound

```
enum arcade::Sound [strong]
```

A type of sound effect to be played

Enumerator

Coin	
Hurt	
GameOver	
Start	
Super	

5.1.2 Function Documentation

5.1.2.1 changePos()

```
std::pair<int, int> arcade::changePos (
    const std::list< Event > & events,
    std::pair< int, int > pos,
    std::vector< std::string > games,
    std::vector< std::string > displays )
```

highlight the selected game or library

Author

leo

Parameters

<i>events</i>	
<i>pos</i>	
<i>games</i>	only use to get the size of the list TODO: change that
<i>displays</i>	only use to get the size of the list TODO: change that

Returns

std::pair<int, int>

5.1.2.2 chargeAllLibs()

```
Core * arcade::chargeAllLibs (
    void )
```

Get the All Libs object and fill core->_displayLibs and core->_gameLibs.

Author

leo

Parameters

<i>core</i>	
<i>firstDisplay</i>	

5.1.2.3 checkLibChange()

```
enum libChange arcade::checkLibChange (
    const std::list< Event > & events )
```


check if the lib has to change

Author

leo

Parameters

<i>events</i>	list of events
---------------	----------------

Returns

enum libChange

5.1.2.4 closeAllLibs()

```
void arcade::closeAllLibs (  
    Core * core )
```

Close all libraries.

Author

leo

Parameters

<i>core</i>	
-------------	--

5.1.2.5 core()

```
int arcade::core (  
    char * displayModule )
```

Core of the program were display and game modules are created,linked and deleted.

Author

leo

Parameters

<i>argc</i>	
<i>argv</i>	

Returns

int

5.1.2.6 createDisplayModule()

```
IDisplayModule * arcade::createDisplayModule (
    void * lib )
```

Create a Display Module object.

Authors

leo

Parameters

<i>lib</i>	created by openLib
------------	--------------------

Exceptions

<i>cannotCreateModule</i>	
---	--

Returns

IDisplayModule*

5.1.2.7 createGameModule()

```
IGameModule * arcade::createGameModule (
    void * lib )
```

Create a Game Module object.

Authors

leo

Parameters

<i>lib</i>	created by openLib
------------	--------------------

Exceptions

<code>cannotCreateModule</code>	
---------------------------------	--

Returns

IGameModule*

5.1.2.8 deleteAllModules()

```
void arcade::deleteAllModules (
    IDisplayModule * displayModule,
    IGameModule * gameModule )
```

delete a display and a game module

Author

leo

Parameters

<i>displayModule</i>	
<i>gameModule</i>	

5.1.2.9 displayMenu()

```
std::pair<int, int> arcade::displayMenu (
    std::pair< int, int > pos,
    std::list< Event > events,
    IDisplayModule * displayModule,
    std::vector< std::string > games,
    std::vector< std::string > displays,
    std::string name )
```

display the menu and highlight the selected game or library

Author

leo

Parameters

<i>pos</i>	
<i>events</i>	
<i>displayModule</i>	
<i>games</i>	
<i>displays</i>	
<i>name</i>	

Returns

`std::pair<int, int> pos`

5.1.2.10 fillDisplays()

```
std::vector<std::string> arcade::fillDisplays (
    Core * core )
```

fill the vector with the displays names

Author

leo

Parameters

<i>core</i>	
-------------	--

Returns

`std::vector<std::string> displays`

5.1.2.11 fillGames()

```
std::vector<std::string> arcade::fillGames (
    Core * core )
```

fill the vector with the games names

Author

leo

Parameters

<i>core</i>	
-------------	--

Returns

`std::vector<std::string> games`

5.1.2.12 findDisplay()

```
std::string arcade::findDisplay (
    Core * core,
    enum displayLibs libs )
```

Find a display library.

Author

leo

Parameters

<i>core</i>	
<i>libs</i>	

Returns

int

5.1.2.13 findDisplayLib()

```
std::string arcade::findDisplayLib (
    Core * core,
    char * file )
```

Find a display library.

Author

leo

Parameters

<i>core</i>	
<i>file</i>	

Returns

int number of the library

5.1.2.14 findGame()

```
std::string arcade::findGame (
    Core * core,
    enum gameLibs libs )
```

Find a game library.

Author

leo

Parameters

<i>core</i>	
<i>libs</i>	

Returns

int

5.1.2.15 gameLoop()

```
int arcade::gameLoop (
    Core * core,
    IDisplayModule * displayModule,
    IGameModule * gameModule,
    std::vector< std::string > games,
    std::vector< std::string > displays,
    std::string gameName,
    std::string displayName )
```

game loop

Author

leo

Parameters

<i>core</i>	
<i>displayModule</i>	
<i>gameModule</i>	

Returns

int TODO cahge that to return nothing or throw an error

5.1.2.16 getDisplayLibName()

```
std::string arcade::getDisplayLibName (
    enum displayLibs displayLib )
```

Get the Display Lib Name object.

Author

leo

Parameters

<i>displayLib</i>	
-------------------	--

Returns

std::string

5.1.2.17 `getGameLibName()`

```
std::string arcade::getGameLibName (
    enum gameLibs gameLib )
```

Get the Game Lib Name object.

Author

leo

Parameters

<i>gameLib</i>	
----------------	--

Returns

std::string

5.1.2.18 `Menu()`

```
int arcade::Menu (
    IDisplayModule * display,
    Core * core,
    std::string firstDisplay )
```

Print the menu and handle the events. THIS FUNCTION CALL THE GAME LOOP.

Author

leo

Parameters

<i>display</i>	
<i>core</i>	

Returns

int

5.1.2.19 nextDisplay()

```

IDisplayModule* arcade::nextDisplay (
    Core * core,
    IDisplayModule * displayModule,
    std::vector< std::string > & displayLibs )

```

5.1.2.20 nextGame()

```

IGameModule* arcade::nextGame (
    Core * core,
    IGameModule * gameModule,
    std::vector< std::string > & gameLibs )

```

5.1.2.21 nextIndex()

```

std::string arcade::nextIndex (
    std::vector< std::string > lib,
    std::string curent )

```

return the next index of a vector and the first one if it's the last

Author

leo

Parameters

<i>lib</i>	list of libs
------------	--------------

Returns

std::string next index

5.1.2.22 openLib()

```
void * arcade::openLib (
    const char * libName )
```

Open a library.

Authors

leo

Parameters

<i>libName</i>	name of the library .so
----------------	-------------------------

Exceptions

<i>cannotOpenLibrary</i>	
--	--

Returns

void*

5.1.2.23 previousIndex()

```
std::string arcade::previousIndex (
    std::vector< std::string > lib,
    std::string curent )
```

return the previous index of a vector and the last one if it's the first

Author

leo

Parameters

<i>lib</i>	list of libs
------------	--------------

Returns

std::string previous index

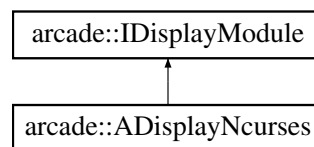
Chapter 6

Class Documentation

6.1 arcade::ADisplayNcurses Class Reference

```
#include <ADisplayNcurses.hpp>
```

Inheritance diagram for arcade::ADisplayNcurses:



Public Member Functions

- [ADisplayNcurses](#) ()
Construct a new arcade::ADisplayNcurses object.
- [~ADisplayNcurses](#) () override
Destroy the arcade::ADisplayNcurses object.
- void [displayBlock](#) ([BlockType](#) blockType, [Position](#) pos) override
Display a character at the position on the screen.
- void [displayText](#) (const std::string &text, [Position](#) pos) override
Display a text at the position on the screen.
- void [clear](#) () override
clear the screen
- void [refresh](#) () override
refresh the screen
- void [playSound](#) ([Sound](#) sound) override
play a sound
- std::list< [Event](#) > [pollEvent](#) () override
poll the event from the keyboard
- void [setWindowSize](#) ([Position](#) size) override
set the window size
- [Position](#) [getWindowSize](#) () override
get window size
- void [setWindowTitle](#) (const std::string &title) override
set window title
- int [getBlockSize](#) ()

Private Attributes

- [Position _windowSize](#)
- `std::string _windowTitle`
- `int _blockSize = 1`

6.1.1 Constructor & Destructor Documentation

6.1.1.1 ADisplayNcurses()

```
arcade::ADisplayNcurses::ADisplayNcurses ( )
```

Construct a new arcade::A Display Ncurses::A Display Ncurses object.

Author

pier-alexandre

6.1.1.2 ~ADisplayNcurses()

```
arcade::ADisplayNcurses::~~ADisplayNcurses ( ) [override]
```

Destroy the arcade::A Display Ncurses::A Display Ncurses object.

Author

pier-alexandre

6.1.2 Member Function Documentation

6.1.2.1 clear()

```
void arcade::ADisplayNcurses::clear ( ) [override], [virtual]
```

clear the screen

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.1.2.2 displayBlock()

```
void arcade::ADisplayNcurses::displayBlock (
    BlockType blockType,
    Position pos ) [override], [virtual]
```

Display a caracter at the position on the screen.

Author

pier-alexandre

Parameters

<i>blockType</i>	
<i>pos</i>	

Implements [arcade::IDisplayModule](#).

6.1.2.3 displayText()

```
void arcade::ADisplayNcurses::displayText (
    const std::string & text,
    Position pos ) [override], [virtual]
```

Display a text at the position on the screen.

Author

pier-alexandre

Parameters

<i>text</i>	
<i>pos</i>	

Implements [arcade::IDisplayModule](#).

6.1.2.4 getBlockSize()

```
int arcade::ADisplayNcurses::getBlockSize ( ) [inline]
```

6.1.2.5 getWindowSize()

```
arcade::Position arcade::ADisplayNcurses::getWindowSize ( ) [override], [virtual]
```

get window size

Returns

[arcade::Position](#)

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.1.2.6 playSound()

```
void arcade::ADisplayNcurses::playSound (
    Sound sound ) [override], [virtual]
```

play a sound

Parameters

<i>sound</i>	
--------------	--

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.1.2.7 pollEvent()

```
std::list< arcade::Event > arcade::ADisplayNcurses::pollEvent ( ) [override], [virtual]
```

poll the event from the keyboard

Returns

std::list<arcade::Event>

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.1.2.8 refresh()

```
void arcade::ADisplayNcurses::refresh ( ) [override], [virtual]
```

refresh the screen

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.1.2.9 setWindowSize()

```
void arcade::ADisplayNcurses::setWindowSize (
    Position size ) [override], [virtual]
```

set the window size

Parameters

<i>size</i>	
-------------	--

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.1.2.10 setWindowTitle()

```
void arcade::ADisplayNcurses::setWindowTitle (
    const std::string & title ) [override], [virtual]
```

set window title

Parameters

<i>std::string</i>	&title
--------------------	--------

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.1.3 Member Data Documentation

6.1.3.1 _blockSize

```
int arcade::ADisplayNcurses::_blockSize = 1 [private]
```

6.1.3.2 _windowSize

```
Position arcade::ADisplayNcurses::_windowSize [private]
```

6.1.3.3 _windowTitle

```
std::string arcade::ADisplayNcurses::_windowTitle [private]
```

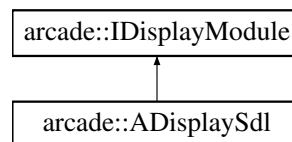
The documentation for this class was generated from the following files:

- [src/Ldisplay/ncurses/ADisplayNcurses.hpp](#)
- [src/Ldisplay/ncurses/ADisplayNcurses.cpp](#)

6.2 arcade::ADisplaySdl Class Reference

```
#include <ADisplaySdl.hpp>
```

Inheritance diagram for arcade::ADisplaySdl:



Public Member Functions

- [ADisplaySdl](#) ()
Construct a new arcade::A Display Sdl::A Display Sdl object.
- [~ADisplaySdl](#) () override
Destroy the arcade::A Display Sdl::A Display Sdl object.
- void [displayBlock](#) ([BlockType](#) blockType, [Position](#) pos) override
display a block of 32 pixel
- void [displayText](#) (const std::string &text, [Position](#) pos) override
display text
- void [clear](#) () override
clear the screen
- void [refresh](#) () override
refresh the screen
- void [close](#) ()
close the window
- void [playSound](#) ([Sound](#) sound) override
play a sound
- std::list< [Event](#) > [pollEvent](#) () override
poll the event from the keyboard
- void [setWindowSize](#) ([Position](#) size) override
set the window size
- [Position](#) [getWindowSize](#) () override
get window size
- void [setWindowTitle](#) (const std::string &title) override
set window title
- int [getBlockSize](#) ()

Private Attributes

- SDL_Renderer * [_renderer](#)
- SDL_Window * [_window](#)
- Mix_Chunk * [_CoinBuffer](#)
- Mix_Chunk * [_hurtBuffer](#)
- Mix_Chunk * [_gameOverBuffer](#)
- Mix_Chunk * [_startBuffer](#)
- Mix_Chunk * [_superBuffer](#)
- SDL_Texture * [_spriteSheet](#)
- SDL_Rect [_spriteSheetRect](#) [5]
- int [_blockSize](#)
- [Position](#) [_windowSize](#)
- std::string [_windowTitle](#)
- TTF_Font * [_font](#)

6.2.1 Constructor & Destructor Documentation

6.2.1.1 ADisplaySdl()

```
arcade::ADisplaySdl::ADisplaySdl ( )
```

Construct a new arcade::A Display Sdl::A Display Sdl object.

Author

pier-alexandre

6.2.1.2 ~ADisplaySdl()

```
arcade::ADisplaySdl::~~ADisplaySdl ( ) [override]
```

Destroy the arcade::A Display Sdl::A Display Sdl object.

Author

pier-alexandre

6.2.2 Member Function Documentation

6.2.2.1 clear()

```
void arcade::ADisplaySdl::clear ( ) [override], [virtual]
```

clear the screen

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.2.2.2 close()

```
void arcade::ADisplaySdl::close ( )
```

close the window

Author

pier-alexandre

6.2.2.3 displayBlock()

```
void arcade::ADisplaySdl::displayBlock (
    BlockType blockType,
    Position pos ) [override], [virtual]
```

display a block of 32 pixel

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.2.2.4 displayText()

```
void arcade::ADisplaySdl::displayText (
    const std::string & text,
    Position pos ) [override], [virtual]
```

display text

Parameters

<i>text</i>	
<i>pos</i>	

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.2.2.5 `getBlockSize()`

```
int arcade::ADisplaySdl::getBlockSize ( ) [inline]
```

6.2.2.6 `getWindowSize()`

```
arcade::Position arcade::ADisplaySdl::getWindowSize ( ) [override], [virtual]
```

get window size

Returns

[arcade::Position](#)

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.2.2.7 `playSound()`

```
void arcade::ADisplaySdl::playSound (
    Sound sound ) [override], [virtual]
```

play a sound

Parameters

<i>sound</i>	
--------------	--

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).**6.2.2.8 pollEvent()**

```
std::list< arcade::Event > arcade::ADisplaySdl::pollEvent ( ) [override], [virtual]
```

poll the event from the keyboard

Returns

std::list<arcade::Event>

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).**6.2.2.9 refresh()**

```
void arcade::ADisplaySdl::refresh ( ) [override], [virtual]
```

refresh the screen

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).**6.2.2.10 setWindowSize()**

```
void arcade::ADisplaySdl::setWindowSize (
    Position size ) [override], [virtual]
```

set the window size

Parameters

<i>size</i>	
-------------	--

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.2.2.11 setWindowTitle()

```
void arcade::ADisplaySdl::setWindowTitle (
    const std::string & title ) [override], [virtual]
```

set window title

Parameters

<i>std::string</i>	&title
--------------------	--------

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.2.3 Member Data Documentation**6.2.3.1 _blockSize**

```
int arcade::ADisplaySdl::_blockSize [private]
```

6.2.3.2 _CoinBuffer

```
Mix_Chunk* arcade::ADisplaySdl::_CoinBuffer [private]
```

6.2.3.3 _font

```
TTF_Font* arcade::ADisplaySdl::_font [private]
```

6.2.3.4 `_gameOverBuffer`

```
Mix_Chunk* arcade::ADisplaySdl::_gameOverBuffer [private]
```

6.2.3.5 `_hurtBuffer`

```
Mix_Chunk* arcade::ADisplaySdl::_hurtBuffer [private]
```

6.2.3.6 `_renderer`

```
SDL_Renderer* arcade::ADisplaySdl::_renderer [private]
```

6.2.3.7 `_spriteSheet`

```
SDL_Texture* arcade::ADisplaySdl::_spriteSheet [private]
```

6.2.3.8 `_spriteSheetRect`

```
SDL_Rect arcade::ADisplaySdl::_spriteSheetRect[5] [private]
```

6.2.3.9 `_startBuffer`

```
Mix_Chunk* arcade::ADisplaySdl::_startBuffer [private]
```

6.2.3.10 `_superBuffer`

```
Mix_Chunk* arcade::ADisplaySdl::_superBuffer [private]
```

6.2.3.11 `_window`

```
SDL_Window* arcade::ADisplaySdl::_window [private]
```

6.2.3.12 _windowSize

`Position` arcade::ADisplaySdl::_windowSize [private]

6.2.3.13 _windowTitle

`std::string` arcade::ADisplaySdl::_windowTitle [private]

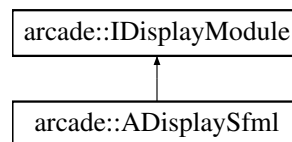
The documentation for this class was generated from the following files:

- [src/Ldisplay/sdl/ADisplaySdl.hpp](#)
- [src/Ldisplay/sdl/ADisplaySdl.cpp](#)

6.3 arcade::ADisplaySfml Class Reference

```
#include <ADisplaySfml.hpp>
```

Inheritance diagram for arcade::ADisplaySfml:



Public Member Functions

- [ADisplaySfml](#) ()
Construct a new arcade::A Display Sfml::A Display Sfml object.
- [~ADisplaySfml](#) () override
Destroy the arcade::A Display Sfml::A Display Sfml object.
- void [displayBlock](#) ([BlockType](#) blockType, [Position](#) pos) override
display a block on the screen
- void [displayText](#) (const std::string &text, [Position](#) pos) override
display a text on the screen
- void [clear](#) () override
clear the screen
- void [refresh](#) () override
refresh the screen
- void [playSound](#) ([Sound](#) sound) override
get the input from the user
- std::list< [Event](#) > [pollEvent](#) () override
get the input from the user
- void [setWindowSize](#) ([Position](#) size) override
set the size of the window
- [Position](#) [getWindowSize](#) () override
get the size of the window
- void [setWindowTitle](#) (const std::string &title) override
set the title of the window
- int [getBlockSize](#) ()

Private Attributes

- [sf::RenderWindow _window](#)
- [sf::Font _font](#)
- [sf::SoundBuffer _CoinBuffer](#)
- [sf::SoundBuffer _hurtBuffer](#)
- [sf::SoundBuffer _gameOverBuffer](#)
- [sf::SoundBuffer _startBuffer](#)
- [sf::SoundBuffer _superBuffer](#)
- [Position _windowSize](#)
- [int _blockSize](#)

6.3.1 Constructor & Destructor Documentation

6.3.1.1 ADisplaySfml()

```
arcade::ADisplaySfml::ADisplaySfml ( )
```

Construct a new arcade::A Display Sfml::A Display Sfml object.

Author

pier-alexandre

6.3.1.2 ~ADisplaySfml()

```
arcade::ADisplaySfml::~~ADisplaySfml ( ) [override]
```

Destroy the arcade::A Display Sfml::A Display Sfml object.

Author

pier-alexandre

6.3.2 Member Function Documentation

6.3.2.1 clear()

```
void arcade::ADisplaySfml::clear ( ) [override], [virtual]
```

clear the screen

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.3.2.2 displayBlock()

```
void arcade::ADisplaySfml::displayBlock (
    BlockType blockType,
    Position pos ) [override], [virtual]
```

display a block on the screen

Parameters

<i>blockType</i>	
<i>pos</i>	

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.3.2.3 displayText()

```
void arcade::ADisplaySfml::displayText (
    const std::string & text,
    Position pos ) [override], [virtual]
```

display a text on the screen

Parameters

<i>text</i>	
<i>pos</i>	

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).**6.3.2.4 getBlockSize()**

```
int arcade::ADisplaySfml::getBlockSize ( ) [inline]
```

6.3.2.5 getWindowSize()

```
arcade::Position arcade::ADisplaySfml::getWindowSize ( ) [override], [virtual]
```

get the size of the window

Returns

std::list<arcade::Event> list of events

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).**6.3.2.6 playSound()**

```
void arcade::ADisplaySfml::playSound (
    Sound sound ) [override], [virtual]
```

get the input from the user

Parameters

<i>sound</i>	
--------------	--

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.3.2.7 pollEvent()

```
std::list< arcade::Event > arcade::ADisplaySfml::pollEvent ( ) [override], [virtual]
```

get the input from the user

Returns

std::list<arcade::Event> list of events

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.3.2.8 refresh()

```
void arcade::ADisplaySfml::refresh ( ) [override], [virtual]
```

refresh the screen

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.3.2.9 setWindowSize()

```
void arcade::ADisplaySfml::setWindowSize (
    Position size ) [override], [virtual]
```

set the size of the window

Returns

std::list<arcade::Event> list of events

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.3.2.10 setTitle()

```
void arcade::ADisplaySfml::setTitle (
    const std::string & title ) [override], [virtual]
```

set the title of the window

Returns

std::list<arcade::Event> list of events

Author

pier-alexandre

Implements [arcade::IDisplayModule](#).

6.3.3 Member Data Documentation

6.3.3.1 _blockSize

```
int arcade::ADisplaySfml::_blockSize [private]
```

6.3.3.2 _CoinBuffer

```
sf::SoundBuffer arcade::ADisplaySfml::_CoinBuffer [private]
```

6.3.3.3 _font

```
sf::Font arcade::ADisplaySfml::_font [private]
```

6.3.3.4 _gameOverBuffer

```
sf::SoundBuffer arcade::ADisplaySfml::_gameOverBuffer [private]
```

6.3.3.5 _hurtBuffer

```
sf::SoundBuffer arcade::ADisplaySfml::_hurtBuffer [private]
```

6.3.3.6 _startBuffer

```
sf::SoundBuffer arcade::ADisplaySfml::_startBuffer [private]
```

6.3.3.7 _superBuffer

```
sf::SoundBuffer arcade::ADisplaySfml::_superBuffer [private]
```

6.3.3.8 _window

```
sf::RenderWindow arcade::ADisplaySfml::_window [private]
```

6.3.3.9 _windowSize

```
Position arcade::ADisplaySfml::_windowSize [private]
```

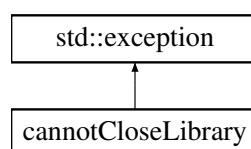
The documentation for this class was generated from the following files:

- src/Ldisplay/sfml/[ADisplaySfml.hpp](#)
- src/Ldisplay/sfml/[ADisplaySfml.cpp](#)

6.4 cannotCloseLibrary Class Reference

```
#include <errors.hpp>
```

Inheritance diagram for cannotCloseLibrary:



Public Member Functions

- virtual const char * [what](#) () const throw ()

6.4.1 Member Function Documentation

6.4.1.1 what()

```
virtual const char* cannotCloseLibrary::what ( ) const throw ( )    [inline], [virtual]
```

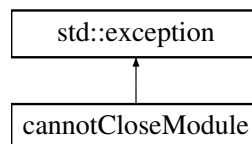
The documentation for this class was generated from the following file:

- src/errors/[errors.hpp](#)

6.5 cannotCloseModule Class Reference

```
#include <errors.hpp>
```

Inheritance diagram for cannotCloseModule:



Public Member Functions

- virtual const char * [what](#) () const throw ()

6.5.1 Member Function Documentation

6.5.1.1 what()

```
virtual const char* cannotCloseModule::what ( ) const throw ( )    [inline], [virtual]
```

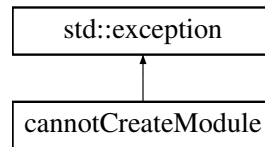
The documentation for this class was generated from the following file:

- src/errors/[errors.hpp](#)

6.6 cannotCreateModule Class Reference

```
#include <errors.hpp>
```

Inheritance diagram for cannotCreateModule:



Public Member Functions

- virtual const char * [what](#) () const throw ()

6.6.1 Member Function Documentation

6.6.1.1 what()

```
virtual const char* cannotCreateModule::what ( ) const throw ( ) [inline], [virtual]
```

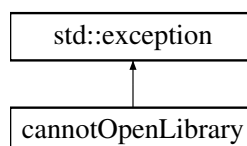
The documentation for this class was generated from the following file:

- [src/errors/errors.hpp](#)

6.7 cannotOpenLibrary Class Reference

```
#include <errors.hpp>
```

Inheritance diagram for cannotOpenLibrary:



Public Member Functions

- virtual const char * [what](#) () const throw ()

6.7.1 Member Function Documentation

6.7.1.1 what()

```
virtual const char* cannotOpenLibrary::what ( ) const throw ( )    [inline], [virtual]
```

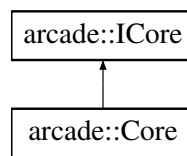
The documentation for this class was generated from the following file:

- [src/errors/errors.hpp](#)

6.8 arcade::Core Class Reference

```
#include <core.hpp>
```

Inheritance diagram for arcade::Core:



Public Member Functions

- [Core](#) ()
- [~Core](#) ()
- void [setName](#) (const std::string &name)
- const std::string & [getName](#) () const

Public Attributes

- std::map< std::string, std::pair< enum [gameLibs](#), void * > > [_gameLibs](#)
- std::map< std::string, std::pair< enum [displayLibs](#), void * > > [_displayLibs](#)

Private Attributes

- std::string [_name](#)

6.8.1 Constructor & Destructor Documentation

6.8.1.1 Core()

```
arcade::Core::Core ( ) [inline]
```

6.8.1.2 ~Core()

```
arcade::Core::~~Core ( ) [inline]
```

6.8.2 Member Function Documentation

6.8.2.1 getName()

```
const std::string& arcade::Core::getName ( ) const [inline], [virtual]
```

Get the username

Implements [arcade::ICore](#).

6.8.2.2 setName()

```
void arcade::Core::setName (
    const std::string & name ) [inline], [virtual]
```

Set the username

Implements [arcade::ICore](#).

6.8.3 Member Data Documentation

6.8.3.1 _displayLibs

```
std::map<std::string, std::pair<enum displayLibs, void *> > arcade::Core::_displayLibs
```

6.8.3.2 `_gameLibs`

```
std::map<std::string, std::pair<enum gameLibs, void *> > arcade::Core::_gameLibs
```

6.8.3.3 `_name`

```
std::string arcade::Core::_name [private]
```

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

- [src/core/core.hpp](#)

6.9 `arcade::Event` Struct Reference

```
#include <Event.hpp>
```

Public Attributes

- [EventType](#) type

6.9.1 Detailed Description

A structure which holds an event, passed from the graphical module to the game module.

6.9.2 Member Data Documentation

6.9.2.1 `type`

```
EventType arcade::Event::type
```

The type of the event which was received

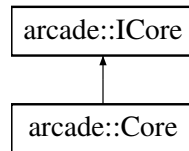
The documentation for this struct was generated from the following file:

- [src/interfaces/Event.hpp](#)

6.10 arcade::ICore Class Reference

```
#include <ICore.hpp>
```

Inheritance diagram for arcade::ICore:



Public Member Functions

- virtual void [setName](#) (const std::string &name)=0
- virtual const std::string & [getName](#) () const =0

6.10.1 Member Function Documentation

6.10.1.1 [getName\(\)](#)

```
virtual const std::string& arcade::ICore::getName ( ) const [pure virtual]
```

Get the username

Implemented in [arcade::Core](#).

6.10.1.2 [setName\(\)](#)

```
virtual void arcade::ICore::setName (
    const std::string & name ) [pure virtual]
```

Set the username

Implemented in [arcade::Core](#).

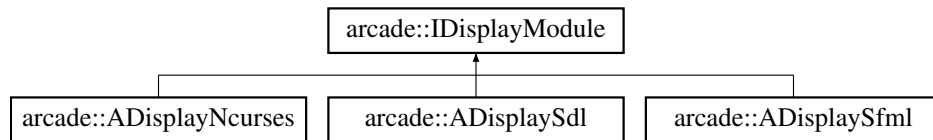
The documentation for this class was generated from the following file:

- [src/interfaces/ICore.hpp](#)

6.11 arcade::IDisplayModule Class Reference

```
#include <IDisplayModule.hpp>
```

Inheritance diagram for arcade::IDisplayModule:



Public Member Functions

- virtual [~IDisplayModule](#) ()=default
- virtual void [displayBlock](#) ([BlockType](#) blockType, [Position](#) pos)=0
- virtual void [displayText](#) (const std::string &text, [Position](#) pos)=0
- virtual void [clear](#) ()=0
- virtual void [refresh](#) ()=0
- virtual void [playSound](#) ([Sound](#) sound)=0
- virtual std::list< [Event](#) > [pollEvent](#) ()=0
- virtual void [setWindowSize](#) ([Position](#) size)=0
- virtual [Position](#) [getWindowSize](#) ()=0
- virtual void [setWindowTitle](#) (const std::string &title)=0

6.11.1 Detailed Description

A graphical module, which handles displaying the arcade games

6.11.2 Constructor & Destructor Documentation

6.11.2.1 ~IDisplayModule()

```
virtual arcade::IDisplayModule::~~IDisplayModule ( ) [virtual], [default]
```

The destructor, which is responsible for cleaning up all resources

6.11.3 Member Function Documentation

6.11.3.1 clear()

```
virtual void arcade::IDisplayModule::clear ( ) [pure virtual]
```

Reset the shown screen

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.2 displayBlock()

```
virtual void arcade::IDisplayModule::displayBlock (
    BlockType blockType,
    Position pos ) [pure virtual]
```

Display a block on screen at the specified position

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.3 displayText()

```
virtual void arcade::IDisplayModule::displayText (
    const std::string & text,
    Position pos ) [pure virtual]
```

Display a string on screen at the specified position

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.4 getWindowSize()

```
virtual Position arcade::IDisplayModule::getWindowSize ( ) [pure virtual]
```

Get window size in blocks.

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.5 playSound()

```
virtual void arcade::IDisplayModule::playSound (
    Sound sound ) [pure virtual]
```

Play a sound

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.6 pollEvent()

```
virtual std::list<Event> arcade::IDisplayModule::pollEvent ( ) [pure virtual]
```

Pop an event. The event will be removed from the list of pending events.

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.7 refresh()

```
virtual void arcade::IDisplayModule::refresh ( ) [pure virtual]
```

Refresh the screen, so that the view is updated

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.8 setWindowSize()

```
virtual void arcade::IDisplayModule::setWindowSize (
    Position size ) [pure virtual]
```

Try setting window size in blocks.

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

6.11.3.9 setWindowTitle()

```
virtual void arcade::IDisplayModule::setWindowTitle (
    const std::string & title ) [pure virtual]
```

Set the title of the window

Implemented in [arcade::ADisplaySfml](#), [arcade::ADisplaySdl](#), and [arcade::ADisplayNcurses](#).

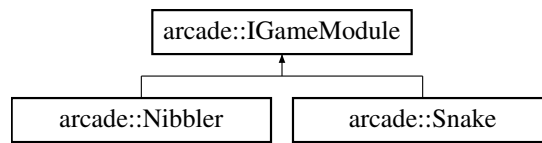
The documentation for this class was generated from the following file:

- [src/interfaces/IDisplayModule.hpp](#)

6.12 arcade::IGameModule Class Reference

```
#include <IGameModule.hpp>
```

Inheritance diagram for arcade::IGameModule:



Public Member Functions

- virtual [~IGameModule](#) ()=default
- virtual void [tick](#) ([IDisplayModule](#) &displayModule, const std::list< [Event](#) > &events)=0
- virtual std::string [getName](#) () const =0
- virtual void [start](#) ([ICore](#) &core)=0

6.12.1 Detailed Description

A game module that handle how the game works

6.12.2 Constructor & Destructor Documentation

6.12.2.1 ~IGameModule()

```
virtual arcade::IGameModule::~~IGameModule ( ) [virtual], [default]
```

The destructor, which is responsible for cleaning up all resources

6.12.3 Member Function Documentation

6.12.3.1 getName()

```
virtual std::string arcade::IGameModule::getName ( ) const [pure virtual]
```

the function which return the name, Duuuuhh~

Implemented in [arcade::Snake](#), and [arcade::Nibbler](#).

6.12.3.2 start()

```
virtual void arcade::IGameModule::start (
    ICore & core ) [pure virtual]
```

the function that start the game

Implemented in [arcade::Snake](#), and [arcade::Nibbler](#).

6.12.3.3 tick()

```
virtual void arcade::IGameModule::tick (
    IDisplayModule & displayModule,
    const std::list< Event > & events ) [pure virtual]
```

handle all command provided by [IDisplayModule](#)

Implemented in [arcade::Snake](#), and [arcade::Nibbler](#).

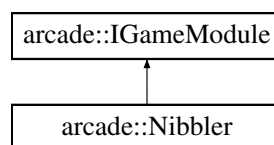
The documentation for this class was generated from the following file:

- [src/interfaces/IGameModule.hpp](#)

6.13 arcade::Nibbler Class Reference

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

Inheritance diagram for arcade::Nibbler:



Public Member Functions

- [Nibbler](#) ()
- [~Nibbler](#) ()
- void [tick](#) ([IDisplayModule](#) &displayModule, const std::list< [Event](#) > &events) override
fonction called by the core to start the game
- void [start](#) ([ICore](#) &core) override
- std::string [getName](#) () const override
- void [setMap](#) ()
- bool [placeFruits](#) (const [arcade::Position](#) &block)
- bool [headSnakeOnWall](#) (const [arcade::Position](#) &block)
- bool [isFruitOnSnake](#) (const [arcade::Position](#) &block)
- void [verifLeft](#) (float x, float y)

- void [verifRight](#) (float x, float y)
- void [verifUp](#) (float x, float y)
- void [verifDown](#) (float x, float y)
- bool [headSnakeOnSnake](#) (const [arcade::Position](#) &block)
- void [setAllFruits](#) ()
 - set new position for the fruit*
- void [printMap](#) ()
 - print _Map*
- void [printSnake](#) ()
 - print _snakeCara*
- void [printFruit](#) ()
 - print _snakeCara*
- void [setNewFruit](#) ()
- void [snakeEatFruit](#) ()
- void [printScoreAndName](#) ()
- void [printHighScore](#) ()
- void [gamePause](#) ()
- void [applyEvent](#) ()
 - parse and aply _events to the game*
- void [changeDirection](#) ()
 - change snake direction and move it*
- void [deathCondition](#) ()
- void [end](#) ()
- void [moveCorps](#) ()
 - move the snake's body to the head position*
- void [moveSnake](#) ()
 - call all move function to move the snake and check if the snake is dead or not*

Private Attributes

- [IDisplayModule](#) * [_display](#)
- std::list< [Event](#) > [_events](#)
- std::map< int, std::vector< [BlockType](#) > > [_Map](#)
- std::vector< [Position](#) > [_wall](#)
- int [_score](#) = 0
- int [_highScore](#) = 0
- float [_speed](#) = 1
- [EventType](#) [_waitDirection](#)
- [EventType](#) [_direction](#)
- std::vector< std::pair< [Event](#), [Position](#) > > [_changeDirectionList](#)
- std::vector< [Position](#) > [_fruit](#)
- std::vector< [Position](#) > [_snakeCara](#)
- std::string [_name](#)

6.13.1 Constructor & Destructor Documentation

6.13.1.1 Nibbler()

```
arcade::Nibbler::Nibbler ( )
```

6.13.1.2 ~Nibbler()

```
arcade::Nibbler::~~Nibbler ( )
```

6.13.2 Member Function Documentation

6.13.2.1 applyEvent()

```
void arcade::Nibbler::applyEvent ( )
```

parse and apply _events to the game

Author

Pier-Alexandre

6.13.2.2 changeDirection()

```
void arcade::Nibbler::changeDirection ( )
```

change snake direction and move it

Author

Pier-Alexandre

6.13.2.3 deathCondition()

```
void arcade::Nibbler::deathCondition ( )
```

6.13.2.4 end()

```
void arcade::Nibbler::end ( )
```

6.13.2.5 gamePause()

```
void arcade::Nibbler::gamePause ( )
```

6.13.2.6 getName()

```
std::string arcade::Nibbler::getName ( ) const [inline], [override], [virtual]
```

the function which return the name, Duuuuhh~

Implements [arcade::IGameModule](#).

6.13.2.7 headSnakeOnSnake()

```
bool arcade::Nibbler::headSnakeOnSnake (
    const arcade::Position & block )
```

6.13.2.8 headSnakeOnWall()

```
bool arcade::Nibbler::headSnakeOnWall (
    const arcade::Position & block )
```

6.13.2.9 isFruitOnSnake()

```
bool arcade::Nibbler::isFruitOnSnake (
    const arcade::Position & block )
```

6.13.2.10 moveCorps()

```
void arcade::Nibbler::moveCorps ( )
```

move the snake's body to the head position

Author

Pier-Alexendre

6.13.2.11 moveSnake()

```
void arcade::Nibbler::moveSnake ( )
```

call all move function to move the snake and check if the snake is dead or not

Author

Pier-Alexendre

6.13.2.12 placeFruits()

```
bool arcade::Nibbler::placeFruits (
    const arcade::Position & block )
```

6.13.2.13 printFruit()

```
void arcade::Nibbler::printFruit ( )
```

print _snakeCara

Author

Pier-Alexendre

6.13.2.14 printHighScore()

```
void arcade::Nibbler::printHighScore ( ) [inline]
```

6.13.2.15 printMap()

```
void arcade::Nibbler::printMap ( )
```

print_Map

Author

Pier-Alexendre

6.13.2.16 printScoreAndName()

```
void arcade::Nibbler::printScoreAndName ( )
```

6.13.2.17 printSnake()

```
void arcade::Nibbler::printSnake ( )
```

print_snakeCara

Author

Pier-Alexendre

6.13.2.18 setAllFruits()

```
void arcade::Nibbler::setAllFruits ( )
```

set new position for the fruit

Author

Pier-Alexendre

6.13.2.19 setMap()

```
void arcade::Nibbler::setMap ( )
```

6.13.2.20 setNewFruit()

```
void arcade::Nibbler::setNewFruit ( )
```

6.13.2.21 snakeEatFruit()

```
void arcade::Nibbler::snakeEatFruit ( )
```

6.13.2.22 start()

```
void arcade::Nibbler::start (
    ICore & core ) [override], [virtual]
```

the function that start the game

Implements [arcade::IGameModule](#).

6.13.2.23 tick()

```
void arcade::Nibbler::tick (
    IDisplayModule & displayModule,
    const std::list< Event > & events ) [override], [virtual]
```

fonction called by the core to start the game

Parameters

<i>_display</i>	the display module
<i>list</i>	of events

Author

Pier-Alexandre

Implements [arcade::IGameModule](#).

6.13.2.24 verifDown()

```
void arcade::Nibbler::verifDown (
    float x,
    float y )
```

6.13.2.25 verifLeft()

```
void arcade::Nibbler::verifLeft (
    float x,
    float y )
```

6.13.2.26 verifRight()

```
void arcade::Nibbler::verifRight (
    float x,
    float y )
```

6.13.2.27 verifUp()

```
void arcade::Nibbler::verifUp (
    float x,
    float y )
```

6.13.3 Member Data Documentation

6.13.3.1 _changeDirectionList

```
std::vector<std::pair<Event, Position> > arcade::Nibbler::_changeDirectionList [private]
```

6.13.3.2 _direction

```
EventType arcade::Nibbler::_direction [private]
```

6.13.3.3 _display

```
IDisplayModule* arcade::Nibbler::_display [private]
```

6.13.3.4 `_events`

```
std::list<Event> arcade::Nibbler::_events [private]
```

6.13.3.5 `_fruit`

```
std::vector<Position> arcade::Nibbler::_fruit [private]
```

6.13.3.6 `_highScore`

```
int arcade::Nibbler::_highScore = 0 [private]
```

6.13.3.7 `_Map`

```
std::map<int, std::vector<BlockType> > arcade::Nibbler::_Map [private]
```

6.13.3.8 `_name`

```
std::string arcade::Nibbler::_name [private]
```

6.13.3.9 `_score`

```
int arcade::Nibbler::_score = 0 [private]
```

6.13.3.10 `_snakeCara`

```
std::vector<Position> arcade::Nibbler::_snakeCara [private]
```

6.13.3.11 `_speed`

```
float arcade::Nibbler::_speed = 1 [private]
```


6.13.3.12 `_waitDirection`

```
EventType arcade::Nibbler::_waitDirection [private]
```

6.13.3.13 `_wall`

```
std::vector<Position> arcade::Nibbler::_wall [private]
```

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

- src/Lgame/nibbler/[nibbler.hpp](#)
- src/Lgame/nibbler/[nibbler.cpp](#)

6.14 arcade::Position Struct Reference

```
#include <IDisplayModule.hpp>
```

Public Attributes

- float [x](#)
- float [y](#)

6.14.1 Detailed Description

A 2D position, in block units

6.14.2 Member Data Documentation

6.14.2.1 `x`

```
float arcade::Position::x
```

6.14.2.2 `y`

```
float arcade::Position::y
```

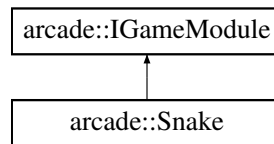
The documentation for this struct was generated from the following file:

- src/interfaces/[IDisplayModule.hpp](#)

6.15 arcade::Snake Class Reference

```
#include <snake.hpp>
```

Inheritance diagram for arcade::Snake:



Public Member Functions

- [Snake](#) ()
Construct a new [arcade::Snake::Snake](#) object.
- [~Snake](#) ()
Destroy the [arcade::Snake::Snake](#) object.
- void [tick](#) ([IDisplayModule](#) &displayModule, const std::list< [Event](#) > &events) override
fonction called by the core to start the game
- void [start](#) ([ICore](#) &core) override
- std::string [getName](#) () const override
- void [printMap](#) ()
print _Map
- void [printSnake](#) ()
print _snakeCara
- void [printFruit](#) ()
print the fruit on the map
- void [setNewFruit](#) ()
set new position for the fruit
- void [snakeEatFruit](#) ()
check if the snake eat the fruit and if yes, add a new block to the snake's body and set a new fruit position
- void [printScoreAndName](#) ()
- void [printHighScore](#) ()
- void [gamePause](#) ()
pause the game and wait for a pause event to resume the game
- void [applyEvent](#) ()
parse and aplly _events to the game
- void [changeDirection](#) ()
change snake direction and move it
- void [deathCondition](#) ()
- void [moveCorps](#) ()
move the snake's body to the head position
- void [moveSnake](#) ()
call all move function to move the snake and check if the snake is dead or not
- void [end](#) ()

Private Attributes

- [IDisplayModule](#) * [_display](#)
- [std::list](#)< [Event](#) > [_events](#)
- [std::map](#)< [int](#), [std::vector](#)< [BlockType](#) > > [_Map](#)
- [int](#) [_score](#)
- [int](#) [_highScore](#) = 0
- [float](#) [_speed](#) = 1
- [EventType](#) [_waitDirection](#)
- [EventType](#) [_direction](#)
- [std::vector](#)< [std::pair](#)< [Event](#), [Position](#) > > [_changeDirectionList](#)
- [Position](#) [_fruit](#)
- [std::vector](#)< [Position](#) > [_snakeCara](#)
- [std::string](#) [_name](#)

6.15.1 Constructor & Destructor Documentation

6.15.1.1 Snake()

```
arcade::Snake::Snake ( )
```

Construct a new [arcade::Snake::Snake](#) object.

Author

Pier-Alexandre

6.15.1.2 ~Snake()

```
arcade::Snake::~Snake ( )
```

Destroy the [arcade::Snake::Snake](#) object.

Author

Pier-Alexandre

6.15.2 Member Function Documentation

6.15.2.1 applyEvent()

```
void arcade::Snake::applyEvent ( )
```

parse and apply _events to the game

Author

Pier-Alexandre

6.15.2.2 changeDirection()

```
void arcade::Snake::changeDirection ( )
```

change snake direction and move it

Author

Pier-Alexandre

6.15.2.3 deathCondition()

```
void arcade::Snake::deathCondition ( )
```

6.15.2.4 end()

```
void arcade::Snake::end ( )
```

6.15.2.5 gamePause()

```
void arcade::Snake::gamePause ( )
```

pause the game and wait for a pause event to resume the game

Author

Pier-Alexandre

6.15.2.6 getName()

```
std::string arcade::Snake::getName ( ) const [inline], [override], [virtual]
```

the function which return the name, Duuuuhh~

Implements [arcade::IGameModule](#).

6.15.2.7 moveCorps()

```
void arcade::Snake::moveCorps ( )
```

move the snake's body to the head position

Author

Pier-Alexendre

6.15.2.8 moveSnake()

```
void arcade::Snake::moveSnake ( )
```

call all move function to move the snake and check if the snake is dead or not

Author

Pier-Alexendre

6.15.2.9 printFruit()

```
void arcade::Snake::printFruit ( )
```

print the fruit on the map

Author

Pier-Alexendre

6.15.2.10 printHighScore()

```
void arcade::Snake::printHighScore ( ) [inline]
```

6.15.2.11 printMap()

```
void arcade::Snake::printMap ( )
```

print_Map

Author

Pier-Alexendre

6.15.2.12 printScoreAndName()

```
void arcade::Snake::printScoreAndName ( )
```

6.15.2.13 printSnake()

```
void arcade::Snake::printSnake ( )
```

print_snakeCara

Author

Pier-Alexendre

6.15.2.14 setNewFruit()

```
void arcade::Snake::setNewFruit ( )
```

set new position for the fruit

Author

Pier-Alexendre

6.15.2.15 snakeEatFruit()

```
void arcade::Snake::snakeEatFruit ( )
```

check if the snake eat the fruit and if yes, add a new block to the snake's body and set a new fruit position

Author

Pier-Alexandre

6.15.2.16 start()

```
void arcade::Snake::start (
    ICore & core ) [override], [virtual]
```

the function that start the game

Implements [arcade::IGameModule](#).

6.15.2.17 tick()

```
void arcade::Snake::tick (
    IDisplayModule & displayModule,
    const std::list< Event > & events ) [override], [virtual]
```

fonction called by the core to start the game

Parameters

<i>_display</i>	the display module
<i>list</i>	of events

Author

Pier-Alexandre

Implements [arcade::IGameModule](#).

6.15.3 Member Data Documentation

6.15.3.1 `_changeDirectionList`

```
std::vector<std::pair<Event, Position> > arcade::Snake::_changeDirectionList [private]
```

6.15.3.2 `_direction`

```
EventType arcade::Snake::_direction [private]
```

6.15.3.3 `_display`

```
IDisplayModule* arcade::Snake::_display [private]
```

6.15.3.4 `_events`

```
std::list<Event> arcade::Snake::_events [private]
```

6.15.3.5 `_fruit`

```
Position arcade::Snake::_fruit [private]
```

6.15.3.6 `_highScore`

```
int arcade::Snake::_highScore = 0 [private]
```

6.15.3.7 `_Map`

```
std::map<int, std::vector<BlockType> > arcade::Snake::_Map [private]
```

6.15.3.8 `_name`

```
std::string arcade::Snake::_name [private]
```


6.15.3.9 `_score`

```
int arcade::Snake::_score [private]
```

6.15.3.10 `_snakeCara`

```
std::vector<Position> arcade::Snake::_snakeCara [private]
```

6.15.3.11 `_speed`

```
float arcade::Snake::_speed = 1 [private]
```

6.15.3.12 `_waitDirection`

```
EventType arcade::Snake::_waitDirection [private]
```

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

- `src/Lgame/snake/snake.hpp`
- `src/Lgame/snake/snake.cpp`

Chapter 7

File Documentation

7.1 src/core/chargeLib.cpp File Reference

```
#include "core.hpp"
```

Namespaces

- [arcade](#)

Functions

- IDisplayModule * [arcade::createDisplayModule](#) (void *lib)
Create a Display Module object.
- IGameModule * [arcade::createGameModule](#) (void *lib)
Create a Game Module object.
- std::string [arcade::nextIndex](#) (std::vector< std::string > lib, std::string curent)
return the next index of a vector and the first one if it's the last
- std::string [arcade::previousIndex](#) (std::vector< std::string > lib, std::string curent)
return the previous index of a vector and the last one if it's the first

7.2 src/core/closeLib.cpp File Reference

```
#include "core.hpp"
```

Namespaces

- [arcade](#)

Functions

- void [arcade::deleteAllModules](#) (IDisplayModule *displayModule, IGameModule *gameModule)
delete a display and a game module
- void [arcade::closeAllLibs](#) (Core *core)
Close all libraries.

7.3 src/core/core.cpp File Reference

```
#include "core.hpp"
#include "../errors/errors.hpp"
#include <iostream>
#include <chrono>
#include <thread>
#include <fstream>
```

Namespaces

- [arcade](#)

Functions

- enum libChange [arcade::checkLibChange](#) (const std::list< Event > &events)
check if the lib has to change
- int [arcade::gameLoop](#) (Core *core, IDisplayModule *displayModule, IGameModule *gameModule, std::vector< std::string > games, std::vector< std::string > displays, std::string gameName, std::string displayName)
game loop
- int [arcade::core](#) (char *displayModule)
Core of the program were display and game modules are created,linked and deleted.
- void [help](#) ()
display help information
- int [main](#) (int argc, char **argv)
Main function of the program.

7.3.1 Function Documentation

7.3.1.1 help()

```
void help ( )
```

display help information

Author

leo

7.3.1.2 main()

```
int main (
    int argc,
    char ** argv )
```

Main function of the program.

Author

Evan

Parameters

<i>argc</i>	
<i>argv</i>	

Returns

int

7.4 src/core/core.hpp File Reference

```
#include "../interfaces/IDisplayModule.hpp"
#include "../interfaces/IGameModule.hpp"
#include "../interfaces/ICore.hpp"
#include "../errors/errors.hpp"
#include <iostream>
#include <vector>
#include <map>
#include <filesystem>
#include <tuple>
#include <dlfcn.h>
#include <cstring>
```

Classes

- class [arcade::Core](#)

Namespaces

- [arcade](#)

Enumerations

- enum `arcade::gameLibs` {
`arcade::Snake` , `arcade::Nibbler` , `arcade::Pacman` , `arcade::Centipede` ,
`arcade::Qix` , `arcade::Solarfox` }
- enum `arcade::displayLibs` {
`arcade::Ncurses` , `arcade::SFML` , `arcade::SDL2` , `arcade::Aalib` ,
`arcade::Caca` , `arcade::OpenGL` , `arcade::Allegro` , `arcade::Gtk` ,
`arcade::Xlib` , `arcade::Irrlicht` , `arcade::Qt5` , `arcade::Vulkain` ,
`arcade::Ndk` }
- enum class `arcade::libChange` {
`arcade::NextGame` , `arcade::PreviousGame` , `arcade::NextLibrary` , `arcade::PreviousLibrary` ,
`arcade::ReturnToSelectionMenu` , `arcade::Restart` , `arcade::Quit` , `arcade::None` }

Functions

- `IDisplayModule * arcade::createDisplayModule` (void *lib)
Create a Display Module object.
- `IGameModule * arcade::createGameModule` (void *lib)
Create a Game Module object.
- `IGameModule * arcade::nextGame` (Core *core, IGameModule *gameModule, std::vector< std::string > &gameLibs)
- `IDisplayModule * arcade::nextDisplay` (Core *core, IDisplayModule *displayModule, std::vector< std::string > &displayLibs)
- `std::string arcade::previousIndex` (std::vector< std::string > lib, std::string curent)
return the previous index of a vector and the last one if it's the first
- `std::string arcade::nextIndex` (std::vector< std::string > lib, std::string curent)
return the next index of a vector and the first one if it's the last
- `Core * arcade::chargeAllLibs` (void)
Get the All Libs object and fill core->_displayLibs and core->_gameLibs.
- `void * arcade::openLib` (const char *libName)
Open a library.
- `void arcade::closeAllLibs` (Core *core)
Close all libraries.
- `void arcade::deleteAllModules` (IDisplayModule *displayModule, IGameModule *gameModule)
delete a display and a game module
- `std::string arcade::findDisplayLib` (Core *core, char *file)
Find a display library.
- `std::string arcade::findDisplay` (Core *core, enum displayLibs libs)
Find a display library.
- `std::string arcade::findGame` (Core *core, enum gameLibs libs)
Find a game library.
- `int arcade::Menu` (IDisplayModule *display, Core *core, std::string firstDisplay)
Print the menu and handle the events. THIS FUNCTION CALL THE GAME LOOP.
- `std::string arcade::getGameLibName` (enum gameLibs gameLib)
Get the Game Lib Name object.
- `std::string arcade::getDisplayLibName` (enum displayLibs displayLib)
Get the Display Lib Name object.
- `int arcade::gameLoop` (Core *core, IDisplayModule *displayModule, IGameModule *gameModule, std::vector< std::string > games, std::vector< std::string > displays, std::string gameName, std::string displayName)
game loop

7.5 src/core/findLib.cpp File Reference

```
#include "core.hpp"
```

Namespaces

- [arcade](#)

Functions

- `std::string arcade::findDisplayLib` (`Core *core`, `char *file`)
Find a display library.
- `std::string arcade::findGame` (`Core *core`, `enum gameLibs libs`)
Find a game library.
- `std::string arcade::findDisplay` (`Core *core`, `enum displayLibs libs`)
Find a display library.

7.6 src/core/initLib.cpp File Reference

```
#include "core.hpp"
```

Namespaces

- [arcade](#)

Functions

- `void * arcade::openLib` (`const char *libName`)
Open a library.
- `Core * arcade::chargeAllLibs` (`void`)
Get the All Libs object and fill core->_displayLibs and core->_gameLibs.

7.7 src/core/menu.cpp File Reference

```
#include "core.hpp"  
#include <vector>
```

Namespaces

- [arcade](#)

Enumerations

- enum [arcade::changes](#) {
[arcade::Left](#) , [arcade::Right](#) , [arcade::Up](#) , [arcade::Down](#) ,
[arcade::None](#) , [arcade::None](#) }

Functions

- std::string [arcade::getGameLibName](#) (enum gameLibs gameLib)
Get the Game Lib Name object.
- std::string [arcade::getDisplayLibName](#) (enum displayLibs displayLib)
Get the Display Lib Name object.
- std::pair< int, int > [arcade::changePos](#) (const std::list< Event > &events, std::pair< int, int > pos, std::vector< std::string > games, std::vector< std::string > displays)
highlight the selected game or library
- std::pair< int, int > [arcade::displayMenu](#) (std::pair< int, int > pos, std::list< Event > events, IDisplayModule *displayModule, std::vector< std::string > games, std::vector< std::string > displays, std::string name)
display the menu and highlight the selected game or library
- std::vector< std::string > [arcade::fillGames](#) (Core *core)
fill the vector with the games names
- std::vector< std::string > [arcade::fillDisplays](#) (Core *core)
fill the vector with the displays names
- int [arcade::Menu](#) (IDisplayModule *display, Core *core, std::string firstDisplay)
Print the menu and handle the events. THIS FUNCTION CALL THE GAME LOOP.

7.8 src/errors/errors.hpp File Reference

```
#include <exception>
#include <string>
```

Classes

- class [cannotOpenLibrary](#)
- class [cannotCloseLibrary](#)
- class [cannotCreateModule](#)
- class [cannotCloseModule](#)

7.9 src/interfaces/Event.hpp File Reference

Classes

- struct [arcade::Event](#)

Namespaces

- [arcade](#)

Enumerations

- enum class [arcade::EventType](#) {
 [arcade::Up](#) , [arcade::Down](#) , [arcade::Left](#) , [arcade::Right](#) ,
 [arcade::Action](#) , [arcade::Pause](#) , [arcade::Quit](#) , [arcade::SwitchToNextGame](#) ,
 [arcade::SwitchToPreviousGame](#) , [arcade::ReturnToSelectionMenu](#) , [arcade::NextLibrary](#) , [arcade::PreviousLibrary](#)
 ,
 [arcade::RestartGame](#) }

7.10 src/interfaces/ICore.hpp File Reference

```
#include <string>
```

Classes

- class [arcade::ICore](#)

Namespaces

- [arcade](#)

7.11 src/interfaces/IDisplayModule.hpp File Reference

```
#include <string>  
#include <list>  
#include "Event.hpp"
```

Classes

- struct [arcade::Position](#)
- class [arcade::IDisplayModule](#)

Namespaces

- [arcade](#)

Enumerations

- enum class [arcade::BlockType](#) {
 [arcade::BrickWall](#) , [arcade::Grass](#) , [arcade::Fruit](#) , [arcade::Coin](#) ,
 [arcade::Character](#) , [arcade::Enemy](#) , [arcade::EnemyVulnerable](#) , [arcade::EnemyDead](#) }
- enum class [arcade::Sound](#) {
 [arcade::Coin](#) , [arcade::Hurt](#) , [arcade::GameOver](#) , [arcade::Start](#) ,
 [arcade::Super](#) }

Functions

- [arcade::IDisplayModule * createDisplayModule \(\)](#)

7.11.1 Function Documentation

7.11.1.1 createDisplayModule()

```
arcade::IDisplayModule* createDisplayModule ( )
```

The entry point of the library, which is responsible for creating the display module

7.12 src/interfaces/IGameModule.hpp File Reference

```
#include "IDisplayModule.hpp"  
#include "ICore.hpp"
```

Classes

- class [arcade::IGameModule](#)

Namespaces

- [arcade](#)

Functions

- [arcade::IGameModule * createGameModule \(\)](#)

7.12.1 Function Documentation

7.12.1.1 createGameModule()

```
arcade::IGameModule* createGameModule ( )
```

The entry point of the library, which is responsible for creating the game module

7.13 src/Ldisplay/ncurses/ADisplayNcurses.cpp File Reference

```
#include "ADisplayNcurses.hpp"
```

Functions

- [arcade::IDisplayModule * createDisplayModule \(\)](#)

7.13.1 Function Documentation

7.13.1.1 createDisplayModule()

```
arcade::IDisplayModule* createDisplayModule ( )
```

The entry point of the library, which is responsible for creating the display module

7.14 src/Ldisplay/ncurses/ADisplayNcurses.hpp File Reference

```
#include <ncurses.h>
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include "../interfaces/IDisplayModule.hpp"
```

Classes

- class [arcade::ADisplayNcurses](#)

Namespaces

- [arcade](#)

7.15 src/Ldisplay/sdl/ADisplaySdl.cpp File Reference

```
#include "ADisplaySdl.hpp"
```

Functions

- [arcade::IDisplayModule * createDisplayModule \(\)](#)

7.15.1 Function Documentation

7.15.1.1 createDisplayModule()

```
arcade::IDisplayModule* createDisplayModule ( )
```

The entry point of the library, which is responsible for creating the display module

7.16 src/Ldisplay/sdl/ADisplaySdl.hpp File Reference

```
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_mixer.h>
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include "../interfaces/IDisplayModule.hpp"
```

Classes

- class [arcade::ADisplaySdl](#)

Namespaces

- [arcade](#)

7.17 src/Ldisplay/sfml/ADisplaySfml.cpp File Reference

```
#include "ADisplaySfml.hpp"
```

Functions

- [arcade::IDisplayModule * createDisplayModule \(\)](#)

7.17.1 Function Documentation

7.17.1.1 createDisplayModule()

```
arcade::IDisplayModule* createDisplayModule ( )
```

The entry point of the library, which is responsible for creating the display module

7.18 src/Ldisplay/sfml/ADisplaySfml.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include "../interfaces/IDisplayModule.hpp"
```

Classes

- class [arcade::ADisplaySfml](#)

Namespaces

- [arcade](#)

7.19 src/Lgame/nibbler/nibbler.cpp File Reference

```
#include "nibbler.hpp"
```

Functions

- void [addVector](#) (std::vector< [arcade::Position](#) > &vec1, const std::vector< [arcade::Position](#) > &vec2)
- [arcade::IGameModule](#) * [createGameModule](#) ()

7.19.1 Function Documentation

7.19.1.1 addVector()

```
void addVector (
    std::vector< arcade::Position > & vec1,
    const std::vector< arcade::Position > & vec2 )
```

7.19.1.2 createGameModule()

```
arcade::IGameModule* createGameModule ( )
```

The entry point of the library, which is responsible for creating the game module

7.20 src/Lgame/nibbler/nibbler.hpp File Reference

```
#include <iostream>
#include <vector>
#include <map>
#include <cmath>
#include <string>
#include "../interfaces/IGameModule.hpp"
```

Classes

- class [arcade::Nibbler](#)

Namespaces

- [arcade](#)

7.21 src/Lgame/snake/snake.cpp File Reference

```
#include "snake.hpp"
```

Functions

- bool [headOnKillBloc](#) (const [arcade::Position](#) &snakeHead, const [arcade::Position](#) &bloc)
- bool [isFruitOnSnake](#) (const std::vector< [arcade::Position](#) > &snakeCara, const [arcade::Position](#) &fruit)
check if the snake is dead or not
- [arcade::IGameModule](#) * [createGameModule](#) ()

7.21.1 Function Documentation

7.21.1.1 createGameModule()

```
arcade::IGameModule* createGameModule ( )
```

The entry point of the library, which is responsible for creating the game module

7.21.1.2 headOnKillBloc()

```
bool headOnKillBloc (
    const arcade::Position & snakeHead,
    const arcade::Position & bloc )
```

7.21.1.3 isFruitOnSnake()

```
bool isFruitOnSnake (
    const std::vector< arcade::Position > & snakeCara,
    const arcade::Position & fruit )
```

check if the snake is dead or not

Parameters

<i>_snakeCara</i>	the snake's body
<i>position</i>	of block

Returns

true if the snake is on block
false if the snake is not on block

Author

Pier-Alexandre

7.22 src/Lgame/snake/snake.hpp File Reference

```
#include <iostream>
#include <vector>
#include <map>
#include <cmath>
#include <string>
#include "../interfaces/IGameModule.hpp"
```

Classes

- class [arcade::Snake](#)

Namespaces

- [arcade](#)

Index

- [_CoinBuffer](#)
 - [arcade::ADisplaySdl, 35](#)
 - [arcade::ADisplaySfml, 42](#)
 - [_Map](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 70](#)
 - [_blockSize](#)
 - [arcade::ADisplayNcurses, 29](#)
 - [arcade::ADisplaySdl, 35](#)
 - [arcade::ADisplaySfml, 42](#)
 - [_changeDirectionList](#)
 - [arcade::Nibbler, 61](#)
 - [arcade::Snake, 69](#)
 - [_direction](#)
 - [arcade::Nibbler, 61](#)
 - [arcade::Snake, 70](#)
 - [_display](#)
 - [arcade::Nibbler, 61](#)
 - [arcade::Snake, 70](#)
 - [_displayLibs](#)
 - [arcade::Core, 47](#)
 - [_events](#)
 - [arcade::Nibbler, 61](#)
 - [arcade::Snake, 70](#)
 - [_font](#)
 - [arcade::ADisplaySdl, 35](#)
 - [arcade::ADisplaySfml, 42](#)
 - [_fruit](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 70](#)
 - [_gameLibs](#)
 - [arcade::Core, 47](#)
 - [_gameOverBuffer](#)
 - [arcade::ADisplaySdl, 35](#)
 - [arcade::ADisplaySfml, 42](#)
 - [_highScore](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 70](#)
 - [_hurtBuffer](#)
 - [arcade::ADisplaySdl, 36](#)
 - [arcade::ADisplaySfml, 42](#)
 - [_name](#)
 - [arcade::Core, 48](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 70](#)
 - [_renderer](#)
 - [arcade::ADisplaySdl, 36](#)
 - [_score](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 70](#)
 - [_snakeCara](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 71](#)
 - [_speed](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 71](#)
 - [_spriteSheet](#)
 - [arcade::ADisplaySdl, 36](#)
 - [_spriteSheetRect](#)
 - [arcade::ADisplaySdl, 36](#)
 - [_startBuffer](#)
 - [arcade::ADisplaySdl, 36](#)
 - [arcade::ADisplaySfml, 43](#)
 - [_superBuffer](#)
 - [arcade::ADisplaySdl, 36](#)
 - [arcade::ADisplaySfml, 43](#)
 - [_waitDirection](#)
 - [arcade::Nibbler, 62](#)
 - [arcade::Snake, 71](#)
 - [_wall](#)
 - [arcade::Nibbler, 63](#)
 - [_window](#)
 - [arcade::ADisplaySdl, 36](#)
 - [arcade::ADisplaySfml, 43](#)
 - [_windowSize](#)
 - [arcade::ADisplayNcurses, 29](#)
 - [arcade::ADisplaySdl, 36](#)
 - [arcade::ADisplaySfml, 43](#)
 - [_windowTitle](#)
 - [arcade::ADisplayNcurses, 29](#)
 - [arcade::ADisplaySdl, 37](#)
 - [~ADisplayNcurses](#)
 - [arcade::ADisplayNcurses, 26](#)
 - [~ADisplaySdl](#)
 - [arcade::ADisplaySdl, 31](#)
 - [~ADisplaySfml](#)
 - [arcade::ADisplaySfml, 38](#)
 - [~Core](#)
 - [arcade::Core, 47](#)
 - [~IDisplayModule](#)
 - [arcade::IDisplayModule, 50](#)
 - [~IGameModule](#)
 - [arcade::IGameModule, 53](#)
 - [~Nibbler](#)
 - [arcade::Nibbler, 56](#)
 - [~Snake](#)
 - [arcade::Snake, 65](#)
- [Aalib](#)

- arcade, [12](#)
- Action
 - arcade, [12](#)
- addVector
 - nibbler.cpp, [83](#)
- ADisplayNcurses
 - arcade::ADisplayNcurses, [26](#)
- ADisplayNcurses.cpp
 - createDisplayModule, [81](#)
- ADisplaySdl
 - arcade::ADisplaySdl, [31](#)
- ADisplaySdl.cpp
 - createDisplayModule, [82](#)
- ADisplaySfml
 - arcade::ADisplaySfml, [38](#)
- ADisplaySfml.cpp
 - createDisplayModule, [83](#)
- Allegro
 - arcade, [12](#)
- applyEvent
 - arcade::Nibbler, [56](#)
 - arcade::Snake, [65](#)
- arcade, [9](#)
 - Aalib, [12](#)
 - Action, [12](#)
 - Allegro, [12](#)
 - BlockType, [11](#)
 - BrickWall, [11](#)
 - Caca, [12](#)
 - Centipede, [13](#)
 - changePos, [13](#)
 - changes, [11](#)
 - Character, [11](#)
 - chargeAllLibs, [14](#)
 - checkLibChange, [14](#)
 - closeAllLibs, [15](#)
 - Coin, [11](#), [13](#)
 - core, [15](#)
 - createDisplayModule, [16](#)
 - createGameModule, [16](#)
 - deleteAllModules, [17](#)
 - displayLibs, [11](#)
 - displayMenu, [17](#)
 - Down, [11](#), [12](#)
 - Enemy, [11](#)
 - EnemyDead, [11](#)
 - EnemyVulnerable, [11](#)
 - EventType, [12](#)
 - fillDisplays, [18](#)
 - fillGames, [18](#)
 - findDisplay, [18](#)
 - findDisplayLib, [19](#)
 - findGame, [19](#)
 - Fruit, [11](#)
 - gameLibs, [12](#)
 - gameLoop, [20](#)
 - GameOver, [13](#)
 - getDisplayLibName, [20](#)
 - getGameLibName, [21](#)
 - Grass, [11](#)
 - Gtk, [12](#)
 - Hurt, [13](#)
 - Irrlicht, [12](#)
 - Left, [11](#), [12](#)
 - libChange, [13](#)
 - Menu, [21](#)
 - Ncurses, [12](#)
 - Ndk, [12](#)
 - nextDisplay, [22](#)
 - NextGame, [13](#)
 - nextGame, [22](#)
 - nextIndex, [22](#)
 - NextLibrary, [12](#), [13](#)
 - Nibbler, [13](#)
 - None, [11](#), [13](#)
 - OpenGL, [12](#)
 - openLib, [23](#)
 - Pacman, [13](#)
 - Pause, [12](#)
 - PreviousGame, [13](#)
 - previousIndex, [23](#)
 - PreviousLibrary, [12](#), [13](#)
 - Qix, [13](#)
 - Qt5, [12](#)
 - Quit, [12](#), [13](#)
 - Restart, [13](#)
 - RestartGame, [12](#)
 - ReturnToSelectionMenu, [12](#), [13](#)
 - Right, [11](#), [12](#)
 - SDL2, [12](#)
 - SFML, [12](#)
 - Snake, [13](#)
 - Solarfox, [13](#)
 - Sound, [13](#)
 - Start, [13](#)
 - Super, [13](#)
 - SwitchToNextGame, [12](#)
 - SwitchToPreviousGame, [12](#)
 - Up, [11](#), [12](#)
 - Vulkain, [12](#)
 - Xlib, [12](#)
- arcade::ADisplayNcurses, [25](#)
 - _blockSize, [29](#)
 - _windowSize, [29](#)
 - _windowTitle, [29](#)
 - ~ADisplayNcurses, [26](#)
 - ADisplayNcurses, [26](#)
 - clear, [26](#)
 - displayBlock, [26](#)
 - displayText, [27](#)
 - getBlockSize, [27](#)
 - getWindowSize, [27](#)
 - playSound, [27](#)
 - pollEvent, [28](#)
 - refresh, [28](#)
 - setWindowSize, [28](#)

- setWindowTitle, 29
- arcade::ADisplaySdl, 30
 - _CoinBuffer, 35
 - _blockSize, 35
 - _font, 35
 - _gameOverBuffer, 35
 - _hurtBuffer, 36
 - _renderer, 36
 - _spriteSheet, 36
 - _spriteSheetRect, 36
 - _startBuffer, 36
 - _superBuffer, 36
 - _window, 36
 - _windowSize, 36
 - _windowTitle, 37
 - ~ADisplaySdl, 31
- ADisplaySdl, 31
 - clear, 31
 - close, 32
 - displayBlock, 32
 - displayText, 32
 - getBlockSize, 33
 - getWindowSize, 33
 - playSound, 33
 - pollEvent, 34
 - refresh, 34
 - setWindowSize, 34
 - setWindowTitle, 35
- arcade::ADisplaySfml, 37
 - _CoinBuffer, 42
 - _blockSize, 42
 - _font, 42
 - _gameOverBuffer, 42
 - _hurtBuffer, 42
 - _startBuffer, 43
 - _superBuffer, 43
 - _window, 43
 - _windowSize, 43
 - ~ADisplaySfml, 38
- ADisplaySfml, 38
 - clear, 38
 - displayBlock, 39
 - displayText, 39
 - getBlockSize, 40
 - getWindowSize, 40
 - playSound, 40
 - pollEvent, 40
 - refresh, 41
 - setWindowSize, 41
 - setWindowTitle, 41
- arcade::Core, 46
 - _displayLibs, 47
 - _gameLibs, 47
 - _name, 48
 - ~Core, 47
- Core, 46
 - getName, 47
 - setName, 47
- arcade::Event, 48
 - type, 48
- arcade::ICore, 49
 - getName, 49
 - setName, 49
- arcade::IDisplayModule, 50
 - ~IDisplayModule, 50
 - clear, 50
 - displayBlock, 51
 - displayText, 51
 - getWindowSize, 51
 - playSound, 51
 - pollEvent, 51
 - refresh, 52
 - setWindowSize, 52
 - setWindowTitle, 52
- arcade::IGameModule, 53
 - ~IGameModule, 53
 - getName, 53
 - start, 53
 - tick, 54
- arcade::Nibbler, 54
 - _Map, 62
 - _changeDirectionList, 61
 - _direction, 61
 - _display, 61
 - _events, 61
 - _fruit, 62
 - _highScore, 62
 - _name, 62
 - _score, 62
 - _snakeCara, 62
 - _speed, 62
 - _waitDirection, 62
 - _wall, 63
 - ~Nibbler, 56
 - applyEvent, 56
 - changeDirection, 56
 - deathCondition, 56
 - end, 56
 - gamePause, 57
 - getName, 57
 - headSnakeOnSnake, 57
 - headSnakeOnWall, 57
 - isFruitOnSnake, 57
 - moveCorps, 57
 - moveSnake, 58
 - Nibbler, 55
 - placeFruits, 58
 - printFruit, 58
 - printHighScore, 58
 - printMap, 58
 - printScoreAndName, 59
 - printSnake, 59
 - setAllFruits, 59
 - setMap, 59
 - setNewFruit, 59
 - snakeEatFruit, 60

- start, 60
- tick, 60
- verifDown, 60
- verifLeft, 60
- verifRight, 61
- verifUp, 61
- arcade::Position, 63
 - x, 63
 - y, 63
- arcade::Snake, 64
 - _Map, 70
 - _changeDirectionList, 69
 - _direction, 70
 - _display, 70
 - _events, 70
 - _fruit, 70
 - _highScore, 70
 - _name, 70
 - _score, 70
 - _snakeCara, 71
 - _speed, 71
 - _waitDirection, 71
 - ~Snake, 65
 - applyEvent, 65
 - changeDirection, 66
 - deathCondition, 66
 - end, 66
 - gamePause, 66
 - getName, 66
 - moveCorps, 67
 - moveSnake, 67
 - printFruit, 67
 - printHighScore, 67
 - printMap, 68
 - printScoreAndName, 68
 - printSnake, 68
 - setNewFruit, 68
 - Snake, 65
 - snakeEatFruit, 68
 - start, 69
 - tick, 69
- BlockType
 - arcade, 11
- BrickWall
 - arcade, 11
- Caca
 - arcade, 12
- cannotCloseLibrary, 43
 - what, 44
- cannotCloseModule, 44
 - what, 44
- cannotCreateModule, 45
 - what, 45
- cannotOpenLibrary, 45
 - what, 46
- Centipede
 - arcade, 13
- changeDirection
 - arcade::Nibbler, 56
 - arcade::Snake, 66
- changePos
 - arcade, 13
- changes
 - arcade, 11
- Character
 - arcade, 11
- chargeAllLibs
 - arcade, 14
- checkLibChange
 - arcade, 14
- clear
 - arcade::ADisplayNcurses, 26
 - arcade::ADisplaySdl, 31
 - arcade::ADisplaySfml, 38
 - arcade::IDisplayModule, 50
- close
 - arcade::ADisplaySdl, 32
- closeAllLibs
 - arcade, 15
- Coin
 - arcade, 11, 13
- Core
 - arcade::Core, 46
- core
 - arcade, 15
- core.cpp
 - help, 74
 - main, 74
- createDisplayModule
 - ADisplayNcurses.cpp, 81
 - ADisplaySdl.cpp, 82
 - ADisplaySfml.cpp, 83
 - arcade, 16
 - IDisplayModule.hpp, 80
- createGameModule
 - arcade, 16
 - IGameModule.hpp, 80
 - nibbler.cpp, 84
 - snake.cpp, 85
- deathCondition
 - arcade::Nibbler, 56
 - arcade::Snake, 66
- deleteAllModules
 - arcade, 17
- displayBlock
 - arcade::ADisplayNcurses, 26
 - arcade::ADisplaySdl, 32
 - arcade::ADisplaySfml, 39
 - arcade::IDisplayModule, 51
- displayLibs
 - arcade, 11
- displayMenu
 - arcade, 17
- displayText
 - arcade::ADisplayNcurses, 27

- arcade::ADisplaySdl, [32](#)
- arcade::ADisplaySfml, [39](#)
- arcade::IDisplayModule, [51](#)
- Down
 - arcade, [11](#), [12](#)
- end
 - arcade::Nibbler, [56](#)
 - arcade::Snake, [66](#)
- Enemy
 - arcade, [11](#)
- EnemyDead
 - arcade, [11](#)
- EnemyVulnerable
 - arcade, [11](#)
- EventType
 - arcade, [12](#)
- fillDisplays
 - arcade, [18](#)
- fillGames
 - arcade, [18](#)
- findDisplay
 - arcade, [18](#)
- findDisplayLib
 - arcade, [19](#)
- findGame
 - arcade, [19](#)
- Fruit
 - arcade, [11](#)
- gameLibs
 - arcade, [12](#)
- gameLoop
 - arcade, [20](#)
- GameOver
 - arcade, [13](#)
- gamePause
 - arcade::Nibbler, [57](#)
 - arcade::Snake, [66](#)
- getBlockSize
 - arcade::ADisplayNcurses, [27](#)
 - arcade::ADisplaySdl, [33](#)
 - arcade::ADisplaySfml, [40](#)
- getDisplayLibName
 - arcade, [20](#)
- getGameLibName
 - arcade, [21](#)
- getName
 - arcade::Core, [47](#)
 - arcade::ICore, [49](#)
 - arcade::IGameModule, [53](#)
 - arcade::Nibbler, [57](#)
 - arcade::Snake, [66](#)
- getWindowSize
 - arcade::ADisplayNcurses, [27](#)
 - arcade::ADisplaySdl, [33](#)
 - arcade::ADisplaySfml, [40](#)
 - arcade::IDisplayModule, [51](#)
- Grass
 - arcade, [11](#)
- Gtk
 - arcade, [12](#)
- headOnKillBloc
 - snake.cpp, [85](#)
- headSnakeOnSnake
 - arcade::Nibbler, [57](#)
- headSnakeOnWall
 - arcade::Nibbler, [57](#)
- help
 - core.cpp, [74](#)
- Hurt
 - arcade, [13](#)
- IDisplayModule.hpp
 - createDisplayModule, [80](#)
- IGameModule.hpp
 - createGameModule, [80](#)
- Irrlicht
 - arcade, [12](#)
- isFruitOnSnake
 - arcade::Nibbler, [57](#)
 - snake.cpp, [85](#)
- Left
 - arcade, [11](#), [12](#)
- libChange
 - arcade, [13](#)
- main
 - core.cpp, [74](#)
- Menu
 - arcade, [21](#)
- moveCorps
 - arcade::Nibbler, [57](#)
 - arcade::Snake, [67](#)
- moveSnake
 - arcade::Nibbler, [58](#)
 - arcade::Snake, [67](#)
- Ncurses
 - arcade, [12](#)
- Ndk
 - arcade, [12](#)
- nextDisplay
 - arcade, [22](#)
- NextGame
 - arcade, [13](#)
- nextGame
 - arcade, [22](#)
- nextIndex
 - arcade, [22](#)
- NextLibrary
 - arcade, [12](#), [13](#)
- Nibbler
 - arcade, [13](#)
 - arcade::Nibbler, [55](#)

- nibbler.cpp
 - addVector, [83](#)
 - createGameModule, [84](#)
- None
 - arcade, [11](#), [13](#)
- OpenGL
 - arcade, [12](#)
- openLib
 - arcade, [23](#)
- Pacman
 - arcade, [13](#)
- Pause
 - arcade, [12](#)
- placeFruits
 - arcade::Nibbler, [58](#)
- playSound
 - arcade::ADisplayNcurses, [27](#)
 - arcade::ADisplaySdl, [33](#)
 - arcade::ADisplaySfml, [40](#)
 - arcade::IDisplayModule, [51](#)
- pollEvent
 - arcade::ADisplayNcurses, [28](#)
 - arcade::ADisplaySdl, [34](#)
 - arcade::ADisplaySfml, [40](#)
 - arcade::IDisplayModule, [51](#)
- PreviousGame
 - arcade, [13](#)
- previousIndex
 - arcade, [23](#)
- PreviousLibrary
 - arcade, [12](#), [13](#)
- printFruit
 - arcade::Nibbler, [58](#)
 - arcade::Snake, [67](#)
- printHighScore
 - arcade::Nibbler, [58](#)
 - arcade::Snake, [67](#)
- printMap
 - arcade::Nibbler, [58](#)
 - arcade::Snake, [68](#)
- printScoreAndName
 - arcade::Nibbler, [59](#)
 - arcade::Snake, [68](#)
- printSnake
 - arcade::Nibbler, [59](#)
 - arcade::Snake, [68](#)
- Qix
 - arcade, [13](#)
- Qt5
 - arcade, [12](#)
- Quit
 - arcade, [12](#), [13](#)
- refresh
 - arcade::ADisplayNcurses, [28](#)
 - arcade::ADisplaySdl, [34](#)
 - arcade::ADisplaySfml, [41](#)
 - arcade::IDisplayModule, [52](#)
- Restart
 - arcade, [13](#)
- RestartGame
 - arcade, [12](#)
- ReturnToSelectionMenu
 - arcade, [12](#), [13](#)
- Right
 - arcade, [11](#), [12](#)
- SDL2
 - arcade, [12](#)
- setAllFruits
 - arcade::Nibbler, [59](#)
- setMap
 - arcade::Nibbler, [59](#)
- setName
 - arcade::Core, [47](#)
 - arcade::ICore, [49](#)
- setNewFruit
 - arcade::Nibbler, [59](#)
 - arcade::Snake, [68](#)
- setWindowSize
 - arcade::ADisplayNcurses, [28](#)
 - arcade::ADisplaySdl, [34](#)
 - arcade::ADisplaySfml, [41](#)
 - arcade::IDisplayModule, [52](#)
- setWindowTitle
 - arcade::ADisplayNcurses, [29](#)
 - arcade::ADisplaySdl, [35](#)
 - arcade::ADisplaySfml, [41](#)
 - arcade::IDisplayModule, [52](#)
- SFML
 - arcade, [12](#)
- Snake
 - arcade, [13](#)
 - arcade::Snake, [65](#)
- snake.cpp
 - createGameModule, [85](#)
 - headOnKillBloc, [85](#)
 - isFruitOnSnake, [85](#)
- snakeEatFruit
 - arcade::Nibbler, [60](#)
 - arcade::Snake, [68](#)
- Solarfox
 - arcade, [13](#)
- Sound
 - arcade, [13](#)
- src/core/chargeLib.cpp, [73](#)
- src/core/closeLib.cpp, [73](#)
- src/core/core.cpp, [74](#)
- src/core/core.hpp, [75](#)
- src/core/findLib.cpp, [77](#)
- src/core/initLib.cpp, [77](#)
- src/core/menu.cpp, [77](#)
- src/errors/errors.hpp, [78](#)
- src/interfaces/Event.hpp, [78](#)
- src/interfaces/ICore.hpp, [79](#)

- src/interfaces/IDisplayModule.hpp, [79](#)
- src/interfaces/IGameModule.hpp, [80](#)
- src/Ldisplay/ncurses/ADisplayNcurses.cpp, [81](#)
- src/Ldisplay/ncurses/ADisplayNcurses.hpp, [81](#)
- src/Ldisplay/sdl/ADisplaySdl.cpp, [81](#)
- src/Ldisplay/sdl/ADisplaySdl.hpp, [82](#)
- src/Ldisplay/sfml/ADisplaySfml.cpp, [82](#)
- src/Ldisplay/sfml/ADisplaySfml.hpp, [83](#)
- src/Lgame/nibbler/nibbler.cpp, [83](#)
- src/Lgame/nibbler/nibbler.hpp, [84](#)
- src/Lgame/snake/snake.cpp, [84](#)
- src/Lgame/snake/snake.hpp, [85](#)
- Start
 - arcade, [13](#)
- start
 - arcade::IGameModule, [53](#)
 - arcade::Nibbler, [60](#)
 - arcade::Snake, [69](#)
- Super
 - arcade, [13](#)
- SwitchToNextGame
 - arcade, [12](#)
- SwitchToPreviousGame
 - arcade, [12](#)
- tick
 - arcade::IGameModule, [54](#)
 - arcade::Nibbler, [60](#)
 - arcade::Snake, [69](#)
- type
 - arcade::Event, [48](#)
- Up
 - arcade, [11](#), [12](#)
- verifDown
 - arcade::Nibbler, [60](#)
- verifLeft
 - arcade::Nibbler, [60](#)
- verifRight
 - arcade::Nibbler, [61](#)
- verifUp
 - arcade::Nibbler, [61](#)
- Vulkain
 - arcade, [12](#)
- what
 - cannotCloseLibrary, [44](#)
 - cannotCloseModule, [44](#)
 - cannotCreateModule, [45](#)
 - cannotOpenLibrary, [46](#)
- x
 - arcade::Position, [63](#)
- Xlib
 - arcade, [12](#)
- y
 - arcade::Position, [63](#)