

cpp_arcade

Generated by Doxygen 1.8.13

Contents

| | | |
|----------|--|-----------|
| 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 | Detailed Description | 10 |
| 5.1.2 | Enumeration Type Documentation | 10 |
| 5.1.2.1 | Keys | 10 |
| 6 | Class Documentation | 11 |
| 6.1 | Arcade::Color Class Reference | 11 |
| 6.1.1 | Detailed Description | 11 |
| 6.1.2 | Constructor & Destructor Documentation | 12 |
| 6.1.2.1 | Color() | 12 |
| 6.1.3 | Member Function Documentation | 12 |
| 6.1.3.1 | getAlpha() | 12 |
| 6.1.3.2 | getBlue() | 12 |

| | | |
|----------|--|----|
| 6.1.3.3 | getGreen() | 13 |
| 6.1.3.4 | getRed() | 13 |
| 6.1.3.5 | operator unsigned char *() | 13 |
| 6.1.3.6 | operator==() | 13 |
| 6.1.3.7 | setAlpha() | 14 |
| 6.1.3.8 | setBlue() | 14 |
| 6.1.3.9 | setColor() | 14 |
| 6.1.3.10 | setGreen() | 14 |
| 6.1.3.11 | setRed() | 15 |
| 6.2 | Arcade::Core Class Reference | 15 |
| 6.3 | DLLoader< T > Class Template Reference | 15 |
| 6.4 | Arcade::IGameLib Class Reference | 16 |
| 6.4.1 | Detailed Description | 16 |
| 6.4.2 | Constructor & Destructor Documentation | 16 |
| 6.4.2.1 | ~IGameLib() | 17 |
| 6.4.3 | Member Function Documentation | 17 |
| 6.4.3.1 | applyEvent() | 17 |
| 6.4.3.2 | getName() | 17 |
| 6.4.3.3 | getScore() | 18 |
| 6.4.3.4 | init() | 18 |
| 6.4.3.5 | refresh() | 18 |
| 6.4.3.6 | stop() | 18 |
| 6.4.3.7 | update() | 19 |
| 6.5 | Arcade::IGraphicLib Class Reference | 19 |
| 6.5.1 | Detailed Description | 20 |
| 6.5.2 | Constructor & Destructor Documentation | 20 |
| 6.5.2.1 | ~IGraphicLib() | 20 |
| 6.5.3 | Member Function Documentation | 21 |
| 6.5.3.1 | clearEvents() | 21 |
| 6.5.3.2 | closeRenderer() | 21 |

| | | |
|----------|---|----|
| 6.5.3.3 | getLastEvent() | 21 |
| 6.5.3.4 | getMaxX() | 22 |
| 6.5.3.5 | getMaxY() | 22 |
| 6.5.3.6 | getName() | 22 |
| 6.5.3.7 | getScreenSize() | 22 |
| 6.5.3.8 | isOpen() | 23 |
| 6.5.3.9 | openRenderer() | 23 |
| 6.5.3.10 | pollEvents() | 23 |
| 6.6 | Arcade::Menu Class Reference | 24 |
| 6.7 | Arcade::NCurses Class Reference | 24 |
| 6.7.1 | Member Function Documentation | 25 |
| 6.7.1.1 | clearEvents() | 25 |
| 6.7.1.2 | closeRenderer() | 26 |
| 6.7.1.3 | getLastEvent() | 26 |
| 6.7.1.4 | getMaxX() | 26 |
| 6.7.1.5 | getMaxY() | 26 |
| 6.7.1.6 | getName() | 27 |
| 6.7.1.7 | getScreenSize() | 27 |
| 6.7.1.8 | isOpen() | 27 |
| 6.7.1.9 | openRenderer() | 27 |
| 6.7.1.10 | pollEvents() | 28 |
| 6.8 | Arcade::Nibbler Class Reference | 28 |
| 6.8.1 | Member Function Documentation | 29 |
| 6.8.1.1 | applyEvent() | 29 |
| 6.8.1.2 | getName() | 30 |
| 6.8.1.3 | getScore() | 30 |
| 6.8.1.4 | init() | 30 |
| 6.8.1.5 | refresh() | 30 |
| 6.8.1.6 | stop() | 31 |
| 6.8.1.7 | update() | 31 |

| | | |
|-----------|--|----|
| 6.9 | Arcade::Pacman Class Reference | 32 |
| 6.9.1 | Member Function Documentation | 33 |
| 6.9.1.1 | applyEvent() | 33 |
| 6.9.1.2 | getName() | 33 |
| 6.9.1.3 | getScore() | 33 |
| 6.9.1.4 | init() | 34 |
| 6.9.1.5 | refresh() | 34 |
| 6.9.1.6 | stop() | 34 |
| 6.9.1.7 | update() | 35 |
| 6.10 | Arcade::PixelBox Class Reference | 35 |
| 6.10.1 | Detailed Description | 36 |
| 6.10.2 | Constructor & Destructor Documentation | 36 |
| 6.10.2.1 | PixelBox() | 36 |
| 6.10.3 | Member Function Documentation | 37 |
| 6.10.3.1 | getHeight() | 37 |
| 6.10.3.2 | getPixel() | 37 |
| 6.10.3.3 | getPixelArray() | 37 |
| 6.10.3.4 | getPos() | 38 |
| 6.10.3.5 | getSize() | 38 |
| 6.10.3.6 | getWidth() | 38 |
| 6.10.3.7 | getX() | 38 |
| 6.10.3.8 | getY() | 39 |
| 6.10.3.9 | putPixel() | 39 |
| 6.10.3.10 | putRect() | 39 |
| 6.10.3.11 | setPos() | 39 |
| 6.10.3.12 | setSize() | 41 |
| 6.11 | Arcade::Score Class Reference | 41 |
| 6.12 | Arcade::Sdl2 Class Reference | 42 |
| 6.12.1 | Member Function Documentation | 43 |
| 6.12.1.1 | clearEvents() | 43 |

| | | |
|-----------|--|----|
| 6.12.1.2 | closeRenderer() | 43 |
| 6.12.1.3 | getLastEvent() | 44 |
| 6.12.1.4 | getMaxX() | 44 |
| 6.12.1.5 | getMaxY() | 44 |
| 6.12.1.6 | getName() | 45 |
| 6.12.1.7 | getScreenSize() | 45 |
| 6.12.1.8 | isOpen() | 45 |
| 6.12.1.9 | openRenderer() | 45 |
| 6.12.1.10 | pollEvents() | 46 |
| 6.13 | Arcade::Sfml Class Reference | 46 |
| 6.13.1 | Member Function Documentation | 47 |
| 6.13.1.1 | clearEvents() | 48 |
| 6.13.1.2 | closeRenderer() | 48 |
| 6.13.1.3 | getLastEvent() | 48 |
| 6.13.1.4 | getMaxX() | 48 |
| 6.13.1.5 | getMaxY() | 49 |
| 6.13.1.6 | getName() | 49 |
| 6.13.1.7 | getScreenSize() | 49 |
| 6.13.1.8 | isOpen() | 49 |
| 6.13.1.9 | openRenderer() | 49 |
| 6.13.1.10 | pollEvents() | 50 |
| 6.14 | Arcade::TextBox Class Reference | 50 |
| 6.14.1 | Detailed Description | 51 |
| 6.14.2 | Constructor & Destructor Documentation | 51 |
| 6.14.2.1 | TextBox() | 52 |
| 6.14.3 | Member Function Documentation | 53 |
| 6.14.3.1 | getBackgroundColor() | 53 |
| 6.14.3.2 | getColor() | 53 |
| 6.14.3.3 | getFontSize() | 54 |
| 6.14.3.4 | getPos() | 54 |

| | | |
|-----------|--|----|
| 6.14.3.5 | getValue() | 54 |
| 6.14.3.6 | setBackgroundColor() | 54 |
| 6.14.3.7 | setColor() | 55 |
| 6.14.3.8 | setFontSize() | 55 |
| 6.14.3.9 | setPos() | 55 |
| 6.14.3.10 | setValue() | 55 |
| 6.15 | Arcade::Vect< T > Class Template Reference | 56 |
| 6.15.1 | Detailed Description | 57 |
| 6.15.2 | Constructor & Destructor Documentation | 57 |
| 6.15.2.1 | Vect() | 57 |
| 6.15.3 | Member Function Documentation | 57 |
| 6.15.3.1 | getX() | 57 |
| 6.15.3.2 | getY() | 58 |
| 6.15.3.3 | operator*() [1/2] | 58 |
| 6.15.3.4 | operator*() [2/2] | 58 |
| 6.15.3.5 | operator*=() [1/2] | 59 |
| 6.15.3.6 | operator*=() [2/2] | 59 |
| 6.15.3.7 | operator+() [1/2] | 59 |
| 6.15.3.8 | operator+() [2/2] | 60 |
| 6.15.3.9 | operator+=() [1/2] | 60 |
| 6.15.3.10 | operator+=() [2/2] | 60 |
| 6.15.3.11 | operator-() [1/2] | 61 |
| 6.15.3.12 | operator-() [2/2] | 61 |
| 6.15.3.13 | operator-=() [1/2] | 62 |
| 6.15.3.14 | operator-=() [2/2] | 62 |
| 6.15.3.15 | operator/() [1/2] | 62 |
| 6.15.3.16 | operator/() [2/2] | 63 |
| 6.15.3.17 | operator/=() [1/2] | 63 |
| 6.15.3.18 | operator/=() [2/2] | 63 |
| 6.15.3.19 | operator==() | 64 |
| 6.15.3.20 | setX() | 64 |
| 6.15.3.21 | setXY() | 64 |
| 6.15.3.22 | setY() | 65 |

| | |
|--|-----------|
| 7 File Documentation | 67 |
| 7.1 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/Color.hpp File Reference | 67 |
| 7.1.1 Detailed Description | 68 |
| 7.2 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/Keys.hpp File Reference | 68 |
| 7.2.1 Detailed Description | 69 |
| 7.3 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/PixelBox.hpp File Reference | 69 |
| 7.3.1 Detailed Description | 70 |
| 7.4 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/TextBox.hpp File Reference | 70 |
| 7.4.1 Detailed Description | 71 |
| 7.5 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/Vect.hpp File Reference | 72 |
| 7.5.1 Detailed Description | 72 |
| 7.6 /home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/IGameLib.hpp File Reference | 73 |
| 7.6.1 Detailed Description | 74 |
| 7.7 /home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/IGraphicLib.hpp File Reference | 74 |
| 7.7.1 Detailed Description | 75 |
| Index | 77 |

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

| | |
|--|---|
| Arcade | |
| Arcade project namespace | 9 |

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

| | |
|----------------------------------|----|
| Arcade::Color | 11 |
| Arcade::Core | 15 |
| DLLoader< T > | 15 |
| Arcade::IGameLib | 16 |
| Arcade::Nibbler | 28 |
| Arcade::Pacman | 32 |
| Arcade::IGraphicLib | 19 |
| Arcade::NCurses | 24 |
| Arcade::Sdl2 | 42 |
| Arcade::Sfml | 46 |
| Arcade::Menu | 24 |
| Arcade::PixelBox | 35 |
| Arcade::Score | 41 |
| Arcade::TextBox | 50 |
| Arcade::Vect< T > | 56 |
| Arcade::Vect< size_t > | 56 |

Chapter 3

Class Index

3.1 Class List

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

| | |
|---------------------------------|----|
| Arcade::Color | |
| Color class | 11 |
| Arcade::Core | 15 |
| DLLoader< T > | 15 |
| Arcade::IGameLib | |
| Game libraries virtual class | 16 |
| Arcade::IGraphicLib | |
| Graphic libraries virtual class | 19 |
| Arcade::Menu | 24 |
| Arcade::NCurses | 24 |
| Arcade::Nibbler | 28 |
| Arcade::Pacman | 32 |
| Arcade::PixelBox | |
| PixelBox class | 35 |
| Arcade::Score | 41 |
| Arcade::Sdl2 | 42 |
| Arcade::Sfml | 46 |
| Arcade::TextBox | |
| TextBox class | 50 |
| Arcade::Vect< T > | |
| Vect class template | 56 |

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

| | |
|---|----|
| /home/rectoria/projects/epitech/CPP/cpp_arcade/games/nibbler/ Nibbler.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/games/pacman/ Pacman.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/include/ Color.hpp Color class, pixel-like | 67 |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/include/ Keys.hpp Keys enum | 68 |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/include/ PixelBox.hpp PixelBox class, similar to a rectangle of pixels | 69 |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/include/ TextBox.hpp TextBox class, similar to a text rectangle | 70 |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/include/ Vect.hpp Project-specific vector template | 72 |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/ IGameLib.hpp Game libraries dedicated class interface | 73 |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/ IGraphicLib.hpp Graphic libraries dedicated class interface | 74 |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/lib/nCurses/ NCurses.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/lib/SDL2/ Sdl2.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/lib/SFML/ Sfml.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/src/core/ Core.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/src/menu/ Menu.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/src/score/ Score.hpp | ?? |
| /home/rectoria/projects/epitech/CPP/cpp_arcade/templates/ DLLoader.hpp | ?? |

Chapter 5

Namespace Documentation

5.1 Arcade Namespace Reference

[Arcade](#) project namespace.

Classes

- class [Color](#)
Color class.
- class [Core](#)
- class [IGameLib](#)
Game libraries virtual class.
- class [IGraphicLib](#)
Graphic libraries virtual class.
- class [Menu](#)
- class [NCurses](#)
- class [Nibbler](#)
- class [Pacman](#)
- class [PictureBox](#)
PictureBox class.
- class [Score](#)
- class [Sdl2](#)
- class [Sfml](#)
- class [TextBox](#)
TextBox class.
- class [Vect](#)
Vect class template.

Enumerations

- enum [Keys](#) {
 NONE, A, B, C,
 D, E, F, G,
 H, I, J, K,
 L, M, N, O,
 P, Q, R, S,
 T, U, V, W,
 X, Y, Z, LEFT,
 RIGHT, UP, DOWN, ENTER,
 SPACE, DELETE, BACKSPACE, TAB,
 ESC, MOUSELEFT, MUSERIGHT }

5.1.1 Detailed Description

[Arcade](#) project namespace.

5.1.2 Enumeration Type Documentation

5.1.2.1 Keys

enum [Arcade::Keys](#)

All those keys should be handled by any graphic libraries or any games

Chapter 6

Class Documentation

6.1 Arcade::Color Class Reference

[Color](#) class.

```
#include <Color.hpp>
```

Public Member Functions

- [Color](#) (unsigned char red=0, unsigned char green=0, unsigned char blue=0, unsigned char alpha=0)
Color class's constructor.
- void [setColor](#) (unsigned char red=0, unsigned char green=0, unsigned char blue=0, unsigned char alpha=0)
Sets the color's subpixels value.
- unsigned char [getRed](#) () const
Red subpixel's getter.
- unsigned char [getGreen](#) () const
Green subpixel's getter.
- unsigned char [getBlue](#) () const
Blue subpixel's getter.
- unsigned char [getAlpha](#) () const
Alpha subpixel's getter.
- void [setRed](#) (unsigned char red)
Red subpixel's setter.
- void [setGreen](#) (unsigned char green)
Green subpixel's setter.
- void [setBlue](#) (unsigned char blue)
Blue subpixel's setter.
- void [setAlpha](#) (unsigned char alpha)
Alpha subpixel's setter.
- [operator unsigned char *](#) ()
- bool [operator==](#) (const [Arcade::Color](#) &other) const

6.1.1 Detailed Description

[Color](#) class.

Class used to represent a pixel

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Color()

```
Arcade::Color::Color (
    unsigned char red = 0,
    unsigned char green = 0,
    unsigned char blue = 0,
    unsigned char alpha = 0 ) [explicit]
```

[Color](#) class's constructor.

Parameters

| | |
|--------------|---|
| <i>red</i> | |
| <i>green</i> | |
| <i>blue</i> | |
| <i>alpha</i> | Creates a new color class instance, each argument being a value between 0-255, representing the value of one of the subpixels (red, green, blue and alpha). |

6.1.3 Member Function Documentation

6.1.3.1 getAlpha()

```
unsigned char Arcade::Color::getAlpha ( ) const
```

Alpha subpixel's getter.

Returns

the alpha subpixel's value

6.1.3.2 getBlue()

```
unsigned char Arcade::Color::getBlue ( ) const
```

Blue subpixel's getter.

Returns

the blue subpixel's value

6.1.3.3 getGreen()

```
unsigned char Arcade::Color::getGreen ( ) const
```

Green subpixel's getter.

Returns

the green subpixel's value

6.1.3.4 getRed()

```
unsigned char Arcade::Color::getRed ( ) const
```

Red subpixel's getter.

Returns

the red subpixel's value

6.1.3.5 operator unsigned char *()

```
Arcade::Color::operator unsigned char * ( ) [explicit]
```

Overloading the cast operator to unsigned char *

Returns

an array of unsigned char * composed of 4 elements, each representing one of the subpixels

6.1.3.6 operator==()

```
bool Arcade::Color::operator== (
    const Arcade::Color & other ) const
```

Overloading the comparison operator

Parameters

| | |
|--------------|------------------------------------|
| <i>other</i> | : the color object to compare with |
|--------------|------------------------------------|

Returns

true if equal, otherwise returns false

6.1.3.7 setAlpha()

```
void Arcade::Color::setAlpha (
    unsigned char alpha )
```

Alpha subpixel's setter.

Sets the value of the alpha's subpixel

6.1.3.8 setBlue()

```
void Arcade::Color::setBlue (
    unsigned char blue )
```

Blue subpixel's setter.

Sets the value of the blue's subpixel

6.1.3.9 setColor()

```
void Arcade::Color::setColor (
    unsigned char red = 0,
    unsigned char green = 0,
    unsigned char blue = 0,
    unsigned char alpha = 0 )
```

Sets the color's subpixels value.

Parameters

| | |
|--------------|--|
| <i>red</i> | |
| <i>green</i> | |
| <i>blue</i> | |
| <i>alpha</i> | Sets the color's object subpixels value, each argument being a value between 0-255, representing the value of one of the subpixels (red, green, blue and alpha). |

6.1.3.10 setGreen()

```
void Arcade::Color::setGreen (
    unsigned char green )
```

Green subpixel's setter.

Sets the value of the green's subpixel

6.1.3.11 setRed()

```
void Arcade::Color::setRed (
    unsigned char red )
```

Red subpixel's setter.

Sets the value of the red's subpixel

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

- [/home/rectoria/projects/epitech/CPP/cpp_arcade/include/Color.hpp](#)
- [/home/rectoria/projects/epitech/CPP/cpp_arcade/include/Color.cpp](#)

6.2 Arcade::Core Class Reference

Public Member Functions

- **Core** (std::string)
- bool **start** ()
- size_t **getLibIdx** () const

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

- [/home/rectoria/projects/epitech/CPP/cpp_arcade/src/core/Core.hpp](#)
- [/home/rectoria/projects/epitech/CPP/cpp_arcade/src/core/Core.cpp](#)

6.3 DLLoader< T > Class Template Reference

Public Member Functions

- **DLLoader** (std::string &filepath, std::string &folder)
- T * **getInstance** ()
- std::string **getFilePath** () const

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

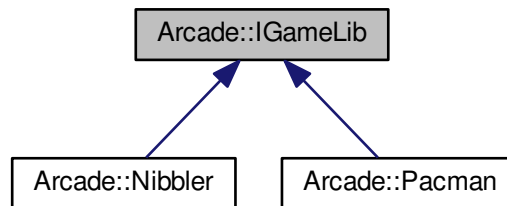
- [/home/rectoria/projects/epitech/CPP/cpp_arcade/templates/DLLoader.hpp](#)

6.4 Arcade::IGameLib Class Reference

Game libraries virtual class.

```
#include <IGameLib.hpp>
```

Inheritance diagram for Arcade::IGameLib:



Public Member Functions

- virtual `~IGameLib()`=default
Destructor.
- virtual const std::string `getName()` const =0
Game name's getter.
- virtual bool `init()`=0
Init the resources needed by the game to run.
- virtual bool `stop()`=0
Unloads the library.
- virtual bool `applyEvent(Arcade::Keys key)`=0
Processes the key obtained by the IGraphicLib from the user to update the game state.
- virtual bool `update()`=0
Updates the game state.
- virtual void `refresh(IGraphicLib &graphicLib)`=0
- virtual size_t `getScore()`=0
Current player score's getter.

6.4.1 Detailed Description

Game libraries virtual class.

Purely virtual class that serves as the basis for all game libraries

6.4.2 Constructor & Destructor Documentation

6.4.2.1 ~IGameLib()

```
virtual Arcade::IGameLib::~~IGameLib ( ) [virtual], [default]
```

Destructor.

[IGameLib](#) class's destructor

6.4.3 Member Function Documentation

6.4.3.1 applyEvent()

```
virtual bool Arcade::IGameLib::applyEvent (
    Arcade::Keys key ) [pure virtual]
```

Processes the key obtained by the [IGraphicLib](#) from the user to update the game state.

Parameters

| | |
|------------|----------------------------------|
| <i>key</i> | : enum value of the obtained key |
|------------|----------------------------------|

Returns

true if the game is still in progress, false in case of defeat

Implemented in [Arcade::Nibbler](#), and [Arcade::Pacman](#).

6.4.3.2 getName()

```
virtual const std::string Arcade::IGameLib::getName ( ) const [pure virtual]
```

Game name's getter.

Returns

a string containing the name of the game

Implemented in [Arcade::Nibbler](#), and [Arcade::Pacman](#).

6.4.3.3 `getScore()`

```
virtual size_t Arcade::IGameLib::getScore ( ) [pure virtual]
```

Current player score's getter.

Returns

the player score

To call at the end of the execution of the game (after the player loose or win) for getting his score

Implemented in [Arcade::Nibbler](#), and [Arcade::Pacman](#).

6.4.3.4 `init()`

```
virtual bool Arcade::IGameLib::init ( ) [pure virtual]
```

Init the resources needed by the game to run.

Returns

true if succeed, otherwise returns false

Implemented in [Arcade::Nibbler](#), and [Arcade::Pacman](#).

6.4.3.5 `refresh()`

```
virtual void Arcade::IGameLib::refresh (
    IGraphicLib & graphicLib ) [pure virtual]
```

Renders the game state to the screen.

Parameters

| | |
|-------------------|---|
| <i>graphicLib</i> | : Loaded graphics library used for rendering. |
|-------------------|---|

This should call `IGraphicLib::refresh()` to display content to the user.

Implemented in [Arcade::Nibbler](#), and [Arcade::Pacman](#).

6.4.3.6 `stop()`

```
virtual bool Arcade::IGameLib::stop ( ) [pure virtual]
```

Unloads the library.

Returns

true if succeed, otherwise returns false

Implemented in [Arcade::Nibbler](#), and [Arcade::Pacman](#).

6.4.3.7 update()

```
virtual bool Arcade::IGameLib::update ( ) [pure virtual]
```

Updates the game state.

Returns

true if the game is still in progress, false in case of defeat

Move the player forward and/or move the NPCs, according to the game's rules

Implemented in [Arcade::Nibbler](#), and [Arcade::Pacman](#).

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

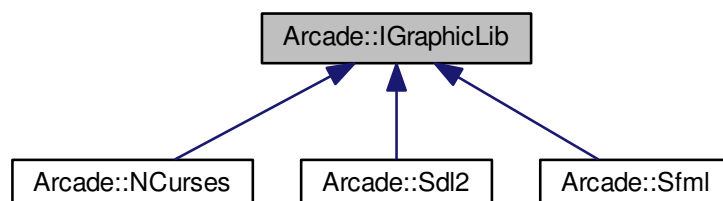
- [/home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/IGameLib.hpp](#)

6.5 Arcade::IGraphicLib Class Reference

Graphic libraries virtual class.

```
#include <IGraphicLib.hpp>
```

Inheritance diagram for Arcade::IGraphicLib:



Public Member Functions

- virtual [~IGraphicLib](#) ()=default
Destructor.
- virtual std::string [getName](#) () const =0
Graphic library name's getter.
- virtual bool [isOpen](#) () const =0
Specifies whether the window is open or not.
- virtual void [closeRenderer](#) ()=0
Closes the rendering support.
- virtual void [openRenderer](#) (std::string const &title)=0
Opens the rendering support.
- virtual void [clearWindow](#) ()=0
Clears the rendering support.
- virtual void [refreshWindow](#) ()=0
Displays the buffered frame to the screen.
- virtual void [drawPixelBox](#) ([PixelBox](#) const &)=0
Draws a [PixelBox](#).
- virtual void [drawText](#) ([TextBox](#) const &)=0
Draws a [TextBox](#).
- virtual bool [pollEvents](#) ()=0
Fetches the events from the user and saves it.
- virtual [Keys](#) [getLastEvent](#) ()=0
Getter of the oldest command in memory.
- virtual void [clearEvents](#) ()=0
Clears the pending commands.
- virtual [Vect](#)< size_t > [getScreenSize](#) () const =0
Getter from the rendering support dimensions.
- virtual size_t [getMaxY](#) () const =0
Getter from the rendering support height.
- virtual size_t [getMaxX](#) () const =0
Getter from the rendering support width.

6.5.1 Detailed Description

Graphic libraries virtual class.

Purely virtual class that serves as the basis for all graphic libraries

6.5.2 Constructor & Destructor Documentation

6.5.2.1 [~IGraphicLib](#)()

```
virtual Arcade::IGraphicLib::~IGraphicLib ( ) [virtual], [default]
```

Destructor.

[IGraphicLib](#) class's destructor

6.5.3 Member Function Documentation

6.5.3.1 clearEvents()

```
virtual void Arcade::IGraphicLib::clearEvents ( ) [pure virtual]
```

Clears the pending commands.

The function deletes all the commands currently stored. They wont be accessible anymore, even with the [getLastEvent\(\)](#) method.

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.2 closeRenderer()

```
virtual void Arcade::IGraphicLib::closeRenderer ( ) [pure virtual]
```

Closes the rendering support.

Usually closes a window. Some graphic library uses other rendering support.

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.3 getLastEvent()

```
virtual Keys Arcade::IGraphicLib::getLastEvent ( ) [pure virtual]
```

Getter of the oldest command in memory.

Returns

the first event of the list.

The function deletes the command if it succeed to retrieves one, using `front()` and `pop_front()` methods

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.4 getMaxX()

```
virtual size_t Arcade::IGraphicLib::getMaxX ( ) const [pure virtual]
```

Getter from the rendering support width.

Returns

the width of the rendering support

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.5 getMaxY()

```
virtual size_t Arcade::IGraphicLib::getMaxY ( ) const [pure virtual]
```

Getter from the rendering support height.

Returns

the height of the rendering support

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.6 getName()

```
virtual std::string Arcade::IGraphicLib::getName ( ) const [pure virtual]
```

Graphic library name's getter.

Returns

a string containing the name of the graphic library

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.7 getScreenSize()

```
virtual Vect<size_t> Arcade::IGraphicLib::getScreenSize ( ) const [pure virtual]
```

Getter from the rendering support dimensions.

Returns

a two dimensions vector containing the width and the height of the rendering support

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.8 isOpen()

```
virtual bool Arcade::IGraphicLib::isOpen ( ) const [pure virtual]
```

Specifies whether the window is open or not.

Returns

true if open, otherwise returns false

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.9 openRenderer()

```
virtual void Arcade::IGraphicLib::openRenderer (
    std::string const & title ) [pure virtual]
```

Opens the rendering support.

Parameters

| | |
|--------------|---|
| <i>title</i> | : Title of the rendering support if supported |
|--------------|---|

Usually opens a window. Some graphic library uses other rendering support.

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

6.5.3.10 pollEvents()

```
virtual bool Arcade::IGraphicLib::pollEvents ( ) [pure virtual]
```

Fetches the events from the user and saves it.

Returns

true if at least one command has been fetched, otherwise returns false

Fetches commands are usually stored inside a `std::vector<Arcade::Keys>` or `std::list<Arcade::Keys>`

Implemented in [Arcade::NCurses](#), [Arcade::Sfml](#), and [Arcade::Sdl2](#).

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

- `/home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/IGraphicLib.hpp`

6.6 Arcade::Menu Class Reference

Public Member Functions

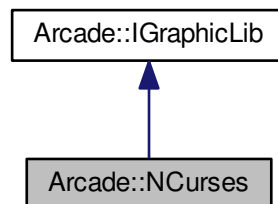
- void **setLists** (std::vector< [DLLoader](#)< [IGameLib](#) > *> *, std::vector< [DLLoader](#)< [IGraphicLib](#) > *> *)
- void **refresh** ([IGraphicLib](#) *, unsigned, [Arcade::Score](#))
- unsigned **applyEvent** ([Keys](#))
- std::string **getUserName** () const

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

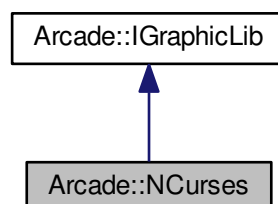
- /home/rectoria/projects/epitech/CPP/cpp_arcade/src/menu/Menu.hpp
- /home/rectoria/projects/epitech/CPP/cpp_arcade/src/menu/Menu.cpp

6.7 Arcade::NCurses Class Reference

Inheritance diagram for Arcade::NCurses:



Collaboration diagram for Arcade::NCurses:



Public Member Functions

- `std::string getName ()` const final
Graphic library name's getter.
- `bool isOpen ()` const final
Specifies whether the window is open or not.
- `void closeRenderer ()` final
Closes the rendering support.
- `void openRenderer (std::string const &title)` final
Opens the rendering support.
- `void clearWindow ()` final
Clears the rendering support.
- `void refreshWindow ()` final
Displays the buffered frame to the screen.
- `void drawPixelBox (PixelBox const &)` final
Draws a [PixelBox](#).
- `void drawText (TextBox const &)` final
Draws a [TextBox](#).
- `bool pollEvents ()` final
Fetches the events from the user and saves it.
- `Keys getLastEvent ()` final
Getter of the oldest command in memory.
- `void clearEvents ()` final
Clears the pending commands.
- `Vect< size_t > getScreenSize ()` const final
Getter from the rendering support dimensions.
- `size_t getMaxY ()` const final
Getter from the rendering support height.
- `size_t getMaxX ()` const final
Getter from the rendering support width.

6.7.1 Member Function Documentation

6.7.1.1 clearEvents()

```
void Arcade::NCurses::clearEvents ( ) [final], [virtual]
```

Clears the pending commands.

The function deletes all the commands currently stored. They wont be accessible anymore, even with the [getLastEvent\(\)](#) method.

Implements [Arcade::IGraphicLib](#).

6.7.1.2 closeRenderer()

```
void Arcade::NCurses::closeRenderer ( ) [final], [virtual]
```

Closes the rendering support.

Usually closes a window. Some graphic library uses other rendering support.

Implements [Arcade::IGraphicLib](#).

6.7.1.3 getLastEvent()

```
Arcade::Keys Arcade::NCurses::getLastEvent ( ) [final], [virtual]
```

Getter of the oldest command in memory.

Returns

the first event of the list.

The function deletes the command if it succeed to retrieves one, using `front()` and `pop_front()` methods

Implements [Arcade::IGraphicLib](#).

6.7.1.4 getMaxX()

```
size_t Arcade::NCurses::getMaxX ( ) const [final], [virtual]
```

Getter from the rendering support width.

Returns

the width of the rendering support

Implements [Arcade::IGraphicLib](#).

6.7.1.5 getMaxY()

```
size_t Arcade::NCurses::getMaxY ( ) const [final], [virtual]
```

Getter from the rendering support height.

Returns

the height of the rendering support

Implements [Arcade::IGraphicLib](#).

6.7.1.6 getName()

```
std::string Arcade::NCurses::getName ( ) const [final], [virtual]
```

Graphic library name's getter.

Returns

a string containing the name of the graphic library

Implements [Arcade::IGraphicLib](#).

6.7.1.7 getScreenSize()

```
Arcade::Vect< size_t > Arcade::NCurses::getScreenSize ( ) const [final], [virtual]
```

Getter from the rendering support dimensions.

Returns

a two dimensions vector containing the width and the height of the rendering support

Implements [Arcade::IGraphicLib](#).

6.7.1.8 isOpen()

```
bool Arcade::NCurses::isOpen ( ) const [final], [virtual]
```

Specifies whether the window is open or not.

Returns

true if open, otherwise returns false

Implements [Arcade::IGraphicLib](#).

6.7.1.9 openRenderer()

```
void Arcade::NCurses::openRenderer (
    std::string const & title ) [final], [virtual]
```

Opens the rendering support.

Parameters

| | |
|--------------|---|
| <i>title</i> | : Title of the rendering support if supported |
|--------------|---|

Usually opens a window. Some graphic library uses other rendering support.

Implements [Arcade::IGraphicLib](#).

6.7.1.10 pollEvents()

```
bool Arcade::NCurses::pollEvents ( ) [final], [virtual]
```

Fetches the events from the user and saves it.

Returns

true if at least one command has been fetched, otherwise returns false

Fetches commands are usually stored inside a `std::vector<Arcade::Keys>` or `std::list<Arcade::Keys>`

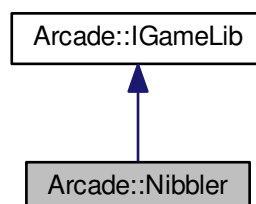
Implements [Arcade::IGraphicLib](#).

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

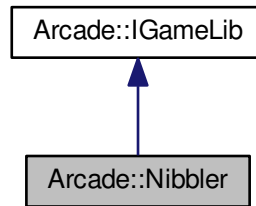
- /home/rectoria/projects/epitech/CPP/cpp_arcade/lib/nCurses/NCurses.hpp
- /home/rectoria/projects/epitech/CPP/cpp_arcade/lib/nCurses/NCurses.cpp

6.8 Arcade::Nibbler Class Reference

Inheritance diagram for Arcade::Nibbler:



Collaboration diagram for Arcade::Nibbler:



Public Member Functions

- `const std::string getName () const final`
Game name's getter.
- `bool init () final`
Init the resources needed by the game to run.
- `bool stop () final`
Unloads the library.
- `bool applyEvent (Keys key) final`
Processes the key obtained by the [IGraphicLib](#) from the user to update the game state.
- `bool update () final`
Updates the game state.
- `void refresh (IGraphicLib &graphicLib) final`
- `size_t getScore () final`
Current player score's getter.

6.8.1 Member Function Documentation

6.8.1.1 applyEvent()

```
bool Arcade::Nibbler::applyEvent (
    Keys key ) [final], [virtual]
```

Processes the key obtained by the [IGraphicLib](#) from the user to update the game state.

Parameters

| | |
|------------|----------------------------------|
| <i>key</i> | : enum value of the obtained key |
|------------|----------------------------------|

Returns

true if the game is still in progress, false in case of defeat

Implements [Arcade::IGameLib](#).

6.8.1.2 getName()

```
const std::string Arcade::Nibbler::getName ( ) const [final], [virtual]
```

Game name's getter.

Returns

a string containing the name of the game

Implements [Arcade::IGameLib](#).

6.8.1.3 getScore()

```
size_t Arcade::Nibbler::getScore ( ) [final], [virtual]
```

Current player score's getter.

Returns

the player score

To call at the end of the execution of the game (after the player loose or win) for getting his score

Implements [Arcade::IGameLib](#).

6.8.1.4 init()

```
bool Arcade::Nibbler::init ( ) [final], [virtual]
```

Init the resources needed by the game to run.

Returns

true if succeed, otherwise returns false

Implements [Arcade::IGameLib](#).

6.8.1.5 refresh()

```
void Arcade::Nibbler::refresh (
    IGraphicLib & graphicLib ) [final], [virtual]
```

Renders the game state to the screen.

Parameters

| | |
|-------------------|---|
| <i>graphicLib</i> | : Loaded graphics library used for rendering. |
|-------------------|---|

This should call `IGraphicLib::refresh()` to display content to the user.

Implements [Arcade::IGameLib](#).

6.8.1.6 stop()

```
bool Arcade::Nibbler::stop ( ) [final], [virtual]
```

Unloads the library.

Returns

true if succeed, otherwise returns false

Implements [Arcade::IGameLib](#).

6.8.1.7 update()

```
bool Arcade::Nibbler::update ( ) [final], [virtual]
```

Updates the game state.

Returns

true if the game is still in progress, false in case of defeat

Move the player forward and/or move the NPCs, according to the game's rules

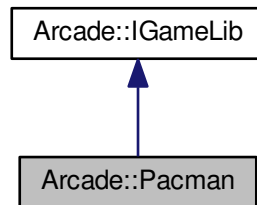
Implements [Arcade::IGameLib](#).

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

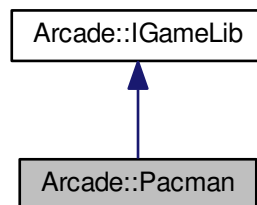
- `/home/rectoria/projects/epitech/CPP/cpp_arcade/games/nibbler/Nibbler.hpp`
- `/home/rectoria/projects/epitech/CPP/cpp_arcade/games/nibbler/Nibbler.cpp`

6.9 Arcade::Pacman Class Reference

Inheritance diagram for Arcade::Pacman:



Collaboration diagram for Arcade::Pacman:



Public Member Functions

- `const std::string getName () const final`
Game name's getter.
- `bool init () final`
Init the resources needed by the game to run.
- `bool stop () final`
Unloads the library.
- `bool applyEvent (Arcade::Keys) final`
Processes the key obtained by the [IGraphicLib](#) from the user to update the game state.
- `bool update () final`
Updates the game state.
- `void refresh (IGraphicLib &) final`
- `size_t getScore () final`
Current player score's getter.

6.9.1 Member Function Documentation

6.9.1.1 applyEvent()

```
bool Arcade::Pacman::applyEvent (
    Arcade::Keys key ) [final], [virtual]
```

Processes the key obtained by the [IGraphicLib](#) from the user to update the game state.

Parameters

| | |
|------------|----------------------------------|
| <i>key</i> | : enum value of the obtained key |
|------------|----------------------------------|

Returns

true if the game is still in progress, false in case of defeat

Implements [Arcade::IGameLib](#).

6.9.1.2 getName()

```
const std::string Arcade::Pacman::getName ( ) const [final], [virtual]
```

Game name's getter.

Returns

a string containing the name of the game

Implements [Arcade::IGameLib](#).

6.9.1.3 getScore()

```
size_t Arcade::Pacman::getScore ( ) [final], [virtual]
```

Current player score's getter.

Returns

the player score

To call at the end of the execution of the game (after the player loose or win) for getting his score

Implements [Arcade::IGameLib](#).

6.9.1.4 init()

```
bool Arcade::Pacman::init ( ) [final], [virtual]
```

Init the resources needed by the game to run.

Returns

true if succeed, otherwise returns false

Implements [Arcade::IGameLib](#).

6.9.1.5 refresh()

```
void Arcade::Pacman::refresh (
    IGraphicLib & graphicLib ) [final], [virtual]
```

Renders the game state to the screen.

Parameters

| | |
|-------------------|---|
| <i>graphicLib</i> | : Loaded graphics library used for rendering. |
|-------------------|---|

This should call IGraphicLib::refresh() to display content to the user.

Implements [Arcade::IGameLib](#).

6.9.1.6 stop()

```
bool Arcade::Pacman::stop ( ) [final], [virtual]
```

Unloads the library.

Returns

true if succeed, otherwise returns false

Implements [Arcade::IGameLib](#).

6.9.1.7 update()

```
bool Arcade::Pacman::update ( ) [final], [virtual]
```

Updates the game state.

Returns

true if the game is still in progress, false in case of defeat

Move the player forward and/or move the NPCs, according to the game's rules

Implements [Arcade::IGameLib](#).

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

- /home/rectoria/projects/epitech/CPP/cpp_arcade/games/pacman/Pacman.hpp
- /home/rectoria/projects/epitech/CPP/cpp_arcade/games/pacman/Pacman.cpp

6.10 Arcade::PixelBox Class Reference

[PixelBox](#) class.

```
#include <PixelBox.hpp>
```

Public Member Functions

- [PixelBox](#) ([Vect](#)< size_t > size=[Vect](#)< size_t >(), [Vect](#)< size_t > pos=[Vect](#)< size_t >(), [Color](#) col=[Color](#)(255, 255, 255))
[PixelBox](#) class's constructor.
- [~PixelBox](#) ()=default
[PixelBox](#) class's destructor.
- size_t [getHeight](#) () const
[PixelBox](#) height's getter.
- size_t [getY](#) () const
[PixelBox](#) Y offset's getter.
- void [setHeight](#) (size_t height)
[PixelBox](#) height setter.
- void [setY](#) (size_t y)
[PixelBox](#) Y offset's getter.
- size_t [getWidth](#) () const
[PixelBox](#) width's getter.
- size_t [getX](#) () const
[PixelBox](#) X offset's getter.
- void [setWidth](#) (size_t width)
[PixelBox](#) height's setter.
- void [setX](#) (size_t x)
[PixelBox](#) X offset's setter.
- [Vect](#)< size_t > [getSize](#) () const

- *PixelFormat dimensions's getter.*
void `setSize` (`Vect`< `size_t` > `size`)
- *PixelFormat dimensions's getter.*
`Vect`< `size_t` > `getPos` () const
- *PixelFormat positions's getter.*
void `setPos` (`Vect`< `size_t` > `pos`)
- *PixelFormat positions's setter.*
void `putPixel` (`Vect`< `size_t` > `pos`, `Color` `col`)
Sets the color of the pixel at the given position.
- `Color` `getPixel` (`Vect`< `size_t` > `pos`) const
Getter from pixel color to given position.
- void `putRect` (`Vect`< `size_t` > `pos`, `Vect`< `size_t` > `size`, `Color` `col`)
Sets the color of many pixels within the pixelBox pixels's array.
- `std::vector`< `Color` > const & `getPixelArray` () const
Getter of the pixels array.

6.10.1 Detailed Description

`PixelFormat` class.

Class used to represent a rectangle of pixels

6.10.2 Constructor & Destructor Documentation

6.10.2.1 PixelBox()

```
Arcade::PixelFormat::PixelFormat (
    Arcade::Vect< size_t > size = Vect<size_t>(),
    Arcade::Vect< size_t > pos = Vect<size_t>(),
    Arcade::Color col = Color(255, 255, 255, 255) ) [explicit]
```

`PixelFormat` class's constructor.

Parameters

| | |
|-------------|--|
| <i>size</i> | : <code>Vect</code> < <code>size_t</code> > containing the width (x) and the height (y) of the pixelBox |
| <i>pos</i> | : <code>Vect</code> < <code>size_t</code> > containing both x and y offsets. Used to place the pixelBox on the rendering support |
| <i>col</i> | : the color with which the array of pixels inside the pixelBox will be created |

Creates a new pixelBox class instance. The first `Vect`<`size_t`> size argument defines the dimensions of the pixelBox. The second `Vect`<`size_t`> pos argument defines the coordinates of the pixelBox's position on the rendering support. It will be the offset applied when rendering it. The third argument defines the color in which the pixels will be created.

6.10.3 Member Function Documentation

6.10.3.1 getHeight()

```
size_t Arcade::PixelFormat::getHeight ( ) const
```

PixelFormat height's getter.

Returns

the pixelBox's height

6.10.3.2 getPixel()

```
Arcade::Color Arcade::PixelFormat::getPixel (
    Arcade::Vect< size_t > pos ) const
```

Getter from pixel color to given position.

Parameters

| | |
|------------|---|
| <i>pos</i> | : The position of the pixel from which the color is requested |
|------------|---|

Returns

the color of the requested pixel

6.10.3.3 getPixelFormatArray()

```
std::vector< Arcade::Color > const & Arcade::PixelFormat::getPixelFormatArray ( ) const
```

Getter of the pixels array.

Returns

a vector of all the pixels of the pixelBox.

6.10.3.4 getPos()

```
Arcade::Vect< size_t > Arcade::PixelBox::getPos ( ) const
```

[PixelBox](#) positions's getter.

Returns

a [Vect<size_t>](#) containing the offsetX (x) and the offsetY (y) of the pixelBox.

6.10.3.5 getSize()

```
Arcade::Vect< size_t > Arcade::PixelBox::getSize ( ) const
```

[PixelBox](#) dimensions's getter.

Returns

a [Vect<size_t>](#) containing the width (x) and the height (y) of the pixelBox.

6.10.3.6 getWidth()

```
size_t Arcade::PixelBox::getWidth ( ) const
```

[PixelBox](#) width's getter.

Returns

the pixelBox's height

6.10.3.7 getX()

```
size_t Arcade::PixelBox::getX ( ) const
```

[PixelBox](#) X offset's getter.

Returns

the pixelBox X's offset

6.10.3.8 getY()

```
size_t Arcade::PixelFormat::getY ( ) const
```

PixelFormat Y offset's getter.

Returns

the pixelBox Y's offset

6.10.3.9 putPixel()

```
void Arcade::PixelFormat::putPixel (
    Arcade::Vect< size_t > pos,
    Arcade::Color col )
```

Sets the color of the pixel at the given position.

Parameters

| | |
|------------|---|
| <i>pos</i> | : The position of the pixel to be modified |
| <i>col</i> | : The new color of the pixel to be modified |

6.10.3.10 putRect()

```
void Arcade::PixelFormat::putRect (
    Arcade::Vect< size_t > pos,
    Arcade::Vect< size_t > size,
    Arcade::Color col )
```

Sets the color of many pixels within the pixelBox pixels's array.

Parameters

| | |
|-------------|---|
| <i>pos</i> | : The position from which the new color has to be applied |
| <i>size</i> | : The dimensions of the chunk of pixels to be modified |
| <i>col</i> | : The new color to apply |

6.10.3.11 setPos()

```
void Arcade::PixelFormat::setPos (
    Arcade::Vect< size_t > pos )
```

[PixelFormat](#) positions's setter.

Parameters

| | |
|------------|--|
| <i>pos</i> | : new positions of the pixelBox pixels's array |
|------------|--|

Takes both new positions as parameter, within a [Vect<size_t>](#)

6.10.3.12 setSize()

```
void Arcade::PixelBox::setSize (
    Arcade::Vect< size_t > size )
```

[PixelBox](#) dimensions's getter.

Parameters

| | |
|-------------|---|
| <i>size</i> | : new dimensions of the pixelBox pixels's array |
|-------------|---|

Takes both new dimensions as parameter, within a [Vect<size_t>](#)

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

- [/home/rectoria/projects/epitech/CPP/cpp_arcade/include/PixelBox.hpp](#)
- [/home/rectoria/projects/epitech/CPP/cpp_arcade/include/PixelBox.cpp](#)

6.11 Arcade::Score Class Reference

Public Member Functions

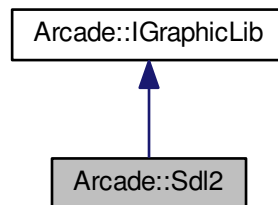
- void **addGame** (std::string &)
- std::map< std::string, std::string > **getGameStats** (std::string &)
- int **getPlayerScore** (std::string &, std::string &)
- void **writeStats** (std::string)
- void **setScore** (std::string &, std::string &, unsigned)

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

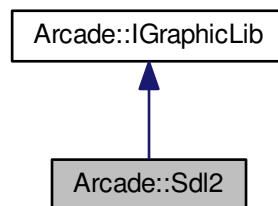
- [/home/rectoria/projects/epitech/CPP/cpp_arcade/src/score/Score.hpp](#)
- [/home/rectoria/projects/epitech/CPP/cpp_arcade/src/score/Score.cpp](#)

6.12 Arcade::Sdl2 Class Reference

Inheritance diagram for Arcade::Sdl2:



Collaboration diagram for Arcade::Sdl2:



Public Member Functions

- `std::string getName ()` const final
Graphic library name's getter.
- `bool isOpen ()` const final
Specifies whether the window is open or not.
- `void closeRenderer ()` final
Closes the rendering support.
- `void openRenderer (std::string const &title)` final
Opens the rendering support.
- `void clearWindow ()` final
Clears the rendering support.
- `void refreshWindow ()` final
Displays the buffered frame to the screen.
- `void drawPixelBox (PixelBox const &)` final
*Draws a *PixelBox*.*
- `void drawText (TextBox const &)` final

- Draws a [TextBox](#).*
- `bool pollEvents ()` final

Fetches the events from the user and saves it.
 - `Keys getLastEvent ()` final

Getter of the oldest command in memory.
 - `void clearEvents ()` final

Clears the pending commands.
 - `Vect< size_t > getScreenSize ()` const final

Getter from the rendering support dimensions.
 - `size_t getMaxY ()` const final

Getter from the rendering support height.
 - `size_t getMaxX ()` const final

Getter from the rendering support width.

6.12.1 Member Function Documentation

6.12.1.1 `clearEvents()`

```
void Arcade::Sdl2::clearEvents ( ) [final], [virtual]
```

Clears the pending commands.

The function deletes all the commands currently stored. They wont be accessible anymore, even with the [getLastEvent\(\)](#) method.

Implements [Arcade::IGraphicLib](#).

6.12.1.2 `closeRenderer()`

```
void Arcade::Sdl2::closeRenderer ( ) [final], [virtual]
```

Closes the rendering support.

Usually closes a window. Some graphic library uses other rendering support.

Implements [Arcade::IGraphicLib](#).

6.12.1.3 getLastEvent()

```
Arcade::Keys Arcade::Sdl2::getLastEvent ( ) [final], [virtual]
```

Getter of the oldest command in memory.

Returns

the first event of the list.

The function deletes the command if it succeed to retrieves one, using front() and pop_front() methods

Implements [Arcade::IGraphicLib](#).

6.12.1.4 getMaxX()

```
size_t Arcade::Sdl2::getMaxX ( ) const [final], [virtual]
```

Getter from the rendering support width.

Returns

the width of the rendering support

Implements [Arcade::IGraphicLib](#).

6.12.1.5 getMaxY()

```
size_t Arcade::Sdl2::getMaxY ( ) const [final], [virtual]
```

Getter from the rendering support height.

Returns

the height of the rendering support

Implements [Arcade::IGraphicLib](#).

6.12.1.6 getName()

```
std::string Arcade::Sdl2::getName ( ) const [final], [virtual]
```

Graphic library name's getter.

Returns

a string containing the name of the graphic library

Implements [Arcade::IGraphicLib](#).

6.12.1.7 getScreenSize()

```
Arcade::Vect< size_t > Arcade::Sdl2::getScreenSize ( ) const [final], [virtual]
```

Getter from the rendering support dimensions.

Returns

a two dimensions vector containing the width and the height of the rendering support

Implements [Arcade::IGraphicLib](#).

6.12.1.8 isOpen()

```
bool Arcade::Sdl2::isOpen ( ) const [final], [virtual]
```

Specifies whether the window is open or not.

Returns

true if open, otherwise returns false

Implements [Arcade::IGraphicLib](#).

6.12.1.9 openRenderer()

```
void Arcade::Sdl2::openRenderer (
    std::string const & title ) [final], [virtual]
```

Opens the rendering support.

Parameters

| | |
|--------------|---|
| <i>title</i> | : Title of the rendering support if supported |
|--------------|---|

Usually opens a window. Some graphic library uses other rendering support.

Implements [Arcade::IGraphicLib](#).

6.12.1.10 pollEvents()

```
bool Arcade::Sdl2::pollEvents ( ) [final], [virtual]
```

Fetches the events from the user and saves it.

Returns

true if at least one command has been fetched, otherwise returns false

Fetches commands are usually stored inside a `std::vector<Arcade::Keys>` or `std::list<Arcade::Keys>`

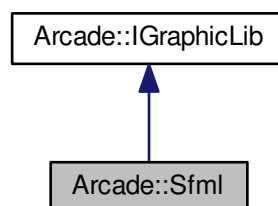
Implements [Arcade::IGraphicLib](#).

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

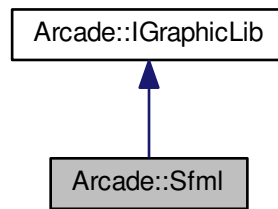
- /home/rectoria/projects/epitech/CPP/cpp_arcade/lib/SDL2/Sdl2.hpp
- /home/rectoria/projects/epitech/CPP/cpp_arcade/lib/SDL2/Sdl2.cpp

6.13 Arcade::Sfml Class Reference

Inheritance diagram for Arcade::Sfml:



Collaboration diagram for Arcade::Sfml:



Public Member Functions

- `std::string getName ()` const final
Graphic library name's getter.
- `bool isOpen ()` const final
Specifies whether the window is open or not.
- `void closeRenderer ()` final
Closes the rendering support.
- `void openRenderer (std::string const &title)` final
Opens the rendering support.
- `void clearWindow ()` final
Clears the rendering support.
- `void refreshWindow ()` final
Displays the buffered frame to the screen.
- `void drawPixelBox (PixelBox const &)` final
Draws a `PixelBox`.
- `void drawText (TextBox const &)` final
Draws a `TextBox`.
- `bool pollEvents ()` final
Fetches the events from the user and saves it.
- `Keys getLastEvent ()` final
Getter of the oldest command in memory.
- `void clearEvents ()` final
Clears the pending commands.
- `Vect< size_t > getScreenSize ()` const final
Getter from the rendering support dimensions.
- `size_t getMaxY ()` const final
Getter from the rendering support height.
- `size_t getMaxX ()` const final
Getter from the rendering support width.

6.13.1 Member Function Documentation

6.13.1.1 clearEvents()

```
void Arcade::Sfml::clearEvents ( ) [final], [virtual]
```

Clears the pending commands.

The function deletes all the commands currently stored. They wont be accessible anymore, even with the [getLastEvent\(\)](#) method.

Implements [Arcade::IGraphicLib](#).

6.13.1.2 closeRenderer()

```
void Arcade::Sfml::closeRenderer ( ) [final], [virtual]
```

Closes the rendering support.

Usually closes a window. Some graphic library uses other rendering support.

Implements [Arcade::IGraphicLib](#).

6.13.1.3 getLastEvent()

```
Arcade::Keys Arcade::Sfml::getLastEvent ( ) [final], [virtual]
```

Getter of the oldest command in memory.

Returns

the first event of the list.

The function deletes the command if it succeed to retrieves one, using `front()` and `pop_front()` methods

Implements [Arcade::IGraphicLib](#).

6.13.1.4 getMaxX()

```
size_t Arcade::Sfml::getMaxX ( ) const [final], [virtual]
```

Getter from the rendering support width.

Returns

the width of the rendering support

Implements [Arcade::IGraphicLib](#).

6.13.1.5 getMaxY()

```
size_t Arcade::Sfml::getMaxY ( ) const [final], [virtual]
```

Getter from the rendering support height.

Returns

the height of the rendering support

Implements [Arcade::IGraphicLib](#).

6.13.1.6 getName()

```
std::string Arcade::Sfml::getName ( ) const [final], [virtual]
```

Graphic library name's getter.

Returns

a string containing the name of the graphic library

Implements [Arcade::IGraphicLib](#).

6.13.1.7 getScreenSize()

```
Arcade::Vect< size_t > Arcade::Sfml::getScreenSize ( ) const [final], [virtual]
```

Getter from the rendering support dimensions.

Returns

a two dimensions vector containing the width and the height of the rendering support

Implements [Arcade::IGraphicLib](#).

6.13.1.8 isOpen()

```
bool Arcade::Sfml::isOpen ( ) const [final], [virtual]
```

Specifies whether the window is open or not.

Returns

true if open, otherwise returns false

Implements [Arcade::IGraphicLib](#).

6.13.1.9 openRenderer()

```
void Arcade::Sfml::openRenderer (
    std::string const & title ) [final], [virtual]
```

Opens the rendering support.

Parameters

| | |
|--------------|---|
| <i>title</i> | : Title of the rendering support if supported |
|--------------|---|

Usually opens a window. Some graphic library uses other rendering support.

Implements [Arcade::IGraphicLib](#).

6.13.1.10 pollEvents()

```
bool Arcade::Sfml::pollEvents ( ) [final], [virtual]
```

Fetches the events from the user and saves it.

Returns

true if at least one command has been fetched, otherwise returns false

Fetches commands are usually stored inside a `std::vector<Arcade::Keys>` or `std::list<Arcade::Keys>`

Implements [Arcade::IGraphicLib](#).

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

- `/home/rectoria/projects/epitech/CPP/cpp_arcade/lib/SFML/Sfml.hpp`
- `/home/rectoria/projects/epitech/CPP/cpp_arcade/lib/SFML/Sfml.cpp`

6.14 Arcade::TextBox Class Reference

[TextBox](#) class.

```
#include <TextBox.hpp>
```

Public Member Functions

- [TextBox](#) (std::string const &text, [Vect](#)< size_t > pos, size_t fontSize=30, [Color](#) color=[Color](#)(255, 255, 255, 255), [Color](#) backgroundColor=[Color](#)(0, 0, 0, 255))
TextBox class's constructor.
- [~TextBox](#) ()=default
PixelBox class's destructor.
- const std::string & [getValue](#) () const
TextBox text's value's getter.
- void [setValue](#) (std::string const &text)
Sets the textBox text's value.
- [Vect](#)< size_t > [getPos](#) () const
TextBox positions's getter.
- void [setPos](#) ([Vect](#)< size_t > pos)
TextBox positions's setter.
- size_t [getX](#) () const
TextBox X offset's getter.
- size_t [getY](#) () const
TextBox Y offset's getter.
- void [setX](#) (size_t x)
TextBox X offset's setter.
- void [setY](#) (size_t y)
TextBox Y offset's setter.
- size_t [getFontSize](#) () const
TextBox's font size's getter.
- void [setFontSize](#) (size_t size)
TextBox's font size's setter.
- [Color](#) [getColor](#) () const
TextBox's text color's getter.
- void [setColor](#) ([Color](#) color)
TextBox's text color's setter.
- [Color](#) [getBackgroundColor](#) () const
TextBox's text background color's getter.
- void [setBackgroundColor](#) ([Color](#) color)
TextBox's text background color's setter.

6.14.1 Detailed Description

[TextBox](#) class.

Class used to represent a rectangle of text

6.14.2 Constructor & Destructor Documentation

6.14.2.1 `TextBox()`

```
Arcade::TextBox::TextBox (
    std::string const & text,
    Arcade::Vect< size_t > pos,
    size_t fontSize = 30,
    Arcade::Color color = Color(255, 255, 255, 255),
    Arcade::Color backgroundColor = Color(0, 0, 0, 255) )
```

`TextBox` class's constructor.

Parameters

| | |
|------------------------|--|
| <i>text</i> | : characters to be apply on the textBox |
| <i>pos</i> | : Vect<size_t> containing both x and y offsets. Used to place the textBox on the rendering support |
| <i>fontSize</i> | : size of the text |
| <i>color</i> | : color of the text |
| <i>backgroundColor</i> | : background color of the text |

Creates a new textBox class instance. The first text argument defines the value of the text within the textBox. The [Vect<size_t>](#) pos argument defines the coordinates of the textBox's position on the rendering support. It will be the offset applied when rendering it. The third fontSize argument defines the size in which the text should be printed. The color's argument defines in which color the characters will be printed. The backgroundColor's argument defines the background color of the characters.

6.14.3 Member Function Documentation

6.14.3.1 getBackgroundColor()

```
Arcade::Color Arcade::TextBox::getBackgroundColor ( ) const
```

[TextBox](#)'s text background color's getter.

Returns

the textBox's text's background color

6.14.3.2 getColor()

```
Arcade::Color Arcade::TextBox::getColor ( ) const
```

[TextBox](#)'s text color's getter.

Returns

the textBox's text's color

6.14.3.3 `getFontSize()`

```
size_t Arcade::TextBox::getFontSize ( ) const
```

`TextBox`'s font size's getter.

Returns

the font size

6.14.3.4 `getPos()`

```
Arcade::Vect< size_t > Arcade::TextBox::getPos ( ) const
```

`TextBox` positions's getter.

Returns

a `Vect<size_t>` containing the offsetX (x) and the offsetY (y) of the textBox.

6.14.3.5 `getValue()`

```
const std::string & Arcade::TextBox::getValue ( ) const
```

`TextBox` text's value's getter.

Returns

the value of the text within textBox

6.14.3.6 `setBackgroundColor()`

```
void Arcade::TextBox::setBackgroundColor (
    Arcade::Color color )
```

`TextBox`'s text background color's setter.

Parameters

| | |
|--------------------|---|
| <code>color</code> | : new background color to apply to text |
|--------------------|---|

6.14.3.7 setColor()

```
void Arcade::TextBox::setColor (
    Arcade::Color color )
```

TextBox's text color's setter.

Parameters

| | |
|--------------|------------------------------|
| <i>color</i> | : new color to apply to text |
|--------------|------------------------------|

6.14.3.8 setFontSize()

```
void Arcade::TextBox::setFontSize (
    size_t size )
```

TextBox's font size's setter.

Parameters

| | |
|-------------|--------------------------------|
| <i>size</i> | : new font size to be assigned |
|-------------|--------------------------------|

6.14.3.9 setPos()

```
void Arcade::TextBox::setPos (
    Arcade::Vect< size_t > pos )
```

TextBox positions's setter.

Parameters

| | |
|------------|--------------------------------|
| <i>pos</i> | : new positions of the textBox |
|------------|--------------------------------|

Takes both new positions as parameter, within a [Vect<size_t>](#)

6.14.3.10 setValue()

```
void Arcade::TextBox::setValue (
    std::string const & text )
```

Sets the textBox text's value.

Parameters

| | |
|-------------|-----------------------|
| <i>text</i> | : new value to assign |
|-------------|-----------------------|

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

- /home/rectoria/projects/epitech/CPP/cpp_arcade/include/TextBox.hpp
- /home/rectoria/projects/epitech/CPP/cpp_arcade/include/TextBox.cpp

6.15 Arcade::Vect< T > Class Template Reference

[Vect](#) class template.

```
#include <Vect.hpp>
```

Public Member Functions

- [Vect](#) (T x=0, T y=0)
Vect class template's constructor.
- void [setXY](#) (T x=0, T y=0)
Vect class template's coordinates's setter.
- void [setX](#) (T x=0)
Vect class template's X coordinate's setter.
- void [setY](#) (T y=0)
Vect class template's Y coordinate's setter.
- T [getX](#) () const
Vect class template's X coordinate's getter.
- T [getY](#) () const
Vect class template's Y coordinate's getter.
- bool [operator==](#) (const [Vect](#)< T > &other) const
Overloading the comparison operator.
- [Vect](#)< T > [operator+](#) (const [Vect](#)< T > &other) const
- [Vect](#)< T > [operator-](#) (const [Vect](#)< T > &other) const
- [Vect](#)< T > [operator*](#) (const [Vect](#)< T > &other) const
- [Vect](#)< T > [operator/](#) (const [Vect](#)< T > &other) const
- [Vect](#)< T > & [operator+=](#) (const [Vect](#)< T > &other)
- [Vect](#)< T > & [operator-=](#) (const [Vect](#)< T > &other)
- [Vect](#)< T > & [operator*=](#) (const [Vect](#)< T > &other)
- [Vect](#)< T > & [operator/=](#) (const [Vect](#)< T > &other)
- [Vect](#)< T > [operator+](#) (const T &other) const
- [Vect](#)< T > [operator-](#) (const T &other) const
- [Vect](#)< T > [operator*](#) (const T &other) const
- [Vect](#)< T > [operator/](#) (const T &other) const
- [Vect](#)< T > & [operator+=](#) (const T &other)
- [Vect](#)< T > & [operator-=](#) (const T &other)
- [Vect](#)< T > & [operator*=](#) (const T &other)
- [Vect](#)< T > & [operator/=](#) (const T &other)

6.15.1 Detailed Description

```
template<typename T>
class Arcade::Vect< T >
```

Vect class template.

Mainly used to store and manage 2 coordinates

6.15.2 Constructor & Destructor Documentation

6.15.2.1 Vect()

```
template<typename T>
Arcade::Vect< T >::Vect (
    T x = 0,
    T y = 0 ) [inline], [explicit]
```

Vect class template's constructor.

Parameters

| | |
|----------|----------------|
| <i>x</i> | : coordinate X |
| <i>y</i> | : coordinate Y |

6.15.3 Member Function Documentation

6.15.3.1 getX()

```
template<typename T>
T Arcade::Vect< T >::getX ( ) const [inline]
```

Vect class template's X coordinate's getter.

Returns

the value of the X coordinate

6.15.3.2 `getY()`

```
template<typename T>
T Arcade::Vect< T >::getY ( ) const [inline]
```

[Vect](#) class template's Y coordinate's getter.

Returns

the value of the Y coordinate

6.15.3.3 `operator*()` [1/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator* (
    const Vect< T > & other ) const [inline]
```

Overloading the multiplication operator

Parameters

| | |
|--------------|--|
| <i>other</i> | : the Vect object to perform the multiplication with |
|--------------|--|

Returns

a new object resulting from the multiplication of the [Vect](#)

6.15.3.4 `operator*()` [2/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator* (
    const T & other ) const [inline]
```

Overloading the multiplication operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the T variable to perform the multiplication with |
|--------------|---|

Returns

a new object resulting from the multiplication

6.15.3.5 `operator*=()` [1/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator*= (
    const Vect< T > & other ) [inline]
```

Overloading the multiplication assignment operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the <code>Vect</code> object to perform the multiplication with |
|--------------|---|

Returns

the object from which this function was called

6.15.3.6 `operator*=()` [2/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator*= (
    const T & other ) [inline]
```

Overloading the multiplication assignment operator

Parameters

| | |
|--------------|--|
| <i>other</i> | : the <code>T</code> variable to perform the multiplication with |
|--------------|--|

Returns

the object from which this function was called

6.15.3.7 `operator+()` [1/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator+ (
    const Vect< T > & other ) const [inline]
```

Overloading the addition operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the <code>Vect</code> object to perform the addition with |
|--------------|---|

Returns

a new object resulting from the addition of the [Vect](#)

6.15.3.8 operator+() [2/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator+ (
    const T & other ) const [inline]
```

Overloading the addition operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the T variable to perform the addition with |
|--------------|---|

Returns

a new object resulting from the addition

6.15.3.9 operator+=() [1/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator+= (
    const Vect< T > & other ) [inline]
```

Overloading the addition assignment operator

Parameters

| | |
|--------------|--|
| <i>other</i> | : the Vect object to perform the addition with |
|--------------|--|

Returns

the object from which this function was called

6.15.3.10 operator+=() [2/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator+= (
    const T & other ) [inline]
```

Overloading the addition assignment operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the T variable to perform the addition with |
|--------------|---|

Returns

the object from which this function was called

6.15.3.11 operator-() [1/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator- (
    const Vect< T > & other ) const [inline]
```

Overloading the subtraction operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the Vect object to perform the subtraction with |
|--------------|---|

Returns

a new object resulting from the subtraction of the Vect

6.15.3.12 operator-() [2/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator- (
    const T & other ) const [inline]
```

Overloading the subtraction operator

Parameters

| | |
|--------------|--|
| <i>other</i> | : the T variable to perform the subtraction with |
|--------------|--|

Returns

a new object resulting from the subtraction

6.15.3.13 operator-=() [1/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator-= (
    const Vect< T > & other ) [inline]
```

Overloading the subtraction assignment operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the Vect object to perform the subtraction with |
|--------------|---|

Returns

the object from which this function was called

6.15.3.14 operator-=() [2/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator-= (
    const T & other ) [inline]
```

Overloading the subtraction assignment operator

Parameters

| | |
|--------------|--|
| <i>other</i> | : the T variable to perform the subtraction with |
|--------------|--|

Returns

the object from which this function was called

6.15.3.15 operator/() [1/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator/ (
    const Vect< T > & other ) const [inline]
```

Overloading the division operator

Parameters

| | |
|--------------|--|
| <i>other</i> | : the Vect object to perform the division with |
|--------------|--|

Returns

a new object resulting from the division of the [Vect](#)

6.15.3.16 `operator/()` [2/2]

```
template<typename T>
Vect<T> Arcade::Vect< T >::operator/ (
    const T & other ) const [inline]
```

Overloading the division operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the T variable to perform the division with |
|--------------|---|

Returns

a new object resulting from the division

6.15.3.17 `operator/=()` [1/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator/= (
    const Vect< T > & other ) [inline]
```

Overloading the division assignment operator

Parameters

| | |
|--------------|--|
| <i>other</i> | : the Vect object to perform the division with |
|--------------|--|

Returns

the object from which this function was called

6.15.3.18 `operator/=()` [2/2]

```
template<typename T>
Vect<T>& Arcade::Vect< T >::operator/= (
    const T & other ) [inline]
```

Overloading the division assignment operator

Parameters

| | |
|--------------|---|
| <i>other</i> | : the T variable to perform the division with |
|--------------|---|

Returns

the object from which this function was called

6.15.3.19 operator==()

```
template<typename T>
bool Arcade::Vect< T >::operator== (
    const Vect< T > & other ) const [inline]
```

Overloading the comparison operator.

Parameters

| | |
|--------------|-----------------------------------|
| <i>other</i> | : the Vect object to compare with |
|--------------|-----------------------------------|

Returns

true if equal, otherwise returns false

6.15.3.20 setX()

```
template<typename T>
void Arcade::Vect< T >::setX (
    T x = 0 ) [inline]
```

Vect class template's X coordinate's setter.

Parameters

| | |
|----------|--------------------|
| <i>x</i> | : new X coordinate |
|----------|--------------------|

6.15.3.21 setXY()

```
template<typename T>
void Arcade::Vect< T >::setXY (
```

```
T x = 0,  
T y = 0 ) [inline]
```

[Vect](#) class template's coordinates's setter.

Parameters

| | |
|----------|--------------------|
| <i>x</i> | : new X coordinate |
| <i>y</i> | : new Y coordinate |

6.15.3.22 setY()

```
template<typename T>  
void Arcade::Vect< T >::setY (  
    T y = 0 ) [inline]
```

[Vect](#) class template's Y coordinate's setter.

Parameters

| | |
|----------|--------------------|
| <i>x</i> | : new Y coordinate |
|----------|--------------------|

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

- /home/rectoria/projects/epitech/CPP/cpp_arcade/include/Vect.hpp

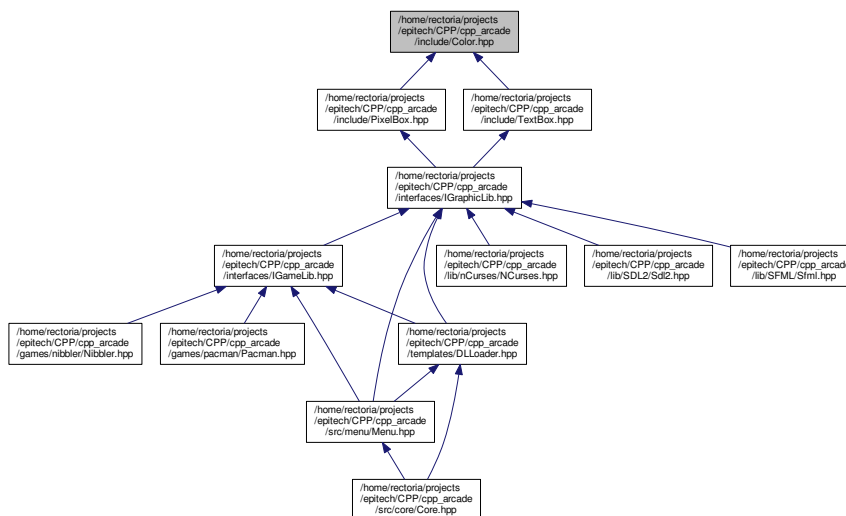
Chapter 7

File Documentation

7.1 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/Color.hpp File Reference

Color class, pixel-like.

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



Classes

- class [Arcade::Color](#)
Color class.

Namespaces

- [Arcade](#)
Arcade project namespace.

7.1.1 Detailed Description

Color class, pixel-like.

Authors

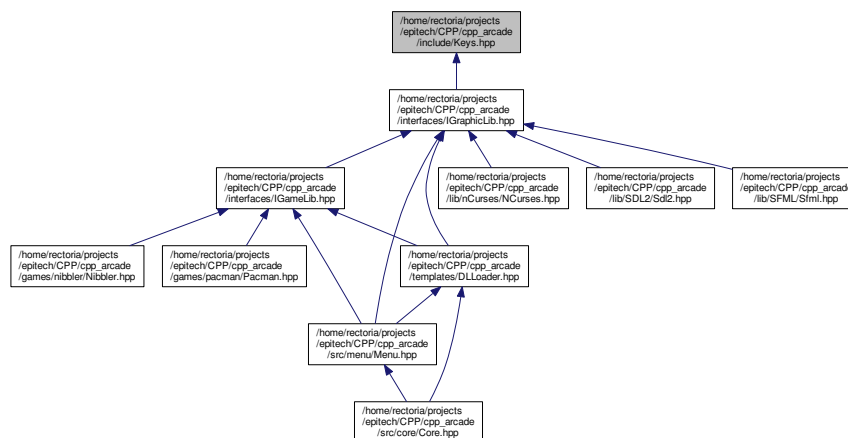
<https://github.com/EPITECH-Strasbourg-2021/CPP-Arcade-Spec>

Class used by games and graphic libraries, as a color's array All functions must be implemented correctly for libraries to function properly.

7.2 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/Keys.hpp File Reference

Keys enum.

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



Namespaces

- [Arcade](#)
Arcade project namespace.

Enumerations

- enum [Arcade::Keys](#) {
NONE, A, B, C,
D, E, F, G,
H, I, J, K,
L, M, N, O,
P, Q, R, S,
T, U, V, W,
X, Y, Z, LEFT,
RIGHT, UP, DOWN, ENTER,
SPACE, DELETE, BACKSPACE, TAB,
ESC, MOUSELEFT, MUSERIGHT }

7.2.1 Detailed Description

Keys enum.

Authors

<https://github.com/EPITECH-Strasbourg-2021/CPP-Arcade-Spec>

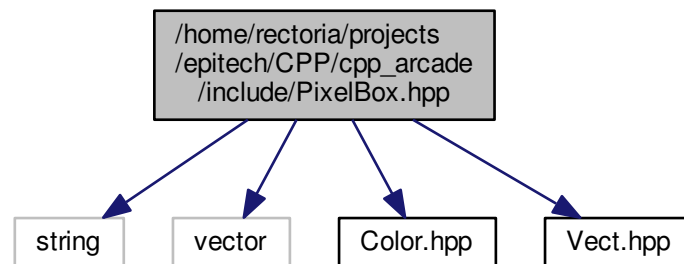
Key Enum, each graphics library must store a map in order to convert the specific library key code into one of this enum code so that it can be used by other components independently of the graphics library.

7.3 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/PixelBox.hpp File Reference

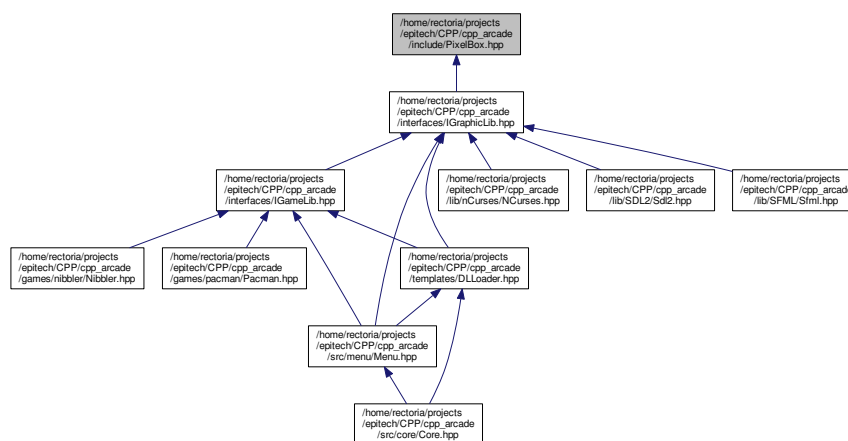
PixelBox class, similar to a rectangle of pixels.

```
#include <string>
#include <vector>
#include "Color.hpp"
#include "Vect.hpp"
```

Include dependency graph for PixelBox.hpp:



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



Classes

- class [Arcade::PixelFormat](#)
PixelFormat class.

Namespaces

- [Arcade](#)
Arcade project namespace.

7.3.1 Detailed Description

PixelFormat class, similar to a rectangle of pixels.

Authors

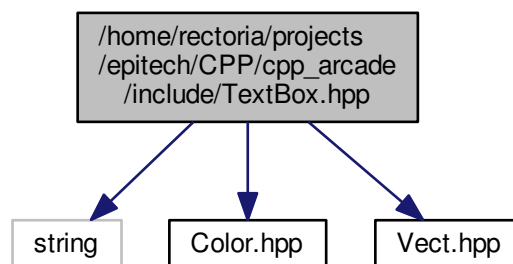
<https://github.com/EPITECH-Strasbourg-2021/CPP-Arcade-Spec>

Class used by games and graphic libraries, similar to a rectangle of pixels. All functions must be implemented correctly for libraries to function properly.

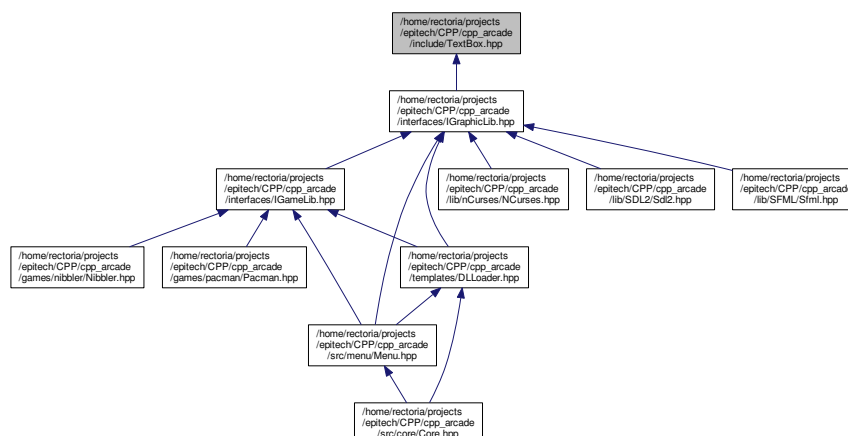
7.4 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/TextBox.hpp File Reference

TextBox class, similar to a text rectangle.

```
#include <string>
#include "Color.hpp"
#include "Vect.hpp"
Include dependency graph for TextBox.hpp:
```



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



Classes

- class [Arcade::TextBox](#)

[TextBox](#) class.

Namespaces

- [Arcade](#)

[Arcade](#) project namespace.

7.4.1 Detailed Description

TextBox class, similar to a text rectangle.

Authors

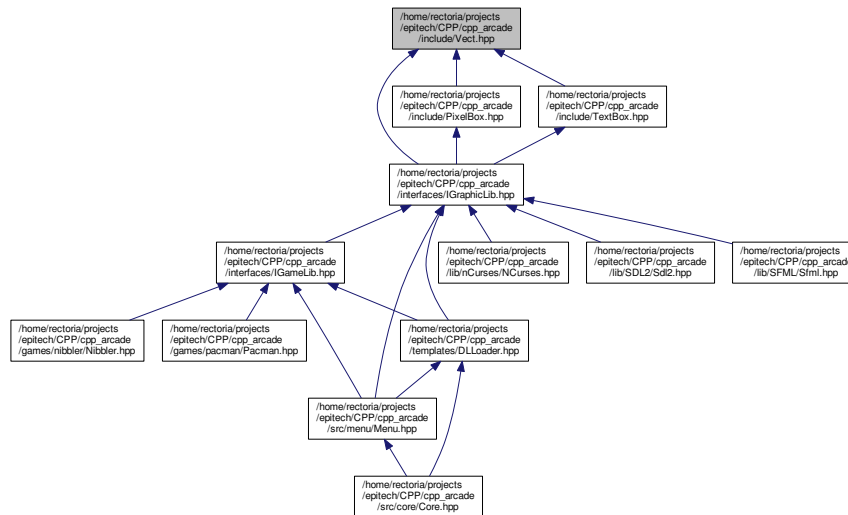
<https://github.com/EPITECH-Strasbourg-2021/CPP-Arcade-Spec>

Class used by games and graphic libraries, similar to a text rectangle. All functions must be implemented correctly for libraries to function properly.

7.5 /home/rectoria/projects/epitech/CPP/cpp_arcade/include/Vect.hpp File Reference

Project-specific vector template.

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



Classes

- class [Arcade::Vect< T >](#)
Vect class template.

Namespaces

- [Arcade](#)
Arcade project namespace.

7.5.1 Detailed Description

Project-specific vector template.

Authors

<https://github.com/EPITECH-Strasbourg-2021/CPP-Arcade-Spec>

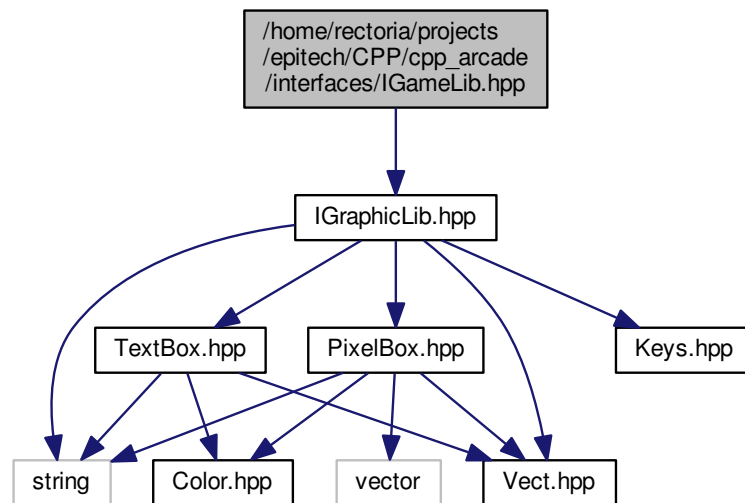
Template used to store and perform arithmetic operations on coordinates.

7.6 /home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/IGameLib.hpp File Reference

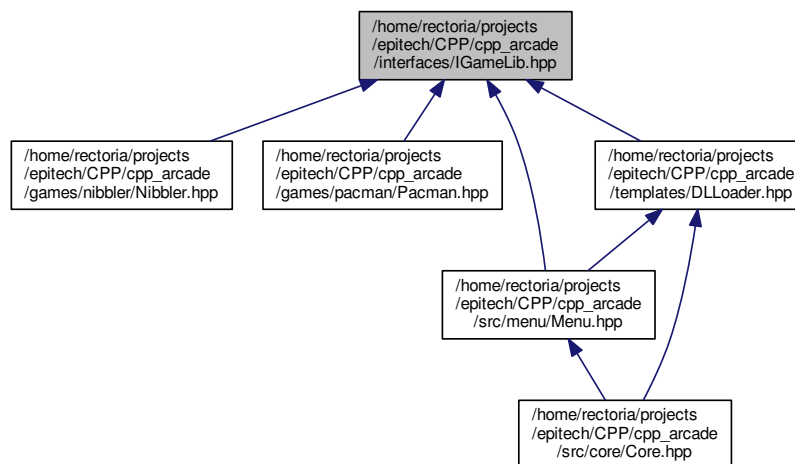
Game libraries dedicated class interface.

```
#include "IGraphicLib.hpp"
```

Include dependency graph for IGameLib.hpp:



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



Classes

- class [Arcade::IGameLib](#)
Game libraries virtual class.

Namespaces

- [Arcade](#)

[Arcade](#) project namespace.

7.6.1 Detailed Description

Game libraries dedicated class interface.

Authors

<https://github.com/EPITECH-Strasbourg-2021/CPP-Arcade-Spec>

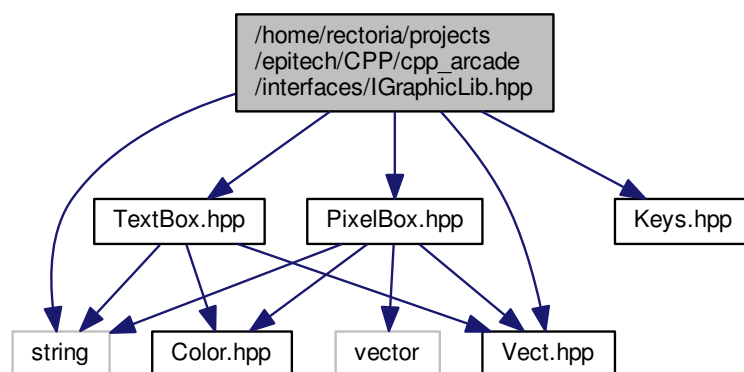
Interface used by game libraries. All functions must be implemented correctly for the kernel to handle the game libraries.

7.7 /home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/IGraphicLib.hpp File Reference

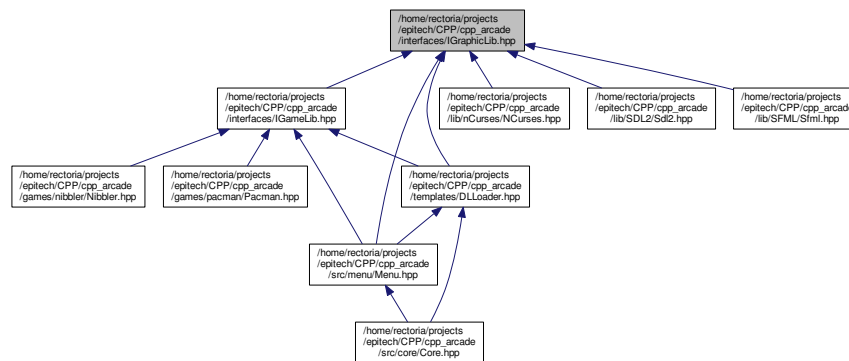
Graphic libraries dedicated class interface.

```
#include <string>
#include "Vect.hpp"
#include "PixelBox.hpp"
#include "TextBox.hpp"
#include "Keys.hpp"
```

Include dependency graph for IGraphicLib.hpp:



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



Classes

- class [Arcade::IGraphicLib](#)
Graphic libraries virtual class.

Namespaces

- [Arcade](#)
Arcade project namespace.

7.7.1 Detailed Description

Graphic libraries dedicated class interface.

Authors

<https://github.com/EPITECH-Strasbourg-2021/CPP-Arcade-Spec>

Interface used by graphic libraries All functions must be implemented correctly for the kernel to handle the graphic libraries.

Index

/home/rectoria/projects/epitech/CPP/cpp_arcade/include/↵
Color.hpp, 67
/home/rectoria/projects/epitech/CPP/cpp_arcade/include/↵
Keys.hpp, 68
/home/rectoria/projects/epitech/CPP/cpp_arcade/include/↵
PixelBox.hpp, 69
/home/rectoria/projects/epitech/CPP/cpp_arcade/include/↵
TextBox.hpp, 70
/home/rectoria/projects/epitech/CPP/cpp_arcade/include/↵
Vect.hpp, 72
/home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/↵
IGameLib.hpp, 73
/home/rectoria/projects/epitech/CPP/cpp_arcade/interfaces/↵
IGraphicLib.hpp, 74
~IGameLib
Arcade::IGameLib, 16
~IGraphicLib
Arcade::IGraphicLib, 20

applyEvent
Arcade::IGameLib, 17
Arcade::Nibbler, 29
Arcade::Pacman, 33

Arcade, 9
Keys, 10

Arcade::Color, 11
Color, 12
getAlpha, 12
getBlue, 12
getGreen, 12
getRed, 13
operator unsigned char *, 13
operator==, 13
setAlpha, 14
setBlue, 14
setColor, 14
setGreen, 14
setRed, 14

Arcade::Core, 15
Arcade::IGameLib, 16
~IGameLib, 16
applyEvent, 17
getName, 17
getScore, 17
init, 18
refresh, 18
stop, 18
update, 19

Arcade::IGraphicLib, 19
~IGraphicLib, 20

clearEvents, 21
closeRenderer, 21
getLastEvent, 21
getMaxX, 21
getMaxY, 22
getName, 22
getScreenSize, 22
isOpen, 22
openRenderer, 23
pollEvents, 23

Arcade::Menu, 24
Arcade::NCurses, 24
clearEvents, 25
closeRenderer, 25
getLastEvent, 26
getMaxX, 26
getMaxY, 26
getName, 26
getScreenSize, 27
isOpen, 27
openRenderer, 27
pollEvents, 28

Arcade::Nibbler, 28
applyEvent, 29
getName, 30
getScore, 30
init, 30
refresh, 30
stop, 31
update, 31

Arcade::Pacman, 32
applyEvent, 33
getName, 33
getScore, 33
init, 33
refresh, 34
stop, 34
update, 34

Arcade::PixelBox, 35
getHeight, 37
getPixel, 37
getPixelArray, 37
getPos, 37
getSize, 38
getWidth, 38
getX, 38
getY, 38
PixelBox, 36
putPixel, 39

- putRect, [39](#)
- setPos, [39](#)
- setSize, [41](#)
- Arcade::Score, [41](#)
- Arcade::Sdl2, [42](#)
 - clearEvents, [43](#)
 - closeRenderer, [43](#)
 - getLastEvent, [43](#)
 - getMaxX, [44](#)
 - getMaxY, [44](#)
 - getName, [44](#)
 - getScreenSize, [45](#)
 - isOpen, [45](#)
 - openRenderer, [45](#)
 - pollEvents, [46](#)
- Arcade::Sfml, [46](#)
 - clearEvents, [47](#)
 - closeRenderer, [48](#)
 - getLastEvent, [48](#)
 - getMaxX, [48](#)
 - getMaxY, [48](#)
 - getName, [49](#)
 - getScreenSize, [49](#)
 - isOpen, [49](#)
 - openRenderer, [49](#)
 - pollEvents, [50](#)
- Arcade::TextBox, [50](#)
 - getBackgroundColor, [53](#)
 - getColor, [53](#)
 - getFontSize, [53](#)
 - getPos, [54](#)
 - getValue, [54](#)
 - setBackgroundColor, [54](#)
 - setColor, [55](#)
 - setFontSize, [55](#)
 - setPos, [55](#)
 - setValue, [55](#)
 - TextBox, [51](#)
- Arcade::Vect
 - getX, [57](#)
 - getY, [57](#)
 - operator*, [58](#)
 - operator*=[58](#), [59](#)
 - operator+, [59](#), [60](#)
 - operator+=, [60](#)
 - operator-, [61](#)
 - operator-=, [61](#), [62](#)
 - operator/, [62](#), [63](#)
 - operator/=, [63](#)
 - operator==, [64](#)
 - setXY, [64](#)
 - setX, [64](#)
 - setY, [65](#)
 - Vect, [57](#)
- Arcade::Vect< T >, [56](#)
- clearEvents
 - Arcade::IGraphicLib, [21](#)
 - Arcade::NCurses, [25](#)
 - Arcade::Sdl2, [43](#)
 - Arcade::Sfml, [47](#)
- closeRenderer
 - Arcade::IGraphicLib, [21](#)
 - Arcade::NCurses, [25](#)
 - Arcade::Sdl2, [43](#)
 - Arcade::Sfml, [48](#)
- Color
 - Arcade::Color, [12](#)
- DLLoader< T >, [15](#)
- getAlpha
 - Arcade::Color, [12](#)
- getBackgroundColor
 - Arcade::TextBox, [53](#)
- getBlue
 - Arcade::Color, [12](#)
- getColor
 - Arcade::TextBox, [53](#)
- getFontSize
 - Arcade::TextBox, [53](#)
- getGreen
 - Arcade::Color, [12](#)
- getHeight
 - Arcade::PictureBox, [37](#)
- getLastEvent
 - Arcade::IGraphicLib, [21](#)
 - Arcade::NCurses, [26](#)
 - Arcade::Sdl2, [43](#)
 - Arcade::Sfml, [48](#)
- getMaxX
 - Arcade::IGraphicLib, [21](#)
 - Arcade::NCurses, [26](#)
 - Arcade::Sdl2, [44](#)
 - Arcade::Sfml, [48](#)
- getMaxY
 - Arcade::IGraphicLib, [22](#)
 - Arcade::NCurses, [26](#)
 - Arcade::Sdl2, [44](#)
 - Arcade::Sfml, [48](#)
- getName
 - Arcade::IGameLib, [17](#)
 - Arcade::IGraphicLib, [22](#)
 - Arcade::NCurses, [26](#)
 - Arcade::Nibbler, [30](#)
 - Arcade::Pacman, [33](#)
 - Arcade::Sdl2, [44](#)
 - Arcade::Sfml, [49](#)
- getPixel
 - Arcade::PictureBox, [37](#)
- getPixelArray
 - Arcade::PictureBox, [37](#)
- getPos
 - Arcade::PictureBox, [37](#)
 - Arcade::TextBox, [54](#)
- getRed
 - Arcade::Color, [13](#)
- getScore

- Arcade::IGameLib, 17
- Arcade::Nibbler, 30
- Arcade::Pacman, 33
- getScreenSize
 - Arcade::IGraphicLib, 22
 - Arcade::NCurses, 27
 - Arcade::Sdl2, 45
 - Arcade::Sfml, 49
- getSize
 - Arcade::PictureBox, 38
- getValue
 - Arcade::TextBox, 54
- getWidth
 - Arcade::PictureBox, 38
- getX
 - Arcade::PictureBox, 38
 - Arcade::Vect, 57
- getY
 - Arcade::PictureBox, 38
 - Arcade::Vect, 57
- init
 - Arcade::IGameLib, 18
 - Arcade::Nibbler, 30
 - Arcade::Pacman, 33
- isOpen
 - Arcade::IGraphicLib, 22
 - Arcade::NCurses, 27
 - Arcade::Sdl2, 45
 - Arcade::Sfml, 49
- Keys
 - Arcade, 10
- openRenderer
 - Arcade::IGraphicLib, 23
 - Arcade::NCurses, 27
 - Arcade::Sdl2, 45
 - Arcade::Sfml, 49
- operator unsigned char *
 - Arcade::Color, 13
- operator*
 - Arcade::Vect, 58
- operator*=
 - Arcade::Vect, 58, 59
- operator+
 - Arcade::Vect, 59, 60
- operator+=
 - Arcade::Vect, 60
- operator-
 - Arcade::Vect, 61
- operator-=
 - Arcade::Vect, 61, 62
- operator/
 - Arcade::Vect, 62, 63
- operator/=
 - Arcade::Vect, 63
- operator==
 - Arcade::Color, 13
- Arcade::Vect, 64
- PictureBox
 - Arcade::PictureBox, 36
- pollEvents
 - Arcade::IGraphicLib, 23
 - Arcade::NCurses, 28
 - Arcade::Sdl2, 46
 - Arcade::Sfml, 50
- putPixel
 - Arcade::PictureBox, 39
- putRect
 - Arcade::PictureBox, 39
- refresh
 - Arcade::IGameLib, 18
 - Arcade::Nibbler, 30
 - Arcade::Pacman, 34
- setAlpha
 - Arcade::Color, 14
- setBackgroundColor
 - Arcade::TextBox, 54
- setBlue
 - Arcade::Color, 14
- setColor
 - Arcade::Color, 14
 - Arcade::TextBox, 55
- setFontSize
 - Arcade::TextBox, 55
- setGreen
 - Arcade::Color, 14
- setPos
 - Arcade::PictureBox, 39
 - Arcade::TextBox, 55
- setRed
 - Arcade::Color, 14
- setSize
 - Arcade::PictureBox, 41
- setValue
 - Arcade::TextBox, 55
- setXY
 - Arcade::Vect, 64
- setX
 - Arcade::Vect, 64
- setY
 - Arcade::Vect, 65
- stop
 - Arcade::IGameLib, 18
 - Arcade::Nibbler, 31
 - Arcade::Pacman, 34
- TextBox
 - Arcade::TextBox, 51
- update
 - Arcade::IGameLib, 19
 - Arcade::Nibbler, 31
 - Arcade::Pacman, 34

Vect

Arcade::Vect, [57](#)