

Arcade

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	Arcade Namespace Reference	7
4.1.1	Detailed Description	8
4.1.2	Enumeration Type Documentation	8
4.1.2.1	Keys	8
5	Class Documentation	11
5.1	Arcade::Color Class Reference	11
5.1.1	Detailed Description	11
5.1.2	Constructor & Destructor Documentation	12
5.1.2.1	Color()	12
5.1.3	Member Function Documentation	12
5.1.3.1	getAlpha()	12
5.1.3.2	getBlue()	12
5.1.3.3	getGreen()	13
5.1.3.4	getRed()	13

5.1.3.5	operator unsigned char *()	13
5.1.3.6	operator==()	13
5.1.3.7	setAlpha()	14
5.1.3.8	setBlue()	14
5.1.3.9	setColor()	14
5.1.3.10	setGreen()	14
5.1.3.11	setRed()	15
5.2	Arcade::IGameLib Class Reference	15
5.2.1	Detailed Description	15
5.2.2	Constructor & Destructor Documentation	15
5.2.2.1	~IGameLib()	16
5.2.3	Member Function Documentation	16
5.2.3.1	applyEvent()	16
5.2.3.2	getName()	16
5.2.3.3	getScore()	16
5.2.3.4	init()	17
5.2.3.5	refresh()	17
5.2.3.6	stop()	17
5.2.3.7	update()	17
5.3	Arcade::IGraphicLib Class Reference	18
5.3.1	Detailed Description	18
5.3.2	Constructor & Destructor Documentation	18
5.3.2.1	~IGraphicLib()	19
5.3.3	Member Function Documentation	19
5.3.3.1	clearEvents()	19
5.3.3.2	clearWindow()	19
5.3.3.3	closeRenderer()	19
5.3.3.4	drawPixelBox()	19
5.3.3.5	drawText()	20
5.3.3.6	getLastEvent()	20

5.3.3.7	getMaxX()	20
5.3.3.8	getMaxY()	20
5.3.3.9	getName()	21
5.3.3.10	getScreenSize()	21
5.3.3.11	isOpen()	21
5.3.3.12	openRenderer()	21
5.3.3.13	pollEvents()	22
5.3.3.14	refreshWindow()	22
5.4	Arcade::PixelBox Class Reference	22
5.4.1	Detailed Description	23
5.4.2	Constructor & Destructor Documentation	23
5.4.2.1	PixelBox()	23
5.4.2.2	~PixelBox()	24
5.4.3	Member Function Documentation	24
5.4.3.1	getHeight()	24
5.4.3.2	getPixel()	24
5.4.3.3	getPixelArray()	24
5.4.3.4	getPos()	25
5.4.3.5	getSize()	25
5.4.3.6	getWidth()	25
5.4.3.7	getX()	25
5.4.3.8	getY()	26
5.4.3.9	putPixel()	26
5.4.3.10	putRect()	26
5.4.3.11	setHeight()	26
5.4.3.12	setPos()	27
5.4.3.13	setSize()	27
5.4.3.14	setWidth()	27
5.4.3.15	setX()	27
5.4.3.16	setY()	28

5.5	Arcade::Scale Class Reference	28
5.5.1	Detailed Description	28
5.5.2	Member Enumeration Documentation	28
5.5.2.1	CENTERING	28
5.5.3	Constructor & Destructor Documentation	29
5.5.3.1	Scale()	29
5.5.3.2	~Scale()	29
5.5.4	Member Function Documentation	29
5.5.4.1	scalePictureBox()	29
5.5.4.2	scaleTextBox()	30
5.5.4.3	setCentering()	30
5.5.4.4	setWindowSize()	31
5.6	Arcade::Scoreboard Class Reference	31
5.6.1	Detailed Description	32
5.6.2	Constructor & Destructor Documentation	32
5.6.2.1	Scoreboard()	32
5.6.2.2	~Scoreboard()	32
5.6.3	Member Function Documentation	32
5.6.3.1	addScores()	32
5.6.3.2	getLastPlayerScore()	32
5.6.3.3	getScoreboard()	33
5.6.3.4	getScores()	33
5.6.3.5	readScoreboard()	33
5.6.3.6	resetScores()	33
5.6.3.7	setGameName()	33
5.6.3.8	setPlayerName()	34
5.6.3.9	subScores()	34
5.7	Arcade::TextBox Class Reference	34
5.7.1	Detailed Description	35
5.7.2	Constructor & Destructor Documentation	35

5.7.2.1	TextBox()	35
5.7.2.2	~TextBox()	36
5.7.3	Member Function Documentation	36
5.7.3.1	getBackgroundColor()	36
5.7.3.2	getColor()	36
5.7.3.3	getFontSize()	36
5.7.3.4	getPos()	37
5.7.3.5	getValue()	37
5.7.3.6	getX()	37
5.7.3.7	getY()	37
5.7.3.8	setBackgroundColor()	37
5.7.3.9	setColor()	38
5.7.3.10	setFontSize()	38
5.7.3.11	setPos()	38
5.7.3.12	setValue()	39
5.7.3.13	setX()	39
5.7.3.14	setY()	39
5.8	Arcade::Vect< T > Class Template Reference	39
5.8.1	Detailed Description	40
5.8.2	Constructor & Destructor Documentation	40
5.8.2.1	Vect()	40
5.8.3	Member Function Documentation	41
5.8.3.1	getX()	41
5.8.3.2	getY()	41
5.8.3.3	operator*() [1/2]	41
5.8.3.4	operator*() [2/2]	42
5.8.3.5	operator*=() [1/2]	42
5.8.3.6	operator*=() [2/2]	42
5.8.3.7	operator+() [1/2]	43
5.8.3.8	operator+() [2/2]	43

5.8.3.9	operator+=() [1/2]	44
5.8.3.10	operator+=() [2/2]	44
5.8.3.11	operator-() [1/2]	44
5.8.3.12	operator-() [2/2]	45
5.8.3.13	operator-=() [1/2]	45
5.8.3.14	operator-=() [2/2]	45
5.8.3.15	operator/() [1/2]	46
5.8.3.16	operator/() [2/2]	46
5.8.3.17	operator/=() [1/2]	47
5.8.3.18	operator/=() [2/2]	47
5.8.3.19	operator==()	47
5.8.3.20	setX()	48
5.8.3.21	setXY()	48
5.8.3.22	setY()	48
6	File Documentation	51
6.1	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Color.cpp File Reference	51
6.2	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Color.hpp File Reference	51
6.2.1	Detailed Description	51
6.3	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/IGameLib.hpp File Reference	52
6.3.1	Detailed Description	52
6.4	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/IGraphicLib.hpp File Reference	52
6.4.1	Detailed Description	53
6.5	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Keys.hpp File Reference	53
6.5.1	Detailed Description	53
6.6	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/PixelBox.cpp File Reference	54
6.7	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/PixelBox.hpp File Reference	54
6.7.1	Detailed Description	54
6.8	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.cpp File Reference	54
6.9	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.hpp File Reference	55
6.10	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scoreboard.cpp File Reference	55
6.11	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scoreboard.hpp File Reference	55
6.11.1	Macro Definition Documentation	56
6.11.1.1	SCOREBOARD	56
6.12	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/TextBox.cpp File Reference	56
6.13	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/TextBox.hpp File Reference	56
6.13.1	Detailed Description	56
6.14	/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Vect.hpp File Reference	57
6.14.1	Detailed Description	57
Index		59

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Arcade	
Arcade project namespace	7

Chapter 2

Class Index

2.1 Class List

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

Arcade::Color	
Color class	11
Arcade::IGameLib	
Game libraries virtual class	15
Arcade::IGraphicLib	
Graphic libraries virtual class	18
Arcade::PixelFormat	
PixelFormat class	22
Arcade::Scale	
Scale class	28
Arcade::Scoreboard	
Scoreboard class	31
Arcade::TextBox	
TextBox class	34
Arcade::Vect< T >	
Vect class template	39

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Color.cpp	51
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Color.hpp	
Color class, pixel-like	51
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/IGameLib.hpp	
Game libraries dedicated class interface	52
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/IGraphicLib.hpp	
Graphic libraries dedicated class interface	52
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Keys.hpp	
Keys enum	53
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/PixelBox.cpp	54
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/PixelBox.hpp	
PixelBox class, similar to a rectangle of pixels	54
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.cpp	54
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.hpp	55
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scoreboard.cpp	55
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scoreboard.hpp	55
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/TextBox.cpp	56
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/TextBox.hpp	
TextBox class, similar to a text rectangle	56
/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Vect.hpp	
Project-specific vector template	57

Chapter 4

Namespace Documentation

4.1 Arcade Namespace Reference

[Arcade](#) project namespace.

Classes

- class [Color](#)
Color class.
- class [IGameLib](#)
Game libraries virtual class.
- class [IGraphicLib](#)
Graphic libraries virtual class.
- class [PixelBox](#)
PixelBox class.
- class [Scale](#)
Scale class.
- class [Scoreboard](#)
Scoreboard class.
- 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](#), [MOUSERIGHT](#) }

4.1.1 Detailed Description

[Arcade](#) project namespace.

4.1.2 Enumeration Type Documentation

4.1.2.1 Keys

enum [Arcade::Keys](#)

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

Enumerator

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	

Enumerator

TAB	
ESC	
MOUSELEFT	
MOUSERIGHT	

Chapter 5

Class Documentation

5.1 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

5.1.1 Detailed Description

[Color](#) class.

Class used to represent a pixel

5.1.2 Constructor & Destructor Documentation

5.1.2.1 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).

5.1.3 Member Function Documentation

5.1.3.1 getAlpha()

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

Alpha subpixel's getter.

Returns

the alpha subpixel's value

5.1.3.2 getBlue()

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

Blue subpixel's getter.

Returns

the blue subpixel's value

5.1.3.3 getGreen()

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

Green subpixel's getter.

Returns

the green subpixel's value

5.1.3.4 getRed()

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

Red subpixel's getter.

Returns

the red subpixel's value

5.1.3.5 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

5.1.3.6 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

5.1.3.7 setAlpha()

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

Alpha subpixel's setter.

Sets the value of the alpha's subpixel

5.1.3.8 setBlue()

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

Blue subpixel's setter.

Sets the value of the blue's subpixel

5.1.3.9 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).

5.1.3.10 setGreen()

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

Green subpixel's setter.

Sets the value of the green's subpixel

5.1.3.11 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/thibrex/epitech/CPP/cpp_arcade/shared_header/Color.hpp](#)
- [/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Color.cpp](#)

5.2 Arcade::IGameLib Class Reference

Game libraries virtual class.

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

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.

5.2.1 Detailed Description

Game libraries virtual class.

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

5.2.2 Constructor & Destructor Documentation

5.2.2.1 ~IGameLib()

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

Destructor.

[IGameLib](#) class's destructor

5.2.3 Member Function Documentation

5.2.3.1 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

5.2.3.2 getName()

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

Game name's getter.

Returns

a string containing the name of the game

5.2.3.3 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

5.2.3.4 init()

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

Init the resources needed by the game to run.

Returns

true if succeed, otherwise returns false

5.2.3.5 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.

5.2.3.6 stop()

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

Unloads the library.

Returns

true if succeed, otherwise returns false

5.2.3.7 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

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

- /home/thibrex/epitech/CPP/cpp_arcade/shared_header/[IGameLib.hpp](#)

5.3 Arcade::IGraphicLib Class Reference

Graphic libraries virtual class.

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

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.

5.3.1 Detailed Description

Graphic libraries virtual class.

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

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ~IGraphicLib()

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

Destructor.

[IGraphicLib](#) class's destructor

5.3.3 Member Function Documentation

5.3.3.1 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.

5.3.3.2 clearWindow()

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

Clears the rendering support.

5.3.3.3 closeRenderer()

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

Closes the rendering support.

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

5.3.3.4 drawPixelBox()

```
virtual void Arcade::IGraphicLib::drawPixelBox (
    PixelBox const & ) [pure virtual]
```

Draws a [PixelBox](#).

5.3.3.5 drawText()

```
virtual void Arcade::IGraphicLib::drawText (
    TextBox const & ) [pure virtual]
```

Draws a [TextBox](#).

5.3.3.6 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

5.3.3.7 getMaxX()

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

Getter from the rendering support width.

Returns

the width of the rendering support

5.3.3.8 getMaxY()

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

Getter from the rendering support height.

Returns

the height of the rendering support

5.3.3.9 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

5.3.3.10 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

5.3.3.11 isOpen()

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

Specifies whether the window is open or not.

Returns

true if open, otherwise returns false

5.3.3.12 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.

5.3.3.13 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>`

5.3.3.14 refreshWindow()

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

Displays the buffered frame to the screen.

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

- /home/thibrex/epitech/CPP/cpp_arcade/shared_header/IGraphicLib.hpp

5.4 Arcade::PixelFormat Class Reference

[PixelFormat](#) class.

```
#include <PixelFormat.hpp>
```

Public Member Functions

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

- PixelFormat X offset's getter.*
- void `setWidth` (`size_t` width)
- PixelFormat height's setter.*
- void `setX` (`size_t` x)
- PixelFormat 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.*

5.4.1 Detailed Description

`PixelFormat` class.

Class used to represent a rectangle of pixels

5.4.2 Constructor & Destructor Documentation

5.4.2.1 PixelBox()

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

`PixelFormat` class's constructor.

Parameters

<i>size</i>	: <code>Vect<size_t></code> containing the width (x) and the height (y) of the pixelBox
<i>pos</i>	: <code>Vect<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 pixel↵

Box. 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.

5.4.2.2 ~PixelFormat()

```
Arcade::PixelFormat::~PixelFormat ( ) [default]
```

[PixelFormat](#) class's destructor.

5.4.3 Member Function Documentation

5.4.3.1 getHeight()

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

[PixelFormat](#) height's getter.

Returns

the pixelBox's height

5.4.3.2 getPixel()

```
Arcade::Color Arcade::PixelFormat::getPixel (
    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

5.4.3.3 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.

5.4.3.4 getPos()

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

[PixelFormat](#) positions's getter.

Returns

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

5.4.3.5 getSize()

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

[PixelFormat](#) dimensions's getter.

Returns

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

5.4.3.6 getWidth()

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

[PixelFormat](#) width's getter.

Returns

the pixelBox's height

5.4.3.7 getX()

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

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

Returns

the pixelBox X's offset

5.4.3.8 getY()

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

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

Returns

the pixelBox Y's offset

5.4.3.9 putPixel()

```
void Arcade::PixelBox::putPixel (
    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

5.4.3.10 putRect()

```
void Arcade::PixelBox::putRect (
    Vect< size_t > pos,
    Vect< size_t > size,
    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

5.4.3.11 setHeight()

```
void Arcade::PixelBox::setHeight (
    size_t height )
```

[PixelFormat](#) height setter.

5.4.3.12 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>](#)

5.4.3.13 setSize()

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

[PixelFormat](#) 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>](#)

5.4.3.14 setWidth()

```
void Arcade::PixelFormat::setWidth (
    size_t width )
```

[PixelFormat](#) height's setter.

5.4.3.15 setX()

```
void Arcade::PixelFormat::setX (
    size_t x )
```

[PixelFormat](#) X offset's setter.

5.4.3.16 setY()

```
void Arcade::PixelBox::setY (
    size_t y )
```

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

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

- [/home/thibrex/epitech/CPP/cpp_arcade/shared_header/PixelBox.hpp](#)
- [/home/thibrex/epitech/CPP/cpp_arcade/shared_header/PixelBox.cpp](#)

5.5 Arcade::Scale Class Reference

[Scale](#) class.

```
#include <Scale.hpp>
```

Public Types

- enum [CENTERING](#) { [NONE](#), [HORIZONTAL](#), [VERTICAL](#), [BOTH](#) }
Enumeration used by setCentering method.

Public Member Functions

- [Scale](#) ()
Scale class's constructor Creates a new Scale class instance.
- [~Scale](#) ()
Scale class's destructor.
- void [setCentering](#) (const [CENTERING](#) &)
Scale centering's setter.
- void [setWindowSize](#) (const [Arcade::Vect](#)< size_t > &windowsSize)
Scale window size's setter.
- void [scalePixelBox](#) (const [Arcade::Vect](#)< size_t > &pos, const [Arcade::Vect](#)< size_t > &size, [Arcade::PixelBox](#) &pixelBox)
Scale main function for PixelBoxes.
- void [scaleTextBox](#) (const [Arcade::Vect](#)< double > &pos, [Arcade::TextBox](#) &textBox)
Scale main function for TextBoxes.

5.5.1 Detailed Description

[Scale](#) class.

Class used to scale PixelBoxes

5.5.2 Member Enumeration Documentation

5.5.2.1 CENTERING

```
enum Arcade::Scale::CENTERING
```

Enumeration used by setCentering method.

Enumerator

NONE	
HORIZONTAL	
VERTICAL	
BOTH	

5.5.3 Constructor & Destructor Documentation

5.5.3.1 Scale()

```
Arcade::Scale::Scale ( )
```

[Scale](#) class's constructor Creates a new [Scale](#) class instance.

5.5.3.2 ~Scale()

```
Arcade::Scale::~~Scale ( )
```

[Scale](#) class's destructor.

5.5.4 Member Function Documentation

5.5.4.1 scalePictureBox()

```
void Arcade::Scale::scalePictureBox (
    const Arcade::Vect< size_t > & pos,
    const Arcade::Vect< size_t > & size,
    Arcade::PictureBox & pictureBox )
```

[Scale](#) main function for PictureBoxes.

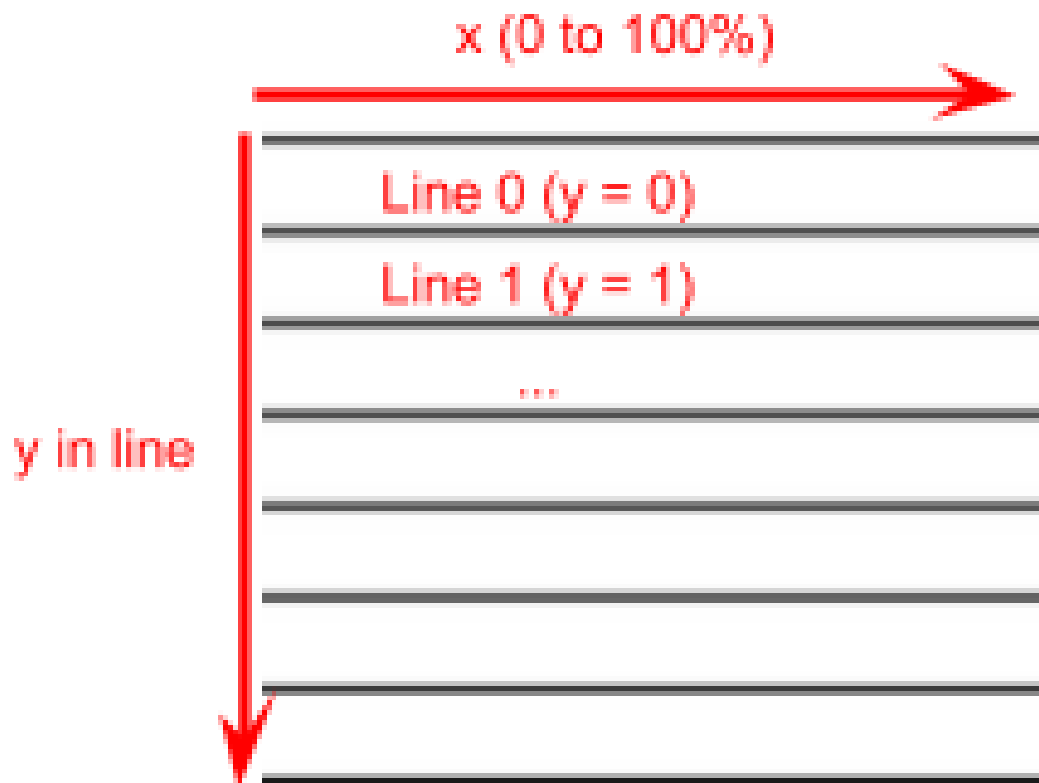
Modifies the [PictureBox](#) giving as parameter and sets the position and size using the [Arcade::Vect](#) pos and size. Your [PictureBox](#) object giving as parameter must be set with the minimal size possible to prevent a possible partial display.

5.5.4.2 `scaleTextBox()`

```
void Arcade::Scale::scaleTextBox (
    const Arcade::Vect< double > & pos,
    Arcade::TextBox & textBox )
```

`Scale` main function for `TextBoxes`.

Modifies the `TextBox` giving as parameter and sets the position using the `Arcade::Vect` pos. Position can be set as double to allow a better vertical precision alignment for advanced graphical libs.



5.5.4.3 `setCentering()`

```
void Arcade::Scale::setCentering (
    const CENTERING & param )
```

`Scale` centering's setter.

Allows the user to define the desired alignment (only working on `PixelBox` object). By default, alignment is set to `NONE` which means that the `PixelBox` object will be set starting from the exact position set by the `scalePixelBox` method.

5.5.4.4 setWindowSize()

```
void Arcade::Scale::setWindowSize (
    const Arcade::Vect< size_t > & windowSize )
```

[Scale](#) window size's setter.

Allows the user to set the window size. If not set, the scaling class may just explode and create a black hole destroying our world. You probably don't want that.

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

- [/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.hpp](#)
- [/home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.cpp](#)

5.6 Arcade::Scoreboard Class Reference

[Scoreboard](#) class.

```
#include <Scoreboard.hpp>
```

Public Member Functions

- [Scoreboard](#) ()
Scoreboard class's constructor.
- [~Scoreboard](#) ()
Scoreboard class's destructor.
- void [setGameName](#) (const std::string &gameName)
Scoreboard game name's setter.
- void [setPlayerName](#) (const std::string &playerName)
Scoreboard player name's setter.
- bool [readScoreboard](#) ()
Scoreboard initializer.
- size_t [getLastPlayerScore](#) ()
Scoreboard last player score's getter.
- std::map< std::string, std::vector< std::string > > [getScoreboard](#) () const
Scoreboard getter.
- size_t [getScores](#) () const
Scoreboard current score's getter.
- void [addScores](#) (const size_t &)
Scoreboard setter.
- void [subScores](#) (const size_t &)
Scoreboard setter.
- void [resetScores](#) ()
Scoreboard reseter.

5.6.1 Detailed Description

[Scoreboard](#) class.

Class used to manage the scoreboard

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Scoreboard()

```
Arcade::Scoreboard::Scoreboard ( )
```

[Scoreboard](#) class's constructor.

Creates a new scoreboard class instance.

5.6.2.2 ~Scoreboard()

```
Arcade::Scoreboard::~~Scoreboard ( )
```

[Scoreboard](#) class's destructor.

5.6.3 Member Function Documentation

5.6.3.1 addScores()

```
void Arcade::Scoreboard::addScores (
    const size_t & points )
```

[Scoreboard](#) setter.

Add score to the current score.

5.6.3.2 getLastPlayerScore()

```
size_t Arcade::Scoreboard::getLastPlayerScore ( )
```

[Scoreboard](#) last player score's getter.

Returns

the score of the last player as a size_t

5.6.3.3 `getScoreboard()`

```
std::map< std::string, std::vector< std::string > > Arcade::Scoreboard::getScoreboard ( )  
const
```

[Scoreboard](#) getter.

Returns

all the scoreboard.

5.6.3.4 `getScores()`

```
size_t Arcade::Scoreboard::getScores ( ) const
```

[Scoreboard](#) current score's getter.

Returns

the current score.

5.6.3.5 `readScoreboard()`

```
bool Arcade::Scoreboard::readScoreboard ( )
```

[Scoreboard](#) initializer.

Reads the file to feed itself.

5.6.3.6 `resetScores()`

```
void Arcade::Scoreboard::resetScores ( )
```

[Scoreboard](#) reseter.

Reset the scores.

5.6.3.7 `setGameName()`

```
void Arcade::Scoreboard::setGameName (  
    const std::string & gameName )
```

[Scoreboard](#) game name's setter.

5.6.3.8 setPlayerName()

```
void Arcade::Scoreboard::setPlayerName (
    const std::string & playerName )
```

[Scoreboard](#) player name's setter.

5.6.3.9 subScores()

```
void Arcade::Scoreboard::subScores (
    const size_t & points )
```

[Scoreboard](#) setter.

Sub score to the current score.

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

- /home/thibrex/epitech/CPP/cpp_arcade/shared_header/[Scoreboard.hpp](#)
- /home/thibrex/epitech/CPP/cpp_arcade/shared_header/[Scoreboard.cpp](#)

5.7 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.*

5.7.1 Detailed Description

`TextBox` class.

Class used to represent a rectangle of text

5.7.2 Constructor & Destructor Documentation

5.7.2.1 TextBox()

```
Arcade::TextBox::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.

Parameters

<i>text</i>	: characters to be apply on the textBox
<i>pos</i>	: <code>Vect<size_t></code> 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.

5.7.2.2 `~TextBox()`

```
Arcade::TextBox::~~TextBox ( ) [default]
```

[PixelFormat](#) class's destructor.

5.7.3 Member Function Documentation

5.7.3.1 `getBackgroundColor()`

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

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

Returns

the `textBox`'s text's background color

5.7.3.2 `getColor()`

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

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

Returns

the `textBox`'s text's color

5.7.3.3 `getFontSize()`

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

[TextBox](#)'s font size's getter.

Returns

the font size

5.7.3.4 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.

5.7.3.5 getValue()

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

TextBox text's value's getter.

Returns

the value of the text within textBox

5.7.3.6 getX()

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

TextBox X offset's getter.

5.7.3.7 getY()

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

TextBox Y offset's getter.

5.7.3.8 setBackgroundColor()

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

TextBox's text background color's setter.

Parameters

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

5.7.3.9 setColor()

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

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

Parameters

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

5.7.3.10 setFontSize()

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

[TextBox](#)'s font size's setter.

Parameters

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

5.7.3.11 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>](#)

5.7.3.12 setValue()

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

Sets the textBox text's value.

Parameters

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

5.7.3.13 setX()

```
void Arcade::TextBox::setX (
    size_t x )
```

[TextBox](#) X offset's setter.

5.7.3.14 setY()

```
void Arcade::TextBox::setY (
    size_t y )
```

[TextBox](#) Y offset's setter.

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

- [/home/thibrex/epitech/CPP/cpp_arcade/shared_header/TextBox.hpp](#)
- [/home/thibrex/epitech/CPP/cpp_arcade/shared_header/TextBox.cpp](#)

5.8 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)`

5.8.1 Detailed Description

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

`Vect` class template.

Mainly used to store and manage 2 coordinates

5.8.2 Constructor & Destructor Documentation

5.8.2.1 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

5.8.3 Member Function Documentation

5.8.3.1 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

5.8.3.2 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

5.8.3.3 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](#)

5.8.3.4 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

5.8.3.5 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 Vect object to perform the multiplication with
--------------	----------------------------------------------------------------------

Returns

the object from which this function was called

5.8.3.6 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 T variable to perform the multiplication with
--------------	-----------------------------------------------------

Returns

the object from which this function was called

5.8.3.7 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 Vect object to perform the addition with
--------------	------------------------------------------------

Returns

a new object resulting from the addition of the Vect

5.8.3.8 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

5.8.3.9 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

5.8.3.10 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

5.8.3.11 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](#)

5.8.3.12 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

5.8.3.13 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

5.8.3.14 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

5.8.3.15 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

5.8.3.16 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

5.8.3.17 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

5.8.3.18 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

5.8.3.19 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

5.8.3.20 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
----------	--------------------

5.8.3.21 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

5.8.3.22 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/thibrex/epitech/CPP/cpp_arcade/shared_header/Vect.hpp

Chapter 6

File Documentation

6.1 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/Color.cpp File Reference

```
#include "Color.hpp"
```

Include dependency graph for Color.cpp:

6.2 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/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.

6.2.1 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.

6.3 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/IGameLib.hpp File Reference

Game libraries dedicated class interface.

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

Include dependency graph for IGameLib.hpp:

Classes

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

Namespaces

- [Arcade](#)
Arcade project namespace.

6.3.1 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.

6.4 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/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.

6.4.1 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.

6.5 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/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](#) {
[Arcade::NONE](#), [Arcade::A](#), [Arcade::B](#), [Arcade::C](#),
[Arcade::D](#), [Arcade::E](#), [Arcade::F](#), [Arcade::G](#),
[Arcade::H](#), [Arcade::I](#), [Arcade::J](#), [Arcade::K](#),
[Arcade::L](#), [Arcade::M](#), [Arcade::N](#), [Arcade::O](#),
[Arcade::P](#), [Arcade::Q](#), [Arcade::R](#), [Arcade::S](#),
[Arcade::T](#), [Arcade::U](#), [Arcade::V](#), [Arcade::W](#),
[Arcade::X](#), [Arcade::Y](#), [Arcade::Z](#), [Arcade::LEFT](#),
[Arcade::RIGHT](#), [Arcade::UP](#), [Arcade::DOWN](#), [Arcade::ENTER](#),
[Arcade::SPACE](#), [Arcade::DELETE](#), [Arcade::BACKSPACE](#), [Arcade::TAB](#),
[Arcade::ESC](#), [Arcade::MOUSELEFT](#), [Arcade::MOUSERIGHT](#) }

6.5.1 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.

6.6 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/PixelBox.cpp File Reference

```
#include "PixelBox.hpp"
```

Include dependency graph for PixelBox.cpp:

6.7 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/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::PixelBox](#)
PixelBox class.

Namespaces

- [Arcade](#)
Arcade project namespace.

6.7.1 Detailed Description

PixelBox 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.

6.8 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include "Scale.hpp"
```

Include dependency graph for Scale.cpp:

6.9 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scale.hpp File Reference

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

Include dependency graph for Scale.hpp: This graph shows which files directly or indirectly include this file:

Classes

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

Namespaces

- [Arcade](#)
Arcade project namespace.

6.10 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scoreboard.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <ios>
#include <sstream>
#include <regex>
#include "Scoreboard.hpp"
```

Include dependency graph for Scoreboard.cpp:

6.11 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/Scoreboard.hpp File Reference

```
#include <vector>
#include <map>
```

Include dependency graph for Scoreboard.hpp: This graph shows which files directly or indirectly include this file:

Classes

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

Namespaces

- [Arcade](#)
Arcade project namespace.

Macros

- `#define SCOREBOARD "scores"`

6.11.1 Macro Definition Documentation

6.11.1.1 SCOREBOARD

```
#define SCOREBOARD "scores"
```

6.12 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/TextBox.cpp File Reference

```
#include "TextBox.hpp"
```

Include dependency graph for TextBox.cpp:

6.13 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/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.

6.13.1 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.

6.14 /home/thibrex/epitech/CPP/cpp_arcade/shared_header/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.

6.14.1 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.

Index

/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Color.cpp, 51
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Color.hpp, 51
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/IGameLib.hpp, 52
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/IGraphicLib.hpp, 52
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Keys.hpp, 53
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/PixelBox.cpp, 54
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/PixelBox.hpp, 54
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Scale.cpp, 54
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Scale.hpp, 55
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Scoreboard.cpp, 55
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Scoreboard.hpp, 55
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/TextBox.cpp, 56
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/TextBox.hpp, 56
/home/thibrex/epitech/CPP/cpp_arcade/shared_↔
 header/Vect.hpp, 57
~IGameLib
 Arcade::IGameLib, 15
~IGraphicLib
 Arcade::IGraphicLib, 18
~PixelBox
 Arcade::PixelBox, 24
~Scale
 Arcade::Scale, 29
~Scoreboard
 Arcade::Scoreboard, 32
~TextBox
 Arcade::TextBox, 36

A
 Arcade, 8
addScores
 Arcade::Scoreboard, 32
applyEvent
 Arcade::IGameLib, 16
Arcade, 7
 A, 8
 B, 8

C, 8
D, 8
E, 8
F, 8
G, 8
H, 8
I, 8
J, 8
K, 8
Keys, 8
L, 8
M, 8
N, 8
O, 8
P, 8
Q, 8
R, 8
S, 8
T, 8
U, 8
UP, 8
V, 8
W, 8
X, 8
Y, 8
Z, 8
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::IGameLib, 15
 ~IGameLib, 15
 applyEvent, 16
 getName, 16
 getScore, 16
 init, 16
 refresh, 17
 stop, 17
 update, 17
Arcade::IGraphicLib, 18

- ~IGraphicLib, 18
- clearEvents, 19
- clearWindow, 19
- closeRenderer, 19
- drawPixelBox, 19
- drawText, 19
- getLastEvent, 20
- getMaxX, 20
- getMaxY, 20
- getName, 20
- getScreenSize, 21
- isOpen, 21
- openRenderer, 21
- pollEvents, 22
- refreshWindow, 22
- Arcade::PixelBox, 22
 - ~PixelBox, 24
 - getHeight, 24
 - getPixel, 24
 - getPixelArray, 24
 - getPos, 25
 - getSize, 25
 - getWidth, 25
 - getX, 25
 - getY, 25
 - PixelBox, 23
 - putPixel, 26
 - putRect, 26
 - setHeight, 26
 - setPos, 27
 - setSize, 27
 - setWidth, 27
 - setX, 27
 - setY, 27
- Arcade::Scale, 28
 - ~Scale, 29
 - CENTERING, 28
 - Scale, 29
 - scalePixelBox, 29
 - scaleTextBox, 29
 - setCentering, 30
 - setWindowSize, 30
- Arcade::Scoreboard, 31
 - ~Scoreboard, 32
 - addScores, 32
 - getLastPlayerScore, 32
 - getScoreboard, 32
 - getScores, 33
 - readScoreboard, 33
 - resetScores, 33
 - Scoreboard, 32
 - setGameName, 33
 - setPlayerName, 33
 - subScores, 34
- Arcade::TextBox, 34
 - ~TextBox, 36
 - getBackgroundColor, 36
 - getColor, 36
 - getFontSize, 36
 - getPos, 36
 - getValue, 37
 - getX, 37
 - getY, 37
 - setBackgroundColor, 37
 - setColor, 38
 - setFontSize, 38
 - setPos, 38
 - setValue, 38
 - setX, 39
 - setY, 39
 - TextBox, 35
- Arcade::Vect
 - getX, 41
 - getY, 41
 - operator*, 41, 42
 - operator*=: 42
 - operator+, 43
 - operator+=, 43, 44
 - operator-, 44, 45
 - operator-=, 45
 - operator/, 46
 - operator/=, 46, 47
 - operator==, 47
 - setXY, 48
 - setX, 48
 - setY, 48
 - Vect, 40
- Arcade::Vect< T >, 39
- B
 - Arcade, 8
- C
 - Arcade, 8
 - CENTERING
 - Arcade::Scale, 28
 - clearEvents
 - Arcade::IGraphicLib, 19
 - clearWindow
 - Arcade::IGraphicLib, 19
 - closeRenderer
 - Arcade::IGraphicLib, 19
 - Color
 - Arcade::Color, 12
- D
 - Arcade, 8
 - drawPixelBox
 - Arcade::IGraphicLib, 19
 - drawText
 - Arcade::IGraphicLib, 19
- E
 - Arcade, 8
- F
 - Arcade, 8

G

- Arcade, [8](#)
- getAlpha
 - Arcade::Color, [12](#)
- getBackgroundColor
 - Arcade::TextBox, [36](#)
- getBlue
 - Arcade::Color, [12](#)
- getColor
 - Arcade::TextBox, [36](#)
- getFontSize
 - Arcade::TextBox, [36](#)
- getGreen
 - Arcade::Color, [12](#)
- getHeight
 - Arcade::PictureBox, [24](#)
- getLastEvent
 - Arcade::IGraphicLib, [20](#)
- getLastPlayerScore
 - Arcade::Scoreboard, [32](#)
- getMaxX
 - Arcade::IGraphicLib, [20](#)
- getMaxY
 - Arcade::IGraphicLib, [20](#)
- getName
 - Arcade::IGameLib, [16](#)
 - Arcade::IGraphicLib, [20](#)
- getPixel
 - Arcade::PictureBox, [24](#)
- getPixelArray
 - Arcade::PictureBox, [24](#)
- getPos
 - Arcade::PictureBox, [25](#)
 - Arcade::TextBox, [36](#)
- getRed
 - Arcade::Color, [13](#)
- getScore
 - Arcade::IGameLib, [16](#)
- getScoreboard
 - Arcade::Scoreboard, [32](#)
- getScores
 - Arcade::Scoreboard, [33](#)
- getScreenSize
 - Arcade::IGraphicLib, [21](#)
- getSize
 - Arcade::PictureBox, [25](#)
- getValue
 - Arcade::TextBox, [37](#)
- getWidth
 - Arcade::PictureBox, [25](#)
- getX
 - Arcade::PictureBox, [25](#)
 - Arcade::TextBox, [37](#)
 - Arcade::Vect, [41](#)
- getY
 - Arcade::PictureBox, [25](#)
 - Arcade::TextBox, [37](#)
 - Arcade::Vect, [41](#)

H

- Arcade, [8](#)

I

- Arcade, [8](#)
- init
 - Arcade::IGameLib, [16](#)
- isOpen
 - Arcade::IGraphicLib, [21](#)

J

- Arcade, [8](#)

K

- Arcade, [8](#)

Keys

- Arcade, [8](#)

L

- Arcade, [8](#)

M

- Arcade, [8](#)

N

- Arcade, [8](#)

O

- Arcade, [8](#)
- openRenderer
 - Arcade::IGraphicLib, [21](#)
- operator unsigned char *
 - Arcade::Color, [13](#)
- operator*
 - Arcade::Vect, [41](#), [42](#)
- operator*=
 - Arcade::Vect, [42](#)
- operator+
 - Arcade::Vect, [43](#)
- operator+=
 - Arcade::Vect, [43](#), [44](#)
- operator-
 - Arcade::Vect, [44](#), [45](#)
- operator-=
 - Arcade::Vect, [45](#)
- operator/
 - Arcade::Vect, [46](#)
- operator/=
 - Arcade::Vect, [46](#), [47](#)
- operator==
 - Arcade::Color, [13](#)
 - Arcade::Vect, [47](#)

P

- Arcade, [8](#)

PictureBox

- Arcade::PictureBox, [23](#)

pollEvents

- Arcade::IGraphicLib, [22](#)

putPixel

Arcade::PixelFormat, 26
 putRect
 Arcade::PixelFormat, 26
 Q
 Arcade, 8
 R
 Arcade, 8
 readScoreboard
 Arcade::Scoreboard, 33
 refresh
 Arcade::IGameLib, 17
 refreshWindow
 Arcade::IGraphicLib, 22
 resetScores
 Arcade::Scoreboard, 33
 S
 Arcade, 8
 SCOREBOARD
 Scoreboard.hpp, 56
 Scale
 Arcade::Scale, 29
 scalePixelFormat
 Arcade::Scale, 29
 scaleTextBox
 Arcade::Scale, 29
 Scoreboard
 Arcade::Scoreboard, 32
 Scoreboard.hpp
 SCOREBOARD, 56
 setAlpha
 Arcade::Color, 14
 setBackgroundColor
 Arcade::TextBox, 37
 setBlue
 Arcade::Color, 14
 setCentering
 Arcade::Scale, 30
 setColor
 Arcade::Color, 14
 Arcade::TextBox, 38
 setFontSize
 Arcade::TextBox, 38
 setGameName
 Arcade::Scoreboard, 33
 setGreen
 Arcade::Color, 14
 setHeight
 Arcade::PixelFormat, 26
 setPlayerName
 Arcade::Scoreboard, 33
 setPos
 Arcade::PixelFormat, 27
 Arcade::TextBox, 38
 setRed
 Arcade::Color, 14
 setSize
 Arcade::PixelFormat, 27
 setValue
 Arcade::TextBox, 38
 setWidth
 Arcade::PixelFormat, 27
 setWindowSize
 Arcade::Scale, 30
 setXY
 Arcade::Vect, 48
 setX
 Arcade::PixelFormat, 27
 Arcade::TextBox, 39
 Arcade::Vect, 48
 setY
 Arcade::PixelFormat, 27
 Arcade::TextBox, 39
 Arcade::Vect, 48
 stop
 Arcade::IGameLib, 17
 subScores
 Arcade::Scoreboard, 34
 T
 Arcade, 8
 TextBox
 Arcade::TextBox, 35
 U
 Arcade, 8
 UP
 Arcade, 8
 update
 Arcade::IGameLib, 17
 V
 Arcade, 8
 Vect
 Arcade::Vect, 40
 W
 Arcade, 8
 X
 Arcade, 8
 Y
 Arcade, 8
 Z
 Arcade, 8