

## Arcade

Generated by Doxygen 1.8.20



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Arcade::AnimBlueGhost Class Reference	5
3.2 Arcade::AnimPacman Class Reference	5
3.3 Arcade::AnimPinkGhost Class Reference	6
3.4 Arcade::AnimRedGhost Class Reference	6
3.5 Arcade::AnimYellowGhost Class Reference	7
3.6 Arcade::ArcadeNcurses Class Reference	7
3.6.1 Member Function Documentation	7
3.6.1.1 getEvents()	8
3.6.1.2 getLibName()	8
3.7 Arcade::ArcadeSfml Class Reference	8
3.7.1 Member Function Documentation	9
3.7.1.1 getEvents()	9
3.7.1.2 getLibName()	9
3.8 Arcade::Component Class Reference	9
3.8.1 Member Function Documentation	10
3.8.1.1 getFile()	11
3.8.1.2 getFontSize()	11
3.8.1.3 getHeight()	11
3.8.1.4 getId()	11
3.8.1.5 getRect()	12
3.8.1.6 getText()	12
3.8.1.7 getType()	12
3.8.1.8 getWidth()	13
3.8.1.9 getX()	13
3.8.1.10 getY()	13
3.8.1.11 setHeight()	13
3.8.1.12 setRect()	14
3.8.1.13 setText()	14
3.8.1.14 setWidth()	14
3.8.1.15 setX()	15
3.8.1.16 setY()	15
3.9 Arcade::ComponentError Class Reference	15
3.10 Arcade::ComponentObject Class Reference	16
3.10.1 Member Function Documentation	17
3.10.1.1 getHeight()	17
3.10.1.2 getRect()	18

3.10.1.3	<a href="#">getWidth()</a>	18
3.10.1.4	<a href="#">getX()</a>	18
3.10.1.5	<a href="#">getY()</a>	18
3.10.1.6	<a href="#">setHeight()</a>	18
3.10.1.7	<a href="#">setRect()</a>	19
3.10.1.8	<a href="#">setWidth()</a>	19
3.11	<a href="#">Arcade::ComponentSound Class Reference</a>	19
3.12	<a href="#">Arcade::ComponentText Class Reference</a>	20
3.12.1	<a href="#">Member Function Documentation</a>	21
3.12.1.1	<a href="#">getFontSize()</a>	21
3.12.1.2	<a href="#">getHeight()</a>	21
3.12.1.3	<a href="#">getRect()</a>	22
3.12.1.4	<a href="#">getText()</a>	22
3.12.1.5	<a href="#">getWidth()</a>	22
3.12.1.6	<a href="#">getX()</a>	23
3.12.1.7	<a href="#">getY()</a>	23
3.12.1.8	<a href="#">setHeight()</a>	23
3.12.1.9	<a href="#">setRect()</a>	23
3.12.1.10	<a href="#">setText()</a>	24
3.12.1.11	<a href="#">setWidth()</a>	24
3.13	<a href="#">Arcade::Core Class Reference</a>	24
3.14	<a href="#">Display Class Reference</a>	25
3.14.1	<a href="#">Member Function Documentation</a>	25
3.14.1.1	<a href="#">display()</a>	25
3.14.1.2	<a href="#">getEvents()</a>	26
3.14.1.3	<a href="#">getLibName()</a>	26
3.15	<a href="#">Display::DisplayException Class Reference</a>	26
3.16	<a href="#">Arcade::Event Class Reference</a>	27
3.16.1	<a href="#">Member Function Documentation</a>	28
3.16.1.1	<a href="#">getKey()</a>	28
3.16.1.2	<a href="#">getMousePos()</a>	28
3.16.1.3	<a href="#">setKey()</a>	28
3.17	<a href="#">Event Class Reference</a>	29
3.17.1	<a href="#">Member Function Documentation</a>	30
3.17.1.1	<a href="#">getData()</a> [1/3]	30
3.17.1.2	<a href="#">getData()</a> [2/3]	30
3.17.1.3	<a href="#">getData()</a> [3/3]	30
3.17.1.4	<a href="#">getKey()</a> [1/3]	31
3.17.1.5	<a href="#">getKey()</a> [2/3]	31
3.17.1.6	<a href="#">getKey()</a> [3/3]	31
3.17.1.7	<a href="#">getMousePos()</a> [1/3]	31
3.17.1.8	<a href="#">getMousePos()</a> [2/3]	32

3.17.1.9 <a href="#">getMousePos()</a> [3/3]	32
3.17.1.10 <a href="#">setKey()</a> [1/3]	32
3.17.1.11 <a href="#">setKey()</a> [2/3]	33
3.17.1.12 <a href="#">setKey()</a> [3/3]	33
3.18 <a href="#">EventNormalizer Class Reference</a>	33
3.19 <a href="#">Arcade::Game Class Reference</a>	34
3.19.1 <a href="#">Member Function Documentation</a>	34
3.19.1.1 <a href="#">getComponents()</a>	34
3.19.1.2 <a href="#">getEvent()</a>	35
3.19.1.3 <a href="#">getGameName()</a>	35
3.19.1.4 <a href="#">getPlayerName()</a>	35
3.19.1.5 <a href="#">sendEvents()</a>	35
3.19.1.6 <a href="#">setPlayerName()</a>	36
3.20 <a href="#">Arcade::IComponent Class Reference</a>	36
3.20.1 <a href="#">Detailed Description</a>	38
3.20.2 <a href="#">Member Function Documentation</a>	38
3.20.2.1 <a href="#">getFile()</a>	38
3.20.2.2 <a href="#">getFontSize()</a>	38
3.20.2.3 <a href="#">getHeight()</a>	39
3.20.2.4 <a href="#">getId()</a>	39
3.20.2.5 <a href="#">getRect()</a>	39
3.20.2.6 <a href="#">getText()</a>	40
3.20.2.7 <a href="#">getType()</a>	40
3.20.2.8 <a href="#">getWidth()</a>	40
3.20.2.9 <a href="#">getX()</a>	41
3.20.2.10 <a href="#">getY()</a>	41
3.20.2.11 <a href="#">setHeight()</a>	41
3.20.2.12 <a href="#">setRect()</a>	41
3.20.2.13 <a href="#">setText()</a>	42
3.20.2.14 <a href="#">setWidth()</a>	42
3.20.2.15 <a href="#">setX()</a>	42
3.20.2.16 <a href="#">setY()</a>	43
3.21 <a href="#">Arcade::IDisplay Class Reference</a>	43
3.21.1 <a href="#">Detailed Description</a>	44
3.21.2 <a href="#">Member Function Documentation</a>	44
3.21.2.1 <a href="#">display()</a>	44
3.21.2.2 <a href="#">getEvents()</a>	44
3.21.2.3 <a href="#">getLibName()</a>	45
3.22 <a href="#">Arcade::IEvent Class Reference</a>	45
3.22.1 <a href="#">Detailed Description</a>	45
3.22.2 <a href="#">Member Function Documentation</a>	45
3.22.2.1 <a href="#">getData()</a>	46

3.22.2.2 getKey()	46
3.22.2.3 getMousePos()	46
3.22.2.4 setKey()	46
3.23 Arcade::IGame Class Reference	47
3.23.1 Detailed Description	48
3.23.2 Member Function Documentation	48
3.23.2.1 getComponents()	48
3.23.2.2 getEvent()	48
3.23.2.3 getGameName()	48
3.23.2.4 getPlayerName()	49
3.23.2.5 sendEvents()	49
3.23.2.6 setPlayerName()	49
3.24 Arcade::IPacmanAnim Class Reference	50
3.25 Arcade::IPacmanComponent Class Reference	50
3.25.1 Member Function Documentation	51
3.25.1.1 getFile()	52
3.25.1.2 getFontSize()	52
3.25.1.3 getHeight()	52
3.25.1.4 getId()	53
3.25.1.5 getRect()	53
3.25.1.6 getText()	53
3.25.1.7 getType()	54
3.25.1.8 getWidth()	54
3.25.1.9 getX()	54
3.25.1.10 getY()	55
3.25.1.11 setHeight()	55
3.25.1.12 setRect()	55
3.25.1.13 setText()	56
3.25.1.14 setWidth()	56
3.25.1.15 setX()	56
3.25.1.16 setY()	57
3.26 Arcade::IPacmanMap Class Reference	57
3.27 Arcade::ISfmlComponent Class Reference	57
3.28 Arcade::ISfmlData Class Reference	58
3.29 Arcade::KeyEvent Class Reference	58
3.30 Arcade::LibHandler Class Reference	59
3.30.1 Detailed Description	59
3.30.2 Member Function Documentation	60
3.30.2.1 getDisplay()	60
3.30.2.2 getGame()	60
3.30.2.3 loadLibrary()	60
3.31 Arcade::LibHandler::LibraryException Class Reference	60

3.31.1 Detailed Description	61
3.32 Arcade::MapError Class Reference	61
3.33 Arcade::MouseEvent Class Reference	62
3.34 Arcade::NcursesEvent Class Reference	62
3.34.1 Member Function Documentation	63
3.34.1.1 getData()	63
3.34.1.2 getKey()	63
3.34.1.3 getMousePos()	64
3.34.1.4 setKey()	64
3.35 Arcade::NibblerComponent Class Reference	65
3.35.1 Member Function Documentation	66
3.35.1.1 getFile()	66
3.35.1.2 getFontSize()	67
3.35.1.3 getHeight()	67
3.35.1.4 getId()	67
3.35.1.5 getRect()	67
3.35.1.6 getText()	68
3.35.1.7 getType()	68
3.35.1.8 getWidth()	68
3.35.1.9 getX()	69
3.35.1.10 getY()	69
3.35.1.11 setHeight()	69
3.35.1.12 setRect()	69
3.35.1.13 setText()	70
3.35.1.14 setWidth()	70
3.35.1.15 setX()	70
3.35.1.16 setY()	71
3.36 Arcade::NibblerGame Class Reference	71
3.36.1 Member Function Documentation	72
3.36.1.1 getComponents()	72
3.36.1.2 getEvent()	72
3.36.1.3 getGameName()	73
3.36.1.4 getPlayerName()	73
3.36.1.5 sendEvents()	73
3.36.1.6 setPlayerName()	73
3.37 Arcade::NibblerGame::NibblerGameException Class Reference	74
3.38 Arcade::ObjectBigPacGum Class Reference	74
3.39 Arcade::ObjectBlueGhost Class Reference	75
3.40 Arcade::ObjectLitPacGum Class Reference	76
3.41 Arcade::ObjectMap Class Reference	76
3.42 Arcade::ObjectPacman Class Reference	77
3.43 Arcade::ObjectPinkGhost Class Reference	77

3.44 Arcade::ObjectRedGhost Class Reference . . . . .	78
3.45 Arcade::ObjectYellowGhost Class Reference . . . . .	79
3.46 Arcade::PacmanAnim Class Reference . . . . .	79
3.47 Arcade::PacmanComponent Class Reference . . . . .	80
3.47.1 Member Function Documentation . . . . .	82
3.47.1.1 getFile() . . . . .	82
3.47.1.2 getFontSize() . . . . .	82
3.47.1.3 getHeight() . . . . .	82
3.47.1.4 getId() . . . . .	83
3.47.1.5 getRect() . . . . .	83
3.47.1.6 getText() . . . . .	83
3.47.1.7 getType() . . . . .	84
3.47.1.8 getWidth() . . . . .	84
3.47.1.9 getX() . . . . .	84
3.47.1.10 getY() . . . . .	84
3.47.1.11 setHeight() . . . . .	84
3.47.1.12 setRect() . . . . .	85
3.47.1.13 setText() . . . . .	85
3.47.1.14 setWidth() . . . . .	85
3.47.1.15 setX() . . . . .	86
3.47.1.16 setY() . . . . .	86
3.48 Arcade::PacmanError Class Reference . . . . .	86
3.49 Arcade::PacmanEvent Class Reference . . . . .	87
3.49.1 Member Function Documentation . . . . .	88
3.49.1.1 getData() . . . . .	88
3.49.1.2 getKey() . . . . .	88
3.49.1.3 getMousePos() . . . . .	88
3.49.1.4 setKey() . . . . .	88
3.50 Arcade::PacmanGame Class Reference . . . . .	89
3.50.1 Member Function Documentation . . . . .	89
3.50.1.1 getComponents() . . . . .	90
3.50.1.2 getEvent() . . . . .	90
3.50.1.3 getGameName() . . . . .	90
3.50.1.4 getPlayerName() . . . . .	90
3.50.1.5 setPlayerName() . . . . .	90
3.51 Arcade::PacmanMap Class Reference . . . . .	91
3.52 Arcade::PacmanScore Class Reference . . . . .	92
3.53 Arcade::Pos Struct Reference . . . . .	92
3.53.1 Detailed Description . . . . .	92
3.54 Arcade::PacmanMap::Pos Struct Reference . . . . .	92
3.55 Arcade::Rect Struct Reference . . . . .	93
3.55.1 Detailed Description . . . . .	93



3.56 Arcade::ScoreBoard Class Reference . . . . .	93
3.57 Arcade::SfmlComponent Class Reference . . . . .	93
3.58 Arcade::SfmlComponentObject Class Reference . . . . .	94
3.59 Arcade::SfmlComponentSound Class Reference . . . . .	95
3.60 Arcade::SfmlComponentText Class Reference . . . . .	96
3.61 Arcade::SfmlData Class Reference . . . . .	96
3.62 Arcade::SfmlDataFont Class Reference . . . . .	97
3.63 Arcade::SfmlDataSound Class Reference . . . . .	97
3.64 Arcade::SfmlDataTexture Class Reference . . . . .	98
3.65 Arcade::SfmlEvent Class Reference . . . . .	98
3.65.1 Member Function Documentation . . . . .	99
3.65.1.1 getData() . . . . .	99
3.65.1.2 getKey() . . . . .	99
3.65.1.3 getMousePos() . . . . .	100
3.65.1.4 setKey() . . . . .	100
3.66 Arcade::Snake Class Reference . . . . .	101
3.66.1 Member Function Documentation . . . . .	102
3.66.1.1 setX() . . . . .	102
3.66.1.2 setY() . . . . .	102
3.67 Arcade::PacmanMap::SPos Struct Reference . . . . .	102
3.68 Arcade::TextGameOver Class Reference . . . . .	103
3.69 Arcade::TextReset Class Reference . . . . .	103
3.70 Arcade::TextScore Class Reference . . . . .	104
3.71 Arcade::TextScoreNb Class Reference . . . . .	104
3.72 Arcade::TextWin Class Reference . . . . .	105
3.73 Arcade::Timer Class Reference . . . . .	105
<b>Index</b>	<b>107</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

Arcade::Core . . . . .	24
EventNormalizer . . . . .	33
exception	
Arcade::LibHandler::LibraryException . . . . .	60
Arcade::NibblerGame::NibblerGameException . . . . .	74
Arcade::PacmanError . . . . .	86
Arcade::ComponentError . . . . .	15
Arcade::MapError . . . . .	61
Display::DisplayException . . . . .	26
Arcade::IComponent . . . . .	36
Arcade::Component . . . . .	9
Arcade::IPacmanComponent . . . . .	50
Arcade::PacmanComponent . . . . .	80
Arcade::ComponentObject . . . . .	16
Arcade::ObjectBigPacGum . . . . .	74
Arcade::ObjectBlueGhost . . . . .	75
Arcade::ObjectLitPacGum . . . . .	76
Arcade::ObjectMap . . . . .	76
Arcade::ObjectPacman . . . . .	77
Arcade::ObjectPinkGhost . . . . .	77
Arcade::ObjectRedGhost . . . . .	78
Arcade::ObjectYellowGhost . . . . .	79
Arcade::ComponentSound . . . . .	19
Arcade::ComponentText . . . . .	20
Arcade::TextGameOver . . . . .	103
Arcade::TextReset . . . . .	103
Arcade::TextScore . . . . .	104
Arcade::TextScoreNb . . . . .	104
Arcade::TextWin . . . . .	105
Arcade::NibblerComponent . . . . .	65
Arcade::Snake . . . . .	101
Arcade::IDisplay . . . . .	43
Arcade::ArcadeNcurses . . . . .	7
Arcade::ArcadeSfml . . . . .	8

Display . . . . .	25
Arcade::IEvent . . . . .	45
Arcade::Event . . . . .	27
Arcade::KeyEvent . . . . .	58
Arcade::MouseEvent . . . . .	62
Arcade::NcursesEvent . . . . .	62
Arcade::PacmanEvent . . . . .	87
Arcade::SfmlEvent . . . . .	98
Event . . . . .	29
Arcade::IGame . . . . .	47
Arcade::Game . . . . .	34
Arcade::NibblerGame . . . . .	71
Arcade::PacmanGame . . . . .	89
Arcade::IPacmanAnim . . . . .	50
Arcade::PacmanAnim . . . . .	79
Arcade::AnimBlueGhost . . . . .	5
Arcade::AnimPacman . . . . .	5
Arcade::AnimPinkGhost . . . . .	6
Arcade::AnimRedGhost . . . . .	6
Arcade::AnimYellowGhost . . . . .	7
Arcade::IPacmanMap . . . . .	57
Arcade::ISfmlComponent . . . . .	57
Arcade::SfmlComponent . . . . .	93
Arcade::SfmlComponentObject . . . . .	94
Arcade::SfmlComponentSound . . . . .	95
Arcade::SfmlComponentText . . . . .	96
Arcade::ISfmlData . . . . .	58
Arcade::SfmlData . . . . .	96
Arcade::SfmlDataFont . . . . .	97
Arcade::SfmlDataSound . . . . .	97
Arcade::SfmlDataTexture . . . . .	98
Arcade::LibHandler . . . . .	59
Arcade::PacmanMap . . . . .	91
Arcade::PacmanScore . . . . .	92
Arcade::Pos . . . . .	92
Arcade::PacmanMap::Pos . . . . .	92
Arcade::Rect . . . . .	93
Arcade::ScoreBoard . . . . .	93
Arcade::PacmanMap::SPos . . . . .	102
Arcade::Timer . . . . .	105

## Chapter 2

# Class Index

### 2.1 Class List

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

Arcade::AnimBlueGhost	5
Arcade::AnimPacman	5
Arcade::AnimPinkGhost	6
Arcade::AnimRedGhost	6
Arcade::AnimYellowGhost	7
Arcade::ArcadeNcurses	7
Arcade::ArcadeSfml	8
Arcade::Component	9
Arcade::ComponentError	15
Arcade::ComponentObject	16
Arcade::ComponentSound	19
Arcade::ComponentText	20
Arcade::Core	24
Display	25
Display::DisplayException	26
Arcade::Event	27
Event	29
EventNormalizer	33
Arcade::Game	34
Arcade::IComponent	
The <a href="#">IComponent</a> interface represents a game entity that can either be displayed or played as a sound	36
Arcade::IDisplay	
This interface defines how to interact with graphic libraries. It represents the IO components. <a href="#">IDisplay</a> 's purpose is to poll events (Inputs), and to draw components on the screen / play sounds (Outputs)	43
Arcade::IEvent	
Represents an event polled by the graphic libraries	45
Arcade::IGame	
<a href="#">IGame</a> represents a game loaded into the core. Its methods allow to get game information, which can be passed down to <a href="#">IDisplay</a> by the core	47
Arcade::IPacmanAnim	50
Arcade::IPacmanComponent	50
Arcade::IPacmanMap	57
Arcade::ISfmlComponent	57

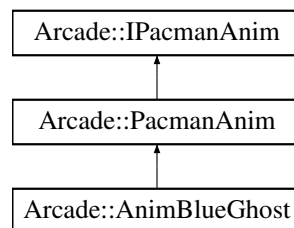
<a href="#">Arcade::ISfmlData</a>	58
<a href="#">Arcade::KeyEvent</a>	58
<a href="#">Arcade::LibHandler</a>	
Encapsulates calls to dlopen, dlsym, dLError and dlclose. Use it to automatically handle library changes	59
<a href="#">Arcade::LibHandler::LibraryException</a>	
Used to throw errors with lib handling	60
<a href="#">Arcade::MapError</a>	61
<a href="#">Arcade::MouseEvent</a>	62
<a href="#">Arcade::NcursesEvent</a>	62
<a href="#">Arcade::NibblerComponent</a>	65
<a href="#">Arcade::NibblerGame</a>	71
<a href="#">Arcade::NibblerGame::NibblerGameException</a>	74
<a href="#">Arcade::ObjectBigPacGum</a>	74
<a href="#">Arcade::ObjectBlueGhost</a>	75
<a href="#">Arcade::ObjectLitPacGum</a>	76
<a href="#">Arcade::ObjectMap</a>	76
<a href="#">Arcade::ObjectPacman</a>	77
<a href="#">Arcade::ObjectPinkGhost</a>	77
<a href="#">Arcade::ObjectRedGhost</a>	78
<a href="#">Arcade::ObjectYellowGhost</a>	79
<a href="#">Arcade::PacmanAnim</a>	79
<a href="#">Arcade::PacmanComponent</a>	80
<a href="#">Arcade::PacmanError</a>	86
<a href="#">Arcade::PacmanEvent</a>	87
<a href="#">Arcade::PacmanGame</a>	89
<a href="#">Arcade::PacmanMap</a>	91
<a href="#">Arcade::PacmanScore</a>	92
<a href="#">Arcade::Pos</a>	
Used for mouse position	92
<a href="#">Arcade::PacmanMap::Pos</a>	92
<a href="#">Arcade::Rect</a>	
The Rect struct give information on a sprite	93
<a href="#">Arcade::ScoreBoard</a>	93
<a href="#">Arcade::SfmlComponent</a>	93
<a href="#">Arcade::SfmlComponentObject</a>	94
<a href="#">Arcade::SfmlComponentSound</a>	95
<a href="#">Arcade::SfmlComponentText</a>	96
<a href="#">Arcade::SfmlData</a>	96
<a href="#">Arcade::SfmlDataFont</a>	97
<a href="#">Arcade::SfmlDataSound</a>	97
<a href="#">Arcade::SfmlDataTexture</a>	98
<a href="#">Arcade::SfmlEvent</a>	98
<a href="#">Arcade::Snake</a>	101
<a href="#">Arcade::PacmanMap::SPos</a>	102
<a href="#">Arcade::TextGameOver</a>	103
<a href="#">Arcade::TextReset</a>	103
<a href="#">Arcade::TextScore</a>	104
<a href="#">Arcade::TextScoreNb</a>	104
<a href="#">Arcade::TextWin</a>	105
<a href="#">Arcade::Timer</a>	105

## Chapter 3

# Class Documentation

### 3.1 Arcade::AnimBlueGhost Class Reference

Inheritance diagram for Arcade::AnimBlueGhost:



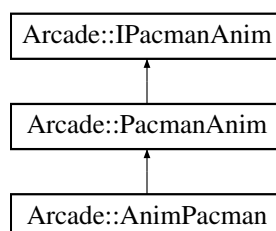
#### Additional Inherited Members

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

- games/pacman/include/PacmanAnim/AnimBlueGhost.hpp
- games/pacman/src/PacmanAnim/AnimBlueGhost.cpp

### 3.2 Arcade::AnimPacman Class Reference

Inheritance diagram for Arcade::AnimPacman:



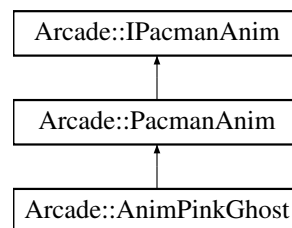
## Additional Inherited Members

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

- games/pacman/include/PacmanAnim/AnimPacman.hpp
- games/pacman/src/PacmanAnim/AnimPacman.cpp

## 3.3 Arcade::AnimPinkGhost Class Reference

Inheritance diagram for Arcade::AnimPinkGhost:



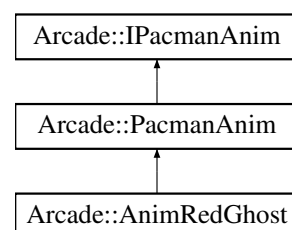
## Additional Inherited Members

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

- games/pacman/include/PacmanAnim/AnimPinkGhost.hpp
- games/pacman/src/PacmanAnim/AnimPinkGhost.cpp

## 3.4 Arcade::AnimRedGhost Class Reference

Inheritance diagram for Arcade::AnimRedGhost:



## Additional Inherited Members

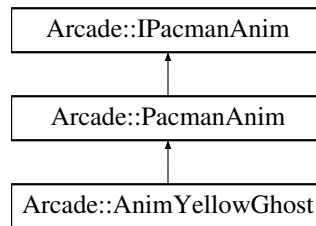
The documentation for this class was generated from the following files:

- games/pacman/include/PacmanAnim/AnimRedGhost.hpp
- games/pacman/src/PacmanAnim/AnimRedGhost.cpp



## 3.5 Arcade::AnimYellowGhost Class Reference

Inheritance diagram for Arcade::AnimYellowGhost:



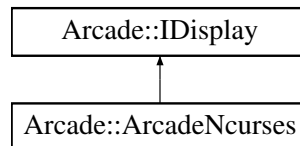
### Additional Inherited Members

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

- games/pacman/include/PacmanAnim/AnimYellowGhost.hpp
- games/pacman/src/PacmanAnim/AnimYellowGhost.cpp

## 3.6 Arcade::ArcadeNcurses Class Reference

Inheritance diagram for Arcade::ArcadeNcurses:



### Public Member Functions

- void **init** ()  
*Initialize the library.*
- void **stop** ()  
*Stop the library.*
- std::string **getLibName** () const  
*Get the Library Name.*
- void **display** (ICompUPtrVect &components)
- IEventUPtrVect & **getEvents** ()  
*Get the Events polled by the library.*
- void **clear** ()  
*Clear graphicals objects list.*

### 3.6.1 Member Function Documentation

### 3.6.1.1 `getEvents()`

```
Arcade::IEventUPtrVect & Arcade::ArcadeNcurses::getEvents ( ) [virtual]
```

Get the Events polled by the library.

#### Returns

`std::vector<std::unique_ptr<IEvent>>&`

Implements [Arcade::IDisplay](#).

### 3.6.1.2 `getLibName()`

```
std::string Arcade::ArcadeNcurses::getLibName ( ) const [virtual]
```

Get the Library Name.

#### Returns

`std::string`

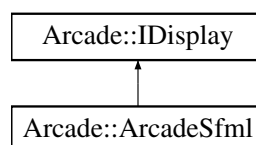
Implements [Arcade::IDisplay](#).

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

- `graphics/ncurses/include/ArcadeNcurses/ArcadeNcurses.hpp`
- `graphics/ncurses/src/ArcadeNcurses/ArcadeNcurses.cpp`

## 3.7 `Arcade::ArcadeSfml` Class Reference

Inheritance diagram for `Arcade::ArcadeSfml`:



### Public Member Functions

- void `init` ()  
*Initialize the library.*
- void `stop` ()  
*Stop the library.*
- `std::string` `getLibName` () const  
*Get the Library Name.*
- void **`display`** (ICompUPtrVect &components)
- IEventUPtrVect & `getEvents` ()  
*Get the Events polled by the library.*
- void `clear` ()  
*Clear graphicals objects list.*

### 3.7.1 Member Function Documentation

#### 3.7.1.1 getEvents()

```
Arcade::IEventUPtrVect & Arcade::ArcadeSfml::getEvents ( ) [virtual]
```

Get the Events polled by the library.

##### Returns

`std::vector<std::unique_ptr<IEvent>>&`

Implements [Arcade::IDisplay](#).

#### 3.7.1.2 getLibName()

```
std::string Arcade::ArcadeSfml::getLibName ( ) const [virtual]
```

Get the Library Name.

##### Returns

`std::string`

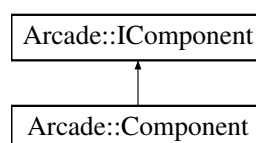
Implements [Arcade::IDisplay](#).

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

- `graphics/sfml/include/ArcadeSfml/ArcadeSfml.hpp`
- `graphics/sfml/src/ArcadeSfml/ArcadeSfml.cpp`

## 3.8 Arcade::Component Class Reference

Inheritance diagram for Arcade::Component:



## Public Member Functions

- **Component** (int id, Type type, std::string file, [Rect](#) rect, int x, int y, int w, int h)
- **Component** (int id, std::string text, std::string file, [Rect](#) rect, int x, int y, int w, int h, int fontSize)
- std::size\_t [getId](#) () const  
*Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.*
- Type [getType](#) () const  
*Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)*
- std::string [getFile](#) () const  
*Get file associated with component Use it to create textures, etc.*
- int [getX](#) () const  
*Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- int [getY](#) () const  
*Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- void [setX](#) (std::size\_t x)  
*Set the X value of the [Component](#).*
- void [setY](#) (std::size\_t y)  
*Set the Y value of the [Component](#).*
- int [getWidth](#) () const  
*Get width of the OBJECT/TEXT [Component](#).*
- int [getHeight](#) () const  
*Get height of the OBJECT/TEXT [Component](#).*
- void [setWidth](#) (const int width)  
*Set the X value of the [Component](#).*
- void [setHeight](#) (const int height)  
*Set the Y value of the [Component](#).*
- [Rect](#) [getRect](#) () const  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- void [setRect](#) ([Rect](#) const rect)  
*Set the [Rect](#) of the OBJECT [Component](#).*
- std::string [getText](#) () const  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- void [setText](#) (std::string const text)  
*Set the text of the [Component](#).*
- int [getFontSize](#) () const  
*Get the font size of the TEXT [Component](#).*

## Additional Inherited Members

### 3.8.1 Member Function Documentation

### 3.8.1.1 getFile()

```
std::string Arcade::Component::getFile ( ) const [virtual]
```

Get file associated with component Use it to create textures, etc.

#### Returns

std::string

Implements [Arcade::IComponent](#).

### 3.8.1.2 getFontSize()

```
int Arcade::Component::getFontSize ( ) const [virtual]
```

Get the font size of the TEXT [Component](#).

#### Returns

int

Implements [Arcade::IComponent](#).

### 3.8.1.3 getHeight()

```
int Arcade::Component::getHeight ( ) const [virtual]
```

Get height of the OBJECT/TEXT [Component](#).

#### Returns

int

Implements [Arcade::IComponent](#).

### 3.8.1.4 getId()

```
std::size_t Arcade::Component::getId ( ) const [virtual]
```

Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.

#### Returns

std::size\_t

Implements [Arcade::IComponent](#).

### 3.8.1.5 getRect()

```
Rect Arcade::Component::getRect ( ) const [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

Returns

Rect

Implements [Arcade::IComponent](#).

### 3.8.1.6 getText()

```
std::string Arcade::Component::getText ( ) const [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

Returns

Rect

Get text from the component Only works with Type (TEXT). Will throw an exception if Type != TEXT

Returns

std::string

Implements [Arcade::IComponent](#).

### 3.8.1.7 getType()

```
Arcade::IComponent::Type Arcade::Component::getType ( ) const [virtual]
```

Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)

Returns

Type

Implements [Arcade::IComponent](#).

### 3.8.1.8 getWidth()

```
int Arcade::Component::getWidth ( ) const [virtual]
```

Get width of the OBJECT/TEXT [Component](#).

Returns

int

Implements [Arcade::IComponent](#).

### 3.8.1.9 getX()

```
int Arcade::Component::getX ( ) const [virtual]
```

Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

intP

Implements [Arcade::IComponent](#).

### 3.8.1.10 getY()

```
int Arcade::Component::getY ( ) const [virtual]
```

Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

int

Implements [Arcade::IComponent](#).

### 3.8.1.11 setHeight()

```
void Arcade::Component::setHeight (
    const int height ) [virtual]
```

Set the Y value of the [Component](#).

**Parameters**

<i>y</i>	
----------	--

Implements [Arcade::IComponent](#).

**3.8.1.12 setRect()**

```
void Arcade::Component::setRect (
    Rect const rect ) [virtual]
```

Set the [Rect](#) of the OBJECT [Component](#).

**Parameters**

<i>rect</i>	
-------------	--

Implements [Arcade::IComponent](#).

**3.8.1.13 setText()**

```
void Arcade::Component::setText (
    std::string const text ) [virtual]
```

Set the text of the [Component](#).

**Parameters**

<i>text</i>	
-------------	--

Implements [Arcade::IComponent](#).

**3.8.1.14 setWidth()**

```
void Arcade::Component::setWidth (
    const int width ) [virtual]
```

Set the X value of the [Component](#).

**Parameters**

<i>x</i>	
----------	--



Implements [Arcade::IComponent](#).

### 3.8.1.15 setX()

```
void Arcade::Component::setX (
    std::size_t x ) [virtual]
```

Set the X value of the [Component](#).

#### Parameters

<i>x</i>	
----------	--

Implements [Arcade::IComponent](#).

### 3.8.1.16 setY()

```
void Arcade::Component::setY (
    std::size_t y ) [virtual]
```

Set the Y value of the [Component](#).

#### Parameters

<i>y</i>	
----------	--

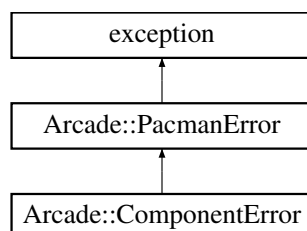
Implements [Arcade::IComponent](#).

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

- games/menu/include/Component.hpp
- games/menu/src/Component.cpp

## 3.9 Arcade::ComponentError Class Reference

Inheritance diagram for Arcade::ComponentError:



## Public Member Functions

- **ComponentError** (const ComponentErrorType type)

## Protected Member Functions

- std::string **\_basicErrorMessage** (const ComponentErrorType type)

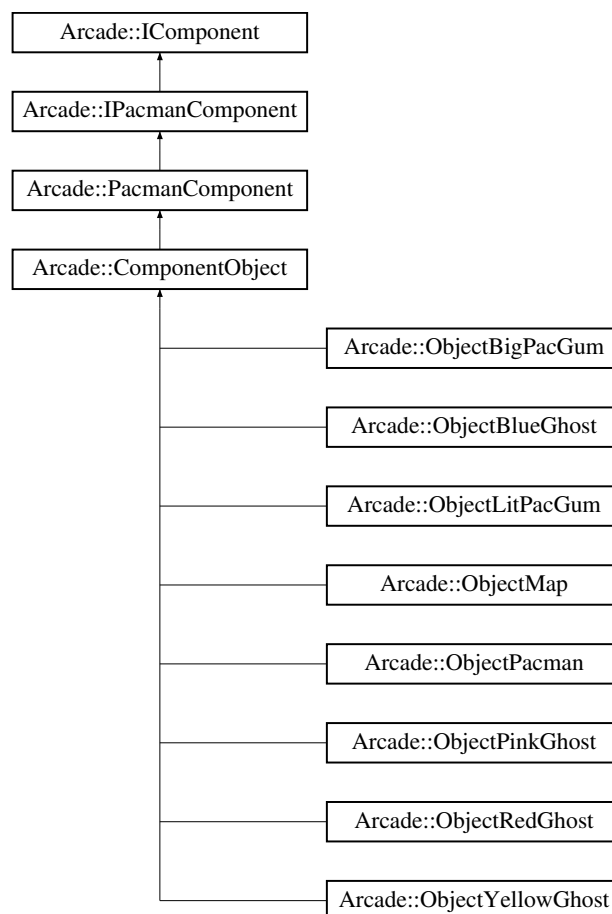
## Additional Inherited Members

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

- games/pacman/include/PacmanError/ComponentError.hpp
- games/pacman/src/PacmanError/ComponentError.cpp

## 3.10 Arcade::ComponentObject Class Reference

Inheritance diagram for Arcade::ComponentObject:



## Public Member Functions

- **ComponentObject** (const std::size\_t &id, const std::string &file, const ComponentObjectType &type)
- ComponentObjectType **getComponentObjectType** () const override
- void **setX** (const size\_t x)
- int **getX** () const override  
*Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- void **setY** (const size\_t y)
- int **getY** () const override  
*Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- void **setWidth** (const int width)  
*Set the X value of the [Component](#).*
- int **getWidth** () const override  
*Get width of the OBJECT/TEXT [Component](#).*
- void **setHeight** (const int height)  
*Set the Y value of the [Component](#).*
- int **getHeight** () const override  
*Get height of the OBJECT/TEXT [Component](#).*
- void **setRect** (const [Rect](#) rect)  
*Set the [Rect](#) of the OBJECT [Component](#).*
- [Rect](#) **getRect** () const override  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- void **setObjectAnimation** (const PacmanAnimType &type)
- void **setObjectCurrentAnimation** (const PacmanAnimMove &move)
- void **animateObject** ()

## Static Public Member Functions

- static IPacCompUPtr **makeObject** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos, const Component←ObjectType &type)

## Additional Inherited Members

### 3.10.1 Member Function Documentation

#### 3.10.1.1 getHeight()

```
int Arcade::ComponentObject::getHeight ( ) const [override], [virtual]
```

Get height of the OBJECT/TEXT [Component](#).

Returns

int

Reimplemented from [Arcade::PacmanComponent](#).

### 3.10.1.2 getRect()

```
Arcade::Rect Arcade::ComponentObject::getRect ( ) const [override], [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

Returns

[Rect](#)

Reimplemented from [Arcade::PacmanComponent](#).

### 3.10.1.3 getWidth()

```
int Arcade::ComponentObject::getWidth ( ) const [override], [virtual]
```

Get width of the OBJECT/TEXT [Component](#).

Returns

int

Reimplemented from [Arcade::PacmanComponent](#).

### 3.10.1.4 getX()

```
int Arcade::ComponentObject::getX ( ) const [override], [virtual]
```

Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

intP

Reimplemented from [Arcade::PacmanComponent](#).

### 3.10.1.5 getY()

```
int Arcade::ComponentObject::getY ( ) const [override], [virtual]
```

Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

int

Reimplemented from [Arcade::PacmanComponent](#).

### 3.10.1.6 setHeight()

```
void Arcade::ComponentObject::setHeight (
    const int height ) [virtual]
```

Set the Y value of the [Component](#).

## Parameters

<i>y</i>	
----------	--

Reimplemented from [Arcade::PacmanComponent](#).

### 3.10.1.7 setRect()

```
void Arcade::ComponentObject::setRect (
    const Rect rect ) [virtual]
```

Set the [Rect](#) of the OBJECT [Component](#).

## Parameters

<i>rect</i>	
-------------	--

Reimplemented from [Arcade::PacmanComponent](#).

### 3.10.1.8 setWidth()

```
void Arcade::ComponentObject::setWidth (
    const int width ) [virtual]
```

Set the X value of the [Component](#).

## Parameters

<i>x</i>	
----------	--

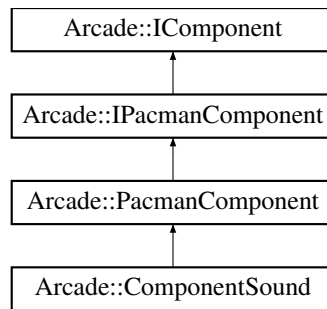
Reimplemented from [Arcade::PacmanComponent](#).

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

- games/pacman/include/PacmanComponent/ComponentObject.hpp
- games/pacman/src/PacmanComponent/ComponentObject.cpp

## 3.11 Arcade::ComponentSound Class Reference

Inheritance diagram for Arcade::ComponentSound:



## Public Member Functions

- **ComponentSound** (const std::size\_t &id, const std::string &file)

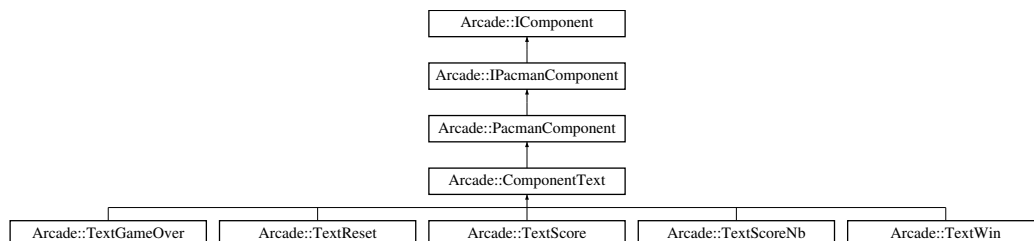
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ComponentSound.hpp
- games/pacman/src/PacmanComponent/ComponentSound.cpp

## 3.12 Arcade::ComponentText Class Reference

Inheritance diagram for `Arcade::ComponentText`:



## Public Member Functions

- **ComponentText** (const std::size\_t &id, const std::string &file, const ComponentTextType &type)
- ComponentTextType **getComponentTextType** () const override
- void **setX** (const size\_t x)
- int **getX** () const override
 

*Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- void **setY** (const size\_t y)
- int **getY** () const override
 

*Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- void **setWidth** (const int width)
 

*Set the X value of the Component.*
- int **getWidth** () const override

- *Get width of the OBJECT/TEXT [Component](#).*
- void [setWidth](#) (const int width)
- *Set the X value of the [Component](#).*
- int [getWidth](#) () const override
- *Get height of the OBJECT/TEXT [Component](#).*
- void [setRect](#) ([Rect](#) const rect)
- *Set the [Rect](#) of the OBJECT [Component](#).*
- [Rect](#) [getRect](#) () const override
- *Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- void [setFont](#) (const int size)
- int [getFontSize](#) () const override
- *Get the font size of the TEXT [Component](#).*
- void [setText](#) (const std::string text)
- *Set the text of the [Component](#).*
- std::string [getText](#) () const override
- *Get animated sprite rect #THROW if no rect or Type != OBJECT.*

## Static Public Member Functions

- static IPacCompUPtr [makeText](#) (const std::size\_t &id, const ComponentTextType &type)

## Additional Inherited Members

### 3.12.1 Member Function Documentation

#### 3.12.1.1 [getFontSize\(\)](#)

```
int Arcade::ComponentText::getFontSize ( ) const [override], [virtual]
```

Get the font size of the TEXT [Component](#).

Returns

int

Reimplemented from [Arcade::PacmanComponent](#).

#### 3.12.1.2 [getHeight\(\)](#)

```
int Arcade::ComponentText::getHeight ( ) const [override], [virtual]
```

Get height of the OBJECT/TEXT [Component](#).

Returns

int

Reimplemented from [Arcade::PacmanComponent](#).

### 3.12.1.3 getRect()

```
Arcade::Rect Arcade::ComponentText::getRect ( ) const [override], [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

Returns

[Rect](#)

Reimplemented from [Arcade::PacmanComponent](#).

### 3.12.1.4 getText()

```
std::string Arcade::ComponentText::getText ( ) const [override], [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

Returns

[Rect](#)

Get text from the component Only works with Type (TEXT). Will throw an exception if Type != TEXT

Returns

std::string

Reimplemented from [Arcade::PacmanComponent](#).

### 3.12.1.5 getWidth()

```
int Arcade::ComponentText::getWidth ( ) const [override], [virtual]
```

Get width of the OBJECT/TEXT [Component](#).

Returns

int

Reimplemented from [Arcade::PacmanComponent](#).



### 3.12.1.6 getX()

```
int Arcade::ComponentText::getX ( ) const [override], [virtual]
```

Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

#### Returns

intP

Reimplemented from [Arcade::PacmanComponent](#).

### 3.12.1.7 getY()

```
int Arcade::ComponentText::getY ( ) const [override], [virtual]
```

Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

#### Returns

int

Reimplemented from [Arcade::PacmanComponent](#).

### 3.12.1.8 setHeight()

```
void Arcade::ComponentText::setHeight (
    const int height ) [virtual]
```

Set the Y value of the [Component](#).

#### Parameters

<i>y</i>	
----------	--

Reimplemented from [Arcade::PacmanComponent](#).

### 3.12.1.9 setRect()

```
void Arcade::ComponentText::setRect (
    Rect const rect ) [virtual]
```

Set the [Rect](#) of the OBJECT [Component](#).

**Parameters**

<i>rect</i>	
-------------	--

Reimplemented from [Arcade::PacmanComponent](#).

**3.12.1.10 setText()**

```
void Arcade::ComponentText::setText (
    const std::string text ) [virtual]
```

Set the text of the [Component](#).

**Parameters**

<i>text</i>	
-------------	--

Reimplemented from [Arcade::PacmanComponent](#).

**3.12.1.11 setWidth()**

```
void Arcade::ComponentText::setWidth (
    const int width ) [virtual]
```

Set the X value of the [Component](#).

**Parameters**

<i>x</i>	
----------	--

Reimplemented from [Arcade::PacmanComponent](#).

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

- games/pacman/include/PacmanComponent/ComponentText.hpp
- games/pacman/src/PacmanComponent/ComponentText.cpp

## 3.13 Arcade::Core Class Reference

**Public Member Functions**

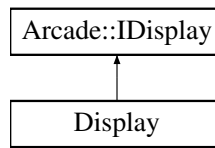
- int **errorHandling** (const int ac, char const \*const \*av)
- void **run** ()

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

- core/include/Core.hpp
- core/src/Core.cpp

## 3.14 Display Class Reference

Inheritance diagram for Display:



### Classes

- class [DisplayException](#)

### Public Member Functions

- void [init](#) () override  
*Initialize the library.*
- void [stop](#) () override  
*Stop the library.*
- std::string [getLibName](#) () const override  
*Get the Library Name.*
- void [display](#) (std::vector< std::unique\_ptr< [Arcade::IComponent](#) >> &components) override  
*Displays components. If Component Type is SOUND, nothing happens.*
- std::vector< std::unique\_ptr< [Arcade::IEvent](#) > > & [getEvents](#) () override  
*Get the Events polled by the library.*
- void [clear](#) ()  
*Clear graphicals objects list.*

### Static Public Member Functions

- static int [calculateXPosition](#) (int x)
- static int [calculateYPosition](#) (int y)
- static int [calculateTextureSize](#) (int sideSize)

#### 3.14.1 Member Function Documentation

##### 3.14.1.1 display()

```
void Display::display (
    std::vector< std::unique_ptr< Arcade::IComponent >> & components ) [override],
[virtual]
```

Displays components. If Component Type is SOUND, nothing happens.

## Parameters

<i>components</i>	
-------------------	--

Implements [Arcade::IDisplay](#).

### 3.14.1.2 `getEvents()`

```
std::vector< std::unique_ptr< Arcade::IEvent > > & Display::getEvents ( ) [override], [virtual]
```

Get the Events polled by the library.

## Returns

```
std::vector<std::unique_ptr<IEvent>>&
```

Implements [Arcade::IDisplay](#).

### 3.14.1.3 `getLibName()`

```
std::string Display::getLibName ( ) const [override], [virtual]
```

Get the Library Name.

## Returns

```
std::string
```

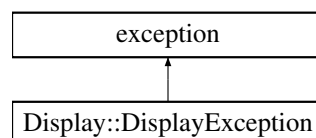
Implements [Arcade::IDisplay](#).

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

- `graphics/sdl2/include/Display.hpp`
- `graphics/sdl2/src/Display.cpp`

## 3.15 `Display::DisplayException` Class Reference

Inheritance diagram for `Display::DisplayException`:



## Public Member Functions

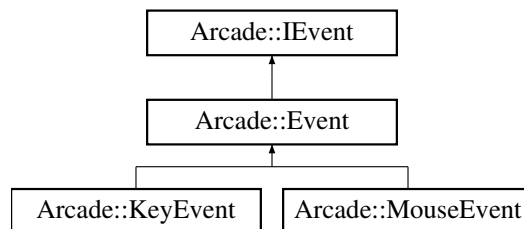
- **DisplayException** (std::string msg)
- const char \* **what** () const noexcept override

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

- graphics/sdl2/include/Display.hpp
- graphics/sdl2/src/Display.cpp

## 3.16 Arcade::Event Class Reference

Inheritance diagram for Arcade::Event:



## Public Types

- enum **Type** { **MOUSE\_EVENT**, **KEY\_EVENT**, **WIN\_EVENT**, **NONE** }

## Public Member Functions

- **Event** ()  
*Constructor with no arguments. Initialize \_type with Type::NONE, \_key with 0 and \_pos with {0, 0}.*
- **Event** (Type event)  
*Constructor with the event type argument. Initialize \_type with the event given, \_key with 0, and \_pos with {0, 0}.*
- **Event** (Type event, size\_t key)  
*Constructor with the event type and key arguments. Initialize \_type with the event given, \_key with the key given, and \_pos with {0, 0}.*
- **Event** (Type event, **Pos** pos)  
*Constructor with the event type and mouse position arguments. Initialize \_type with the event given, \_key with 0, and \_pos with the position given.*
- **Event** (Type event, size\_t key, **Pos** pos)  
*Constructor with the event type, key and mouse position arguments. Initialize \_type with the event given, \_key with the key given, and \_pos with the position given.*
- **~Event** ()  
*Just a destructor, nothing to see here.*
- std::size\_t **getKey** () const  
*Get the Key value of the event.*
- void **setKey** (std::size\_t const key)  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- **Pos** **getMousePos** () const  
*Get the Mouse Position Position is expressed in the **Pos** structure {x, y}.*

### 3.16.1 Member Function Documentation

#### 3.16.1.1 getKey()

```
std::size_t Event::getKey ( ) const [virtual]
```

Get the Key value of the event.

##### Returns

std::size\_t

Implements [Arcade::IEvent](#).

#### 3.16.1.2 getMousePos()

```
Arcade::Pos Event::getMousePos ( ) const [virtual]
```

Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.

##### Returns

[Pos](#)

Implements [Arcade::IEvent](#).

#### 3.16.1.3 setKey()

```
void Event::setKey (
    std::size_t const key ) [virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.

##### Parameters

<i>key</i>	
------------	--

##### Returns

std::size\_t

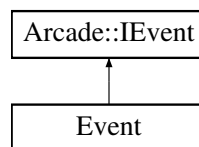
Implements [Arcade::IEvent](#).

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

- core/include/Event.hpp
- core/src/Event.cpp

## 3.17 Event Class Reference

Inheritance diagram for Event:



### Public Member Functions

- **Event** (std::size\_t key)
- **Event** (std::size\_t key, std::string data)
- std::size\_t **getKey** () const  
*Get the Key value of the event.*
- void **setKey** (std::size\_t const key)  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- **Arcade::Pos getMousePos** () const  
*Get the Mouse Position Position is expressed in the Pos structure {x, y}.*
- std::string **getData** () const  
*Get Data from [Event](#).*
- **Event** (std::size\_t key)
- **Event** (std::size\_t key, std::string data)
- std::size\_t **getKey** () const  
*Get the Key value of the event.*
- void **setKey** (std::size\_t const key)  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- **Arcade::Pos getMousePos** () const  
*Get the Mouse Position Position is expressed in the Pos structure {x, y}.*
- std::string **getData** () const  
*Get Data from [Event](#).*
- **Event** (std::size\_t key)
- **Event** (std::size\_t key, int x, int y)
- std::size\_t **getKey** () const  
*Get the Key value of the event.*
- void **setKey** (std::size\_t const key)  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- **Arcade::Pos getMousePos** () const  
*Get the Mouse Position Position is expressed in the Pos structure {x, y}.*
- std::string **getData** () const  
*Get Data from [Event](#).*

### 3.17.1 Member Function Documentation

#### 3.17.1.1 `getData()` [1/3]

```
std::string Event::getData ( ) const [virtual]
```

Get Data from [Event](#).

##### Returns

std::string

Implements [Arcade::IEvent](#).

#### 3.17.1.2 `getData()` [2/3]

```
std::string Event::getData ( ) const [virtual]
```

Get Data from [Event](#).

##### Returns

std::string

Implements [Arcade::IEvent](#).

#### 3.17.1.3 `getData()` [3/3]

```
std::string Event::getData ( ) const [virtual]
```

Get Data from [Event](#).

##### Returns

std::string

Implements [Arcade::IEvent](#).



#### 3.17.1.4 getKey() [1/3]

```
std::size_t Event::getKey ( ) const [virtual]
```

Get the Key value of the event.

##### Returns

std::size\_t

Implements [Arcade::IEvent](#).

#### 3.17.1.5 getKey() [2/3]

```
std::size_t Event::getKey ( ) const [virtual]
```

Get the Key value of the event.

##### Returns

std::size\_t

Implements [Arcade::IEvent](#).

#### 3.17.1.6 getKey() [3/3]

```
std::size_t Event::getKey ( ) const [virtual]
```

Get the Key value of the event.

##### Returns

std::size\_t

Implements [Arcade::IEvent](#).

#### 3.17.1.7 getMousePos() [1/3]

```
Arcade::Pos Event::getMousePos ( ) const [virtual]
```

Get the Mouse Position Position is expressed in the Pos structure {x, y}.

##### Returns

Pos

Implements [Arcade::IEvent](#).

### 3.17.1.8 `getMousePos()` [2/3]

```
Arcade::Pos Event::getMousePos ( ) const [virtual]
```

Get the Mouse Position Position is expressed in the Pos structure {x, y}.

#### Returns

Pos

Implements [Arcade::IEvent](#).

### 3.17.1.9 `getMousePos()` [3/3]

```
Arcade::Pos Event::getMousePos ( ) const [virtual]
```

Get the Mouse Position Position is expressed in the Pos structure {x, y}.

#### Returns

Pos

Implements [Arcade::IEvent](#).

### 3.17.1.10 `setKey()` [1/3]

```
void Event::setKey (
    std::size_t const key ) [virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.

#### Parameters

<i>key</i>	
------------	--

#### Returns

void

Implements [Arcade::IEvent](#).

**3.17.1.11 setKey()** [2/3]

```
void Event::setKey (
    std::size_t const key ) [virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.

**Parameters**

<i>key</i>	
------------	--

**Returns**

void

Implements [Arcade::IEvent](#).

**3.17.1.12 setKey()** [3/3]

```
void Event::setKey (
    std::size_t const key ) [virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.

**Parameters**

<i>key</i>	
------------	--

**Returns**

void

Implements [Arcade::IEvent](#).

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

- games/menu/include/Event.hpp
- games/menu/src/Event.cpp

## 3.18 EventNormalizer Class Reference

**Public Member Functions**

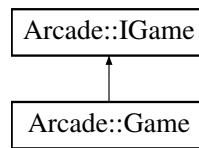
- void **normalizeEvents** (std::vector< std::unique\_ptr< [Arcade::IEvent](#) >> &events, const std::string &name)

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

- core/include/EventNormalizer.hpp
- core/src/EventNormalizer.cpp

## 3.19 Arcade::Game Class Reference

Inheritance diagram for Arcade::Game:



### Public Member Functions

- **Game** (std::string name)
- void **init** ()  
*initilaze the game*
- void **stop** ()  
*cleanup the game*
- std::string **getGameName** () const  
*Get the **Game** name.*
- std::vector< std::unique\_ptr< **IComponent** > > & **getComponents** ()  
*Get the IComponents from the game Their order in the vector determines in which order they are displayed.*
- void **sendEvents** (std::vector< std::unique\_ptr< **IEvent** >> &events)  
*This function is called to send the events polled by the libs to the game. You can use this function as your game entrypoint.*
- **IEvent** \* **getEvent** ()  
*Get special events from **IGame**. It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.*
- void **setPlayerName** (std::string const name)  
*Set the Player Name.*
- std::string **getPlayerName** () const  
*Get the Player Name.*

### 3.19.1 Member Function Documentation

#### 3.19.1.1 getComponents()

```
std::vector< std::unique_ptr< IComponent > > & Arcade::Game::getComponents ( ) [virtual]
```

Get the IComponents from the game Their order in the vector determines in which order they are displayed.

#### Returns

std::vector<std::unique\_ptr<IComponent>>&

Implements [Arcade::IGame](#).

### 3.19.1.2 `getEvent()`

```
IEvent * Arcade::Game::getEvent ( ) [virtual]
```

Get special events from [IGame](#). It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.

#### Returns

`IEvent*`

Implements [Arcade::IGame](#).

### 3.19.1.3 `getGameName()`

```
std::string Arcade::Game::getGameName ( ) const [virtual]
```

Get the [Game](#) name.

#### Returns

`std::string`

Implements [Arcade::IGame](#).

### 3.19.1.4 `getPlayerName()`

```
std::string Arcade::Game::getPlayerName ( ) const [virtual]
```

Get the Player Name.

#### Returns

`std::string`

Implements [Arcade::IGame](#).

### 3.19.1.5 `sendEvents()`

```
void Arcade::Game::sendEvents (
    std::vector< std::unique_ptr< IEvent >> & events ) [virtual]
```

This function is called to send the events polled by the libs to the game. You can use this function as your game entrypoint.

## Parameters

<i>events</i>	
---------------	--

Implements [Arcade::IGame](#).

### 3.19.1.6 setPlayerName()

```
void Arcade::Game::setPlayerName (
    std::string const name ) [virtual]
```

Set the Player Name.

## Parameters

<i>name</i>	
-------------	--

Implements [Arcade::IGame](#).

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

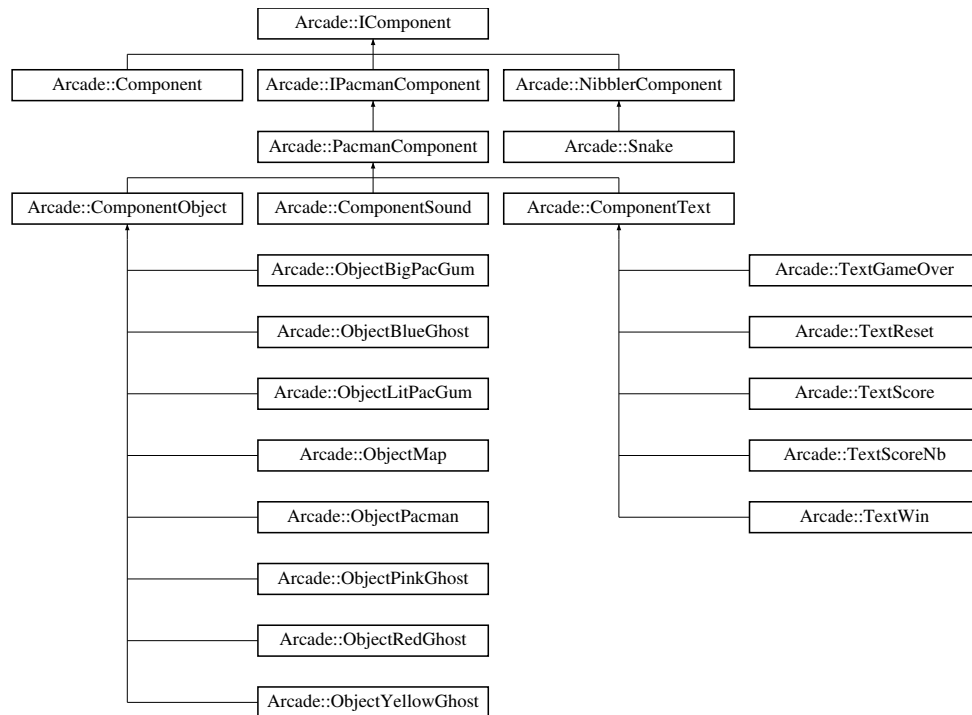
- games/menu/include/Game.hpp
- games/menu/src/Game.cpp

## 3.20 Arcade::IComponent Class Reference

The [IComponent](#) interface represents a game entity that can either be displayed or played as a sound.

```
#include <IComponent.hpp>
```

Inheritance diagram for Arcade::IComponent:



## Public Types

- enum **Type** { **OBJECT**, **SOUND**, **TEXT** }

## Public Member Functions

- virtual std::size\_t **getId** () const =0  
*Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.*
- virtual Type **getType** () const =0  
*Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)*
- virtual std::string **getFile** () const =0  
*Get file associated with component Use it to create textures, etc.*
- virtual int **getX** () const =0  
*Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- virtual int **getY** () const =0  
*Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- virtual std::string **getText** () const =0  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- virtual Rect **getRect** () const =0  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- virtual int **getWidth** () const =0  
*Get width of the OBJECT/TEXT Component.*
- virtual int **getHeight** () const =0  
*Get height of the OBJECT/TEXT Component.*
- virtual int **getFontSize** () const =0  
*Get the font size of the TEXT Component.*
- virtual void **setX** (std::size\_t const x)=0

- Set the *X* value of the [Component](#).  
 • virtual void [setY](#) (std::size\_t const y)=0  
 Set the *Y* value of the [Component](#).
- virtual void [setWidth](#) (const int width)=0  
 Set the *X* value of the [Component](#).
- virtual void [setHeight](#) (const int height)=0  
 Set the *Y* value of the [Component](#).
- virtual void [setText](#) (std::string const text)=0  
 Set the text of the [Component](#).
- virtual void [setRect](#) ([Rect](#) const rect)=0  
 Set the [Rect](#) of the *OBJECT* [Component](#).

### 3.20.1 Detailed Description

The [IComponent](#) interface represents a game entity that can either be displayed or played as a sound.

### 3.20.2 Member Function Documentation

#### 3.20.2.1 [getFile\(\)](#)

```
virtual std::string Arcade::IComponent::getFile ( ) const [pure virtual]
```

Get file associated with component Use it to create textures, etc.

Returns

std::string

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

#### 3.20.2.2 [getFontSize\(\)](#)

```
virtual int Arcade::IComponent::getFontSize ( ) const [pure virtual]
```

Get the font size of the TEXT [Component](#).

Returns

int

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).



### 3.20.2.3 getHeight()

```
virtual int Arcade::IComponent::getHeight ( ) const [pure virtual]
```

Get height of the OBJECT/TEXT [Component](#).

#### Returns

int

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::ComponentObject](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.4 getId()

```
virtual std::size_t Arcade::IComponent::getId ( ) const [pure virtual]
```

Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.

#### Returns

std::size\_t

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.5 getRect()

```
virtual Rect Arcade::IComponent::getRect ( ) const [pure virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

#### Returns

[Rect](#)

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::ComponentObject](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.6 getText()

```
virtual std::string Arcade::IComponent::getText ( ) const [pure virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

Returns

[Rect](#)

Get text from the component Only works with Type (TEXT). Will throw an exception if Type != TEXT

Returns

std::string

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.7 getType()

```
virtual Type Arcade::IComponent::getType ( ) const [pure virtual]
```

Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)

Returns

Type

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.8 getWidth()

```
virtual int Arcade::IComponent::getWidth ( ) const [pure virtual]
```

Get width of the OBJECT/TEXT [Component](#).

Returns

int

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::ComponentObject](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.9 getX()

```
virtual int Arcade::IComponent::getX ( ) const [pure virtual]
```

Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

intP

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::ComponentObject](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.10 getY()

```
virtual int Arcade::IComponent::getY ( ) const [pure virtual]
```

Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

int

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::ComponentObject](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.11 setHeight()

```
virtual void Arcade::IComponent::setHeight (
    const int height ) [pure virtual]
```

Set the Y value of the [Component](#).

Parameters

y	
---	--

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::ComponentObject](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.12 setRect()

```
virtual void Arcade::IComponent::setRect (
    Rect const rect ) [pure virtual]
```

Set the [Rect](#) of the [OBJECT Component](#).

#### Parameters

<i>rect</i>	
-------------	--

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::NibblerComponent](#), [Arcade::Component](#), and [Arcade::ComponentObject](#).

### 3.20.2.13 setText()

```
virtual void Arcade::IComponent::setText (
    std::string const text ) [pure virtual]
```

Set the text of the [Component](#).

#### Parameters

<i>text</i>	
-------------	--

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::NibblerComponent](#), [Arcade::Component](#), and [Arcade::ComponentText](#).

### 3.20.2.14 setWidth()

```
virtual void Arcade::IComponent::setWidth (
    const int width ) [pure virtual]
```

Set the X value of the [Component](#).

#### Parameters

<i>x</i>	
----------	--

Implemented in [Arcade::IPacmanComponent](#), [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), [Arcade::ComponentObject](#), [Arcade::NibblerComponent](#), and [Arcade::Component](#).

### 3.20.2.15 setX()

```
virtual void Arcade::IComponent::setX (
    std::size_t const x ) [pure virtual]
```

Set the X value of the [Component](#).

## Parameters

<i>x</i>	
----------	--

Implemented in [Arcade::Snake](#), [Arcade::NibblerComponent](#), [Arcade::Component](#), [Arcade::IPacmanComponent](#), and [Arcade::PacmanComponent](#).

**3.20.2.16 setY()**

```
virtual void Arcade::IComponent::setY (
    std::size_t const y ) [pure virtual]
```

Set the Y value of the [Component](#).

## Parameters

<i>y</i>	
----------	--

Implemented in [Arcade::Snake](#), [Arcade::NibblerComponent](#), [Arcade::Component](#), [Arcade::IPacmanComponent](#), and [Arcade::PacmanComponent](#).

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

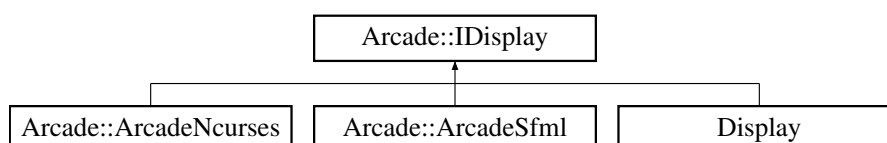
- include/IComponent.hpp

## 3.21 Arcade::IDisplay Class Reference

This interface defines how to interact with graphic libraries. It represents the IO components. [IDisplay](#)'s purpose is to poll events (Inputs), and to draw components on the screen / play sounds (Outputs)

```
#include <IDisplay.hpp>
```

Inheritance diagram for Arcade::IDisplay:



## Public Member Functions

- virtual void [init](#) ()=0  
*Initialize the library.*
- virtual void [stop](#) ()=0  
*Stop the library.*
- virtual std::string [getLibName](#) () const =0  
*Get the Library Name.*
- virtual void [display](#) (std::vector< std::unique\_ptr< [IComponent](#) >> &components)=0  
*Displays components. If [Component](#) Type is SOUND, nothing happens.*
- virtual std::vector< std::unique\_ptr< [IEvent](#) > > & [getEvents](#) ()=0  
*Get the Events polled by the library.*
- virtual void [clear](#) ()=0  
*Clear graphicals objects list.*

### 3.21.1 Detailed Description

This interface defines how to interact with graphic libraries. It represents the IO components. [IDisplay](#)'s purpose is to poll events (Inputs), and to draw components on the screen / play sounds (Outputs)

### 3.21.2 Member Function Documentation

#### 3.21.2.1 [display\(\)](#)

```
virtual void Arcade::IDisplay::display (
    std::vector< std::unique_ptr< IComponent >> & components ) [pure virtual]
```

Displays components. If [Component](#) Type is SOUND, nothing happens.

##### Parameters

<a href="#">components</a>	
----------------------------	--

Implemented in [Display](#).

#### 3.21.2.2 [getEvents\(\)](#)

```
virtual std::vector<std::unique_ptr<IEvent> >& Arcade::IDisplay::getEvents ( ) [pure virtual]
```

Get the Events polled by the library.

##### Returns

std::vector<std::unique\_ptr<[IEvent](#)>>&

Implemented in [Display](#), [Arcade::ArcadeSfml](#), and [Arcade::ArcadeNcurses](#).

### 3.21.2.3 getLibName()

```
virtual std::string Arcade::IDisplay::getLibName ( ) const [pure virtual]
```

Get the Library Name.

#### Returns

std::string

Implemented in [Display](#), [Arcade::ArcadeSfml](#), and [Arcade::ArcadeNcurses](#).

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

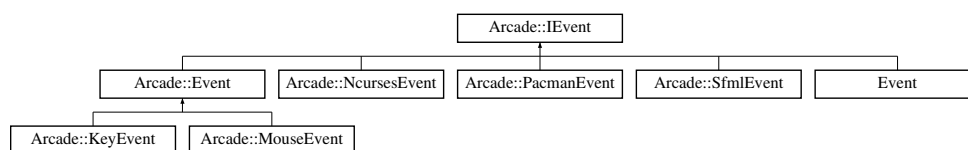
- include/IDisplay.hpp

## 3.22 Arcade::IEvent Class Reference

Represents an event polled by the graphic libraries.

```
#include <IEvent.hpp>
```

Inheritance diagram for Arcade::IEvent:



### Public Member Functions

- virtual std::size\_t [getKey](#) ( ) const =0  
*Get the Key value of the event.*
- virtual void [setKey](#) (std::size\_t const key)=0  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- virtual [Pos](#) [getMousePos](#) ( ) const =0  
*Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.*
- virtual std::string [getData](#) ( ) const =0  
*Get Data from [Event](#).*

### 3.22.1 Detailed Description

Represents an event polled by the graphic libraries.

### 3.22.2 Member Function Documentation

### 3.22.2.1 `getData()`

```
virtual std::string Arcade::IEvent::getData ( ) const [pure virtual]
```

Get Data from [Event](#).

#### Returns

`std::string`

Implemented in [Arcade::SfmlEvent](#), [Event](#), [Arcade::NcursesEvent](#), [Arcade::PacmanEvent](#), [Event](#), and [Event](#).

### 3.22.2.2 `getKey()`

```
virtual std::size_t Arcade::IEvent::getKey ( ) const [pure virtual]
```

Get the Key value of the event.

#### Returns

`std::size_t`

Implemented in [Arcade::SfmlEvent](#), [Event](#), [Arcade::NcursesEvent](#), [Arcade::PacmanEvent](#), [Event](#), [Event](#), and [Arcade::Event](#).

### 3.22.2.3 `getMousePos()`

```
virtual Pos Arcade::IEvent::getMousePos ( ) const [pure virtual]
```

Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.

#### Returns

[Pos](#)

Implemented in [Arcade::SfmlEvent](#), [Event](#), [Arcade::NcursesEvent](#), [Arcade::PacmanEvent](#), [Event](#), [Event](#), and [Arcade::Event](#).

### 3.22.2.4 `setKey()`

```
virtual void Arcade::IEvent::setKey (
    std::size_t const key ) [pure virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.



## Parameters

<i>key</i>	
------------	--

## Returns

void

Implemented in [Arcade::SfmlEvent](#), [Event](#), [Arcade::NcursesEvent](#), [Arcade::PacmanEvent](#), [Event](#), [Event](#), and [Arcade::Event](#).

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

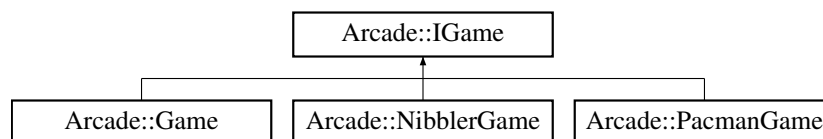
- include/IEvent.hpp

## 3.23 Arcade::IGame Class Reference

[IGame](#) represents a game loaded into the core. Its methods allow to get game information, which can be passed down to [IDisplay](#) by the core.

```
#include <IGame.hpp>
```

Inheritance diagram for `Arcade::IGame`:



### Public Member Functions

- virtual void [init](#) ()=0  
*initilaze the game*
- virtual void [stop](#) ()=0  
*cleanup the game*
- virtual std::string [getGameName](#) () const =0  
*Get the [Game](#) name.*
- virtual std::vector< std::unique\_ptr< [IComponent](#) > > & [getComponents](#) ()=0  
*Get the IComponents from the game Their order in the vector determines in which order they are displayed.*
- virtual void [sendEvents](#) (std::vector< std::unique\_ptr< [IEvent](#) > > &events)=0  
*This function is called to send the events polled by the libs to the game. You can use this function as your game entrypoint.*
- virtual [IEvent](#) \* [getEvent](#) ()=0  
*Get special events from [IGame](#). It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.*
- virtual void [setPlayerName](#) (std::string const name)=0  
*Set the Player Name.*
- virtual std::string [getPlayerName](#) () const =0  
*Get the Player Name.*

### 3.23.1 Detailed Description

[IGame](#) represents a game loaded into the core. Its methods allow to get game information, which can be passed down to [IDisplay](#) by the core.

### 3.23.2 Member Function Documentation

#### 3.23.2.1 `getComponents()`

```
virtual std::vector<std::unique_ptr<IComponent>> & Arcade::IGame::getComponents ( ) [pure virtual]
```

Get the [IComponents](#) from the game Their order in the vector determines in which order they are displayed.

##### Returns

`std::vector<std::unique_ptr<IComponent>>&`

Implemented in [Arcade::PacmanGame](#), [Arcade::NibblerGame](#), and [Arcade::Game](#).

#### 3.23.2.2 `getEvent()`

```
virtual IEvent* Arcade::IGame::getEvent ( ) [pure virtual]
```

Get special events from [IGame](#). It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.

##### Returns

`IEvent*`

Implemented in [Arcade::PacmanGame](#), [Arcade::NibblerGame](#), and [Arcade::Game](#).

#### 3.23.2.3 `getGameName()`

```
virtual std::string Arcade::IGame::getGameName ( ) const [pure virtual]
```

Get the [Game](#) name.

##### Returns

`std::string`

Implemented in [Arcade::PacmanGame](#), [Arcade::NibblerGame](#), and [Arcade::Game](#).

### 3.23.2.4 getPlayerName()

```
virtual std::string Arcade::IGame::getPlayerName ( ) const [pure virtual]
```

Get the Player Name.

#### Returns

std::string

Implemented in [Arcade::PacmanGame](#), [Arcade::NibblerGame](#), and [Arcade::Game](#).

### 3.23.2.5 sendEvents()

```
virtual void Arcade::IGame::sendEvents (
    std::vector< std::unique_ptr< IEvent >> & events ) [pure virtual]
```

This function is called to send the events polled by the libs to the game. You can use this function as your game entrypoint.

#### Parameters

<i>events</i>	
---------------	--

Implemented in [Arcade::NibblerGame](#), and [Arcade::Game](#).

### 3.23.2.6 setPlayerName()

```
virtual void Arcade::IGame::setPlayerName (
    std::string const name ) [pure virtual]
```

Set the Player Name.

#### Parameters

<i>name</i>	
-------------	--

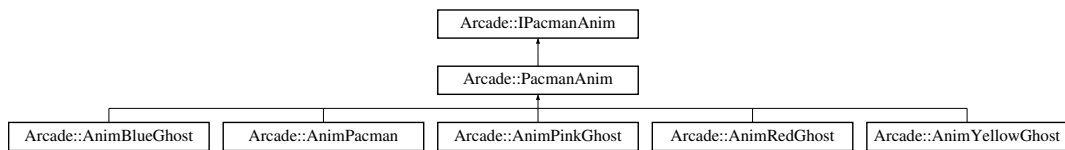
Implemented in [Arcade::PacmanGame](#), [Arcade::NibblerGame](#), and [Arcade::Game](#).

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

- include/IGame.hpp

### 3.24 Arcade::IPacmanAnim Class Reference

Inheritance diagram for Arcade::IPacmanAnim:



#### Public Member Functions

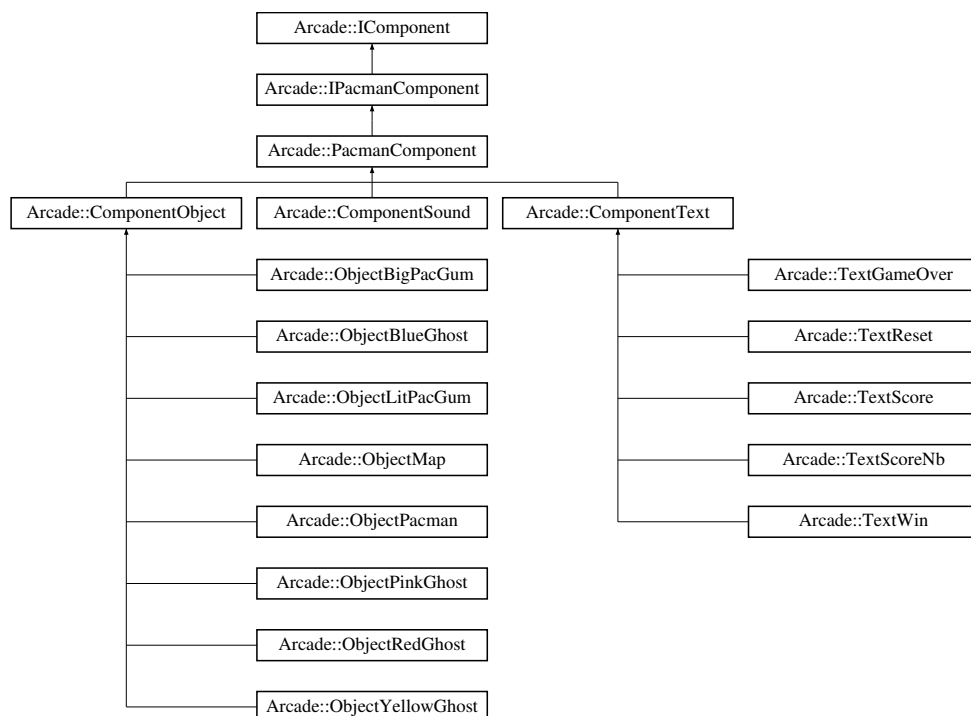
- virtual void **setAnim** (const RectList &rectList, const float &delay, const PacmanAnimMove &type)=0
- virtual void **setCurrentAnim** (const PacmanAnimMove &currentAnim)=0
- virtual PacmanAnimMove **getCurrentAnim** (void) const =0
- virtual **Rect animate** ()=0

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

- games/pacman/include/PacmanAnim/IPacmanAnim.hpp

### 3.25 Arcade::IPacmanComponent Class Reference

Inheritance diagram for Arcade::IPacmanComponent:



## Public Member Functions

- virtual std::size\_t **getId** () const =0  
*Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.*
- virtual Type **getType** () const =0  
*Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)*
- virtual ComponentObjectType **getComponentObjectType** () const =0
- virtual ComponentTextType **getComponentTextType** () const =0
- virtual std::string **getFile** () const =0  
*Get file associated with component Use it to create textures, etc.*
- virtual void **setX** (std::size\_t const x)=0  
*Set the X value of the [Component](#).*
- virtual int **getX** () const override=0  
*Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- virtual void **setY** (std::size\_t const y)=0  
*Set the Y value of the [Component](#).*
- virtual int **getY** () const override=0  
*Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- virtual void **setWidth** (const int width)=0  
*Set the X value of the [Component](#).*
- virtual int **getWidth** () const override=0  
*Get width of the OBJECT/TEXT [Component](#).*
- virtual void **setHeight** (const int height)=0  
*Set the Y value of the [Component](#).*
- virtual int **getHeight** () const override=0  
*Get height of the OBJECT/TEXT [Component](#).*
- virtual void **setRect** (Rect const rect)=0  
*Set the [Rect](#) of the OBJECT [Component](#).*
- virtual Rect **getRect** () const override=0  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- virtual void **setFontSize** (const int size)=0
- virtual int **getFontSize** () const override=0  
*Get the font size of the TEXT [Component](#).*
- virtual void **setText** (std::string const text)=0  
*Set the text of the [Component](#).*
- virtual std::string **getText** () const override=0  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- virtual void **setObjectAnimation** (const PacmanAnimType &type)=0
- virtual void **setObjectCurrentAnimation** (const PacmanAnimMove &move)=0
- virtual void **animateObject** ()=0

## Additional Inherited Members

### 3.25.1 Member Function Documentation

### 3.25.1.1 getFile()

```
virtual std::string Arcade::IPacmanComponent::getFile ( ) const [pure virtual]
```

Get file associated with component Use it to create textures, etc.

#### Returns

std::string

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#).

### 3.25.1.2 getFontSize()

```
virtual int Arcade::IPacmanComponent::getFontSize ( ) const [override], [pure virtual]
```

Get the font size of the TEXT [Component](#).

#### Returns

int

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), and [Arcade::ComponentText](#).

### 3.25.1.3 getHeight()

```
virtual int Arcade::IPacmanComponent::getHeight ( ) const [override], [pure virtual]
```

Get height of the OBJECT/TEXT [Component](#).

#### Returns

int

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

#### 3.25.1.4 getId()

```
virtual std::size_t Arcade::IPacmanComponent::getId ( ) const [pure virtual]
```

Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.

##### Returns

std::size\_t

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#).

#### 3.25.1.5 getRect()

```
virtual Rect Arcade::IPacmanComponent::getRect ( ) const [override], [pure virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

##### Returns

[Rect](#)

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

#### 3.25.1.6 getText()

```
virtual std::string Arcade::IPacmanComponent::getText ( ) const [override], [pure virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

##### Returns

[Rect](#)

Get text from the component Only works with Type (TEXT). Will throw an exception if Type != TEXT

##### Returns

std::string

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), and [Arcade::ComponentText](#).

### 3.25.1.7 getType()

```
virtual Type Arcade::IPacmanComponent::getType ( ) const [pure virtual]
```

Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)

#### Returns

Type

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#).

### 3.25.1.8 getWidth()

```
virtual int Arcade::IPacmanComponent::getWidth ( ) const [override], [pure virtual]
```

Get width of the OBJECT/TEXT [Component](#).

#### Returns

int

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.25.1.9 getX()

```
virtual int Arcade::IPacmanComponent::getX ( ) const [override], [pure virtual]
```

Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

#### Returns

intP

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).



### 3.25.1.10 getY()

```
virtual int Arcade::IPacmanComponent::getY ( ) const [override], [pure virtual]
```

Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

#### Returns

int

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.25.1.11 setHeight()

```
virtual void Arcade::IPacmanComponent::setHeight (
    const int height ) [pure virtual]
```

Set the Y value of the [Component](#).

#### Parameters

<i>y</i>	
----------	--

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.25.1.12 setRect()

```
virtual void Arcade::IPacmanComponent::setRect (
    Rect const rect ) [pure virtual]
```

Set the [Rect](#) of the OBJECT [Component](#).

#### Parameters

<i>rect</i>	
-------------	--

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.25.1.13 setText()

```
virtual void Arcade::IPacmanComponent::setText (
    std::string const text ) [pure virtual]
```

Set the text of the [Component](#).

#### Parameters

<i>text</i>	
-------------	--

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), and [Arcade::ComponentText](#).

### 3.25.1.14 setWidth()

```
virtual void Arcade::IPacmanComponent::setWidth (
    const int width ) [pure virtual]
```

Set the X value of the [Component](#).

#### Parameters

<i>x</i>	
----------	--

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#), [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.25.1.15 setX()

```
virtual void Arcade::IPacmanComponent::setX (
    std::size_t const x ) [pure virtual]
```

Set the X value of the [Component](#).

#### Parameters

<i>x</i>	
----------	--

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#).

### 3.25.1.16 setY()

```
virtual void Arcade::IPacmanComponent::setY (
    std::size_t const y ) [pure virtual]
```

Set the Y value of the [Component](#).

#### Parameters

<i>y</i>	
----------	--

Implements [Arcade::IComponent](#).

Implemented in [Arcade::PacmanComponent](#).

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

- games/pacman/include/PacmanComponent/IPacmanComponent.hpp

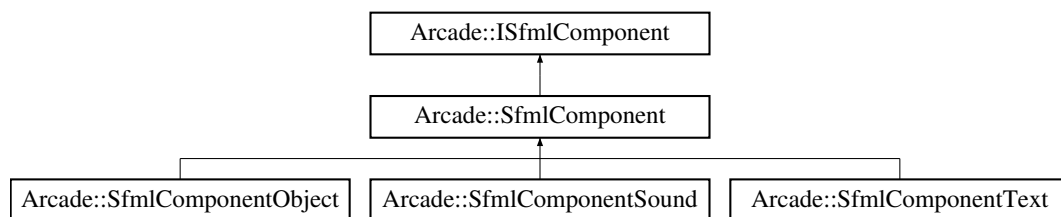
## 3.26 Arcade::IPacmanMap Class Reference

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

- games/pacman/include/PacmanMap/IPacmanMap.hpp

## 3.27 Arcade::ISfmlComponent Class Reference

Inheritance diagram for Arcade::ISfmlComponent:



### Public Member Functions

- virtual std::size\_t **getId** () const =0
- virtual IComponent::Type **getType** () const =0
- virtual std::string **getFile** () const =0
- virtual void **setPos** (const int x, const int y)=0
- virtual sf::Vector2f **getPos** () const =0
- virtual void **setRect** (const [Rect](#) rect)=0
- virtual void **setWidth** (const int width)=0
- virtual int **getWidth** () const =0

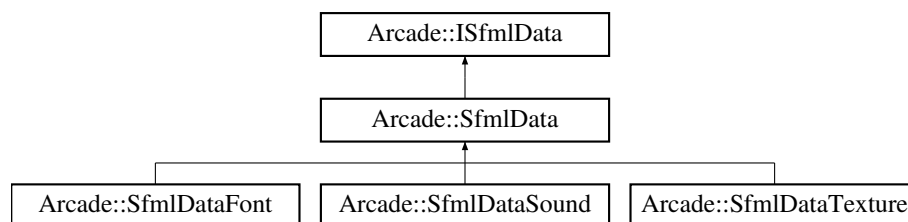
- virtual void **setHeight** (const int height)=0
- virtual int **getHeight** () const =0
- virtual void **setFontSize** (const int fontSize)=0
- virtual int **getFontSize** () const =0
- virtual sf::IntRect **getRect** () const =0
- virtual void **setText** (const std::string &text)=0
- virtual std::string **getText** () const =0
- virtual void **draw** (sf::RenderWindow &>window)=0
- virtual void **setUse** (bool use)=0
- virtual bool **isUsed** () const =0

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

- graphics/sfml/include/SfmlComponent/ISfmlComponent.hpp

### 3.28 Arcade::ISfmlData Class Reference

Inheritance diagram for Arcade::ISfmlData:



#### Public Member Functions

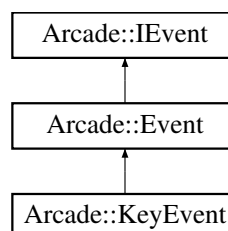
- virtual std::string **getPath** () const =0
- virtual SfmlDataType **getType** () const =0
- virtual const sf::Font & **getFont** () const =0
- virtual const sf::SoundBuffer & **getSound** () const =0
- virtual const sf::Texture & **getTexture** () const =0

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

- graphics/sfml/include/SfmlData/ISfmlData.hpp

### 3.29 Arcade::KeyEvent Class Reference

Inheritance diagram for Arcade::KeyEvent:



## Public Types

- enum **Type** { **KEY\_PRESS**, **KEY\_RELEASE**, **NONE** }

## Public Member Functions

- [KeyEvent](#) ()  
*Constructor with no arguments. Initialize \_type with Type::NONE and \_key with 0.*
- [KeyEvent](#) (Type event)  
*Constructor with the event type argument. Initialize \_type with the event given and \_key with 0.*
- [KeyEvent](#) (Type event, size\_t key)  
*Constructor with the event type and key arguments. Initialize \_type with the event given and \_key with the key given.*
- [~KeyEvent](#) ()  
*Just a destructor, nothing to see here.*

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

- core/include/KeyEvent.hpp
- core/src/KeyEvent.cpp

## 3.30 Arcade::LibHandler Class Reference

Encapsulates calls to dlopen, dlsym, dlerror and dlclose. Use it to automatically handle library changes.

```
#include <LibHandler.hpp>
```

## Classes

- class [LibraryException](#)  
*Used to throw errors with lib handling.*

## Public Member Functions

- void [loadLibrary](#) (std::string name)  
*Load a library (game | display) This automatically handles switches if a library is already loaded.*
- [IDisplay](#) \* [getDisplay](#) ()  
*Returns and instance of [IDisplay](#). Use it throughout the core loop to interact with it.*
- [IGame](#) \* [getGame](#) ()  
*Returns and instance of [IGame](#). Use it throughout the core loop to interact with it.*
- void **changeGame** (char which)
- void **changeDisplay** (char which)
- void **shutdown** ()
- const std::set< std::string > & **getValidDisplays** () const

### 3.30.1 Detailed Description

Encapsulates calls to dlopen, dlsym, dlerror and dlclose. Use it to automatically handle library changes.

### 3.30.2 Member Function Documentation

#### 3.30.2.1 `getDisplay()`

```
IDisplay * Arcade::LibHandler::getDisplay ( )
```

Returns and instance of `IDisplay`. Use it throughout the core loop to interact with it.

##### Returns

`std::shared_ptr<IDisplay>`

#### 3.30.2.2 `getGame()`

```
IGame * Arcade::LibHandler::getGame ( )
```

Returns and instance of `IGame`. Use it throughout the core loop to interact with it.

##### Returns

`std::shared_ptr<IGame>`

#### 3.30.2.3 `loadLibrary()`

```
void Arcade::LibHandler::loadLibrary (
    std::string name )
```

Load a library (game | display) This automatically handles switches if a library is already loaded.

##### Parameters

<i>name</i>	
-------------	--

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

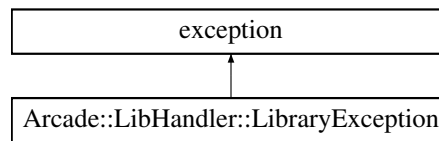
- `core/include/LibHandler.hpp`
- `core/src/LibHandler.cpp`

### 3.31 `Arcade::LibHandler::LibraryException` Class Reference

Used to throw errors with lib handling.

```
#include <LibHandler.hpp>
```

Inheritance diagram for Arcade::LibHandler::LibraryException:



## Public Member Functions

- **LibraryException** (std::string msg)
- const char \* **what** () const noexcept override

### 3.31.1 Detailed Description

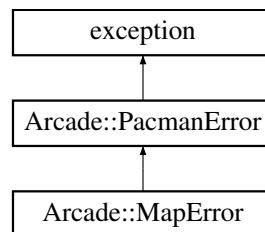
Used to throw errors with lib handling.

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

- core/include/LibHandler.hpp
- core/src/LibHandler.cpp

## 3.32 Arcade::MapError Class Reference

Inheritance diagram for Arcade::MapError:



## Public Member Functions

- **MapError** (const MapErrorType type)
- **MapError** (const MapErrorType type, const size\_t line)

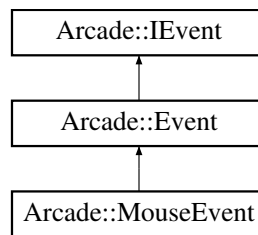
## Additional Inherited Members

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

- games/pacman/include/PacmanError/MapError.hpp
- games/pacman/src/PacmanError/MapError.cpp

### 3.33 Arcade::MouseEvent Class Reference

Inheritance diagram for Arcade::MouseEvent:



#### Public Types

- enum **Type** {  
**MOUSE\_MOVED**, **MOUSE\_ENTER**, **MOUSE\_LEAVE**, **RIGHT\_CLICK**,  
**LEFT\_CLICK**, **MIDDLE\_CLICK**, **NONE** }

#### Public Member Functions

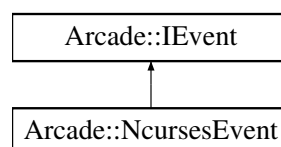
- [MouseEvent](#) ()  
*Constructor with no arguments. Initialize `_type` with `Type::NONE` and `_pos` with `{0, 0}`.*
- [MouseEvent](#) (Type event)  
*Constructor with the event type argument. Initialize `_type` with the event given and `_pos` with `{0, 0}`.*
- [MouseEvent](#) (Type event, [Pos](#) pos)  
*Constructor with the event type and mouse position arguments. Initialize `_type` with the event given and `_pos` with the position given.*
- [~MouseEvent](#) ()  
*Just a destructor, nothing to see here.*

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

- core/include/MouseEvent.hpp
- core/src/MouseEvent.cpp

### 3.34 Arcade::NcursesEvent Class Reference

Inheritance diagram for Arcade::NcursesEvent:





## Public Member Functions

- **NcursesEvent** (int event)
- `std::size_t getKey ()` const  
*Get the Key value of the event.*
- `void setKey (std::size_t const key)`  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- `Pos getMousePos ()` const  
*Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.*
- `std::string getData ()` const  
*Get Data from [Event](#).*

### 3.34.1 Member Function Documentation

#### 3.34.1.1 `getData()`

```
std::string Arcade::NcursesEvent::getData ( ) const [virtual]
```

Get Data from [Event](#).

##### Returns

`std::string`

Implements [Arcade::IEvent](#).

#### 3.34.1.2 `getKey()`

```
std::size_t Arcade::NcursesEvent::getKey ( ) const [virtual]
```

Get the Key value of the event.

##### Returns

`std::size_t`

Implements [Arcade::IEvent](#).

### 3.34.1.3 getMousePos()

```
Arcade::Pos Arcade::NcursesEvent::getMousePos ( ) const [virtual]
```

Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.

Returns

[Pos](#)

Implements [Arcade::IEvent](#).

### 3.34.1.4 setKey()

```
void Arcade::NcursesEvent::setKey (
    std::size_t const key ) [virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.

## Parameters

<i>key</i>	
------------	--

## Returns

void

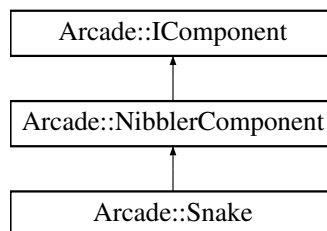
Implements [Arcade::IEvent](#).

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

- graphics/ncurses/include/NcursesEvent/NcursesEvent.hpp
- graphics/ncurses/src/NcursesEvent/NcursesEvent.cpp

## 3.35 Arcade::NibblerComponent Class Reference

Inheritance diagram for Arcade::NibblerComponent:



### Public Member Functions

- **NibblerComponent** (int id, Type type, std::string file, [Rect](#) rect, int x, int y, int w, int h)
- **NibblerComponent** (int id, std::string text, std::string file, [Rect](#) rect, int x, int y, int w, int h, int fontSize)
- std::size\_t [getId](#) () const  
*Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.*
- Type [getType](#) () const  
*Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)*
- std::string [getFile](#) () const  
*Get file associated with component Use it to create textures, etc.*
- int [getX](#) () const  
*Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- int [getY](#) () const  
*Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- virtual void [setX](#) (std::size\_t x)  
*Set the X value of the [Component](#).*
- virtual void [setY](#) (std::size\_t y)  
*Set the Y value of the [Component](#).*
- int [getWidth](#) () const  
*Get width of the OBJECT/TEXT [Component](#).*

- int [getHeight](#) () const  
*Get height of the OBJECT/TEXT [Component](#).*
- void [setWidth](#) (const int width)  
*Set the X value of the [Component](#).*
- void [setHeight](#) (const int height)  
*Set the Y value of the [Component](#).*
- [Rect](#) [getRect](#) () const  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- void [setRect](#) ([Rect](#) const rect)  
*Set the [Rect](#) of the OBJECT [Component](#).*
- std::string [getText](#) () const  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- void [setText](#) (std::string const text)  
*Set the text of the [Component](#).*
- int [getFontSize](#) () const  
*Get the font size of the TEXT [Component](#).*

## Protected Attributes

- std::size\_t **\_id**
- Type **\_type**
- std::string **\_file**
- int **\_x**
- int **\_y**
- int **\_w**
- int **\_h**
- [Rect](#) **\_rect**
- std::string **\_text**
- int **\_fontSize**

## Additional Inherited Members

### 3.35.1 Member Function Documentation

#### 3.35.1.1 [getFile](#)()

```
std::string Arcade::NibblerComponent::getFile ( ) const [virtual]
```

Get file associated with component Use it to create textures, etc.

#### Returns

std::string

Implements [Arcade::IComponent](#).

### 3.35.1.2 getFontSize()

```
int Arcade::NibblerComponent::getFontSize ( ) const [virtual]
```

Get the font size of the TEXT [Component](#).

#### Returns

int

Implements [Arcade::IComponent](#).

### 3.35.1.3 getHeight()

```
int Arcade::NibblerComponent::getHeight ( ) const [virtual]
```

Get height of the OBJECT/TEXT [Component](#).

#### Returns

int

Implements [Arcade::IComponent](#).

### 3.35.1.4 getId()

```
std::size_t Arcade::NibblerComponent::getId ( ) const [virtual]
```

Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.

#### Returns

std::size\_t

Implements [Arcade::IComponent](#).

### 3.35.1.5 getRect()

```
Rect Arcade::NibblerComponent::getRect ( ) const [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

#### Returns

[Rect](#)

Implements [Arcade::IComponent](#).

### 3.35.1.6 getText()

```
std::string Arcade::NibblerComponent::getText ( ) const [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

Returns

[Rect](#)

Get text from the component Only works with Type (TEXT). Will throw an exception if Type != TEXT

Returns

std::string

Implements [Arcade::IComponent](#).

### 3.35.1.7 getType()

```
Arcade::IComponent::Type Arcade::NibblerComponent::getType ( ) const [virtual]
```

Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)

Returns

Type

Implements [Arcade::IComponent](#).

### 3.35.1.8 getWidth()

```
int Arcade::NibblerComponent::getWidth ( ) const [virtual]
```

Get width of the OBJECT/TEXT [Component](#).

Returns

int

Implements [Arcade::IComponent](#).

### 3.35.1.9 getX()

```
int Arcade::NibblerComponent::getX ( ) const [virtual]
```

Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

#### Returns

intP

Implements [Arcade::IComponent](#).

### 3.35.1.10 getY()

```
int Arcade::NibblerComponent::getY ( ) const [virtual]
```

Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

#### Returns

int

Implements [Arcade::IComponent](#).

### 3.35.1.11 setHeight()

```
void Arcade::NibblerComponent::setHeight (
    const int height ) [virtual]
```

Set the Y value of the [Component](#).

#### Parameters

<i>y</i>	
----------	--

Implements [Arcade::IComponent](#).

### 3.35.1.12 setRect()

```
void Arcade::NibblerComponent::setRect (
    Rect const rect ) [virtual]
```

Set the [Rect](#) of the OBJECT [Component](#).

## Parameters

<i>rect</i>	
-------------	--

Implements [Arcade::IComponent](#).

**3.35.1.13 setText()**

```
void Arcade::NibblerComponent::setText (
    std::string const text ) [virtual]
```

Set the text of the [Component](#).

## Parameters

<i>text</i>	
-------------	--

Implements [Arcade::IComponent](#).

**3.35.1.14 setWidth()**

```
void Arcade::NibblerComponent::setWidth (
    const int width ) [virtual]
```

Set the X value of the [Component](#).

## Parameters

<i>x</i>	
----------	--

Implements [Arcade::IComponent](#).

**3.35.1.15 setX()**

```
void Arcade::NibblerComponent::setX (
    std::size_t x ) [virtual]
```

Set the X value of the [Component](#).

## Parameters

<i>x</i>	
----------	--



Implements [Arcade::IComponent](#).

Reimplemented in [Arcade::Snake](#).

#### 3.35.1.16 setY()

```
void Arcade::NibblerComponent::setY (
    std::size_t y ) [virtual]
```

Set the Y value of the [Component](#).

##### Parameters

<i>y</i>	
----------	--

Implements [Arcade::IComponent](#).

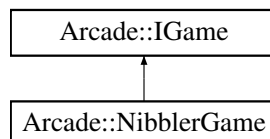
Reimplemented in [Arcade::Snake](#).

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

- games/nibbler/include/NibblerComponent.hpp
- games/nibbler/src/NibblerComponent.cpp

## 3.36 Arcade::NibblerGame Class Reference

Inheritance diagram for Arcade::NibblerGame:



### Classes

- class [NibblerGameException](#)

## Public Member Functions

- void `init` ()  
*initilize the game*
- void `stop` ()  
*cleanup the game*
- std::string `getGameName` () const  
*Get the [Game](#) name.*
- std::vector< std::unique\_ptr< [IComponent](#) > > & `getComponents` ()  
*Get the IComponents from the game Their order in the vector determines in which order they are displayed.*
- void `sendEvents` (std::vector< std::unique\_ptr< [IEvent](#) >> &events)  
*This function is called to send the events polled by the libs to the game. You can use this function as your game entrypt.*
- [IEvent](#) \* `getEvent` ()  
*Get special events from [IGame](#). It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.*
- void `setPlayerName` (std::string const name)  
*Set the Player Name.*
- std::string `getPlayerName` () const  
*Get the Player Name.*
- std::unique\_ptr< [IComponent](#) > & `getComponentByld` (int id)

### 3.36.1 Member Function Documentation

#### 3.36.1.1 getComponents()

```
std::vector< std::unique_ptr< IComponent > > & Arcade::NibblerGame::getComponents ( ) [virtual]
```

Get the IComponents from the game Their order in the vector determines in which order they are displayed.

##### Returns

std::vector<std::unique\_ptr<IComponent>>&

Implements [Arcade::IGame](#).

#### 3.36.1.2 getEvent()

```
IEvent * Arcade::NibblerGame::getEvent ( ) [virtual]
```

Get special events from [IGame](#). It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.

##### Returns

IEvent\*

Implements [Arcade::IGame](#).

### 3.36.1.3 `getGameName()`

```
std::string Arcade::NibblerGame::getGameName ( ) const [virtual]
```

Get the [Game](#) name.

#### Returns

`std::string`

Implements [Arcade::IGame](#).

### 3.36.1.4 `getPlayerName()`

```
std::string Arcade::NibblerGame::getPlayerName ( ) const [virtual]
```

Get the Player Name.

#### Returns

`std::string`

Implements [Arcade::IGame](#).

### 3.36.1.5 `sendEvents()`

```
void Arcade::NibblerGame::sendEvents (
    std::vector< std::unique_ptr< IEvent >> & events ) [virtual]
```

This function is called to send the events polled by the libs to the game. You can use this function as your game entrypoint.

#### Parameters

<i>events</i>	
---------------	--

Implements [Arcade::IGame](#).

### 3.36.1.6 `setPlayerName()`

```
void Arcade::NibblerGame::setPlayerName (
    std::string const name ) [virtual]
```

Set the Player Name.

## Parameters

<i>name</i>	
-------------	--

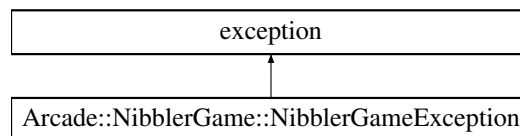
Implements [Arcade::IGame](#).

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

- games/nibbler/include/NibblerGame.hpp
- games/nibbler/src/NibblerGame.cpp

### 3.37 Arcade::NibblerGame::NibblerGameException Class Reference

Inheritance diagram for Arcade::NibblerGame::NibblerGameException:



#### Public Member Functions

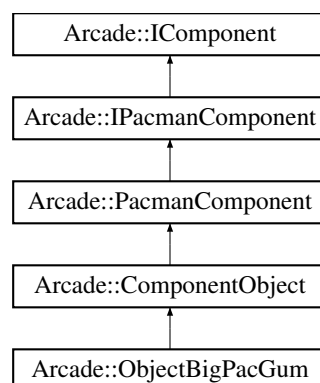
- **NibblerGameException** (std::string msg)
- const char \* **what** () const noexcept override

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

- games/nibbler/include/NibblerGame.hpp
- games/nibbler/src/NibblerGame.cpp

### 3.38 Arcade::ObjectBigPacGum Class Reference

Inheritance diagram for Arcade::ObjectBigPacGum:



## Public Member Functions

- **ObjectBigPacGum** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

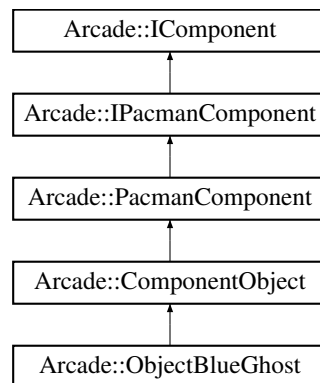
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ObjectBigPacGum.hpp
- games/pacman/src/PacmanComponent/ObjectBigPacGum.cpp

## 3.39 Arcade::ObjectBlueGhost Class Reference

Inheritance diagram for Arcade::ObjectBlueGhost:



## Public Member Functions

- **ObjectBlueGhost** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

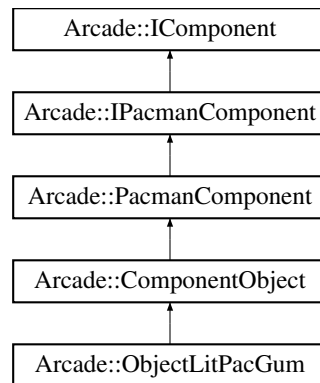
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ObjectBlueGhost.hpp
- games/pacman/src/PacmanComponent/ObjectBlueGhost.cpp

### 3.40 Arcade::ObjectLitPacGum Class Reference

Inheritance diagram for Arcade::ObjectLitPacGum:



#### Public Member Functions

- **ObjectLitPacGum** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

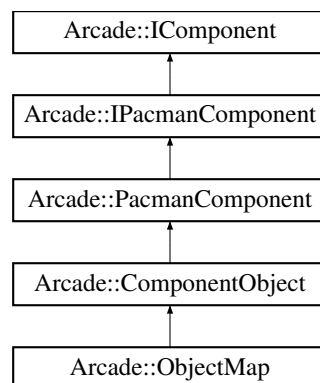
#### Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ObjectLitPacGum.hpp
- games/pacman/src/PacmanComponent/ObjectLitPacGum.cpp

### 3.41 Arcade::ObjectMap Class Reference

Inheritance diagram for Arcade::ObjectMap:



#### Public Member Functions

- **ObjectMap** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

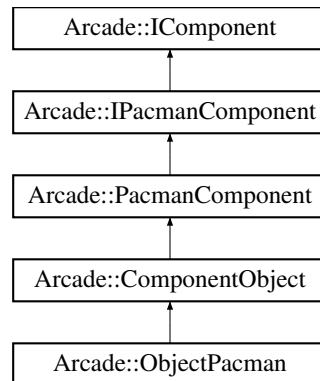
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ObjectMap.hpp
- games/pacman/src/PacmanComponent/ObjectMap.cpp

## 3.42 Arcade::ObjectPacman Class Reference

Inheritance diagram for Arcade::ObjectPacman:



## Public Member Functions

- **ObjectPacman** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

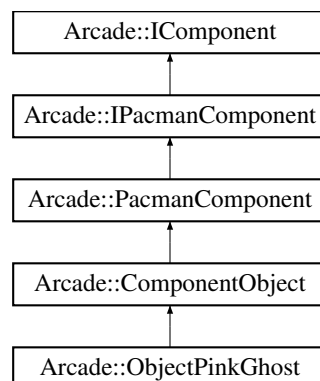
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ObjectPacman.hpp
- games/pacman/src/PacmanComponent/ObjectPacman.cpp

## 3.43 Arcade::ObjectPinkGhost Class Reference

Inheritance diagram for Arcade::ObjectPinkGhost:



## Public Member Functions

- **ObjectPinkGhost** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

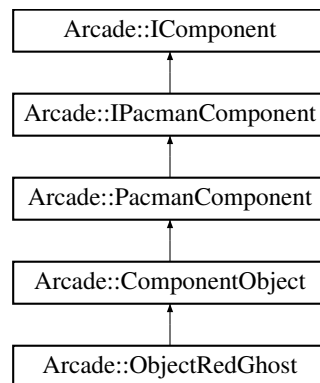
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ObjectPinkGhost.hpp
- games/pacman/src/PacmanComponent/ObjectPinkGhost.cpp

## 3.44 Arcade::ObjectRedGhost Class Reference

Inheritance diagram for Arcade::ObjectRedGhost:



## Public Member Functions

- **ObjectRedGhost** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

## Additional Inherited Members

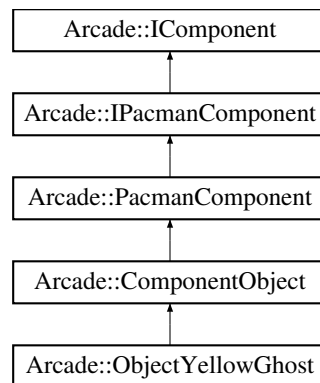
The documentation for this class was generated from the following files:

- games/pacman/include/PacmanComponent/ObjectRedGhost.hpp
- games/pacman/src/PacmanComponent/ObjectRedGhost.cpp



## 3.45 Arcade::ObjectYellowGhost Class Reference

Inheritance diagram for Arcade::ObjectYellowGhost:



### Public Member Functions

- **ObjectYellowGhost** (const std::size\_t &id, const [PacmanMap::Pos](#) &pos)

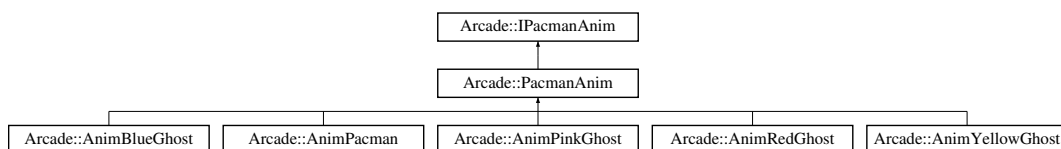
### Additional Inherited Members

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

- games/pacman/include/PacmanComponent/ObjectYellowGhost.hpp
- games/pacman/src/PacmanComponent/ObjectYellowGhost.cpp

## 3.46 Arcade::PacmanAnim Class Reference

Inheritance diagram for Arcade::PacmanAnim:



### Public Member Functions

- void **setAnim** (const RectList &rectList, const float &delay, const PacmanAnimMove &type)
- void **setCurrentAnim** (const PacmanAnimMove &currentAnim)
- PacmanAnimMove **getCurrentAnim** (void) const
- [Rect](#) **animate** ()

## Static Public Member Functions

- static IPacAnimUptr **makeAnim** (const PacmanAnimType &type)

## Protected Member Functions

- void **\_changeRectPos** (RectList &list, const PacmanAnimMove &type)

## Protected Attributes

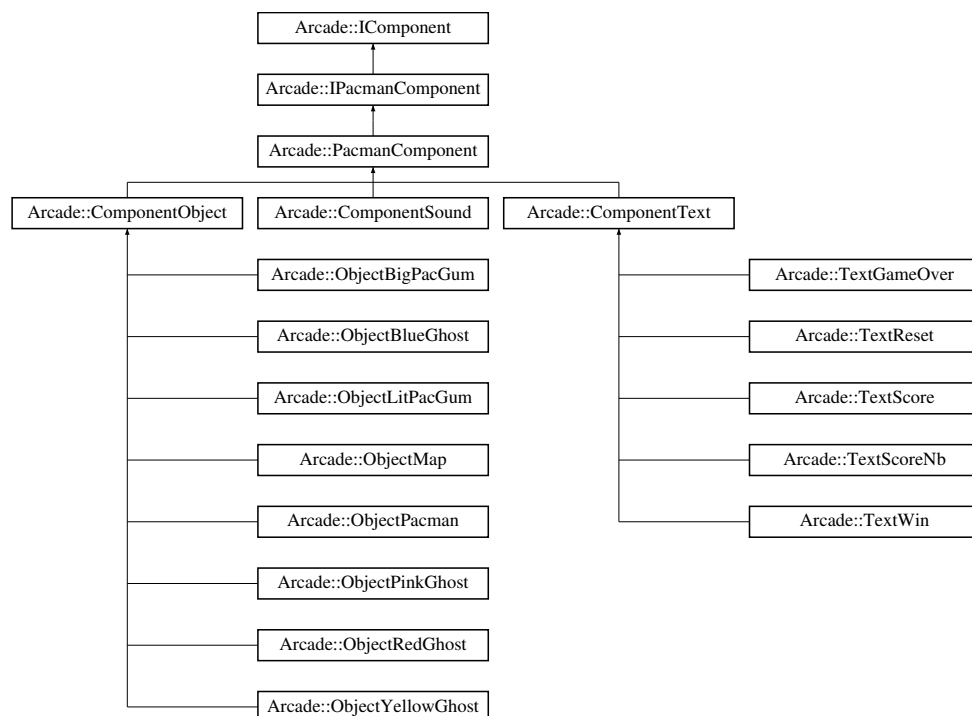
- [Rect](#) **\_currentRect**
- PacmanAnimMove **\_currentAnim**
- std::chrono::time\_point< std::chrono::high\_resolution\_clock > **\_lastTime**
- float **\_delay**
- AnimTypeRectListMap **\_map**

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

- games/pacman/include/PacmanAnim/PacmanAnim.hpp
- games/pacman/src/PacmanAnim/PacmanAnim.cpp

## 3.47 Arcade::PacmanComponent Class Reference

Inheritance diagram for Arcade::PacmanComponent:



## Public Member Functions

- **PacmanComponent** (const std::size\_t &id, const Type &type, const std::string &file)
- std::size\_t **getId** () const  
*Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.*
- Type **getType** () const  
*Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)*
- virtual ComponentObjectType **getComponentObjectType** () const
- virtual ComponentTextType **getComponentTextType** () const
- std::string **getFile** () const  
*Get file associated with component Use it to create textures, etc.*
- virtual void **setX** (std::size\_t const x)  
*Set the X value of the Component.*
- virtual int **getX** () const override  
*Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- virtual void **setY** (std::size\_t const y)  
*Set the Y value of the Component.*
- virtual int **getY** () const override  
*Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.*
- virtual void **setWidth** (const int width)  
*Set the X value of the Component.*
- virtual int **getWidth** () const override  
*Get width of the OBJECT/TEXT Component.*
- virtual void **setHeight** (const int height)  
*Set the Y value of the Component.*
- virtual int **getHeight** () const override  
*Get height of the OBJECT/TEXT Component.*
- virtual void **setRect** (Rect const rect)  
*Set the Rect of the OBJECT Component.*
- virtual Rect **getRect** () const override  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- virtual void **setFontSize** (const int size)
- virtual int **getFontSize** () const override  
*Get the font size of the TEXT Component.*
- virtual void **setText** (std::string const text)  
*Set the text of the Component.*
- virtual std::string **getText** () const override  
*Get animated sprite rect #THROW if no rect or Type != OBJECT.*
- virtual void **setObjectAnimation** (const PacmanAnimType &type)
- virtual void **setObjectCurrentAnimation** (const PacmanAnimMove &move)
- virtual void **animateObject** ()

## Static Public Member Functions

- static IPacCompUPtr **makePacmanComponent** (const std::size\_t &id, const Type &type, const std::string &file)

## Protected Attributes

- const std::size\_t **\_id**
- const Type **\_type**
- const std::string **\_file**

## Additional Inherited Members

### 3.47.1 Member Function Documentation

#### 3.47.1.1 getFile()

```
std::string Arcade::PacmanComponent::getFile ( ) const [virtual]
```

Get file associated with component Use it to create textures, etc.

##### Returns

std::string

Implements [Arcade::IPacmanComponent](#).

#### 3.47.1.2 getFontSize()

```
int Arcade::PacmanComponent::getFontSize ( ) const [override], [virtual]
```

Get the font size of the TEXT [Component](#).

##### Returns

int

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#).

#### 3.47.1.3 getHeight()

```
int Arcade::PacmanComponent::getHeight ( ) const [override], [virtual]
```

Get height of the OBJECT/TEXT [Component](#).

##### Returns

int

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

#### 3.47.1.4 getId()

```
std::size_t Arcade::PacmanComponent::getId ( ) const [virtual]
```

Get the component ID The component ID is used to identify a component, which can be useful with user/component interaction.

##### Returns

std::size\_t

Implements [Arcade::IPacmanComponent](#).

#### 3.47.1.5 getRect()

```
Arcade::Rect Arcade::PacmanComponent::getRect ( ) const [override], [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

##### Returns

[Rect](#)

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

#### 3.47.1.6 getText()

```
std::string Arcade::PacmanComponent::getText ( ) const [override], [virtual]
```

Get animated sprite rect #THROW if no rect or Type != OBJECT.

##### Returns

[Rect](#)

Get text from the component Only works with Type (TEXT). Will throw an exception if Type != TEXT

##### Returns

std::string

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#).

### 3.47.1.7 getType()

```
Arcade::IComponent::Type Arcade::PacmanComponent::getType ( ) const [virtual]
```

Get component type Use it to know how component need to be displayed (OBJECT | SOUND | TEXT)

Returns

Type

Implements [Arcade::IPacmanComponent](#).

### 3.47.1.8 getWidth()

```
int Arcade::PacmanComponent::getWidth ( ) const [override], [virtual]
```

Get width of the OBJECT/TEXT [Component](#).

Returns

int

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.47.1.9 getX()

```
int Arcade::PacmanComponent::getX ( ) const [override], [virtual]
```

Get X position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

intP

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.47.1.10 getY()

```
int Arcade::PacmanComponent::getY ( ) const [override], [virtual]
```

Get Y position of the component Only works with Type (OBJECT/TEXT). Will throw an exception if Type = SOUND.

Returns

int

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

### 3.47.1.11 setHeight()

```
void Arcade::PacmanComponent::setHeight (
    const int height ) [virtual]
```

Set the Y value of the [Component](#).

## Parameters

<i>y</i>	
----------	--

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

**3.47.1.12 setRect()**

```
void Arcade::PacmanComponent::setRect (
    Rect const rect ) [virtual]
```

Set the [Rect](#) of the OBJECT [Component](#).

## Parameters

<i>rect</i>	
-------------	--

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

**3.47.1.13 setText()**

```
void Arcade::PacmanComponent::setText (
    std::string const text ) [virtual]
```

Set the text of the [Component](#).

## Parameters

<i>text</i>	
-------------	--

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#).

**3.47.1.14 setWidth()**

```
void Arcade::PacmanComponent::setWidth (
    const int width ) [virtual]
```

Set the X value of the [Component](#).

**Parameters**

<i>x</i>	
----------	--

Implements [Arcade::IPacmanComponent](#).

Reimplemented in [Arcade::ComponentText](#), and [Arcade::ComponentObject](#).

**3.47.1.15 setX()**

```
void Arcade::PacmanComponent::setX (
    std::size_t const x ) [virtual]
```

Set the X value of the [Component](#).

**Parameters**

<i>x</i>	
----------	--

Implements [Arcade::IPacmanComponent](#).

**3.47.1.16 setY()**

```
void Arcade::PacmanComponent::setY (
    std::size_t const y ) [virtual]
```

Set the Y value of the [Component](#).

**Parameters**

<i>y</i>	
----------	--

Implements [Arcade::IPacmanComponent](#).

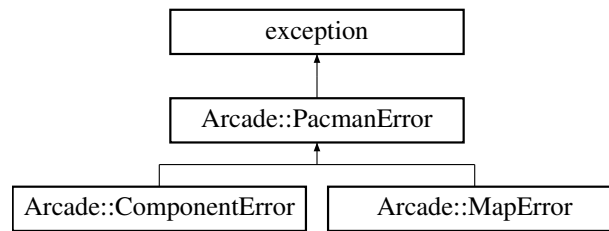
The documentation for this class was generated from the following files:

- games/pacman/include/PacmanComponent/PacmanComponent.hpp
- games/pacman/src/PacmanComponent/PacmanComponent.cpp

## 3.48 Arcade::PacmanError Class Reference

Inheritance diagram for Arcade::PacmanError:





## Public Member Functions

- **PacmanError** (const std::string &message)
- const char \* **what** () const noexcept override

## Protected Attributes

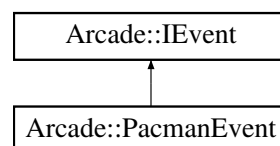
- const std::string **\_message**

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

- games/pacman/include/PacmanError/PacmanError.hpp
- games/pacman/src/PacmanError/PacmanError.cpp

## 3.49 Arcade::PacmanEvent Class Reference

Inheritance diagram for Arcade::PacmanEvent:



## Public Member Functions

- **PacmanEvent** (const std::size\_t key)
- **PacmanEvent** (const std::size\_t &key, const std::string &data)
- std::size\_t **getKey** () const  
*Get the Key value of the event.*
- void **setKey** (std::size\_t const key)  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- **Pos** **getMousePos** () const  
*Get the Mouse Position Position is expressed in the **Pos** structure {x, y}.*
- std::string **getData** () const  
*Get Data from **Event**.*

### 3.49.1 Member Function Documentation

#### 3.49.1.1 `getData()`

```
std::string Arcade::PacmanEvent::getData ( ) const [virtual]
```

Get Data from [Event](#).

##### Returns

`std::string`

Implements [Arcade::IEvent](#).

#### 3.49.1.2 `getKey()`

```
std::size_t Arcade::PacmanEvent::getKey ( ) const [virtual]
```

Get the Key value of the event.

##### Returns

`std::size_t`

Implements [Arcade::IEvent](#).

#### 3.49.1.3 `getMousePos()`

```
Arcade::Pos Arcade::PacmanEvent::getMousePos ( ) const [virtual]
```

Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.

##### Returns

[Pos](#)

Implements [Arcade::IEvent](#).

#### 3.49.1.4 `setKey()`

```
void Arcade::PacmanEvent::setKey (
    std::size_t const key ) [virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.

## Parameters

<i>key</i>	
------------	--

## Returns

void

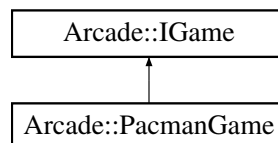
Implements [Arcade::IEvent](#).

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

- games/pacman/include/PacmanEvent/PacmanEvent.hpp
- games/pacman/src/PacmanEvent/PacmanEvent.cpp

## 3.50 Arcade::PacmanGame Class Reference

Inheritance diagram for Arcade::PacmanGame:



### Public Member Functions

- void [init](#) ()  
*initilize the game*
- void [stop](#) ()  
*cleanup the game*
- std::string [getGameName](#) () const  
*Get the [Game](#) name.*
- ICompUPtrVect & [getComponents](#) ()  
*Get the IComponents from the game Their order in the vector determines in which order they are displayed.*
- void [sendEvents](#) (IEventUPtrVect &events)
- [IEvent](#) \* [getEvent](#) ()  
*Get special events from [IGame](#). It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.*
- void [setPlayerName](#) (std::string const name)  
*Set the Player Name.*
- std::string [getPlayerName](#) () const  
*Get the Player Name.*

### 3.50.1 Member Function Documentation

### 3.50.1.1 getComponents()

```
Arcade::ICompUPtrVect & Arcade::PacmanGame::getComponents ( ) [virtual]
```

Get the IComponents from the game Their order in the vector determines in which order they are displayed.

#### Returns

std::vector<std::unique\_ptr<IComponent>>&

Implements [Arcade::IGame](#).

### 3.50.1.2 getEvent()

```
Arcade::IEvent * Arcade::PacmanGame::getEvent ( ) [virtual]
```

Get special events from [IGame](#). It can be used to pass back data to the arcade core, and to indicate maybe a lib change call.

#### Returns

IEvent\*

Implements [Arcade::IGame](#).

### 3.50.1.3 getGameName()

```
std::string Arcade::PacmanGame::getGameName ( ) const [virtual]
```

Get the [Game](#) name.

#### Returns

std::string

Implements [Arcade::IGame](#).

### 3.50.1.4 getPlayerName()

```
std::string Arcade::PacmanGame::getPlayerName ( ) const [virtual]
```

Get the Player Name.

#### Returns

std::string

Implements [Arcade::IGame](#).

### 3.50.1.5 setPlayerName()

```
void Arcade::PacmanGame::setPlayerName (
    std::string const name ) [virtual]
```

Set the Player Name.

## Parameters

<i>name</i>	
-------------	--

Implements [Arcade::IGame](#).

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

- games/pacman/include/PacmanGame/PacmanGame.hpp
- games/pacman/src/PacmanGame/PacmanGame.cpp

## 3.51 Arcade::PacmanMap Class Reference

### Classes

- struct [Pos](#)
- struct [SPos](#)

### Public Types

- enum **PlayerMove** {  
    **LEFT, RIGHT, UP, DOWN,**  
    **DEATH, WEEKNESS** }

### Public Member Functions

- [Pos](#) **getPlayerPos** () const
- [Pos](#) **getEnemyPos** (const size\_t &id) const
- bool **movePlayer** (const PlayerMove &move)
- void **moveEnemies** ()
- PlayerMove **getPlayerMove** () const
- PlayerMove **getEnemyMove** (const size\_t &id) const
- void **computeBigPacGums** (IPacCompUPtrVect &components, size\_t &compCompt, size\_t &gumsCount)
- void **computeLittlePacGums** (IPacCompUPtrVect &components, size\_t &compCompt, size\_t &gumsCount)
- void **checkGumsHitbox** (IPacCompUPtrVect &components, size\_t &score, size\_t &gumsCount)

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

- games/pacman/include/PacmanMap/PacmanMap.hpp
- games/pacman/src/PacmanMap/PacmanMap.cpp

## 3.52 Arcade::PacmanScore Class Reference

### Public Member Functions

- void **addScore** (std::string name, int score)
- std::map< std::string, int > **getScores** () const
- int **getScore** (const std::string name) const
- void **saveScores** ()
- void **loadScores** (std::string game)

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

- games/pacman/include/PacmanScore/PacmanScore.hpp
- games/pacman/src/PacmanScore/PacmanScore.cpp

## 3.53 Arcade::Pos Struct Reference

Used for mouse position.

```
#include <IEvent.hpp>
```

### Public Attributes

- int **x**
- int **y**

### 3.53.1 Detailed Description

Used for mouse position.

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

- include/IEvent.hpp

## 3.54 Arcade::PacmanMap::Pos Struct Reference

### Public Attributes

- int **x**
- int **y**

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

- games/pacman/include/PacmanMap/PacmanMap.hpp

## 3.55 Arcade::Rect Struct Reference

The [Rect](#) struct give information on a sprite.

```
#include <IComponent.hpp>
```

### Public Attributes

- int **x**
- int **y**
- int **width**
- int **height**

### 3.55.1 Detailed Description

The [Rect](#) struct give information on a sprite.

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

- include/IComponent.hpp

## 3.56 Arcade::ScoreBoard Class Reference

### Public Member Functions

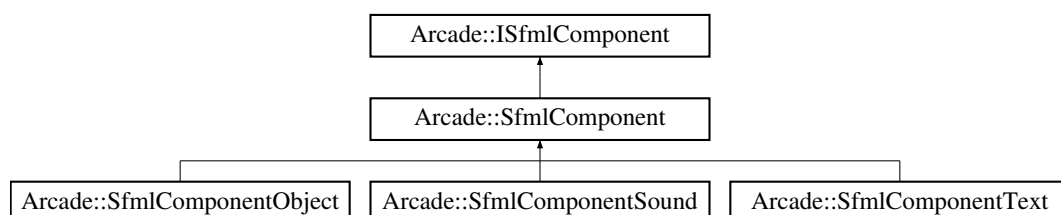
- void **addScore** (std::string name, int score)
- std::map< std::string, int > **getScores** () const
- void **saveScores** ()
- void **loadScores** (std::string game)

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

- games/nibbler/include/ScoreBoard.hpp
- games/nibbler/src/ScoreBoard.cpp

## 3.57 Arcade::SfmlComponent Class Reference

Inheritance diagram for Arcade::SfmlComponent:



## Public Member Functions

- **SfmlComponent** (const size\_t &id, const IComponent::Type &type, const std::unique\_ptr< [ISfmlData](#) > &data)
- std::size\_t **getId** () const
- IComponent::Type **getType** () const
- std::string **getFile** () const
- virtual void **setPos** (const int x, const int y)
- virtual sf::Vector2f **getPos** () const
- virtual void **setRect** (const [Rect](#) rect)
- virtual sf::IntRect **getRect** () const
- virtual void **setWidth** (const int width)
- virtual int **getWidth** () const
- virtual void **setHeight** (const int height)
- virtual int **getHeight** () const
- virtual void **setFontSize** (const int fontSize)
- virtual int **getFontSize** () const
- virtual void **setText** (const std::string &text)
- virtual std::string **getText** () const
- virtual void **draw** (sf::RenderWindow &window)
- void **setUse** (bool use)
- bool **isUsed** () const

## Static Public Member Functions

- static std::unique\_ptr< [ISfmlComponent](#) > **sfmlComponentFactory** (const size\_t &id, const IComponent::Type &type, const std::unique\_ptr< [ISfmlData](#) > &data)

## Protected Attributes

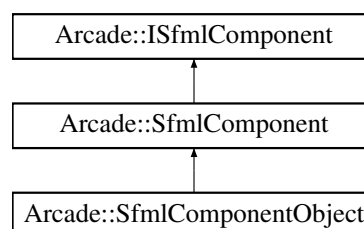
- const size\_t **\_id**
- const IComponent::Type **\_type**
- const std::unique\_ptr< [ISfmlData](#) > & **\_data**
- bool **\_use**

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

- graphics/sfml/include/SfmlComponent/SfmlComponent.hpp
- graphics/sfml/src/SfmlComponent/SfmlComponent.cpp

## 3.58 Arcade::SfmlComponentObject Class Reference

Inheritance diagram for Arcade::SfmlComponentObject:





## Public Member Functions

- **SfmlComponentObject** (const size\_t &id, const std::unique\_ptr< [ISfmlData](#) > &data)
- void **setPos** (const int x, const int y)
- sf::Vector2f **getPos** () const override
- void **setRect** (const [Rect](#) rect)
- sf::IntRect **getRect** () const override
- void **setWidth** (const int width)
- int **getWidth** () const
- void **setHeight** (const int height)
- int **getHeight** () const
- void **draw** (sf::RenderWindow &window)

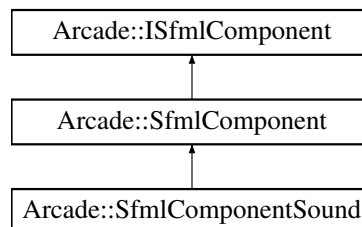
## Additional Inherited Members

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

- graphics/sfml/include/SfmlComponent/SfmlComponentObject.hpp
- graphics/sfml/src/SfmlComponent/SfmlComponentObject.cpp

## 3.59 Arcade::SfmlComponentSound Class Reference

Inheritance diagram for Arcade::SfmlComponentSound:



## Public Member Functions

- **SfmlComponentSound** (const size\_t &id, const std::unique\_ptr< [ISfmlData](#) > &data)

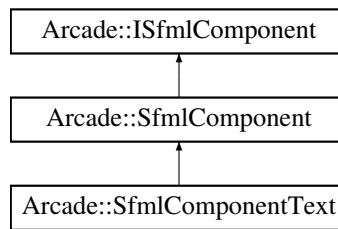
## Additional Inherited Members

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

- graphics/sfml/include/SfmlComponent/SfmlComponentSound.hpp
- graphics/sfml/src/SfmlComponent/SfmlComponentSound.cpp

### 3.60 Arcade::SfmlComponentText Class Reference

Inheritance diagram for Arcade::SfmlComponentText:



#### Public Member Functions

- **SfmlComponentText** (const size\_t &id, const std::unique\_ptr< [ISfmlData](#) > &data)
- void **setPos** (const int x, const int y)
- sf::Vector2f **getPos** () const override
- void **setFontSize** (const int fontSize)
- int **getFontSize** () const
- void **setText** (const std::string &text)
- std::string **getText** () const override
- void **draw** (sf::RenderWindow &window)

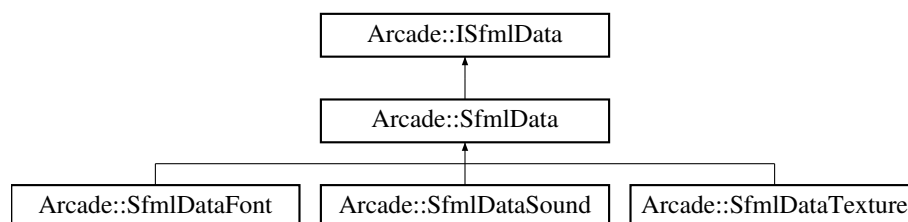
#### Additional Inherited Members

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

- graphics/sfml/include/SfmlComponent/SfmlComponentText.hpp
- graphics/sfml/src/SfmlComponent/SfmlComponentText.cpp

### 3.61 Arcade::SfmlData Class Reference

Inheritance diagram for Arcade::SfmlData:



#### Public Member Functions

- **SfmlData** (const std::string &path, const SfmlDataType &type)
- std::string **getPath** () const
- SfmlDataType **getType** () const
- virtual const sf::Font & **getFont** () const
- virtual const sf::SoundBuffer & **getSound** () const
- virtual const sf::Texture & **getTexture** () const

### Static Public Member Functions

- static ISfmlDataUPtr **SfmlDataFactory** (const std::string &path, const IComponent::Type &type)

### Protected Attributes

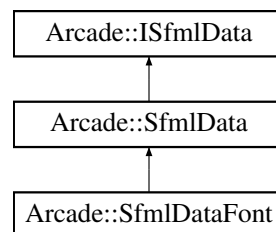
- const std::string **\_path**
- const SfmlDataType **\_type**

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

- graphics/sfml/include/SfmlData/SfmlData.hpp
- graphics/sfml/src/SfmlData/SfmlData.cpp

## 3.62 Arcade::SfmlDataFont Class Reference

Inheritance diagram for Arcade::SfmlDataFont:



### Public Member Functions

- **SfmlDataFont** (const std::string &path)
- const sf::Font & **getFont** () const

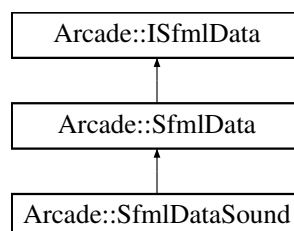
### Additional Inherited Members

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

- graphics/sfml/include/SfmlData/SfmlDataFont.hpp
- graphics/sfml/src/SfmlData/SfmlDataFont.cpp

## 3.63 Arcade::SfmlDataSound Class Reference

Inheritance diagram for Arcade::SfmlDataSound:



## Public Member Functions

- **SfmlDataSound** (const std::string &path)
- const sf::SoundBuffer & **getSound** () const

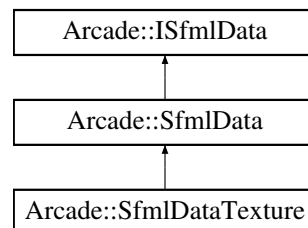
## Additional Inherited Members

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

- graphics/sfml/include/SfmlData/SfmlDataSound.hpp
- graphics/sfml/src/SfmlData/SfmlDataSound.cpp

## 3.64 Arcade::SfmlDataTexture Class Reference

Inheritance diagram for Arcade::SfmlDataTexture:



## Public Member Functions

- **SfmlDataTexture** (const std::string &path)
- const sf::Texture & **getTexture** () const

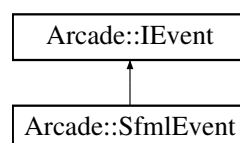
## Additional Inherited Members

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

- graphics/sfml/include/SfmlData/SfmlDataTexture.hpp
- graphics/sfml/src/SfmlData/SfmlDataTexture.cpp

## 3.65 Arcade::SfmlEvent Class Reference

Inheritance diagram for Arcade::SfmlEvent:



## Public Member Functions

- **SfmlEvent** (const sf::Event &event, const sf::RenderWindow &window)
- std::size\_t [getKey](#) () const  
*Get the Key value of the event.*
- void [setKey](#) (std::size\_t const key)  
*Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.*
- [Pos](#) [getMousePos](#) () const  
*Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.*
- std::string [getData](#) () const  
*Get Data from [Event](#).*

### 3.65.1 Member Function Documentation

#### 3.65.1.1 [getData\(\)](#)

```
std::string Arcade::SfmlEvent::getData ( ) const [virtual]
```

Get Data from [Event](#).

##### Returns

std::string

Implements [Arcade::IEvent](#).

#### 3.65.1.2 [getKey\(\)](#)

```
std::size_t Arcade::SfmlEvent::getKey ( ) const [virtual]
```

Get the Key value of the event.

##### Returns

std::size\_t

Implements [Arcade::IEvent](#).

### 3.65.1.3 `getMousePos()`

```
Arcade::Pos Arcade::SfmlEvent::getMousePos ( ) const [virtual]
```

Get the Mouse Position Position is expressed in the [Pos](#) structure {x, y}.

Returns

[Pos](#)

Implements [Arcade::IEvent](#).

### 3.65.1.4 `setKey()`

```
void Arcade::SfmlEvent::setKey (
    std::size_t const key ) [virtual]
```

Set the Key value of the event. This is only to be used by the core, to uniformize the value between different libraries.

## Parameters

<i>key</i>	
------------	--

## Returns

void

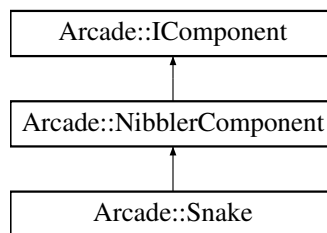
Implements [Arcade::IEvent](#).

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

- graphics/sfml/include/SfmlEvent/SfmlEvent.hpp
- graphics/sfml/src/SfmlEvent/SfmlEvent.cpp

## 3.66 Arcade::Snake Class Reference

Inheritance diagram for Arcade::Snake:



### Public Types

- enum **direction** { **up**, **down**, **left**, **right** }

### Public Member Functions

- **Snake** (int id, [NibblerGame](#) \*game)
- void **setX** (std::size\_t x) override  
*Set the X value of the [Component](#).*
- void **setY** (std::size\_t y) override  
*Set the Y value of the [Component](#).*
- std::size\_t **getLastX** () const
- std::size\_t **getLastY** () const
- void **addTail** (int id)
- void **operator+=** (Snake::direction dir)
- void **setNextMove** (Snake::direction dir)
- direction **getNextMove** () const
- intMap **getEmplacement** () const
- bool **lost** () const

## Additional Inherited Members

### 3.66.1 Member Function Documentation

#### 3.66.1.1 setX()

```
void Arcade::Snake::setX (
    std::size_t x ) [override], [virtual]
```

Set the X value of the [Component](#).

##### Parameters

<i>x</i>	
----------	--

Reimplemented from [Arcade::NibblerComponent](#).

#### 3.66.1.2 setY()

```
void Arcade::Snake::setY (
    std::size_t y ) [override], [virtual]
```

Set the Y value of the [Component](#).

##### Parameters

<i>y</i>	
----------	--

Reimplemented from [Arcade::NibblerComponent](#).

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

- games/nibbler/include/Snake.hpp
- games/nibbler/src/Snake.cpp

## 3.67 Arcade::PacmanMap::SPos Struct Reference

### Public Attributes

- `size_t x`
- `size_t y`

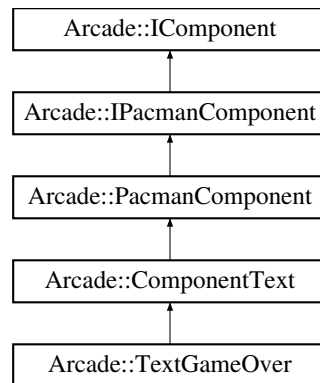
The documentation for this struct was generated from the following file:

- games/pacman/include/PacmanMap/PacmanMap.hpp



## 3.68 Arcade::TextGameOver Class Reference

Inheritance diagram for Arcade::TextGameOver:



### Public Member Functions

- **TextGameOver** (const std::size\_t &id)

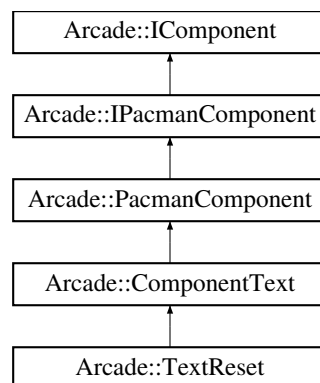
### Additional Inherited Members

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

- games/pacman/include/PacmanComponent/TextGameOver.hpp
- games/pacman/src/PacmanComponent/TextGameOver.cpp

## 3.69 Arcade::TextReset Class Reference

Inheritance diagram for Arcade::TextReset:



### Public Member Functions

- **TextReset** (const std::size\_t &id)

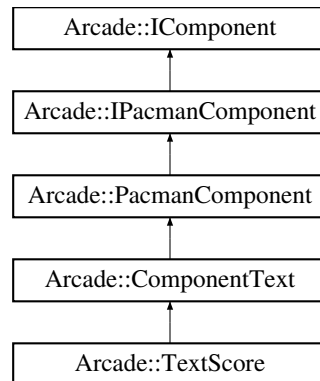
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/TextReset.hpp
- games/pacman/src/PacmanComponent/TextReset.cpp

## 3.70 Arcade::TextScore Class Reference

Inheritance diagram for Arcade::TextScore:



## Public Member Functions

- **TextScore** (const std::size\_t &id)

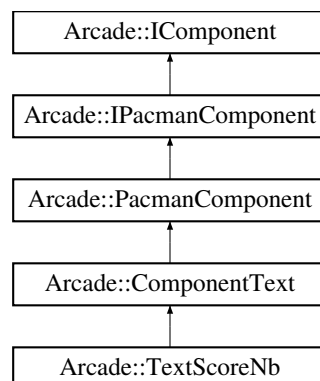
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/TextScore.hpp
- games/pacman/src/PacmanComponent/TextScore.cpp

## 3.71 Arcade::TextScoreNb Class Reference

Inheritance diagram for Arcade::TextScoreNb:



## Public Member Functions

- **TextScoreNb** (const std::size\_t &id)

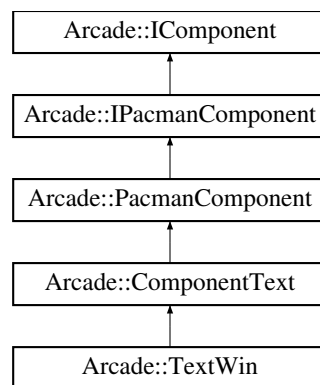
## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/TextScoreNb.hpp
- games/pacman/src/PacmanComponent/TextScoreNb.cpp

## 3.72 Arcade::TextWin Class Reference

Inheritance diagram for Arcade::TextWin:



## Public Member Functions

- **TextWin** (const std::size\_t &id)

## Additional Inherited Members

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

- games/pacman/include/PacmanComponent/TextWin.hpp
- games/pacman/src/PacmanComponent/TextWin.cpp

## 3.73 Arcade::Timer Class Reference

## Public Member Functions

- void **start** ()
- void **pause** ()
- void **reset** ()
- std::chrono::duration< double > **getElapsedTime** () const

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

- games/nibbler/include/Timer.hpp
- games/nibbler/src/Timer.cpp



# Index

- Arcade::AnimBlueGhost, [5](#)
- Arcade::AnimPacman, [5](#)
- Arcade::AnimPinkGhost, [6](#)
- Arcade::AnimRedGhost, [6](#)
- Arcade::AnimYellowGhost, [7](#)
- Arcade::ArcadeNcurses, [7](#)
  - getEvents, [7](#)
  - getLibName, [8](#)
- Arcade::ArcadeSfml, [8](#)
  - getEvents, [9](#)
  - getLibName, [9](#)
- Arcade::Component, [9](#)
  - getFile, [10](#)
  - getFontSize, [11](#)
  - getHeight, [11](#)
  - getId, [11](#)
  - getRect, [11](#)
  - getText, [12](#)
  - getType, [12](#)
  - getWidth, [12](#)
  - getX, [13](#)
  - getY, [13](#)
  - setHeight, [13](#)
  - setRect, [14](#)
  - setText, [14](#)
  - setWidth, [14](#)
  - setX, [15](#)
  - setY, [15](#)
- Arcade::ComponentError, [15](#)
- Arcade::ComponentObject, [16](#)
  - getHeight, [17](#)
  - getRect, [17](#)
  - getWidth, [18](#)
  - getX, [18](#)
  - getY, [18](#)
  - setHeight, [18](#)
  - setRect, [19](#)
  - setWidth, [19](#)
- Arcade::ComponentSound, [19](#)
- Arcade::ComponentText, [20](#)
  - getFontSize, [21](#)
  - getHeight, [21](#)
  - getRect, [21](#)
  - getText, [22](#)
  - getWidth, [22](#)
  - getX, [22](#)
  - getY, [23](#)
  - setHeight, [23](#)
  - setRect, [23](#)
  - setText, [24](#)
  - setWidth, [24](#)
- Arcade::Core, [24](#)
- Arcade::Event, [27](#)
  - getKey, [28](#)
  - getMousePos, [28](#)
  - setKey, [28](#)
- Arcade::Game, [34](#)
  - getComponents, [34](#)
  - getEvent, [34](#)
  - getGameName, [35](#)
  - getPlayerName, [35](#)
  - sendEvents, [35](#)
  - setPlayerName, [36](#)
- Arcade::IComponent, [36](#)
  - getFile, [38](#)
  - getFontSize, [38](#)
  - getHeight, [38](#)
  - getId, [39](#)
  - getRect, [39](#)
  - getText, [39](#)
  - getType, [40](#)
  - getWidth, [40](#)
  - getX, [40](#)
  - getY, [41](#)
  - setHeight, [41](#)
  - setRect, [41](#)
  - setText, [42](#)
  - setWidth, [42](#)
  - setX, [42](#)
  - setY, [43](#)
- Arcade::IDisplay, [43](#)
  - display, [44](#)
  - getEvents, [44](#)
  - getLibName, [44](#)
- Arcade::IEvent, [45](#)
  - getData, [45](#)
  - getKey, [46](#)
  - getMousePos, [46](#)
  - setKey, [46](#)
- Arcade::IGame, [47](#)
  - getComponents, [48](#)
  - getEvent, [48](#)
  - getGameName, [48](#)
  - getPlayerName, [48](#)
  - sendEvents, [49](#)
  - setPlayerName, [49](#)
- Arcade::IPacmanAnim, [50](#)
- Arcade::IPacmanComponent, [50](#)

- getFile, [51](#)
- getFontSize, [52](#)
- getHeight, [52](#)
- getId, [52](#)
- getRect, [53](#)
- getText, [53](#)
- getType, [53](#)
- getWidth, [54](#)
- getX, [54](#)
- getY, [54](#)
- setHeight, [55](#)
- setRect, [55](#)
- setText, [55](#)
- setWidth, [56](#)
- setX, [56](#)
- setY, [56](#)
- Arcade::IPacmanMap, [57](#)
- Arcade::ISfmlComponent, [57](#)
- Arcade::ISfmlData, [58](#)
- Arcade::KeyEvent, [58](#)
- Arcade::LibHandler, [59](#)
  - getDisplay, [60](#)
  - getGame, [60](#)
  - loadLibrary, [60](#)
- Arcade::LibHandler::LibraryException, [60](#)
- Arcade::MapError, [61](#)
- Arcade::MouseEvent, [62](#)
- Arcade::NcursesEvent, [62](#)
  - getData, [63](#)
  - getKey, [63](#)
  - getMousePos, [63](#)
  - setKey, [64](#)
- Arcade::NibblerComponent, [65](#)
  - getFile, [66](#)
  - getFontSize, [66](#)
  - getHeight, [67](#)
  - getId, [67](#)
  - getRect, [67](#)
  - getText, [67](#)
  - getType, [68](#)
  - getWidth, [68](#)
  - getX, [68](#)
  - getY, [69](#)
  - setHeight, [69](#)
  - setRect, [69](#)
  - setText, [70](#)
  - setWidth, [70](#)
  - setX, [70](#)
  - setY, [71](#)
- Arcade::NibblerGame, [71](#)
  - getComponents, [72](#)
  - getEvent, [72](#)
  - getGameName, [72](#)
  - getPlayerName, [73](#)
  - sendEvents, [73](#)
  - setPlayerName, [73](#)
- Arcade::NibblerGame::NibblerGameException, [74](#)
- Arcade::ObjectBigPacGum, [74](#)
- Arcade::ObjectBlueGhost, [75](#)
- Arcade::ObjectLitPacGum, [76](#)
- Arcade::ObjectMap, [76](#)
- Arcade::ObjectPacman, [77](#)
- Arcade::ObjectPinkGhost, [77](#)
- Arcade::ObjectRedGhost, [78](#)
- Arcade::ObjectYellowGhost, [79](#)
- Arcade::PacmanAnim, [79](#)
- Arcade::PacmanComponent, [80](#)
  - getFile, [82](#)
  - getFontSize, [82](#)
  - getHeight, [82](#)
  - getId, [82](#)
  - getRect, [83](#)
  - getText, [83](#)
  - getType, [83](#)
  - getWidth, [84](#)
  - getX, [84](#)
  - getY, [84](#)
  - setHeight, [84](#)
  - setRect, [85](#)
  - setText, [85](#)
  - setWidth, [85](#)
  - setX, [86](#)
  - setY, [86](#)
- Arcade::PacmanError, [86](#)
- Arcade::PacmanEvent, [87](#)
  - getData, [88](#)
  - getKey, [88](#)
  - getMousePos, [88](#)
  - setKey, [88](#)
- Arcade::PacmanGame, [89](#)
  - getComponents, [89](#)
  - getEvent, [90](#)
  - getGameName, [90](#)
  - getPlayerName, [90](#)
  - setPlayerName, [90](#)
- Arcade::PacmanMap, [91](#)
- Arcade::PacmanMap::Pos, [92](#)
- Arcade::PacmanMap::SPos, [102](#)
- Arcade::PacmanScore, [92](#)
- Arcade::Pos, [92](#)
- Arcade::Rect, [93](#)
- Arcade::ScoreBoard, [93](#)
- Arcade::SfmlComponent, [93](#)
- Arcade::SfmlComponentObject, [94](#)
- Arcade::SfmlComponentSound, [95](#)
- Arcade::SfmlComponentText, [96](#)
- Arcade::SfmlData, [96](#)
- Arcade::SfmlDataFont, [97](#)
- Arcade::SfmlDataSound, [97](#)
- Arcade::SfmlDataTexture, [98](#)
- Arcade::SfmlEvent, [98](#)
  - getData, [99](#)
  - getKey, [99](#)
  - getMousePos, [99](#)
  - setKey, [100](#)
- Arcade::Snake, [101](#)

- setX, [102](#)
- setY, [102](#)
- Arcade::TextGameOver, [103](#)
- Arcade::TextReset, [103](#)
- Arcade::TextScore, [104](#)
- Arcade::TextScoreNb, [104](#)
- Arcade::TextWin, [105](#)
- Arcade::Timer, [105](#)
- Display, [25](#)
  - display, [25](#)
  - getEvents, [26](#)
  - getLibName, [26](#)
- display
  - Arcade::IDisplay, [44](#)
  - Display, [25](#)
- Display::DisplayException, [26](#)
- Event, [29](#)
  - getData, [30](#)
  - getKey, [30](#), [31](#)
  - getMousePos, [31](#), [32](#)
  - setKey, [32](#), [33](#)
- EventNormalizer, [33](#)
- getComponents
  - Arcade::Game, [34](#)
  - Arcade::IGame, [48](#)
  - Arcade::NibblerGame, [72](#)
  - Arcade::PacmanGame, [89](#)
- getData
  - Arcade::IEvent, [45](#)
  - Arcade::NcursesEvent, [63](#)
  - Arcade::PacmanEvent, [88](#)
  - Arcade::SfmlEvent, [99](#)
  - Event, [30](#)
- getDisplay
  - Arcade::LibHandler, [60](#)
- getEvent
  - Arcade::Game, [34](#)
  - Arcade::IGame, [48](#)
  - Arcade::NibblerGame, [72](#)
  - Arcade::PacmanGame, [90](#)
- getEvents
  - Arcade::ArcadeNcurses, [7](#)
  - Arcade::ArcadeSfml, [9](#)
  - Arcade::IDisplay, [44](#)
  - Display, [26](#)
- getFile
  - Arcade::Component, [10](#)
  - Arcade::IComponent, [38](#)
  - Arcade::IPacmanComponent, [51](#)
  - Arcade::NibblerComponent, [66](#)
  - Arcade::PacmanComponent, [82](#)
- getFontSize
  - Arcade::Component, [11](#)
  - Arcade::ComponentText, [21](#)
  - Arcade::IComponent, [38](#)
  - Arcade::IPacmanComponent, [52](#)
- Arcade::NibblerComponent, [66](#)
- Arcade::PacmanComponent, [82](#)
- getGame
  - Arcade::LibHandler, [60](#)
- getGameName
  - Arcade::Game, [35](#)
  - Arcade::IGame, [48](#)
  - Arcade::NibblerGame, [72](#)
  - Arcade::PacmanGame, [90](#)
- getHeight
  - Arcade::Component, [11](#)
  - Arcade::ComponentObject, [17](#)
  - Arcade::ComponentText, [21](#)
  - Arcade::IComponent, [38](#)
  - Arcade::IPacmanComponent, [52](#)
  - Arcade::NibblerComponent, [67](#)
  - Arcade::PacmanComponent, [82](#)
- getId
  - Arcade::Component, [11](#)
  - Arcade::IComponent, [39](#)
  - Arcade::IPacmanComponent, [52](#)
  - Arcade::NibblerComponent, [67](#)
  - Arcade::PacmanComponent, [82](#)
- getKey
  - Arcade::Event, [28](#)
  - Arcade::IEvent, [46](#)
  - Arcade::NcursesEvent, [63](#)
  - Arcade::PacmanEvent, [88](#)
  - Arcade::SfmlEvent, [99](#)
  - Event, [30](#), [31](#)
- getLibName
  - Arcade::ArcadeNcurses, [8](#)
  - Arcade::ArcadeSfml, [9](#)
  - Arcade::IDisplay, [44](#)
  - Display, [26](#)
- getMousePos
  - Arcade::Event, [28](#)
  - Arcade::IEvent, [46](#)
  - Arcade::NcursesEvent, [63](#)
  - Arcade::PacmanEvent, [88](#)
  - Arcade::SfmlEvent, [99](#)
  - Event, [31](#), [32](#)
- getPlayerName
  - Arcade::Game, [35](#)
  - Arcade::IGame, [48](#)
  - Arcade::NibblerGame, [73](#)
  - Arcade::PacmanGame, [90](#)
- getRect
  - Arcade::Component, [11](#)
  - Arcade::ComponentObject, [17](#)
  - Arcade::ComponentText, [21](#)
  - Arcade::IComponent, [39](#)
  - Arcade::IPacmanComponent, [53](#)
  - Arcade::NibblerComponent, [67](#)
  - Arcade::PacmanComponent, [83](#)
- getText
  - Arcade::Component, [12](#)
  - Arcade::ComponentText, [22](#)

- Arcade::IComponent, 39
- Arcade::IPacmanComponent, 53
- Arcade::NibblerComponent, 67
- Arcade::PacmanComponent, 83
- getType
  - Arcade::Component, 12
  - Arcade::IComponent, 40
  - Arcade::IPacmanComponent, 53
  - Arcade::NibblerComponent, 68
  - Arcade::PacmanComponent, 83
- getWidth
  - Arcade::Component, 12
  - Arcade::ComponentObject, 18
  - Arcade::ComponentText, 22
  - Arcade::IComponent, 40
  - Arcade::IPacmanComponent, 54
  - Arcade::NibblerComponent, 68
  - Arcade::PacmanComponent, 84
- getX
  - Arcade::Component, 13
  - Arcade::ComponentObject, 18
  - Arcade::ComponentText, 22
  - Arcade::IComponent, 40
  - Arcade::IPacmanComponent, 54
  - Arcade::NibblerComponent, 68
  - Arcade::PacmanComponent, 84
- getY
  - Arcade::Component, 13
  - Arcade::ComponentObject, 18
  - Arcade::ComponentText, 23
  - Arcade::IComponent, 41
  - Arcade::IPacmanComponent, 54
  - Arcade::NibblerComponent, 69
  - Arcade::PacmanComponent, 84
- loadLibrary
  - Arcade::LibHandler, 60
- sendEvents
  - Arcade::Game, 35
  - Arcade::IGame, 49
  - Arcade::NibblerGame, 73
- setHeight
  - Arcade::Component, 13
  - Arcade::ComponentObject, 18
  - Arcade::ComponentText, 23
  - Arcade::IComponent, 41
  - Arcade::IPacmanComponent, 55
  - Arcade::NibblerComponent, 69
  - Arcade::PacmanComponent, 84
- setKey
  - Arcade::Event, 28
  - Arcade::IEvent, 46
  - Arcade::NcursesEvent, 64
  - Arcade::PacmanEvent, 88
  - Arcade::SfmlEvent, 100
  - Event, 32, 33
- setPlayerName
  - Arcade::Game, 36
- Arcade::IGame, 49
- Arcade::NibblerGame, 73
- Arcade::PacmanGame, 90
- setRect
  - Arcade::Component, 14
  - Arcade::ComponentObject, 19
  - Arcade::ComponentText, 23
  - Arcade::IComponent, 41
  - Arcade::IPacmanComponent, 55
  - Arcade::NibblerComponent, 69
  - Arcade::PacmanComponent, 85
- setText
  - Arcade::Component, 14
  - Arcade::ComponentText, 24
  - Arcade::IComponent, 42
  - Arcade::IPacmanComponent, 55
  - Arcade::NibblerComponent, 70
  - Arcade::PacmanComponent, 85
- setWidth
  - Arcade::Component, 14
  - Arcade::ComponentObject, 19
  - Arcade::ComponentText, 24
  - Arcade::IComponent, 42
  - Arcade::IPacmanComponent, 56
  - Arcade::NibblerComponent, 70
  - Arcade::PacmanComponent, 85
- setX
  - Arcade::Component, 15
  - Arcade::IComponent, 42
  - Arcade::IPacmanComponent, 56
  - Arcade::NibblerComponent, 70
  - Arcade::PacmanComponent, 86
  - Arcade::Snake, 102
- setY
  - Arcade::Component, 15
  - Arcade::IComponent, 43
  - Arcade::IPacmanComponent, 56
  - Arcade::NibblerComponent, 71
  - Arcade::PacmanComponent, 86
  - Arcade::Snake, 102