

arcade

0.1.0

Generated by Doxygen 1.9.1



<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 Arcade Namespace Reference	9
5.1.1 Detailed Description	10
5.1.2 Enumeration Type Documentation	10
5.1.2.1 GameEvent	10
<b>6 Class Documentation</b>	<b>11</b>
6.1 Arcade::Core Class Reference	11
6.1.1 Detailed Description	12
6.1.2 Member Function Documentation	12
6.1.2.1 gameLoop()	12
6.1.2.2 handleEvents()	12
6.1.2.3 parser()	12
6.1.2.4 runArcade()	12
6.1.2.5 setMode()	13
6.2 Arcade::Core::CoreException Class Reference	13
6.2.1 Detailed Description	14
6.3 Arcade::IGame Class Reference	14
6.3.1 Detailed Description	14
6.4 Arcade::IRenderer Class Reference	14
6.4.1 Detailed Description	15
6.4.2 Member Function Documentation	15
6.4.2.1 createSprite()	15
6.4.2.2 createText()	16
6.4.2.3 displaySprite()	16
6.4.2.4 displayText()	17
6.4.2.5 getEvent()	17
6.4.2.6 isASCII()	17
6.4.2.7 loadFont()	17
6.4.2.8 loadTexture()	18
6.4.2.9 openWindow()	18
6.4.2.10 setTitle()	19

6.5 Arcade::ISprite Class Reference	19
6.5.1 Detailed Description	19
6.5.2 Member Function Documentation	19
6.5.2.1 createSprite()	19
6.5.2.2 setPosition()	20
6.6 Arcade::Ncurses Class Reference	20
6.6.1 Detailed Description	21
6.6.2 Member Function Documentation	21
6.6.2.1 createSprite()	21
6.6.2.2 createText()	22
6.6.2.3 displaySprite()	22
6.6.2.4 displayText()	22
6.6.2.5 getEvent()	23
6.6.2.6 isASCII()	23
6.6.2.7 loadFont()	23
6.6.2.8 loadTexture()	24
6.6.2.9 openWindow()	24
6.6.2.10 setTitle()	24
6.7 Arcade::RuntimeException Class Reference	26
6.7.1 Detailed Description	26
6.8 Arcade::Sdl2 Class Reference	26
6.8.1 Detailed Description	27
6.8.2 Member Function Documentation	27
6.8.2.1 createSprite()	27
6.8.2.2 createText()	28
6.8.2.3 displaySprite()	28
6.8.2.4 displayText()	29
6.8.2.5 getEvent()	29
6.8.2.6 isASCII()	29
6.8.2.7 loadFont()	29
6.8.2.8 loadTexture()	30
6.8.2.9 openWindow()	30
6.8.2.10 setTitle()	31
6.9 Arcade::Sfml Class Reference	31
6.9.1 Detailed Description	32
6.9.2 Member Function Documentation	32
6.9.2.1 createSprite()	32
6.9.2.2 createText()	33
6.9.2.3 displaySprite()	33
6.9.2.4 displayText()	33
6.9.2.5 getEvent()	34
6.9.2.6 isASCII()	34

6.9.2.7 loadFont()	34
6.9.2.8 loadTexture()	35
6.9.2.9 openWindow()	35
6.9.2.10 setTitle()	35
<b>7 File Documentation</b>	<b>37</b>
7.1 Core/include/Arcade/abstractions/IGame.hpp File Reference	37
7.1.1 Detailed Description	37
7.2 Core/include/Arcade/abstractions/IRenderer.hpp File Reference	37
7.2.1 Detailed Description	38
7.3 Core/include/Arcade/abstractions/ISprite.hpp File Reference	38
7.3.1 Detailed Description	38
7.4 Core/include/Arcade/Constants.hpp File Reference	38
7.4.1 Detailed Description	39
7.5 Core/include/Arcade/Core.hpp File Reference	39
7.5.1 Detailed Description	39
7.6 Core/include/Arcade/Enum.hpp File Reference	39
7.6.1 Detailed Description	40
7.7 Core/include/Arcade/RuntimeException.hpp File Reference	40
7.7.1 Detailed Description	40
7.8 Graphics/NCurses/include/Arcade/NCurses.hpp File Reference	40
7.8.1 Detailed Description	41
7.9 Graphics/SDL2/include/Arcade/Sdl2.hpp File Reference	41
7.9.1 Detailed Description	41
7.10 Graphics/SFML/include/Arcade/Sfml.hpp File Reference	42
7.10.1 Detailed Description	42
<b>Index</b>	<b>43</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

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

<a href="#">Arcade</a>	Main namespace for the <a href="#">Arcade</a> project . . . . .	9
------------------------	---	---





## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Arcade::Core . . . . .	11
std::exception	
Arcade::Core::CoreException . . . . .	13
Arcade::RuntimeException . . . . .	26
Arcade::IGame . . . . .	14
Arcade::IRenderer . . . . .	14
Arcade::Ncurses . . . . .	20
Arcade::Sdl2 . . . . .	26
Arcade::Sfml . . . . .	31
Arcade::ISprite . . . . .	19



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Arcade::Core</a>	
<a href="#">Core</a> class . . . . .	11
<a href="#">Arcade::Core::CoreException</a>	
<a href="#">CoreException</a> class . . . . .	13
<a href="#">Arcade::IGame</a>	
Game Interface . . . . .	14
<a href="#">Arcade::IRenderer</a>	
Graphic Interface . . . . .	14
<a href="#">Arcade::ISprite</a>	
Sprite Interface . . . . .	19
<a href="#">Arcade::Ncurses</a>	
<a href="#">Ncurses</a> class . . . . .	20
<a href="#">Arcade::RuntimeException</a>	
<a href="#">RuntimeException</a> class . . . . .	26
<a href="#">Arcade::Sdl2</a>	
<a href="#">Sdl2</a> class . . . . .	26
<a href="#">Arcade::Sfml</a>	
<a href="#">Sfml</a> class . . . . .	31



## Chapter 4

# File Index

### 4.1 File List

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

Core/include/Arcade/ <a href="#">Constants.hpp</a>	
File containing constant values used in the <a href="#">Arcade</a> project	38
Core/include/Arcade/ <a href="#">Core.hpp</a>	
File declaring Core	39
Core/include/Arcade/ <a href="#">Enum.hpp</a>	
File declaring Enum	39
Core/include/Arcade/ <a href="#">RuntimeException.hpp</a>	
File declaring RuntimeException	40
Core/include/Arcade/abstractions/ <a href="#">IGame.hpp</a>	
File declaring Game Interface	37
Core/include/Arcade/abstractions/ <a href="#">IRenderer.hpp</a>	
File declaring Graphic Interface	37
Core/include/Arcade/abstractions/ <a href="#">ISprite.hpp</a>	
File declaring Sprite Interface	38
Graphics/NCurses/include/Arcade/ <a href="#">Ncurses.hpp</a>	
File declaring Ncurses	40
Graphics/SDL2/include/Arcade/ <a href="#">Sdl2.hpp</a>	
File declaring Sdl2 class	41
Graphics/SFML/include/Arcade/ <a href="#">Sfml.hpp</a>	
File declaring Sfml class	42



## Chapter 5

# Namespace Documentation

### 5.1 Arcade Namespace Reference

Main namespace for the [Arcade](#) project.

#### Classes

- class [IGame](#)  
*Game Interface.*
- class [IRenderer](#)  
*Graphic Interface.*
- class [ISprite](#)  
*Sprite Interface.*
- class [Core](#)  
*Core class.*
- class [RuntimeException](#)  
*RuntimeException class.*
- class [Sdl2](#)  
*Sdl2 class.*
- class [Ncurses](#)  
*Ncurses class.*
- class [Sfml](#)  
*Sfml class.*

#### Enumerations

- enum class **CoreMode** { **GAME** , **MENU** , **PAUSE** , **QUIT** }
- enum class **GameMode** { **GAME** , **GAME\_OVER** , **MENU** , **PAUSE** , **QUIT** }
- enum class [GameEvent](#) { **UP** , **DOWN** , **LEFT** , **RIGHT** , **SHOOT** , **PAUSE** , **QUIT** , **NONE** }  
*Enum for the GameEvent.*

### 5.1.1 Detailed Description

Main namespace for the [Arcade](#) project.

## 5.1.2 Enumeration Type Documentation

### 5.1.2.1 GameEvent

enum [Arcade::GameEvent](#) [strong]

Enum for the GameEvent.

Set of enum used for the game event



## Chapter 6

# Class Documentation

### 6.1 Arcade::Core Class Reference

[Core](#) class.

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

#### Classes

- class [CoreException](#)  
*CoreException* class.

#### Public Member Functions

- **Core** (const [Core](#) &)=delete
- **Core & operator=** (const [Core](#) &)=delete
- **Core** (const [Core](#) &&)=delete
- **Core & operator=** (const [Core](#) &&)=delete
- void [parser](#) (const std::string &path)  
*parse arguments*
- int [runArcade](#) (const std::string &path)  
*Run Arcade.*
- void [gameLoop](#) ()  
*Game loop.*
- void [handleEvents](#) (const [GameEvent](#) &event)  
*Handle events.*
- void [setMode](#) (const CoreMode &gameMode)  
*Set the game mode.*
- void [closeWindow](#) ()  
*close the window*
- void [loadGraphic](#) ()  
*load graphic*
- void [displayMenu](#) ()  
*display menu*

### 6.1.1 Detailed Description

[Core](#) class.

[Core](#) class for the [Arcade](#) project

### 6.1.2 Member Function Documentation

#### 6.1.2.1 `gameLoop()`

```
void Arcade::Core::gameLoop ( )
```

Game loop.

Main game loop

#### 6.1.2.2 `handleEvents()`

```
void Arcade::Core::handleEvents (
    const GameEvent & event )
```

Handle events.

##### Parameters

<i>event</i>	
--------------	--

#### 6.1.2.3 `parser()`

```
void Arcade::Core::parser (
    const std::string & path )
```

parse arguments

##### Parameters

<i>path</i>	
-------------	--

#### 6.1.2.4 `runArcade()`

```
int Arcade::Core::runArcade (
```

```
const std::string & path )
```

Run [Arcade](#).

#### Parameters

<i>path</i>	
-------------	--

#### Returns

int

#### 6.1.2.5 setMode()

```
void Arcade::Core::setMode (
    const CoreMode & gameMode ) [inline]
```

Set the game mode.

#### Parameters

<i>gameMode</i>	
-----------------	--

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

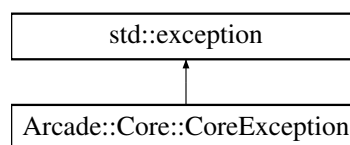
- Core/include/Arcade/[Core.hpp](#)

## 6.2 Arcade::Core::CoreException Class Reference

[CoreException](#) class.

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

Inheritance diagram for Arcade::Core::CoreException:



### Public Member Functions

- **CoreException** (std::string msg)
- **CoreException** (const [CoreException](#) &)=delete
- [CoreException](#) & **operator=** (const [CoreException](#) &)=delete
- **CoreException** (const [CoreException](#) &&)=delete
- [CoreException](#) & **operator=** (const [CoreException](#) &&)=delete
- const char \* **what** () const noexcept override

### 6.2.1 Detailed Description

[CoreException](#) class.

Exception class for the [Core](#) class

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

- [Core/include/Arcade/Core.hpp](#)

## 6.3 Arcade::IGame Class Reference

Game Interface.

```
#include <Arcade/abstractions/IGame.hpp>
```

### Public Member Functions

- **IGame** (const [IGame](#) &)=delete
- **IGame** & **operator=** (const [IGame](#) &)=delete
- virtual void **start** ()=0  
*start the game*
- virtual void **stop** ()=0  
*stop the game*

### 6.3.1 Detailed Description

Game Interface.

Set of method to implement for the business logic of the [Arcade](#) project

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

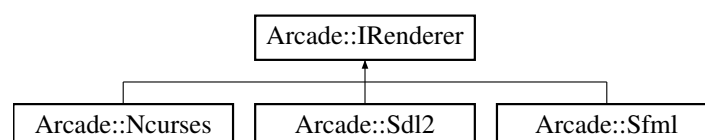
- [Core/include/Arcade/abstractions/IGame.hpp](#)

## 6.4 Arcade::IRenderer Class Reference

Graphic Interface.

```
#include <Arcade/abstractions/IRenderer.hpp>
```

Inheritance diagram for `Arcade::IRenderer`:



## Public Member Functions

- virtual void [openWindow](#) (unsigned int width, unsigned int height)=0  
*Create a window with the specified dimensions.*
- virtual void [displayWindow](#) ()=0  
*Display the current window.*
- virtual void [clearWindow](#) ()=0  
*Clear the contents of the current window.*
- virtual void [closeWindow](#) ()=0  
*Close the current window.*
- virtual void [setTitle](#) (const std::string &title)=0  
*Set the title of the window.*
- virtual [GameEvent](#) [getEvent](#) ()=0  
*Retrieve the current event from the window.*
- virtual bool [isASCII](#) ()=0  
*Check if the renderer supports only ASCII characters.*
- virtual bool [loadTexture](#) (const std::string &filepath, const std::string &name)=0  
*Load a texture from a file.*
- virtual bool [loadFont](#) (const std::string &filepath, const std::string &name)=0  
*Load a font from a file.*
- virtual void [createSprite](#) (const std::string &textureName, float pos\_x, float pos\_y, float scale\_x, float scale\_y)=0  
*Create a sprite with the specified parameters.*
- virtual void [displaySprite](#) (const std::string &spriteName)=0  
*Display a sprite with the specified name.*
- virtual void [createText](#) (const std::string &fontName, const std::string &text, int size, float pos\_x, float pos\_y)=0  
*Create a text object with the specified parameters.*
- virtual void [displayText](#) (const std::string &textName)=0  
*Display a text object with the specified name.*

### 6.4.1 Detailed Description

Graphic Interface.

Set of method for the [Arcade](#) project

### 6.4.2 Member Function Documentation

#### 6.4.2.1 createSprite()

```
virtual void Arcade::IRenderer::createSprite (
    const std::string & textureName,
    float pos_x,
    float pos_y,
    float scale_x,
    float scale_y ) [pure virtual]
```

Create a sprite with the specified parameters.

## Parameters

<i>textureName</i>	The name of the texture to use for the sprite
<i>pos_x</i>	The x-coordinate of the sprite's position
<i>pos_y</i>	The y-coordinate of the sprite's position
<i>scale_x</i>	The scale factor for the sprite's width
<i>scale_y</i>	The scale factor for the sprite's height

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

### 6.4.2.2 createText()

```
virtual void Arcade::IRenderer::createText (
    const std::string & fontName,
    const std::string & text,
    int size,
    float pos_x,
    float pos_y ) [pure virtual]
```

Create a text object with the specified parameters.

## Parameters

<i>fontName</i>	The name of the font to use for the text
<i>text</i>	The text to display
<i>size</i>	The size of the text
<i>pos_x</i>	The x-coordinate of the text's position
<i>pos_y</i>	The y-coordinate of the text's position

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

### 6.4.2.3 displaySprite()

```
virtual void Arcade::IRenderer::displaySprite (
    const std::string & spriteName ) [pure virtual]
```

Display a sprite with the specified name.

## Parameters

<i>spriteName</i>	The name of the sprite to display
-------------------	-----------------------------------

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

#### 6.4.2.4 displayText()

```
virtual void Arcade::IRenderer::displayText (
    const std::string & textName ) [pure virtual]
```

Display a text object with the specified name.

##### Parameters

<i>textName</i>	The name of the text to display
-----------------	---------------------------------

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

#### 6.4.2.5 getEvent()

```
virtual GameEvent Arcade::IRenderer::getEvent ( ) [pure virtual]
```

Retrieve the current event from the window.

##### Returns

The current event

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

#### 6.4.2.6 isASCII()

```
virtual bool Arcade::IRenderer::isASCII ( ) [pure virtual]
```

Check if the renderer supports only ASCII characters.

##### Returns

True if the renderer supports ASCII, false otherwise

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

#### 6.4.2.7 loadFont()

```
virtual bool Arcade::IRenderer::loadFont (
    const std::string & filepath,
    const std::string & name ) [pure virtual]
```

Load a font from a file.

**Parameters**

<i>filepath</i>	The path to the font file
<i>name</i>	The name to associate with the font

**Returns**

True if the font was loaded successfully, false otherwise

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

**6.4.2.8 loadTexture()**

```
virtual bool Arcade::IRenderer::loadTexture (
    const std::string & filepath,
    const std::string & name ) [pure virtual]
```

Load a texture from a file.

**Parameters**

<i>filepath</i>	The path to the texture file
<i>name</i>	The name to associate with the texture

**Returns**

True if the texture was loaded successfully, false otherwise

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

**6.4.2.9 openWindow()**

```
virtual void Arcade::IRenderer::openWindow (
    unsigned int width,
    unsigned int height ) [pure virtual]
```

Create a window with the specified dimensions.

**Parameters**

<i>width</i>	The width of the window
<i>height</i>	The height of the window

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



### 6.4.2.10 setTitle()

```
virtual void Arcade::IRenderer::setTitle (
    const std::string & title ) [pure virtual]
```

Set the title of the window.

#### Parameters

<i>title</i>	The new title of the window
--------------	-----------------------------

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

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

- Core/include/Arcade/abstractions/[IRenderer.hpp](#)

## 6.5 Arcade::ISprite Class Reference

Sprite Interface.

```
#include <Arcade/abstractions/ISprite.hpp>
```

### Public Member Functions

- [ISprite](#) (const [ISprite](#) &)=delete
- [ISprite](#) & **operator=** (const [ISprite](#) &)=delete
- virtual void [createSprite](#) (const std::vector< std::string > &sprite)=0  
*create the sprite*
- virtual void [draw](#) ()=0  
*draw the sprite*
- virtual void [setPosition](#) (int x, int y)=0  
*set the position of the sprite*

### 6.5.1 Detailed Description

Sprite Interface.

Set of method to implement for the business logic of the [Arcade](#) project

### 6.5.2 Member Function Documentation

#### 6.5.2.1 createSprite()

```
virtual void Arcade::ISprite::createSprite (
    const std::vector< std::string > & sprite ) [pure virtual]
```

create the sprite

## Parameters

<i>sprite</i>	
---------------	--

create the sprite with the given sprite, can be a filePath, or a ASCII art

**6.5.2.2 setPosition()**

```
virtual void Arcade::ISprite::setPosition (
    int x,
    int y ) [pure virtual]
```

set the position of the sprite

## Parameters

<i>x</i>	
<i>y</i>	

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

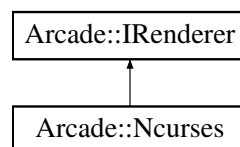
- Core/include/Arcade/abstractions/[ISprite.hpp](#)

**6.6 Arcade::Ncurses Class Reference**

[Ncurses](#) class.

```
#include <Arcade/Ncurses.hpp>
```

Inheritance diagram for Arcade::Ncurses:

**Public Member Functions**

- [GameEvent](#) [getEvent](#) () override  
*Retrieve the current event from the window.*
- void [openWindow](#) (unsigned int width, unsigned int height) override  
*Create a window with the specified dimensions.*
- void [displayWindow](#) () override  
*Display the current window.*
- void [clearWindow](#) () override  
*Clear the contents of the current window.*

- void `closeWindow` () override  
*Close the current window.*
- void `setTitle` (const std::string &title) override  
*Set the title of the window.*
- bool `loadTexture` (const std::string &texture, const std::string &name) override  
*Load a texture from a file.*
- bool `loadFont` (const std::string &filepath, const std::string &name) override  
*Load a font from a file.*
- void `createSprite` (const std::string &textureName, float pos\_x, float pos\_y, float scale\_x, float scale\_y) override  
*Create a sprite with the specified parameters.*
- void `createText` (const std::string &fontName, const std::string &text, int size, float pos\_x, float pos\_y) override  
*Create a text object with the specified parameters.*
- void `displaySprite` (const std::string &spriteName) override  
*Display a sprite with the specified name.*
- void `displayText` (const std::string &textName) override  
*Display a text object with the specified name.*
- bool `isASCII` () override  
*Check if the renderer supports only ASCII characters.*

### 6.6.1 Detailed Description

`Ncurses` class.

`Ncurses` class for the `Arcade` project

### 6.6.2 Member Function Documentation

#### 6.6.2.1 createSprite()

```
void Arcade::Ncurses::createSprite (
    const std::string & textureName,
    float pos_x,
    float pos_y,
    float scale_x,
    float scale_y ) [inline], [override], [virtual]
```

Create a sprite with the specified parameters.

##### Parameters

<i>textureName</i>	The name of the texture to use for the sprite
<i>pos_x</i>	The x-coordinate of the sprite's position
<i>pos_y</i>	The y-coordinate of the sprite's position
<i>scale_x</i>	The scale factor for the sprite's width
<i>scale_y</i>	The scale factor for the sprite's height

Implements [Arcade::IRenderer](#).

### 6.6.2.2 createText()

```
void Arcade::Ncurses::createText (
    const std::string & fontName,
    const std::string & text,
    int size,
    float pos_x,
    float pos_y ) [inline], [override], [virtual]
```

Create a text object with the specified parameters.

#### Parameters

<i>fontName</i>	The name of the font to use for the text
<i>text</i>	The text to display
<i>size</i>	The size of the text
<i>pos_x</i>	The x-coordinate of the text's position
<i>pos_y</i>	The y-coordinate of the text's position

Implements [Arcade::IRenderer](#).

### 6.6.2.3 displaySprite()

```
void Arcade::Ncurses::displaySprite (
    const std::string & spriteName ) [inline], [override], [virtual]
```

Display a sprite with the specified name.

#### Parameters

<i>spriteName</i>	The name of the sprite to display
-------------------	-----------------------------------

Implements [Arcade::IRenderer](#).

### 6.6.2.4 displayText()

```
void Arcade::Ncurses::displayText (
    const std::string & textName ) [inline], [override], [virtual]
```

Display a text object with the specified name.

## Parameters

<i>textName</i>	The name of the text to display
-----------------	---------------------------------

Implements [Arcade::IRenderer](#).

### 6.6.2.5 `getEvent()`

```
GameEvent Arcade::Ncurses::getEvent ( ) [override], [virtual]
```

Retrieve the current event from the window.

## Returns

The current event

Implements [Arcade::IRenderer](#).

### 6.6.2.6 `isASCII()`

```
bool Arcade::Ncurses::isASCII ( ) [inline], [override], [virtual]
```

Check if the renderer supports only ASCII characters.

## Returns

True if the renderer supports ASCII, false otherwise

Implements [Arcade::IRenderer](#).

### 6.6.2.7 `loadFont()`

```
bool Arcade::Ncurses::loadFont (
    const std::string & filepath,
    const std::string & name ) [inline], [override], [virtual]
```

Load a font from a file.

## Parameters

<i>filepath</i>	The path to the font file
<i>name</i>	The name to associate with the font

**Returns**

True if the font was loaded successfully, false otherwise

Implements [Arcade::IRenderer](#).

**6.6.2.8 loadTexture()**

```
bool Arcade::Ncurses::loadTexture (
    const std::string & filepath,
    const std::string & name ) [inline], [override], [virtual]
```

Load a texture from a file.

**Parameters**

<i>filepath</i>	The path to the texture file
<i>name</i>	The name to associate with the texture

**Returns**

True if the texture was loaded successfully, false otherwise

Implements [Arcade::IRenderer](#).

**6.6.2.9 openWindow()**

```
void Arcade::Ncurses::openWindow (
    unsigned int width,
    unsigned int height ) [override], [virtual]
```

Create a window with the specified dimensions.

**Parameters**

<i>width</i>	The width of the window
<i>height</i>	The height of the window

Implements [Arcade::IRenderer](#).

**6.6.2.10 setTitle()**

```
void Arcade::Ncurses::setTitle (
    const std::string & title ) [inline], [override], [virtual]
```

Set the title of the window.

## Parameters

<i>title</i>	The new title of the window
--------------	-----------------------------

Implements [Arcade::IRenderer](#).

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

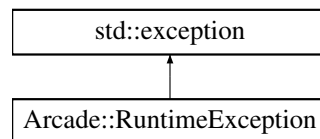
- Graphics/NCurses/include/Arcade/[Ncurses.hpp](#)

## 6.7 Arcade::RuntimeException Class Reference

[RuntimeException](#) class.

```
#include <Arcade/RuntimeException.hpp>
```

Inheritance diagram for `Arcade::RuntimeException`:



### Public Member Functions

- **RuntimeException** (std::string msg)
- **RuntimeException** ([RuntimeException](#) const &)=delete
- const char \* **what** () const noexcept override

### 6.7.1 Detailed Description

[RuntimeException](#) class.

[RuntimeException](#) class for the [Arcade](#) project

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

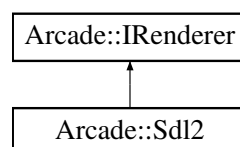
- Core/include/Arcade/[RuntimeException.hpp](#)

## 6.8 Arcade::Sdl2 Class Reference

[Sdl2](#) class.

```
#include <Arcade/Sdl2.hpp>
```

Inheritance diagram for `Arcade::Sdl2`:





## Public Member Functions

- [GameEvent](#) [getEvent](#) () override  
*Retrieve the current event from the window.*
- void [openWindow](#) (unsigned int width, unsigned int height) override  
*Create a window with the specified dimensions.*
- void [displayWindow](#) () override  
*Display the current window.*
- void [clearWindow](#) () override  
*Clear the contents of the current window.*
- void [closeWindow](#) () override  
*Close the current window.*
- void [setTitle](#) (const std::string &title) override  
*Set the title of the window.*
- bool [loadTexture](#) (const std::string &filePath, const std::string &name) override  
*Load a texture from a file.*
- bool [loadFont](#) (const std::string &filePath, const std::string &name) override  
*Load a font from a file.*
- void [createSprite](#) (const std::string &textureName, float pos\_x, float pos\_y, float scale\_x, float scale\_y) override  
*Create a sprite with the specified parameters.*
- void [createText](#) (const std::string &fontName, const std::string &text, int size, float pos\_x, float pos\_y) override  
*Create a text object with the specified parameters.*
- void [displaySprite](#) (const std::string &spriteName) override  
*Display a sprite with the specified name.*
- void [displayText](#) (const std::string &textName) override  
*Display a text object with the specified name.*
- bool [isASCII](#) () override  
*Check if the renderer supports only ASCII characters.*

### 6.8.1 Detailed Description

[Sdl2](#) class.

[Sdl2](#) class for the [Arcade](#) project

### 6.8.2 Member Function Documentation

#### 6.8.2.1 createSprite()

```
void Arcade::Sdl2::createSprite (
    const std::string & textureName,
    float pos_x,
    float pos_y,
    float scale_x,
    float scale_y ) [override], [virtual]
```

Create a sprite with the specified parameters.

## Parameters

<i>textureName</i>	The name of the texture to use for the sprite
<i>pos_x</i>	The x-coordinate of the sprite's position
<i>pos_y</i>	The y-coordinate of the sprite's position
<i>scale_x</i>	The scale factor for the sprite's width
<i>scale_y</i>	The scale factor for the sprite's height

Implements [Arcade::IRenderer](#).

### 6.8.2.2 createText()

```
void Arcade::Sdl2::createText (
    const std::string & fontName,
    const std::string & text,
    int size,
    float pos_x,
    float pos_y ) [override], [virtual]
```

Create a text object with the specified parameters.

## Parameters

<i>fontName</i>	The name of the font to use for the text
<i>text</i>	The text to display
<i>size</i>	The size of the text
<i>pos_x</i>	The x-coordinate of the text's position
<i>pos_y</i>	The y-coordinate of the text's position

Implements [Arcade::IRenderer](#).

### 6.8.2.3 displaySprite()

```
void Arcade::Sdl2::displaySprite (
    const std::string & spriteName ) [override], [virtual]
```

Display a sprite with the specified name.

## Parameters

<i>spriteName</i>	The name of the sprite to display
-------------------	-----------------------------------

Implements [Arcade::IRenderer](#).

#### 6.8.2.4 displayText()

```
void Arcade::Sdl2::displayText (
    const std::string & textName ) [override], [virtual]
```

Display a text object with the specified name.

##### Parameters

<i>textName</i>	The name of the text to display
-----------------	---------------------------------

Implements [Arcade::IRenderer](#).

#### 6.8.2.5 getEvent()

```
GameEvent Arcade::Sdl2::getEvent ( ) [override], [virtual]
```

Retrieve the current event from the window.

##### Returns

The current event

Implements [Arcade::IRenderer](#).

#### 6.8.2.6 isASCII()

```
bool Arcade::Sdl2::isASCII ( ) [inline], [override], [virtual]
```

Check if the renderer supports only ASCII characters.

##### Returns

True if the renderer supports ASCII, false otherwise

Implements [Arcade::IRenderer](#).

#### 6.8.2.7 loadFont()

```
bool Arcade::Sdl2::loadFont (
    const std::string & filepath,
    const std::string & name ) [override], [virtual]
```

Load a font from a file.

**Parameters**

<i>filepath</i>	The path to the font file
<i>name</i>	The name to associate with the font

**Returns**

True if the font was loaded successfully, false otherwise

Implements [Arcade::IRenderer](#).

**6.8.2.8 loadTexture()**

```
bool Arcade::Sdl2::loadTexture (
    const std::string & filepath,
    const std::string & name ) [override], [virtual]
```

Load a texture from a file.

**Parameters**

<i>filepath</i>	The path to the texture file
<i>name</i>	The name to associate with the texture

**Returns**

True if the texture was loaded successfully, false otherwise

Implements [Arcade::IRenderer](#).

**6.8.2.9 openWindow()**

```
void Arcade::Sdl2::openWindow (
    unsigned int width,
    unsigned int height ) [override], [virtual]
```

Create a window with the specified dimensions.

**Parameters**

<i>width</i>	The width of the window
<i>height</i>	The height of the window

Implements [Arcade::IRenderer](#).

### 6.8.2.10 setTitle()

```
void Arcade::Sdl2::setTitle (
    const std::string & title ) [inline], [override], [virtual]
```

Set the title of the window.

#### Parameters

<i>title</i>	The new title of the window
--------------	-----------------------------

Implements [Arcade::IRenderer](#).

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

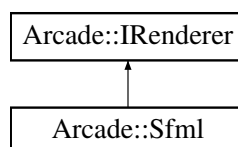
- Graphics/SDL2/include/Arcade/[Sdl2.hpp](#)

## 6.9 Arcade::Sfml Class Reference

[Sfml](#) class.

```
#include <Arcade/Sfml.hpp>
```

Inheritance diagram for Arcade::Sfml:



### Public Member Functions

- [GameEvent](#) [getEvent](#) () override  
*Retrieve the current event from the window.*
- void [openWindow](#) (unsigned int width, unsigned int height) override  
*Create a window with the specified dimensions.*
- void [displayWindow](#) () override  
*Display the current window.*
- void [clearWindow](#) () override  
*Clear the contents of the current window.*
- void [closeWindow](#) () override  
*Close the current window.*
- void [setTitle](#) (const std::string &title) override  
*Set the title of the window.*
- bool [loadTexture](#) (const std::string &texture, const std::string &name) override

*Load a texture from a file.*

- bool [loadFont](#) (const std::string &filepath, const std::string &name) override

*Load a font from a file.*

- void [createSprite](#) (const std::string &textureName, float pos\_x, float pos\_y, float scale\_x, float scale\_y) override

*Create a sprite with the specified parameters.*

- void [createText](#) (const std::string &fontName, const std::string &text, int size, float pos\_x, float pos\_y) override

*Create a text object with the specified parameters.*

- void [displaySprite](#) (const std::string &spriteName) override

*Display a sprite with the specified name.*

- void [displayText](#) (const std::string &textName) override

*Display a text object with the specified name.*

- bool [isASCII](#) () override

*Check if the renderer supports only ASCII characters.*

## 6.9.1 Detailed Description

[Sfml](#) class.

[Sfml](#) class for the [Arcade](#) project

## 6.9.2 Member Function Documentation

### 6.9.2.1 createSprite()

```
void Arcade::Sfml::createSprite (
    const std::string & textureName,
    float pos_x,
    float pos_y,
    float scale_x,
    float scale_y ) [override], [virtual]
```

Create a sprite with the specified parameters.

#### Parameters

<i>textureName</i>	The name of the texture to use for the sprite
<i>pos_x</i>	The x-coordinate of the sprite's position
<i>pos_y</i>	The y-coordinate of the sprite's position
<i>scale_x</i>	The scale factor for the sprite's width
<i>scale_y</i>	The scale factor for the sprite's height

Implements [Arcade::IRenderer](#).

### 6.9.2.2 createText()

```
void Arcade::Sfml::createText (
    const std::string & fontName,
    const std::string & text,
    int size,
    float pos_x,
    float pos_y ) [override], [virtual]
```

Create a text object with the specified parameters.

#### Parameters

<i>fontName</i>	The name of the font to use for the text
<i>text</i>	The text to display
<i>size</i>	The size of the text
<i>pos_x</i>	The x-coordinate of the text's position
<i>pos_y</i>	The y-coordinate of the text's position

Implements [Arcade::IRenderer](#).

### 6.9.2.3 displaySprite()

```
void Arcade::Sfml::displaySprite (
    const std::string & spriteName ) [override], [virtual]
```

Display a sprite with the specified name.

#### Parameters

<i>spriteName</i>	The name of the sprite to display
-------------------	-----------------------------------

Implements [Arcade::IRenderer](#).

### 6.9.2.4 displayText()

```
void Arcade::Sfml::displayText (
    const std::string & textName ) [override], [virtual]
```

Display a text object with the specified name.

#### Parameters

<i>textName</i>	The name of the text to display
-----------------	---------------------------------

Implements [Arcade::IRenderer](#).

#### 6.9.2.5 `getEvent()`

```
GameEvent Arcade::Sfml::getEvent ( ) [override], [virtual]
```

Retrieve the current event from the window.

##### Returns

The current event

Implements [Arcade::IRenderer](#).

#### 6.9.2.6 `isASCII()`

```
bool Arcade::Sfml::isASCII ( ) [inline], [override], [virtual]
```

Check if the renderer supports only ASCII characters.

##### Returns

True if the renderer supports ASCII, false otherwise

Implements [Arcade::IRenderer](#).

#### 6.9.2.7 `loadFont()`

```
bool Arcade::Sfml::loadFont (
    const std::string & filepath,
    const std::string & name ) [override], [virtual]
```

Load a font from a file.

##### Parameters

<i>filepath</i>	The path to the font file
<i>name</i>	The name to associate with the font

##### Returns

True if the font was loaded successfully, false otherwise



Implements [Arcade::IRenderer](#).

### 6.9.2.8 loadTexture()

```
bool Arcade::Sfml::loadTexture (
    const std::string & filepath,
    const std::string & name ) [override], [virtual]
```

Load a texture from a file.

#### Parameters

<i>filepath</i>	The path to the texture file
<i>name</i>	The name to associate with the texture

#### Returns

True if the texture was loaded successfully, false otherwise

Implements [Arcade::IRenderer](#).

### 6.9.2.9 openWindow()

```
void Arcade::Sfml::openWindow (
    unsigned int width,
    unsigned int height ) [override], [virtual]
```

Create a window with the specified dimensions.

#### Parameters

<i>width</i>	The width of the window
<i>height</i>	The height of the window

Implements [Arcade::IRenderer](#).

### 6.9.2.10 setTitle()

```
void Arcade::Sfml::setTitle (
    const std::string & title ) [inline], [override], [virtual]
```

Set the title of the window.

**Parameters**

<i>title</i>	The new title of the window
--------------	-----------------------------

Implements [Arcade::IRenderer](#).

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

- Graphics/SFML/include/Arcade/[Sfml.hpp](#)

## Chapter 7

# File Documentation

### 7.1 Core/include/Arcade/abstractions/IGame.hpp File Reference

File declaring Game Interface.

#### Classes

- class [Arcade::IGame](#)  
*Game Interface.*

#### Namespaces

- [Arcade](#)  
*Main namespace for the [Arcade](#) project.*

#### 7.1.1 Detailed Description

File declaring Game Interface.

### 7.2 Core/include/Arcade/abstractions/IRenderer.hpp File Reference

File declaring Graphic Interface.

```
#include <string>
#include "Arcade/Enum.hpp"
```

#### Classes

- class [Arcade::IRenderer](#)  
*Graphic Interface.*

## Namespaces

- [Arcade](#)

*Main namespace for the [Arcade](#) project.*

### 7.2.1 Detailed Description

File declaring Graphic Interface.

## 7.3 Core/include/Arcade/abstractions/ISprite.hpp File Reference

File declaring Sprite Interface.

```
#include <string>
#include <vector>
```

## Classes

- class [Arcade::ISprite](#)

*Sprite Interface.*

## Namespaces

- [Arcade](#)

*Main namespace for the [Arcade](#) project.*

### 7.3.1 Detailed Description

File declaring Sprite Interface.

## 7.4 Core/include/Arcade/Constants.hpp File Reference

File containing constant values used in the [Arcade](#) project.

## Variables

- constexpr int [EPITECH\\_ERROR](#) = 84  
*Exit code for Epitech errors.*
- constexpr int [SUCCESS](#) = 0  
*Exit code indicating that the program has executed successfully.*
- constexpr int [ERROR](#) = -1  
*Exit code indicating that an error has occurred.*

### 7.4.1 Detailed Description

File containing constant values used in the [Arcade](#) project.

This file provides a set of constant values that are used throughout the [Arcade](#) project.

## 7.5 Core/include/Arcade/Core.hpp File Reference

File declaring Core.

```
#include <memory>
#include "Arcade/Enum.hpp"
#include "Arcade/abstractions/IRenderer.hpp"
```

### Classes

- class [Arcade::Core](#)  
*Core class.*
- class [Arcade::Core::CoreException](#)  
*CoreException class.*

### Namespaces

- [Arcade](#)  
*Main namespace for the [Arcade](#) project.*

### 7.5.1 Detailed Description

File declaring Core.

## 7.6 Core/include/Arcade/Enum.hpp File Reference

File declaring Enum.

### Namespaces

- [Arcade](#)  
*Main namespace for the [Arcade](#) project.*

## Enumerations

- enum class **CoreMode** { **GAME** , **MENU** , **PAUSE** , **QUIT** }
- enum class **GameMode** { **GAME** , **GAME\_OVER** , **MENU** , **PAUSE** , **QUIT** }
- enum class [Arcade::GameEvent](#) { **UP** , **DOWN** , **LEFT** , **RIGHT** , **SHOOT** , **PAUSE** , **QUIT** , **NONE** }

*Enum for the GameEvent.*

### 7.6.1 Detailed Description

File declaring Enum.

Set of enum used in the [Arcade](#) project

## 7.7 Core/include/Arcade/RuntimeException.hpp File Reference

File declaring RuntimeException.

```
#include <iostream>
#include <exception>
```

## Classes

- class [Arcade::RuntimeException](#)  
*RuntimeException class.*

## Namespaces

- [Arcade](#)  
*Main namespace for the [Arcade](#) project.*

### 7.7.1 Detailed Description

File declaring RuntimeException.

## 7.8 Graphics/NCurses/include/Arcade/Ncurses.hpp File Reference

File declaring Ncurses.

```
#include <ncurses.h>
#include "Arcade/abstractions/IRenderer.hpp"
```

## Classes

- class [Arcade::Ncurses](#)  
*Ncurses class.*

## Namespaces

- [Arcade](#)  
*Main namespace for the [Arcade](#) project.*

### 7.8.1 Detailed Description

File declaring Ncurses.

## 7.9 Graphics/SDL2/include/Arcade/Sdl2.hpp File Reference

File declaring Sdl2 class.

```
#include <iostream>
#include <map>
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include "Arcade/abstractions/IRenderer.hpp"
```

## Classes

- class [Arcade::Sdl2](#)  
*Sdl2 class.*

## Namespaces

- [Arcade](#)  
*Main namespace for the [Arcade](#) project.*

## Variables

- constexpr std::string\_view **WINDOW\_TITLE** = "Arcade - SDL2"

### 7.9.1 Detailed Description

File declaring Sdl2 class.

## 7.10 Graphics/SFML/include/Arcade/Sfml.hpp File Reference

File declaring Sfml class.

```
#include <SFML/Graphics.hpp>
#include "Arcade/abstractions/IRenderer.hpp"
```

### Classes

- class [Arcade::Sfml](#)  
*[Sfml](#) class.*

### Namespaces

- [Arcade](#)  
*Main namespace for the [Arcade](#) project.*

#### 7.10.1 Detailed Description

File declaring Sfml class.



# Index

- Arcade, [9](#)
  - GameEvent, [10](#)
- Arcade::Core, [11](#)
  - gameLoop, [12](#)
  - handleEvents, [12](#)
  - parser, [12](#)
  - runArcade, [12](#)
  - setMode, [13](#)
- Arcade::Core::CoreException, [13](#)
- Arcade::IGame, [14](#)
- Arcade::IRenderer, [14](#)
  - createSprite, [15](#)
  - createText, [16](#)
  - displaySprite, [16](#)
  - displayText, [16](#)
  - getEvent, [17](#)
  - isASCII, [17](#)
  - loadFont, [17](#)
  - loadTexture, [18](#)
  - openWindow, [18](#)
  - setTitle, [19](#)
- Arcade::ISprite, [19](#)
  - createSprite, [19](#)
  - setPosition, [20](#)
- Arcade::Ncurses, [20](#)
  - createSprite, [21](#)
  - createText, [22](#)
  - displaySprite, [22](#)
  - displayText, [22](#)
  - getEvent, [23](#)
  - isASCII, [23](#)
  - loadFont, [23](#)
  - loadTexture, [24](#)
  - openWindow, [24](#)
  - setTitle, [24](#)
- Arcade::RuntimeException, [26](#)
- Arcade::Sdl2, [26](#)
  - createSprite, [27](#)
  - createText, [28](#)
  - displaySprite, [28](#)
  - displayText, [28](#)
  - getEvent, [29](#)
  - isASCII, [29](#)
  - loadFont, [29](#)
  - loadTexture, [30](#)
  - openWindow, [30](#)
  - setTitle, [31](#)
- Arcade::Sfml, [31](#)
  - createSprite, [32](#)
  - createText, [32](#)
  - displaySprite, [33](#)
  - displayText, [33](#)
  - getEvent, [34](#)
  - isASCII, [34](#)
  - loadFont, [34](#)
  - loadTexture, [35](#)
  - openWindow, [35](#)
  - setTitle, [35](#)
- Core/include/Arcade/abstractions/IGame.hpp, [37](#)
- Core/include/Arcade/abstractions/IRenderer.hpp, [37](#)
- Core/include/Arcade/abstractions/ISprite.hpp, [38](#)
- Core/include/Arcade/Constants.hpp, [38](#)
- Core/include/Arcade/Core.hpp, [39](#)
- Core/include/Arcade/Enum.hpp, [39](#)
- Core/include/Arcade/RuntimeException.hpp, [40](#)
- createSprite
  - Arcade::IRenderer, [15](#)
  - Arcade::ISprite, [19](#)
  - Arcade::Ncurses, [21](#)
  - Arcade::Sdl2, [27](#)
  - Arcade::Sfml, [32](#)
- createText
  - Arcade::IRenderer, [16](#)
  - Arcade::Ncurses, [22](#)
  - Arcade::Sdl2, [28](#)
  - Arcade::Sfml, [32](#)
- displaySprite
  - Arcade::IRenderer, [16](#)
  - Arcade::Ncurses, [22](#)
  - Arcade::Sdl2, [28](#)
  - Arcade::Sfml, [33](#)
- displayText
  - Arcade::IRenderer, [16](#)
  - Arcade::Ncurses, [22](#)
  - Arcade::Sdl2, [28](#)
  - Arcade::Sfml, [33](#)
- GameEvent
  - Arcade, [10](#)
- gameLoop
  - Arcade::Core, [12](#)
- getEvent
  - Arcade::IRenderer, [17](#)
  - Arcade::Ncurses, [23](#)
  - Arcade::Sdl2, [29](#)
  - Arcade::Sfml, [34](#)
- Graphics/Ncurses/include/Arcade/Ncurses.hpp, [40](#)

Graphics/SDL2/include/Arcade/Sdl2.hpp, [41](#)  
Graphics/SFML/include/Arcade/Sfml.hpp, [42](#)

handleEvents  
    Arcade::Core, [12](#)

isASCII  
    Arcade::IRenderer, [17](#)  
    Arcade::Ncurses, [23](#)  
    Arcade::Sdl2, [29](#)  
    Arcade::Sfml, [34](#)

loadFont  
    Arcade::IRenderer, [17](#)  
    Arcade::Ncurses, [23](#)  
    Arcade::Sdl2, [29](#)  
    Arcade::Sfml, [34](#)

loadTexture  
    Arcade::IRenderer, [18](#)  
    Arcade::Ncurses, [24](#)  
    Arcade::Sdl2, [30](#)  
    Arcade::Sfml, [35](#)

openWindow  
    Arcade::IRenderer, [18](#)  
    Arcade::Ncurses, [24](#)  
    Arcade::Sdl2, [30](#)  
    Arcade::Sfml, [35](#)

parser  
    Arcade::Core, [12](#)

runArcade  
    Arcade::Core, [12](#)

setMode  
    Arcade::Core, [13](#)

setPosition  
    Arcade::ISprite, [20](#)

setTitle  
    Arcade::IRenderer, [19](#)  
    Arcade::Ncurses, [24](#)  
    Arcade::Sdl2, [31](#)  
    Arcade::Sfml, [35](#)