

arcade

0.1.0

Generated by Doxygen 1.9.1

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 Arcade Namespace Reference	9
5.1.1 Detailed Description	10
5.1.2 Enumeration Type Documentation	10
5.1.2.1 KeyboardEvents	10
6 Class Documentation	11
6.1 Arcade::AGame Class Reference	11
6.1.1 Detailed Description	12
6.1.2 Member Function Documentation	12
6.1.2.1 getDirection()	12
6.1.2.2 getGameMode()	12
6.1.2.3 getScore()	13
6.1.2.4 getUsername()	13
6.1.2.5 randomize()	13
6.1.2.6 setDirection()	14
6.2 Arcade::Clock Class Reference	14
6.2.1 Member Function Documentation	14
6.2.1.1 getElapsedTime()	15
6.3 Arcade::Core Class Reference	15
6.3.1 Detailed Description	16
6.3.2 Member Function Documentation	16
6.3.2.1 getNextLibIndex()	16
6.3.2.2 getPreviousLibIndex()	17
6.3.2.3 getRenderer()	17
6.3.2.4 loadRenderer()	17
6.3.2.5 mainLoop()	18
6.3.2.6 runArcade()	18
6.3.2.7 setMode()	18
6.3.2.8 switchLib()	18
6.4 Arcade::Core::CoreException Class Reference	19
6.4.1 Detailed Description	19

6.5 Arcade::EventManager Class Reference	20
6.5.1 Detailed Description	20
6.5.2 Member Function Documentation	20
6.5.2.1 handleEvent()	20
6.6 Arcade::IGame Class Reference	20
6.6.1 Detailed Description	21
6.6.2 Member Function Documentation	21
6.6.2.1 getDirection()	22
6.6.2.2 getGameMode()	22
6.6.2.3 getScore()	22
6.6.2.4 getUserName()	22
6.6.2.5 randomize()	23
6.6.2.6 setDirection()	23
6.7 Arcade::IRenderer Class Reference	24
6.7.1 Detailed Description	25
6.7.2 Member Function Documentation	25
6.7.2.1 createSprite()	25
6.7.2.2 createText()	25
6.7.2.3 displaySprite()	26
6.7.2.4 displayText()	26
6.7.2.5 getEvent()	26
6.7.2.6 getWindow()	27
6.7.2.7 isASCII()	27
6.7.2.8 loadFont()	27
6.7.2.9 loadSound()	28
6.7.2.10 loadTexture()	28
6.7.2.11 moveSprite()	28
6.7.2.12 moveText()	29
6.7.2.13 rotateSprite()	29
6.7.2.14 setSize()	29
6.8 Arcade::IWindow Class Reference	30
6.8.1 Detailed Description	30
6.8.2 Member Function Documentation	30
6.8.2.1 openWindow()	30
6.8.2.2 setTitle()	31
6.9 Arcade::NCursesRenderer 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()	34

6.9.2.5 getEvent()	34
6.9.2.6 getWindow()	34
6.9.2.7 isASCII()	34
6.9.2.8 loadFont()	35
6.9.2.9 loadSound()	36
6.9.2.10 loadTexture()	36
6.9.2.11 moveSprite()	37
6.9.2.12 moveText()	37
6.9.2.13 rotateSprite()	37
6.9.2.14 setSize()	38
6.10 Arcade::NCursesWindow Class Reference	38
6.10.1 Detailed Description	39
6.10.2 Member Function Documentation	39
6.10.2.1 getWindow()	39
6.10.2.2 openWindow()	39
6.10.2.3 setTitle()	39
6.11 Arcade::Nibbler Class Reference	40
6.11.1 Member Function Documentation	41
6.11.1.1 setDirection()	41
6.12 Arcade::RuntimeException Class Reference	41
6.12.1 Detailed Description	41
6.13 Arcade::Sdl2Renderer Class Reference	42
6.13.1 Detailed Description	43
6.13.2 Member Function Documentation	43
6.13.2.1 createSprite()	43
6.13.2.2 createText()	43
6.13.2.3 displaySprite()	44
6.13.2.4 displayText()	44
6.13.2.5 getEvent()	44
6.13.2.6 getWindow()	45
6.13.2.7 isASCII()	45
6.13.2.8 loadFont()	45
6.13.2.9 loadSound()	46
6.13.2.10 loadTexture()	46
6.13.2.11 moveSprite()	46
6.13.2.12 moveText()	47
6.13.2.13 rotateSprite()	47
6.13.2.14 setSize()	47
6.14 Arcade::Sdl2Window Class Reference	48
6.14.1 Detailed Description	48
6.14.2 Member Function Documentation	49
6.14.2.1 getRenderer()	49

6.14.2.2 getWindow()	49
6.14.2.3 openWindow()	49
6.14.2.4 setTitle()	49
6.15 Arcade::SfmlRenderer Class Reference	50
6.15.1 Detailed Description	51
6.15.2 Member Function Documentation	51
6.15.2.1 createSprite()	51
6.15.2.2 createText()	52
6.15.2.3 displaySprite()	52
6.15.2.4 displayText()	52
6.15.2.5 getEvent()	53
6.15.2.6 getWindow()	53
6.15.2.7 isASCII()	53
6.15.2.8 loadFont()	53
6.15.2.9 loadSound()	54
6.15.2.10 loadTexture()	54
6.15.2.11 moveSprite()	55
6.15.2.12 moveText()	55
6.15.2.13 rotateSprite()	55
6.15.2.14 setInput()	56
6.15.2.15 setSize()	56
6.16 Arcade::SfmlWindow Class Reference	56
6.16.1 Detailed Description	57
6.16.2 Member Function Documentation	57
6.16.2.1 getWindow()	57
6.16.2.2 openWindow()	57
6.16.2.3 setTitle()	58
6.17 Arcade::Snake Class Reference	58
6.18 Arcade::Time Class Reference	59
7 File Documentation	61
7.1 Core/include/Arcade/abstractions/AGame.hpp File Reference	61
7.1.1 Detailed Description	61
7.2 Core/include/Arcade/abstractions/IGame.hpp File Reference	61
7.2.1 Detailed Description	62
7.3 Core/include/Arcade/abstractions/IRenderer.hpp File Reference	62
7.3.1 Detailed Description	62
7.4 Core/include/Arcade/abstractions/IWindow.hpp File Reference	62
7.4.1 Detailed Description	63
7.5 Core/include/Arcade/Constants.hpp File Reference	63
7.5.1 Detailed Description	63
7.6 Core/include/Arcade/Core.hpp File Reference	64

7.6.1 Detailed Description	64
7.7 Core/include/Arcade/EventManager.hpp File Reference	64
7.7.1 Detailed Description	65
7.8 Core/include/Arcade/RuntimeException.hpp File Reference	65
7.8.1 Detailed Description	65
7.9 Games/Nibbler/include/Arcade/Nibbler.hpp File Reference	65
7.9.1 Detailed Description	66
7.10 Games/Snake/include/Arcade/Snake.hpp File Reference	66
7.10.1 Detailed Description	66
7.11 Graphics/NCurses/include/Arcade/NCursesRenderer.hpp File Reference	66
7.11.1 Detailed Description	67
7.12 Graphics/NCurses/include/Arcade/NCursesWindow.hpp File Reference	67
7.12.1 Detailed Description	67
7.13 Graphics/SDL2/include/Arcade/Sdl2Renderer.hpp File Reference	67
7.13.1 Detailed Description	68
7.14 Graphics/SDL2/include/Arcade/Sdl2Window.hpp File Reference	68
7.14.1 Detailed Description	68
7.15 Graphics/SFML/include/Arcade/SfmlRenderer.hpp File Reference	68
7.15.1 Detailed Description	69
7.16 Graphics/SFML/include/Arcade/SfmlWindow.hpp File Reference	69
7.16.1 Detailed Description	69
Index	71

Chapter 1

Namespace Index

1.1 Namespace List

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

Arcade	Main namespace for the Arcade project	9
------------------------	---	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Arcade::Clock	14
Arcade::Core	15
Arcade::EventManager	20
std::exception	
Arcade::Core::CoreException	19
Arcade::RuntimeException	41
Arcade::IGame	20
Arcade::AGame	11
Arcade::Nibbler	40
Arcade::Snake	58
Arcade::IRenderer	24
Arcade::NCursesRenderer	31
Arcade::Sdl2Renderer	42
Arcade::SfmlRenderer	50
Arcade::IWindow	30
Arcade::NCursesWindow	38
Arcade::Sdl2Window	48
Arcade::SfmlWindow	56
Arcade::Time	59

Chapter 3

Class Index

3.1 Class List

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

Arcade::AGame	
Game Abstract Class	11
Arcade::Clock	14
Arcade::Core	
Core class	15
Arcade::Core::CoreException	
CoreException class	19
Arcade::EventManager	
EventManager class	20
Arcade::IGame	
Game Interface	20
Arcade::IRenderer	
Graphic Interface	24
Arcade::IWindow	
Window Interface	30
Arcade::NCursesRenderer	
NCursesRenderer class	31
Arcade::NCursesWindow	
NCursesWindow class	38
Arcade::Nibbler	40
Arcade::RuntimeException	
RuntimeException class	41
Arcade::Sdl2Renderer	
Sdl2Renderer class	42
Arcade::Sdl2Window	
Sdl2Window class	48
Arcade::SfmlRenderer	
SfmlRenderer class	50
Arcade::SfmlWindow	
SfmlWindow class	56
Arcade::Snake	58
Arcade::Time	59

Chapter 4

File Index

4.1 File List

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

Core/include/Arcade/ Constants.hpp	
File containing constant values used in the Arcade project	63
Core/include/Arcade/ Core.hpp	
File declaring Core	64
Core/include/Arcade/ EventManager.hpp	
File declaring EventsManager	64
Core/include/Arcade/ RuntimeException.hpp	
File declaring RuntimeException	65
Core/include/Arcade/abstractions/ AGame.hpp	
File declaring Game Abstract Class	61
Core/include/Arcade/abstractions/ IGame.hpp	
File declaring Game Interface	61
Core/include/Arcade/abstractions/ IRenderer.hpp	
File declaring Graphic Interface	62
Core/include/Arcade/abstractions/ IWindow.hpp	
File declaring Window Interface	62
Core/include/Arcade/Clock/ Clock.hpp	??
Core/include/Arcade/Clock/ Time.hpp	??
Games/Nibbler/include/Arcade/ Nibbler.hpp	
Defines the Nibbler class	65
Games/Snake/include/Arcade/ Snake.hpp	
Defines the Snake class	66
Graphics/NCurses/include/Arcade/ NCursesRenderer.hpp	
File declaring NCursesRenderer	66
Graphics/NCurses/include/Arcade/ NCursesWindow.hpp	
File declaring NCursesWindow	67
Graphics/SDL2/include/Arcade/ Sdl2Renderer.hpp	
File declaring Sdl2Renderer class	67
Graphics/SDL2/include/Arcade/ Sdl2Window.hpp	
File declaring Sdl2Window class	68
Graphics/SFML/include/Arcade/ SfmlRenderer.hpp	
File declaring SfmlRenderer class	68
Graphics/SFML/include/Arcade/ SfmlWindow.hpp	
File declaring SfmlWindow class	69

Chapter 5

Namespace Documentation

5.1 Arcade Namespace Reference

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

Classes

- class [AGame](#)
Game Abstract Class.
- class [IGame](#)
Game Interface.
- class [IRenderer](#)
Graphic Interface.
- class [IWindow](#)
Window Interface.
- class [Clock](#)
- class [Time](#)
- class [Core](#)
Core class.
- class [EventManager](#)
EventManager class.
- class [RuntimeException](#)
RuntimeException class.
- class [Sdl2Renderer](#)
Sdl2Renderer class.
- class [Sdl2Window](#)
Sdl2Window class.
- class [NCursesRenderer](#)
NCursesRenderer class.
- class [NCursesWindow](#)
NCursesWindow class.
- class [SfmlRenderer](#)
SfmlRenderer class.
- class [SfmlWindow](#)
SfmlWindow class.
- class [Nibbler](#)
- class [Snake](#)

Enumerations

- enum class **CoreMode** { **GAME** , **LOGIN** , **MENU** , **QUIT** }
- enum class **GameMode** { **GAME** , **GAME_OVER** , **LOGIN** , **MENU** , **PAUSE** }
- enum class **Direction** { **UP** , **DOWN** , **LEFT** , **RIGHT** , **NONE** }
- enum class **KeyboardEvents** { **DOWN** , **ENTER** , **ESC** , **F1** , **F2** , **LEFT** , **RIGHT** , **TEXTENTERED** , **UP** , **NONE** }

Enum for the KeyboardEvents.

5.1.1 Detailed Description

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

5.1.2 Enumeration Type Documentation

5.1.2.1 KeyboardEvents

```
enum Arcade::KeyboardEvents [strong]
```

Enum for the KeyboardEvents.

Set of enum used for the game event

Chapter 6

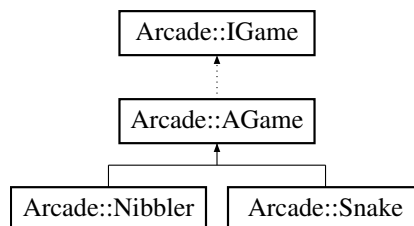
Class Documentation

6.1 Arcade::AGame Class Reference

Game Abstract Class.

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

Inheritance diagram for Arcade::AGame:



Public Member Functions

- void **start** (std::reference_wrapper< [IRenderer](#) > renderer, const [Clock](#) &clock) override
start the game
- void **stop** () override
stop the game
- int **getScore** () const override
Get the score.
- GameMode **getGameMode** () const override
Get the game mode.
- Direction **getDirection** () const override
Get the direction.
- void **setDirection** (const Direction &direction) override
Set the direction.
- void **getUserName** (const std::string &userName) override
Get the user name.
- int **randomize** (int min, int max) override
Randomize a number.

Protected Attributes

- `std::vector< std::vector< std::pair< float, float > > > _map {0}`
- `int _score {0}`
- `Clock _clock {}`
- `int _lastMilliseconds {0}`
- `std::string _userName {}`
- `IRenderer * _renderer {nullptr}`
- `GameMode _gameMode {GameMode::GAME}`
- `Direction _direction {Direction::RIGHT}`
- `std::vector< Direction > _prevDirection {0, Direction::NONE}`

6.1.1 Detailed Description

Game Abstract Class.

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

6.1.2 Member Function Documentation

6.1.2.1 `getDirection()`

```
Direction Arcade::AGame::getDirection ( ) const [inline], [override], [virtual]
```

Get the direction.

Returns

The direction

Implements [Arcade::IGame](#).

6.1.2.2 `getGameMode()`

```
GameMode Arcade::AGame::getGameMode ( ) const [inline], [override], [virtual]
```

Get the game mode.

Returns

The game mode

Implements [Arcade::IGame](#).

6.1.2.3 getScore()

```
int Arcade::AGame::getScore ( ) const [inline], [override], [virtual]
```

Get the score.

Returns

The score

Implements [Arcade::IGame](#).

6.1.2.4 getUserName()

```
void Arcade::AGame::getUserName (
    const std::string & name ) [inline], [override], [virtual]
```

Get the user name.

Parameters

<i>name</i>	The user name
-------------	---------------

Implements [Arcade::IGame](#).

6.1.2.5 randomize()

```
int Arcade::AGame::randomize (
    int min,
    int max ) [inline], [override], [virtual]
```

Randomize a number.

Parameters

<i>min</i>	The minimum value
<i>max</i>	The maximum value

Returns

The random number

Implements [Arcade::IGame](#).

6.1.2.6 setDirection()

```
void Arcade::AGame::setDirection (
    const Direction & direction ) [inline], [override], [virtual]
```

Set the direction.

Parameters

<i>direction</i>	The direction
------------------	---------------

Implements [Arcade::IGame](#).

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

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

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

6.2 Arcade::Clock Class Reference

Public Types

- using [TimePoint](#) = std::chrono::time_point< std::chrono::high_resolution_clock >
TimePoint is a type alias for a time point which is a very long and complicated type in the standard library.

Public Member Functions

- [Clock](#) ()
Construct a new [Clock](#) object.
- void [restart](#) ()
Restart the clock.
- void [pause](#) ()
Pause the clock.
- void [resume](#) ()
Resume the clock.
- [Time](#) [getElapsedTime](#) () const
Get the elapsed time since the last restart.

6.2.1 Member Function Documentation

6.2.1.1 getElapsedTime()

```
Time Arcade::Clock::getElapsedTime ( ) const
```

Get the elapsed time since the last restart.

Returns

[Time](#) The elapsed time

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

- Core/include/Arcade/Clock/Clock.hpp

6.3 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
- int [runArcade](#) (const std::string &path)
Run Arcade.
- void [getLibraries](#) (const std::string &libPath)
Get actives libraries.
- void [closeGameLibrary](#) ()
Close game library.
- void [switchGraphicLibrary](#) ()
Switch Graphic library.
- template<typename T >
void [switchLib](#) (const std::string &libPath)
Switch library.
- void [mainLoop](#) ()
Game loop.
- void [setMode](#) (const CoreMode &gameMode)
Set the game mode.
- void [loadRenderer](#) (const std::string &path)
Load renderer.

- void `loadMenu` ()
load Menu
- void `displayMenu` ()
display menu
- `IRenderer` * `getRenderer` ()
Get the renderer.
- void `moveCursorDown` ()
Move cursor down, in menu.
- void `moveCursorUp` ()
Move cursor up, in menu.
- void `launchGame` ()
Launch game, call to entry point games.
- void `setGameIndex` (size_t index)
set game index, used for launch game
- CoreMode `getMode` ()
get core mode
- void `setUserName` (const std::string &userName)
set user name
- void `loadLogin` ()
load Login page, instead of game or menu page
- void `displayLoginScreen` ()
display login page

Static Public Member Functions

- static size_t `getNextLibIndex` (std::vector< std::string > &libs, size_t currentIndex)
Get next library index.
- static size_t `getPreviousLibIndex` (std::vector< std::string > &libs, size_t currentIndex)
Get previous library index.

6.3.1 Detailed Description

`Core` class.

`Core` class for the `Arcade` project

6.3.2 Member Function Documentation

6.3.2.1 getNextLibIndex()

```
static size_t Arcade::Core::getNextLibIndex (
    std::vector< std::string > & libs,
    size_t currentIndex ) [static]
```

Get next library index.

Parameters

<i>libs</i>	
<i>currentIndex</i>	

Returns

size_t

6.3.2.2 getPreviousLibIndex()

```
static size_t Arcade::Core::getPreviousLibIndex (
    std::vector< std::string > & libs,
    size_t currentIndex ) [static]
```

Get previous library index.

Parameters

<i>libs</i>	
<i>currentIndex</i>	

Returns

size_t

6.3.2.3 getRenderer()

```
IRenderer* Arcade::Core::getRenderer ( ) [inline]
```

Get the renderer.

Returns

The renderer

6.3.2.4 loadRenderer()

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

Load renderer.

Parameters

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

6.3.2.5 mainLoop()

```
void Arcade::Core::mainLoop ( )
```

Game loop.

Main game loop

6.3.2.6 runArcade()

```
int Arcade::Core::runArcade (
    const std::string & path )
```

Run [Arcade](#).

Parameters

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

Returns

int

6.3.2.7 setMode()

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

Set the game mode.

Parameters

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

6.3.2.8 switchLib()

```
template<typename T >
```

```
void Arcade::Core::switchLib (
    const std::string & libPath )
```

Switch library.

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>libPath</i>	
----------------	--

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

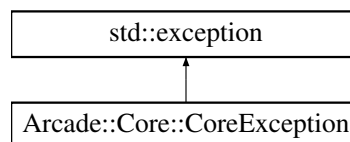
- Core/include/Arcade/[Core.hpp](#)

6.4 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

6.4.1 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](#)

6.5 Arcade::EventManager Class Reference

[EventManager](#) class.

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

Static Public Member Functions

- `template<typename T >`
`static void handleEvent (const std::map< const Arcade::KeyboardEvents, std::function< void(T &)>>>`
`&eventMap, T &object, const Arcade::KeyboardEvents &event)`
Handle the event.

6.5.1 Detailed Description

[EventManager](#) class.

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

6.5.2 Member Function Documentation

6.5.2.1 `handleEvent()`

```
template<typename T >
static void Arcade::EventManager::handleEvent (
    const std::map< const Arcade::KeyboardEvents, std::function< void(T &)>>> & eventMap,
    T & object,
    const Arcade::KeyboardEvents & event ) [inline], [static]
```

Handle the event.

Parameters

<i>eventMap</i>	The map of events
<i>object</i>	The object to handle
<i>event</i>	The event to handle

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

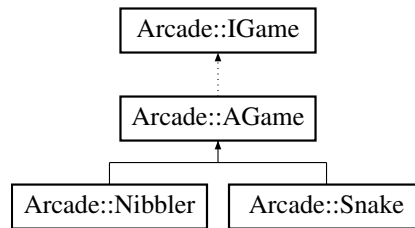
- `Core/include/Arcade/EventsManager.hpp`

6.6 Arcade::IGame Class Reference

Game Interface.

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

Inheritance diagram for Arcade::IGame:



Public Member Functions

- virtual void [start](#) (std::reference_wrapper< [IRenderer](#) > renderer, const [Clock](#) &clock)=0
start the game
- virtual void [loadGame](#) ()=0
load the game
- virtual void [gameLoop](#) ()=0
display the game
- virtual void [stop](#) ()=0
stop the game
- virtual void [handleEvents](#) (const [KeyboardEvents](#) &event)=0
handle the game event
- virtual int [getScore](#) () const =0
Get the score.
- virtual GameMode [getGameMode](#) () const =0
Get the game mode.
- virtual Direction [getDirection](#) () const =0
Get the direction.
- virtual void [setDirection](#) (const Direction &direction)=0
Set the direction.
- virtual int [randomize](#) (int min, int max)=0
Randomize a number.
- virtual void [getUserName](#) (const std::string &name)=0
Get the user name.

6.6.1 Detailed Description

Game Interface.

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

6.6.2 Member Function Documentation

6.6.2.1 `getDirection()`

```
virtual Direction Arcade::IGame::getDirection ( ) const [pure virtual]
```

Get the direction.

Returns

The direction

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

6.6.2.2 `getGameMode()`

```
virtual GameMode Arcade::IGame::getGameMode ( ) const [pure virtual]
```

Get the game mode.

Returns

The game mode

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

6.6.2.3 `getScore()`

```
virtual int Arcade::IGame::getScore ( ) const [pure virtual]
```

Get the score.

Returns

The score

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

6.6.2.4 `getUserName()`

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

Get the user name.

Parameters

<i>name</i>	The user name
-------------	---------------

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

6.6.2.5 randomize()

```
virtual int Arcade::IGame::randomize (
    int min,
    int max ) [pure virtual]
```

Randomize a number.

Parameters

<i>min</i>	The minimum value
<i>max</i>	The maximum value

Returns

The random number

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

6.6.2.6 setDirection()

```
virtual void Arcade::IGame::setDirection (
    const Direction & direction ) [pure virtual]
```

Set the direction.

Parameters

<i>direction</i>	The direction
------------------	---------------

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

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

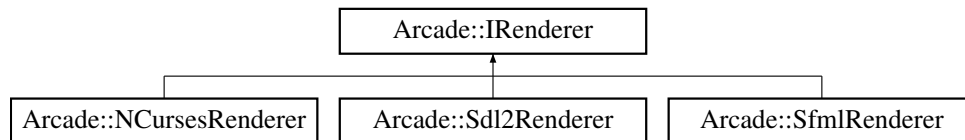
- Core/include/Arcade/abstractions/[IGame.hpp](#)

6.7 Arcade::IRenderer Class Reference

Graphic Interface.

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

Inheritance diagram for Arcade::IRenderer:



Public Member Functions

- virtual [IWindow](#) * [getWindow](#) ()=0
Retrieve the window associated with the renderer.
- virtual [KeyboardEvents](#) [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.
- virtual void [moveText](#) (const std::string &textName, int pos_x, int pos_y)=0
Move a text object to the specified position.
- virtual void [loadSound](#) (const std::string &filePath)=0
Load a sound for menu.
- virtual void [stopSound](#) ()=0
Stop the sound for menu.
- virtual void [moveSprite](#) (const std::string &spriteName, float pos_x, float pos_y)=0
Move sprite to the specified position.
- virtual void [rotateSprite](#) (const std::string &spriteName, float angle)=0
Rotate sprite to the specified angle.
- virtual std::string [updateTextBox](#) ()=0
Update the text box for username input.
- virtual void [setSize](#) (float x, float y)=0
Set the size of the renderer.
- virtual void [deleteResources](#) ()=0
Clear the renderer.

6.7.1 Detailed Description

Graphic Interface.

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

6.7.2 Member Function Documentation

6.7.2.1 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::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.2 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::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.3 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::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.4 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::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.5 getEvent()

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

Retrieve the current event from the window.

Returns

The current event

Implemented in [Arcade::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.6 getWindow()

```
virtual IWindow* Arcade::IRenderer::getWindow ( ) [pure virtual]
```

Retrieve the window associated with the renderer.

Returns

The window associated with the renderer

Implemented in [Arcade::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.7 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::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.8 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::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.9 loadSound()

```
virtual void Arcade::IRenderer::loadSound (
    const std::string & filePath ) [pure virtual]
```

Load a sound for menu.

Parameters

<i>filePath</i>	The path to the sound file
-----------------	----------------------------

Implemented in [Arcade::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.10 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::SfmlRenderer](#), [Arcade::NCursesRenderer](#), and [Arcade::Sdl2Renderer](#).

6.7.2.11 moveSprite()

```
virtual void Arcade::IRenderer::moveSprite (
    const std::string & spriteName,
    float pos_x,
    float pos_y ) [pure virtual]
```

Move sprite to the specified position.

Parameters

<i>spriteName</i>	The name of the sprite to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implemented in [Arcade::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.12 moveText()

```
virtual void Arcade::IRenderer::moveText (
    const std::string & textName,
    int pos_x,
    int pos_y ) [pure virtual]
```

Move a text object to the specified position.

Parameters

<i>textName</i>	The name of the text to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implemented in [Arcade::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.13 rotateSprite()

```
virtual void Arcade::IRenderer::rotateSprite (
    const std::string & spriteName,
    float angle ) [pure virtual]
```

Rotate sprite to the specified angle.

Parameters

<i>spriteName</i>	The name of the sprite to rotate
<i>angle</i>	The angle to rotate the sprite

Implemented in [Arcade::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

6.7.2.14 setSize()

```
virtual void Arcade::IRenderer::setSize (
    float x,
    float y ) [pure virtual]
```

Set the size of the renderer.

Parameters

<i>x</i>	The width of the renderer
<i>y</i>	The height of the renderer

Implemented in [Arcade::SfmlRenderer](#), [Arcade::Sdl2Renderer](#), and [Arcade::NCursesRenderer](#).

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

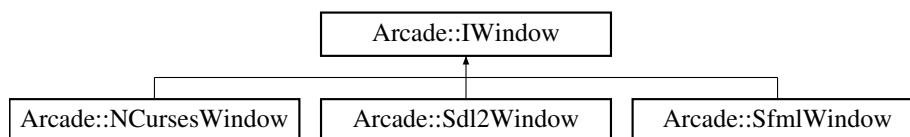
- [Core/include/Arcade/abstractions/IRenderer.hpp](#)

6.8 Arcade::IWindow Class Reference

Window Interface.

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

Inheritance diagram for Arcade::IWindow:



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.

6.8.1 Detailed Description

Window Interface.

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

6.8.2 Member Function Documentation

6.8.2.1 openWindow()

```
virtual void Arcade::IWindow::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::SfmlWindow](#), [Arcade::Sdl2Window](#), and [Arcade::NCursesWindow](#).

6.8.2.2 setTitle()

```
virtual void Arcade::IWindow::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::SfmlWindow](#), [Arcade::Sdl2Window](#), and [Arcade::NCursesWindow](#).

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

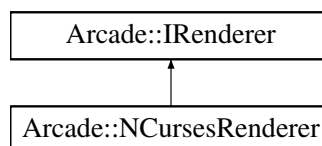
- [Core/include/Arcade/abstractions/IWindow.hpp](#)

6.9 Arcade::NCursesRenderer Class Reference

[NCursesRenderer](#) class.

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

Inheritance diagram for Arcade::NCursesRenderer:

**Public Member Functions**

- [KeyboardEvents](#) [getEvent](#) () override
Retrieve the current event from the window.
- bool [loadTexture](#) (const std::string &texture, const std::string &name) override
Load a texture 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 [displaySprite](#) (const std::string &spriteName) override

Display a sprite with the specified name.
- bool [loadFont](#) (const std::string &filepath, const std::string &name) override

Load a font from a file.
- 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 [displayText](#) (const std::string &textName) override

Display a text object with the specified name.
- void [moveText](#) (const std::string &textName, int pos_x, int pos_y) override

Move a text object to the specified position.
- bool [isASCII](#) () override

Check if the renderer supports only ASCII characters.
- [IWindow](#) * [getWindow](#) () override

Retrieve the window associated with the renderer.
- void [loadSound](#) (const std::string &filePath) override

Load a sound for menu.
- void [stopSound](#) () override

Stop the sound for menu.
- void [moveSprite](#) (const std::string &spriteName, float pos_x, float pos_y) override

Move sprite to the specified position.
- void [rotateSprite](#) (const std::string &spriteName, float angle) override

Rotate sprite to the specified angle.
- std::string [updateTextBox](#) () override

Update the text box for username input.
- void [setSize](#) (float x, float y) override

Set the size of the renderer.
- void [deleteResources](#) () override

Clear the renderer.

6.9.1 Detailed Description

[NCursesRenderer](#) class.

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

6.9.2 Member Function Documentation

6.9.2.1 createSprite()

```
void Arcade::NCursesRenderer::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](#).

6.9.2.2 createText()

```
void Arcade::NCursesRenderer::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::NCursesRenderer::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](#).

6.9.2.4 displayText()

```
void Arcade::NCursesRenderer::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()

```
KeyboardEvents Arcade::NCursesRenderer::getEvent ( ) [override], [virtual]
```

Retrieve the current event from the window.

Returns

The current event

Implements [Arcade::IRenderer](#).

6.9.2.6 getWindow()

```
IWindow\* Arcade::NCursesRenderer::getWindow ( ) [inline], [override], [virtual]
```

Retrieve the window associated with the renderer.

Returns

The window associated with the renderer

Implements [Arcade::IRenderer](#).

6.9.2.7 isASCII()

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

Check if the renderer supports only ASCII characters.

Returns

True if the renderer supports ASCII, false otherwise

Implements [Arcade::IRenderer](#).

6.9.2.8 loadFont()

```
bool Arcade::NCursesRenderer::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](#).

6.9.2.9 loadSound()

```
void Arcade::NCursesRenderer::loadSound (
    const std::string & filePath ) [inline], [override], [virtual]
```

Load a sound for menu.

Parameters

<i>filePath</i>	The path to the sound file
-----------------	----------------------------

Implements [Arcade::IRenderer](#).

6.9.2.10 loadTexture()

```
bool Arcade::NCursesRenderer::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](#).

6.9.2.11 moveSprite()

```
void Arcade::NCursesRenderer::moveSprite (
    const std::string & spriteName,
    float pos_x,
    float pos_y ) [inline], [override], [virtual]
```

Move sprite to the specified position.

Parameters

<i>spriteName</i>	The name of the sprite to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implements [Arcade::IRenderer](#).

6.9.2.12 moveText()

```
void Arcade::NCursesRenderer::moveText (
    const std::string & textName,
    int pos_x,
    int pos_y ) [override], [virtual]
```

Move a text object to the specified position.

Parameters

<i>textName</i>	The name of the text to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implements [Arcade::IRenderer](#).

6.9.2.13 rotateSprite()

```
void Arcade::NCursesRenderer::rotateSprite (
    const std::string & spriteName,
    float angle ) [inline], [override], [virtual]
```

Rotate sprite to the specified angle.

Parameters

<i>spriteName</i>	The name of the sprite to rotate
<i>angle</i>	The angle to rotate the sprite

Implements [Arcade::IRenderer](#).

6.9.2.14 setSize()

```
void Arcade::NCursesRenderer::setSize (
    float x,
    float y ) [inline], [override], [virtual]
```

Set the size of the renderer.

Parameters

<i>x</i>	The width of the renderer
<i>y</i>	The height of the renderer

Implements [Arcade::IRenderer](#).

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

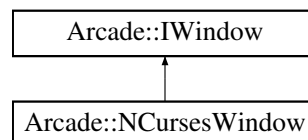
- Graphics/NCurses/include/Arcade/[NCursesRenderer.hpp](#)

6.10 Arcade::NCursesWindow Class Reference

[NCursesWindow](#) class.

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

Inheritance diagram for `Arcade::NCursesWindow`:



Public Member Functions

- 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.
- WINDOW * [getWindow](#) ()
Get the current window.

6.10.1 Detailed Description

[NCursesWindow](#) class.

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

6.10.2 Member Function Documentation

6.10.2.1 getWindow()

```
WINDOW* Arcade::NCursesWindow::getWindow ( ) [inline]
```

Get the current window.

Returns

The current window

6.10.2.2 openWindow()

```
void Arcade::NCursesWindow::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::IWindow](#).

6.10.2.3 setTitle()

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

Set the title of the window.

Parameters

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

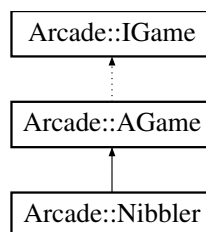
Implements [Arcade::IWindow](#).

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

- Graphics/NCurses/include/Arcade/NCursesWindow.hpp

6.11 Arcade::Nibbler Class Reference

Inheritance diagram for Arcade::Nibbler:



Public Member Functions

- void [loadGame](#) () override
load the game
- void [gameLoop](#) () override
display the game
- void [handleEvents](#) (const [KeyboardEvents](#) &event) override
handle the game event
- void [setDirection](#) (const Direction &direction) override
Set the direction.
- void **moveSnake** ()
- void **moveBody** ()
- void **createMap** ()
- void **createWall** ()
- void **checkLose** ()
- void **displaySnake** ()
- void **isEating** ()
- void **checkIfCanMove** (std::pair< size_t, size_t > pos, const Direction &direction)
- std::pair< size_t, size_t > **getBodyPosition** ()
- bool **findDirection** (const Direction &direction)
- void **createFood** ()
- void **displayFood** ()
- void **loadNibbler** ()
- void **loadASCIINibbler** ()
- void **createSnake** ()

Additional Inherited Members

6.11.1 Member Function Documentation

6.11.1.1 setDirection()

```
void Arcade::Nibbler::setDirection (
    const Direction & direction ) [override], [virtual]
```

Set the direction.

Parameters

<i>direction</i>	The direction
------------------	---------------

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

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

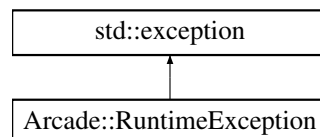
- Games/Nibbler/include/Arcade/[Nibbler.hpp](#)

6.12 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

6.12.1 Detailed Description

[RuntimeException](#) class.

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

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

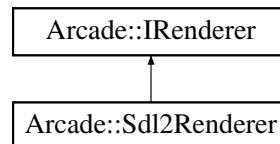
- Core/include/Arcade/[RuntimeException.hpp](#)

6.13 Arcade::Sdl2Renderer Class Reference

[Sdl2Renderer](#) class.

```
#include <Arcade/Sdl2Renderer.hpp>
```

Inheritance diagram for Arcade::Sdl2Renderer:



Public Member Functions

- [KeyboardEvents](#) [getEvent](#) () override
Retrieve the current event from 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.
- void [moveText](#) (const std::string &textName, int pos_x, int pos_y) override
Move a text object to the specified position.
- bool [isASCII](#) () override
Check if the renderer supports only ASCII characters.
- [IWindow](#) * [getWindow](#) () override
Retrieve the window associated with the renderer.
- void [loadSound](#) (const std::string &filePath) override
Load a sound for menu.
- void [stopSound](#) () override
Stop the sound for menu.
- std::string [updateTextBox](#) () override
Update the text box for username input.
- void [setSize](#) (float x, float y) override
Set the size of the renderer.
- void [moveSprite](#) (const std::string &spriteName, float pos_x, float pos_y) override
Move sprite to the specified position.
- void [rotateSprite](#) (const std::string &spriteName, float angle) override
Rotate sprite to the specified angle.
- void [deleteResources](#) () override
Clear the renderer.

6.13.1 Detailed Description

[Sdl2Renderer](#) class.

[Sdl2Renderer](#) class for the [Arcade](#) project

6.13.2 Member Function Documentation

6.13.2.1 createSprite()

```
void Arcade::Sdl2Renderer::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](#).

6.13.2.2 createText()

```
void Arcade::Sdl2Renderer::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](#).

6.13.2.3 displaySprite()

```
void Arcade::Sdl2Renderer::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](#).

6.13.2.4 displayText()

```
void Arcade::Sdl2Renderer::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](#).

6.13.2.5 getEvent()

```
KeyboardEvents Arcade::Sdl2Renderer::getEvent ( ) [override], [virtual]
```

Retrieve the current event from the window.

Returns

The current event

Implements [Arcade::IRenderer](#).

6.13.2.6 getWindow()

```
IWindow* Arcade::Sdl2Renderer::getWindow ( ) [inline], [override], [virtual]
```

Retrieve the window associated with the renderer.

Returns

The window associated with the renderer

Implements [Arcade::IRenderer](#).

6.13.2.7 isASCII()

```
bool Arcade::Sdl2Renderer::isASCII ( ) [inline], [override], [virtual]
```

Check if the renderer supports only ASCII characters.

Returns

True if the renderer supports ASCII, false otherwise

Implements [Arcade::IRenderer](#).

6.13.2.8 loadFont()

```
bool Arcade::Sdl2Renderer::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](#).

6.13.2.9 loadSound()

```
void Arcade::Sdl2Renderer::loadSound (
    const std::string & filePath ) [override], [virtual]
```

Load a sound for menu.

Parameters

<i>filePath</i>	The path to the sound file
-----------------	----------------------------

Implements [Arcade::IRenderer](#).

6.13.2.10 loadTexture()

```
bool Arcade::Sdl2Renderer::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](#).

6.13.2.11 moveSprite()

```
void Arcade::Sdl2Renderer::moveSprite (
    const std::string & spriteName,
    float pos_x,
    float pos_y ) [override], [virtual]
```

Move sprite to the specified position.

Parameters

<i>spriteName</i>	The name of the sprite to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implements [Arcade::IRenderer](#).

6.13.2.12 moveText()

```
void Arcade::Sdl2Renderer::moveText (
    const std::string & textName,
    int pos_x,
    int pos_y ) [override], [virtual]
```

Move a text object to the specified position.

Parameters

<i>textName</i>	The name of the text to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implements [Arcade::IRenderer](#).

6.13.2.13 rotateSprite()

```
void Arcade::Sdl2Renderer::rotateSprite (
    const std::string & spriteName,
    float angle ) [override], [virtual]
```

Rotate sprite to the specified angle.

Parameters

<i>spriteName</i>	The name of the sprite to rotate
<i>angle</i>	The angle to rotate the sprite

Implements [Arcade::IRenderer](#).

6.13.2.14 setSize()

```
void Arcade::Sdl2Renderer::setSize (
    float x,
    float y ) [inline], [override], [virtual]
```

Set the size of the renderer.

Parameters

<i>x</i>	The width of the renderer
<i>y</i>	The height of the renderer

Implements [Arcade::IRenderer](#).

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

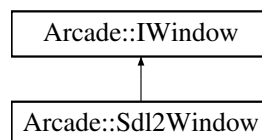
- Graphics/SDL2/include/Arcade/[Sdl2Renderer.hpp](#)

6.14 Arcade::Sdl2Window Class Reference

[Sdl2Window](#) class.

```
#include <Arcade/Sdl2Window.hpp>
```

Inheritance diagram for Arcade::Sdl2Window:



Public Member Functions

- 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.
- SDL_Window * [getWindow](#) ()
Get the current window.
- SDL_Renderer * [getRenderer](#) ()
Get the current renderer.

6.14.1 Detailed Description

[Sdl2Window](#) class.

[Sdl2Window](#) class for the [Arcade](#) project

6.14.2 Member Function Documentation

6.14.2.1 getRenderer()

```
SDL_Renderer* Arcade::Sdl2Window::getRenderer ( ) [inline]
```

Get the current renderer.

Returns

The current renderer

6.14.2.2 getWindow()

```
SDL_Window* Arcade::Sdl2Window::getWindow ( ) [inline]
```

Get the current window.

Returns

The current window

6.14.2.3 openWindow()

```
void Arcade::Sdl2Window::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::IWindow](#).

6.14.2.4 setTitle()

```
void Arcade::Sdl2Window::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::IWindow](#).

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

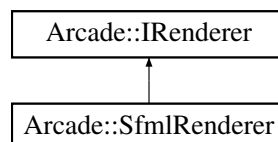
- Graphics/SDL2/include/Arcade/[Sdl2Window.hpp](#)

6.15 Arcade::SfmlRenderer Class Reference

[SfmlRenderer](#) class.

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

Inheritance diagram for `Arcade::SfmlRenderer`:



Public Member Functions

- bool [loadFont](#) (const std::string &filepath, const std::string &name) override
Load a font from a file.
- 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 [moveText](#) (const std::string &textName, int pos_x, int pos_y) override
Move a text object to the specified position.
- void [displayText](#) (const std::string &textName) override
Display a text object with the specified name.
- bool [loadTexture](#) (const std::string &texture, const std::string &name) override
Load a texture 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 [displaySprite](#) (const std::string &spriteName) override
Display a sprite with the specified name.
- std::string [updateTextBox](#) () override
Update the text box for username input.
- void [loadSound](#) (const std::string &filePath) override
Load a sound for menu.
- void [stopSound](#) () override
Stop the sound for menu.

- [IWindow](#) * [getWindow](#) () override
Retrieve the window associated with the renderer.
- [KeyboardEvents](#) [getEvent](#) () override
Retrieve the current event from the window.
- bool [isASCII](#) () override
Check if the renderer supports only ASCII characters.
- void [setSize](#) (float x, float y) override
Set the size of the renderer.
- void [setInput](#) (const sf::Event &event)
Set the Input object.
- void [moveSprite](#) (const std::string &spriteName, float pos_x, float pos_y) override
Move sprite to the specified position.
- void [rotateSprite](#) (const std::string &spriteName, float angle) override
Rotate sprite to the specified angle.
- void [deleteResources](#) () override
Clear the renderer.

6.15.1 Detailed Description

[SfmlRenderer](#) class.

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

6.15.2 Member Function Documentation

6.15.2.1 createSprite()

```
void Arcade::SfmlRenderer::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](#).

6.15.2.2 createText()

```
void Arcade::SfmlRenderer::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](#).

6.15.2.3 displaySprite()

```
void Arcade::SfmlRenderer::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](#).

6.15.2.4 displayText()

```
void Arcade::SfmlRenderer::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](#).

6.15.2.5 getEvent()

```
KeyboardEvents Arcade::SfmlRenderer::getEvent ( ) [override], [virtual]
```

Retrieve the current event from the window.

Returns

The current event

Implements [Arcade::IRenderer](#).

6.15.2.6 getWindow()

```
IWindow* Arcade::SfmlRenderer::getWindow ( ) [inline], [override], [virtual]
```

Retrieve the window associated with the renderer.

Returns

The window associated with the renderer

Implements [Arcade::IRenderer](#).

6.15.2.7 isASCII()

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

Check if the renderer supports only ASCII characters.

Returns

True if the renderer supports ASCII, false otherwise

Implements [Arcade::IRenderer](#).

6.15.2.8 loadFont()

```
bool Arcade::SfmlRenderer::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](#).

6.15.2.9 loadSound()

```
void Arcade::SfmlRenderer::loadSound (
    const std::string & filePath ) [override], [virtual]
```

Load a sound for menu.

Parameters

<i>filePath</i>	The path to the sound file
-----------------	----------------------------

Implements [Arcade::IRenderer](#).

6.15.2.10 loadTexture()

```
bool Arcade::SfmlRenderer::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](#).

6.15.2.11 moveSprite()

```
void Arcade::SfmlRenderer::moveSprite (
    const std::string & spriteName,
    float pos_x,
    float pos_y ) [override], [virtual]
```

Move sprite to the specified position.

Parameters

<i>spriteName</i>	The name of the sprite to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implements [Arcade::IRenderer](#).

6.15.2.12 moveText()

```
void Arcade::SfmlRenderer::moveText (
    const std::string & textName,
    int pos_x,
    int pos_y ) [override], [virtual]
```

Move a text object to the specified position.

Parameters

<i>textName</i>	The name of the text to move
<i>pos_x</i>	The x-coordinate of the new position
<i>pos_y</i>	The y-coordinate of the new position

Implements [Arcade::IRenderer](#).

6.15.2.13 rotateSprite()

```
void Arcade::SfmlRenderer::rotateSprite (
    const std::string & spriteName,
    float angle ) [override], [virtual]
```

Rotate sprite to the specified angle.

Parameters

<i>spriteName</i>	The name of the sprite to rotate
<i>angle</i>	The angle to rotate the sprite

Implements [Arcade::IRenderer](#).

6.15.2.14 setInput()

```
void Arcade::SfmlRenderer::setInput (
    const sf::Event & event )
```

Set the Input object.

used for username input

Parameters

<i>event</i>	The event to set
--------------	------------------

6.15.2.15 setSize()

```
void Arcade::SfmlRenderer::setSize (
    float x,
    float y ) [inline], [override], [virtual]
```

Set the size of the renderer.

Parameters

<i>x</i>	The width of the renderer
<i>y</i>	The height of the renderer

Implements [Arcade::IRenderer](#).

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

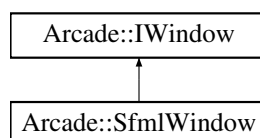
- Graphics/SFML/include/Arcade/[SfmlRenderer.hpp](#)

6.16 Arcade::SfmlWindow Class Reference

[SfmlWindow](#) class.

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

Inheritance diagram for Arcade::SfmlWindow:



Public Member Functions

- 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.
- sf::RenderWindow * [getWindow](#) ()
Get the current window.

6.16.1 Detailed Description

[SfmlWindow](#) class.

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

6.16.2 Member Function Documentation

6.16.2.1 getWindow()

```
sf::RenderWindow* Arcade::SfmlWindow::getWindow ( ) [inline]
```

Get the current window.

Returns

The current window

6.16.2.2 openWindow()

```
void Arcade::SfmlWindow::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::IWindow](#).

6.16.2.3 setTitle()

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

Set the title of the window.

Parameters

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

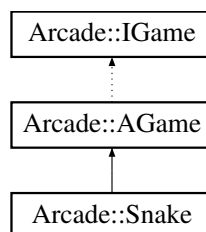
Implements [Arcade::IWindow](#).

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

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

6.17 Arcade::Snake Class Reference

Inheritance diagram for Arcade::Snake:



Public Member Functions

- void [loadGame](#) () override
load the game
- void [gameLoop](#) () override
display the game
- void [handleEvents](#) (const [KeyboardEvents](#) &event) override
handle the game event
- void **moveSnake** ()
- void **moveBody** ()
- void **createMap** ()
- void **replaceFood** ()
- void **checkLose** ()
- void **displaySnake** ()
- void **isEating** ()
- std::pair< size_t, size_t > **getBodyPosition** ()
- void **loadSnake** ()
- void **loadAsciiSnake** ()

Additional Inherited Members

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

- Games/Snake/include/Arcade/[Snake.hpp](#)

6.18 Arcade::Time Class Reference

Public Member Functions

- [Time](#) (double seconds)
Construct a new [Time](#) object.
- int [asSeconds](#) () const
Transform the time to seconds.
- int [asMilliseconds](#) () const
Transform the time to milliseconds.
- int [asMicroseconds](#) () const
Transform the time to microseconds.

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

- Core/include/Arcade/Clock/Time.hpp

Chapter 7

File Documentation

7.1 Core/include/Arcade/abstractions/AGame.hpp File Reference

File declaring Game Abstract Class.

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

Classes

- class [Arcade::AGame](#)
Game Abstract Class.

Namespaces

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

7.1.1 Detailed Description

File declaring Game Abstract Class.

7.2 Core/include/Arcade/abstractions/IGame.hpp File Reference

File declaring Game Interface.

```
#include <functional>
#include "Arcade/abstractions/IRenderer.hpp"
#include "Arcade/Constants.hpp"
#include "Arcade/Clock/Clock.hpp"
```

Classes

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

Namespaces

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

7.2.1 Detailed Description

File declaring Game Interface.

7.3 Core/include/Arcade/abstractions/IRenderer.hpp File Reference

File declaring Graphic Interface.

```
#include "Arcade/Constants.hpp"  
#include "Arcade/abstractions/IWindow.hpp"
```

Classes

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

Namespaces

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

7.3.1 Detailed Description

File declaring Graphic Interface.

7.4 Core/include/Arcade/abstractions/IWindow.hpp File Reference

File declaring Window Interface.

```
#include <iostream>
```

Classes

- class [Arcade::IWindow](#)
Window Interface.

Namespaces

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

7.4.1 Detailed Description

File declaring Window Interface.

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

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

Namespaces

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

Enumerations

- enum class **CoreMode** { **GAME** , **LOGIN** , **MENU** , **QUIT** }
- enum class **GameMode** { **GAME** , **GAME_OVER** , **LOGIN** , **MENU** , **PAUSE** }
- enum class **Direction** { **UP** , **DOWN** , **LEFT** , **RIGHT** , **NONE** }
- enum class [Arcade::KeyboardEvents](#) { **DOWN** , **ENTER** , **ESC** , **F1** , **F2** , **LEFT** , **RIGHT** , **TEXTENTERED** , **UP** , **NONE** }

Enum for the KeyboardEvents.

7.5.1 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.

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

File declaring Core.

```
#include <memory>
#include <vector>
#include "Arcade/abstractions/IRenderer.hpp"
#include "Arcade/abstractions/IWindow.hpp"
#include "Arcade/abstractions/IGame.hpp"
```

Classes

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

Namespaces

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

7.6.1 Detailed Description

File declaring Core.

7.7 Core/include/Arcade/EventManager.hpp File Reference

File declaring EventManager.

```
#include <map>
#include <functional>
#include "Arcade/Core.hpp"
#include "Arcade/Constants.hpp"
```

Classes

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

Namespaces

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

7.7.1 Detailed Description

File declaring EventsManager.

7.8 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.

7.8.1 Detailed Description

File declaring RuntimeException.

7.9 Games/Nibbler/include/Arcade/Nibbler.hpp File Reference

Defines the Nibbler class.

```
#include "Arcade/abstractions/AGame.hpp"
```

Classes

- class [Arcade::Nibbler](#)

Namespaces

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

7.9.1 Detailed Description

Defines the Nibbler class.

7.10 Games/Snake/include/Arcade/Snake.hpp File Reference

Defines the Snake class.

```
#include "Arcade/abstractions/AGame.hpp"
```

Classes

- class [Arcade::Snake](#)

Namespaces

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

7.10.1 Detailed Description

Defines the Snake class.

7.11 Graphics/NCurses/include/Arcade/NCursesRenderer.hpp File Reference

File declaring NCursesRenderer.

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

Classes

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

Namespaces

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

7.11.1 Detailed Description

File declaring NCursesRenderer.

7.12 Graphics/NCurses/include/Arcade/NCursesWindow.hpp File Reference

File declaring NCursesWindow.

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

Classes

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

Namespaces

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

7.12.1 Detailed Description

File declaring NCursesWindow.

7.13 Graphics/SDL2/include/Arcade/Sdl2Renderer.hpp File Reference

File declaring Sdl2Renderer class.

```
#include <map>
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_mixer.h>
#include "Arcade/abstractions/IRenderer.hpp"
#include "Arcade/Sdl2Window.hpp"
```

Classes

- class [Arcade::Sdl2Renderer](#)
Sdl2Renderer class.

Namespaces

- [Arcade](#)

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

7.13.1 Detailed Description

File declaring Sdl2Renderer class.

7.14 Graphics/SDL2/include/Arcade/Sdl2Window.hpp File Reference

File declaring Sdl2Window class.

```
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_image.h>
#include "Arcade/abstractions/IWindow.hpp"
```

Classes

- class [Arcade::Sdl2Window](#)
Sdl2Window class.

Namespaces

- [Arcade](#)

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

7.14.1 Detailed Description

File declaring Sdl2Window class.

7.15 Graphics/SFML/include/Arcade/SfmlRenderer.hpp File Reference

File declaring SfmlRenderer class.

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

Classes

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

Namespaces

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

7.15.1 Detailed Description

File declaring SfmlRenderer class.

7.16 Graphics/SFML/include/Arcade/SfmlWindow.hpp File Reference

File declaring SfmlWindow class.

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

Classes

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

Namespaces

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

7.16.1 Detailed Description

File declaring SfmlWindow class.

Index

- Arcade, 9
 - KeyboardEvents, 10
- Arcade::AGame, 11
 - getDirection, 12
 - getGameMode, 12
 - getScore, 12
 - getUserName, 13
 - randomize, 13
 - setDirection, 13
- Arcade::Clock, 14
 - getElapsedTime, 14
- Arcade::Core, 15
 - getNextLibIndex, 16
 - getPreviousLibIndex, 17
 - getRenderer, 17
 - loadRenderer, 17
 - mainLoop, 18
 - runArcade, 18
 - setMode, 18
 - switchLib, 18
- Arcade::Core::CoreException, 19
- Arcade::EventManager, 20
 - handleEvent, 20
- Arcade::IGame, 20
 - getDirection, 21
 - getGameMode, 22
 - getScore, 22
 - getUserName, 22
 - randomize, 23
 - setDirection, 23
- Arcade::IRenderer, 24
 - createSprite, 25
 - createText, 25
 - displaySprite, 26
 - displayText, 26
 - getEvent, 26
 - getWindow, 26
 - isASCII, 27
 - loadFont, 27
 - loadSound, 27
 - loadTexture, 28
 - moveSprite, 28
 - moveText, 29
 - rotateSprite, 29
 - setSize, 29
- Arcade::IWindow, 30
 - openWindow, 30
 - setTitle, 31
- Arcade::NCursesRenderer, 31
 - createSprite, 32
 - createText, 33
 - displaySprite, 33
 - displayText, 33
 - getEvent, 34
 - getWindow, 34
 - isASCII, 34
 - loadFont, 34
 - loadSound, 36
 - loadTexture, 36
 - moveSprite, 36
 - moveText, 37
 - rotateSprite, 37
 - setSize, 38
- Arcade::NCursesWindow, 38
 - getWindow, 39
 - openWindow, 39
 - setTitle, 39
- Arcade::Nibbler, 40
 - setDirection, 41
- Arcade::RuntimeException, 41
- Arcade::Sdl2Renderer, 42
 - createSprite, 43
 - createText, 43
 - displaySprite, 44
 - displayText, 44
 - getEvent, 44
 - getWindow, 44
 - isASCII, 45
 - loadFont, 45
 - loadSound, 45
 - loadTexture, 46
 - moveSprite, 46
 - moveText, 47
 - rotateSprite, 47
 - setSize, 47
- Arcade::Sdl2Window, 48
 - getRenderer, 49
 - getWindow, 49
 - openWindow, 49
 - setTitle, 49
- Arcade::SfmlRenderer, 50
 - createSprite, 51
 - createText, 52
 - displaySprite, 52
 - displayText, 52
 - getEvent, 53
 - getWindow, 53
 - isASCII, 53

- loadFont, [53](#)
- loadSound, [54](#)
- loadTexture, [54](#)
- moveSprite, [54](#)
- moveText, [55](#)
- rotateSprite, [55](#)
- setInput, [56](#)
- setSize, [56](#)
- Arcade::SfmlWindow, [56](#)
 - getWindow, [57](#)
 - openWindow, [57](#)
 - setTitle, [58](#)
- Arcade::Snake, [58](#)
- Arcade::Time, [59](#)
- Core/include/Arcade/abstractions/AGame.hpp, [61](#)
- Core/include/Arcade/abstractions/IGame.hpp, [61](#)
- Core/include/Arcade/abstractions/IRenderer.hpp, [62](#)
- Core/include/Arcade/abstractions/IWindow.hpp, [62](#)
- Core/include/Arcade/Constants.hpp, [63](#)
- Core/include/Arcade/Core.hpp, [64](#)
- Core/include/Arcade/EventManager.hpp, [64](#)
- Core/include/Arcade/RuntimeException.hpp, [65](#)
- createSprite
 - Arcade::IRenderer, [25](#)
 - Arcade::NCursesRenderer, [32](#)
 - Arcade::Sdl2Renderer, [43](#)
 - Arcade::SfmlRenderer, [51](#)
- createText
 - Arcade::IRenderer, [25](#)
 - Arcade::NCursesRenderer, [33](#)
 - Arcade::Sdl2Renderer, [43](#)
 - Arcade::SfmlRenderer, [52](#)
- displaySprite
 - Arcade::IRenderer, [26](#)
 - Arcade::NCursesRenderer, [33](#)
 - Arcade::Sdl2Renderer, [44](#)
 - Arcade::SfmlRenderer, [52](#)
- displayText
 - Arcade::IRenderer, [26](#)
 - Arcade::NCursesRenderer, [33](#)
 - Arcade::Sdl2Renderer, [44](#)
 - Arcade::SfmlRenderer, [52](#)
- Games/Nibbler/include/Arcade/Nibbler.hpp, [65](#)
- Games/Snake/include/Arcade/Snake.hpp, [66](#)
- getDirection
 - Arcade::AGame, [12](#)
 - Arcade::IGame, [21](#)
- getElapsedTime
 - Arcade::Clock, [14](#)
- getEvent
 - Arcade::IRenderer, [26](#)
 - Arcade::NCursesRenderer, [34](#)
 - Arcade::Sdl2Renderer, [44](#)
 - Arcade::SfmlRenderer, [53](#)
- getGameMode
 - Arcade::AGame, [12](#)
- Arcade::IGame, [22](#)
- getNextLibIndex
 - Arcade::Core, [16](#)
- getPreviousLibIndex
 - Arcade::Core, [17](#)
- getRenderer
 - Arcade::Core, [17](#)
 - Arcade::Sdl2Window, [49](#)
- getScore
 - Arcade::AGame, [12](#)
 - Arcade::IGame, [22](#)
- getUserName
 - Arcade::AGame, [13](#)
 - Arcade::IGame, [22](#)
- getWindow
 - Arcade::IRenderer, [26](#)
 - Arcade::NCursesRenderer, [34](#)
 - Arcade::NCursesWindow, [39](#)
 - Arcade::Sdl2Renderer, [44](#)
 - Arcade::Sdl2Window, [49](#)
 - Arcade::SfmlRenderer, [53](#)
 - Arcade::SfmlWindow, [57](#)
- Graphics/NCurses/include/Arcade/NCursesRenderer.hpp, [66](#)
- Graphics/NCurses/include/Arcade/NCursesWindow.hpp, [67](#)
- Graphics/SDL2/include/Arcade/Sdl2Renderer.hpp, [67](#)
- Graphics/SDL2/include/Arcade/Sdl2Window.hpp, [68](#)
- Graphics/SFML/include/Arcade/SfmlRenderer.hpp, [68](#)
- Graphics/SFML/include/Arcade/SfmlWindow.hpp, [69](#)
- handleEvent
 - Arcade::EventManager, [20](#)
- isASCII
 - Arcade::IRenderer, [27](#)
 - Arcade::NCursesRenderer, [34](#)
 - Arcade::Sdl2Renderer, [45](#)
 - Arcade::SfmlRenderer, [53](#)
- KeyboardEvents
 - Arcade, [10](#)
- loadFont
 - Arcade::IRenderer, [27](#)
 - Arcade::NCursesRenderer, [34](#)
 - Arcade::Sdl2Renderer, [45](#)
 - Arcade::SfmlRenderer, [53](#)
- loadRenderer
 - Arcade::Core, [17](#)
- loadSound
 - Arcade::IRenderer, [27](#)
 - Arcade::NCursesRenderer, [36](#)
 - Arcade::Sdl2Renderer, [45](#)
 - Arcade::SfmlRenderer, [54](#)
- loadTexture
 - Arcade::IRenderer, [28](#)
 - Arcade::NCursesRenderer, [36](#)
 - Arcade::Sdl2Renderer, [46](#)

- Arcade::SfmlRenderer, [54](#)
- mainLoop
 - Arcade::Core, [18](#)
- moveSprite
 - Arcade::IRenderer, [28](#)
 - Arcade::NCursesRenderer, [36](#)
 - Arcade::Sdl2Renderer, [46](#)
 - Arcade::SfmlRenderer, [54](#)
- moveText
 - Arcade::IRenderer, [29](#)
 - Arcade::NCursesRenderer, [37](#)
 - Arcade::Sdl2Renderer, [47](#)
 - Arcade::SfmlRenderer, [55](#)
- openWindow
 - Arcade::IWindow, [30](#)
 - Arcade::NCursesWindow, [39](#)
 - Arcade::Sdl2Window, [49](#)
 - Arcade::SfmlWindow, [57](#)
- randomize
 - Arcade::AGame, [13](#)
 - Arcade::IGame, [23](#)
- rotateSprite
 - Arcade::IRenderer, [29](#)
 - Arcade::NCursesRenderer, [37](#)
 - Arcade::Sdl2Renderer, [47](#)
 - Arcade::SfmlRenderer, [55](#)
- runArcade
 - Arcade::Core, [18](#)
- setDirection
 - Arcade::AGame, [13](#)
 - Arcade::IGame, [23](#)
 - Arcade::Nibbler, [41](#)
- setInput
 - Arcade::SfmlRenderer, [56](#)
- setMode
 - Arcade::Core, [18](#)
- setSize
 - Arcade::IRenderer, [29](#)
 - Arcade::NCursesRenderer, [38](#)
 - Arcade::Sdl2Renderer, [47](#)
 - Arcade::SfmlRenderer, [56](#)
- setTitle
 - Arcade::IWindow, [31](#)
 - Arcade::NCursesWindow, [39](#)
 - Arcade::Sdl2Window, [49](#)
 - Arcade::SfmlWindow, [58](#)
- switchLib
 - Arcade::Core, [18](#)