

Arcade

Generated by Doxygen 1.9.7

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
6 Class Documentation	11
6.1 arcade::ADisplayModule Class Reference	11
6.1.1 Constructor & Destructor Documentation	12
6.1.1.1 ADisplayModule()	12
6.1.1.2 ~ADisplayModule()	12
6.1.2 Member Function Documentation	12
6.1.2.1 clearWindow()	12
6.1.2.2 displayWindow()	12
6.1.2.3 drawAllSprite()	12
6.1.2.4 drawSprite()	13
6.1.2.5 drawText()	13
6.1.2.6 getCoreModule()	13
6.1.2.7 getInput()	13
6.1.2.8 getName()	13
6.1.2.9 setCoreModule()	13
6.1.3 Member Data Documentation	14
6.1.3.1 _coreModule	14
6.1.3.2 _input	14
6.2 arcade::AGameModule Class Reference	14
6.2.1 Constructor & Destructor Documentation	15
6.2.1.1 AGameModule()	15
6.2.1.2 ~AGameModule()	15
6.2.2 Member Function Documentation	15
6.2.2.1 getCoreModule()	15
6.2.2.2 getDirection()	15
6.2.2.3 getGameStatus()	16
6.2.2.4 getLayerCell()	16
6.2.2.5 getName()	16
6.2.2.6 handleKeyEvents()	16

6.2.2.7	init()	16
6.2.2.8	setCoreModule()	16
6.2.2.9	setDirection()	17
6.2.2.10	setGameStatus()	17
6.2.2.11	updateGame()	17
6.3	arcade::NCurses Class Reference	17
6.3.1	Constructor & Destructor Documentation	18
6.3.1.1	NCurses()	18
6.3.1.2	~NCurses()	19
6.3.2	Member Function Documentation	19
6.3.2.1	clearWindow()	19
6.3.2.2	displayWindow()	19
6.3.2.3	drawAllSprite()	19
6.3.2.4	drawSprite()	19
6.3.2.5	drawText()	19
6.3.2.6	getInput()	20
6.3.2.7	getName()	20
6.3.3	Member Data Documentation	20
6.3.3.1	_window	20
6.4	arcade::Pacman Class Reference	20
6.4.1	Constructor & Destructor Documentation	21
6.4.1.1	Pacman()	21
6.4.1.2	~Pacman()	21
6.4.2	Member Function Documentation	21
6.4.2.1	getName()	21
6.4.2.2	handleKeyEvents()	21
6.4.2.3	init()	22
6.4.2.4	movePacman()	22
6.4.2.5	updateGame()	22
6.5	arcade::Sdl2 Class Reference	22
6.5.1	Constructor & Destructor Documentation	24
6.5.1.1	Sdl2()	24
6.5.1.2	~Sdl2()	24
6.5.2	Member Function Documentation	24
6.5.2.1	clearWindow()	24
6.5.2.2	displayWindow()	24
6.5.2.3	drawAllSprite()	24
6.5.2.4	drawSprite()	25
6.5.2.5	drawText()	25
6.5.2.6	getInput()	25
6.5.2.7	getName()	26
6.5.3	Member Data Documentation	26

6.5.3.1 _renderer	26
6.5.3.2 _window	26
6.6 arcade::Sfml Class Reference	26
6.6.1 Constructor & Destructor Documentation	27
6.6.1.1 Sfml()	27
6.6.1.2 ~Sfml()	28
6.6.2 Member Function Documentation	28
6.6.2.1 clearWindow()	28
6.6.2.2 displayWindow()	28
6.6.2.3 drawAllSprite()	28
6.6.2.4 drawSprite()	28
6.6.2.5 drawText()	28
6.6.2.6 getInput()	29
6.6.2.7 getName()	29
6.6.3 Member Data Documentation	29
6.6.3.1 _texture	29
6.6.3.2 _window	29
6.7 arcade::Snake Class Reference	29
6.7.1 Constructor & Destructor Documentation	30
6.7.1.1 Snake()	30
6.7.1.2 ~Snake()	31
6.7.2 Member Function Documentation	31
6.7.2.1 getName()	31
6.7.2.2 handleKeyEvents()	31
6.7.2.3 init()	31
6.7.2.4 moveSnake()	31
6.7.2.5 updateGame()	32
7 File Documentation	33
7.1 /home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.cpp File Reference	33
7.2 /home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.hpp File Reference	33
7.3 AGameModule.hpp	33
7.4 /home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/Pacman.cpp File Reference	34
7.4.1 Function Documentation	34
7.4.1.1 entryPoint()	34
7.4.1.2 getName()	34
7.4.1.3 getType()	34
7.5 /home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/Pacman.hpp File Reference	35
7.5.1 Macro Definition Documentation	35
7.5.1.1 SPEED_SNAKE	35
7.6 Pacman.hpp	35
7.7 /home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.cpp File Reference	36

7.7.1 Function Documentation	36
7.7.1.1 <code>entryPoint()</code>	36
7.7.1.2 <code>getName()</code>	36
7.7.1.3 <code>getType()</code>	36
7.8 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.hpp</code> File Reference	36
7.8.1 Macro Definition Documentation	37
7.8.1.1 <code>MAX_SNAKE_SIZE</code>	37
7.8.1.2 <code>SPEED_SNAKE</code>	37
7.9 <code>Snake.hpp</code>	37
7.10 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplayModule.cpp</code> File Reference	37
7.11 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplayModule.hpp</code> File Reference	38
7.12 <code>ADisplayModule.hpp</code>	38
7.13 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/NCurses.cpp</code> File Reference	38
7.13.1 Function Documentation	39
7.13.1.1 <code>entryPoint()</code>	39
7.13.1.2 <code>getName()</code>	39
7.13.1.3 <code>getType()</code>	39
7.14 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/NCurses.hpp</code> File Reference	39
7.15 <code>NCurses.hpp</code>	40
7.16 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.cpp</code> File Reference	40
7.16.1 Function Documentation	40
7.16.1.1 <code>entryPoint()</code>	40
7.16.1.2 <code>getName()</code>	41
7.16.1.3 <code>getType()</code>	41
7.17 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.hpp</code> File Reference	41
7.18 <code>Sdl2.hpp</code>	41
7.19 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.cpp</code> File Reference	42
7.19.1 Function Documentation	42
7.19.1.1 <code>entryPoint()</code>	42
7.19.1.2 <code>getName()</code>	42
7.19.1.3 <code>getType()</code>	42
7.20 <code>/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.hpp</code> File Reference	42
7.21 <code>Sfml.hpp</code>	43
7.22 <code>/home/arthur/tek2/OOP/projects/Arcade/src/CoreModule.cpp</code> File Reference	43
7.23 <code>/home/arthur/tek2/OOP/projects/Arcade/src/ErrorHandler.cpp</code> File Reference	43
7.23.1 Function Documentation	44
7.23.1.1 <code>checkPath()</code>	44
7.24 <code>/home/arthur/tek2/OOP/projects/Arcade/src/Main.cpp</code> File Reference	44
7.24.1 Function Documentation	44
7.24.1.1 <code>main()</code>	44

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

arcade	9
------------------------	-------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

arcade::IDisplayModule	
arcade::ADisplayModule	11
arcade::NCurses	17
arcade::Sdl2	22
arcade::Sfml	26
arcade::IGameModule	
arcade::AGameModule	14
arcade::Pacman	20
arcade::Snake	29

Chapter 3

Class Index

3.1 Class List

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

arcade::ADisplayModule	11
arcade::AGameModule	14
arcade::NCurses	17
arcade::Pacman	20
arcade::Sdl2	22
arcade::Sfml	26
arcade::Snake	29

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/arthur/tek2/OOP/projects/Arcade/lib/games/ AGameModule.cpp	33
/home/arthur/tek2/OOP/projects/Arcade/lib/games/ AGameModule.hpp	33
/home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/ Pacman.cpp	34
/home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/ Pacman.hpp	35
/home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/ Snake.cpp	36
/home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/ Snake.hpp	36
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ ADisplayModule.cpp	37
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ ADisplayModule.hpp	38
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/ NCurses.cpp	38
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/ NCurses.hpp	39
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/ Sdl2.cpp	40
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/ Sdl2.hpp	41
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/ Sfml.cpp	42
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/ Sfml.hpp	42
/home/arthur/tek2/OOP/projects/Arcade/src/ CoreModule.cpp	43
/home/arthur/tek2/OOP/projects/Arcade/src/ ErrorHandling.cpp	43
/home/arthur/tek2/OOP/projects/Arcade/src/ Main.cpp	44

Chapter 5

Namespace Documentation

5.1 arcade Namespace Reference

Classes

- class [ADisplayModule](#)
- class [AGameModule](#)
- class [NCurses](#)
- class [Pacman](#)
- class [Sdl2](#)
- class [Sfml](#)
- class [Snake](#)

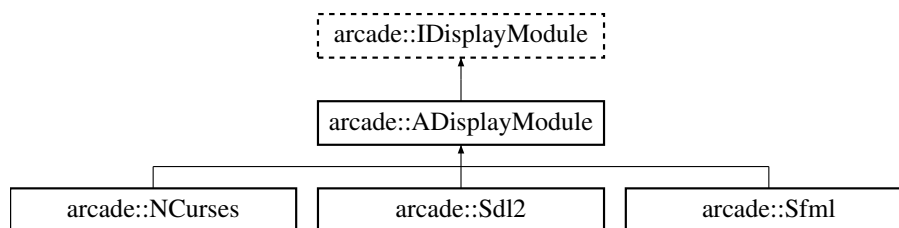
Chapter 6

Class Documentation

6.1 arcade::ADisplayModule Class Reference

```
#include <ADisplayModule.hpp>
```

Inheritance diagram for arcade::ADisplayModule:



Public Member Functions

- [ADisplayModule](#) ()
Construct a new `arcade::ADisplayModule::ADisplayModule` object.
- [~ADisplayModule](#) ()
Destroy the `arcade::ADisplayModule::ADisplayModule` object.
- virtual std::string [getName](#) () const =0
- void [setCoreModule](#) (arcade::CoreModule *coreModule)
set the core module to the display module
- arcade::CoreModule * [getCoreModule](#) () const
get the core module
- virtual void [clearWindow](#) ()=0
- virtual void [displayWindow](#) ()=0
- virtual arcade::KeyboardInput [getInput](#) ()=0
- virtual void [drawSprite](#) (std::pair< char, std::string > sprite, int x, int y, int width, int height)=0
- virtual void [drawAllSprite](#) (std::pair< char, std::string > sprite, std::vector< std::pair< int, int > > coordinates, int width, int height)=0
- virtual void [drawText](#) (const std::string text, int x, int y, int size)=0

Protected Attributes

- `arcade::KeyboardInput` [_input](#)
- `arcade::CoreModule` * [_coreModule](#)

6.1.1 Constructor & Destructor Documentation

6.1.1.1 `ADisplayModule()`

```
arcade::ADisplayModule::ADisplayModule ( )
```

Construct a new [arcade::ADisplayModule::ADisplayModule](#) object.

6.1.1.2 `~ADisplayModule()`

```
arcade::ADisplayModule::~~ADisplayModule ( )
```

Destroy the [arcade::ADisplayModule::ADisplayModule](#) object.

6.1.2 Member Function Documentation

6.1.2.1 `clearWindow()`

```
virtual void arcade::ADisplayModule::clearWindow ( ) [pure virtual]
```

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

6.1.2.2 `displayWindow()`

```
virtual void arcade::ADisplayModule::displayWindow ( ) [pure virtual]
```

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

6.1.2.3 `drawAllSprite()`

```
virtual void arcade::ADisplayModule::drawAllSprite (
    std::pair< char, std::string > sprite,
    std::vector< std::pair< int, int > > coordinates,
    int width,
    int height ) [pure virtual]
```

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

6.1.2.4 drawSprite()

```
virtual void arcade::ADisplayModule::drawSprite (
    std::pair< char, std::string > sprite,
    int x,
    int y,
    int width,
    int height ) [pure virtual]
```

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

6.1.2.5 drawText()

```
virtual void arcade::ADisplayModule::drawText (
    const std::string text,
    int x,
    int y,
    int size ) [pure virtual]
```

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

6.1.2.6 getCoreModule()

```
arcade::CoreModule * arcade::ADisplayModule::getCoreModule ( ) const
```

get the core module

Returns

arcade::CoreModule *

6.1.2.7 getInput()

```
virtual arcade::KeyboardInput arcade::ADisplayModule::getInput ( ) [pure virtual]
```

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

6.1.2.8 getName()

```
virtual std::string arcade::ADisplayModule::getName ( ) const [pure virtual]
```

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

6.1.2.9 setCoreModule()

```
void arcade::ADisplayModule::setCoreModule (
    arcade::CoreModule * coreModule )
```

set the core module to the display module

Parameters

<code>coreModule</code>	
-------------------------	--

6.1.3 Member Data Documentation

6.1.3.1 `_coreModule`

```
arcade::CoreModule* arcade::ADisplayModule::_coreModule [protected]
```

6.1.3.2 `_input`

```
arcade::KeyboardInput arcade::ADisplayModule::_input [protected]
```

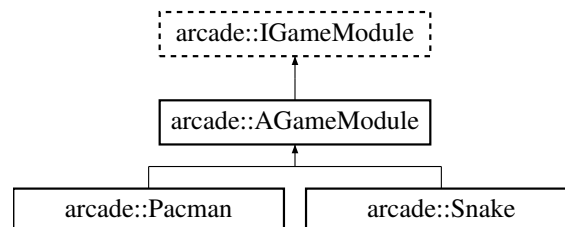
The documentation for this class was generated from the following files:

- [/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplayModule.hpp](#)
- [/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplayModule.cpp](#)

6.2 arcade::AGameModule Class Reference

```
#include <AGameModule.hpp>
```

Inheritance diagram for arcade::AGameModule:



Public Member Functions

- [AGameModule](#) ()
Construct a new arcade::A Game Module::A Game Module object.
- [~AGameModule](#) ()
Destroy the arcade::A Game Module::A Game Module object.
- virtual void [updateGame](#) ()=0
- virtual std::string [getName](#) ()=0
- void [setGameStatus](#) (GameStatus status)
- GameStatus [getGameStatus](#) () const
get the status of the game
- void [setCoreModule](#) (arcade::CoreModule *coreModule)
set the core module to the game module

- arcade::CoreModule * [getCoreModule](#) () const
get the core module
- void [setDirection](#) (arcade::KeyboardInput direction)
set the direction of the game
- arcade::KeyboardInput [getDirection](#) () const
get the direction of the game
- virtual void [init](#) ()=0
- virtual void [handdleKeyEvents](#) (arcade::KeyboardInput key)=0
- int [getLayerCell](#) (int layer, int x, int y) const
get the cell of the actual layer

6.2.1 Constructor & Destructor Documentation

6.2.1.1 AGameModule()

```
arcade::AGameModule::AGameModule ( )
```

Construct a new arcade::A Game Module::A Game Module object.

6.2.1.2 ~AGameModule()

```
arcade::AGameModule::~~AGameModule ( )
```

Destroy the arcade::A Game Module::A Game Module object.

6.2.2 Member Function Documentation

6.2.2.1 getCoreModule()

```
arcade::CoreModule * arcade::AGameModule::getCoreModule ( ) const
```

get the core module

Returns

arcade::CoreModule *

6.2.2.2 getDirection()

```
arcade::KeyboardInput arcade::AGameModule::getDirection ( ) const
```

get the direction of the game

Returns

arcade::KeyboardInput

6.2.2.3 `getGameStatus()`

```
arcade::AGameModule::GameStatus arcade::AGameModule::getGameStatus ( ) const
```

get the status of the game

Returns

`arcade::AGameModule::GameStatus`

6.2.2.4 `getLayerCell()`

```
int arcade::AGameModule::getLayerCell (
    int layer,
    int x,
    int y ) const
```

get the cell of the actual layer

Parameters

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

Returns

`int`

6.2.2.5 `getName()`

```
virtual std::string arcade::AGameModule::getName ( ) [pure virtual]
```

Implemented in [arcade::Pacman](#), and [arcade::Snake](#).

6.2.2.6 `handdleKeyEvents()`

```
virtual void arcade::AGameModule::handdleKeyEvents (
    arcade::KeyboardInput key ) [pure virtual]
```

Implemented in [arcade::Pacman](#), and [arcade::Snake](#).

6.2.2.7 `init()`

```
virtual void arcade::AGameModule::init ( ) [pure virtual]
```

Implemented in [arcade::Pacman](#), and [arcade::Snake](#).

6.2.2.8 `setCoreModule()`

```
void arcade::AGameModule::setCoreModule (
    arcade::CoreModule * coreModule )
```

set the core module to the game module

Parameters

<i>coreModule</i>	
-------------------	--

6.2.2.9 setDirection()

```
void arcade::AGameModule::setDirection (
    arcade::KeyboardInput direction )
```

set the direction of the game

Parameters

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

6.2.2.10 setGameStatus()

```
void arcade::AGameModule::setGameStatus (
    GameState status )
```

6.2.2.11 updateGame()

```
virtual void arcade::AGameModule::updateGame ( ) [pure virtual]
```

Implemented in [arcade::Pacman](#), and [arcade::Snake](#).

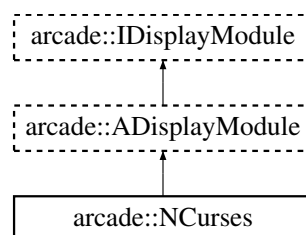
The documentation for this class was generated from the following files:

- [/home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.hpp](#)
- [/home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.cpp](#)

6.3 arcade::NCurses Class Reference

```
#include <NCurses.hpp>
```

Inheritance diagram for arcade::NCurses:



Public Member Functions

- [NCurses](#) ()
- [~NCurses](#) ()
- `std::string` [getName](#) () const
- void [clearWindow](#) ()
- `arcade::KeyboardInput` [getInput](#) ()
- void [displayWindow](#) ()
- void [drawText](#) (const `std::string` text, int x, int y, int size)
- void [drawSprite](#) (`std::pair`< char, `std::string` > sprite, int x, int y, int width, int height)
- void [drawAllSprite](#) (`std::pair`< char, `std::string` > sprite, `std::vector`< `std::pair`< int, int > > coordinates, int width, int height)

Public Member Functions inherited from [arcade::ADisplayModule](#)

- [ADisplayModule](#) ()
Construct a new `arcade::ADisplayModule::ADisplayModule` object.
- [~ADisplayModule](#) ()
Destroy the `arcade::ADisplayModule::ADisplayModule` object.
- virtual `std::string` [getName](#) () const =0
- void [setCoreModule](#) (`arcade::CoreModule` *coreModule)
set the core module to the display module
- `arcade::CoreModule` * [getCoreModule](#) () const
get the core module
- virtual void [clearWindow](#) ()=0
- virtual void [displayWindow](#) ()=0
- virtual `arcade::KeyboardInput` [getInput](#) ()=0
- virtual void [drawSprite](#) (`std::pair`< char, `std::string` > sprite, int x, int y, int width, int height)=0
- virtual void [drawAllSprite](#) (`std::pair`< char, `std::string` > sprite, `std::vector`< `std::pair`< int, int > > coordinates, int width, int height)=0
- virtual void [drawText](#) (const `std::string` text, int x, int y, int size)=0

Protected Attributes

- WINDOW * [_window](#)

Protected Attributes inherited from [arcade::ADisplayModule](#)

- `arcade::KeyboardInput` [_input](#)
- `arcade::CoreModule` * [_coreModule](#)

6.3.1 Constructor & Destructor Documentation

6.3.1.1 NCurses()

```
arcade::NCurses::NCurses ( )
```


6.3.1.2 ~NCurses()

```
arcade::NCurses::~~NCurses ( )
```

6.3.2 Member Function Documentation

6.3.2.1 clearWindow()

```
void arcade::NCurses::clearWindow ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.3.2.2 displayWindow()

```
void arcade::NCurses::displayWindow ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.3.2.3 drawAllSprite()

```
void arcade::NCurses::drawAllSprite (
    std::pair< char, std::string > sprite,
    std::vector< std::pair< int, int > > coordinates,
    int width,
    int height ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.3.2.4 drawSprite()

```
void arcade::NCurses::drawSprite (
    std::pair< char, std::string > sprite,
    int x,
    int y,
    int width,
    int height ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.3.2.5 drawText()

```
void arcade::NCurses::drawText (
    const std::string text,
    int x,
    int y,
    int size ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.3.2.6 getInput()

```
arcade::KeyboardInput arcade::NCurses::getInput ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.3.2.7 getName()

```
std::string arcade::NCurses::getName ( ) const [virtual]
```

Implements [arcade::ADisplayModule](#).

6.3.3 Member Data Documentation

6.3.3.1 _window

```
WINDOW* arcade::NCurses::_window [protected]
```

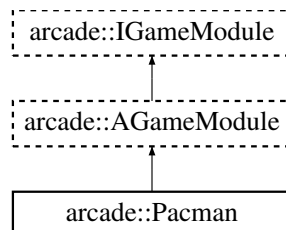
The documentation for this class was generated from the following files:

- /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/[NCurses.hpp](#)
- /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/[NCurses.cpp](#)

6.4 arcade::Pacman Class Reference

```
#include <Pacman.hpp>
```

Inheritance diagram for arcade::Pacman:



Public Member Functions

- [Pacman](#) ()
Construct a new [arcade::Pacman::Pacman](#) object.
- [~Pacman](#) ()
Destroy the [arcade::Pacman::Pacman](#) object.
- void [init](#) ()
- void [updateGame](#) ()
update the game
- std::string [getName](#) ()
get the name of the game
- arcade::GameData [movePacman](#) ()
move the [Pacman](#)
- void [handleKeyEvents](#) (arcade::KeyboardInput key)
handle key events

Public Member Functions inherited from [arcade::AGameModule](#)

- [AGameModule](#) ()
Construct a new arcade::A Game Module::A Game Module object.
- [~AGameModule](#) ()
Destroy the arcade::A Game Module::A Game Module object.
- virtual void [updateGame](#) ()=0
- virtual std::string [getName](#) ()=0
- void [setGameStatus](#) (GameStatus status)
- GameStatus [getGameStatus](#) () const
get the status of the game
- void [setCoreModule](#) (arcade::CoreModule *coreModule)
set the core module to the game module
- arcade::CoreModule * [getCoreModule](#) () const
get the core module
- void [setDirection](#) (arcade::KeyboardInput direction)
set the direction of the game
- arcade::KeyboardInput [getDirection](#) () const
get the direction of the game
- virtual void [init](#) ()=0
- virtual void [handleKeyEvents](#) (arcade::KeyboardInput key)=0
- int [getLayerCell](#) (int layer, int x, int y) const
get the cell of the actual layer

6.4.1 Constructor & Destructor Documentation

6.4.1.1 Pacman()

```
arcade::Pacman::Pacman ( )
```

Construct a new [arcade::Pacman::Pacman](#) object.

6.4.1.2 ~Pacman()

```
arcade::Pacman::~~Pacman ( )
```

Destroy the [arcade::Pacman::Pacman](#) object.

6.4.2 Member Function Documentation

6.4.2.1 getName()

```
std::string arcade::Pacman::getName ( ) [virtual]
```

get the name of the game

Returns

std::string

Implements [arcade::AGameModule](#).

6.4.2.2 handleKeyEvents()

```
void arcade::Pacman::handleKeyEvents (
    arcade::KeyboardInput key ) [virtual]
```

handle key events

Parameters

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

Implements [arcade::AGameModule](#).

6.4.2.3 init()

```
void arcade::Pacman::init ( ) [virtual]
```

Implements [arcade::AGameModule](#).

6.4.2.4 movePacman()

```
arcade::GameData arcade::Pacman::movePacman ( )
```

move the [Pacman](#)

Parameters

<i>display_info</i>	
---------------------	--

6.4.2.5 updateGame()

```
void arcade::Pacman::updateGame ( ) [virtual]
```

update the game

Implements [arcade::AGameModule](#).

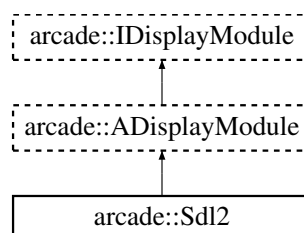
The documentation for this class was generated from the following files:

- /home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/[Pacman.hpp](#)
- /home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/[Pacman.cpp](#)

6.5 arcade::Sdl2 Class Reference

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

Inheritance diagram for arcade::Sdl2:



Public Member Functions

- [Sdl2](#) ()
- [~Sdl2](#) ()
- `std::string` [getName](#) () const
get the name of the library
- void [clearWindow](#) ()
clear the window
- void [displayWindow](#) ()
- `arcade::KeyboardInput` [getInput](#) ()
get the input from the window
- void [drawText](#) (const `std::string` text, int x, int y, int size)
draw text on the window
- void [drawSprite](#) (`std::pair`< char, `std::string` > sprite, int x, int y, int width, int height)
draw a sprite on the window
- void [drawAllSprite](#) (`std::pair`< char, `std::string` > sprite, `std::vector`< `std::pair`< int, int > > coordinates, int width, int height)
draw a sprite on the window

Public Member Functions inherited from [arcade::ADisplayModule](#)

- [ADisplayModule](#) ()
Construct a new [arcade::ADisplayModule::ADisplayModule](#) object.
- [~ADisplayModule](#) ()
Destroy the [arcade::ADisplayModule::ADisplayModule](#) object.
- virtual `std::string` [getName](#) () const =0
- void [setCoreModule](#) (`arcade::CoreModule` *coreModule)
set the core module to the display module
- `arcade::CoreModule` * [getCoreModule](#) () const
get the core module
- virtual void [clearWindow](#) ()=0
- virtual void [displayWindow](#) ()=0
- virtual `arcade::KeyboardInput` [getInput](#) ()=0
- virtual void [drawSprite](#) (`std::pair`< char, `std::string` > sprite, int x, int y, int width, int height)=0
- virtual void [drawAllSprite](#) (`std::pair`< char, `std::string` > sprite, `std::vector`< `std::pair`< int, int > > coordinates, int width, int height)=0
- virtual void [drawText](#) (const `std::string` text, int x, int y, int size)=0

Protected Attributes

- `SDL_Renderer` * [_renderer](#)
- `SDL_Window` * [_window](#)

Protected Attributes inherited from [arcade::ADisplayModule](#)

- `arcade::KeyboardInput` [_input](#)
- `arcade::CoreModule` * [_coreModule](#)

6.5.1 Constructor & Destructor Documentation

6.5.1.1 Sdl2()

```
arcade::Sdl2::Sdl2 ( )
```

6.5.1.2 ~Sdl2()

```
arcade::Sdl2::~~Sdl2 ( )
```

6.5.2 Member Function Documentation

6.5.2.1 clearWindow()

```
void arcade::Sdl2::clearWindow ( ) [virtual]
```

clear the window

Implements [arcade::ADisplayModule](#).

6.5.2.2 displayWindow()

```
void arcade::Sdl2::displayWindow ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.5.2.3 drawAllSprite()

```
void arcade::Sdl2::drawAllSprite (
    std::pair< char, std::string > sprite,
    std::vector< std::pair< int, int > > coordinates,
    int width,
    int height ) [virtual]
```

draw a sprite on the window

Parameters

<i>sprite</i>	sprite to display
<i>coordinates</i>	coordinates of the sprite
<i>width</i>	width of the sprite
<i>height</i>	height of the sprite
<i>rotation</i>	rotation of the sprite

Implements [arcade::ADisplayModule](#).

6.5.2.4 drawSprite()

```
void arcade::Sdl2::drawSprite (
    std::pair< char, std::string > sprite,
    int x,
    int y,
    int width,
    int height ) [virtual]
```

draw a sprite on the window

Parameters

<i>path</i>	path to the sprite
<i>x</i>	x position of the sprite
<i>y</i>	y position of the sprite
<i>width</i>	width of the sprite
<i>height</i>	height of the sprite

Implements [arcade::ADisplayModule](#).

6.5.2.5 drawText()

```
void arcade::Sdl2::drawText (
    const std::string text,
    int x,
    int y,
    int size ) [virtual]
```

draw text on the window

Parameters

<i>text</i>	text to display
<i>x</i>	x position of the text
<i>y</i>	y position of the text
<i>size</i>	size of the text

Implements [arcade::ADisplayModule](#).

6.5.2.6 getInput()

```
arcade::KeyboardInput arcade::Sdl2::getInput ( ) [virtual]
```

get the input from the window

Returns

arcade::KeyboardInput

Implements [arcade::ADisplayModule](#).

6.5.2.7 getName()

```
std::string arcade::Sdl2::getName ( ) const [virtual]
```

get the name of the library

Returns

std::string

Implements [arcade::ADisplayModule](#).

6.5.3 Member Data Documentation

6.5.3.1 _renderer

```
SDL_Renderer* arcade::Sdl2::_renderer [protected]
```

6.5.3.2 _window

```
SDL_Window* arcade::Sdl2::_window [protected]
```

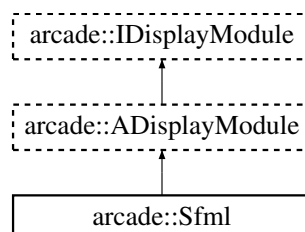
The documentation for this class was generated from the following files:

- [/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.hpp](#)
- [/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.cpp](#)

6.6 arcade::Sfml Class Reference

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

Inheritance diagram for arcade::Sfml:



Public Member Functions

- [Sfml](#) ()
- [~Sfml](#) ()
- std::string [getName](#) () const
- void [clearWindow](#) ()
- void [displayWindow](#) ()
- arcade::KeyboardInput [getInput](#) ()
- void [drawText](#) (const std::string text, int x, int y, int size)
- void [drawSprite](#) (std::pair< char, std::string > sprite, int x, int y, int width, int height)
- void [drawAllSprite](#) (std::pair< char, std::string > sprite, std::vector< std::pair< int, int > > coordinates, int width, int height)

Public Member Functions inherited from [arcade::ADisplayModule](#)

- [ADisplayModule](#) ()
Construct a new [arcade::ADisplayModule::ADisplayModule](#) object.
- [~ADisplayModule](#) ()
Destroy the [arcade::ADisplayModule::ADisplayModule](#) object.
- virtual std::string [getName](#) () const =0
- void [setCoreModule](#) (arcade::CoreModule *coreModule)
set the core module to the display module
- arcade::CoreModule * [getCoreModule](#) () const
get the core module
- virtual void [clearWindow](#) ()=0
- virtual void [displayWindow](#) ()=0
- virtual arcade::KeyboardInput [getInput](#) ()=0
- virtual void [drawSprite](#) (std::pair< char, std::string > sprite, int x, int y, int width, int height)=0
- virtual void [drawAllSprite](#) (std::pair< char, std::string > sprite, std::vector< std::pair< int, int > > coordinates, int width, int height)=0
- virtual void [drawText](#) (const std::string text, int x, int y, int size)=0

Protected Attributes

- sf::RenderWindow * [_window](#)
- sf::Texture [_texture](#)

Protected Attributes inherited from [arcade::ADisplayModule](#)

- arcade::KeyboardInput [_input](#)
- arcade::CoreModule * [_coreModule](#)

6.6.1 Constructor & Destructor Documentation

6.6.1.1 Sfml()

```
arcade::Sfml::Sfml ( )
```

6.6.1.2 ~Sfml()

```
arcade::Sfml::~~Sfml ( )
```

6.6.2 Member Function Documentation

6.6.2.1 clearWindow()

```
void arcade::Sfml::clearWindow ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.6.2.2 displayWindow()

```
void arcade::Sfml::displayWindow ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.6.2.3 drawAllSprite()

```
void arcade::Sfml::drawAllSprite (
    std::pair< char, std::string > sprite,
    std::vector< std::pair< int, int > > coordinates,
    int width,
    int height ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.6.2.4 drawSprite()

```
void arcade::Sfml::drawSprite (
    std::pair< char, std::string > sprite,
    int x,
    int y,
    int width,
    int height ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.6.2.5 drawText()

```
void arcade::Sfml::drawText (
    const std::string text,
    int x,
    int y,
    int size ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.6.2.6 getInput()

```
arcade::KeyboardInput arcade::Sfml::getInput ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

6.6.2.7 getName()

```
std::string arcade::Sfml::getName ( ) const [virtual]
```

Implements [arcade::ADisplayModule](#).

6.6.3 Member Data Documentation

6.6.3.1 _texture

```
sf::Texture arcade::Sfml::_texture [protected]
```

6.6.3.2 _window

```
sf::RenderWindow* arcade::Sfml::_window [protected]
```

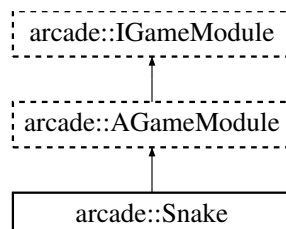
The documentation for this class was generated from the following files:

- [/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.hpp](#)
- [/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.cpp](#)

6.7 arcade::Snake Class Reference

```
#include <Snake.hpp>
```

Inheritance diagram for arcade::Snake:



Public Member Functions

- [Snake](#) ()
Construct a new `arcade::Snake::Snake` object.
- [~Snake](#) ()
Destroy the `arcade::Snake::Snake` object.
- `std::string` [getName](#) ()
get the name of the game
- `void` [init](#) ()
- `void` [updateGame](#) ()
update the game
- `arcade::GameData` [moveSnake](#) ()
move the snake
- `void` [handleKeyEvents](#) (`arcade::KeyboardInput` key)
handle key events

Public Member Functions inherited from [arcade::AGameModule](#)

- [AGameModule](#) ()
Construct a new `arcade::A Game Module::A Game Module` object.
- [~AGameModule](#) ()
Destroy the `arcade::A Game Module::A Game Module` object.
- `virtual void` [updateGame](#) ()=0
- `virtual std::string` [getName](#) ()=0
- `void` [setGameStatus](#) (`GameStatus` status)
- `GameStatus` [getGameStatus](#) () const
get the status of the game
- `void` [setCoreModule](#) (`arcade::CoreModule *coreModule`)
set the core module to the game module
- `arcade::CoreModule *` [getCoreModule](#) () const
get the core module
- `void` [setDirection](#) (`arcade::KeyboardInput` direction)
set the direction of the game
- `arcade::KeyboardInput` [getDirection](#) () const
get the direction of the game
- `virtual void` [init](#) ()=0
- `virtual void` [handleKeyEvents](#) (`arcade::KeyboardInput` key)=0
- `int` [getLayerCell](#) (`int` layer, `int` x, `int` y) const
get the cell of the actual layer

6.7.1 Constructor & Destructor Documentation

6.7.1.1 Snake()

```
arcade::Snake::Snake ( )
```

Construct a new [arcade::Snake::Snake](#) object.

6.7.1.2 ~Snake()

```
arcade::Snake::~~Snake ( )
```

Destroy the [arcade::Snake::Snake](#) object.

6.7.2 Member Function Documentation

6.7.2.1 getName()

```
std::string arcade::Snake::getName ( ) [virtual]
```

get the name of the game

Returns

std::string

Implements [arcade::AGameModule](#).

6.7.2.2 handdleKeyEvents()

```
void arcade::Snake::handdleKeyEvents (
    arcade::KeyboardInput key ) [virtual]
```

handle key events

Parameters

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

Implements [arcade::AGameModule](#).

6.7.2.3 init()

```
void arcade::Snake::init ( ) [virtual]
```

Implements [arcade::AGameModule](#).

6.7.2.4 moveSnake()

```
arcade::GameData arcade::Snake::moveSnake ( )
```

move the snake

Parameters

<i>display_info</i>	
---------------------	--

6.7.2.5 updateGame()

```
void arcade::Snake::updateGame ( ) [virtual]
```

update the game

Implements [arcade::AGameModule](#).

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

- [/home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.hpp](#)
- [/home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.cpp](#)

Chapter 7

File Documentation

7.1 /home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.cpp File Reference

```
#include "AGameModule.hpp"
```

7.2 /home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.hpp File Reference

```
#include <arcade/IGameModule.hpp>
```

Classes

- class [arcade::AGameModule](#)

Namespaces

- namespace [arcade](#)

7.3 AGameModule.hpp

[Go to the documentation of this file.](#)

```
00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Arcade
00004 ** File description:
00005 ** AGameModule
00006 */
00007
00008 #ifndef AGAMEMODULE_HPP_
00009 #define AGAMEMODULE_HPP_
00010
00011 #include <arcade/IGameModule.hpp>
00012
```

```

00013 namespace arcade
00014 {
00015     class AGameModule : virtual public arcade::IGameModule
00016     {
00017     public:
00018         AGameModule();
00019         ~AGameModule();
00020
00021         virtual void updateGame() = 0;
00022
00023         virtual std::string getName() = 0;
00024
00025         void setGameStatus(GameStatus status);
00026         GameStatus getGameStatus() const;
00027
00028         void setCoreModule(arcade::CoreModule *coreModule);
00029         arcade::CoreModule *getCoreModule() const;
00030
00031         void setDirection(arcade::KeyboardInput direction);
00032         arcade::KeyboardInput getDirection() const;
00033
00034         virtual void init() = 0;
00035         virtual void handleKeyEvents(arcade::KeyboardInput key) = 0;
00036
00037         int getLayerCell(int layer, int x, int y) const;
00038     };
00039 }; // namespace arcade
00040
00041
00042 #endif /* !IGAMEMODULE_HPP_ */

```

7.4 /home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/↵ Pacman.cpp File Reference

```
#include "Pacman.hpp"
```

Functions

- `std::unique_ptr< arcade::IGameModule > entryPoint ()`
generate entry point for the game library
- `arcade::ModuleType getType ()`
- `std::string getName ()`

7.4.1 Function Documentation

7.4.1.1 entryPoint()

```
std::unique_ptr< arcade::IGameModule > entryPoint ( )
```

generate entry point for the game library

7.4.1.2 getName()

```
std::string getName ( )
```

7.4.1.3 getType()

```
arcade::ModuleType getType ( )
```


7.5 /home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/Pacman.hpp File Reference

```
#include "../AGameModule.hpp"
```

Classes

- class [arcade::Pacman](#)

Namespaces

- namespace [arcade](#)

Macros

- #define [SPEED_SNAKE](#) 1

7.5.1 Macro Definition Documentation

7.5.1.1 SPEED_SNAKE

```
#define SPEED_SNAKE 1
```

7.6 Pacman.hpp

[Go to the documentation of this file.](#)

```
00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Arcade
00004 ** File description:
00005 ** Pacman
00006 */
00007
00008 #ifndef PACMAN_HPP_
00009 #define PACMAN_HPP_
00010 #define SPEED_SNAKE 1
00011
00012 #include "../AGameModule.hpp"
00013
00014 namespace arcade {
00015     class Pacman : virtual public arcade::AGameModule {
00016     public:
00017         Pacman();
00018         ~Pacman();
00019         void init();
00020         void updateGame();
00021         std::string getName();
00022         arcade::GameData movePacman();
00023         void handleKeyEvents(arcade::KeyboardInput key);
00024
00025     protected:
00026     };
00027 }; // namespace arcade
00028
00029 #endif /* !PACMAN_HPP_ */
```

7.7 /home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.cpp

File Reference

```
#include "Snake.hpp"
```

Functions

- `std::unique_ptr< arcade::IGameModule > entryPoint ()`
generate entry point for the game library
- `arcade::ModuleType getType ()`
- `std::string getName ()`

7.7.1 Function Documentation

7.7.1.1 `entryPoint()`

```
std::unique_ptr< arcade::IGameModule > entryPoint ( )
```

generate entry point for the game library

7.7.1.2 `getName()`

```
std::string getName ( )
```

7.7.1.3 `getType()`

```
arcade::ModuleType getType ( )
```

7.8 /home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.hpp

File Reference

```
#include "../AGameModule.hpp"
```

Classes

- class [arcade::Snake](#)

Namespaces

- namespace [arcade](#)

Macros

- `#define SPEED_SNAKE 1`
- `#define MAX_SNAKE_SIZE 361`

7.8.1 Macro Definition Documentation**7.8.1.1 MAX_SNAKE_SIZE**

```
#define MAX_SNAKE_SIZE 361
```

7.8.1.2 SPEED_SNAKE

```
#define SPEED_SNAKE 1
```

7.9 Snake.hpp

[Go to the documentation of this file.](#)

```
00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Arcade
00004  ** File description:
00005  ** Snake
00006  */
00007
00008 #ifndef SNAKE_HPP_
00009 #define SNAKE_HPP_
00010 #define SPEED_SNAKE 1
00011 #define MAX_SNAKE_SIZE 361
00012
00013 #include "../AGameModule.hpp"
00014
00015 namespace arcade {
00016 class Snake : virtual public arcade::AGameModule {
00017 public:
00018     Snake();
00019     ~Snake();
00020     std::string getName();
00021     void init();
00022     void updateGame();
00023     arcade::GameData moveSnake();
00024     void handleKeyEvents(arcade::KeyboardInput key);
00025
00026 protected:
00027 };
00028 }; // namespace arcade
00029
00030 #endif /* !SNAKE_HPP_ */
```

**7.10 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplay↵
Module.cpp File Reference**

```
#include "ADisplayModule.hpp"
```

7.11 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplayModule.hpp File Reference

```
#include <arcade/IDisplayModule.hpp>
```

Classes

- class [arcade::ADisplayModule](#)

Namespaces

- namespace [arcade](#)

7.12 ADisplayModule.hpp

[Go to the documentation of this file.](#)

```
00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** bsArcade
00004 ** File description:
00005 ** ADisplayModule
00006 */
00007
00008
00009 #ifndef ADISPLAYMODULE_HPP_
00010 #define ADISPLAYMODULE_HPP_
00011
00012 #include <arcade/IDisplayModule.hpp>
00013
00014 namespace arcade {
00015     class CoreModule;
00016     class ADisplayModule : virtual public arcade::IDisplayModule {
00017     public:
00018         ADisplayModule();
00019         ~ADisplayModule();
00020
00021         virtual std::string getName() const = 0;
00022
00023         void setCoreModule(arcade::CoreModule *coreModule);
00024         arcade::CoreModule *getCoreModule() const;
00025
00026         virtual void clearWindow() = 0;
00027         virtual void displayWindow() = 0;
00028         virtual arcade::KeyboardInput getInput() = 0;
00029         virtual void drawSprite(std::pair<char, std::string> sprite, int x, int y, int width, int height) =
00030 0;
00031         virtual void drawAllSprite(std::pair<char, std::string> sprite, std::vector<std::pair<int, int>
00032 coordinates, int width, int height) = 0;
00033         virtual void drawText(const std::string text, int x, int y, int size) = 0;
00034
00035     protected:
00036         arcade::KeyboardInput _input;
00037         arcade::CoreModule *_coreModule;
00038     };
00039 }; // namespace arcade
00040
00041 #endif /* !ADISPLAYMODULE_HPP_ */
```

7.13 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/NCurses.cpp File Reference

```
#include "NCurses.hpp"
```

Functions

- `std::unique_ptr< arcade::IDisplayModule > entryPoint ()`
entry point
- `arcade::ModuleType getType ()`
- `std::string getName ()`

7.13.1 Function Documentation

7.13.1.1 `entryPoint()`

```
std::unique_ptr< arcade::IDisplayModule > entryPoint ( )
```

entry point

7.13.1.2 `getName()`

```
std::string getName ( )
```

7.13.1.3 `getType()`

```
arcade::ModuleType getType ( )
```

7.14 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/↵ NCurses.hpp File Reference

```
#include "../ADisplayModule.hpp"  
#include <ncurses.h>
```

Classes

- class [arcade::NCurses](#)

Namespaces

- namespace [arcade](#)

7.15 NCurses.hpp

[Go to the documentation of this file.](#)

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Arcade
00004 ** File description:
00005 ** NCurses
00006 */
00007
00008 #ifndef NCURSES_HPP_
00009 #define NCURSES_HPP_
00010
00011 #include "../ADisplayModule.hpp"
00012 #include <ncurses.h>
00013
00014 namespace arcade {
00015 class NCurses : virtual public arcade::ADisplayModule {
00016 public:
00017     NCurses();
00018     ~NCurses();
00019     std::string getName() const;
00020     void clearWindow();
00021     arcade::KeyboardInput getInput();
00022     void displayWindow();
00023     void drawText(const std::string text, int x, int y, int size);
00024     void drawSprite(std::pair<char, std::string> sprite, int x, int y, int width, int height);
00025     void drawAllSprite(std::pair<char, std::string> sprite, std::vector<std::pair<int, int>> coordinates,
00026                        int width, int height);
00027 protected:
00028     WINDOW *_window;
00029 };
00030 }; // namespace arcade
00031
00032 #endif /* !NCURSES_HPP_ */

```

7.16 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.cpp

File Reference

```

#include "ErrorHandling.hpp"
#include "Sdl2.hpp"
#include <iostream>

```

Functions

- `std::unique_ptr< arcade::IDisplayModule > entryPoint ()`
entry point for the library
- `arcade::ModuleType getType ()`
- `std::string getName ()`

7.16.1 Function Documentation

7.16.1.1 `entryPoint()`

```
std::unique_ptr< arcade::IDisplayModule > entryPoint ( )
```

entry point for the library

Returns

`arcade::Sdl2 *`

7.16.1.2 getName()

```
std::string getName ( )
```

7.16.1.3 getType()

```
arcade::ModuleType getType ( )
```

7.17 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.hpp File Reference

```
#include "../ADisplayModule.hpp"
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_image.h>
```

Classes

- class [arcade::Sdl2](#)

Namespaces

- namespace [arcade](#)

7.18 Sdl2.hpp

[Go to the documentation of this file.](#)

```
00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Arcade
00004 ** File description:
00005 ** SDL2
00006 */
00007
00008 #ifndef SDL2_HPP_
00009 #define SDL2_HPP_
00010
00011 #include "../ADisplayModule.hpp"
00012 #include <SDL2/SDL.h>
00013 #include <SDL2/SDL_ttf.h>
00014 #include <SDL2/SDL_image.h>
00015
00016 namespace arcade {
00017 class Sdl2: virtual public arcade::ADisplayModule {
00018 public:
00019     Sdl2();
00020     ~Sdl2();
00021     std::string getName() const;
00022     void clearWindow();
00023     void displayWindow();
00024     arcade::KeyboardInput getInput();
00025     void drawText(const std::string text, int x, int y, int size);
00026     void drawSprite(std::pair<char, std::string> sprite, int x, int y, int width, int height);
00027     void drawAllSprite(std::pair<char, std::string> sprite, std::vector<std::pair<int, int>> coordinates,
00028 int width, int height);
00029 protected:
00030     SDL_Renderer *_renderer;
00031     SDL_Window *_window;
00032 private:
00033 };
00034 }; // namespace arcade
00035
00036 #endif /* !SDL2_HPP_ */
```

7.19 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.cpp

File Reference

```
#include <ErrorHandling.hpp>
#include "Sfml.hpp"
```

Functions

- `std::unique_ptr< arcade::IDisplayModule > entryPoint ()`
entry point for the library
- `arcade::ModuleType getType ()`
- `std::string getName ()`

7.19.1 Function Documentation

7.19.1.1 `entryPoint()`

```
std::unique_ptr< arcade::IDisplayModule > entryPoint ( )
```

entry point for the library

Returns

[arcade::Sfml](#)*

7.19.1.2 `getName()`

```
std::string getName ( )
```

7.19.1.3 `getType()`

```
arcade::ModuleType getType ( )
```

7.20 /home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.hpp

File Reference

```
#include "../ADisplayModule.hpp"
#include <SFML/Graphics.hpp>
```

Classes

- class [arcade::Sfml](#)

Namespaces

- namespace [arcade](#)

7.21 Sfml.hpp

[Go to the documentation of this file.](#)

```
00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Arcade
00004 ** File description:
00005 ** SFML
00006 */
00007
00008 #ifndef SFML_HPP_
00009 #define SFML_HPP_
00010
00011 #include "../ADisplayModule.hpp"
00012 #include <SFML/Graphics.hpp>
00013
00014 namespace arcade {
00015     class Sfml : virtual public arcade::ADisplayModule {
00016     public:
00017         Sfml();
00018         ~Sfml();
00019
00020         std::string getName() const;
00021
00022         void clearWindow();
00023         void displayWindow();
00024         arcade::KeyboardInput getInput();
00025         void drawText(const std::string text, int x, int y, int size);
00026         void drawSprite(std::pair<char, std::string> sprite, int x, int y, int width, int height);
00027         void drawAllSprite(std::pair<char, std::string> sprite, std::vector<std::pair<int, int> coordinates,
00028                             int width, int height);
00029     protected:
00030         sf::RenderWindow *_window;
00031         sf::Texture _texture;
00032     };
00033 }; // namespace arcade
00034
00035 #endif /* !SFML_HPP_ */
```

7.22 /home/arthur/tek2/OOP/projects/Arcade/src/CoreModule.cpp File Reference

```
#include <CoreModule.hpp>
```

7.23 /home/arthur/tek2/OOP/projects/Arcade/src/ErrorHandler.cpp File Reference

```
#include <dlfcn.h>
#include <iostream>
#include <ErrorHandler.hpp>
```

Functions

- void [checkPath](#) (const char *path)

7.23.1 Function Documentation

7.23.1.1 checkPath()

```
void checkPath (
    const char * path )
```

7.24 /home/arthur/tek2/OOP/projects/Arcade/src/Main.cpp File Reference

```
#include <iostream>
#include <arcade/IShell.hpp>
#include <Shell.hpp>
#include <ErrorHandling.hpp>
```

Functions

- int [main](#) (int const argc, char const *const *argv)
check param launch arcade

7.24.1 Function Documentation

7.24.1.1 main()

```
int main (
    int const argc,
    char const *const * argv )
```

check param launch arcade

Parameters

<i>argc</i>	
<i>argv</i>	

Returns

int

Index

/home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.cpp, arcade::AGameModule, 15
33
~NCurses
/home/arthur/tek2/OOP/projects/Arcade/lib/games/AGameModule.h, arcade::AGameModule, 18
33
~Pacman
/home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/Pacman.cpp, pacman::Pacman, 21
34
~Sdl2
/home/arthur/tek2/OOP/projects/Arcade/lib/games/pacman/Pacman.h, pacman::Sdl2, 24
35
~Sfml
/home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.cpp, arcade::Sfml, 27
36
~Snake
/home/arthur/tek2/OOP/projects/Arcade/lib/games/snake/Snake.h, arcade::Snake, 30
36, 37
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplayModule.cpp, ADisplayModule
37
arcade::ADisplayModule, 12
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ADisplayModule.h, AGameModule
38
arcade::AGameModule, 15
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/NCurses.cpp, arcade, 9
38
arcade::ADisplayModule, 11
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/ncurses/NCurses.h, coreModule, 14
39, 40
_input, 14
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.cpp, ADisplayModule, 12
40
ADisplayModule, 12
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sdl2/Sdl2.h, clearWindow, 12
41
displayWindow, 12
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.cpp, drawAllSprite, 12
42
drawSprite, 12
/home/arthur/tek2/OOP/projects/Arcade/lib/graphics/sfml/Sfml.h, drawText, 13
42, 43
getCoreModule, 13
/home/arthur/tek2/OOP/projects/Arcade/src/CoreModule.cpp, getInput, 13
43
getName, 13
/home/arthur/tek2/OOP/projects/Arcade/src/ErrorHandling.cpp, setCoreModule, 13
43
arcade::AGameModule, 14
/home/arthur/tek2/OOP/projects/Arcade/src/Main.cpp, ~AGameModule, 15
44
AGameModule, 15
_coreModule
getCoreModule, 15
arcade::ADisplayModule, 14
getDirection, 15
_input
getGameStatus, 15
arcade::ADisplayModule, 14
getLayerCell, 16
_renderer
getName, 16
arcade::Sdl2, 26
handleKeyEvents, 16
_texture
init, 16
arcade::Sfml, 29
setCoreModule, 16
_window
setDirection, 17
arcade::NCurses, 20
setGameStatus, 17
arcade::Sdl2, 26
updateGame, 17
arcade::Sfml, 29
arcade::NCurses, 17
~ADisplayModule
_window, 20
arcade::ADisplayModule, 12
~NCurses, 18
~AGameModule
clearWindow, 19
displayWindow, 19

- drawAllSprite, 19
- drawSprite, 19
- drawText, 19
- getInput, 19
- getName, 20
- NCurses, 18
- arcade::Pacman, 20
 - ~Pacman, 21
 - getName, 21
 - handleKeyEvents, 21
 - init, 22
 - movePacman, 22
 - Pacman, 21
 - updateGame, 22
- arcade::Sdl2, 22
 - _renderer, 26
 - _window, 26
 - ~Sdl2, 24
 - clearWindow, 24
 - displayWindow, 24
 - drawAllSprite, 24
 - drawSprite, 24
 - drawText, 25
 - getInput, 25
 - getName, 25
 - Sdl2, 24
- arcade::Sfml, 26
 - _texture, 29
 - _window, 29
 - ~Sfml, 27
 - clearWindow, 28
 - displayWindow, 28
 - drawAllSprite, 28
 - drawSprite, 28
 - drawText, 28
 - getInput, 28
 - getName, 29
 - Sfml, 27
- arcade::Snake, 29
 - ~Snake, 30
 - getName, 31
 - handleKeyEvents, 31
 - init, 31
 - moveSnake, 31
 - Snake, 30
 - updateGame, 32
- checkPath
 - ErrorHandling.cpp, 44
- clearWindow
 - arcade::ADisplayModule, 12
 - arcade::NCurses, 19
 - arcade::Sdl2, 24
 - arcade::Sfml, 28
- displayWindow
 - arcade::ADisplayModule, 12
 - arcade::NCurses, 19
 - arcade::Sdl2, 24
- arcade::Sfml, 28
- drawAllSprite
 - arcade::ADisplayModule, 12
 - arcade::NCurses, 19
 - arcade::Sdl2, 24
 - arcade::Sfml, 28
- drawSprite
 - arcade::ADisplayModule, 12
 - arcade::NCurses, 19
 - arcade::Sdl2, 24
 - arcade::Sfml, 28
- drawText
 - arcade::ADisplayModule, 13
 - arcade::NCurses, 19
 - arcade::Sdl2, 25
 - arcade::Sfml, 28
- entryPoint
 - NCurses.cpp, 39
 - Pacman.cpp, 34
 - Sdl2.cpp, 40
 - Sfml.cpp, 42
 - Snake.cpp, 36
- ErrorHandling.cpp
 - checkPath, 44
- getCoreModule
 - arcade::ADisplayModule, 13
 - arcade::AGameModule, 15
- getDirection
 - arcade::AGameModule, 15
- getGameStatus
 - arcade::AGameModule, 15
- getInput
 - arcade::ADisplayModule, 13
 - arcade::NCurses, 19
 - arcade::Sdl2, 25
 - arcade::Sfml, 28
- getLayerCell
 - arcade::AGameModule, 16
- getName
 - arcade::ADisplayModule, 13
 - arcade::AGameModule, 16
 - arcade::NCurses, 20
 - arcade::Pacman, 21
 - arcade::Sdl2, 25
 - arcade::Sfml, 29
 - arcade::Snake, 31
 - NCurses.cpp, 39
 - Pacman.cpp, 34
 - Sdl2.cpp, 40
 - Sfml.cpp, 42
 - Snake.cpp, 36
- getType
 - NCurses.cpp, 39
 - Pacman.cpp, 34
 - Sdl2.cpp, 41
 - Sfml.cpp, 42
 - Snake.cpp, 36

- handleKeyEvents
 - arcade::AGameModule, [16](#)
 - arcade::Pacman, [21](#)
 - arcade::Snake, [31](#)
- init
 - arcade::AGameModule, [16](#)
 - arcade::Pacman, [22](#)
 - arcade::Snake, [31](#)
- main
 - Main.cpp, [44](#)
- Main.cpp
 - main, [44](#)
- MAX_SNAKE_SIZE
 - Snake.hpp, [37](#)
- movePacman
 - arcade::Pacman, [22](#)
- moveSnake
 - arcade::Snake, [31](#)
- NCurses
 - arcade::NCurses, [18](#)
- NCurses.cpp
 - entryPoint, [39](#)
 - getName, [39](#)
 - getType, [39](#)
- Pacman
 - arcade::Pacman, [21](#)
- Pacman.cpp
 - entryPoint, [34](#)
 - getName, [34](#)
 - getType, [34](#)
- Pacman.hpp
 - SPEED_SNAKE, [35](#)
- Sdl2
 - arcade::Sdl2, [24](#)
- Sdl2.cpp
 - entryPoint, [40](#)
 - getName, [40](#)
 - getType, [41](#)
- setCoreModule
 - arcade::ADisplayModule, [13](#)
 - arcade::AGameModule, [16](#)
- setDirection
 - arcade::AGameModule, [17](#)
- setGameStatus
 - arcade::AGameModule, [17](#)
- Sfml
 - arcade::Sfml, [27](#)
- Sfml.cpp
 - entryPoint, [42](#)
 - getName, [42](#)
 - getType, [42](#)
- Snake
 - arcade::Snake, [30](#)
- Snake.cpp
 - entryPoint, [36](#)
 - getName, [36](#)
 - getType, [36](#)
- Snake.hpp
 - MAX_SNAKE_SIZE, [37](#)
 - SPEED_SNAKE, [37](#)
- SPEED_SNAKE
 - Pacman.hpp, [35](#)
 - Snake.hpp, [37](#)
- updateGame
 - arcade::AGameModule, [17](#)
 - arcade::Pacman, [22](#)
 - arcade::Snake, [32](#)