

## Arcade

Generated by Doxygen 1.9.7



<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 arcade Namespace Reference	9
<b>6 Class Documentation</b>	<b>11</b>
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()	12
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
6.2 arcade::AGameModule Class Reference	14
6.2.1 Constructor & Destructor Documentation	14
6.2.1.1 AGameModule()	14
6.2.1.2 ~AGameModule()	14
6.2.2 Member Function Documentation	15
6.2.2.1 getCoreModule()	15
6.2.2.2 getDirection()	15
6.2.2.3 getGameStatus()	15
6.2.2.4 getLayerCell()	15
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()	16

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()	18
6.3.2 Member Function Documentation	18
6.3.2.1 clearWindow()	18
6.3.2.2 displayWindow()	18
6.3.2.3 drawAllSprite()	19
6.3.2.4 drawSprite()	19
6.3.2.5 drawText()	19
6.3.2.6 getInput()	19
6.3.2.7 getName()	19
6.3.3 Member Data Documentation	19
6.3.3.1 _window	19
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()	21
6.4.2.4 updateGame()	21
6.5 arcade::Sdl2 Class Reference	22
6.5.1 Constructor & Destructor Documentation	23
6.5.1.1 Sdl2()	23
6.5.1.2 ~Sdl2()	23
6.5.2 Member Function Documentation	23
6.5.2.1 clearWindow()	23
6.5.2.2 displayWindow()	23
6.5.2.3 drawAllSprite()	23
6.5.2.4 drawSprite()	23
6.5.2.5 drawText()	24
6.5.2.6 getInput()	24
6.5.2.7 getName()	24
6.5.3 Member Data Documentation	25
6.5.3.1 _font	25
6.5.3.2 _renderer	25
6.5.3.3 _window	25
6.6 arcade::Sfml Class Reference	25

6.6.1 Constructor & Destructor Documentation	26
6.6.1.1 SfmI()	26
6.6.1.2 ~SfmI()	26
6.6.2 Member Function Documentation	26
6.6.2.1 clearWindow()	26
6.6.2.2 displayWindow()	27
6.6.2.3 drawAllSprite()	27
6.6.2.4 drawSprite()	27
6.6.2.5 drawText()	27
6.6.2.6 getInput()	27
6.6.2.7 getName()	27
6.6.3 Member Data Documentation	28
6.6.3.1 _font	28
6.6.3.2 _texture	28
6.6.3.3 _window	28
6.7 arcade::Snake Class Reference	28
6.7.1 Constructor & Destructor Documentation	29
6.7.1.1 Snake()	29
6.7.1.2 ~Snake()	29
6.7.2 Member Function Documentation	29
6.7.2.1 generateFood()	29
6.7.2.2 getName()	29
6.7.2.3 handleKeyEvents()	29
6.7.2.4 init()	30
6.7.2.5 moveSnake()	30
6.7.2.6 updateGame()	30
<b>7 File Documentation</b>	<b>31</b>
7.1 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGameModule.cpp File Reference	31
7.2 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGameModule.hpp File Reference	31
7.3 AGameModule.hpp	31
7.4 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/pacman/Pacman.cpp File Reference	32
7.4.1 Function Documentation	32
7.4.1.1 entryPoint()	32
7.4.1.2 getName()	32
7.4.1.3 getType()	32
7.5 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/pacman/Pacman.hpp File Reference	33
7.6 Pacman.hpp	33
7.7 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/Snake.cpp File Reference	33
7.7.1 Function Documentation	34
7.7.1.1 entryPoint()	34
7.7.1.2 getName()	34

7.7.1.3 <a href="#">getType()</a>	34
7.8 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/Snake.hpp File Reference</a>	34
7.8.1 Macro Definition Documentation	35
7.8.1.1 BODY	35
7.8.1.2 EMPTY	35
7.8.1.3 FOOD	35
7.8.1.4 FOOD_LAYER	35
7.8.1.5 HEAD	35
7.8.1.6 MAP_LAYER	35
7.8.1.7 SNAKE_LAYER	35
7.8.1.8 SNAKE_SPEED	35
7.8.1.9 WALL	35
7.9 Snake.hpp	36
7.10 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ADisplayModule.cpp File Reference</a>	36
7.11 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ADisplayModule.hpp File Reference</a>	36
7.12 ADisplayModule.hpp	37
7.13 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/NCurses.cpp File Reference</a>	37
7.13.1 Function Documentation	37
7.13.1.1 <a href="#">entryPoint()</a>	37
7.13.1.2 <a href="#">getName()</a>	38
7.13.1.3 <a href="#">getType()</a>	38
7.14 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/NCurses.hpp File Reference</a>	38
7.15 NCurses.hpp	38
7.16 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/Sdl2.cpp File Reference</a>	39
7.16.1 Function Documentation	39
7.16.1.1 <a href="#">entryPoint()</a>	39
7.16.1.2 <a href="#">getName()</a>	39
7.16.1.3 <a href="#">getType()</a>	39
7.17 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/Sdl2.hpp File Reference</a>	39
7.18 Sdl2.hpp	40
7.19 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/Sfml.cpp File Reference</a>	40
7.19.1 Function Documentation	41
7.19.1.1 <a href="#">entryPoint()</a>	41
7.19.1.2 <a href="#">getName()</a>	41
7.19.1.3 <a href="#">getType()</a>	41
7.20 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/Sfml.hpp File Reference</a>	41
7.21 Sfml.hpp	42
7.22 <a href="#">/home/tmendy/Documents/Tek2/OOP/Arcade/src/CoreModule.cpp File Reference</a>	42
7.22.1 Function Documentation	43
7.22.1.1 <a href="#">addScoreInFile()</a>	43
7.22.1.2 <a href="#">generateFocusVersion()</a>	43
7.22.1.3 <a href="#">getScoreFromFile()</a>	43

---

7.22.1.4 max_len_line()	43
7.22.1.5 split_str()	43
7.23 /home/tmendy/Documents/Tek2/OOP/Arcade/src/Main.cpp File Reference	43
7.23.1 Function Documentation	44
7.23.1.1 help()	44
7.23.1.2 is_good_graphic_lib()	44
7.23.1.3 main()	44
7.24 /home/tmendy/Documents/Tek2/OOP/Arcade/src/Shell.cpp File Reference	44
<b>Index</b>	<b>45</b>





# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">arcade</a>	.....	9
------------------------	-------	---



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

arcade::IDisplayModule	
arcade::ADisplayModule . . . . .	<a href="#">11</a>
arcade::NCurses . . . . .	<a href="#">17</a>
arcade::Sdl2 . . . . .	<a href="#">22</a>
arcade::Sfml . . . . .	<a href="#">25</a>
arcade::IGameModule	
arcade::AGameModule . . . . .	<a href="#">14</a>
arcade::Pacman . . . . .	<a href="#">20</a>
arcade::Snake . . . . .	<a href="#">28</a>



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">arcade::ADisplayModule</a>	11
<a href="#">arcade::AGameModule</a>	14
<a href="#">arcade::NCurses</a>	17
<a href="#">arcade::Pacman</a>	20
<a href="#">arcade::Sdl2</a>	22
<a href="#">arcade::Sfml</a>	25
<a href="#">arcade::Snake</a>	28



# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

/home/tmندی/Documents/Tek2/OOP/Arcade/lib/games/ <a href="#">AGameModule.cpp</a> . . . . .	31
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/games/ <a href="#">AGameModule.hpp</a> . . . . .	31
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/games/pacman/ <a href="#">Pacman.cpp</a> . . . . .	32
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/games/pacman/ <a href="#">Pacman.hpp</a> . . . . .	33
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/games/snake/ <a href="#">Snake.cpp</a> . . . . .	33
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/games/snake/ <a href="#">Snake.hpp</a> . . . . .	34
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/ <a href="#">ADisplayModule.cpp</a> . . . . .	36
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/ <a href="#">ADisplayModule.hpp</a> . . . . .	36
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/ <a href="#">NCurses.cpp</a> . . . . .	37
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/ <a href="#">NCurses.hpp</a> . . . . .	38
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/ <a href="#">Sdl2.cpp</a> . . . . .	39
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/ <a href="#">Sdl2.hpp</a> . . . . .	39
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/ <a href="#">Sfml.cpp</a> . . . . .	40
/home/tmندی/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/ <a href="#">Sfml.hpp</a> . . . . .	41
/home/tmندی/Documents/Tek2/OOP/Arcade/src/ <a href="#">CoreModule.cpp</a> . . . . .	42
/home/tmندی/Documents/Tek2/OOP/Arcade/src/ <a href="#">Main.cpp</a> . . . . .	43
/home/tmندی/Documents/Tek2/OOP/Arcade/src/ <a href="#">Shell.cpp</a> . . . . .	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](#)



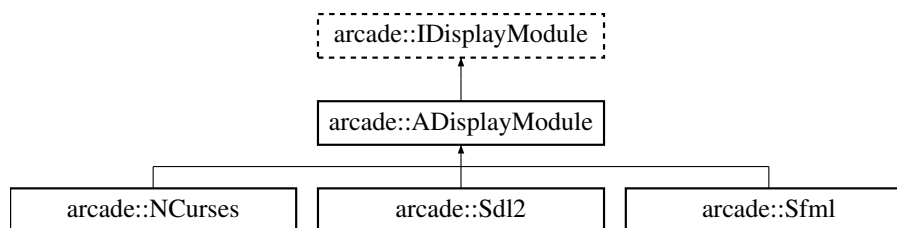
## Chapter 6

# 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](#) ()=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

## 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 ( ) [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

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

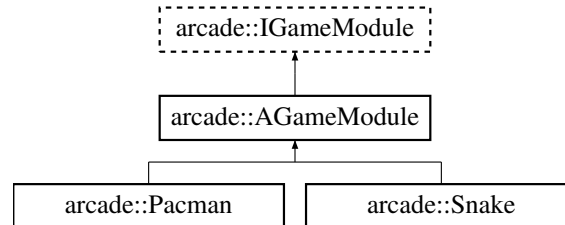
The documentation for this class was generated from the following files:

- [/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ADisplayModule.hpp](#)
- [/home/tmendy/Documents/Tek2/OOP/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 std::string [getName](#) ()=0
- virtual void [init](#) ()=0
- virtual void [updateGame](#) ()=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*
- int [getLayerCell](#) (int layer, int x, int y) const  
*get the cell of the map*
- virtual void [handleKeyEvents](#) (arcade::KeyboardInput key)=0

### 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 map

#### 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 handleKeyEvents()**

```
virtual void arcade::AGameModule::handleKeyEvents (
    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 (
    GameStatus status )
```

### 6.2.2.11 updateGame()

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

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

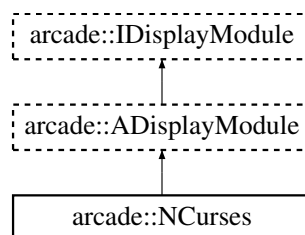
The documentation for this class was generated from the following files:

- [/home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGameModule.hpp](#)
- [/home/tmendy/Documents/Tek2/OOP/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](#) ()
- [void clearWindow](#) ()
- [void displayWindow](#) ()
- [arcade::KeyboardInput getInput](#) ()
- [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)
- [void drawText](#) (const std::string text, int x, int y, int size)

## 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](#) ()=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](#)

## 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 ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

## 6.3.3 Member Data Documentation

### 6.3.3.1 \_window

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

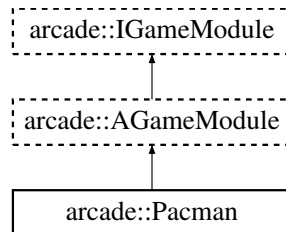
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/NCurses.hpp
- /home/tmendy/Documents/Tek2/OOP/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.*
- `std::string` [getName](#) ()
- `void` [init](#) ()
- `void` [updateGame](#) ()
- `void` [handleKeyEvents](#) (arcade::KeyboardInput key)

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

- [AGameModule](#) ()  
*Construct a new [arcade::AGameModule::AGameModule](#) object.*
- [~AGameModule](#) ()  
*Destroy the [arcade::AGameModule::AGameModule](#) object.*
- `virtual std::string` [getName](#) ()=0
- `virtual void` [init](#) ()=0
- `virtual void` [updateGame](#) ()=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*
- `int` [getLayerCell](#) (int layer, int x, int y) const  
*get the cell of the map*
- `virtual void` [handleKeyEvents](#) (arcade::KeyboardInput key)=0

## 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]
```

Implements [arcade::AGameModule](#).

### 6.4.2.2 handleKeyEvents()

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

Implements [arcade::AGameModule](#).

### 6.4.2.3 init()

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

Implements [arcade::AGameModule](#).

### 6.4.2.4 updateGame()

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

Implements [arcade::AGameModule](#).

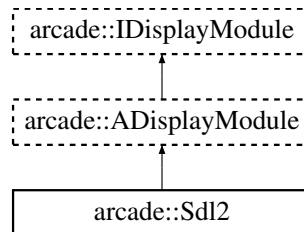
The documentation for this class was generated from the following files:

- [/home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/pacman/Pacman.hpp](#)
- [/home/tmendy/Documents/Tek2/OOP/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](#) ()
- `void` [clearWindow](#) ()  
*clear the window*
- `void` [displayWindow](#) ()
- `arcade::KeyboardInput` [getInput](#) ()  
*get the input from 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)
- `void` [drawText](#) (`const` `std::string` text, `int` x, `int` y, `int` size)  
*draw text 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](#) ()=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\_Window \* [\\_window](#)
- TTF\_Font \* [\\_font](#)
- SDL\_Renderer \* [\\_renderer](#)

## 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]
```

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 ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).



## 6.5.3 Member Data Documentation

### 6.5.3.1 \_font

TTF\_Font\* arcade::Sdl2::\_font [protected]

### 6.5.3.2 \_renderer

SDL\_Renderer\* arcade::Sdl2::\_renderer [protected]

### 6.5.3.3 \_window

SDL\_Window\* arcade::Sdl2::\_window [protected]

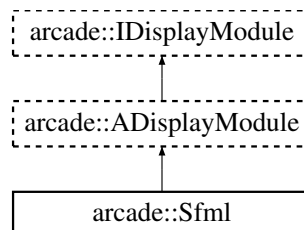
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/[Sdl2.hpp](#)
- /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/[Sdl2.cpp](#)

## 6.6 arcade::Sfml Class Reference

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

Inheritance diagram for arcade::Sfml:



### Public Member Functions

- [Sfml](#) ()  
Construct a new [arcade::Sfml::Sfml](#) object.
- [~Sfml](#) ()  
Destroy the [arcade::Sfml::Sfml](#) object.
- std::string [getName](#) ()
- void [clearWindow](#) ()
- void [displayWindow](#) ()
- arcade::KeyboardInput [getInput](#) ()
- 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)
- void [drawText](#) (const std::string text, int x, int y, int size)

## 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](#) ()=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](#)
- sf::Font [\\_font](#)

## 6.6.1 Constructor & Destructor Documentation

### 6.6.1.1 Sfml()

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

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

### 6.6.1.2 ~Sfml()

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

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

## 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 ( ) [virtual]
```

Implements [arcade::ADisplayModule](#).

## 6.6.3 Member Data Documentation

### 6.6.3.1 `_font`

```
sf::Font arcade::Sfml::_font [protected]
```

### 6.6.3.2 `_texture`

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

### 6.6.3.3 `_window`

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

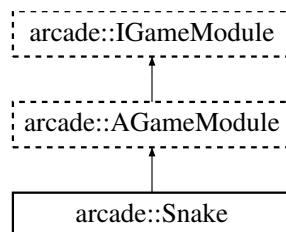
The documentation for this class was generated from the following files:

- [/home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/Sfml.hpp](#)
- [/home/tmendy/Documents/Tek2/OOP/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](#) ()
- `void` [init](#) ()
- `void` [updateGame](#) ()  
update the game
- `arcade::GameData` [moveSnake](#) ()
- `void` [handleKeyEvents](#) (`arcade::KeyboardInput` key)  
handle key events
- `bool` [generateFood](#) (`std::vector< std::vector< arcade::entity > > &entities`)

## 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 std::string [getName](#) ()=0
- virtual void [init](#) ()=0
- virtual void [updateGame](#) ()=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*
- int [getLayerCell](#) (int layer, int x, int y) const  
*get the cell of the map*
- virtual void [handleKeyEvents](#) (arcade::KeyboardInput key)=0

## 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 generateFood()

```
bool arcade::Snake::generateFood (
    std::vector< std::vector< arcade::entity > > & entities )
```

### 6.7.2.2 getName()

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

Implements [arcade::AGameModule](#).

### 6.7.2.3 handleKeyEvents()

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

handle key events

#### Parameters

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

Implements [arcade::AGameModule](#).

#### 6.7.2.4 init()

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

Implements [arcade::AGameModule](#).

#### 6.7.2.5 moveSnake()

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

#### 6.7.2.6 updateGame()

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

update the game

Implements [arcade::AGameModule](#).

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

- [/home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/Snake.hpp](#)
- [/home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/Snake.cpp](#)

## Chapter 7

# File Documentation

### 7.1 `/home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGameModule.cpp` File Reference

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

### 7.2 `/home/tmendy/Documents/Tek2/OOP/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 class AGameModule : virtual public arcade::IGameModule {
00015 public:
00016     AGameModule();
00017     ~AGameModule();
00018
00019     virtual std::string getName() = 0;
00020
00021     virtual void init() = 0;
00022     virtual void updateGame() = 0;
00023
00024     void setGameStatus(GameStatus status);
00025     GameStatus getGameStatus() const;
00026
00027     void setCoreModule(arcade::CoreModule *coreModule);
00028     arcade::CoreModule *getCoreModule() const;
00029
00030     void setDirection(arcade::KeyboardInput direction);
00031     arcade::KeyboardInput getDirection() const;
00032
00033     int getLayerCell(int layer, int x, int y) const;
00034
00035     virtual void handleKeyEvents(arcade::KeyboardInput key) = 0;
00036 };
00037 }; // namespace arcade
00038
00039 #endif /* !IGAMEMODULE_HPP_ */

```

## 7.4 /home/tmendy/Documents/Tek2/OOP/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/tmendy/Documents/Tek2/OOP/Arcade/lib/games/pacman/↵ Pacman.hpp File Reference

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

### Classes

- class [arcade::Pacman](#)

### Namespaces

- namespace [arcade](#)

## 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
00011 #include "../AGameModule.hpp"
00012
00013 namespace arcade {
00014 class Pacman : virtual public arcade::AGameModule {
00015 public:
00016     Pacman();
00017     ~Pacman();
00018     std::string getName();
00019     void init();
00020     void updateGame();
00021     void handleKeyEvents(arcade::KeyboardInput key){};
00022
00023 protected:
00024 };
00025 }; // namespace arcade
00026
00027 #endif /* !PACMAN_HPP_ */
```

## 7.7 /home/tmendy/Documents/Tek2/OOP/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/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/`↵ **Snake.hpp File Reference**

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

### Classes

- class [arcade::Snake](#)

### Namespaces

- namespace [arcade](#)

### Macros

- `#define` [HEAD](#) 'P'
- `#define` [BODY](#) '\*'
- `#define` [FOOD](#) 'F'
- `#define` [WALL](#) '#'
- `#define` [EMPTY](#) ''
- `#define` [SNAKE\\_SPEED](#) 1
- `#define` [MAP\\_LAYER](#) 0
- `#define` [FOOD\\_LAYER](#) 1
- `#define` [SNAKE\\_LAYER](#) 2

## 7.8.1 Macro Definition Documentation

### 7.8.1.1 BODY

```
#define BODY '*'
```

### 7.8.1.2 EMPTY

```
#define EMPTY ' '
```

### 7.8.1.3 FOOD

```
#define FOOD 'F'
```

### 7.8.1.4 FOOD\_LAYER

```
#define FOOD_LAYER 1
```

### 7.8.1.5 HEAD

```
#define HEAD 'P'
```

### 7.8.1.6 MAP\_LAYER

```
#define MAP_LAYER 0
```

### 7.8.1.7 SNAKE\_LAYER

```
#define SNAKE_LAYER 2
```

### 7.8.1.8 SNAKE\_SPEED

```
#define SNAKE_SPEED 1
```

### 7.8.1.9 WALL

```
#define WALL '#'
```

## 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
00011 #include "../AGameModule.hpp"
00012
00013 namespace arcade {
00014 class Snake : virtual public arcade::AGameModule {
00015 public:
00016 #define HEAD 'P'
00017 #define BODY '*'
00018 #define FOOD 'F'
00019 #define WALL '#'
00020 #define EMPTY ' '
00021 #define SNAKE_SPEED 1
00022 #define MAP_LAYER 0
00023 #define FOOD_LAYER 1
00024 #define SNAKE_LAYER 2
00025 Snake();
00026 ~Snake();
00027 std::string getName();
00028 void init();
00029 void updateGame();
00030 arcade::GameData moveSnake();
00031 void handleKeyEvents(arcade::KeyboardInput key);
00032 bool generateFood(std::vector<std::vector<arcade::entity>> &entities);
00033 };
00034 }; // namespace arcade
00035
00036 #endif /* !SNAKE_HPP_ */
```

## 7.10 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/↵ ADisplayModule.cpp File Reference

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

## 7.11 /home/tmendy/Documents/Tek2/OOP/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 #ifndef ADISPLAYMODULE_HPP_
00009 #define ADISPLAYMODULE_HPP_
00010
00011 #include <arcade/IDisplayModule.hpp>
00012
00013 namespace arcade {
00014 class ADisplayModule : virtual public arcade::IDisplayModule {
00015 public:
00016     ADisplayModule();
00017     ~ADisplayModule();
00018
00019     virtual std::string getName() = 0;
00020
00021     void setCoreModule(arcade::CoreModule *coreModule);
00022     arcade::CoreModule *getCoreModule() const;
00023
00024     virtual void clearWindow() = 0;
00025     virtual void displayWindow() = 0;
00026     virtual arcade::KeyboardInput getInput() = 0;
00027     virtual void drawSprite(std::pair<char, std::string> sprite,
00028                             int x,
00029                             int y,
00030                             int width,
00031                             int height) = 0;
00032     virtual void drawAllSprite(std::pair<char, std::string> sprite,
00033                                std::vector<std::pair<int, int> > coordinates,
00034                                int width,
00035                                int height) = 0;
00036     virtual void drawText(const std::string text, int x, int y, int size) = 0;
00037 };
00038 }; // namespace arcade
00039
00040 #endif /* !ADISPLAYMODULE_HPP_ */
```

## 7.13 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/↵ NCurses.cpp File Reference

```
#include "NCurses.hpp"
#include "Error.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/tmendy/Documents/Tek2/OOP/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();
00020     void clearWindow();
00021     void displayWindow();
00022     arcade::KeyboardInput getInput();
00023     void drawSprite(
00024         std::pair<char, std::string> sprite, int x, int y, int width, int height);
00025     void drawAllSprite(std::pair<char, std::string> sprite,
00026                       std::vector<std::pair<int, int>> coordinates,
00027                       int width,
00028                       int height);
00029     void drawText(const std::string text, int x, int y, int size);
00030
00031 protected:
00032     WINDOW *_window;
00033 };
00034 }; // namespace arcade
00035
00036 #endif /* !NCURSES_HPP_ */
```

## 7.16 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/Sdl2.cpp File Reference

```
#include "Sdl2.hpp"
#include "Error.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/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/Sdl2.hpp File Reference

```
#include "../ADisplayModule.hpp"
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_ttf.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_image.h>
00014 #include <SDL2/SDL_ttf.h>
00015
00016 namespace arcade {
00017 class Sdl2 : virtual public arcade::ADisplayModule {
00018 public:
00019     Sdl2();
00020     ~Sdl2();
00021     std::string getName();
00022     void clearWindow();
00023     void displayWindow();
00024     arcade::KeyboardInput getInput();
00025     void drawSprite (
00026         std::pair<char, std::string> sprite, int x, int y, int width, int height);
00027     void drawAllSprite(std::pair<char, std::string> sprite,
00028         std::vector<std::pair<int, int>> coordinates,
00029         int width,
00030         int height);
00031     void drawText(const std::string text, int x, int y, int size);
00032
00033 protected:
00034     SDL_Window *_window;
00035     TTF_Font *_font;
00036     SDL_Renderer *_renderer;
00037 };
00038 }; // namespace arcade
00039
00040 #endif /* !SDL2_HPP_ */
```

## 7.19 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/↵ Sfml.cpp File Reference

```
#include "Sfml.hpp"
#include "Error.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/tmendy/Documents/Tek2/OOP/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     std::string getName();
00020
00021     void clearWindow();
00022     void displayWindow();
00023     arcade::KeyboardInput getInput();
00024     void drawSprite(
00025         std::pair<char, std::string> sprite, int x, int y, int width, int height);
00026     void drawAllSprite(std::pair<char, std::string> sprite,
00027                       std::vector<std::pair<int, int> coordinates,
00028                               int width,
00029                               int height);
00030     void drawText(const std::string text, int x, int y, int size);
00031
00032 protected:
00033     sf::RenderWindow _window;
00034     sf::Texture _texture;
00035     sf::Font _font;
00036 };
00037 }; // namespace arcade
00038
00039 #endif /* !SFML_HPP_ */
```

## 7.22 /home/tmendy/Documents/Tek2/OOP/Arcade/src/CoreModule.cpp

### File Reference

```
#include <CoreModule.hpp>
#include <Error.hpp>
#include <fcntl.h>
#include <fstream>
#include <sstream>
#include <sys/stat.h>
#include <unistd.h>
```

### Functions

- `std::vector< std::string > split_str` (`std::string const &str`, `const char delim`)
- `static void generateFocusVersion` (`std::string &section`, `int len`)
- `int max_len_line` (`std::string str`)
- `std::vector< std::pair< std::string, int > > getScoreFromFile` (`std::string moduleName`)
- `static void addScoreInFile` (`std::string path`, `int score`, `std::string username`)

## 7.22.1 Function Documentation

### 7.22.1.1 addScoreInFile()

```
static void addScoreInFile (
    std::string path,
    int score,
    std::string username ) [static]
```

### 7.22.1.2 generateFocusVersion()

```
static void generateFocusVersion (
    std::string & section,
    int len ) [static]
```

### 7.22.1.3 getScoreFromFile()

```
std::vector< std::pair< std::string, int > > getScoreFromFile (
    std::string moduleName )
```

### 7.22.1.4 max\_len\_line()

```
int max_len_line (
    std::string str )
```

### 7.22.1.5 split\_str()

```
std::vector< std::string > split_str (
    std::string const & str,
    const char delim )
```

## 7.23 /home/tmendy/Documents/Tek2/OOP/Arcade/src/Main.cpp File Reference

```
#include <Arcade.hpp>
#include <CoreModule.hpp>
#include <iostream>
#include <unistd.h>
#include "Shell.hpp"
```

### Functions

- bool [is\\_good\\_graphic\\_lib](#) (char \*path\_graphic\_lib)  
*check if the library is a good graphic library*
- void [help](#) (void)  
*display the help*
- int [main](#) (int ac, char \*\*av)

## 7.23.1 Function Documentation

### 7.23.1.1 help()

```
void help (
    void )
```

display the help

### 7.23.1.2 is\_good\_graphic\_lib()

```
bool is_good_graphic_lib (
    char * path_graphic_lib )
```

check if the library is a good graphic library

#### Parameters

<i>path_graphic_lib</i>	path of the graphic library
-------------------------	-----------------------------

#### Returns

true if the library is a good graphic library  
false if the library is not a good graphic library

### 7.23.1.3 main()

```
int main (
    int ac,
    char ** av )
```

## 7.24 /home/tmendy/Documents/Tek2/OOP/Arcade/src/Shell.cpp File Reference

```
#include "Shell.hpp"
#include <Arcade.hpp>
#include <CoreModule.hpp>
```

# Index

/home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGameModule.cpp, 31  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGameModule.hpp, 31  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/pacman/Sdl2Pacman.cpp, 32  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/pacman/Sdl2Pacman.hpp, 33  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/Snake.cpp, 33  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/Snake.hpp, 34, 36  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ADisplayModule.cpp, 36  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ADisplayModule.hpp, 36, 37  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/NCurses.cpp, 37  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/NCurses.hpp, 38  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/Sdl2.cpp, 39  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/Sdl2.hpp, 39, 40  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/Sfml.cpp, 40  
 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/Sfml.hpp, 41, 42  
 /home/tmendy/Documents/Tek2/OOP/Arcade/src/CoreModule.cpp, 42  
 /home/tmendy/Documents/Tek2/OOP/Arcade/src/Main.cpp, 43  
 /home/tmendy/Documents/Tek2/OOP/Arcade/src/Shell.cpp, 44  
 \_font  
     arcade::Sdl2, 25  
     arcade::Sfml, 28  
 \_renderer  
     arcade::Sdl2, 25  
 \_texture  
     arcade::Sfml, 28  
 \_window  
     arcade::NCurses, 19  
     arcade::Sdl2, 25  
     arcade::Sfml, 28  
 ~ADisplayModule  
     arcade::ADisplayModule, 12  
 ~AGameModule  
     arcade::AGameModule, 14  
 arcade::NCurses, 18  
 arcade::Pacman, 21  
 arcade::Sdl2, 23  
 arcade::Sfml, 26  
 arcade::Snake, 29  
 addScoreInFile  
 CoreModule.cpp, 43  
 ADisplayModule  
     arcade::ADisplayModule, 12  
 AGameModule  
     arcade::AGameModule, 14  
 arcade, 9  
 arcade::ADisplayModule, 11  
     ~ADisplayModule, 12  
 ADisplayModule, 12  
     clearWindow, 12  
     displayWindow, 12  
     drawAllSprite, 12  
     drawSprite, 12  
     drawText, 12  
 getCoreModule, 13  
 getInput, 13  
 getName, 13  
 setCoreModule, 13  
 arcade::AGameModule, 14  
     ~AGameModule, 14  
 AGameModule, 14  
     getCoreModule, 15  
     getDirection, 15  
     getGameStatus, 15  
     getLayerCell, 15  
     getName, 16  
     handleKeyEvents, 16  
     init, 16  
     setCoreModule, 16  
     setDirection, 16  
     setGameStatus, 16  
     updateGame, 17  
 arcade::NCurses, 17  
     \_window, 19  
     ~NCurses, 18  
     clearWindow, 18  
     displayWindow, 18  
     drawAllSprite, 18

- drawSprite, 19
- drawText, 19
- getInput, 19
- getName, 19
- NCurses, 18
- arcade::Pacman, 20
  - ~Pacman, 21
  - getName, 21
  - handleKeyEvents, 21
  - init, 21
  - Pacman, 21
  - updateGame, 21
- arcade::Sdl2, 22
  - \_font, 25
  - \_renderer, 25
  - \_window, 25
  - ~Sdl2, 23
  - clearWindow, 23
  - displayWindow, 23
  - drawAllSprite, 23
  - drawSprite, 23
  - drawText, 24
  - getInput, 24
  - getName, 24
  - Sdl2, 23
- arcade::Sfml, 25
  - \_font, 28
  - \_texture, 28
  - \_window, 28
  - ~Sfml, 26
  - clearWindow, 26
  - displayWindow, 26
  - drawAllSprite, 27
  - drawSprite, 27
  - drawText, 27
  - getInput, 27
  - getName, 27
  - Sfml, 26
- arcade::Snake, 28
  - ~Snake, 29
  - generateFood, 29
  - getName, 29
  - handleKeyEvents, 29
  - init, 30
  - moveSnake, 30
  - Snake, 29
  - updateGame, 30
- BODY
  - Snake.hpp, 35
- clearWindow
  - arcade::ADisplayModule, 12
  - arcade::NCurses, 18
  - arcade::Sdl2, 23
  - arcade::Sfml, 26
- CoreModule.cpp
  - addScoreInFile, 43
  - generateFocusVersion, 43
  - getScoreFromFile, 43
  - max\_len\_line, 43
  - split\_str, 43
- displayWindow
  - arcade::ADisplayModule, 12
  - arcade::NCurses, 18
  - arcade::Sdl2, 23
  - arcade::Sfml, 26
- drawAllSprite
  - arcade::ADisplayModule, 12
  - arcade::NCurses, 18
  - arcade::Sdl2, 23
  - arcade::Sfml, 27
- drawSprite
  - arcade::ADisplayModule, 12
  - arcade::NCurses, 19
  - arcade::Sdl2, 23
  - arcade::Sfml, 27
- drawText
  - arcade::ADisplayModule, 12
  - arcade::NCurses, 19
  - arcade::Sdl2, 24
  - arcade::Sfml, 27
- EMPTY
  - Snake.hpp, 35
- entryPoint
  - NCurses.cpp, 37
  - Pacman.cpp, 32
  - Sdl2.cpp, 39
  - Sfml.cpp, 41
  - Snake.cpp, 34
- FOOD
  - Snake.hpp, 35
- FOOD\_LAYER
  - Snake.hpp, 35
- generateFocusVersion
  - CoreModule.cpp, 43
- generateFood
  - arcade::Snake, 29
- getCoreModule
  - arcade::ADisplayModule, 13
  - arcade::AGameModule, 15
- getDirection
  - arcade::AGameModule, 15
- getGameStatus
  - arcade::AGameModule, 15
- getInput
  - arcade::ADisplayModule, 13
  - arcade::NCurses, 19
  - arcade::Sdl2, 24
  - arcade::Sfml, 27
- getLayerCell
  - arcade::AGameModule, 15
- getName
  - arcade::ADisplayModule, 13

- arcade::AGameModule, [16](#)
- arcade::NCurses, [19](#)
- arcade::Pacman, [21](#)
- arcade::Sdl2, [24](#)
- arcade::Sfml, [27](#)
- arcade::Snake, [29](#)
- NCurses.cpp, [37](#)
- Pacman.cpp, [32](#)
- Sdl2.cpp, [39](#)
- Sfml.cpp, [41](#)
- Snake.cpp, [34](#)
- getScoreFromFile
  - CoreModule.cpp, [43](#)
- getType
  - NCurses.cpp, [38](#)
  - Pacman.cpp, [32](#)
  - Sdl2.cpp, [39](#)
  - Sfml.cpp, [41](#)
  - Snake.cpp, [34](#)
- handleKeyEvents
  - arcade::AGameModule, [16](#)
  - arcade::Pacman, [21](#)
  - arcade::Snake, [29](#)
- HEAD
  - Snake.hpp, [35](#)
- help
  - Main.cpp, [44](#)
- init
  - arcade::AGameModule, [16](#)
  - arcade::Pacman, [21](#)
  - arcade::Snake, [30](#)
- is\_good\_graphic\_lib
  - Main.cpp, [44](#)
- main
  - Main.cpp, [44](#)
- Main.cpp
  - help, [44](#)
  - is\_good\_graphic\_lib, [44](#)
  - main, [44](#)
- MAP\_LAYER
  - Snake.hpp, [35](#)
- max\_len\_line
  - CoreModule.cpp, [43](#)
- moveSnake
  - arcade::Snake, [30](#)
- NCurses
  - arcade::NCurses, [18](#)
- NCurses.cpp
  - entryPoint, [37](#)
  - getName, [37](#)
  - getType, [38](#)
- Pacman
  - arcade::Pacman, [21](#)
- Pacman.cpp
  - entryPoint, [32](#)
  - getName, [32](#)
  - getType, [32](#)
- Sdl2
  - arcade::Sdl2, [23](#)
- Sdl2.cpp
  - entryPoint, [39](#)
  - getName, [39](#)
  - getType, [39](#)
- setCoreModule
  - arcade::ADisplayModule, [13](#)
  - arcade::AGameModule, [16](#)
- setDirection
  - arcade::AGameModule, [16](#)
- setGameStatus
  - arcade::AGameModule, [16](#)
- Sfml
  - arcade::Sfml, [26](#)
- Sfml.cpp
  - entryPoint, [41](#)
  - getName, [41](#)
  - getType, [41](#)
- Snake
  - arcade::Snake, [29](#)
- Snake.cpp
  - entryPoint, [34](#)
  - getName, [34](#)
  - getType, [34](#)
- Snake.hpp
  - BODY, [35](#)
  - EMPTY, [35](#)
  - FOOD, [35](#)
  - FOOD\_LAYER, [35](#)
  - HEAD, [35](#)
  - MAP\_LAYER, [35](#)
  - SNAKE\_LAYER, [35](#)
  - SNAKE\_SPEED, [35](#)
  - WALL, [35](#)
- SNAKE\_LAYER
  - Snake.hpp, [35](#)
- SNAKE\_SPEED
  - Snake.hpp, [35](#)
- split\_str
  - CoreModule.cpp, [43](#)
- updateGame
  - arcade::AGameModule, [17](#)
  - arcade::Pacman, [21](#)
  - arcade::Snake, [30](#)
- WALL
  - Snake.hpp, [35](#)