Arcade

Generated by Doxygen 1.9.7

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 arcade Namespace Reference

Classes

- class ADisplayModule
- class AGameModule
- struct DataGhost
- struct DataPacman
- class NCurses
- struct Node
- class Pacman
- class Sdl2
- class Sfml
- class Snake

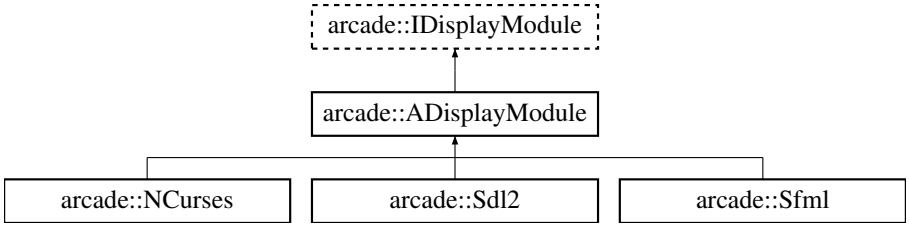
Chapter 6

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

coreModule	

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]
```

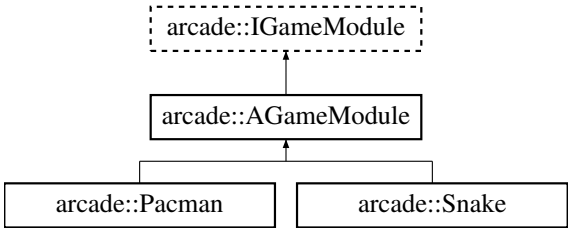
The documentation for this class was generated from the following files:

- /home/aleachlodnik/Arcade/lib/graphics/ADisplayModule.hpp
- /home/aleachlodnik/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
- 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

x	
y	

Returns

int

6.2.2.5 handddleKeyEvents()

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

Implemented in arcade::Pacman, and arcade::Snake.

6.2.2.6 init()

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

Implemented in arcade::Pacman, and arcade::Snake.

6.2.2.7 setCoreModule()

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

set the core module to the game module

Parameters

coreModule	

6.2.2.8 setDirection()

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

set the direction of the game

Parameters

direction	

6.2.2.9 setGameStatus()

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

6.2.2.10 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/aleachlodnik/Arcade/lib/games/AGameModule.hpp
- /home/aleachlodnik/Arcade/lib/games/AGameModule.cpp

6.3 CompareNode Struct Reference

Public Member Functions

- bool operator() (const arcade::Node lhs, const arcade::Node rhs) const

6.3.1 Member Function Documentation

6.3.1.1 operator()()

```
bool CompareNode::operator() (
            const arcade::Node lhs,
            const arcade::Node rhs ) const  [inline]
```

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

- /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.cpp

6.4 arcade::DataGhost Struct Reference

`#include <Pacman.hpp>`

Public Attributes

- bool isScared
- bool isDead
- std::pair< int, int > initialPos
- std::vector< arcade::Node > path
- int actualPathIndex
- std::chrono::time_point< std::chrono::system_clock > ghostTimer

6.4.1 Member Data Documentation

6.4.1.1 actualPathIndex

```
int arcade::DataGhost::actualPathIndex
```

6.4.1.2 ghostTimer

```
std::chrono::time_point<std::chrono::system_clock> arcade::DataGhost::ghostTimer
```

6.4.1.3 initialPos

```
std::pair<int, int> arcade::DataGhost::initialPos
```

6.4.1.4 isDead

```
bool arcade::DataGhost::isDead
```

6.4.1.5 isScared

```
bool arcade::DataGhost::isScared
```

6.4.1.6 path

```
std::vector<arcade::Node> arcade::DataGhost::path
```

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

- /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.hpp

6.5 arcade::DataPacman Struct Reference

`#include <Pacman.hpp>`

Public Attributes

- bool isBoosted
- std::chrono::time_point< std::chrono::system_clock > _pacmanTimer

6.5.1 Member Data Documentation

6.5.1.1 _pacmanTimer

`std::chrono::time_point<std::chrono::system_clock> arcade::DataPacman::_pacmanTimer`

6.5.1.2 isBoosted

`bool arcade::DataPacman::isBoosted`

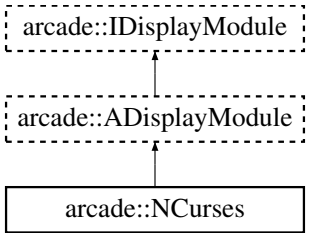
The documentation for this struct was generated from the following file:

- /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.hpp

6.6 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

6.6.1 Constructor & Destructor Documentation

6.6.1.1 NCurses()

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

6.6.1.2 ~NCurses()

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

6.6.2 Member Function Documentation

6.6.2.1 clearWindow()

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

Implements arcade::ADisplayModule.

6.6.2.2 displayWindow()

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

Implements arcade::ADisplayModule.

6.6.2.3 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.

6.6.2.4 drawSprite()

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

Implements arcade::ADisplayModule.

6.6.2.5 drawText()

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

Implements arcade::ADisplayModule.

6.6.2.6 getInput()

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

Implements arcade::ADisplayModule.

6.6.2.7 getName()

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

Implements arcade::ADisplayModule.

6.6.3 Member Data Documentation

6.6.3.1 _window

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

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

- /home/aleachlodnik/Arcade/lib/graphics/ncurses/NCurses.hpp
- /home/aleachlodnik/Arcade/lib/graphics/ncurses/NCurses.cpp

6.7 arcade::Node Struct Reference

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

Public Attributes

- std::pair< int, int > position
- float f
- float g
- float h

6.7.1 Member Data Documentation

6.7.1.1 f

```
float arcade::Node::f
```

6.7.1.2 g

```
float arcade::Node::g
```

6.7.1.3 h

```
float arcade::Node::h
```

6.7.1.4 position

```
std::pair<int, int> arcade::Node::position
```

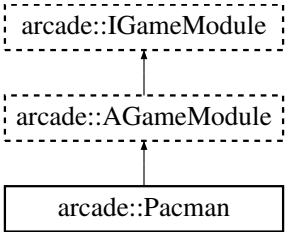
The documentation for this struct was generated from the following file:

- /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.hpp

6.8 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::vector< std::vector< arcade::entity > > moveEntities (std::vector< std::vector< arcade::entity > > layers)
- void handdleKeyEvents (arcade::KeyboardInput key)
- bool isOver (std::vector< std::vector< arcade::entity > > layers)
- void updateTimers (std::vector< std::vector< arcade::entity > > layers)
- bool isPacgumEaten (std::pair< int, int > pos, std::vector< std::vector< arcade::entity > > layers)
- bool isCoinEaten (std::pair< int, int > pos, std::vector< std::vector< arcade::entity > > layers)

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
- 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

Private Attributes

- int _levelSpeed = 0
- arcade::DataPacman _pacmanData
- std::vector< arcade::DataGhost > _ghostData

6.8.1 Constructor & Destructor Documentation

6.8.1.1 Pacman()

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

Construct a new arcade::Pacman::Pacman object.

6.8.1.2 ~Pacman()

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

Destroy the arcade::Pacman::Pacman object.

6.8.2 Member Function Documentation

6.8.2.1 handdleKeyEvents()

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

Implements arcade::AGameModule.

6.8.2.2 init()

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

Implements arcade::AGameModule.

6.8.2.3 isCoinEaten()

```
bool arcade::Pacman::isCoinEaten (
            std::pair< int, int > pos,
            std::vector< std::vector< arcade::entity > > layers )
```

6.8.2.4 isOver()

```
bool arcade::Pacman::isOver (
            std::vector< std::vector< arcade::entity > > layers )
```

6.8.2.5 isPacgumEaten()

```
bool arcade::Pacman::isPacgumEaten (
            std::pair< int, int > pos,
            std::vector< std::vector< arcade::entity > > layers )
```

6.8.2.6 moveEntities()

```
std::vector< std::vector< arcade::entity > > arcade::Pacman::moveEntities (
            std::vector< std::vector< arcade::entity > > layers )
```

6.8.2.7 updateGame()

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

update the game

Implements arcade::AGameModule.

6.8.2.8 updateTimers()

```
void arcade::Pacman::updateTimers (
            std::vector< std::vector< arcade::entity > > layers )
```

6.8.3 Member Data Documentation

6.8.3.1 _ghostData

```
std::vector<arcade::DataGhost> arcade::Pacman::_ghostData  [private]
```

6.8.3.2 _levelSpeed

```
int arcade::Pacman::_levelSpeed = 0  [private]
```

6.8.3.3 _pacmanData

```
arcade::DataPacman arcade::Pacman::_pacmanData  [private]
```

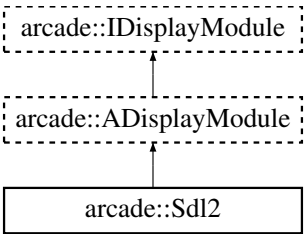
The documentation for this class was generated from the following files:

- /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.hpp
- /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.cpp

6.9 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

6.9.1 Constructor & Destructor Documentation

6.9.1.1 Sdl2()

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

6.9.1.2 ~Sdl2()

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

6.9.2 Member Function Documentation

6.9.2.1 clearWindow()

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

clear the window

Implements arcade::ADisplayModule.

6.9.2.2 displayWindow()

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

Implements arcade::ADisplayModule.

6.9.2.3 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

sprite	sprite to display
coordinates	coordinates of the sprite
width	width of the sprite
height	height of the sprite
rotation	rotation of the sprite

Implements arcade::ADisplayModule.

6.9.2.4 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

path	path to the sprite
x	x position of the sprite
y	y position of the sprite
width	width of the sprite
height	height of the sprite

Implements arcade::ADisplayModule.

6.9.2.5 drawText()

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

draw text on the window

Parameters

text	text to display
x	x position of the text
y	y position of the text
size	size of the text

Implements arcade::ADisplayModule.

6.9.2.6 getInput()

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

get the input from the window

Returns

arcade::KeyboardInput

Implements arcade::ADisplayModule.

6.9.2.7 getName()

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

get the name of the library

Returns

std::string

Implements arcade::ADisplayModule.

6.9.3 Member Data Documentation

6.9.3.1 _renderer

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

6.9.3.2 _window

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

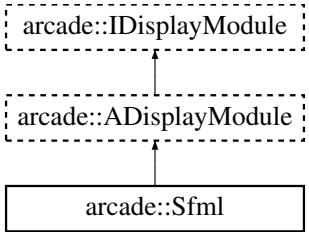
The documentation for this class was generated from the following files:

- /home/aleachlodnik/Arcade/lib/graphics/sdl2/Sdl2.hpp
- /home/aleachlodnik/Arcade/lib/graphics/sdl2/Sdl2.cpp

6.10 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

6.10.1 Constructor & Destructor Documentation

6.10.1.1 Sfml()

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

6.10.1.2 ~Sfml()

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

6.10.2 Member Function Documentation

6.10.2.1 clearWindow()

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

Implements arcade::ADisplayModule.

6.10.2.2 displayWindow()

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

Implements arcade::ADisplayModule.

6.10.2.3 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.

6.10.2.4 drawSprite()

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

Implements arcade::ADisplayModule.

6.10.2.5 drawText()

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

Implements arcade::ADisplayModule.

6.10.2.6 getInput()

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

Implements arcade::ADisplayModule.

6.10.2.7 getName()

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

Implements arcade::ADisplayModule.

6.10.3 Member Data Documentation

6.10.3.1 _texture

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

6.10.3.2 _window

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

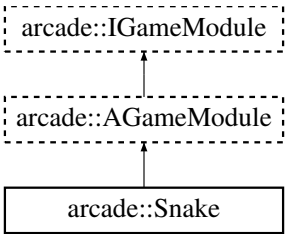
The documentation for this class was generated from the following files:

- /home/aleachlodnik/Arcade/lib/graphics/sfml/Sfml.hpp
- /home/aleachlodnik/Arcade/lib/graphics/sfml/Sfml.cpp

6.11 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.
- void init ()
- void updateGame ()

  update the game
- arcade::GameData moveSnake ()

  move the snake
- void handdleKeyEvents (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
- 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

6.11.1 Constructor & Destructor Documentation

6.11.1.1 Snake()

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

Construct a new arcade::Snake::Snake object.

6.11.1.2 ~Snake()

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

Destroy the arcade::Snake::Snake object.

6.11.2 Member Function Documentation

6.11.2.1 handdleKeyEvents()

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

handle key events

Parameters

key	

Implements arcade::AGameModule.

6.11.2.2 init()

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

Implements arcade::AGameModule.

6.11.2.3 moveSnake()

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

move the snake

Parameters

display_info	

6.11.2.4 updateGame()

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

update the game

Implements arcade::AGameModule.

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

- /home/aleachlodnik/Arcade/lib/games/snake/Snake.hpp
- /home/aleachlodnik/Arcade/lib/games/snake/Snake.cpp

Chapter 7

File Documentation

7.1 /home/aleachlodnik/Arcade/lib/games/AGameModule.cpp File Reference

```
#include ¨AGameModule.hpp¨
```

7.2 /home/aleachlodnik/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     void setGameStatus(GameStatus status);
00024     GameStatus getGameStatus() const;
00025
00026     void setCoreModule(arcade::CoreModule *coreModule);
00027     arcade::CoreModule *getCoreModule() const;
00028
00029     void setDirection(arcade::KeyboardInput direction);
00030     arcade::KeyboardInput getDirection() const;
00031
00032     virtual void init() = 0;
00033     virtual void handdleKeyEvents(arcade::KeyboardInput key) = 0;
00034
00035     int getLayerCell(int layer, int x, int y) const;
00036
00037   };
00038 }; // namespace arcade
00039
00040 #endif /* !IGAMEMODULE_HPP_ */
```

7.4 /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.cpp File Reference

```
#include ¨Pacman.hpp¨
```

Classes

- struct CompareNode

Functions

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

  generate entry point for the game library
- arcade::ModuleType getType ()
- std::string getName ()
- int manhattanDistance (const arcade::Node a, const arcade::Node b)
- bool isValid (const std::vector< std::vector< int > > map, arcade::Node p)
- std::vector< arcade::Node > getNeighbors (const std::vector< std::vector< int > > map, arcade::Node p)
- std::vector< std::vector< int > > layersToMap (std::vector< arcade::entity > layer)
- std::vector< arcade::Node > aStar (std::vector< std::vector< arcade::entity > > layers, arcade::Node start, arcade::Node end)
- arcade::Node pairToNode (std::pair< int, int > pair)

7.4.1 Function Documentation

7.4.1.1 aStar()

```
std::vector< arcade::Node > aStar (
            std::vector< std::vector< arcade::entity > > layers,
            arcade::Node start,
            arcade::Node end )
```

7.4.1.2 entryPoint()

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

generate entry point for the game library

7.4.1.3 getName()

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

7.4.1.4 getNeighbors()

```
std::vector< arcade::Node > getNeighbors (
            const std::vector< std::vector< int > > map,
            arcade::Node p )
```

7.4.1.5 getType()

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

7.4.1.6 isValid()

```
bool isValid (
            const std::vector< std::vector< int > > map,
            arcade::Node p )
```

7.4.1.7 layersToMap()

```
std::vector< std::vector< int > > layersToMap (
            std::vector< arcade::entity > layer )
```

7.4.1.8 manhattanDistance()

```
int manhattanDistance (
            const arcade::Node a,
            const arcade::Node b )
```

7.4.1.9 pairToNode()

```
arcade::Node pairToNode (
            std::pair< int, int > pair )
```

7.5 /home/aleachlodnik/Arcade/lib/games/pacman/Pacman.hpp File Reference

```
#include ¨../AGameModule.hpp¨
#include <queue>
#include <cmath>
#include <iostream>
#include <limits>
#include <algorithm>
#include <chrono>
```

Classes

- struct arcade::Node
- struct arcade::DataPacman
- struct arcade::DataGhost
- class arcade::Pacman

Namespaces

- namespace arcade

Macros

- #define SPEED_PACMAN 1

7.5.1 Macro Definition Documentation

7.5.1.1 SPEED_PACMAN

```
#define SPEED_PACMAN 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_PACMAN 1
00011
00012 #include "../AGameModule.hpp"
00013 #include <queue>
00014 #include <cmath>
00015 #include <iostream>
00016 #include <limits>
00017 #include <algorithm>
00018 #include <chrono>
```

```
00019
00020 namespace arcade {
00021
00022 struct Node {
00023   std::pair<int, int> position;
00024   float f, g, h;
00025 };
00026
00027 struct DataPacman {
00028   bool isBoosted;
00029   std::chrono::time_point<std::chrono::system_clock> _pacmanTimer;
00030 };
00031
00032 struct DataGhost {
00033   bool isScared;
00034   bool isDead;
00035   std::pair<int, int> initialPos;
00036   std::vector<arcade::Node> path;
00037   int actualPathIndex;
00038   std::chrono::time_point<std::chrono::system_clock> ghostTimer;
00039 };
00040
00041 class Pacman : virtual public arcade::AGameModule {
00042 public:
00043   Pacman();
00044   ~Pacman();
00045   void init();
00046   void updateGame();
00047   std::vector<std::vector<arcade::entity» moveEntities(std::vector<std::vector<arcade::entity»
      layers);
00048   void handdleKeyEvents(arcade::KeyboardInput key);
00049   bool isOver(std::vector<std::vector<arcade::entity» layers);
00050   void updateTimers(std::vector<std::vector<arcade::entity» layers);
00051   bool isPacgumEaten(std::pair<int, int> pos, std::vector<std::vector<arcade::entity» layers);
00052   bool isCoinEaten(std::pair<int, int> pos, std::vector<std::vector<arcade::entity» layers);
00053
00054 protected:
00055 private:
00056   int _levelSpeed = 0;
00057   arcade::DataPacman _pacmanData;
00058   std::vector<arcade::DataGhost> _ghostData;
00059 };
00060 }; // namespace arcade
00061
00062 #endif /* !PACMAN_HPP_ */
```

7.7 /home/aleachlodnik/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/aleachlodnik/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   void init();
00021   void updateGame();
00022   arcade::GameData moveSnake();
00023   void handdleKeyEvents(arcade::KeyboardInput key);
00024
00025 protected:
00026 };
00027 }; // namespace arcade
00028
00029 #endif /* !SNAKE_HPP_ */
```

7.10 /home/aleachlodnik/Arcade/lib/graphics/ADisplayModule.cpp File Reference

```
#include ¨ADisplayModule.hpp¨
```

7.11 /home/aleachlodnik/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) =
      0;
00030   virtual void drawAllSprite(std::pair<char, std::string> sprite, std::vector<std::pair<int, int»
      coordinates, int width, int height) = 0;
00031   virtual void drawText(const std::string text, int x, int y, int size) = 0;
00032
00033 protected:
00034   arcade::KeyboardInput _input;
00035   arcade::CoreModule *_coreModule;
00036 };
00037 }; // namespace arcade
00038
00039 #endif /* !ADISPLAYMODULE_HPP_ */
```

7.13 /home/aleachlodnik/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/aleachlodnik/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,
      int width, int height);
00026
00027 protected:
00028   WINDOW *_window;
00029 };
00030 }; // namespace arcade
00031
00032 #endif /* !NCURSES_HPP_ */
```

7.16 /home/aleachlodnik/Arcade/lib/graphics/sdl2/Sdl2.cpp File Reference

```
#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/aleachlodnik/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,
      int width, int height);
00028
00029 protected:
00030   SDL_Renderer *_renderer;
00031   SDL_Window *_window;
00032 private:
00033 };
00034 }; // namespace arcade
00035
00036 #endif /* !SDL2_HPP_ */
```

7.19 /home/aleachlodnik/Arcade/lib/graphics/sfml/Sfml.cpp File Reference

```
#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/aleachlodnik/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,
      int width, int height);
00028
00029 protected:
00030   sf::RenderWindow *_window;
00031   sf::Texture _texture;
00032 };
00033 }; // namespace arcade
00034
00035 #endif /* !SFML_HPP_ */
```

7.22 /home/aleachlodnik/Arcade/src/CoreModule.cpp File Reference

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

7.23 /home/aleachlodnik/Arcade/src/ErrorHandling.cpp File Reference

```
#include <dlfcn.h>
#include <iostream>
#include <ErrorHandling.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/aleachlodnik/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

argc	
argv	

Returns

int

Index