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

namespace for the arcade project

Classes

- class ADisplayModule
- class AGameModule
- class CoreModule
- class IModule

    Interface for the modules.
- class NCurses
- class Pacman
- class Sdl2
- class Sfml
- class Snake

5.1.1 Detailed Description

namespace for the arcade project

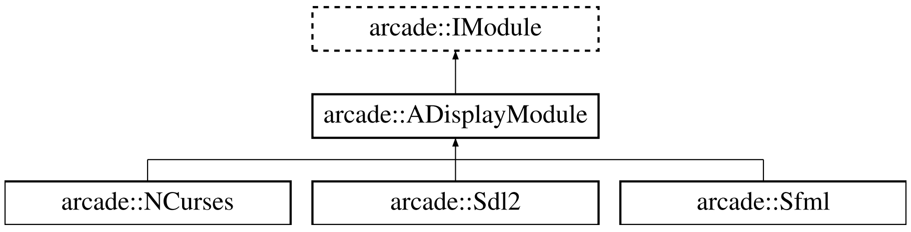
Chapter 6

Class Documentation

6.1 arcade::ADisplayModule Class Reference

`#include <ADisplayModule.hpp>`

Inheritance diagram for arcade::ADisplayModule:



Public Types

- enum DisplayStatus {
  RUNNING , PAUSED , SELECTION , GAMEOVER ,
  WIN }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

Public Member Functions

- ADisplayModule ()

  Construct a new arcade::ADisplayModule::ADisplayModule object.
- ∼ADisplayModule ()

  Destroy the arcade::ADisplayModule::ADisplayModule object.
- virtual void init ()=0
- virtual void stop ()=0
- virtual void display ()=0
- void setDisplayStatus (DisplayStatus status)

  set the status of the display module
- DisplayStatus getDisplayStatus () const

  return the status of the display module
- virtual const arcade::IModule::LibName getName () const =0
- const arcade::IModule::ModuleType getType () const

  return the type of the module
- arcade::IModule::KeyboardInput getInput () const

  get input from the user
- void sendGameData (arcade::IModule::GameData data)

  receive send data of the game module to the display module

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Protected Attributes

- void ∗ _window
- void ∗ _texture
- void ∗ _event
- arcade::IModule::GameData _gameData
- arcade::IModule::KeyboardInput _input
- DisplayStatus _displayStatus

6.1.1 Member Enumeration Documentation

6.1.1.1 DisplayStatus

```
enum arcade::ADisplayModule::DisplayStatus
```

Enumerator

RUNNING	
PAUSED	
SELECTION	
GAMEOVER	
WIN	

6.1.2 Constructor & Destructor Documentation

6.1.2.1 ADisplayModule()

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

Construct a new arcade::ADisplayModule::ADisplayModule object.

6.1.2.2 ∼ADisplayModule()

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

Destroy the arcade::ADisplayModule::ADisplayModule object.

6.1.3 Member Function Documentation

6.1.3.1 display()

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

Implemented in arcade::NCurses, arcade::Sdl2, and arcade::Sfml.

6.1.3.2 getDisplayStatus()

```
arcade::ADisplayModule::DisplayStatus arcade::ADisplayModule::getDisplayStatus ( ) const
```

return the status of the display module

Returns

arcade::ADisplayModule::DisplayStatus status of the display module

6.1.3.3 getInput()

```
arcade::IModule::KeyboardInput arcade::ADisplayModule::getInput ( ) const
```

get input from the user

Returns

arcade::IModule::KeyboardInput

6.1.3.4 getName()

```
virtual const arcade::IModule::LibName arcade::ADisplayModule::getName ( ) const [pure virtual]
```

Implements arcade::IModule.

Implemented in arcade::NCurses, arcade::Sdl2, and arcade::Sfml.

6.1.3.5 getType()

```
const arcade::IModule::ModuleType arcade::ADisplayModule::getType ( ) const  [virtual]
```

return the type of the module

Returns

const arcade::IModule::ModuleType

Implements arcade::IModule.

6.1.3.6 init()

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

Implements arcade::IModule.

Implemented in arcade::NCurses, arcade::Sdl2, and arcade::Sfml.

6.1.3.7 sendGameData()

```
void arcade::ADisplayModule::sendGameData (
             arcade::IModule::GameData data )
```

receive send data of the game module to the display module

Parameters

data	of the game module (score, map, player position)

6.1.3.8 setDisplayStatus()

```
void arcade::ADisplayModule::setDisplayStatus (
             DisplayStatus status )
```

set the status of the display module

Parameters

status	of the display module

6.1.3.9 stop()

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

Implements arcade::IModule.

Implemented in arcade::NCurses, arcade::Sdl2, and arcade::Sfml.

6.1.4 Member Data Documentation

6.1.4.1 _displayStatus

```
DisplayStatus arcade::ADisplayModule::_displayStatus  [protected]
```

6.1.4.2 _event

```
void* arcade::ADisplayModule::_event  [protected]
```

6.1.4.3 _gameData

```
arcade::IModule::GameData arcade::ADisplayModule::_gameData  [protected]
```

6.1.4.4 _input

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

6.1.4.5 _texture

```
void* arcade::ADisplayModule::_texture  [protected]
```

6.1.4.6 _window

```
void* arcade::ADisplayModule::_window  [protected]
```

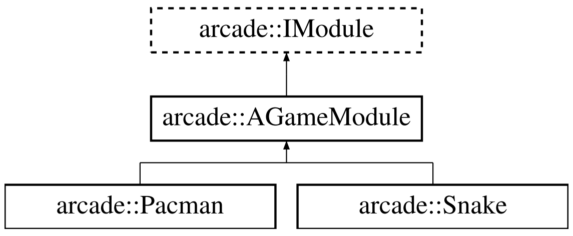
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/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 Types

- enum GameStatus { RUNNING , PAUSED , GAMEOVER , WIN }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

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 init ()=0
- virtual void stop ()=0
- void setGameStatus (GameStatus status)
- GameStatus getDisplayStatus () const

  get the status of the game
- virtual const arcade::IModule::LibName getName () const =0
- const arcade::IModule::ModuleType getType () const

  get the name of the game library
- void sendInput (arcade::IModule::KeyboardInput input)

  receive input from the graphic module
- arcade::IModule::GameData sendGameData ()

  send the game data to the graphic module

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Protected Attributes

- arcade::IModule::GameData _gameData
- arcade::IModule::KeyboardInput _input
- GameStatus _gameStatus

6.2.1 Member Enumeration Documentation

6.2.1.1 GameStatus

```
enum arcade::AGameModule::GameStatus
```

Enumerator

RUNNING	
PAUSED	
GAMEOVER	
WIN	

6.2.2 Constructor & Destructor Documentation

6.2.2.1 AGameModule()

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

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

6.2.2.2 ∼AGameModule()

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

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

6.2.3 Member Function Documentation

6.2.3.1 getDisplayStatus()

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

get the status of the game

Returns

arcade::AGameModule::GameStatus

6.2.3.2 getName()

```
virtual const arcade::IModule::LibName arcade::AGameModule::getName ( ) const  [pure virtual]
```

Implements arcade::IModule.

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

6.2.3.3 getType()

```
const arcade::IModule::ModuleType arcade::AGameModule::getType ( ) const  [virtual]
```

get the name of the game library

Returns

const arcade::IModule::ModuleType

Implements arcade::IModule.

6.2.3.4 init()

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

Implements arcade::IModule.

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

6.2.3.5 sendGameData()

```
arcade::IModule::GameData arcade::AGameModule::sendGameData ( )
```

send the game data to the graphic module

Returns

arcade::IModule::GameData

6.2.3.6 sendInput()

```
void arcade::AGameModule::sendInput (
            arcade::IModule::KeyboardInput input )
```

receive input from the graphic module

Parameters

input	KeyboardInput

6.2.3.7 setGameStatus()

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

6.2.3.8 stop()

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

Implements arcade::IModule.

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

6.2.4 Member Data Documentation

6.2.4.1 _gameData

```
arcade::IModule::GameData arcade::AGameModule::_gameData  [protected]
```

6.2.4.2 _gameStatus

```
GameStatus arcade::AGameModule::_gameStatus  [protected]
```

6.2.4.3 _input

```
arcade::IModule::KeyboardInput arcade::AGameModule::_input  [protected]
```

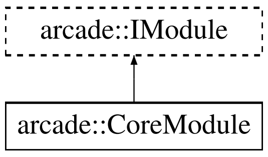
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/game/AGameModule.hpp
- /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGameModule.cpp

6.3 arcade::CoreModule Class Reference

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

Inheritance diagram for arcade::CoreModule:



Public Types

- enum CoreStatus { RUNNING , SELECTION }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

Public Member Functions

- CoreModule ()

  Construct a new arcade::Core Module::Core Module object.
- ∼CoreModule ()

  Destroy the arcade::Core Module::Core Module object.
- void init ()

  load the libraries in the given path
- void stop ()

  stop the core module
- const LibName getName () const

  get the name of the library
- const ModuleType getType () const

  get the type of the library
- void setCoreStatus (CoreStatus status)

  get the status of the core module
- CoreStatus getCoreStatus () const

  get the status of the core module
- std::unique_ptr< ADisplayModule > getDisplayModule ()

  get the display module
- std::unique_ptr< AGameModule > getGameModule ()

  get the game module
- void setModule (arcade::IModule::LibName name, arcade::IModule::ModuleType type)

  set graphic or game module to the core module
- std::vector< std::string > getLib (std::string pathLib)

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Protected Attributes

- CoreStatus _coreStatus
- std::unique_ptr< arcade::ADisplayModule > _displayModule
- std::unique_ptr< arcade::AGameModule > _gameModule

6.3.1 Member Enumeration Documentation

6.3.1.1 CoreStatus

```
enum arcade::CoreModule::CoreStatus
```

Enumerator

RUNNING	
SELECTION	

6.3.2 Constructor & Destructor Documentation

6.3.2.1 CoreModule()

```
arcade::CoreModule::CoreModule ( )
```

Construct a new arcade::Core Module::Core Module object.

6.3.2.2 ∼CoreModule()

```
arcade::CoreModule::∼CoreModule ( )
```

Destroy the arcade::Core Module::Core Module object.

6.3.3 Member Function Documentation

6.3.3.1 getCoreStatus()

```
arcade::CoreModule::CoreStatus arcade::CoreModule::getCoreStatus ( ) const
```

get the status of the core module

Returns

CoreStatus

6.3.3.2 getDisplayModule()

```
std::unique_ptr< arcade::ADisplayModule > arcade::CoreModule::getDisplayModule ( )
```

get the display module

Returns

std::unique_ptr<arcade::ADisplayModule>

6.3.3.3 getGameModule()

```
std::unique_ptr< arcade::AGameModule > arcade::CoreModule::getGameModule ( )
```

get the game module

Returns

std::unique_ptr<arcade::AGameModule>

6.3.3.4 getLib()

```
std::vector< std::string > arcade::CoreModule::getLib (
            std::string pathLib )
```

6.3.3.5 getName()

```
const arcade::IModule::LibName arcade::CoreModule::getName ( ) const  [virtual]
```

get the name of the library

Returns

const arcade::IModule::LibName

Implements arcade::IModule.

6.3.3.6 getType()

```
const arcade::IModule::ModuleType arcade::CoreModule::getType ( ) const  [virtual]
```

get the type of the library

Returns

const arcade::IModule::ModuleType

Implements arcade::IModule.

6.3.3.7 init()

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

load the libraries in the given path

Parameters

pathLib	path to the libraries

Implements arcade::IModule.

6.3.3.8 setCoreStatus()

```
void arcade::CoreModule::setCoreStatus (
            CoreStatus status )
```

get the status of the core module

Returns

CoreStatus

6.3.3.9 setModule()

```
void arcade::CoreModule::setModule (
            arcade::IModule::LibName name,
            arcade::IModule::ModuleType type )
```

set graphic or game module to the core module

Parameters

name	of the module (snake, pacman, ncurses, sdl2)
type	of the module (graphic or game)

6.3.3.10 stop()

```
void arcade::CoreModule::stop ( )  [virtual]
```

stop the core module

Implements arcade::IModule.

6.3.4 Member Data Documentation

6.3.4.1 _coreStatus

```
CoreStatus arcade::CoreModule::_coreStatus  [protected]
```

6.3.4.2 _displayModule

```
std::unique_ptr<arcade::ADisplayModule> arcade::CoreModule::_displayModule  [protected]
```

6.3.4.3 _gameModule

```
std::unique_ptr<arcade::AGameModule> arcade::CoreModule::_gameModule  [protected]
```

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

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/CoreModule.hpp
- /home/tmendy/Documents/Tek2/OOP/Arcade/lib/CoreModule.cpp

6.4 DLLoader< T > Class Template Reference

```
#include <DLLoader.hpp>
```

Public Member Functions

- DLLoader (const std::string &libPath)
- ∼DLLoader ()
- T ∗ getInstance (const std::string &funcName)

Private Attributes

- void ∗ handle

6.4.1 Constructor & Destructor Documentation

6.4.1.1 DLLoader()

```
template<typename T >
DLLoader< T >::DLLoader (
            const std::string & libPath )  [inline]
```

6.4.1.2 ∼DLLoader()

```
template<typename T >
DLLoader< T >::∼DLLoader ( )  [inline]
```

6.4.2 Member Function Documentation

6.4.2.1 getInstance()

```
template<typename T >
T * DLLoader< T >::getInstance (
            const std::string & funcName )  [inline]
```

6.4.3 Member Data Documentation

6.4.3.1 handle

```
template<typename T >
void* DLLoader< T >::handle  [private]
```

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

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/DLLoader.hpp

6.5 arcade::IModule::GameData Struct Reference

information about the game from the game module to the graphic module

```
#include <IModule.hpp>
```

Public Attributes

- std::vector< std::vector< int > > display_info
- std::map< unsigned int, std::string > sprite_value

6.5.1 Detailed Description

information about the game from the game module to the graphic module

6.5.2 Member Data Documentation

6.5.2.1 display_info

```
std::vector<std::vector<int> > arcade::IModule::GameData::display_info
```

6.5.2.2 sprite_value

```
std::map<unsigned int, std::string> arcade::IModule::GameData::sprite_value
```

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

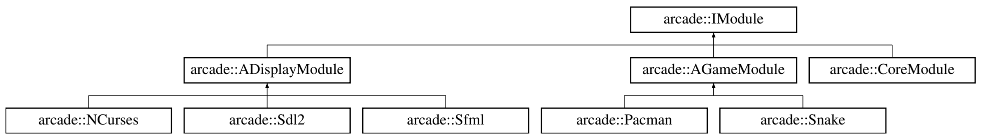
- /home/tmendy/Documents/Tek2/OOP/Arcade/include/IModule.hpp

6.6 arcade::IModule Class Reference

Interface for the modules.

```
#include <IModule.hpp>
```

Inheritance diagram for arcade::IModule:



Classes

- struct GameData

  information about the game from the game module to the graphic module

Public Types

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

Public Member Functions

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

6.6.1 Detailed Description

Interface for the modules.

6.6.2 Member Enumeration Documentation

6.6.2.1 KeyboardInput

```
enum arcade::IModule::KeyboardInput
```

all the possible keyboard inputs

Enumerator

UP	
DOWN	
LEFT	
RIGHT	
SPACE	
ENTER	
ESCAPE	
A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
NONE	

6.6.2.2 LibName

```
enum arcade::IModule::LibName
```

all the possible library names

Enumerator

SNAKE	
NIBBLER	
PACMAN	
QIX	
CENTIPEDE	
SOLARFOX	
NCURSES	
SDL	
SFML	
OPENGL	
UNKNOWN	

6.6.2.3 ModuleType

```
enum arcade::IModule::ModuleType
```

all the possible module types

Enumerator

GRAPHIC	
GAME	
CORE	

6.6.3 Constructor & Destructor Documentation

6.6.3.1 IModule()

```
arcade::IModule::IModule ( )  [inline]
```

6.6.3.2 ∼IModule()

```
virtual arcade::IModule::∼IModule ( )  [inline], [virtual]
```

6.6.4 Member Function Documentation

6.6.4.1 getName()

```
virtual const LibName arcade::IModule::getName ( ) const  [pure virtual]
```

Implemented in arcade::CoreModule, arcade::Pacman, arcade::Snake, arcade::NCurses, arcade::Sdl2, arcade::Sfml, arcade::AGameModule, and arcade::ADisplayModule.

6.6.4.2 getType()

```
virtual const ModuleType arcade::IModule::getType ( ) const  [pure virtual]
```

Implemented in arcade::CoreModule, arcade::AGameModule, and arcade::ADisplayModule.

6.6.4.3 init()

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

Implemented in arcade::CoreModule, arcade::Pacman, arcade::Snake, arcade::NCurses, arcade::Sdl2, arcade::Sfml, arcade::AGameModule, and arcade::ADisplayModule.

6.6.4.4 stop()

```
virtual void arcade::IModule::stop ( )  [pure virtual]
```

Implemented in arcade::CoreModule, arcade::Pacman, arcade::Snake, arcade::NCurses, arcade::Sdl2, arcade::Sfml, arcade::AGameModule, and arcade::ADisplayModule.

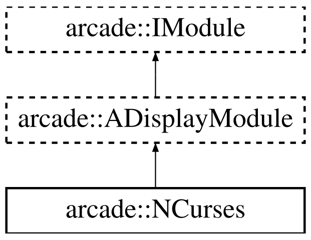
The documentation for this class was generated from the following file:

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/IModule.hpp

6.7 arcade::NCurses Class Reference

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

Inheritance diagram for arcade::NCurses:



Public Member Functions

- NCurses ()
- ∼NCurses ()
- void init ()
- void stop ()
- void display ()
- const arcade::IModule::LibName getName () const

Public Member Functions inherited from arcade::ADisplayModule

- ADisplayModule ()

  Construct a new arcade::ADisplayModule::ADisplayModule object.
- ∼ADisplayModule ()

  Destroy the arcade::ADisplayModule::ADisplayModule object.
- virtual void init ()=0
- virtual void stop ()=0
- virtual void display ()=0
- void setDisplayStatus (DisplayStatus status)

  set the status of the display module
- DisplayStatus getDisplayStatus () const

  return the status of the display module
- virtual const arcade::IModule::LibName getName () const =0
- const arcade::IModule::ModuleType getType () const

  return the type of the module
- arcade::IModule::KeyboardInput getInput () const

  get input from the user
- void sendGameData (arcade::IModule::GameData data)

  receive send data of the game module to the display module

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Additional Inherited Members

Public Types inherited from arcade::ADisplayModule

- enum DisplayStatus {
  RUNNING , PAUSED , SELECTION , GAMEOVER ,
  WIN }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs

- enum ModuleType { GRAPHIC , GAME , CORE }

    all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

    all the possible library names

Protected Attributes inherited from arcade::ADisplayModule

- void ∗ _window
- void ∗ _texture
- void ∗ _event
- arcade::IModule::GameData _gameData
- arcade::IModule::KeyboardInput _input
- DisplayStatus _displayStatus

6.7.1 Constructor & Destructor Documentation

6.7.1.1 NCurses()

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

6.7.1.2 ∼NCurses()

```
arcade::NCurses::∼NCurses ( )
```

6.7.2 Member Function Documentation

6.7.2.1 display()

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

Implements arcade::ADisplayModule.

6.7.2.2 getName()

```
const arcade::IModule::LibName arcade::NCurses::getName ( ) const [virtual]
```

Implements arcade::ADisplayModule.

6.7.2.3 init()

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

Implements arcade::ADisplayModule.

6.7.2.4 stop()

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

Implements arcade::ADisplayModule.

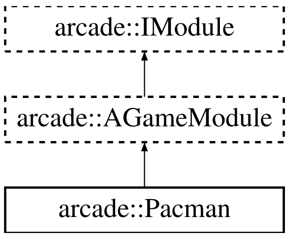
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/NCurses.hpp
- /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/NCurses.cpp

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 stop ()
- const arcade::IModule::LibName getName () const

  return the name of the game

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 init ()=0
- virtual void stop ()=0
- void setGameStatus (GameStatus status)
- GameStatus getDisplayStatus () const

  get the status of the game
- virtual const arcade::IModule::LibName getName () const =0
- const arcade::IModule::ModuleType getType () const

  get the name of the game library
- void sendInput (arcade::IModule::KeyboardInput input)

  receive input from the graphic module
- arcade::IModule::GameData sendGameData ()

  send the game data to the graphic module

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Additional Inherited Members

Public Types inherited from arcade::AGameModule

- enum GameStatus { RUNNING , PAUSED , GAMEOVER , WIN }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

Protected Attributes inherited from arcade::AGameModule

- arcade::IModule::GameData _gameData
- arcade::IModule::KeyboardInput _input
- GameStatus _gameStatus

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 getName()

```
const arcade::IModule::LibName arcade::Pacman::getName ( ) const  [virtual]
```

return the name of the game

Returns

const arcade::IModule::LibName

Implements arcade::AGameModule.

6.8.2.2 init()

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

Implements arcade::AGameModule.

6.8.2.3 stop()

```
void arcade::Pacman::stop ( )  [inline], [virtual]
```

Implements arcade::AGameModule.

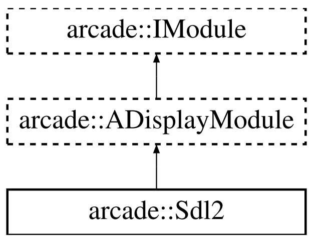
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/game/Pacman.hpp
- /home/tmendy/Documents/Tek2/OOP/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 ()
- void init ()

  initailize the SDL2 module and create a window
- void stop ()

  stop the SDL2 module and destroy the window
- void display ()

  display information on the window
- const arcade::IModule::LibName getName () const

  return the name of the module

Public Member Functions inherited from arcade::ADisplayModule

- ADisplayModule ()

  Construct a new arcade::ADisplayModule::ADisplayModule object.
- ∼ADisplayModule ()

  Destroy the arcade::ADisplayModule::ADisplayModule object.
- virtual void init ()=0
- virtual void stop ()=0
- virtual void display ()=0
- void setDisplayStatus (DisplayStatus status)

  set the status of the display module
- DisplayStatus getDisplayStatus () const

  return the status of the display module
- virtual const arcade::IModule::LibName getName () const =0
- const arcade::IModule::ModuleType getType () const

  return the type of the module
- arcade::IModule::KeyboardInput getInput () const

  get input from the user
- void sendGameData (arcade::IModule::GameData data)

  receive send data of the game module to the display module

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Additional Inherited Members

Public Types inherited from arcade::ADisplayModule

- enum DisplayStatus {
  RUNNING , PAUSED , SELECTION , GAMEOVER ,
  WIN }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

Protected Attributes inherited from arcade::ADisplayModule

- void ∗ _window
- void ∗ _texture
- void ∗ _event
- arcade::IModule::GameData _gameData
- arcade::IModule::KeyboardInput _input
- DisplayStatus _displayStatus

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 display()

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

display information on the window

Implements arcade::ADisplayModule.

6.9.2.2 getName()

```
const arcade::IModule::LibName arcade::Sdl2::getName ( ) const  [virtual]
```

return the name of the module

Returns

const arcade::IModule::LibName

Implements arcade::ADisplayModule.

6.9.2.3 init()

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

initailize the SDL2 module and create a window

Implements arcade::ADisplayModule.

6.9.2.4 stop()

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

stop the SDL2 module and destroy the window

Implements arcade::ADisplayModule.

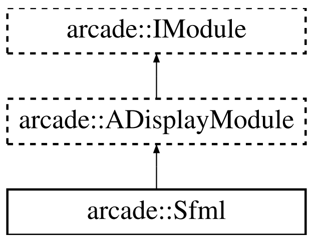
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/Sdl2.hpp
- /home/tmendy/Documents/Tek2/OOP/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 ()
- arcade::IModule::KeyboardInput getInput ()
- void init ()
- void stop ()
- void display ()
- const arcade::IModule::LibName getName () const

Public Member Functions inherited from arcade::ADisplayModule

- ADisplayModule ()

  Construct a new arcade::ADisplayModule::ADisplayModule object.
- ∼ADisplayModule ()

  Destroy the arcade::ADisplayModule::ADisplayModule object.
- virtual void init ()=0
- virtual void stop ()=0
- virtual void display ()=0
- void setDisplayStatus (DisplayStatus status)

  set the status of the display module
- DisplayStatus getDisplayStatus () const

  return the status of the display module
- virtual const arcade::IModule::LibName getName () const =0
- const arcade::IModule::ModuleType getType () const

  return the type of the module
- arcade::IModule::KeyboardInput getInput () const

  get input from the user
- void sendGameData (arcade::IModule::GameData data)

  receive send data of the game module to the display module

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Private Attributes

- sf::Texture _texture
- sf::Font _font

Additional Inherited Members

Public Types inherited from arcade::ADisplayModule

- enum DisplayStatus {
  RUNNING , PAUSED , SELECTION , GAMEOVER ,
  WIN }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

Protected Attributes inherited from arcade::ADisplayModule

- void ∗ _window
- void ∗ _texture
- void ∗ _event
- arcade::IModule::GameData _gameData
- arcade::IModule::KeyboardInput _input
- DisplayStatus _displayStatus

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 display()

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

Implements arcade::ADisplayModule.

6.10.2.2 getInput()

`arcade::IModule::KeyboardInput arcade::Sfml::getInput ( )`

6.10.2.3 getName()

`const arcade::IModule::LibName arcade::Sfml::getName ( ) const [virtual]`

Implements arcade::ADisplayModule.

6.10.2.4 init()

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

Implements arcade::ADisplayModule.

6.10.2.5 stop()

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

Implements arcade::ADisplayModule.

6.10.3 Member Data Documentation

6.10.3.1 _font

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

6.10.3.2 _texture

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

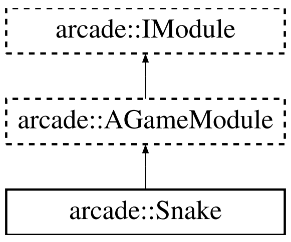
The documentation for this class was generated from the following files:

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/Sfml.hpp
- /home/tmendy/Documents/Tek2/OOP/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 ()

    init the game
- void stop ()

    stop the game
- const arcade::IModule::LibName getName () const

    return the name of the game library

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 init ()=0
- virtual void stop ()=0
- void setGameStatus (GameStatus status)
- GameStatus getDisplayStatus () const

    get the status of the game
- virtual const arcade::IModule::LibName getName () const =0
- const arcade::IModule::ModuleType getType () const

    get the name of the game library
- void sendInput (arcade::IModule::KeyboardInput input)

    receive input from the graphic module
- arcade::IModule::GameData sendGameData ()

    send the game data to the graphic module

Public Member Functions inherited from arcade::IModule

- IModule ()
- virtual ∼IModule ()
- virtual void init ()=0
- virtual void stop ()=0
- virtual const LibName getName () const =0
- virtual const ModuleType getType () const =0

Additional Inherited Members

Public Types inherited from arcade::AGameModule

- enum GameStatus { RUNNING , PAUSED , GAMEOVER , WIN }

Public Types inherited from arcade::IModule

- enum KeyboardInput {
  UP , DOWN , LEFT , RIGHT ,
  SPACE , ENTER , ESCAPE , A ,
  B , C , D , E ,
  F , G , H , I ,
  J , K , L , M ,
  N , O , P , Q ,
  R , S , T , U ,
  V , W , X , Y ,
  Z , NONE }

  all the possible keyboard inputs
- enum ModuleType { GRAPHIC , GAME , CORE }

  all the possible module types
- enum LibName {
  SNAKE , NIBBLER , PACMAN , QIX ,
  CENTIPEDE , SOLARFOX , NCURSES , SDL ,
  SFML , OPENGL , UNKNOWN }

  all the possible library names

Protected Attributes inherited from arcade::AGameModule

- arcade::IModule::GameData _gameData
- arcade::IModule::KeyboardInput _input
- GameStatus _gameStatus

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 getName()

```
const arcade::IModule::LibName arcade::Snake::getName ( ) const  [virtual]
```

return the name of the game library

Returns

const arcade::IModule::LibName

Implements arcade::AGameModule.

6.11.2.2 init()

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

init the game

Implements arcade::AGameModule.

6.11.2.3 stop()

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

stop the game

Implements arcade::AGameModule.

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

- /home/tmendy/Documents/Tek2/OOP/Arcade/include/game/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/include/Core↵ Module.hpp File Reference

```
#include "ADisplayModule.hpp"
#include "AGameModule.hpp"
#include "IModule.hpp"
#include <memory>
#include <iostream>
#include <dirent.h>
```

Classes

- class arcade::CoreModule

Namespaces

- namespace arcade

  namespace for the arcade project

7.2 CoreModule.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** CoreModule
*/

#ifndef COREMODULE_HPP_
#define COREMODULE_HPP_

#include "ADisplayModule.hpp"
#include "AGameModule.hpp"
#include "IModule.hpp"
#include <memory>
#include <iostream>
#include <dirent.h>
```

```

namespace arcade {
class CoreModule : virtual public arcade::IModule {
public:
  CoreModule();
  ~CoreModule();
  void init();
  void stop();
  const LibName getName() const;
  const ModuleType getType() const;
  enum CoreStatus { RUNNING, SELECTION };
  void setCoreStatus(CoreStatus status);
  CoreStatus getCoreStatus() const;
  std::unique_ptr<ADisplayModule> getDisplayModule();
  std::unique_ptr<AGameModule> getGameModule();
  void setModule(arcade::IModule::LibName name,
                 arcade::IModule::ModuleType type);
  std::vector<std::string> getLib(std::string pathLib);

protected:
  CoreStatus _coreStatus;
  std::unique_ptr<arcade::ADisplayModule> _displayModule;
  std::unique_ptr<arcade::AGameModule> _gameModule;
};
}; // namespace arcade

#endif /* !COREMODULE_HPP_ */
```

7.3 /home/tmendy/Documents/Tek2/OOP/Arcade/include/DLLoader.hpp File Reference

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

Classes

- class DLLoader< T >

7.4 DLLoader.hpp

Go to the documentation of this file.

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

template <typename T>
class DLLoader {
private:
  void *handle;

public:
  DLLoader(const std::string &libPath)
  {
    handle = dlopen(libPath.c_str(), RTLD_LAZY);
    if (!handle) {
      std::cerr « dlerror() « std::endl;
      exit(1);
    }
  }

  ~DLLoader()
  {
    if (handle)
      dlclose(handle);
  }

  T *getInstance(const std::string &funcName)
  {
    void *sym = dlsym(handle, funcName.c_str());
    if (!sym) {
      std::cerr « dlerror() « std::endl;
      exit(1);
    }
    return reinterpret_cast<T *(*)()>(sym)();
  }
};
```

7.5 /home/tmendy/Documents/Tek2/OOP/Arcade/include/game/AGame↵Module.hpp File Reference

```
#include ¨IModule.hpp¨
```

Classes

- class arcade::AGameModule

Namespaces

- namespace arcade

  namespace for the arcade project

7.6 AGameModule.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** AGameModule
*/

#ifndef AGAMEMODULE_HPP_
#define AGAMEMODULE_HPP_

#include "IModule.hpp"

namespace arcade {
class AGameModule : virtual public arcade::IModule {
public:
  enum GameStatus { RUNNING, PAUSED, GAMEOVER, WIN };
  AGameModule();
  ~AGameModule();
  virtual void init() = 0;
  virtual void stop() = 0;

  void setGameStatus(GameStatus status);
  GameStatus getDisplayStatus() const;

  virtual const arcade::IModule::LibName getName() const = 0;
  const arcade::IModule::ModuleType getType() const;

  void sendInput(arcade::IModule::KeyboardInput input);
  arcade::IModule::GameData sendGameData();

protected:
  arcade::IModule::GameData _gameData;
  arcade::IModule::KeyboardInput _input;
  GameStatus _gameStatus;
};
}; // namespace arcade

#endif /* !IGAMEMODULE_HPP_ */
```

7.7 /home/tmendy/Documents/Tek2/OOP/Arcade/include/game/←Pacman.hpp File Reference

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

Classes

- class arcade::Pacman

Namespaces

- namespace arcade

  namespace for the arcade project

7.8 Pacman.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** Pacman
*/

#ifndef PACMAN_HPP_
#define PACMAN_HPP_

#include "AGameModule.hpp"

namespace arcade {
class Pacman : virtual public arcade::AGameModule {
public:
  Pacman();
  ~Pacman();
  void init(){};
  void stop(){};
  const arcade::IModule::LibName getName() const;

protected:
private:
};
}; // namespace arcade

#endif /* !PACMAN_HPP_ */
```

7.9 /home/tmendy/Documents/Tek2/OOP/Arcade/include/game/←Snake.hpp File Reference

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

Classes

- class arcade::Snake

Namespaces

- namespace arcade

  namespace for the arcade project

7.10 Snake.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** Snake
*/

#ifndef SNAKE_HPP_
#define SNAKE_HPP_

#include "AGameModule.hpp"

namespace arcade {
class Snake : virtual public arcade::AGameModule {
public:
  Snake();
  ~Snake();
  void init();
  void stop();
  const arcade::IModule::LibName getName() const;

protected:
private:
};
}; // namespace arcade

#endif /* !SNAKE_HPP_ */
```

7.11 /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/↩ ADisplayModule.hpp File Reference

```
#include "IModule.hpp"
```

Classes

- class arcade::ADisplayModule

Namespaces

- namespace arcade

  namespace for the arcade project

7.12 ADisplayModule.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** bsArcade
** File description:
** ADisplayModule
*/

#ifndef ADISPLAYMODULE_HPP_
#define ADISPLAYMODULE_HPP_

#include "IModule.hpp"

namespace arcade {
class ADisplayModule : virtual public arcade::IModule {
public:
  enum DisplayStatus { RUNNING, PAUSED, SELECTION, GAMEOVER, WIN };
  ADisplayModule();
  ~ADisplayModule();
  virtual void init() = 0;
  virtual void stop() = 0;

  virtual void display() = 0;
  void setDisplayStatus(DisplayStatus status);
  DisplayStatus getDisplayStatus() const;

  virtual const arcade::IModule::LibName getName() const = 0;
  const arcade::IModule::ModuleType getType() const;

  arcade::IModule::KeyboardInput getInput() const;
  void sendGameData(arcade::IModule::GameData data);

protected:
  void *_window;
  void *_texture;
  void *_event;
  arcade::IModule::GameData _gameData;
  arcade::IModule::KeyboardInput _input;
  DisplayStatus _displayStatus;
};
}; // namespace arcade

#endif /* !ADISPLAYMODULE_HPP_ */
```

7.13 /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/↵ NCurses.hpp File Reference

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

Classes

- class arcade::NCurses

Namespaces

- namespace arcade

  namespace for the arcade project

7.14 NCurses.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** NCurses
*/

#ifndef NCURSES_HPP_
#define NCURSES_HPP_

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

namespace arcade {
class NCurses : virtual public arcade::ADisplayModule {
public:
  NCurses();
  ~NCurses();
  void init();
  void stop();
  void display();
  const arcade::IModule::LibName getName() const;

protected:
private:
};
}; // namespace arcade

#endif /* !NCURSES_HPP_ */
```

7.15 /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/↩ Sdl2.hpp File Reference

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

Classes

- class arcade::Sdl2

Namespaces

- namespace arcade

  namespace for the arcade project

7.16 Sdl2.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** SDL2
*/

#ifndef SDL2_HPP_
#define SDL2_HPP_

```

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

namespace arcade {
class Sdl2 : virtual public arcade::ADisplayModule {
public:
  Sdl2();
  ~Sdl2();
  void init();
  void stop();
  void display();
  const arcade::IModule::LibName getName() const;

protected:
private:
};
}; // namespace arcade

#endif /* !SDL2_HPP_ */
```

7.17 /home/tmendy/Documents/Tek2/OOP/Arcade/include/graphic/↩ Sfml.hpp File Reference

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

Classes

- class arcade::Sfml

Namespaces

- namespace arcade

  namespace for the arcade project

7.18 Sfml.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** SFML
*/

#ifndef SFML_HPP_
#define SFML_HPP_

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

namespace arcade {
class Sfml : virtual public arcade::ADisplayModule {
public:
  Sfml();
  ~Sfml();

  arcade::IModule::KeyboardInput getInput();
  void init();
  void stop();
  void display();
  const arcade::IModule::LibName getName() const;

protected:
private:
  sf::Texture _texture;
  sf::Font _font;
};
}; // namespace arcade

#endif /* !SFML_HPP_ */
```

7.19 /home/tmendy/Documents/Tek2/OOP/Arcade/include/IModule.hpp File Reference

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

Classes

- class arcade::IModule

    Interface for the modules.
- struct arcade::IModule::GameData

    information about the game from the game module to the graphic module

Namespaces

- namespace arcade

    namespace for the arcade project

7.20 IModule.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** IModule
*/

#ifndef IMODULE_HPP_
#define IMODULE_HPP_

#include <map> // Include the necessary header file
#include <string>
#include <vector>

namespace arcade {
class IModule {
public:
  IModule(){};
  virtual ~IModule(){};
  virtual void init() = 0;
  virtual void stop() = 0;
  struct GameData {
    std::vector<std::vector<int» display_info;
    std::map<unsigned int, std::string> sprite_value;
  };

  enum KeyboardInput {
    UP,
    DOWN,
    LEFT,
    RIGHT,
    SPACE,
    ENTER,
    ESCAPE,
    A,
    B,
    C,
    D,
    E,
    F,
    G,
    H,
    I,
    J,
    K,
    L,
    M,
    N,
    O,
    P,
    Q,
    R,
    S,
    T,
    U,
    V,
    W,
    X,
    Y,
    Z,
    NONE
  };

  enum ModuleType { GRAPHIC, GAME, CORE };

  enum LibName {
    // GAME
    SNAKE,
    NIBBLER,
    PACMAN,
    QIX,
    CENTIPEDE,
    SOLARFOX,
    // GRAPHIC
    NCURSES,
    SDL,
    SFML,
    OPENGL,
    // CORE
    UNKNOWN
  };
  virtual const LibName getName() const = 0;
  virtual const ModuleType getType() const = 0;
};
}; // namespace arcade

#endif /* !IMODULE_HPP_ */
```

7.21 /home/tmendy/Documents/Tek2/OOP/Arcade/include/Macros.hpp File Reference

Macros

- #define OK 0
- #define KO 84
- #define ERROR (-1)

7.21.1 Macro Definition Documentation

7.21.1.1 ERROR

```
#define ERROR (-1)
```

7.21.1.2 KO

```
#define KO 84
```

7.21.1.3 OK

```
#define OK 0
```

7.22 Macros.hpp

Go to the documentation of this file.

```
/*
** EPITECH PROJECT, 2024
** Arcade
** File description:
** Macros
*/

#ifndef MACROS_HPP_
#define MACROS_HPP_
#define OK 0
#define KO 84
#define ERROR (-1)

#endif /* !MACROS_HPP_ */
```

7.23 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/CoreModule.cpp File Reference

```
#include "CoreModule.hpp"
#include <NCurses.hpp>
#include <Pacman.hpp>
#include <Sdl2.hpp>
#include <Snake.hpp>
```

Functions

- std::vector< std::string > getLib (std::string pathLib)

  get the list of libraries in the given path

7.23.1 Function Documentation

7.23.1.1 getLib()

```
std::vector< std::string > getLib (
             std::string pathLib )
```

get the list of libraries in the given path

Parameters

pathLib	path to the libraries

Returns

std::vector<std::string> list of libraries

7.24 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/AGame↩ Module.cpp File Reference

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

7.25 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/pacman/↩ Pacman.cpp File Reference

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

Functions

- void init ()
  initialize the game
- void stop ()
  stop the game
- arcade::Pacman ∗ entryPoint ()
  generate entry point for the game library

7.25.1 Function Documentation

7.25.1.1 entryPoint()

```
arcade::Pacman * entryPoint ( )
```

generate entry point for the game library

7.25.1.2 init()

```
void init ( )
```

initialize the game

7.25.1.3 stop()

```
void stop ( )
```

stop the game

7.26 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/games/snake/↩ Snake.cpp File Reference

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

Functions

- arcade::Snake * entryPoint ()

  generate entry point for the game library

7.26.1 Function Documentation

7.26.1.1 entryPoint()

```
arcade::Snake * entryPoint ( )
```

generate entry point for the game library

7.27 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/↩ ADisplayModule.cpp File Reference

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

7.28 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/ncurses/↩ NCurses.cpp File Reference

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

Functions

- arcade::NCurses * entryPoint ()

7.28.1 Function Documentation

7.28.1.1 entryPoint()

```
arcade::NCurses * entryPoint ( )
```

7.29 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sdl2/↩ Sdl2.cpp File Reference

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

Functions

- arcade::Sdl2 ∗ entryPoint ()

7.29.1 Function Documentation

7.29.1.1 entryPoint()

```
arcade::Sdl2 * entryPoint ( )
```

7.30 /home/tmendy/Documents/Tek2/OOP/Arcade/lib/graphics/sfml/↩ Sfml.cpp File Reference

```
#include "Sfml.hpp"
```

Functions

- arcade::Sfml ∗ entryPoint ()

7.30.1 Function Documentation

7.30.1.1 entryPoint()

```
arcade::Sfml * entryPoint ( )
```

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

```
#include "DLLoader.hpp"
#include "IModule.hpp"
#include "Macros.hpp"
#include <ADisplayModule.hpp>
#include <cstring>
#include <iostream>
#include <libgen.h>
#include <unistd.h>
```

Functions

- int arcadeRe (char ∗path_graphic_lib)

  launch the arcade
- 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)

7.31.1 Function Documentation

7.31.1.1 arcadeRe()

```
int arcadeRe (
            char * path_graphic_lib )
```

launch the arcade

Parameters

path_graphic_lib	path of the graphic library

Returns

int OK if the arcade is launched

7.31.1.2 help()

```
void help (
            void  )
```

display the help

7.31.1.3 is_good_graphic_lib()

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

check if the library is a good graphic library

Parameters

path_graphic_lib	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

7.31.1.4 main()

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

Index