Arcade Project

Generated by Doxygen 1.8.18

Chapter 1

OOP_arcade_2019

1.1 Introduction

Arcade is a gaming platform: a program that lets the user choose a game to play and keeps a register of player scores.

To be able to deal with the elements of your gaming plateform at run-time, your graphic libraries and your games must be implemented as dynamic libraries, loaded at runtime.

Each GUI available for the program must be used as a shared library that will be loaded and used dynamically by the main program.

1.2 Add a dynamic library

Warning

> Your library must be have an entry point (a.k.a a function) named `entry`.

Place your library (only the `.so` extension is supported) in the `lib` or `games` folder depending if it's a graphical library or a game. Your file name must be respect the following pattern: `lib_arcade_$gamename.so` or `lib_arcade_$libraryname.so`

Warning

> The graphical library's entry point must return a `IGraphic *` instance
>
> The game library's entry point must return a `IGame *` instance

1.3 Collaboration

The project has been made in collaboration with:

- Louise KLEIVER - Leo KAIDER - Loic BRANSTETT
- Victor LIMBACH - Leo SEICHEPINE - Maxime SCHAEFFER

  Authors

  > Clément RUFFINONI - Quentin HALTER - Antoine PRONNIER

Chapter 2

OOP_arcade_2019

2.1 Introduction

Arcade is a gaming platform: a program that lets the user choose a game to play and keeps a register of player scores.

To be able to deal with the elements of your gaming plateform at run-time, your graphic libraries and your games must be implemented as dynamic libraries, loaded at runtime.

Each GUI available for the program must be used as a shared library that will be loaded and used dynamically by the main program.

2.2 Add a dynamic library

[!] Your library must be have an entry point (a.k.a a function) named `entry`. [!]

Place your library (only the `.so` extension is supported) in the `lib` or `games` folder depending if it's a graphical library or a game. Your file name must be respect the following pattern: `lib_arcade_$gamename.so` or `lib_arcade_$libraryname.so`.

[!] The graphical library's entry point must return a `IGraphic *` instance. [!]

[!] The game library's entry point must return a `IGame *` instance [!]

2.3 Installation

2.3.1 SFML installation:

https://www.sfml-dev.org/download/sfml/2.5.1/

2.3.2 SDL installation:

2.3.2.0.1 Debian-based: `sudo apt-get install libsdl-dev;`
`sudo apt-get install libsdl2-ttf-dev;`

2.3.2.0.2 Red hat-based:

```
sudo yum install SDL2-devel;
sudo yum install SDL2_ttf-devel;
```

2.3.3 Documentation with Doxygen:

The documentation can be generated with Doxygen. All the public methods, static variables and class are documented.

2.3.3.0.1 Installation:

- First, you need to install Doxygen: http://www.doxygen.nl/manual/install.html
- You can now generate all HTML files by running:
  ```
  doxygen DocFileDoxygen;
  ```
- When it's done, naviguate to the `html` folder and open the file `index.html`.
- Done, you have the whole documentation.

2.3.3.0.2 PDF version:

Associated with Doxygen, we use PDF Latex. You can download it at: https://miktex.org/download

You may require some additional package, if so download them by running:

```
sudo apt-get install texlive-latex-base texlive-fonts-recommended texlive-fonts-extra texlive-latex-extra
```

- You can now generate the PDF file:
  ```
  doxygen DocFileDoxygen;
  cd latex;
  make;
  ```

2.4 Collaboration

The project has been made in collaboration with:

- Louise KLEIVER - Leo KAIDER - Loic BRANSTETT
- Victor LIMBACH - Leo SEICHEPINE - Maxime SCHAEFFER

Authors: Clément RUFFINONI - Quentin HALTER - Antoine PRONNIER

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

Chapter 7

Namespace Documentation

7.1 Core Namespace Reference

Namespaces

- Exceptions

Classes

- class Core

7.2 Core::Exceptions Namespace Reference

Classes

- class EmptyMandatoryFolder
- class ExitGame
- class InvalidScorePath
- class MissingMandatoryFolder
- class UnableCreateFolder
- class UnknownGraphicalLib

7.3 Game Namespace Reference

Classes

- class Nibbler
- class Pacman
- class TestGame

7.4 Graphic Namespace Reference

Namespaces

- Exceptions

Classes

- class Ncurses
- class SDL2
- class SFML

7.4.1 Detailed Description

Graphic namespace include all graphical library.

7.5 Graphic::Exceptions Namespace Reference

Classes

- class LoadFontFailed

7.6 Score Namespace Reference

Namespaces

- Exceptions

Classes

- class File
- class Game

7.6.1 Detailed Description

Namespace Score:

The score namespace group 2 classes which are related to the score made in game and one class reserved to the exceptions.

7.7 Score::Exceptions Namespace Reference

Classes

- class InvalidFile

7.8 SoLoader Namespace Reference

Namespaces

- Exceptions

Classes

- class SoLoader

7.9 SoLoader::Exceptions Namespace Reference

Classes

- class InvalidEntryPoint
- class InvalidSO

7.9.1 Detailed Description

The namespace Exceptions regroup all exceptions related to the SoLoader namespace

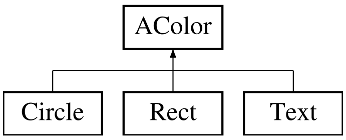
Chapter 8

Class Documentation

8.1 AColor Class Reference

```
#include <Drawables.hpp>
```

Inheritance diagram for AColor:



Public Member Functions

- AColor (const Color &color)
- ∼AColor ()=default
- void setColor (const Color &color)
- void setColorRed (unsigned char red)
- void setColorGreen (unsigned char green)
- void setColorBlue (unsigned char blue)
- void setColorAlpha (unsigned char alpha)
- Color getColor (void) const
- unsigned char getColorRed (void) const
- unsigned char getColorGreen (void) const
- unsigned char getColorBlue (void) const
- unsigned char getColorAlpha (void) const

Protected Attributes

- Color _color

8.1.1 Constructor & Destructor Documentation

8.1.1.1 AColor()

```
AColor::AColor (
            const Color & color )  [inline]
```

8.1.1.2 ∼AColor()

```
AColor::∼AColor ( )  [default]
```

8.1.2 Member Function Documentation

8.1.2.1 getColor()

```
Color AColor::getColor (
            void  ) const  [inline]
```

8.1.2.2 getColorAlpha()

```
unsigned char AColor::getColorAlpha (
            void  ) const  [inline]
```

8.1.2.3 getColorBlue()

```
unsigned char AColor::getColorBlue (
            void  ) const  [inline]
```

8.1.2.4 getColorGreen()

```
unsigned char AColor::getColorGreen (
            void  ) const  [inline]
```

8.1.2.5 getColorRed()

```
unsigned char AColor::getColorRed (
             void  ) const  [inline]
```

8.1.2.6 setColor()

```
void AColor::setColor (
             const Color & color )  [inline]
```

8.1.2.7 setColorAlpha()

```
void AColor::setColorAlpha (
             unsigned char alpha )  [inline]
```

8.1.2.8 setColorBlue()

```
void AColor::setColorBlue (
             unsigned char blue )  [inline]
```

8.1.2.9 setColorGreen()

```
void AColor::setColorGreen (
             unsigned char green )  [inline]
```

8.1.2.10 setColorRed()

```
void AColor::setColorRed (
             unsigned char red )  [inline]
```

8.1.3 Member Data Documentation

8.1.3.1 _color

```
Color AColor::_color  [protected]
```

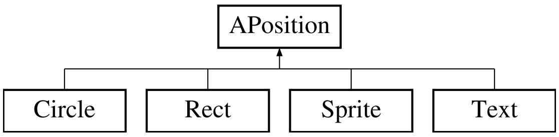
The documentation for this class was generated from the following file:

- src/graphic/Drawables.hpp

8.2 APosition Class Reference

```
#include <Drawables.hpp>
```

Inheritance diagram for APosition:



Public Member Functions

- APosition (const Vector2f &pos)
- ∼APosition ()=default
- void setPosition (const Vector2f pos)
- void setPositionX (float posX)
- void setPositionY (float posY)
- void move (const Vector2f pos)
- void moveX (float movx)
- void moveY (float movy)
- Vector2f getPosition (void) const
- float getPositionX (void) const
- float getPositionY (void) const

Protected Attributes

- Vector2f _pos

8.2.1 Constructor & Destructor Documentation

8.2.1.1 APosition()

```
APosition::APosition (
                const Vector2f & pos )  [inline]
```

8.2.1.2 ∼APosition()

```
APosition::∼APosition ( )  [default]
```

8.2.2 Member Function Documentation

8.2.2.1 getPosition()

```
Vector2f APosition::getPosition (
             void  ) const  [inline]
```

8.2.2.2 getPositionX()

```
float APosition::getPositionX (
             void  ) const  [inline]
```

8.2.2.3 getPositionY()

```
float APosition::getPositionY (
             void  ) const  [inline]
```

8.2.2.4 move()

```
void APosition::move (
             const Vector2f pos )  [inline]
```

8.2.2.5 moveX()

```
void APosition::moveX (
             float movx )  [inline]
```

8.2.2.6 moveY()

```
void APosition::moveY (
             float movy )  [inline]
```

8.2.2.7 setPosition()

```
void APosition::setPosition (
             const Vector2f pos )  [inline]
```

8.2.2.8 setPositionX()

```
void APosition::setPositionX (
             float posX )  [inline]
```

8.2.2.9 setPositionY()

```
void APosition::setPositionY (
             float posY )  [inline]
```

8.2.3 Member Data Documentation

8.2.3.1 _pos

```
Vector2f APosition::_pos  [protected]
```

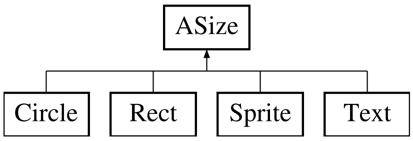
The documentation for this class was generated from the following file:

- src/graphic/Drawables.hpp

8.3 ASize Class Reference

```
#include <Drawables.hpp>
```

Inheritance diagram for ASize:



Public Member Functions

- ASize (const Vector2f &size)
- ∼ASize ()=default
- void setSize (const Vector2f &size)
- void setSizeX (float sizeX)
- void setSizeY (float sizeY)
- Vector2f getSize (void) const
- float getSizeX (void) const
- float getSizeY (void) const

Protected Attributes

- Vector2f _size

8.3.1 Constructor & Destructor Documentation

8.3.1.1 ASize()

```
ASize::ASize (
              const Vector2f & size )  [inline]
```

8.3.1.2 ∼ASize()

```
ASize::∼ASize ( )  [default]
```

8.3.2 Member Function Documentation

8.3.2.1 getSize()

```
Vector2f ASize::getSize (
              void  ) const  [inline]
```

8.3.2.2 getSizeX()

```
float ASize::getSizeX (
              void  ) const  [inline]
```

8.3.2.3 getSizeY()

```
float ASize::getSizeY (
             void  ) const  [inline]
```

8.3.2.4 setSize()

```
void ASize::setSize (
             const Vector2f & size )  [inline]
```

8.3.2.5 setSizeX()

```
void ASize::setSizeX (
             float sizeX )  [inline]
```

8.3.2.6 setSizeY()

```
void ASize::setSizeY (
             float sizeY )  [inline]
```

8.3.3 Member Data Documentation

8.3.3.1 _size

```
Vector2f ASize::_size  [protected]
```

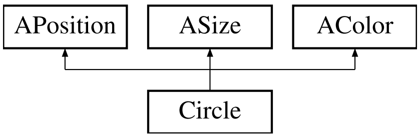
The documentation for this class was generated from the following file:

- src/graphic/Drawables.hpp

8.4 Circle Class Reference

```
#include <Drawables.hpp>
```

Inheritance diagram for Circle:



Public Member Functions

- Circle (Vector2f pos=Vector2f(0, 0), Vector2f size=Vector2f(0, 0), Color color=Color(0, 0, 0, 0))
- ∼Circle ()=default

Additional Inherited Members

8.4.1 Constructor & Destructor Documentation

8.4.1.1 Circle()

```
Circle::Circle (
          Vector2f pos = Vector2f(0, 0),
          Vector2f size = Vector2f(0, 0),
          Color color = Color(0, 0, 0, 0) )  [inline]
```

8.4.1.2 ∼Circle()

```
Circle::∼Circle ( )  [default]
```

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

- src/graphic/Drawables.hpp

8.5 Color Struct Reference

```
#include <Drawables.hpp>
```

Public Member Functions

- bool operator== (const Color a) const
- Color (unsigned char red, unsigned char blue, unsigned char green, unsigned char alpha)

Static Public Member Functions

- static Color Red ()
- static Color Blue ()
- static Color Green ()
- static Color Black ()
- static Color White ()
- static Color Transparent ()

Public Attributes

- unsigned char red
- unsigned char blue
- unsigned char green
- unsigned char alpha

8.5.1 Constructor & Destructor Documentation

8.5.1.1 Color()

```
Color::Color (
             unsigned char red,
             unsigned char blue,
             unsigned char green,
             unsigned char alpha )  [inline]
```

8.5.2 Member Function Documentation

8.5.2.1 Black()

```
static Color Color::Black ( )  [inline], [static]
```

8.5.2.2 Blue()

```
static Color Color::Blue ( )  [inline], [static]
```

8.5.2.3 Green()

```
static Color Color::Green ( )  [inline], [static]
```

8.5.2.4 operator==()

```
bool Color::operator== (
             const Color a ) const  [inline]
```

8.5.2.5 Red()

```
static Color Color::Red ( )  [inline], [static]
```

8.5.2.6 Transparent()

```
static Color Color::Transparent ( )  [inline], [static]
```

8.5.2.7 White()

```
static Color Color::White ( )  [inline], [static]
```

8.5.3 Member Data Documentation

8.5.3.1 alpha

```
unsigned char Color::alpha
```

8.5.3.2 blue

```
unsigned char Color::blue
```

8.5.3.3 green

```
unsigned char Color::green
```

8.5.3.4 red

```
unsigned char Color::red
```

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

- src/graphic/Drawables.hpp

8.6 Core::Core Class Reference

```
#include <Core.hpp>
```

Public Member Functions

- Core (const std::string &graphicalLib)
- ∼Core ()=default
- void useGraphic (const std::string &filename)
- void useGame (const std::string &filename)
- void run ()

Static Public Member Functions

- static Score::File loadScore (const std::string &gameName)

Static Public Attributes

- constexpr static const char ∗ SCORE_PATH = "./games/score/"
- static const uint FRAME_PER_TIME = 17

8.6.1 Detailed Description

Class Core:

This class is the main one which is coordinate the graphic and the games libraries.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 Core()

```
Core::Core::Core (
            const std::string & graphicalLib )  [explicit]
```

Constructor for the Core class.

Parameters

graphicalLib	: The path to the graphical library to load.

Exceptions

SoLoader::Exceptions::InvalidSO	

Exceptions

SoLoader::Exceptions::InvalidEntryPoint	
Core::Exceptions::EmptyMandatoryFolder	
Core::Exceptions::UnknownGraphicalLib	
Core::Exceptions::UnableCreateFolder	
Core::Exceptions::MissingMandatoryFolder	

8.6.2.2 ∼Core()

```
Core::Core::∼Core ( )  [default]
```

8.6.3 Member Function Documentation

8.6.3.1 loadScore()

```
Score::File Core::Core::loadScore (
             const std::string & gameName )  [static]
```

Load a score file. The score's files path is defined by the static public variable `SCORE_PATH`.

Parameters

gameName	: The game's name score to load.

Exceptions

Core::Exceptions::InvalidScorePath	

Returns

8.6.3.2 run()

```
void Core::Core::run ( )
```

Launch the main loop for the project. If the loop is over, the programme ends.

8.6.3.3 useGame()

```
void Core::Core::useGame (
             const std::string & filename )
```

Load a game library and store the instance. Can throw exceptions.

Parameters

filename	: The path to the game library.

Exceptions

SoLoader::Exceptions::InvalidSO	
SoLoader::Exceptions::InvalidEntryPoint	

8.6.3.4 useGraphic()

```
void Core::Core::useGraphic (
             const std::string & filename )
```

Load a graphical library and store the instance. Can throw exceptions.

Parameters

filename	: The path to the graphical library.

Exceptions

SoLoader::Exceptions::InvalidSO	
SoLoader::Exceptions::InvalidEntryPoint	

8.6.4 Member Data Documentation

8.6.4.1 FRAME_PER_TIME

```
const uint Core::Core::FRAME_PER_TIME = 17  [static]
```

1000 / 60 = 17 (∼ 16,666666667)

- 1000 milliseconds = 1 sec

- 60 FPS (frame per seconds) or the number of total images that we want per second -> 17 = number of frame / millisecond
- We choose millisecond as the architecture stipulate it

8.6.4.2 SCORE_PATH

```
constexpr static const char* Core::Core::SCORE_PATH = "./games/score/"  [static], [constexpr]
```

See `loadScore` description.

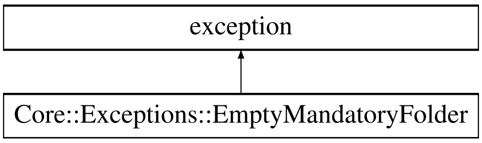
The documentation for this class was generated from the following files:

- src/core/Core.hpp
- src/core/Core.cpp
- src/core/KeyEvents.cpp
- src/core/menu/Menu.cpp
- src/core/menu/Strip.cpp
- src/core/menu/Utils.cpp

8.7 Core::Exceptions::EmptyMandatoryFolder Class Reference

```
#include <Core.hpp>
```

Inheritance diagram for Core::Exceptions::EmptyMandatoryFolder:



Public Member Functions

- EmptyMandatoryFolder (const std::string &name) noexcept
- EmptyMandatoryFolder ()=delete
- const char * what () const noexcept override

8.7.1 Detailed Description

EmptyMandatoryFolder is an exception throw when the mandatory folders are empty. That means there is no library (games or graphical libraries) inside one nor both empty.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 EmptyMandatoryFolder() [1/2]

```
Core::Exceptions::EmptyMandatoryFolder::EmptyMandatoryFolder (
            const std::string & name )  [explicit], [noexcept]
```

8.7.2.2 EmptyMandatoryFolder() [2/2]

```
Core::Exceptions::EmptyMandatoryFolder::EmptyMandatoryFolder ( )  [delete]
```

8.7.3 Member Function Documentation

8.7.3.1 what()

```
const char * Core::Exceptions::EmptyMandatoryFolder::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred.

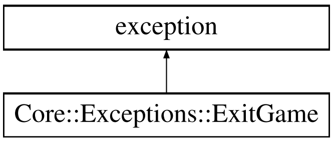
The documentation for this class was generated from the following files:

- src/core/Core.hpp
- src/core/Exceptions.cpp

8.8 Core::Exceptions::ExitGame Class Reference

```
#include <Core.hpp>
```

Inheritance diagram for Core::Exceptions::ExitGame:



Public Member Functions

- ExitGame () noexcept=default
- const char * what () const noexcept override

8.8.1 Detailed Description

ExitGame is a minor exception throw when the program must exit gracefully.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 ExitGame()

```
Core::Exceptions::ExitGame::ExitGame ( )  [default], [noexcept]
```

8.8.3 Member Function Documentation

8.8.3.1 what()

```
const char * Core::Exceptions::ExitGame::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

> An explicit message explaining why the error occurred.

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

- src/core/Core.hpp
- src/core/Exceptions.cpp

8.9 Score::File Class Reference

```
#include <Score.hpp>
```

Classes

- class PlayerData

Public Member Functions

- File (const std::string &content) noexcept
- File ()=default
- void addEntry (const PlayerData &user)
- std::list< PlayerData > getListUser ()

8.9.1 Constructor & Destructor Documentation

8.9.1.1 File() [1/2]

```
Score::File::File (
            const std::string & content )  [explicit], [noexcept]
```

Initialize a File instance with the content of a file.

Parameters

content	: Content of the file which will be initialized

8.9.1.2 File() [2/2]

```
Score::File::File ( )  [default]
```

8.9.2 Member Function Documentation

8.9.2.1 addEntry()

```
void Score::File::addEntry (
             const PlayerData & user )
```

Add an entry to the list of registered users. Users are PlayerData type

Parameters

user	The user to add in the list

8.9.2.2 getListUser()

```
std::list< Score::File::PlayerData > Score::File::getListUser ( )
```

Get a list of all registered users. Users are PlayerData type

Returns

A list of PlayerData

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

- src/core/score/Score.hpp
- src/core/score/File.cpp

8.10 Score::Game Class Reference

```
#include <Score.hpp>
```

Public Member Functions

- Game ()
- void addLetter (const char &letter)
- const char & getLetter (const std::size_t idx) const
- void setLetter (const std::size_t idx, const char c)
- void save () const
- std::string getAuthor () const
- void setScore (const std::size_t score)
- void setGame (const std::string &game)
- void reset ()
- void removeLastLetter ()

Static Public Attributes

- static const std::size_t MAX_AUTHOR_NAME = 8

8.10.1 Detailed Description

Class Game:

The class is dedicated to the score produced directly in-game. It's composed of the author and his score.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 Game()

```
Score::Game::Game ( )
```

Basic constructor which is initialize private values

8.10.3 Member Function Documentation

8.10.3.1 addLetter()

```
void Score::Game::addLetter (
             const char & letter )
```

Add a letter to the author's name

Parameters

letter	: The letter to be added

8.10.3.2 getAuthor()

```
std::string Score::Game::getAuthor ( ) const
```

Get the author's name saved in private

Returns

A string with the author's name

8.10.3.3 getLetter()

```
const char & Score::Game::getLetter (
             const std::size_t idx ) const
```

Get a letter at an certain index

Parameters

idx	: The index of the letter

Exceptions

std::out_of_range	: If the index is not in the range

Returns

The letter as char

8.10.3.4 removeLastLetter()

```
void Score::Game::removeLastLetter ( )
```

Remove the last letter of the author's name

Exceptions

std::out_of_range	: If the author's name is empty

8.10.3.5 reset()

```
void Score::Game::reset ( )
```

Reset the score's amount

8.10.3.6 save()

```
void Score::Game::save ( ) const
```

Save the current score with the author's name into a file. The file path is defined by class Core and the game's name

Exceptions

Score::Exceptions::InvalidFile	: Cannot create/open the file

8.10.3.7 setGame()

```
void Score::Game::setGame (
             const std::string & game )
```

Set the game's name which will be used when the file is saved

Parameters

game	: The game's name

8.10.3.8 setLetter()

```
void Score::Game::setLetter (
             const std::size_t idx,
             const char c )
```

Set a letter at the certain index

Parameters

idx	: The index of the letter
c	: The letter which will be replaced

Exceptions

std::out_of_range	: If the index is not in the range

8.10.3.9 setScore()

```
void Score::Game::setScore (
             const std::size_t score )
```

Set the score's amount to `score`

Parameters

score	New score's amount

8.10.4 Member Data Documentation

8.10.4.1 MAX_AUTHOR_NAME

```
const std::size_t Score::Game::MAX_AUTHOR_NAME = 8  [static]
```

The maximum length of the author's name

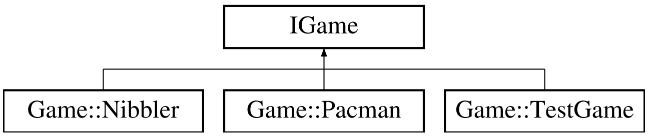
The documentation for this class was generated from the following files:

- src/core/score/Score.hpp
- src/core/score/Game.cpp

8.11 IGame Class Reference

```
#include <IGame.hpp>
```

Inheritance diagram for IGame:



Public Types

- typedef std::map< std::string, std::string > GameDataType

Public Member Functions

- virtual ~IGame ()=default
- virtual void handleEvent (std::string &name)=0
- virtual void handleUpdate (int elapsedTime)=0
- virtual void handleRender (IGraphicRenderer &renderer)=0
- virtual GameDataType getGameData ()=0
- virtual void setGameData (GameDataType &data)=0

8.11.1 Member Typedef Documentation

8.11.1.1 GameDataType

```
typedef std::map<std::string, std::string> IGame::GameDataType
```

8.11.2 Constructor & Destructor Documentation

8.11.2.1 ~IGame()

```
virtual IGame::~IGame ( )  [virtual], [default]
```

8.11.3 Member Function Documentation

8.11.3.1 getGameData()

```
virtual GameDataType IGame::getGameData ( )  [pure virtual]
```

Implemented in Game::Nibbler, Game::Pacman, and Game::TestGame.

8.11.3.2 handleEvent()

```
virtual void IGame::handleEvent (
             std::string & name )  [pure virtual]
```

Implemented in Game::Nibbler, Game::Pacman, and Game::TestGame.

8.11.3.3 handleRender()

```
virtual void IGame::handleRender (
             IGraphicRenderer & renderer )  [pure virtual]
```

Implemented in Game::Nibbler, Game::Pacman, and Game::TestGame.

8.11.3.4 handleUpdate()

```
virtual void IGame::handleUpdate (
             int elapsedTime )  [pure virtual]
```

Implemented in Game::Nibbler, Game::Pacman, and Game::TestGame.

8.11.3.5 setGameData()

```
virtual void IGame::setGameData (
             GameDataType & data )  [pure virtual]
```

Implemented in Game::Nibbler, Game::Pacman, and Game::TestGame.

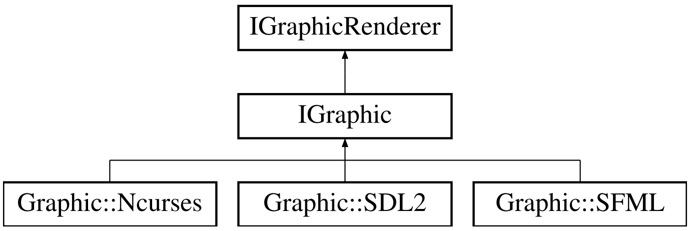
The documentation for this class was generated from the following file:

- src/game/IGame.hpp

8.12 IGraphic Class Reference

```
#include <IGraphic.hpp>
```

Inheritance diagram for IGraphic:



Public Member Functions

- IGraphic ()=default
- IGraphic (const IGraphic &)=delete
- virtual ∼IGraphic ()=default
- virtual bool isOperational ()=0
- virtual std::string handleEvent ()=0

8.12.1 Constructor & Destructor Documentation

8.12.1.1 IGraphic() [1/2]

```
IGraphic::IGraphic ( )  [default]
```

8.12.1.2 IGraphic() [2/2]

```
IGraphic::IGraphic (
            const IGraphic &  )  [delete]
```

8.12.1.3 ∼IGraphic()

```
virtual IGraphic::∼IGraphic ( )  [virtual], [default]
```

8.12.2 Member Function Documentation

8.12.2.1 handleEvent()

```
virtual std::string IGraphic::handleEvent ( )  [pure virtual]
```

The function handle keyboard events

Returns

A string with the latest event. The string must be one element of the structure KeyboardEvent_s.

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

8.12.2.2 isOperational()

```
virtual bool IGraphic::isOperational ( ) [pure virtual]
```

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

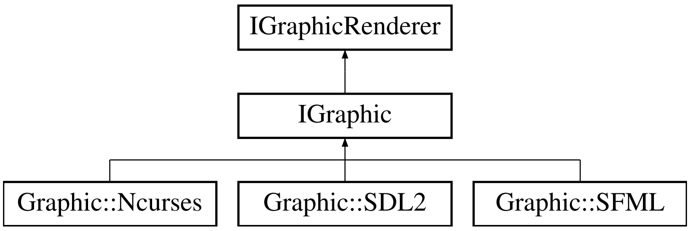
The documentation for this class was generated from the following file:

- src/graphic/IGraphic.hpp

8.13 IGraphicRenderer Class Reference

```
#include <IGraphic.hpp>
```

Inheritance diagram for IGraphicRenderer:



Public Member Functions

- virtual ∼IGraphicRenderer ()=default
- virtual void drawScreen ()=0
- virtual void clearScreen ()=0
- virtual void drawRect (Rect rect)=0
- virtual void drawCircle (Circle circle)=0
- virtual void drawSprite (Sprite sprite)=0
- virtual void drawText (Text text)=0
- IGraphicRenderer & operator= (const IGraphicRenderer &)=delete

8.13.1 Detailed Description

The IGraphicRenderer class render the graphical aspect of a game.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 ∼IGraphicRenderer()

```
virtual IGraphicRenderer::∼IGraphicRenderer ( ) [virtual], [default]
```

8.13.3 Member Function Documentation

8.13.3.1 clearScreen()

```
virtual void IGraphicRenderer::clearScreen ( )  [pure virtual]
```

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

8.13.3.2 drawCircle()

```
virtual void IGraphicRenderer::drawCircle (
            Circle circle )  [pure virtual]
```

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

8.13.3.3 drawRect()

```
virtual void IGraphicRenderer::drawRect (
            Rect rect )  [pure virtual]
```

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

8.13.3.4 drawScreen()

```
virtual void IGraphicRenderer::drawScreen ( )  [pure virtual]
```

Draw all visual elements on the screen. Up to each library graphical library to handle the way they manages their graphical entities.

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

8.13.3.5 drawSprite()

```
virtual void IGraphicRenderer::drawSprite (
            Sprite sprite )  [pure virtual]
```

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

8.13.3.6 drawText()

```
virtual void IGraphicRenderer::drawText (
            Text text )  [pure virtual]
```

Implemented in Graphic::Ncurses, Graphic::SFML, and Graphic::SDL2.

8.13.3.7 operator=()

```
IGraphicRenderer& IGraphicRenderer::operator= (
            const IGraphicRenderer &  )  [delete]
```

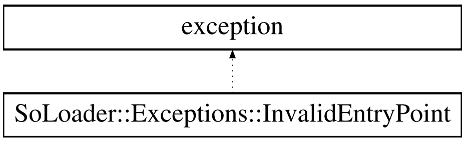
The documentation for this class was generated from the following file:

- src/graphic/IGraphic.hpp

8.14 SoLoader::Exceptions::InvalidEntryPoint Class Reference

```
#include <Exceptions.hpp>
```

Inheritance diagram for SoLoader::Exceptions::InvalidEntryPoint:



Public Member Functions

- InvalidEntryPoint (const std::string &DLLName) noexcept
- InvalidEntryPoint ()=delete
- const char ∗ what () const noexcept override

8.14.1 Detailed Description

InvalidEntryPoint is an exception throw when the SO library previously doesn't have the entry point specified by SoLoader::SoLoader::ENTRY_POINT_NAME.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 InvalidEntryPoint() [1/2]

```
SoLoader::Exceptions::InvalidEntryPoint::InvalidEntryPoint (
             const std::string & DLLName )  [explicit], [noexcept]
```

8.14.2.2 InvalidEntryPoint() [2/2]

```
SoLoader::Exceptions::InvalidEntryPoint::InvalidEntryPoint ( )  [delete]
```

8.14.3 Member Function Documentation

8.14.3.1 what()

```
const char * SoLoader::Exceptions::InvalidEntryPoint::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred.

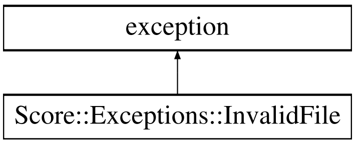
The documentation for this class was generated from the following files:

- src/soLoader/Exceptions.hpp
- src/soLoader/Exceptions.cpp

8.15 Score::Exceptions::InvalidFile Class Reference

```
#include <Score.hpp>
```

Inheritance diagram for Score::Exceptions::InvalidFile:



Public Member Functions

- InvalidFile ()=default
- const char ∗ what () const noexcept override

8.15.1 Detailed Description

InvalidFile is an exception throw when a file cannot be open nor created

8.15.2 Constructor & Destructor Documentation

8.15.2.1 InvalidFile()

```
Score::Exceptions::InvalidFile::InvalidFile ( )  [default]
```

8.15.3 Member Function Documentation

8.15.3.1 what()

```
const char * Score::Exceptions::InvalidFile::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred

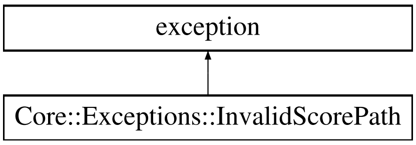
The documentation for this class was generated from the following files:

- src/core/score/Score.hpp
- src/core/score/Exceptions.cpp

8.16 Core::Exceptions::InvalidScorePath Class Reference

```
#include <Core.hpp>
```

Inheritance diagram for Core::Exceptions::InvalidScorePath:



Public Member Functions

- InvalidScorePath (const std::string &name) noexcept
- InvalidScorePath ()=delete
- const char * what () const noexcept override

8.16.1 Detailed Description

InvalidScorePath is a minor exception throw when the path to load a score file is invalid. The program must handle this exception since it's a minor one.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 InvalidScorePath() [1/2]

```
Core::Exceptions::InvalidScorePath::InvalidScorePath (
             const std::string & name )  [explicit], [noexcept]
```

8.16.2.2 InvalidScorePath() [2/2]

```
Core::Exceptions::InvalidScorePath::InvalidScorePath ( )  [delete]
```

8.16.3 Member Function Documentation

8.16.3.1 what()

```
const char * Core::Exceptions::InvalidScorePath::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred.

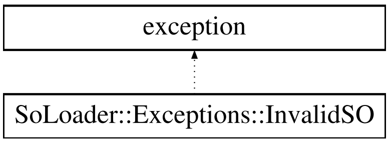
The documentation for this class was generated from the following files:

- src/core/Core.hpp
- src/core/Exceptions.cpp

8.17 SoLoader::Exceptions::InvalidSO Class Reference

```
#include <Exceptions.hpp>
```

Inheritance diagram for SoLoader::Exceptions::InvalidSO:



Public Member Functions

- InvalidSO (const std::string &DLLPath) noexcept
- InvalidSO ()=delete
- const char ∗ what () const noexcept override

8.17.1 Detailed Description

InvalidSO is an exception throw when the SO library passed as argument cannot be open by `dlopen`. It may be due to several factors, refer to the error message displayed if this occurs.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 InvalidSO() [1/2]

```
SoLoader::Exceptions::InvalidSO::InvalidSO (
            const std::string & DLLPath )  [explicit], [noexcept]
```

8.17.2.2 InvalidSO() [2/2]

```
SoLoader::Exceptions::InvalidSO::InvalidSO ( )  [delete]
```

8.17.3 Member Function Documentation

8.17.3.1 what()

```
const char * SoLoader::Exceptions::InvalidSO::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred.

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

- src/soLoader/Exceptions.hpp
- src/soLoader/Exceptions.cpp

8.18 KeyboardEvent_s Struct Reference

```
#include <IGraphic.hpp>
```

Static Public Attributes

- static constexpr const char * NEXT_GAME = "next_game"
- static constexpr const char * PREV_GAME = "prev_game"
- static constexpr const char * RESTART = "restart"
- static constexpr const char * NEXT_GRAPHIC = "next_graphic"
- static constexpr const char * PREV_GRAPHIC = "prev_graphic"
- static constexpr const char * RIGHT = "right"
- static constexpr const char * UP = "up"
- static constexpr const char * DOWN = "down"
- static constexpr const char * LEFT = "left"
- static constexpr const char * ENTER = "enter"
- static constexpr const char * ESC = "menu"
- static constexpr const char * SPACE = "space"
- static constexpr const char * UNKNOWN = "unknown"

8.18.1 Detailed Description

The structure KeyboardEvent_s define generic values for the different action that can occurs in the program. The associated key for each key is defined inside the private function: Core::Core::handleInternalKey and is displayed during runtime.

8.18.2 Member Data Documentation

8.18.2.1 DOWN

```
constexpr const char* KeyboardEvent_s::DOWN = "down"  [static], [constexpr]
```

8.18.2.2 ENTER

```
constexpr const char* KeyboardEvent_s::ENTER = "enter"  [static], [constexpr]
```

8.18.2.3 ESC

```
constexpr const char* KeyboardEvent_s::ESC = "menu"  [static], [constexpr]
```

8.18.2.4 LEFT

```
constexpr const char* KeyboardEvent_s::LEFT = "left"  [static], [constexpr]
```

8.18.2.5 NEXT_GAME

```
constexpr const char* KeyboardEvent_s::NEXT_GAME = "next_game"  [static], [constexpr]
```

8.18.2.6 NEXT_GRAPHIC

```
constexpr const char* KeyboardEvent_s::NEXT_GRAPHIC = "next_graphic"  [static], [constexpr]
```

8.18.2.7 PREV_GAME

```
constexpr const char* KeyboardEvent_s::PREV_GAME = "prev_game"  [static], [constexpr]
```

8.18.2.8 PREV_GRAPHIC

```
constexpr const char* KeyboardEvent_s::PREV_GRAPHIC = "prev_graphic"  [static], [constexpr]
```

8.18.2.9 RESTART

```
constexpr const char* KeyboardEvent_s::RESTART = "restart"  [static], [constexpr]
```

8.18.2.10 RIGHT

```
constexpr const char* KeyboardEvent_s::RIGHT = "right"  [static], [constexpr]
```

8.18.2.11 SPACE

```
constexpr const char* KeyboardEvent_s::SPACE = "space"  [static], [constexpr]
```

8.18.2.12 UNKNOWN

```
constexpr const char* KeyboardEvent_s::UNKNOWN = "unknown"  [static], [constexpr]
```

8.18.2.13 UP

```
constexpr const char* KeyboardEvent_s::UP = "up"  [static], [constexpr]
```

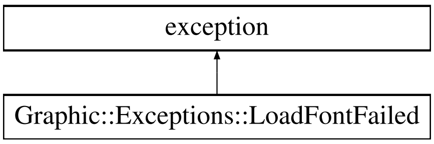
The documentation for this struct was generated from the following file:

- src/graphic/IGraphic.hpp

8.19 Graphic::Exceptions::LoadFontFailed Class Reference

```
#include <Exceptions.hpp>
```

Inheritance diagram for Graphic::Exceptions::LoadFontFailed:



Public Member Functions

- LoadFontFailed (const LoadFontFailed &a) noexcept
- LoadFontFailed (const std::string &fontName) noexcept
- const char ∗ what () const noexcept override

8.19.1 Constructor & Destructor Documentation

8.19.1.1 LoadFontFailed() [1/2]

```
Graphic::Exceptions::LoadFontFailed::LoadFontFailed (
            const LoadFontFailed & a )  [inline], [noexcept]
```

8.19.1.2 LoadFontFailed() [2/2]

```
Graphic::Exceptions::LoadFontFailed::LoadFontFailed (
            const std::string & fontName )  [inline], [explicit], [noexcept]
```

8.19.2 Member Function Documentation

8.19.2.1 what()

```
const char* Graphic::Exceptions::LoadFontFailed::what ( ) const  [inline], [override], [noexcept]
```

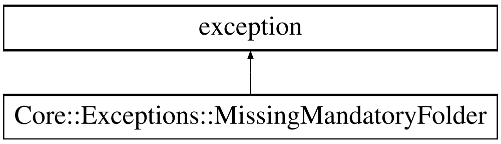
The documentation for this class was generated from the following file:

- src/lib/graphic/Exceptions.hpp

8.20 Core::Exceptions::MissingMandatoryFolder Class Reference

```
#include <Core.hpp>
```

Inheritance diagram for Core::Exceptions::MissingMandatoryFolder:



Public Member Functions

- MissingMandatoryFolder (const std::string &name) noexcept
- MissingMandatoryFolder ()=delete
- const char ∗ what () const noexcept override

8.20.1 Detailed Description

MissingMandatoryFolder is an exception throw when the mandatory folders are missing. These folders are required by the project 'Arcade'.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 MissingMandatoryFolder() [1/2]

```
Core::Exceptions::MissingMandatoryFolder::MissingMandatoryFolder (
            const std::string & name )  [explicit], [noexcept]
```

8.20.2.2 MissingMandatoryFolder() [2/2]

```
Core::Exceptions::MissingMandatoryFolder::MissingMandatoryFolder ( )  [delete]
```

8.20.3 Member Function Documentation

8.20.3.1 what()

```
const char * Core::Exceptions::MissingMandatoryFolder::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred.

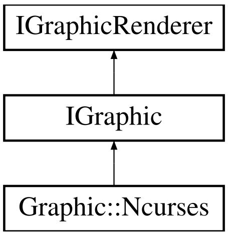
The documentation for this class was generated from the following files:

- src/core/Core.hpp
- src/core/Exceptions.cpp

8.21 Graphic::Ncurses Class Reference

```
#include <Ncurses.hpp>
```

Inheritance diagram for Graphic::Ncurses:



Public Member Functions

- Ncurses ()
- ∼Ncurses () override
- void clearScreen () override
- void drawCircle (Circle circle) override
- void drawRect (Rect rect) override
- void drawScreen () override
- void drawSprite (Sprite sprite) override
- void drawText (Text text) override
- std::string handleEvent () override
- bool isOperational () override

Public Attributes

- std::array< Color, 8 > TRANSLATE_COLORS

Static Public Attributes

- static const int WINDOW_WIDTH = 100
- static const int WINDOW_HEIGHT = 40

8.21.1 Constructor & Destructor Documentation

8.21.1.1 Ncurses()

```
Graphic::Ncurses::Ncurses ( )
```

Basic constructor for SFML. It creates the window with some specific parameters (framerate, window's name, ...).

8.21.1.2 ∼Ncurses()

```
Graphic::Ncurses::∼Ncurses ( )  [override]
```

8.21.2 Member Function Documentation

8.21.2.1 clearScreen()

```
void Graphic::Ncurses::clearScreen ( )  [override], [virtual]
```

Implements IGraphicRenderer.

8.21.2.2 drawCircle()

```
void Graphic::Ncurses::drawCircle (
            Circle circle )  [override], [virtual]
```

Implements IGraphicRenderer.

8.21.2.3 drawRect()

```
void Graphic::Ncurses::drawRect (
            Rect rect )  [override], [virtual]
```

Implements IGraphicRenderer.

8.21.2.4 drawScreen()

```
void Graphic::Ncurses::drawScreen ( )  [override], [virtual]
```

Draw all visual elements on the screen. Up to each library graphical library to handle the way they manages their graphical entities.

Implements IGraphicRenderer.

8.21.2.5 drawSprite()

```
void Graphic::Ncurses::drawSprite (
            Sprite sprite )  [override], [virtual]
```

Implements IGraphicRenderer.

8.21.2.6 drawText()

```
void Graphic::Ncurses::drawText (
            Text text )  [override], [virtual]
```

Implements IGraphicRenderer.

8.21.2.7 handleEvent()

```
std::string Graphic::Ncurses::handleEvent ( )  [override], [virtual]
```

The function handle keyboard events

Returns

A string with the latest event. The string must be one element of the structure KeyboardEvent_s.

Implements IGraphic.

8.21.2.8 isOperational()

```
bool Graphic::Ncurses::isOperational ( )  [override], [virtual]
```

Implements IGraphic.

8.21.3 Member Data Documentation

8.21.3.1 TRANSLATE_COLORS

```
std::array<Color, 8> Graphic::Ncurses::TRANSLATE_COLORS
```

Initial value:

```
= {
                Color {Color::Black().red, Color::Black().blue, Color::Black().green, Color::Black().alpha},
                Color {Color::Red().red, Color::Red().blue, Color::Red().green, Color::Red().alpha},
                Color {Color::Green().red, Color::Green().blue, Color::Green().green, Color::Green().alpha},
                Color {230, 0, 230, 255},
                Color {Color::Blue().red, Color::Blue().blue, Color::Blue().green, Color::Blue().alpha},
                Color {255, 255, 0, Color::Blue().alpha},
                Color {0, 255, 255, Color::Blue().alpha},
                Color {Color::White().red, Color::White().blue, Color::White().green, Color::White().alpha},
            }
```

This array correspond to the liaison between NCurses' colors and the color's value within the Color class.

8.21.3.2 WINDOW_HEIGHT

```
const int Graphic::Ncurses::WINDOW_HEIGHT = 40  [static]
```

8.21.3.3 WINDOW_WIDTH

```
const int Graphic::Ncurses::WINDOW_WIDTH = 100  [static]
```

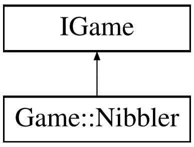
The documentation for this class was generated from the following files:

- src/lib/graphic/ncurses/Ncurses.hpp
- src/lib/graphic/ncurses/Ncurses.cpp

8.22 Game::Nibbler Class Reference

```
#include <Nibbler.hpp>
```

Inheritance diagram for Game::Nibbler:



Public Member Functions

- Nibbler ()
- ∼Nibbler () override=default
- void handleEvent (std::string &name) override
- void handleRender (IGraphicRenderer &renderer) override
- void handleUpdate (int elapsedTime) override
- void setGameData (GameDataType &data) override
- GameDataType getGameData () override

Additional Inherited Members

8.22.1 Constructor & Destructor Documentation

8.22.1.1 Nibbler()

```
Game::Nibbler::Nibbler ( )
```

Class Nibbler:

This class is the main class for the game Nibbler. Inherits from IGame Interface Constructor for the Nibbler class.

8.22.1.2 ∼Nibbler()

```
Game::Nibbler::∼Nibbler ( )  [override], [default]
```

8.22.2 Member Function Documentation

8.22.2.1 getGameData()

```
IGame::GameDataType Game::Nibbler::getGameData ( )  [override], [virtual]
```

Get game data.

Implements IGame.

8.22.2.2 handleEvent()

```
void Game::Nibbler::handleEvent (
             std::string & name )  [override], [virtual]
```

Handle an event from the IGraphic library.

Parameters

name	: The name of the event.

Implements IGame.

8.22.2.3 handleRender()

```
void Game::Nibbler::handleRender (
            IGraphicRenderer & renderer )  [override], [virtual]
```

Handle screen graphic actualization.

Parameters

renderer	: The graphic lib to render.

Implements IGame.

8.22.2.4 handleUpdate()

```
void Game::Nibbler::handleUpdate (
            int elapsedTime )  [override], [virtual]
```

Handle clock time for the game.

Parameters

elapsedTime	: The time length between two calls.

Implements IGame.

8.22.2.5 setGameData()

```
void Game::Nibbler::setGameData (
            IGame::GameDataType & data )  [override], [virtual]
```

Update game data.

Parameters

data	: The game data to send.

Implements IGame.

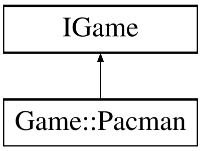
The documentation for this class was generated from the following files:

- src/lib/game/nibbler/Nibbler.hpp
- src/lib/game/nibbler/Draw.cpp
- src/lib/game/nibbler/Nibbler.cpp

8.23 Game::Pacman Class Reference

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

Inheritance diagram for Game::Pacman:



Public Member Functions

- Pacman ()
- ∼Pacman () override=default
- void handleEvent (std::string &name) override
- void handleRender (IGraphicRenderer &renderer) override
- void handleUpdate (int elapsedTime) override
- void setGameData (GameDataType &data) override
- GameDataType getGameData () override

Additional Inherited Members

8.23.1 Constructor & Destructor Documentation

8.23.1.1 Pacman()

```
Game::Pacman::Pacman ( )
```

8.23.1.2 ∼Pacman()

```
Game::Pacman::∼Pacman ( )  [override], [default]
```

8.23.2 Member Function Documentation

8.23.2.1 getGameData()

```
IGame::GameDataType Game::Pacman::getGameData ( )  [override], [virtual]
```

Implements IGame.

8.23.2.2 handleEvent()

```
void Game::Pacman::handleEvent (
             std::string & name )  [override], [virtual]
```

Implements IGame.

8.23.2.3 handleRender()

```
void Game::Pacman::handleRender (
             IGraphicRenderer & renderer )  [override], [virtual]
```

Implements IGame.

8.23.2.4 handleUpdate()

```
void Game::Pacman::handleUpdate (
             int elapsedTime )  [override], [virtual]
```

Implements IGame.

8.23.2.5 setGameData()

```
void Game::Pacman::setGameData (
             IGame::GameDataType & data )  [override], [virtual]
```

Implements IGame.

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

- src/lib/game/pacman/Pacman.hpp
- src/lib/game/pacman/Draw.cpp
- src/lib/game/pacman/Map.cpp
- src/lib/game/pacman/MoveEnemy.cpp
- src/lib/game/pacman/MoveEntity.cpp
- src/lib/game/pacman/MovePlayer.cpp
- src/lib/game/pacman/Pacman.cpp
- src/lib/game/pacman/Pathfinding.cpp

8.24 Score::File::PlayerData Class Reference

```
#include <Score.hpp>
```

Public Member Functions

- PlayerData (const std::string &name, const long score)
- std::string toStr () const

Public Attributes

- std::string name
- long score

8.24.1 Detailed Description

PlayerData is a class the represents a player in-game with his score and his name.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 PlayerData()

```
Score::File::PlayerData::PlayerData (
            const std::string & name,
            const long score )
```

Initialize the class with his name and his score

Parameters

name	: The author's name
score	: The author's score

8.24.3 Member Function Documentation

8.24.3.1 toStr()

```
std::string Score::File::PlayerData::toStr ( ) const
```

Transform the internal information into a string

Returns

: A string formatted as: `Name = Score`

8.24.4 Member Data Documentation

8.24.4.1 name

```
std::string Score::File::PlayerData::name
```

Player's name

8.24.4.2 score

```
long Score::File::PlayerData::score
```

Player's score

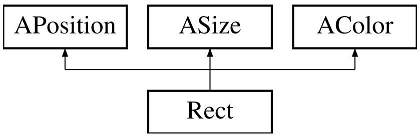
The documentation for this class was generated from the following files:

- src/core/score/Score.hpp
- src/core/score/File.cpp

8.25 Rect Class Reference

```
#include <Drawables.hpp>
```

Inheritance diagram for Rect:



Public Member Functions

- Rect (Vector2f pos=Vector2f(0, 0), Vector2f size=Vector2f(0, 0), Color color=Color(0, 0, 0, 0))
- ∼Rect ()=default

Additional Inherited Members

8.25.1 Constructor & Destructor Documentation

8.25.1.1 Rect()

```
Rect::Rect (
            Vector2f pos = Vector2f(0, 0),
            Vector2f size = Vector2f(0, 0),
            Color color = Color(0, 0, 0, 0) )  [inline]
```

8.25.1.2 ∼Rect()

```
Rect::∼Rect ( )  [default]
```

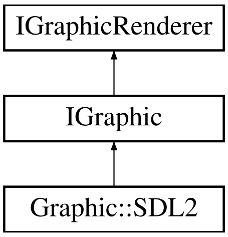
The documentation for this class was generated from the following file:

- src/graphic/Drawables.hpp

8.26 Graphic::SDL2 Class Reference

```
#include <SDL2.hpp>
```

Inheritance diagram for Graphic::SDL2:



Public Member Functions

- SDL2 ()
- ∼SDL2 () override
- void clearScreen () override
- void drawCircle (Circle circle) override
- void drawRect (Rect rect) override
- void drawScreen () override
- void drawSprite (Sprite sprite) override
- void drawText (Text text) override
- std::string handleEvent () override
- bool isOperational () override

Static Public Attributes

- static const uint WINDOW_WIDTH = 800
- static const uint WINDOW_HEIGHT = 800

8.26.1 Constructor & Destructor Documentation

8.26.1.1 SDL2()

```
Graphic::SDL2::SDL2 ( )
```

8.26.1.2 ∼SDL2()

```
Graphic::SDL2::∼SDL2 ( ) [override]
```

8.26.2 Member Function Documentation

8.26.2.1 clearScreen()

```
void Graphic::SDL2::clearScreen ( ) [override], [virtual]
```

Implements IGraphicRenderer.

8.26.2.2 drawCircle()

```
void Graphic::SDL2::drawCircle (
            Circle circle ) [override], [virtual]
```

Implements IGraphicRenderer.

8.26.2.3 drawRect()

```
void Graphic::SDL2::drawRect (
            Rect rect ) [override], [virtual]
```

Implements IGraphicRenderer.

8.26.2.4 drawScreen()

```
void Graphic::SDL2::drawScreen ( )  [override], [virtual]
```

Draw all visual elements on the screen. Up to each library graphical library to handle the way they manages their graphical entities.

Implements IGraphicRenderer.

8.26.2.5 drawSprite()

```
void Graphic::SDL2::drawSprite (
            Sprite sprite )  [override], [virtual]
```

Implements IGraphicRenderer.

8.26.2.6 drawText()

```
void Graphic::SDL2::drawText (
            Text text )  [override], [virtual]
```

Implements IGraphicRenderer.

8.26.2.7 handleEvent()

```
std::string Graphic::SDL2::handleEvent ( )  [override], [virtual]
```

The function handle keyboard events

Returns

A string with the latest event. The string must be one element of the structure KeyboardEvent_s.

Implements IGraphic.

8.26.2.8 isOperational()

```
bool Graphic::SDL2::isOperational ( )  [override], [virtual]
```

Implements IGraphic.

8.26.3 Member Data Documentation

8.26.3.1 WINDOW_HEIGHT

```
const uint Graphic::SDL2::WINDOW_HEIGHT = 800  [static]
```

8.26.3.2 WINDOW_WIDTH

```
const uint Graphic::SDL2::WINDOW_WIDTH = 800  [static]
```

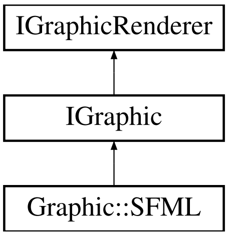
The documentation for this class was generated from the following files:

- src/lib/graphic/sdl2/SDL2.hpp
- src/lib/graphic/sdl2/SDL2.cpp

8.27 Graphic::SFML Class Reference

```
#include <SFML.hpp>
```

Inheritance diagram for Graphic::SFML:



Public Member Functions

- SFML ()
- ∼SFML () override=default
- void clearScreen () override
- void drawCircle (Circle circle) override
- void drawRect (Rect rect) override
- void drawScreen () override
- void drawSprite (Sprite sprite) override
- void drawText (Text text) override
- std::string handleEvent () override
- bool isOperational () override

Static Public Attributes

- static const uint WINDOW_WIDTH = 800
- static const uint WINDOW_HEIGHT = 800

8.27.1 Constructor & Destructor Documentation

8.27.1.1 SFML()

```
Graphic::SFML::SFML ( )
```

Basic constructor for SFML. It creates the window with some specific parameters (framerate, window's name, ...).

8.27.1.2 ∼SFML()

```
Graphic::SFML::∼SFML ( ) [override], [default]
```

8.27.2 Member Function Documentation

8.27.2.1 clearScreen()

```
void Graphic::SFML::clearScreen ( ) [override], [virtual]
```

Implements IGraphicRenderer.

8.27.2.2 drawCircle()

```
void Graphic::SFML::drawCircle (
            Circle circle ) [override], [virtual]
```

Implements IGraphicRenderer.

8.27.2.3 drawRect()

```
void Graphic::SFML::drawRect (
            Rect rect ) [override], [virtual]
```

Implements IGraphicRenderer.

8.27.2.4 drawScreen()

```
void Graphic::SFML::drawScreen ( )  [override], [virtual]
```

Draw all visual elements on the screen. Up to each library graphical library to handle the way they manages their graphical entities.

Implements IGraphicRenderer.

8.27.2.5 drawSprite()

```
void Graphic::SFML::drawSprite (
             Sprite sprite )  [override], [virtual]
```

Implements IGraphicRenderer.

8.27.2.6 drawText()

```
void Graphic::SFML::drawText (
             Text text )  [override], [virtual]
```

Implements IGraphicRenderer.

8.27.2.7 handleEvent()

```
std::string Graphic::SFML::handleEvent ( )  [override], [virtual]
```

The function handle keyboard events

Returns

A string with the latest event. The string must be one element of the structure KeyboardEvent_s.

Implements IGraphic.

8.27.2.8 isOperational()

```
bool Graphic::SFML::isOperational ( )  [override], [virtual]
```

Implements IGraphic.

8.27.3 Member Data Documentation

8.27.3.1 WINDOW_HEIGHT

```
const uint Graphic::SFML::WINDOW_HEIGHT = 800  [static]
```

8.27.3.2 WINDOW_WIDTH

```
const uint Graphic::SFML::WINDOW_WIDTH = 800  [static]
```

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

- src/lib/graphic/sfml/SFML.hpp
- src/lib/graphic/sfml/SFML.cpp

8.28 SoLoader::SoLoader< T > Class Template Reference

```
#include <SoLoader.hpp>
```

Public Member Functions

- SoLoader ()
- void changeSo (const std::string &DLLPath)
- ∼SoLoader ()
- T ∗ operator-> ()
- T ∗ getInstance ()
- std::string getLibPath ()

Static Public Attributes

- static constexpr const char ∗ ENTRY_POINT_NAME = "entry"

8.28.1 Detailed Description

template<typename T>
class SoLoader::SoLoader< T >

The class allows to load a library and associate a type determined by the template T.

Template Parameters

T	Type which will be associated to the library.

8.28.2 Constructor & Destructor Documentation

8.28.2.1 SoLoader()

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

8.28.2.2 ∼SoLoader()

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

The destructor closes the current loaded library if there is one

8.28.3 Member Function Documentation

8.28.3.1 changeSo()

```
template<typename T >
void SoLoader::SoLoader< T >::changeSo (
             const std::string & DLLPath )  [inline]
```

Change the current loaded library to the new one specified by the parameter. If there was already a library loaded, it closes it before loading the new one.

Parameters

DLLPath	: The new library's path.

Exceptions

SoLoader::Exceptions::InvalidSO	
SoLoader::Exceptions::InvalidEntryPoint	

8.28.3.2 getInstance()

```
template<typename T >
T* SoLoader::SoLoader< T >::getInstance ( )  [inline]
```

Get the current instance loaded

Returns

The current instance

8.28.3.3 getLibPath()

```
template<typename T >
std::string SoLoader::SoLoader< T >::getLibPath ( )  [inline]
```

Get the library's path loaded

Returns

A string to the library's path

8.28.3.4 operator->()

```
template<typename T >
T* SoLoader::SoLoader< T >::operator-> ( )  [inline]
```

Overloading the -> operator to access to the members functions of the T type.

Warning

It leads to a crash if there is no instance loaded

Returns

The current instance loaded.

8.28.4 Member Data Documentation

8.28.4.1 ENTRY_POINT_NAME

```
template<typename T >
constexpr const char* SoLoader::SoLoader< T >::ENTRY_POINT_NAME = "entry"  [static], [constexpr]
```

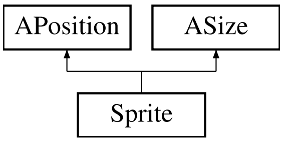
The documentation for this class was generated from the following file:

- src/soLoader/SoLoader.hpp

8.29 Sprite Class Reference

```
#include <Drawables.hpp>
```

Inheritance diagram for Sprite:



Public Member Functions

- Sprite (const std::string &name, Vector2f pos=Vector2f(0, 0), Vector2f size=Vector2f(0, 0))
- ∼Sprite ()=default
- const std::string & getTextureName (void) const

Additional Inherited Members

8.29.1 Constructor & Destructor Documentation

8.29.1.1 Sprite()

```
Sprite::Sprite (
            const std::string & name,
            Vector2f pos = Vector2f(0, 0),
            Vector2f size = Vector2f(0, 0) )  [inline]
```

8.29.1.2 ∼Sprite()

```
Sprite::∼Sprite ( )  [default]
```

8.29.2 Member Function Documentation

8.29.2.1 getTextureName()

```
const std::string& Sprite::getTextureName (
            void  ) const  [inline]
```

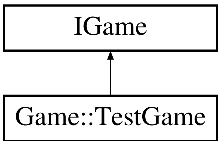
The documentation for this class was generated from the following file:

- src/graphic/Drawables.hpp

8.30 Game::TestGame Class Reference

```
#include <TestGame.hpp>
```

Inheritance diagram for Game::TestGame:



Public Member Functions

- TestGame ()=default
- ∼TestGame () override=default
- void handleEvent (std::string &name) override
- void handleRender (IGraphicRenderer &renderer) override
- void handleUpdate (int elapsedTime) override
- void setGameData (GameDataType &data) override
- GameDataType getGameData () override

Additional Inherited Members

8.30.1 Constructor & Destructor Documentation

8.30.1.1 TestGame()

```
Game::TestGame::TestGame ( )  [default]
```

8.30.1.2 ∼TestGame()

```
Game::TestGame::∼TestGame ( )  [override], [default]
```

8.30.2 Member Function Documentation

8.30.2.1 getGameData()

```
IGame::GameDataType Game::TestGame::getGameData ( )  [override], [virtual]
```

Implements IGame.

8.30.2.2 handleEvent()

```
void Game::TestGame::handleEvent (
            std::string & name )  [override], [virtual]
```

Implements IGame.

8.30.2.3 handleRender()

```
void Game::TestGame::handleRender (
            IGraphicRenderer & renderer )  [override], [virtual]
```

Implements IGame.

8.30.2.4 handleUpdate()

```
void Game::TestGame::handleUpdate (
            int elapsedTime )  [override], [virtual]
```

Implements IGame.

8.30.2.5 setGameData()

```
void Game::TestGame::setGameData (
            IGame::GameDataType & data )  [override], [virtual]
```

Implements IGame.

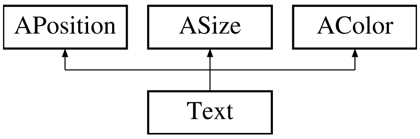
The documentation for this class was generated from the following files:

- src/lib/game/test/TestGame.hpp
- src/lib/game/test/TestGame.cpp

8.31 Text Class Reference

```
#include <Drawables.hpp>
```

Inheritance diagram for Text:



Public Member Functions

- Text (const std::string &text, Vector2f pos=Vector2f(0, 0), Vector2f size=Vector2f(0, 0), Color color=Color(0, 0, 0, 0))
- ∼Text ()=default
- const std::string & getText (void) const
- void setText (const std::string &newText)

Additional Inherited Members

8.31.1 Constructor & Destructor Documentation

8.31.1.1 Text()

```
Text::Text (
            const std::string & text,
            Vector2f pos = Vector2f(0, 0),
            Vector2f size = Vector2f(0, 0),
            Color color = Color(0, 0, 0, 0) )  [inline]
```

8.31.1.2 ∼Text()

```
Text::∼Text ( )  [default]
```

8.31.2 Member Function Documentation

8.31.2.1 getText()

```
const std::string& Text::getText (
             void  ) const  [inline]
```

8.31.2.2 setText()

```
void Text::setText (
             const std::string & newText )  [inline]
```

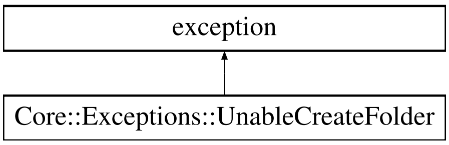
The documentation for this class was generated from the following file:

- src/graphic/Drawables.hpp

8.32 Core::Exceptions::UnableCreateFolder Class Reference

```
#include <Core.hpp>
```

Inheritance diagram for Core::Exceptions::UnableCreateFolder:



Public Member Functions

- UnableCreateFolder () noexcept=default
- const char ∗ what () const noexcept override

8.32.1 Detailed Description

UnableCreateFolder is an exception throw when the program cannot create the folder which will gather score files together.

8.32.2 Constructor & Destructor Documentation

8.32.2.1 UnableCreateFolder()

```
Core::Exceptions::UnableCreateFolder::UnableCreateFolder ( )  [default], [noexcept]
```

8.32.3 Member Function Documentation

8.32.3.1 what()

```
const char * Core::Exceptions::UnableCreateFolder::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred.

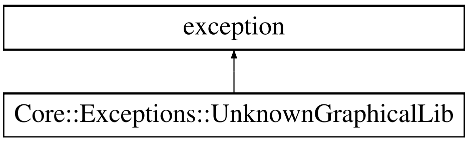
The documentation for this class was generated from the following files:

- src/core/Core.hpp
- src/core/Exceptions.cpp

8.33 Core::Exceptions::UnknownGraphicalLib Class Reference

```
#include <Core.hpp>
```

Inheritance diagram for Core::Exceptions::UnknownGraphicalLib:



Public Member Functions

- UnknownGraphicalLib (const std::string &name) noexcept
- UnknownGraphicalLib ()=delete
- const char ∗ what () const noexcept override

8.33.1 Detailed Description

UnknownGraphicalLib is a minor exception throw when the initial graphical library has been correctly loaded but cannot be found anymore, anywhere.

8.33.2 Constructor & Destructor Documentation

8.33.2.1 UnknownGraphicalLib() [1/2]

```
Core::Exceptions::UnknownGraphicalLib::UnknownGraphicalLib (
            const std::string & name )  [explicit], [noexcept]
```

8.33.2.2 UnknownGraphicalLib() [2/2]

```
Core::Exceptions::UnknownGraphicalLib::UnknownGraphicalLib ( )  [delete]
```

8.33.3 Member Function Documentation

8.33.3.1 what()

```
const char * Core::Exceptions::UnknownGraphicalLib::what ( ) const  [override], [noexcept]
```

Required by std::exception

Returns

An explicit message explaining why the error occurred.

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

- src/core/Core.hpp
- src/core/Exceptions.cpp

8.34 Vector2< T > Struct Template Reference

```
#include <Drawables.hpp>
```

Public Member Functions

- Vector2 (T pX, T pY)
- Vector2 operator+ (Vector2 const &with) const
- Vector2 & operator+= (Vector2 const &with)
- Vector2 operator- (Vector2 const &with) const
- Vector2 & operator-= (Vector2 const &with)
- bool operator== (Vector2 const &a) const

Public Attributes

- T x
- T y

8.34.1 Constructor & Destructor Documentation

8.34.1.1 Vector2()

```
template<class T >
Vector2< T >::Vector2 (
             T pX,
             T pY )  [inline]
```

8.34.2 Member Function Documentation

8.34.2.1 operator+()

```
template<class T >
Vector2 Vector2< T >::operator+ (
             Vector2< T > const & with ) const  [inline]
```

8.34.2.2 operator+=()

```
template<class T >
Vector2& Vector2< T >::operator+= (
             Vector2< T > const & with )  [inline]
```

8.34.2.3 operator-()

```
template<class T >
Vector2 Vector2< T >::operator- (
             Vector2< T > const & with ) const  [inline]
```

8.34.2.4 operator-=()

```
template<class T >
Vector2& Vector2< T >::operator-= (
             Vector2< T > const & with )  [inline]
```

8.34.2.5 operator==()

```
template<class T >
bool Vector2< T >::operator== (
             Vector2< T > const & a ) const  [inline]
```

8.34.3 Member Data Documentation

8.34.3.1 x

```
template<class T >
T Vector2< T >::x
```

8.34.3.2 y

```
template<class T >
T Vector2< T >::y
```

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

- src/graphic/Drawables.hpp

8.35 Vector3< T > Struct Template Reference

```
#include <Drawables.hpp>
```

Public Member Functions

- Vector3 (T pX, T pY, T pZ)
- Vector3 operator+ (Vector3 const &with) const
- Vector3 & operator+= (Vector3 const &with)
- Vector3 operator- (Vector3 const &with) const
- Vector3 & operator-= (Vector3 const &with)
- bool operator== (Vector3 const &a) const

Public Attributes

- T x
- T y
- T z

8.35.1 Constructor & Destructor Documentation

8.35.1.1 Vector3()

```
template<class T >
Vector3< T >::Vector3 (
             T pX,
             T pY,
             T pZ )  [inline]
```

8.35.2 Member Function Documentation

8.35.2.1 operator+()

```
template<class T >
Vector3 Vector3< T >::operator+ (
             Vector3< T > const & with ) const  [inline]
```

8.35.2.2 operator+=()

```
template<class T >
Vector3& Vector3< T >::operator+= (
             Vector3< T > const & with )  [inline]
```

8.35.2.3 operator-()

```
template<class T >
Vector3 Vector3< T >::operator- (
             Vector3< T > const & with ) const  [inline]
```

8.35.2.4 operator-=()

```
template<class T >
Vector3& Vector3< T >::operator-= (
             Vector3< T > const & with )  [inline]
```

8.35.2.5 operator==()

```
template<class T >
bool Vector3< T >::operator== (
             Vector3< T > const & a ) const  [inline]
```

8.35.3 Member Data Documentation

8.35.3.1 x

```
template<class T >
T Vector3< T >::x
```

8.35.3.2 y

```
template<class T >
T Vector3< T >::y
```

8.35.3.3 z

```
template<class T >
T Vector3< T >::z
```

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

- src/graphic/Drawables.hpp

Chapter 9

File Documentation

9.1 CMakeLists.txt File Reference

9.2 README.md File Reference

9.3 src/core/Core.cpp File Reference

```
#include <iostream>
#include <algorithm>
#include <chrono>
#include <sys/stat.h>
#include <fcntl.h>
#include <dirent.h>
#include "soLoader/SoLoader.hpp"
#include "Core.hpp"
```

9.4 src/core/Core.hpp File Reference

```
#include <string>
#include <list>
#include <core/score/Score.hpp>
#include <experimental/filesystem>
#include "game/IGame.hpp"
#include "graphic/IGraphic.hpp"
#include "soLoader/SoLoader.hpp"
```

Classes

- class Core::Core
- class Core::Exceptions::UnableCreateFolder
- class Core::Exceptions::MissingMandatoryFolder
- class Core::Exceptions::EmptyMandatoryFolder
- class Core::Exceptions::InvalidScorePath
- class Core::Exceptions::UnknownGraphicalLib
- class Core::Exceptions::ExitGame

Namespaces

- Core
- Core::Exceptions

Macros

- #define IS_IN_GAME(a) (a->_selection["games"] == -1)

9.4.1 Macro Definition Documentation

9.4.1.1 IS_IN_GAME

```
#define IS_IN_GAME(
              a ) (a->_selection["games"] == -1)
```

9.5 src/core/Exceptions.cpp File Reference

```
#include "Core.hpp"
```

9.6 src/core/score/Exceptions.cpp File Reference

```
#include "core/score/Score.hpp"
```

9.7 src/soLoader/Exceptions.cpp File Reference

```
#include "SoLoader.hpp"
#include "Exceptions.hpp"
```

9.8 src/core/KeyEvents.cpp File Reference

```
#include "Core.hpp"
```

9.9 src/core/menu/Menu.cpp File Reference

```
#include "core/Core.hpp"
#include "core/score/Score.hpp"
```

9.10 src/core/menu/Strip.cpp File Reference

```
#include "core/Core.hpp"
```

Variables

- static const Vector2f DEFAULT_STRIP_TEXT_SIZE = {30.f, 5.f}

9.10.1 Variable Documentation

9.10.1.1 DEFAULT_STRIP_TEXT_SIZE

```
const Vector2f DEFAULT_STRIP_TEXT_SIZE = {30.f, 5.f}  [static]
```

9.11 src/core/menu/Utils.cpp File Reference

```
#include <cstring>
#include <fstream>
#include "core/Core.hpp"
```

9.12 src/core/score/File.cpp File Reference

```
#include <algorithm>
#include "core/score/Score.hpp"
```

9.13 src/core/score/Game.cpp File Reference

```
#include <fstream>
#include "core/Core.hpp"
#include "Score.hpp"
```

9.14 src/core/score/Score.hpp File Reference

```
#include <string>
#include <list>
```

Classes

- class Score::Game
- class Score::File
- class Score::File::PlayerData
- class Score::Exceptions::InvalidFile

Namespaces

- Score
- Score::Exceptions

9.15 src/game/IGame.hpp File Reference

```
#include <map>
#include <string>
#include "graphic/IGraphic.hpp"
```

Classes

- class IGame

9.16 src/graphic/Drawables.hpp File Reference

```
#include <string>
```

Classes

- struct Vector2< T >
- struct Vector3< T >
- struct Color
- class APosition
- class ASize
- class AColor
- class Rect
- class Circle
- class Text
- class Sprite

Typedefs

- typedef Vector2< float > Vector2f
- typedef Vector2< unsigned int > Vector2u
- typedef Vector2< int > Vector2i
- typedef Vector3< float > Vector3f
- typedef Vector3< unsigned int > Vector3u
- typedef Vector3< int > Vector3i

9.16.1 Typedef Documentation

9.16.1.1 Vector2f

```
typedef Vector2<float> Vector2f
```

9.16.1.2 Vector2i

```
typedef Vector2<int> Vector2i
```

9.16.1.3 Vector2u

```
typedef Vector2<unsigned int> Vector2u
```

9.16.1.4 Vector3f

```
typedef Vector3<float> Vector3f
```

9.16.1.5 Vector3i

```
typedef Vector3<int> Vector3i
```

9.16.1.6 Vector3u

```
typedef Vector3<unsigned int> Vector3u
```

9.17 src/graphic/IGraphic.hpp File Reference

```
#include "lib/graphic/Exceptions.hpp"
#include "Drawables.hpp"
```

Classes

- struct KeyboardEvent_s
- class IGraphicRenderer
- class IGraphic

Macros

- #define FONT_FILENAME "/usr/share/fonts/liberation/LiberationMono-Regular.ttf"
- #define PERCENTAGE(a) (a / 100.f)

9.17.1 Macro Definition Documentation

9.17.1.1 FONT_FILENAME

```
#define FONT_FILENAME "/usr/share/fonts/liberation/LiberationMono-Regular.ttf"
```

Common macro defined between groups.

9.17.1.2 PERCENTAGE

```
#define PERCENTAGE(
             a ) (a / 100.f)
```

9.18 src/lib/game/nibbler/Draw.cpp File Reference

```
#include "Nibbler.hpp"
```

Variables

- static const Vector2f DEFAULT_TEXT_SIZE = {35.f, 5.f}

9.18.1 Variable Documentation

9.18.1.1 DEFAULT_TEXT_SIZE

```
const Vector2f DEFAULT_TEXT_SIZE = {35.f, 5.f}  [static]
```

9.19 src/lib/game/pacman/Draw.cpp File Reference

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

Variables

- static const Vector2f DEFAULT_TEXT_SIZE = {10.f, 5.f}

9.19.1 Variable Documentation

9.19.1.1 DEFAULT_TEXT_SIZE

```
const Vector2f DEFAULT_TEXT_SIZE = {10.f, 5.f}  [static]
```

9.20 src/lib/game/nibbler/Nibbler.cpp File Reference

```
#include <algorithm>
#include <iostream>
#include <random>
#include <array>
#include "Nibbler.hpp"
```

Functions

- IGame * entry ()
- __attribute__ ((constructor)) void load()
- __attribute__ ((destructor)) void unload()

Variables

- static Game::Nibbler * instance

9.20.1 Function Documentation

9.20.1.1 __attribute__() [1/2]

```
__attribute__ (
              (constructor)   )
```

9.20.1.2 __attribute__() [2/2]

```
__attribute__ (
              (destructor)   )
```

9.20.1.3 entry()

```
IGame* entry ( )
```

9.20.2 Variable Documentation

9.20.2.1 instance

```
Game::Nibbler* instance  [static]
```

9.21 src/lib/game/nibbler/Nibbler.hpp File Reference

```
#include <vector>
#include <list>
#include "graphic/Drawables.hpp"
#include "game/IGame.hpp"
#include "core/score/Score.hpp"
```

Classes

- class Game::Nibbler

Namespaces

- Game

Macros

- #define IS_GAME_IN_PROGRESS(a) (a->_state == Nibbler::GAME_STATE::GAME)

9.21.1 Macro Definition Documentation

9.21.1.1 IS_GAME_IN_PROGRESS

```
#define IS_GAME_IN_PROGRESS(
              a ) (a->_state == Nibbler::GAME_STATE::GAME)
```

9.22 src/lib/game/pacman/Map.cpp File Reference

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

9.23 src/lib/game/pacman/MoveEnemy.cpp File Reference

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

9.24 src/lib/game/pacman/MoveEntity.cpp File Reference

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

9.25 src/lib/game/pacman/MovePlayer.cpp File Reference

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

9.26 src/lib/game/pacman/Pacman.cpp File Reference

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

Functions

- IGame ∗ entry ()
- __attribute__ ((constructor)) void load()
- __attribute__ ((destructor)) void unload()

Variables

- static Game::Pacman ∗ instance

9.26.1 Function Documentation

9.26.1.1 __attribute__() [1/2]

```
__attribute__ (
              (constructor) )
```

9.26.1.2 __attribute__() [2/2]

```
__attribute__ (
              (destructor) )
```

9.26.1.3 entry()

```
IGame* entry ( )
```

9.26.2 Variable Documentation

9.26.2.1 instance

```
Game::Pacman* instance  [static]
```

9.27 src/lib/game/pacman/Pacman.hpp File Reference

```
#include <vector>
#include <list>
#include <core/score/Score.hpp>
#include "game/IGame.hpp"
```

Classes

- class Game::Pacman

Namespaces

- Game

Macros

- #define IS_GAME_IN_PROGRESS(a) (a->_state == Pacman::GAME_STATE::GAME)

9.27.1 Macro Definition Documentation

9.27.1.1 IS_GAME_IN_PROGRESS

```
#define IS_GAME_IN_PROGRESS(
            a ) (a->_state == Pacman::GAME_STATE::GAME)
```

9.28 src/lib/game/pacman/Pathfinding.cpp File Reference

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

9.29 src/lib/game/test/TestGame.cpp File Reference

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

Functions

- IGame ∗ entry ()
- __attribute__ ((constructor)) void load()
- __attribute__ ((destructor)) void unload()

Variables

- static Game::TestGame ∗ instance

9.29.1 Function Documentation

9.29.1.1 __attribute__() [1/2]

```
__attribute__ (
              (constructor)  )
```

9.29.1.2 __attribute__() [2/2]

```
__attribute__ (
              (destructor)  )
```

9.29.1.3 entry()

```
IGame* entry ( )
```

9.29.2 Variable Documentation

9.29.2.1 instance

```
Game::TestGame* instance  [static]
```

9.30 src/lib/game/test/TestGame.hpp File Reference

```
#include "game/IGame.hpp"
```

Classes

- class Game::TestGame

Namespaces

- Game

9.31 src/lib/graphic/Exceptions.hpp File Reference

```
#include <string>
```

Classes

- class Graphic::Exceptions::LoadFontFailed

Namespaces

- Graphic
- Graphic::Exceptions

9.32 src/soLoader/Exceptions.hpp File Reference

```
#include <string>
```

Classes

- class SoLoader::Exceptions::InvalidSO
- class SoLoader::Exceptions::InvalidEntryPoint

Namespaces

- SoLoader
- SoLoader::Exceptions

9.33 src/lib/graphic/ncurses/Ncurses.cpp File Reference

```
#include <cstring>
#include <chrono>
#include <algorithm>
#include <thread>
#include <cmath>
#include "core/Core.hpp"
#include "Ncurses.hpp"
```

Functions

- IGraphic ∗ entry ()
- __attribute__ ((constructor)) void load()
- __attribute__ ((destructor)) void unload()

Variables

- static std::shared_ptr< Graphic::Ncurses ∗ > instance

9.33.1 Function Documentation

9.33.1.1 __attribute__() [1/2]

```
__attribute__ (
              (constructor)  )
```

9.33.1.2 __attribute__() [2/2]

```
__attribute__ (
              (destructor)  )
```

9.33.1.3 entry()

```
IGraphic* entry ( )
```

9.33.2 Variable Documentation

9.33.2.1 instance

```
std::shared_ptr<Graphic::Ncurses *> instance  [static]
```

9.34 src/lib/graphic/ncurses/Ncurses.hpp File Reference

```
#include "graphic/IGraphic.hpp"
#include <curses.h>
```

Classes

- class Graphic::Ncurses

Namespaces

- Graphic

Macros

- #define KEY_ESCAPE 27
- #define CUSTOM_KEY_ENTER 10

9.34.1 Macro Definition Documentation

9.34.1.1 CUSTOM_KEY_ENTER

```
#define CUSTOM_KEY_ENTER 10
```

Since the KEY_ENTER NCurses macro doesn't correspond to the actual ENTER key, we define a custom one.

9.34.1.2 KEY_ESCAPE

```
#define KEY_ESCAPE 27
```

The KEY_ESCAPE macro doesn't exists in NCurses, we add it.

9.35 src/lib/graphic/sdl2/SDL2.cpp File Reference

```
#include <iostream>
#include "core/Core.hpp"
#include "SDL2.hpp"
```

Functions

- IGraphic ∗ entry ()
- __attribute__ ((constructor)) void load()
- __attribute__ ((destructor)) void unload()

Variables

- static std::shared_ptr< Graphic::SDL2 ∗ > instance

9.35.1 Function Documentation

9.35.1.1 __attribute__() [1/2]

```
__attribute__ (
             (constructor)  )
```

9.35.1.2 __attribute__() [2/2]

```
__attribute__ (
             (destructor)  )
```

9.35.1.3 entry()

```
IGraphic* entry ( )
```

9.35.2 Variable Documentation

9.35.2.1 instance

```
std::shared_ptr<Graphic::SDL2 *> instance  [static]
```

9.36 src/lib/graphic/sdl2/SDL2.hpp File Reference

```
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include <vector>
#include <iostream>
#include "graphic/IGraphic.hpp"
```

Classes

- class Graphic::SDL2

Namespaces

- Graphic

9.37 src/lib/graphic/sfml/SFML.cpp File Reference

```
#include "lib/graphic/Exceptions.hpp"
#include "SFML.hpp"
#include <memory>
```

Functions

- IGraphic ∗ entry ()
- __attribute__ ((constructor)) void load()
- __attribute__ ((destructor)) void unload()

Variables

- static std::shared_ptr< Graphic::SFML ∗ > instance

9.37.1 Function Documentation

9.37.1.1 __attribute__() [1/2]

```
__attribute__ (
            (constructor) )
```

9.37.1.2 __attribute__() [2/2]

```
__attribute__ (
            (destructor) )
```

9.37.1.3 entry()

```
IGraphic* entry ( )
```

9.37.2 Variable Documentation

9.37.2.1 instance

```
std::shared_ptr<Graphic::SFML *> instance  [static]
```

9.38 src/lib/graphic/sfml/SFML.hpp File Reference

```
#include <memory>
#include <SFML/Graphics.hpp>
#include "graphic/IGraphic.hpp"
```

Classes

- class Graphic::SFML

Namespaces

- Graphic

9.39 src/Main.cpp File Reference

```
#include <iostream>
#include "core/Core.hpp"
#include "soLoader/SoLoader.hpp"
```

Functions

- int main (const int ac, const char **av)

9.39.1 Function Documentation

9.39.1.1 main()

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

9.40 src/soLoader/SoLoader.hpp File Reference

```
#include <string>
#include <dlfcn.h>
#include <iostream>
#include "soLoader/Exceptions.hpp"
```

Classes

- class SoLoader::SoLoader< T >

Namespaces

- SoLoader

Index