

## Arcade Project

Generated by Doxygen 1.8.18



<b>1 OOP_arcade_2019</b>	<b>1</b>
1.1 Introduction	1
1.2 Add a dynamic library	1
1.3 Collaboration	1
<b>2 OOP_arcade_2019</b>	<b>3</b>
2.1 Introduction	3
2.2 Add a dynamic library	3
2.3 Installation	3
2.3.1 SFML installation:	3
2.3.2 SDL installation:	3
2.3.3 Documentation with Doxygen:	4
2.4 Collaboration	4
<b>3 Namespace Index</b>	<b>5</b>
3.1 Namespace List	5
<b>4 Hierarchical Index</b>	<b>7</b>
4.1 Class Hierarchy	7
<b>5 Class Index</b>	<b>9</b>
5.1 Class List	9
<b>6 File Index</b>	<b>11</b>
6.1 File List	11
<b>7 Namespace Documentation</b>	<b>13</b>
7.1 Core Namespace Reference	13
7.2 Core::Exceptions Namespace Reference	13
7.3 Game Namespace Reference	13
7.4 Graphic Namespace Reference	14
7.4.1 Detailed Description	14
7.5 Graphic::Exceptions Namespace Reference	14
7.6 Score Namespace Reference	14
7.6.1 Detailed Description	14
7.7 Score::Exceptions Namespace Reference	15
7.8 SoLoader Namespace Reference	15
7.9 SoLoader::Exceptions Namespace Reference	15
7.9.1 Detailed Description	15
<b>8 Class Documentation</b>	<b>17</b>
8.1 AColor Class Reference	17
8.1.1 Constructor & Destructor Documentation	17
8.1.1.1 AColor()	18
8.1.1.2 ~AColor()	18

8.1.2 Member Function Documentation	18
8.1.2.1 getColor()	18
8.1.2.2 getColorAlpha()	18
8.1.2.3 getColorBlue()	18
8.1.2.4 getColorGreen()	18
8.1.2.5 getColorRed()	19
8.1.2.6 setColor()	19
8.1.2.7 setColorAlpha()	19
8.1.2.8 setColorBlue()	19
8.1.2.9 setColorGreen()	19
8.1.2.10 setColorRed()	19
8.1.3 Member Data Documentation	19
8.1.3.1 _color	20
8.2 APosition Class Reference	20
8.2.1 Constructor & Destructor Documentation	20
8.2.1.1 APosition()	20
8.2.1.2 ~APosition()	21
8.2.2 Member Function Documentation	21
8.2.2.1 getPosition()	21
8.2.2.2 getPositionX()	21
8.2.2.3 getPositionY()	21
8.2.2.4 move()	21
8.2.2.5 moveX()	21
8.2.2.6 moveY()	22
8.2.2.7 setPosition()	22
8.2.2.8 setPositionX()	22
8.2.2.9 setPositionY()	22
8.2.3 Member Data Documentation	22
8.2.3.1 _pos	22
8.3 ASize Class Reference	22
8.3.1 Constructor & Destructor Documentation	23
8.3.1.1 ASize()	23
8.3.1.2 ~ASize()	23
8.3.2 Member Function Documentation	23
8.3.2.1 getSize()	23
8.3.2.2 getSizeX()	23
8.3.2.3 getSizeY()	24
8.3.2.4 setSize()	24
8.3.2.5 setSizeX()	24
8.3.2.6 setSizeY()	24
8.3.3 Member Data Documentation	24
8.3.3.1 _size	24

8.4 Circle Class Reference	24
8.4.1 Constructor & Destructor Documentation	25
8.4.1.1 Circle()	25
8.4.1.2 ~Circle()	25
8.5 Color Struct Reference	25
8.5.1 Constructor & Destructor Documentation	26
8.5.1.1 Color()	26
8.5.2 Member Function Documentation	26
8.5.2.1 Black()	26
8.5.2.2 Blue()	26
8.5.2.3 Green()	26
8.5.2.4 operator==()	26
8.5.2.5 Red()	27
8.5.2.6 Transparent()	27
8.5.2.7 White()	27
8.5.3 Member Data Documentation	27
8.5.3.1 alpha	27
8.5.3.2 blue	27
8.5.3.3 green	27
8.5.3.4 red	27
8.6 Core::Core Class Reference	28
8.6.1 Detailed Description	28
8.6.2 Constructor & Destructor Documentation	28
8.6.2.1 Core()	28
8.6.2.2 ~Core()	29
8.6.3 Member Function Documentation	29
8.6.3.1 loadScore()	29
8.6.3.2 run()	29
8.6.3.3 useGame()	30
8.6.3.4 useGraphic()	30
8.6.4 Member Data Documentation	30
8.6.4.1 FRAME_PER_TIME	30
8.6.4.2 SCORE_PATH	31
8.7 Core::Exceptions::EmptyMandatoryFolder Class Reference	31
8.7.1 Detailed Description	31
8.7.2 Constructor & Destructor Documentation	31
8.7.2.1 EmptyMandatoryFolder() [1/2]	32
8.7.2.2 EmptyMandatoryFolder() [2/2]	32
8.7.3 Member Function Documentation	32
8.7.3.1 what()	32
8.8 Core::Exceptions::ExitGame Class Reference	32
8.8.1 Detailed Description	33

8.8.2 Constructor & Destructor Documentation	33
8.8.2.1 ExitGame()	33
8.8.3 Member Function Documentation	33
8.8.3.1 what()	33
8.9 Score::File Class Reference	33
8.9.1 Constructor & Destructor Documentation	34
8.9.1.1 File() [1/2]	34
8.9.1.2 File() [2/2]	35
8.9.2 Member Function Documentation	35
8.9.2.1 addEntry()	35
8.9.2.2 getListUser()	35
8.10 Score::Game Class Reference	35
8.10.1 Detailed Description	36
8.10.2 Constructor & Destructor Documentation	36
8.10.2.1 Game()	36
8.10.3 Member Function Documentation	36
8.10.3.1 addLetter()	36
8.10.3.2 getAuthor()	37
8.10.3.3 getLetter()	37
8.10.3.4 removeLastLetter()	37
8.10.3.5 reset()	38
8.10.3.6 save()	38
8.10.3.7 setGame()	38
8.10.3.8 setLetter()	38
8.10.3.9 setScore()	39
8.10.4 Member Data Documentation	39
8.10.4.1 MAX_AUTHOR_NAME	39
8.11 IGame Class Reference	39
8.11.1 Member Typedef Documentation	40
8.11.1.1 GameDataType	40
8.11.2 Constructor & Destructor Documentation	40
8.11.2.1 ~IGame()	40
8.11.3 Member Function Documentation	40
8.11.3.1 getGameData()	40
8.11.3.2 handleEvent()	41
8.11.3.3 handleRender()	41
8.11.3.4 handleUpdate()	41
8.11.3.5 setGameData()	41
8.12 IGraphic Class Reference	41
8.12.1 Constructor & Destructor Documentation	42
8.12.1.1 IGraphic() [1/2]	42
8.12.1.2 IGraphic() [2/2]	42

8.12.1.3 ~IGraphic()	42
8.12.2 Member Function Documentation	42
8.12.2.1 handleEvent()	42
8.12.2.2 isOperational()	43
8.13 IGraphicRenderer Class Reference	43
8.13.1 Detailed Description	43
8.13.2 Constructor & Destructor Documentation	43
8.13.2.1 ~IGraphicRenderer()	43
8.13.3 Member Function Documentation	44
8.13.3.1 clearScreen()	44
8.13.3.2 drawCircle()	44
8.13.3.3 drawRect()	44
8.13.3.4 drawScreen()	44
8.13.3.5 drawSprite()	44
8.13.3.6 drawText()	45
8.13.3.7 operator=()	45
8.14 SoLoader::Exceptions::InvalidEntryPoint Class Reference	45
8.14.1 Detailed Description	45
8.14.2 Constructor & Destructor Documentation	45
8.14.2.1 InvalidEntryPoint() [1/2]	46
8.14.2.2 InvalidEntryPoint() [2/2]	46
8.14.3 Member Function Documentation	46
8.14.3.1 what()	46
8.15 Score::Exceptions::InvalidFile Class Reference	46
8.15.1 Detailed Description	47
8.15.2 Constructor & Destructor Documentation	47
8.15.2.1 InvalidFile()	47
8.15.3 Member Function Documentation	47
8.15.3.1 what()	47
8.16 Core::Exceptions::InvalidScorePath Class Reference	47
8.16.1 Detailed Description	48
8.16.2 Constructor & Destructor Documentation	48
8.16.2.1 InvalidScorePath() [1/2]	48
8.16.2.2 InvalidScorePath() [2/2]	48
8.16.3 Member Function Documentation	48
8.16.3.1 what()	48
8.17 SoLoader::Exceptions::InvalidSO Class Reference	49
8.17.1 Detailed Description	49
8.17.2 Constructor & Destructor Documentation	49
8.17.2.1 InvalidSO() [1/2]	49
8.17.2.2 InvalidSO() [2/2]	49
8.17.3 Member Function Documentation	49

8.17.3.1 what()	50
8.18 KeyboardEvent_s Struct Reference	50
8.18.1 Detailed Description	50
8.18.2 Member Data Documentation	50
8.18.2.1 DOWN	51
8.18.2.2 ENTER	51
8.18.2.3 ESC	51
8.18.2.4 LEFT	51
8.18.2.5 NEXT_GAME	51
8.18.2.6 NEXT_GRAPHIC	51
8.18.2.7 PREV_GAME	51
8.18.2.8 PREV_GRAPHIC	51
8.18.2.9 RESTART	52
8.18.2.10 RIGHT	52
8.18.2.11 SPACE	52
8.18.2.12 UNKNOWN	52
8.18.2.13 UP	52
8.19 Graphic::Exceptions::LoadFontFailed Class Reference	52
8.19.1 Constructor & Destructor Documentation	53
8.19.1.1 LoadFontFailed() [1/2]	53
8.19.1.2 LoadFontFailed() [2/2]	53
8.19.2 Member Function Documentation	53
8.19.2.1 what()	53
8.20 Core::Exceptions::MissingMandatoryFolder Class Reference	53
8.20.1 Detailed Description	54
8.20.2 Constructor & Destructor Documentation	54
8.20.2.1 MissingMandatoryFolder() [1/2]	54
8.20.2.2 MissingMandatoryFolder() [2/2]	54
8.20.3 Member Function Documentation	54
8.20.3.1 what()	54
8.21 Graphic::Ncurses Class Reference	55
8.21.1 Constructor & Destructor Documentation	55
8.21.1.1 Ncurses()	55
8.21.1.2 ~Ncurses()	56
8.21.2 Member Function Documentation	56
8.21.2.1 clearScreen()	56
8.21.2.2 drawCircle()	56
8.21.2.3 drawRect()	56
8.21.2.4 drawScreen()	56
8.21.2.5 drawSprite()	57
8.21.2.6 drawText()	57
8.21.2.7 handleEvent()	57



8.21.2.8 isOperational()	57
8.21.3 Member Data Documentation	57
8.21.3.1 TRANSLATE_COLORS	58
8.21.3.2 WINDOW_HEIGHT	58
8.21.3.3 WINDOW_WIDTH	58
8.22 Game::Nibbler Class Reference	58
8.22.1 Constructor & Destructor Documentation	59
8.22.1.1 Nibbler()	59
8.22.1.2 ~Nibbler()	59
8.22.2 Member Function Documentation	59
8.22.2.1 getGameData()	59
8.22.2.2 handleEvent()	59
8.22.2.3 handleRender()	60
8.22.2.4 handleUpdate()	60
8.22.2.5 setGameData()	60
8.23 Game::Pacman Class Reference	61
8.23.1 Constructor & Destructor Documentation	61
8.23.1.1 Pacman()	61
8.23.1.2 ~Pacman()	61
8.23.2 Member Function Documentation	61
8.23.2.1 getGameData()	62
8.23.2.2 handleEvent()	62
8.23.2.3 handleRender()	62
8.23.2.4 handleUpdate()	62
8.23.2.5 setGameData()	62
8.24 Score::File::PlayerData Class Reference	63
8.24.1 Detailed Description	63
8.24.2 Constructor & Destructor Documentation	63
8.24.2.1 PlayerData()	63
8.24.3 Member Function Documentation	63
8.24.3.1 toStr()	63
8.24.4 Member Data Documentation	64
8.24.4.1 name	64
8.24.4.2 score	64
8.25 Rect Class Reference	64
8.25.1 Constructor & Destructor Documentation	64
8.25.1.1 Rect()	65
8.25.1.2 ~Rect()	65
8.26 Graphic::SDL2 Class Reference	65
8.26.1 Constructor & Destructor Documentation	66
8.26.1.1 SDL2()	66
8.26.1.2 ~SDL2()	66

8.26.2 Member Function Documentation	66
8.26.2.1 clearScreen()	66
8.26.2.2 drawCircle()	66
8.26.2.3 drawRect()	66
8.26.2.4 drawScreen()	67
8.26.2.5 drawSprite()	67
8.26.2.6 drawText()	67
8.26.2.7 handleEvent()	67
8.26.2.8 isOperational()	67
8.26.3 Member Data Documentation	68
8.26.3.1 WINDOW_HEIGHT	68
8.26.3.2 WINDOW_WIDTH	68
8.27 Graphic::SFML Class Reference	68
8.27.1 Constructor & Destructor Documentation	69
8.27.1.1 SFML()	69
8.27.1.2 ~SFML()	69
8.27.2 Member Function Documentation	69
8.27.2.1 clearScreen()	69
8.27.2.2 drawCircle()	69
8.27.2.3 drawRect()	69
8.27.2.4 drawScreen()	70
8.27.2.5 drawSprite()	70
8.27.2.6 drawText()	70
8.27.2.7 handleEvent()	70
8.27.2.8 isOperational()	70
8.27.3 Member Data Documentation	71
8.27.3.1 WINDOW_HEIGHT	71
8.27.3.2 WINDOW_WIDTH	71
8.28 SoLoader::SoLoader< T > Class Template Reference	71
8.28.1 Detailed Description	71
8.28.2 Constructor & Destructor Documentation	72
8.28.2.1 SoLoader()	72
8.28.2.2 ~SoLoader()	72
8.28.3 Member Function Documentation	72
8.28.3.1 changeSo()	72
8.28.3.2 getInstance()	73
8.28.3.3 getLibPath()	73
8.28.3.4 operator->()	73
8.28.4 Member Data Documentation	73
8.28.4.1 ENTRY_POINT_NAME	74
8.29 Sprite Class Reference	74
8.29.1 Constructor & Destructor Documentation	74

8.29.1.1 Sprite()	74
8.29.1.2 ~Sprite()	74
8.29.2 Member Function Documentation	75
8.29.2.1 getTextureName()	75
8.30 Game::TestGame Class Reference	75
8.30.1 Constructor & Destructor Documentation	75
8.30.1.1 TestGame()	75
8.30.1.2 ~TestGame()	76
8.30.2 Member Function Documentation	76
8.30.2.1 getGameData()	76
8.30.2.2 handleEvent()	76
8.30.2.3 handleRender()	76
8.30.2.4 handleUpdate()	76
8.30.2.5 setGameData()	77
8.31 Text Class Reference	77
8.31.1 Constructor & Destructor Documentation	77
8.31.1.1 Text()	77
8.31.1.2 ~Text()	78
8.31.2 Member Function Documentation	78
8.31.2.1 getText()	78
8.31.2.2 setText()	78
8.32 Core::Exceptions::UnableCreateFolder Class Reference	78
8.32.1 Detailed Description	78
8.32.2 Constructor & Destructor Documentation	79
8.32.2.1 UnableCreateFolder()	79
8.32.3 Member Function Documentation	79
8.32.3.1 what()	79
8.33 Core::Exceptions::UnknownGraphicalLib Class Reference	79
8.33.1 Detailed Description	80
8.33.2 Constructor & Destructor Documentation	80
8.33.2.1 UnknownGraphicalLib() [1/2]	80
8.33.2.2 UnknownGraphicalLib() [2/2]	80
8.33.3 Member Function Documentation	80
8.33.3.1 what()	80
8.34 Vector2< T > Struct Template Reference	80
8.34.1 Constructor & Destructor Documentation	81
8.34.1.1 Vector2()	81
8.34.2 Member Function Documentation	81
8.34.2.1 operator+()	81
8.34.2.2 operator+=()	81
8.34.2.3 operator-()	82
8.34.2.4 operator-=()	82

8.34.2.5 operator==()	82
8.34.3 Member Data Documentation	82
8.34.3.1 x	82
8.34.3.2 y	82
8.35 Vector3< T > Struct Template Reference	82
8.35.1 Constructor & Destructor Documentation	83
8.35.1.1 Vector3()	83
8.35.2 Member Function Documentation	83
8.35.2.1 operator+()	83
8.35.2.2 operator+=()	83
8.35.2.3 operator-()	84
8.35.2.4 operator-=()	84
8.35.2.5 operator==()	84
8.35.3 Member Data Documentation	84
8.35.3.1 x	84
8.35.3.2 y	84
8.35.3.3 z	84
<b>9 File Documentation</b>	<b>85</b>
9.1 CMakeLists.txt File Reference	85
9.2 README.md File Reference	85
9.3 src/core/Core.cpp File Reference	85
9.4 src/core/Core.hpp File Reference	85
9.4.1 Macro Definition Documentation	86
9.4.1.1 IS_IN_GAME	86
9.5 src/core/Exceptions.cpp File Reference	86
9.6 src/core/score/Exceptions.cpp File Reference	86
9.7 src/soLoader/Exceptions.cpp File Reference	86
9.8 src/core/KeyEvents.cpp File Reference	86
9.9 src/core/menu/Menu.cpp File Reference	87
9.10 src/core/menu/Strip.cpp File Reference	87
9.10.1 Variable Documentation	87
9.10.1.1 DEFAULT_STRIP_TEXT_SIZE	87
9.11 src/core/menu/Utils.cpp File Reference	87
9.12 src/core/score/File.cpp File Reference	87
9.13 src/core/score/Game.cpp File Reference	87
9.14 src/core/score/Score.hpp File Reference	88
9.15 src/game/IGame.hpp File Reference	88
9.16 src/graphic/Drawables.hpp File Reference	88
9.16.1 Typedef Documentation	89
9.16.1.1 Vector2f	89
9.16.1.2 Vector2i	89

9.16.1.3 Vector2u . . . . .	89
9.16.1.4 Vector3f . . . . .	89
9.16.1.5 Vector3i . . . . .	89
9.16.1.6 Vector3u . . . . .	89
9.17 src/graphic/IGraphic.hpp File Reference . . . . .	90
9.17.1 Macro Definition Documentation . . . . .	90
9.17.1.1 FONT_FILENAME . . . . .	90
9.17.1.2 PERCENTAGE . . . . .	90
9.18 src/lib/game/nibbler/Draw.cpp File Reference . . . . .	90
9.18.1 Variable Documentation . . . . .	90
9.18.1.1 DEFAULT_TEXT_SIZE . . . . .	91
9.19 src/lib/game/pacman/Draw.cpp File Reference . . . . .	91
9.19.1 Variable Documentation . . . . .	91
9.19.1.1 DEFAULT_TEXT_SIZE . . . . .	91
9.20 src/lib/game/nibbler/Nibbler.cpp File Reference . . . . .	91
9.20.1 Function Documentation . . . . .	91
9.20.1.1 __attribute__((__unused__)) [1/2] . . . . .	92
9.20.1.2 __attribute__((__unused__)) [2/2] . . . . .	92
9.20.1.3 entry() . . . . .	92
9.20.2 Variable Documentation . . . . .	92
9.20.2.1 instance . . . . .	92
9.21 src/lib/game/nibbler/Nibbler.hpp File Reference . . . . .	92
9.21.1 Macro Definition Documentation . . . . .	93
9.21.1.1 IS_GAME_IN_PROGRESS . . . . .	93
9.22 src/lib/game/pacman/Map.cpp File Reference . . . . .	93
9.23 src/lib/game/pacman/MoveEnemy.cpp File Reference . . . . .	93
9.24 src/lib/game/pacman/MoveEntity.cpp File Reference . . . . .	93
9.25 src/lib/game/pacman/MovePlayer.cpp File Reference . . . . .	93
9.26 src/lib/game/pacman/Pacman.cpp File Reference . . . . .	93
9.26.1 Function Documentation . . . . .	94
9.26.1.1 __attribute__((__unused__)) [1/2] . . . . .	94
9.26.1.2 __attribute__((__unused__)) [2/2] . . . . .	94
9.26.1.3 entry() . . . . .	94
9.26.2 Variable Documentation . . . . .	94
9.26.2.1 instance . . . . .	94
9.27 src/lib/game/pacman/Pacman.hpp File Reference . . . . .	95
9.27.1 Macro Definition Documentation . . . . .	95
9.27.1.1 IS_GAME_IN_PROGRESS . . . . .	95
9.28 src/lib/game/pacman/Pathfinding.cpp File Reference . . . . .	95
9.29 src/lib/game/test/TestGame.cpp File Reference . . . . .	95
9.29.1 Function Documentation . . . . .	96
9.29.1.1 __attribute__((__unused__)) [1/2] . . . . .	96

9.29.1.2 <code>__attribute__()</code> [2/2]	96
9.29.1.3 <code>entry()</code>	96
9.29.2 Variable Documentation	96
9.29.2.1 instance	96
9.30 <code>src/lib/game/test/TestGame.hpp</code> File Reference	96
9.31 <code>src/lib/graphic/Exceptions.hpp</code> File Reference	97
9.32 <code>src/soLoader/Exceptions.hpp</code> File Reference	97
9.33 <code>src/lib/graphic/ncurses/Ncurses.cpp</code> File Reference	97
9.33.1 Function Documentation	98
9.33.1.1 <code>__attribute__()</code> [1/2]	98
9.33.1.2 <code>__attribute__()</code> [2/2]	98
9.33.1.3 <code>entry()</code>	98
9.33.2 Variable Documentation	98
9.33.2.1 instance	98
9.34 <code>src/lib/graphic/ncurses/Ncurses.hpp</code> File Reference	98
9.34.1 Macro Definition Documentation	99
9.34.1.1 <code>CUSTOM_KEY_ENTER</code>	99
9.34.1.2 <code>KEY_ESCAPE</code>	99
9.35 <code>src/lib/graphic/sdl2/SDL2.cpp</code> File Reference	99
9.35.1 Function Documentation	100
9.35.1.1 <code>__attribute__()</code> [1/2]	100
9.35.1.2 <code>__attribute__()</code> [2/2]	100
9.35.1.3 <code>entry()</code>	100
9.35.2 Variable Documentation	100
9.35.2.1 instance	100
9.36 <code>src/lib/graphic/sdl2/SDL2.hpp</code> File Reference	100
9.37 <code>src/lib/graphic/sfml/SFML.cpp</code> File Reference	101
9.37.1 Function Documentation	101
9.37.1.1 <code>__attribute__()</code> [1/2]	101
9.37.1.2 <code>__attribute__()</code> [2/2]	101
9.37.1.3 <code>entry()</code>	101
9.37.2 Variable Documentation	102
9.37.2.1 instance	102
9.38 <code>src/lib/graphic/sfml/SFML.hpp</code> File Reference	102
9.39 <code>src/Main.cpp</code> File Reference	102
9.39.1 Function Documentation	102
9.39.1.1 <code>main()</code>	102
9.40 <code>src/soLoader/SoLoader.hpp</code> File Reference	103

# 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 platform 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 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 respect the following pattern: `lib_arcade_$gamenamename.so` or `lib_arcade_$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 platform 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 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 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.sfm1-dev.org/download/sfm1/2.5.1/>

#### 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, navigate 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:

Core	13
Core::Exceptions	13
Game	13
Graphic	14
Graphic::Exceptions	14
Score	14
Score::Exceptions	15
SoLoader	15
SoLoader::Exceptions	15



## Chapter 4

# Hierarchical Index

### 4.1 Class Hierarchy

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

AColor . . . . .	17
Circle . . . . .	24
Rect . . . . .	64
Text . . . . .	77
APosition . . . . .	20
Circle . . . . .	24
Rect . . . . .	64
Sprite . . . . .	74
Text . . . . .	77
ASize . . . . .	22
Circle . . . . .	24
Rect . . . . .	64
Sprite . . . . .	74
Text . . . . .	77
Color . . . . .	25
Core::Core . . . . .	28
exception	
Core::Exceptions::EmptyMandatoryFolder . . . . .	31
Core::Exceptions::ExitGame . . . . .	32
Core::Exceptions::InvalidScorePath . . . . .	47
Core::Exceptions::MissingMandatoryFolder . . . . .	53
Core::Exceptions::UnableCreateFolder . . . . .	78
Core::Exceptions::UnknownGraphicalLib . . . . .	79
Graphic::Exceptions::LoadFontFailed . . . . .	52
Score::Exceptions::InvalidFile . . . . .	46
SoLoader::Exceptions::InvalidEntryPoint . . . . .	45
SoLoader::Exceptions::InvalidSO . . . . .	49
Score::File . . . . .	33
Score::Game . . . . .	35
IGame . . . . .	39
Game::Nibbler . . . . .	58
Game::Pacman . . . . .	61
Game::TestGame . . . . .	75
IGraphicRenderer . . . . .	43

IGraphic . . . . .	41
Graphic::Ncurses . . . . .	55
Graphic::SDL2 . . . . .	65
Graphic::SFML . . . . .	68
KeyboardEvent_s . . . . .	50
Score::File::PlayerData . . . . .	63
SoLoader::SoLoader< T > . . . . .	71
SoLoader::SoLoader< IGame > . . . . .	71
SoLoader::SoLoader< IGraphic > . . . . .	71
Vector2< T > . . . . .	80
Vector2< float > . . . . .	80
Vector2< int > . . . . .	80
Vector3< T > . . . . .	82

## Chapter 5

# Class Index

### 5.1 Class List

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

AColor	17
APosition	20
ASize	22
Circle	24
Color	25
Core::Core	28
Core::Exceptions::EmptyMandatoryFolder	31
Core::Exceptions::ExitGame	32
Score::File	33
Score::Game	35
IGame	39
IGraphic	41
IGraphicRenderer	43
SoLoader::Exceptions::InvalidEntryPoint	45
Score::Exceptions::InvalidFile	46
Core::Exceptions::InvalidScorePath	47
SoLoader::Exceptions::InvalidSO	49
KeyboardEvent_s	50
Graphic::Exceptions::LoadFontFailed	52
Core::Exceptions::MissingMandatoryFolder	53
Graphic::Ncurses	55
Game::Nibbler	58
Game::Pacman	61
Score::File::PlayerData	63
Rect	64
Graphic::SDL2	65
Graphic::SFML	68
SoLoader::SoLoader< T >	71
Sprite	74
Game::TestGame	75
Text	77
Core::Exceptions::UnableCreateFolder	78
Core::Exceptions::UnknownGraphicalLib	79
Vector2< T >	80
Vector3< T >	82





## Chapter 6

# File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

src/Main.cpp	102
src/core/Core.cpp	85
src/core/Core.hpp	85
src/core/Exceptions.cpp	86
src/core/KeyEvents.cpp	86
src/core/menu/Menu.cpp	87
src/core/menu/Strip.cpp	87
src/core/menu/Utils.cpp	87
src/core/score/Exceptions.cpp	86
src/core/score/File.cpp	87
src/core/score/Game.cpp	87
src/core/score/Score.hpp	88
src/game/IGame.hpp	88
src/graphic/Drawables.hpp	88
src/graphic/IGraphic.hpp	90
src/lib/game/nibbler/Draw.cpp	90
src/lib/game/nibbler/Nibbler.cpp	91
src/lib/game/nibbler/Nibbler.hpp	92
src/lib/game/pacman/Draw.cpp	91
src/lib/game/pacman/Map.cpp	93
src/lib/game/pacman/MoveEnemy.cpp	93
src/lib/game/pacman/MoveEntity.cpp	93
src/lib/game/pacman/MovePlayer.cpp	93
src/lib/game/pacman/Pacman.cpp	93
src/lib/game/pacman/Pacman.hpp	95
src/lib/game/pacman/Pathfinding.cpp	95
src/lib/game/test/TestGame.cpp	95
src/lib/game/test/TestGame.hpp	96
src/lib/graphic/Exceptions.hpp	97
src/lib/graphic/ncurses/Ncurses.cpp	97
src/lib/graphic/ncurses/Ncurses.hpp	98
src/lib/graphic/sdl2/SDL2.cpp	99
src/lib/graphic/sdl2/SDL2.hpp	100
src/lib/graphic/sfml/SFML.cpp	101
src/lib/graphic/sfml/SFML.hpp	102
src/soLoader/Exceptions.cpp	86
src/soLoader/Exceptions.hpp	97
src/soLoader/SoLoader.hpp	103



## 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



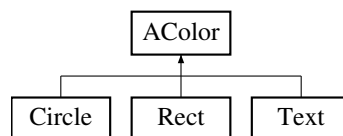
## Chapter 8

# 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]
```

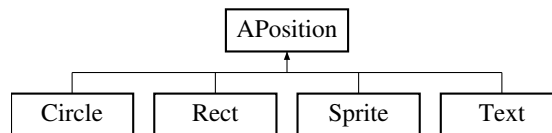
The documentation for this class was generated from the following file:

- `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]
```

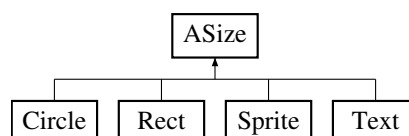
The documentation for this class was generated from the following file:

- 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]
```

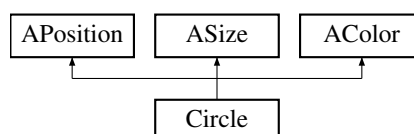
The documentation for this class was generated from the following file:

- 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

<i>graphicalLib</i>	: The path to the graphical library to load.
---------------------	--

#### Exceptions

<a href="#">SoLoader::Exceptions::InvalidSO</a>	
---	--

## Exceptions

<a href="#"><i>SoLoader::Exceptions::InvalidEntryPoint</i></a>	
<a href="#"><i>Core::Exceptions::EmptyMandatoryFolder</i></a>	
<a href="#"><i>Core::Exceptions::UnknownGraphicalLib</i></a>	
<a href="#"><i>Core::Exceptions::UnableCreateFolder</i></a>	
<a href="#"><i>Core::Exceptions::MissingMandatoryFolder</i></a>	

## 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

<i>gameName</i>	: The game's name score to load.
-----------------	----------------------------------

## Exceptions

<a href="#"><i>Core::Exceptions::InvalidScorePath</i></a>	
---	--

## 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

<i>filename</i>	: The path to the game library.
-----------------	---------------------------------

#### Exceptions

<a href="#"><i>SoLoader::Exceptions::InvalidSO</i></a>	
<a href="#"><i>SoLoader::Exceptions::InvalidEntryPoint</i></a>	

### 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

<i>filename</i>	: The path to the graphical library.
-----------------	--------------------------------------

#### Exceptions

<a href="#"><i>SoLoader::Exceptions::InvalidSO</i></a>	
<a href="#"><i>SoLoader::Exceptions::InvalidEntryPoint</i></a>	

## 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.

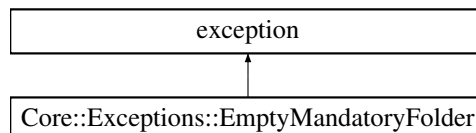
The documentation for this class was generated from the following files:

- [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.

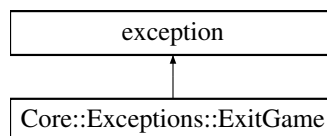
The documentation for this class was generated from the following files:

- [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

<i>content</i>	: 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

<i>user</i>	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

<i>letter</i>	: 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

<i>idx</i>	: The index of the letter
------------	---------------------------

## Exceptions

<i>std::out_of_range</i>	: 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

<code>std::out_of_range</code>	: 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

<code>Score::Exceptions::InvalidFile</code>	: 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

<code>game</code>	: 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

<i>idx</i>	: The index of the letter
<i>c</i>	: The letter which will be replaced

## Exceptions

<code>std::out_of_range</code>	: 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

<i>score</i>	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

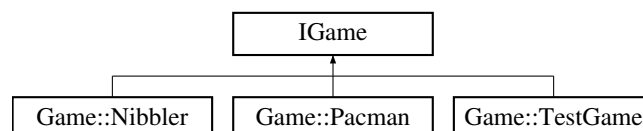
The documentation for this class was generated from the following files:

- [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 > [GameDataTypes](#)

## 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 [GameDataTypes](#) [getGameData](#) ()=0
- virtual void [setGameData](#) ([GameDataTypes](#) &data)=0

### 8.11.1 Member Typedef Documentation

#### 8.11.1.1 GameDataTypes

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

### 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 GameDataTypes 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](#).

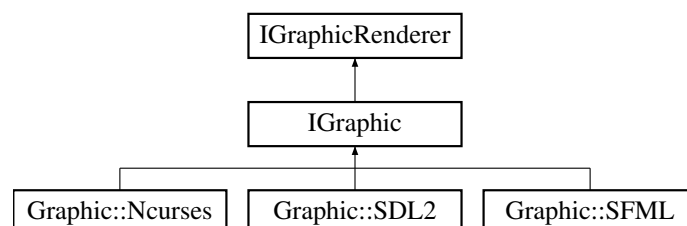
The documentation for this class was generated from the following file:

- [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](#).

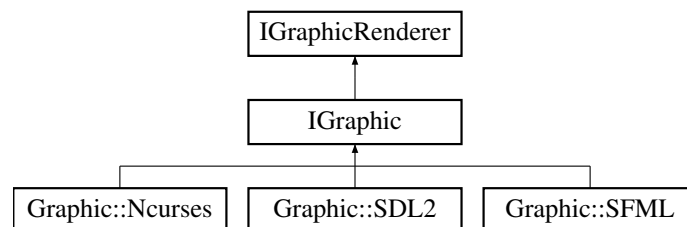
The documentation for this class was generated from the following file:

- [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]
```

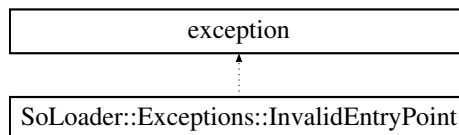
The documentation for this class was generated from the following file:

- [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.

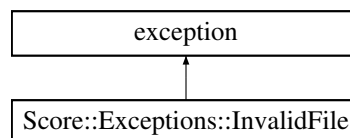
The documentation for this class was generated from the following files:

- [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

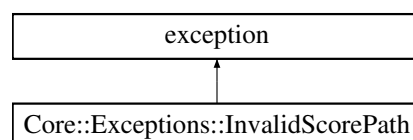
The documentation for this class was generated from the following files:

- [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.

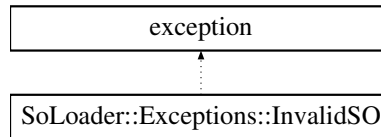
The documentation for this class was generated from the following files:

- 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]
```

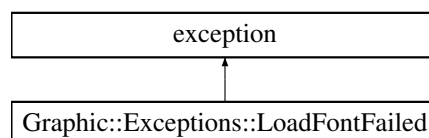
The documentation for this struct was generated from the following file:

- [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]
```

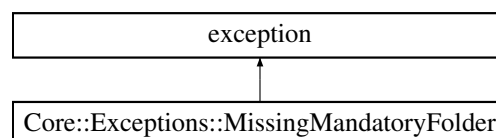
The documentation for this class was generated from the following file:

- 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.

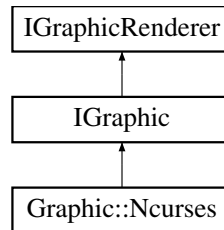
The documentation for this class was generated from the following files:

- 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]
```

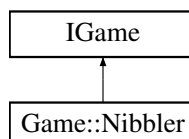
The documentation for this class was generated from the following files:

- [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](#) (GameDataTypes &data) override
- [GameDataTypes](#) [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::GameDataTypes 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

<i>name</i>	: 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

<i>renderer</i>	: 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

<i>elapsedTime</i>	: The time length between two calls.
--------------------	--------------------------------------

Implements [IGame](#).

### 8.22.2.5 setGameData()

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

Update game data.

#### Parameters

<i>data</i>	: The game data to send.
-------------	--------------------------

Implements [IGame](#).

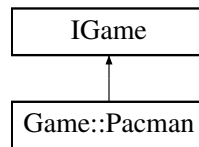
The documentation for this class was generated from the following files:

- [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](#) (GameDataTypes &data) override
- [GameDataTypes](#) [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::GameDataTypes 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::GameDataTypes & 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

<i>name</i>	: The author's name
<i>score</i>	: 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

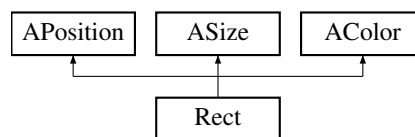
The documentation for this class was generated from the following files:

- [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]
```

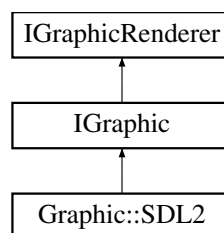
The documentation for this class was generated from the following file:

- [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]
```

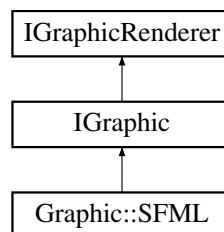
The documentation for this class was generated from the following files:

- [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

<i>T</i>	Type which will be associated to the library.
----------	---

## 8.28.2 Constructor &amp; 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

<i>DLLPath</i>	: The new library's path.
----------------	---------------------------

## Exceptions

<i>SoLoader::Exceptions::InvalidSO</i>	
<i>SoLoader::Exceptions::InvalidEntryPoint</i>	

### 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]
```

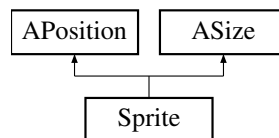
The documentation for this class was generated from the following file:

- 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]
```

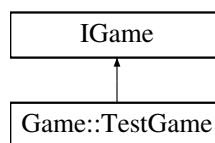
The documentation for this class was generated from the following file:

- 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](#) ([GameData Type](#) &data) override
- [GameData Type](#) [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::GameDataTypes 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](#).

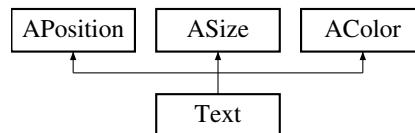
The documentation for this class was generated from the following files:

- [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]
```

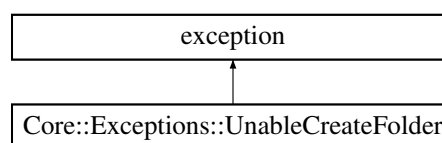
The documentation for this class was generated from the following file:

- [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.

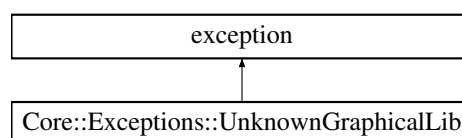
The documentation for this class was generated from the following files:

- [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 exist 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

- `__attribute__`
    - Ncurses.cpp, 98
    - Nibbler.cpp, 91, 92
    - Pacman.cpp, 94
    - SDL2.cpp, 100
    - SFML.cpp, 101
    - TestGame.cpp, 96
  - `_color`
    - AColor, 19
  - `_pos`
    - APosition, 22
  - `_size`
    - ASize, 24
  - `~AColor`
    - AColor, 18
  - `~APosition`
    - APosition, 20
  - `~ASize`
    - ASize, 23
  - `~Circle`
    - Circle, 25
  - `~Core`
    - Core::Core, 29
  - `~IGame`
    - IGame, 40
  - `~IGraphic`
    - IGraphic, 42
  - `~IGraphicRenderer`
    - IGraphicRenderer, 43
  - `~Ncurses`
    - Graphic::Ncurses, 55
  - `~Nibbler`
    - Game::Nibbler, 59
  - `~Pacman`
    - Game::Pacman, 61
  - `~Rect`
    - Rect, 65
  - `~SDL2`
    - Graphic::SDL2, 66
  - `~SFML`
    - Graphic::SFML, 69
  - `~SoLoader`
    - SoLoader::SoLoader< T >, 72
  - `~Sprite`
    - Sprite, 74
  - `~TestGame`
    - Game::TestGame, 75
  - `~Text`
    - Text, 77

- AColor, 17
  - `_color`, 19
  - `~AColor`, 18
  - AColor, 17
  - `getColor`, 18
  - `getColorAlpha`, 18
  - `getColorBlue`, 18
  - `getColorGreen`, 18
  - `getColorRed`, 18
  - `setColor`, 19
  - `setColorAlpha`, 19
  - `setColorBlue`, 19
  - `setColorGreen`, 19
  - `setColorRed`, 19
- addEntry
  - Score::File, 35
- addLetter
  - Score::Game, 36
- alpha
  - Color, 27
- APosition, 20
  - `_pos`, 22
  - `~APosition`, 20
  - APosition, 20
  - `getPosition`, 21
  - `getPositionX`, 21
  - `getPositionY`, 21
  - `move`, 21
  - `moveX`, 21
  - `moveY`, 21
  - `setPosition`, 22
  - `setPositionX`, 22
  - `setPositionY`, 22
- ASize, 22
  - `_size`, 24
  - `~ASize`, 23
  - ASize, 23
  - `getSize`, 23
  - `getSizeX`, 23
  - `getSizeY`, 23
  - `setSize`, 24
  - `setSizeX`, 24
  - `setSizeY`, 24
- Black
  - Color, 26
- Blue
  - Color, 26
- blue
  - Color, 27

- changeSo
  - SoLoader::SoLoader< T >, 72
- Circle, 24
  - ~Circle, 25
  - Circle, 25
- clearScreen
  - Graphic::Ncurses, 56
  - Graphic::SDL2, 66
  - Graphic::SFML, 69
  - IGraphicRenderer, 44
- CMakeLists.txt, 85
- Color, 25
  - alpha, 27
  - Black, 26
  - Blue, 26
  - blue, 27
  - Color, 26
  - Green, 26
  - green, 27
  - operator==, 26
  - Red, 26
  - red, 27
  - Transparent, 27
  - White, 27
- Core, 13
  - Core::Core, 28
- Core.hpp
  - IS\_IN\_GAME, 86
- Core::Core, 28
  - ~Core, 29
  - Core, 28
  - FRAME\_PER\_TIME, 30
  - loadScore, 29
  - run, 29
  - SCORE\_PATH, 31
  - useGame, 29
  - useGraphic, 30
- Core::Exceptions, 13
- Core::Exceptions::EmptyMandatoryFolder, 31
  - EmptyMandatoryFolder, 31, 32
  - what, 32
- Core::Exceptions::ExitGame, 32
  - ExitGame, 33
  - what, 33
- Core::Exceptions::InvalidScorePath, 47
  - InvalidScorePath, 48
  - what, 48
- Core::Exceptions::MissingMandatoryFolder, 53
  - MissingMandatoryFolder, 54
  - what, 54
- Core::Exceptions::UnableCreateFolder, 78
  - UnableCreateFolder, 79
  - what, 79
- Core::Exceptions::UnknownGraphicalLib, 79
  - UnknownGraphicalLib, 80
  - what, 80
- CUSTOM\_KEY\_ENTER
  - Ncurses.hpp, 99
- DEFAULT\_STRIP\_TEXT\_SIZE
  - Strip.cpp, 87
- DEFAULT\_TEXT\_SIZE
  - Draw.cpp, 90, 91
- DOWN
  - KeyboardEvent\_s, 50
- Draw.cpp
  - DEFAULT\_TEXT\_SIZE, 90, 91
- Drawables.hpp
  - Vector2f, 89
  - Vector2i, 89
  - Vector2u, 89
  - Vector3f, 89
  - Vector3i, 89
  - Vector3u, 89
- drawCircle
  - Graphic::Ncurses, 56
  - Graphic::SDL2, 66
  - Graphic::SFML, 69
  - IGraphicRenderer, 44
- drawRect
  - Graphic::Ncurses, 56
  - Graphic::SDL2, 66
  - Graphic::SFML, 69
  - IGraphicRenderer, 44
- drawScreen
  - Graphic::Ncurses, 56
  - Graphic::SDL2, 66
  - Graphic::SFML, 69
  - IGraphicRenderer, 44
- drawSprite
  - Graphic::Ncurses, 56
  - Graphic::SDL2, 67
  - Graphic::SFML, 70
  - IGraphicRenderer, 44
- drawText
  - Graphic::Ncurses, 57
  - Graphic::SDL2, 67
  - Graphic::SFML, 70
  - IGraphicRenderer, 44
- EmptyMandatoryFolder
  - Core::Exceptions::EmptyMandatoryFolder, 31, 32
- ENTER
  - KeyboardEvent\_s, 51
- entry
  - Ncurses.cpp, 98
  - Nibbler.cpp, 92
  - Pacman.cpp, 94
  - SDL2.cpp, 100
  - SFML.cpp, 101
  - TestGame.cpp, 96
- ENTRY\_POINT\_NAME
  - SoLoader::SoLoader< T >, 73
- ESC
  - KeyboardEvent\_s, 51
- ExitGame
  - Core::Exceptions::ExitGame, 33



- File
  - Score::File, [34](#), [35](#)
- FONT\_FILENAME
  - IGraphic.hpp, [90](#)
- FRAME\_PER\_TIME
  - Core::Core, [30](#)
- Game, [13](#)
  - Score::Game, [36](#)
- Game::Nibbler, [58](#)
  - ~Nibbler, [59](#)
  - getGameData, [59](#)
  - handleEvent, [59](#)
  - handleRender, [59](#)
  - handleUpdate, [60](#)
  - Nibbler, [59](#)
  - setGameData, [60](#)
- Game::Pacman, [61](#)
  - ~Pacman, [61](#)
  - getGameData, [61](#)
  - handleEvent, [62](#)
  - handleRender, [62](#)
  - handleUpdate, [62](#)
  - Pacman, [61](#)
  - setGameData, [62](#)
- Game::TestGame, [75](#)
  - ~TestGame, [75](#)
  - getGameData, [76](#)
  - handleEvent, [76](#)
  - handleRender, [76](#)
  - handleUpdate, [76](#)
  - setGameData, [76](#)
  - TestGame, [75](#)
- GameDataType
  - IGame, [40](#)
- getAuthor
  - Score::Game, [37](#)
- getColor
  - AColor, [18](#)
- getColorAlpha
  - AColor, [18](#)
- getColorBlue
  - AColor, [18](#)
- getColorGreen
  - AColor, [18](#)
- getColorRed
  - AColor, [18](#)
- getGameData
  - Game::Nibbler, [59](#)
  - Game::Pacman, [61](#)
  - Game::TestGame, [76](#)
  - IGame, [40](#)
- getInstance
  - SoLoader::SoLoader< T >, [73](#)
- getLetter
  - Score::Game, [37](#)
- getLibPath
  - SoLoader::SoLoader< T >, [73](#)
- getListUser
  - Score::File, [35](#)
- getPosition
  - APosition, [21](#)
- getPositionX
  - APosition, [21](#)
- getPositionY
  - APosition, [21](#)
- getSize
  - ASize, [23](#)
- getSizeX
  - ASize, [23](#)
- getSizeY
  - ASize, [23](#)
- getText
  - Text, [78](#)
- getTextureName
  - Sprite, [75](#)
- Graphic, [14](#)
  - Graphic::Exceptions, [14](#)
  - Graphic::Exceptions::LoadFontFailed, [52](#)
    - LoadFontFailed, [53](#)
    - what, [53](#)
  - Graphic::Ncurses, [55](#)
    - ~Ncurses, [55](#)
    - clearScreen, [56](#)
    - drawCircle, [56](#)
    - drawRect, [56](#)
    - drawScreen, [56](#)
    - drawSprite, [56](#)
    - drawText, [57](#)
    - handleEvent, [57](#)
    - isOperational, [57](#)
    - Ncurses, [55](#)
    - TRANSLATE\_COLORS, [57](#)
    - WINDOW\_HEIGHT, [58](#)
    - WINDOW\_WIDTH, [58](#)
  - Graphic::SDL2, [65](#)
    - ~SDL2, [66](#)
    - clearScreen, [66](#)
    - drawCircle, [66](#)
    - drawRect, [66](#)
    - drawScreen, [66](#)
    - drawSprite, [67](#)
    - drawText, [67](#)
    - handleEvent, [67](#)
    - isOperational, [67](#)
    - SDL2, [66](#)
    - WINDOW\_HEIGHT, [68](#)
    - WINDOW\_WIDTH, [68](#)
  - Graphic::SFML, [68](#)
    - ~SFML, [69](#)
    - clearScreen, [69](#)
    - drawCircle, [69](#)
    - drawRect, [69](#)
    - drawScreen, [69](#)
    - drawSprite, [70](#)
    - drawText, [70](#)
    - handleEvent, [70](#)

- isOperational, 70
- SFML, 69
- WINDOW\_HEIGHT, 71
- WINDOW\_WIDTH, 71
- Green
  - Color, 26
- green
  - Color, 27
- handleEvent
  - Game::Nibbler, 59
  - Game::Pacman, 62
  - Game::TestGame, 76
  - Graphic::Ncurses, 57
  - Graphic::SDL2, 67
  - Graphic::SFML, 70
  - IGame, 40
  - IGraphic, 42
- handleRender
  - Game::Nibbler, 59
  - Game::Pacman, 62
  - Game::TestGame, 76
  - IGame, 41
- handleUpdate
  - Game::Nibbler, 60
  - Game::Pacman, 62
  - Game::TestGame, 76
  - IGame, 41
- IGame, 39
  - ~IGame, 40
  - GameData Type, 40
  - getGameData, 40
  - handleEvent, 40
  - handleRender, 41
  - handleUpdate, 41
  - setGameData, 41
- IGraphic, 41
  - ~IGraphic, 42
  - handleEvent, 42
  - IGraphic, 42
  - isOperational, 42
- IGraphic.hpp
  - FONT\_FILENAME, 90
  - PERCENTAGE, 90
- IGraphicRenderer, 43
  - ~IGraphicRenderer, 43
  - clearScreen, 44
  - drawCircle, 44
  - drawRect, 44
  - drawScreen, 44
  - drawSprite, 44
  - drawText, 44
  - operator=, 45
- instance
  - Ncurses.cpp, 98
  - Nibbler.cpp, 92
  - Pacman.cpp, 94
  - SDL2.cpp, 100
  - SFML.cpp, 102
  - TestGame.cpp, 96
- InvalidEntryPoint
  - SoLoader::Exceptions::InvalidEntryPoint, 45, 46
- InvalidFile
  - Score::Exceptions::InvalidFile, 47
- InvalidScorePath
  - Core::Exceptions::InvalidScorePath, 48
- InvalidSO
  - SoLoader::Exceptions::InvalidSO, 49
- IS\_GAME\_IN\_PROGRESS
  - Nibbler.hpp, 93
  - Pacman.hpp, 95
- IS\_IN\_GAME
  - Core.hpp, 86
- isOperational
  - Graphic::Ncurses, 57
  - Graphic::SDL2, 67
  - Graphic::SFML, 70
  - IGraphic, 42
- KEY\_ESCAPE
  - Ncurses.hpp, 99
- KeyboardEvent\_s, 50
  - DOWN, 50
  - ENTER, 51
  - ESC, 51
  - LEFT, 51
  - NEXT\_GAME, 51
  - NEXT\_GRAPHIC, 51
  - PREV\_GAME, 51
  - PREV\_GRAPHIC, 51
  - RESTART, 51
  - RIGHT, 52
  - SPACE, 52
  - UNKNOWN, 52
  - UP, 52
- LEFT
  - KeyboardEvent\_s, 51
- LoadFontFailed
  - Graphic::Exceptions::LoadFontFailed, 53
- loadScore
  - Core::Core, 29
- main
  - Main.cpp, 102
- Main.cpp
  - main, 102
- MAX\_AUTHOR\_NAME
  - Score::Game, 39
- MissingMandatoryFolder
  - Core::Exceptions::MissingMandatoryFolder, 54
- move
  - APosition, 21
- moveX
  - APosition, 21
- moveY
  - APosition, 21

name  
     Score::File::PlayerData, 64  
 Ncurses  
     Graphic::Ncurses, 55  
 Ncurses.cpp  
     \_\_attribute\_\_, 98  
     entry, 98  
     instance, 98  
 Ncurses.hpp  
     CUSTOM\_KEY\_ENTER, 99  
     KEY\_ESCAPE, 99  
 NEXT\_GAME  
     KeyboardEvent\_s, 51  
 NEXT\_GRAPHIC  
     KeyboardEvent\_s, 51  
 Nibbler  
     Game::Nibbler, 59  
 Nibbler.cpp  
     \_\_attribute\_\_, 91, 92  
     entry, 92  
     instance, 92  
 Nibbler.hpp  
     IS\_GAME\_IN\_PROGRESS, 93  
  
 operator+  
     Vector2< T >, 81  
     Vector3< T >, 83  
 operator+=  
     Vector2< T >, 81  
     Vector3< T >, 83  
 operator-  
     Vector2< T >, 81  
     Vector3< T >, 83  
 operator->  
     SoLoader::SoLoader< T >, 73  
 operator-=  
     Vector2< T >, 82  
     Vector3< T >, 84  
 operator=  
     IGraphicRenderer, 45  
 operator==  
     Color, 26  
     Vector2< T >, 82  
     Vector3< T >, 84  
  
 Pacman  
     Game::Pacman, 61  
 Pacman.cpp  
     \_\_attribute\_\_, 94  
     entry, 94  
     instance, 94  
 Pacman.hpp  
     IS\_GAME\_IN\_PROGRESS, 95  
 PERCENTAGE  
     IGraphic.hpp, 90  
 PlayerData  
     Score::File::PlayerData, 63  
 PREV\_GAME  
     KeyboardEvent\_s, 51  
  
 PREV\_GRAPHIC  
     KeyboardEvent\_s, 51  
  
 README.md, 85  
 Rect, 64  
     ~Rect, 65  
     Rect, 64  
 Red  
     Color, 26  
 red  
     Color, 27  
 removeLastLetter  
     Score::Game, 37  
 reset  
     Score::Game, 38  
 RESTART  
     KeyboardEvent\_s, 51  
 RIGHT  
     KeyboardEvent\_s, 52  
 run  
     Core::Core, 29  
  
 save  
     Score::Game, 38  
 Score, 14  
 score  
     Score::File::PlayerData, 64  
 Score::Exceptions, 15  
 Score::Exceptions::InvalidFile, 46  
     InvalidFile, 47  
     what, 47  
 Score::File, 33  
     addEntry, 35  
     File, 34, 35  
     getListUser, 35  
 Score::File::PlayerData, 63  
     name, 64  
     PlayerData, 63  
     score, 64  
     toStr, 63  
 Score::Game, 35  
     addLetter, 36  
     Game, 36  
     getAuthor, 37  
     getLetter, 37  
     MAX\_AUTHOR\_NAME, 39  
     removeLastLetter, 37  
     reset, 38  
     save, 38  
     setGame, 38  
     setLetter, 38  
     setScore, 39  
 SCORE\_PATH  
     Core::Core, 31  
 SDL2  
     Graphic::SDL2, 66  
 SDL2.cpp  
     \_\_attribute\_\_, 100  
     entry, 100

- instance, 100
- setColor
  - AColor, 19
- setColorAlpha
  - AColor, 19
- setColorBlue
  - AColor, 19
- setColorGreen
  - AColor, 19
- setColorRed
  - AColor, 19
- setGame
  - Score::Game, 38
- setGameData
  - Game::Nibbler, 60
  - Game::Pacman, 62
  - Game::TestGame, 76
  - IGame, 41
- setLetter
  - Score::Game, 38
- setPosition
  - APosition, 22
- setPositionX
  - APosition, 22
- setPositionY
  - APosition, 22
- setScore
  - Score::Game, 39
- setSize
  - ASize, 24
- setSizeX
  - ASize, 24
- setSizeY
  - ASize, 24
- setText
  - Text, 78
- SFML
  - Graphic::SFML, 69
- SFML.cpp
  - \_\_attribute\_\_, 101
  - entry, 101
  - instance, 102
- SoLoader, 15
  - SoLoader::SoLoader< T >, 72
- SoLoader::Exceptions, 15
- SoLoader::Exceptions::InvalidEntryPoint, 45
  - InvalidEntryPoint, 45, 46
  - what, 46
- SoLoader::Exceptions::InvalidSO, 49
  - InvalidSO, 49
  - what, 49
- SoLoader::SoLoader< T >, 71
  - ~SoLoader, 72
  - changeSo, 72
  - ENTRY\_POINT\_NAME, 73
  - getInstance, 73
  - getLibPath, 73
  - operator->, 73
  - SoLoader, 72
  - SPACE
    - KeyboardEvent\_s, 52
  - Sprite, 74
    - ~Sprite, 74
    - getTextureName, 75
    - Sprite, 74
  - src/core/Core.cpp, 85
  - src/core/Core.hpp, 85
  - src/core/Exceptions.cpp, 86
  - src/core/KeyEvents.cpp, 86
  - src/core/menu/Menu.cpp, 87
  - src/core/menu/Strip.cpp, 87
  - src/core/menu/Utils.cpp, 87
  - src/core/score/Exceptions.cpp, 86
  - src/core/score/File.cpp, 87
  - src/core/score/Game.cpp, 87
  - src/core/score/Score.hpp, 88
  - src/game/IGame.hpp, 88
  - src/graphic/Drawables.hpp, 88
  - src/graphic/IGraphic.hpp, 90
  - src/lib/game/nibbler/Draw.cpp, 90
  - src/lib/game/nibbler/Nibbler.cpp, 91
  - src/lib/game/nibbler/Nibbler.hpp, 92
  - src/lib/game/pacman/Draw.cpp, 91
  - src/lib/game/pacman/Map.cpp, 93
  - src/lib/game/pacman/MoveEnemy.cpp, 93
  - src/lib/game/pacman/MoveEntity.cpp, 93
  - src/lib/game/pacman/MovePlayer.cpp, 93
  - src/lib/game/pacman/Pacman.cpp, 93
  - src/lib/game/pacman/Pacman.hpp, 95
  - src/lib/game/pacman/Pathfinding.cpp, 95
  - src/lib/game/test/TestGame.cpp, 95
  - src/lib/game/test/TestGame.hpp, 96
  - src/lib/graphic/Exceptions.hpp, 97
  - src/lib/graphic/ncurses/Ncurses.cpp, 97
  - src/lib/graphic/ncurses/Ncurses.hpp, 98
  - src/lib/graphic/sdl2/SDL2.cpp, 99
  - src/lib/graphic/sdl2/SDL2.hpp, 100
  - src/lib/graphic/sfml/SFML.cpp, 101
  - src/lib/graphic/sfml/SFML.hpp, 102
  - src/Main.cpp, 102
  - src/soLoader/Exceptions.cpp, 86
  - src/soLoader/Exceptions.hpp, 97
  - src/soLoader/SoLoader.hpp, 103
  - Strip.cpp
    - DEFAULT\_STRIP\_TEXT\_SIZE, 87
- TestGame
  - Game::TestGame, 75
- TestGame.cpp
  - \_\_attribute\_\_, 96
  - entry, 96
  - instance, 96
- Text, 77
  - ~Text, 77
  - getText, 78
  - setText, 78
  - Text, 77

toStr  
    Score::File::PlayerData, [63](#)  
TRANSLATE\_COLORS  
    Graphic::Ncurses, [57](#)  
Transparent  
    Color, [27](#)  
  
UnableCreateFolder  
    Core::Exceptions::UnableCreateFolder, [79](#)  
UNKNOWN  
    KeyboardEvent\_s, [52](#)  
UnknownGraphicalLib  
    Core::Exceptions::UnknownGraphicalLib, [80](#)  
UP  
    KeyboardEvent\_s, [52](#)  
useGame  
    Core::Core, [29](#)  
useGraphic  
    Core::Core, [30](#)  
  
Vector2  
    Vector2< T >, [81](#)  
Vector2< T >, [80](#)  
    operator+, [81](#)  
    operator+=", [81](#)  
    operator-, [81](#)  
    operator-=, [82](#)  
    operator==, [82](#)  
    Vector2, [81](#)  
    x, [82](#)  
    y, [82](#)  
Vector2f  
    Drawables.hpp, [89](#)  
Vector2i  
    Drawables.hpp, [89](#)  
Vector2u  
    Drawables.hpp, [89](#)  
Vector3  
    Vector3< T >, [83](#)  
Vector3< T >, [82](#)  
    operator+, [83](#)  
    operator+=", [83](#)  
    operator-, [83](#)  
    operator-=, [84](#)  
    operator==, [84](#)  
    Vector3, [83](#)  
    x, [84](#)  
    y, [84](#)  
    z, [84](#)  
Vector3f  
    Drawables.hpp, [89](#)  
Vector3i  
    Drawables.hpp, [89](#)  
Vector3u  
    Drawables.hpp, [89](#)  
  
what  
    Core::Exceptions::EmptyMandatoryFolder, [32](#)  
    Core::Exceptions::ExitGame, [33](#)  
    Core::Exceptions::InvalidScorePath, [48](#)  
    Core::Exceptions::MissingMandatoryFolder, [54](#)  
    Core::Exceptions::UnableCreateFolder, [79](#)  
    Core::Exceptions::UnknownGraphicalLib, [80](#)  
    Graphic::Exceptions::LoadFontFailed, [53](#)  
    Score::Exceptions::InvalidFile, [47](#)  
    SoLoader::Exceptions::InvalidEntryPoint, [46](#)  
    SoLoader::Exceptions::InvalidSO, [49](#)  
  
White  
    Color, [27](#)  
WINDOW\_HEIGHT  
    Graphic::Ncurses, [58](#)  
    Graphic::SDL2, [68](#)  
    Graphic::SFML, [71](#)  
WINDOW\_WIDTH  
    Graphic::Ncurses, [58](#)  
    Graphic::SDL2, [68](#)  
    Graphic::SFML, [71](#)  
  
x  
    Vector2< T >, [82](#)  
    Vector3< T >, [84](#)  
  
y  
    Vector2< T >, [82](#)  
    Vector3< T >, [84](#)  
  
z  
    Vector3< T >, [84](#)