

GRA RPG

DM & KK

Generated by Doxygen 1.8.18

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 AddHeroe_Window Class Reference	7
4.1.1 Detailed Description	8
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 AddHeroe_Window()	8
4.1.2.2 ~AddHeroe_Window()	8
4.1.3 Member Function Documentation	9
4.1.3.1 Start()	9
4.1.4 Member Data Documentation	9
4.1.4.1 arial	9
4.1.4.2 choose	9
4.1.4.3 dbcontroler	9
4.1.4.4 flaga	9
4.1.4.5 Logo_Button	10
4.1.4.6 Name	10
4.1.4.7 name	10
4.1.4.8 profession	10
4.1.4.9 rectangle_add	10
4.1.4.10 rectangle_choose	10
4.1.4.11 rectangle_name	11
4.1.4.12 window	11
4.2 Archer Class Reference	11
4.2.1 Detailed Description	11
4.2.2 Constructor & Destructor Documentation	12
4.2.2.1 Archer()	12
4.2.3 Member Function Documentation	12
4.2.3.1 attack()	12
4.2.3.2 block()	13
4.2.3.3 clone()	13
4.3 Arena_Window Class Reference	13
4.3.1 Detailed Description	14
4.3.2 Constructor & Destructor Documentation	14
4.3.2.1 Arena_Window()	14
4.3.2.2 ~Arena_Window()	15

4.3.3 Member Function Documentation	15
4.3.3.1 getClickHeal()	15
4.3.3.2 randMonster()	15
4.3.3.3 Start()	15
4.4 Button Class Reference	15
4.4.1 Detailed Description	17
4.4.2 Constructor & Destructor Documentation	18
4.4.2.1 Button() [1/2]	18
4.4.2.2 Button() [2/2]	18
4.4.2.3 ~Button()	18
4.4.3 Member Function Documentation	18
4.4.3.1 getButton()	19
4.4.3.2 getText()	19
4.4.3.3 setButton()	19
4.4.3.4 setText()	19
4.4.4 Member Data Documentation	20
4.4.4.1 button	20
4.4.4.2 font	20
4.4.4.3 text	20
4.4.4.4 write	20
4.5 Champion Class Reference	20
4.5.1 Detailed Description	22
4.5.2 Constructor & Destructor Documentation	22
4.5.2.1 Champion()	22
4.5.3 Member Function Documentation	24
4.5.3.1 addDefence()	24
4.5.3.2 addDexterity()	25
4.5.3.3 addExperience()	25
4.5.3.4 addHealth()	25
4.5.3.5 addLuck()	25
4.5.3.6 addMagic()	27
4.5.3.7 addMoney()	27
4.5.3.8 addStrength()	27
4.5.3.9 attack()	28
4.5.3.10 block()	28
4.5.3.11 clone()	28
4.5.3.12 getDefence()	28
4.5.3.13 getDexterity()	28
4.5.3.14 getExperince()	29
4.5.3.15 getHealth()	29
4.5.3.16 getId()	29
4.5.3.17 getLevel()	29

4.5.3.18 getLuck()	30
4.5.3.19 getMagic()	30
4.5.3.20 getMoney()	30
4.5.3.21 getName()	30
4.5.3.22 getStrength()	31
4.5.3.23 heal()	31
4.5.3.24 setHealth()	31
4.5.3.25 takeMoney()	31
4.5.4 Friends And Related Function Documentation	32
4.5.4.1 operator<<	32
4.5.5 Member Data Documentation	32
4.5.5.1 defence	32
4.5.5.2 dexterity	32
4.5.5.3 experience	32
4.5.5.4 health	32
4.5.5.5 id_champion	32
4.5.5.6 level	33
4.5.5.7 luck	33
4.5.5.8 magic	33
4.5.5.9 money	33
4.5.5.10 name	33
4.5.5.11 strength	33
4.6 Create_Window Class Reference	33
4.6.1 Detailed Description	34
4.6.2 Constructor & Destructor Documentation	34
4.6.2.1 Create_Window()	34
4.6.2.2 ~Create_Window()	35
4.6.3 Member Function Documentation	35
4.6.3.1 Start()	35
4.6.4 Member Data Documentation	35
4.6.4.1 arial	35
4.6.4.2 dbcontroler	35
4.6.4.3 flaga	36
4.6.4.4 Login	36
4.6.4.5 login	36
4.6.4.6 Logo_Button	36
4.6.4.7 Password	36
4.6.4.8 password	36
4.6.4.9 rectangle_create	37
4.6.4.10 rectangle_pass	37
4.6.4.11 rectangle_user	37
4.6.4.12 window	37

4.7 DBcontroler Class Reference	37
4.7.1 Detailed Description	38
4.7.2 Constructor & Destructor Documentation	38
4.7.2.1 DBcontroler()	39
4.7.2.2 ~DBcontroler()	39
4.7.3 Member Function Documentation	39
4.7.3.1 addChampion()	39
4.7.3.2 addHeroes()	39
4.7.3.3 addUser()	40
4.7.3.4 checkUser() [1/2]	40
4.7.3.5 checkUser() [2/2]	40
4.7.3.6 deleteHeroe()	41
4.7.3.7 deleteHeroes()	41
4.7.3.8 getChampion()	41
4.7.3.9 getChampions()	42
4.7.3.10 getProf()	42
4.7.3.11 getUser()	42
4.7.3.12 randomChampion()	43
4.7.3.13 updateChampion()	43
4.8 ErrorShow_Window Class Reference	43
4.8.1 Detailed Description	44
4.8.2 Constructor & Destructor Documentation	44
4.8.2.1 ErrorShow_Window()	44
4.8.2.2 ~ErrorShow_Window()	44
4.8.3 Member Function Documentation	44
4.8.3.1 Start()	44
4.9 Fight Class Reference	44
4.9.1 Detailed Description	45
4.9.2 Constructor & Destructor Documentation	45
4.9.2.1 Fight() [1/2]	45
4.9.2.2 Fight() [2/2]	45
4.9.2.3 ~Fight()	46
4.9.3 Member Function Documentation	46
4.9.3.1 getChamp1()	46
4.9.3.2 getChamp2()	46
4.9.3.3 heal()	46
4.9.3.4 startFigft()	47
4.9.4 Friends And Related Function Documentation	47
4.9.4.1 operator<<	47
4.10 Login_Window Class Reference	47
4.10.1 Detailed Description	48
4.10.2 Constructor & Destructor Documentation	48

4.10.2.1 Login_Window()	48
4.10.2.2 ~Login_Window()	49
4.10.3 Member Function Documentation	49
4.10.3.1 Start()	49
4.10.4 Member Data Documentation	49
4.10.4.1 arial	49
4.10.4.2 dbcontroler	49
4.10.4.3 flaga	49
4.10.4.4 Login	50
4.10.4.5 login	50
4.10.4.6 Logo_Button	50
4.10.4.7 Password	50
4.10.4.8 password	50
4.10.4.9 rectangle_create	50
4.10.4.10 rectangle_log	51
4.10.4.11 rectangle_pass	51
4.10.4.12 rectangle_user	51
4.10.4.13 window	51
4.11 Map_Windows Class Reference	51
4.11.1 Detailed Description	53
4.11.2 Constructor & Destructor Documentation	53
4.11.2.1 Map_Windows() [1/2]	53
4.11.2.2 Map_Windows() [2/2]	53
4.11.2.3 ~Map_Windows()	53
4.11.3 Member Function Documentation	54
4.11.3.1 setPosition_figure()	54
4.11.3.2 Start()	54
4.11.4 Member Data Documentation	54
4.11.4.1 clock	54
4.11.4.2 col_l_d_1	54
4.11.4.3 col_l_d_2	54
4.11.4.4 col_l_u_1	55
4.11.4.5 col_l_u_2	55
4.11.4.6 col_r_d_1	55
4.11.4.7 col_r_d_2	55
4.11.4.8 col_r_d_3	55
4.11.4.9 col_r_u_1	55
4.11.4.10 col_r_u_2	56
4.11.4.11 dbcontroler	56
4.11.4.12 rectSourceSprite	56
4.11.4.13 viewItem_figure	56
4.11.4.14 viewItem_key_arena	56

4.11.4.15 viewItem_key_rest	56
4.11.4.16 viewItem_key_shop	57
4.11.4.17 window	57
4.12 Menu Class Reference	57
4.12.1 Detailed Description	58
4.12.2 Constructor & Destructor Documentation	59
4.12.2.1 Menu() [1/2]	59
4.12.2.2 Menu() [2/2]	59
4.12.2.3 ~Menu()	59
4.12.3 Member Function Documentation	59
4.12.3.1 Start()	59
4.12.4 Member Data Documentation	60
4.12.4.1 add_b	60
4.12.4.2 arial	60
4.12.4.3 background	60
4.12.4.4 backgroundTab	60
4.12.4.5 backgroundImage	60
4.12.4.6 backroundTable	61
4.12.4.7 buttonExit	61
4.12.4.8 buttonStart	61
4.12.4.9 choose	61
4.12.4.10 dbcontroler	61
4.12.4.11 delete_b	61
4.12.4.12 height	62
4.12.4.13 rectangle_choose	62
4.12.4.14 res	62
4.12.4.15 row	62
4.12.4.16 size	62
4.12.4.17 Text_Data	62
4.12.4.18 title	63
4.12.4.19 width	63
4.12.4.20 window	63
4.13 Monster Class Reference	63
4.13.1 Detailed Description	64
4.13.2 Constructor & Destructor Documentation	64
4.13.2.1 Monster() [1/2]	64
4.13.2.2 Monster() [2/2]	64
4.13.3 Member Function Documentation	65
4.13.3.1 attack()	65
4.13.3.2 block()	65
4.13.3.3 clone()	65
4.14 MyException Struct Reference	66

4.14.1 Detailed Description	66
4.14.2 Constructor & Destructor Documentation	66
4.14.2.1 MyException()	66
4.14.3 Member Function Documentation	67
4.14.3.1 what()	67
4.14.4 Member Data Documentation	67
4.14.4.1 error	67
4.15 Restaurant_Window Class Reference	67
4.15.1 Detailed Description	68
4.15.2 Constructor & Destructor Documentation	68
4.15.2.1 Restaurant_Window()	68
4.15.2.2 ~Restaurant_Window()	69
4.15.3 Member Function Documentation	69
4.15.3.1 Start()	69
4.15.4 Member Data Documentation	69
4.15.4.1 arial	69
4.15.4.2 backgroundTab	69
4.15.4.3 backroundTable	69
4.15.4.4 buttonStart	70
4.15.4.5 choose	70
4.15.4.6 dbcontroler	70
4.15.4.7 rectangle_choose	70
4.15.4.8 res	70
4.15.4.9 row	70
4.15.4.10 size	71
4.15.4.11 Text_Data	71
4.16 RestaurantFight_Window Class Reference	71
4.16.1 Detailed Description	72
4.16.2 Constructor & Destructor Documentation	72
4.16.2.1 RestaurantFight_Window()	72
4.16.2.2 ~RestaurantFight_Window()	72
4.16.3 Member Function Documentation	72
4.16.3.1 Start()	72
4.17 Shop_Windows Class Reference	73
4.17.1 Detailed Description	73
4.17.2 Constructor & Destructor Documentation	73
4.17.2.1 Shop_Windows()	73
4.17.2.2 ~Shop_Windows()	74
4.17.3 Member Function Documentation	74
4.17.3.1 moneycheck()	74
4.17.3.2 Start()	74
4.18 View_Item Class Reference	74

4.18.1 Detailed Description	75
4.18.2 Constructor & Destructor Documentation	75
4.18.2.1 View_Item() [1/2]	75
4.18.2.2 View_Item() [2/2]	75
4.18.2.3 ~View_Item()	76
4.18.3 Member Function Documentation	76
4.18.3.1 getImage()	76
4.18.3.2 set_Image()	76
4.18.4 Member Data Documentation	76
4.18.4.1 Image	77
4.18.4.2 item	77
4.19 Warrior Class Reference	77
4.19.1 Detailed Description	77
4.19.2 Constructor & Destructor Documentation	78
4.19.2.1 Warrior()	78
4.19.3 Member Function Documentation	78
4.19.3.1 attack()	78
4.19.3.2 block()	79
4.19.3.3 clone()	79
4.20 Window_Base Class Reference	79
4.20.1 Detailed Description	80
4.20.2 Constructor & Destructor Documentation	80
4.20.2.1 Window_Base() [1/2]	80
4.20.2.2 Window_Base() [2/2]	80
4.20.2.3 ~Window_Base()	81
4.20.3 Member Function Documentation	81
4.20.3.1 getWindows()	81
4.20.4 Member Data Documentation	81
4.20.4.1 background	81
4.20.4.2 backgroundImage	81
4.20.4.3 height	81
4.20.4.4 title	81
4.20.4.5 width	82
4.20.4.6 window	82
4.21 Wizzard Class Reference	82
4.21.1 Detailed Description	82
4.21.2 Constructor & Destructor Documentation	82
4.21.2.1 Wizzard()	83
4.21.3 Member Function Documentation	83
4.21.3.1 attack()	83
4.21.3.2 clone()	84

5 File Documentation	85
5.1 AddHeroe_Window.cpp File Reference	85
5.2 AddHeroe_Window.h File Reference	85
5.3 Archer.cpp File Reference	85
5.4 Archer.h File Reference	86
5.5 Arena_Window.cpp File Reference	86
5.5.1 Function Documentation	86
5.5.1.1 setText()	86
5.6 Arena_Window.h File Reference	86
5.7 Button.cpp File Reference	87
5.8 Button.h File Reference	87
5.9 Champion.cpp File Reference	87
5.9.1 Function Documentation	87
5.9.1.1 operator<<()	87
5.10 Champion.h File Reference	88
5.11 Create_Window.cpp File Reference	88
5.12 Create_Window.h File Reference	88
5.13 DBcontroler.cpp File Reference	88
5.14 DBcontroler.h File Reference	88
5.15 ErrorShow_Window.cpp File Reference	89
5.16 ErrorShow_Window.h File Reference	89
5.17 Fight.cpp File Reference	89
5.17.1 Function Documentation	89
5.17.1.1 operator<<()	89
5.18 Fight.h File Reference	90
5.19 Login_Window.cpp File Reference	90
5.20 Login_Window.h File Reference	90
5.21 Map_Windows.cpp File Reference	90
5.22 Map_Windows.h File Reference	91
5.23 Menu.cpp File Reference	91
5.24 Menu.h File Reference	91
5.25 Monster.cpp File Reference	91
5.26 Monster.h File Reference	92
5.27 Restaurant_Window.cpp File Reference	92
5.28 Restaurant_Window.h File Reference	92
5.29 RestaurantFight_Window.h File Reference	92
5.30 ResteurantFight_Window.cpp File Reference	93
5.30.1 Function Documentation	93
5.30.1.1 setText1()	93
5.31 Shop_Windows.cpp File Reference	93
5.32 Shop_Windows.h File Reference	93
5.33 Source.cpp File Reference	94

5.33.1 Function Documentation	94
5.33.1.1 main()	94
5.34 ThrowLoad.cpp File Reference	94
5.35 View_Item.cpp File Reference	94
5.36 View_Item.h File Reference	94
5.37 Warrior.cpp File Reference	95
5.38 Warrior.h File Reference	95
5.39 Window_Base.cpp File Reference	95
5.40 Window_Base.h File Reference	95
5.41 Wizzard.cpp File Reference	96
5.42 Wizzard.h File Reference	96
Index	97

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Button	15
Champion	20
Archer	11
Monster	63
Warrior	77
Wizzard	82
Create_Window	33
DBcontroler	37
ErrorShow_Window	43
exception	
MyException	66
Fight	44
Login_Window	47
View_Item	74
Window_Base	79
AddHeroe_Window	7
Arena_Window	13
Map_Windows	51
Menu	57
Restaurant_Window	67
RestaurantFight_Window	71
Shop_Windows	73

Chapter 2

Class Index

2.1 Class List

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

AddHeroe_Window	Class window support add character to user	7
Archer	Class Archer Inherits a champion	11
Arena_Window	Class arena to representation graphic Window object Arena	13
Button	Class to creat object button	15
Champion	Class Champion use to create character	20
Create_Window	Class window support create user	33
DBcontroler	Class responsible for database operation	37
ErrorShow_Window	Classe represenr window to show a error messge	43
Fight	Class support fighting between player vs monster or player vs player	44
Login_Window	Class window support login user	47
Map_Windows	Class use to show graphic city and supporting all activities Inherits a Window_Base	51
Menu	Class Menu represent window Menu inherit Window_Base	57
Monster	Class Monster Inherits a champion	63
MyException	Struct to service exception	66
Restaurant_Window	Class restaurant to representation graphic Window object restaurant inherit Window_Base	67
RestaurantFight_Window	Class to representation graphic Window object Fight betwen players	71
Shop_Windows	Class shop to representation graphic Window object shop	73
View_Item	Class use to representation some graphics	74

Warrior	Class Warrior Inherits a champion	77
Window_Base	Base class use to make graphic interface	79
Wizzard	Class Wizzard Inherits a champion	82

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

AddHeroe_Window.cpp	85
AddHeroe_Window.h	85
Archer.cpp	85
Archer.h	86
Arena_Window.cpp	86
Arena_Window.h	86
Button.cpp	87
Button.h	87
Champion.cpp	87
Champion.h	88
Create_Window.cpp	88
Create_Window.h	88
DBcontroler.cpp	88
DBcontroler.h	88
ErrorShow_Window.cpp	89
ErrorShow_Window.h	89
Fight.cpp	89
Fight.h	90
Login_Window.cpp	90
Login_Window.h	90
Map_Windows.cpp	90
Map_Windows.h	91
Menu.cpp	91
Menu.h	91
Monster.cpp	91
Monster.h	92
Restaurant_Window.cpp	92
Restaurant_Window.h	92
RestaurantFight_Window.h	92
ResteurantFight_Window.cpp	93
Shop_Windows.cpp	93
Shop_Windows.h	93
Source.cpp	94
ThrowLoad.cpp	94
View_Item.cpp	94

View_Item.h	94
Warrior.cpp	95
Warrior.h	95
Window_Base.cpp	95
Window_Base.h	95
Wizzard.cpp	96
Wizzard.h	96

Chapter 4

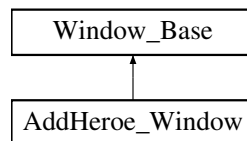
Class Documentation

4.1 AddHeroe_Window Class Reference

Class window support add character to user.

```
#include <AddHeroe_Window.h>
```

Inheritance diagram for AddHeroe_Window:



Public Member Functions

- [AddHeroe_Window](#) (int idUser)
Constructor window.
- [~AddHeroe_Window](#) ()
Deconstructor window.
- void [Start](#) ()
Function start showing window with user creation new character.

Public Attributes

- [Window_Base](#) window
Base window class use to create window.
- Sprite [Logo_Button](#)
Button to validation add champion to user.
- Font [arial](#)
Used font.
- Text [Name](#)
Name of new character.
- [DBcontroler](#) dbcontroler

- Data base controller.*
 - string [name](#)
 - Name of new character.*
 - int [profession](#)
 - Profession id.*
 - int [flaga](#)
 - Flag to selection class profession.*
 - int [choose](#)
 - Player selection witch class selected.*
 - RectangleShape [rectangle_name](#)
 - Mouse click field to enter user name character.*
 - RectangleShape [rectangle_add](#)
 - Mouse click field to validate the data.*
 - RectangleShape * [rectangle_choose](#)
 - Mouse click field to check proffesion of character.*

Additional Inherited Members

4.1.1 Detailed Description

Class window support add character to user.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 AddHeroe_Window()

```
AddHeroe_Window::AddHeroe_Window (
    int idUser )
```

Constructor window.

Parameters

<i>idUser</i>	user's id
---------------	-----------

4.1.2.2 ~AddHeroe_Window()

```
AddHeroe_Window::~~AddHeroe_Window ( )
```

Deconstructor window.

4.1.3 Member Function Documentation

4.1.3.1 Start()

```
void AddHeroe_Window::Start ( )
```

Function start showing window with user creation new character.

4.1.4 Member Data Documentation

4.1.4.1 arial

```
Font AddHeroe_Window::arial
```

Used font.

4.1.4.2 choose

```
int AddHeroe_Window::choose
```

Player selection witch class selected.

4.1.4.3 dbcontroler

```
DBcontroler AddHeroe_Window::dbcontroler
```

Data base controller.

4.1.4.4 flaga

```
int AddHeroe_Window::flaga
```

Flag to selection class profession.

4.1.4.5 Logo_Button

```
Sprite AddHeroe_Window::Logo_Button
```

[Button](#) to validation add champion to user.

4.1.4.6 Name

```
Text AddHeroe_Window::Name
```

Name of new character.

4.1.4.7 name

```
string AddHeroe_Window::name
```

Name of new character.

4.1.4.8 profession

```
int AddHeroe_Window::profession
```

Profession id.

4.1.4.9 rectangle_add

```
RectangleShape AddHeroe_Window::rectangle_add
```

Mouse click field to validate the data.

4.1.4.10 rectangle_choose

```
RectangleShape* AddHeroe_Window::rectangle_choose
```

Mouse click field to check proffesion of character.

4.1.4.11 rectangle_name

`RectangleShape AddHeroe_Window::rectangle_name`

Mouse click field to enter user name character.

4.1.4.12 window

`Window_Base AddHeroe_Window::window`

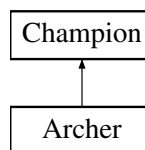
Base window class use to create window.

4.2 Archer Class Reference

Class `Archer` Inherits a champion.

```
#include <Archer.h>
```

Inheritance diagram for Archer:



Public Member Functions

- `Archer` (const unsigned id, const std::string n, unsigned int l, unsigned int e, double m, int he, int st, int de, int ma, int def, int lu)
Constructor `Archer` cause a champion constructor.
- virtual int `attack` () override
Virtual function attack.
- virtual int `block` () override
Virtual function block.
- virtual `Archer * clone` () const
Virtual function clone to clone object.

Additional Inherited Members

4.2.1 Detailed Description

Class `Archer` Inherits a champion.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Archer()

```
Archer::Archer (
    const unsigned id,
    const std::string n,
    unsigned int l,
    unsigned int e,
    double m,
    int he,
    int st,
    int de,
    int ma,
    int def,
    int lu ) [inline]
```

Constructor [Archer](#) cause a champion constructor.

Parameters

<i>write</i>	- The inscription that will be shown on the button
<i>id</i>	- Id champion in database
<i>n</i>	- Name champion
<i>l</i>	- Level champion
<i>e</i>	- Experience champion
<i>m</i>	- Money champion
<i>he</i>	- Health champion
<i>st</i>	- Strength champion
<i>de</i>	- Dexerity Champion
<i>ma</i>	- Magic champion
<i>def</i>	- Defence champion
<i>lu</i>	- Lucky champion

4.2.3 Member Function Documentation

4.2.3.1 attack()

```
int Archer::attack ( ) [override], [virtual]
```

Virtual function attack.

Returns

value attack

Reimplemented from [Champion](#).

4.2.3.2 block()

```
int Archer::block ( ) [override], [virtual]
```

Virtual function block.

Returns

value block

Reimplemented from [Champion](#).

4.2.3.3 clone()

```
virtual Archer* Archer::clone ( ) const [inline], [virtual]
```

Virtual function clone to clone object.

Returns

Pointer object [Archer](#)

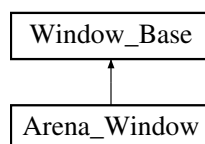
Implements [Champion](#).

4.3 Arena_Window Class Reference

Class arena to representation graphic Window object Arena.

```
#include <Arena_Window.h>
```

Inheritance diagram for Arena_Window:



Public Member Functions

- [Arena_Window](#) ([Champion](#) *champ, int idUse)

Constructor [Arena_Window](#)

Parameters

Champ	- Object type Champion
idUse	- Parameter type int to storage id User

- [~Arena_Window](#) ()
Destructor Class Arena_Windows.
- void [Start](#) ()
Function Start to show graphic interface.
- RectangleShape [getClickHeal](#) ()
Function return object type RectangleShape to show some graphics

Parameters

Champ	- Object type Champion
idUse	- Parameter type int to storage id User

*Returns**Object type RectangleShape*

- void [randMonster](#) ()
Function use to random monsters.

Additional Inherited Members**4.3.1 Detailed Description**

Class arena to representation graphic Window object Arena.

4.3.2 Constructor & Destructor Documentation**4.3.2.1 Arena_Window()**

```
Arena_Window::Arena_Window (
    Champion * champ,
    int idUse )
```

Constructor [Arena_Window](#)

Parameters

<i>Champ</i>	- Object type Champion
<i>idUse</i>	- Parameter type int to storage id User

4.3.2.2 ~Arena_Window()

```
Arena_Window::~~Arena_Window ( )
```

Destructor Class Arena_Windows.

4.3.3 Member Function Documentation

4.3.3.1 getClickHeal()

```
RectangleShape Arena_Window::getClickHeal ( ) [inline]
```

Function return object type RectangleShape to show some graphics

Parameters

<i>Champ</i>	- Object type Champion
<i>idUse</i>	- Parameter type int to storage id User

Returns

Object type RectangleShape

4.3.3.2 randMonster()

```
void Arena_Window::randMonster ( )
```

Function use to random monsters.

4.3.3.3 Start()

```
void Arena_Window::Start ( )
```

Function Start to show graphic interface.

4.4 Button Class Reference

Class to creat object button.

```
#include <Button.h>
```

Public Member Functions

- [Button](#) ()
Constructor Class [Button](#).
- [Button](#) (String [text](#), int pos_x, int pos_y, Color color, int text_pos_x, int text_pos_y)
Constructor Class [Button](#)

Parameters

text	- Text on the button
pos_x	- Position width button
pos_y	- Position height button
color	- Color button
text_↔ pos_x	- Position width text on button
text_↔ pos_y	- Position height text on button

Returns

- [~Button](#) ()
Destructor [Button](#).
- void [setButton](#) (String [write](#), int width, int height, Color color, int pos_x, int pos_y, int pos_x_txt, int pos_y_txt)
Function to set value

Parameters

write	- The inscription that will be shown on the button
width	- Width button
height	- Height button
color	- Color background button
pos_x	- Position x button
pos_y	- Position y button
pos_x↔ _txt	- Possition x text on button
pos_y↔ _txt	- Possition y text on button

- RectangleShape [getButton](#) ()
Function get RectangleShape use to show create graphic button.
- Text [getText](#) ()
Function return object Text to printig text.
- void [setText](#) (string [write](#))
Function set new text on button.

Public Attributes

- String [write](#)
String to keep text button.
- RectangleShape [button](#)
Object temporary use to create view button.
- Text [text](#)
Object use to show string button.
- Font [font](#)
Object font use load Font used for entering text.

4.4.1 Detailed Description

Class to creat object button.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Button() [1/2]

```
Button::Button ( )
```

Constructor Class [Button](#).

4.4.2.2 Button() [2/2]

```
Button::Button (
    String text,
    int pos_x,
    int pos_y,
    Color color,
    int text_pos_x,
    int text_pos_y )
```

Constructor Class [Button](#)

Parameters

<i>text</i>	- Text on the button
<i>pos_x</i>	- Position width button
<i>pos_y</i>	- Position height button
<i>color</i>	- Color button
<i>text_pos_x</i>	- Position width text on button
<i>text_pos_y</i>	- Position height text on button

Returns

4.4.2.3 ~Button()

```
Button::~~Button ( )
```

Destructor [Button](#).

4.4.3 Member Function Documentation

4.4.3.1 getButton()

```
RectangleShape Button::getButton ( )
```

Function get RectangleShape use to show create graphic button.

4.4.3.2 getText()

```
Text Button::getText ( )
```

Function return object Text to printnig text.

4.4.3.3 setButton()

```
void Button::setButton (
    String write,
    int width,
    int height,
    Color color,
    int pos_x,
    int pos_y,
    int pos_x_txt,
    int pos_y_txt )
```

Function to set value

Parameters

<i>write</i>	- The inscription that will be shown on the button
<i>width</i>	- Width button
<i>height</i>	- Height button
<i>color</i>	- Color background button
<i>pos_x</i>	- Position x button
<i>pos_y</i>	- Position y button
<i>pos_x_txt</i>	- Possition x text on button
<i>pos_y_txt</i>	- Possition y text on button

4.4.3.4 setText()

```
void Button::setText (
    string write )
```

Function set new text on button.

4.4.4 Member Data Documentation

4.4.4.1 button

```
RectangleShape Button::button
```

Object temporary use to create view button.

4.4.4.2 font

```
Font Button::font
```

Object font use load Font used for entering text.

4.4.4.3 text

```
Text Button::text
```

Object use to show string button.

4.4.4.4 write

```
String Button::write
```

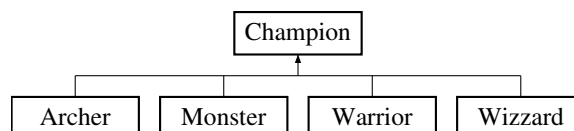
String to keep text button.

4.5 Champion Class Reference

Class [Champion](#) use to create character.

```
#include <Champion.h>
```

Inheritance diagram for Champion:



Public Member Functions

- [Champion](#) (const unsigned id, const std::string n, unsigned int l, unsigned int e, double m, int he, int st, int de, int ma, int def, int lu)
Constructor [Champion](#).
- virtual int [attack](#) ()
Virtual function attack.
- virtual int [block](#) ()
Virtual function block.
- int [heal](#) ()
Function return Helath the value to heal the hero.
- void [setHealth](#) (int hp)
Function set Health.
- virtual [Champion](#) * [clone](#) () const =0
Virtual method clone copy pointer.
- int [getHealth](#) () const
Function get Health champion.
- std::string [getName](#) ()
Function get Name champion.
- void [addHealth](#) (int hp)
Function add Health.
- void [addStrength](#) (int stren)
Function add Strength.
- void [addDexterity](#) (int dex)
Function add Dexterity.
- void [addMagic](#) (int mag)
Function add Magic.
- void [addDefence](#) (int def)
Function add Health.
- void [addLuck](#) (int luc)
Function add Lucky.
- void [addMoney](#) (int mon)
Function add Money.
- void [addExperience](#) (int exp)
Function add Experience.
- void [takeMoney](#) (double mon)
Function add Money.
- int [getId](#) ()
Get Id.
- int [getExperience](#) ()
Get Experience.
- int [getLevel](#) ()
Get Level.
- int [getMoney](#) ()
Get Money.
- int [getStrength](#) ()
Get Strength.
- int [getDexterity](#) ()
Get Dexterity.
- int [getMagic](#) ()
Get Magic.

- int [getDefence](#) ()
Get Defence.
- int [getLuck](#) ()
Get Lucky.

Protected Attributes

- const unsigned int [id_champion](#)
- const std::string [name](#)
- unsigned int [level](#)
- unsigned int [experience](#)
- double [money](#)
- int [health](#)
- int [strength](#)
- int [dexterity](#)
- int [magic](#)
- int [defence](#)
- int [luck](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [Champion](#) &p)
Showing a value represent champion.

4.5.1 Detailed Description

Class [Champion](#) use to create character.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 [Champion\(\)](#)

```
Champion::Champion (  
    const unsigned id,  
    const std::string n,  
    unsigned int l,  
    unsigned int e,  
    double m,  
    int he,  
    int st,  
    int de,  
    int ma,  
    int def,  
    int lu )
```

Constructor [Champion](#).

Parameters

<i>write</i>	- The inscription that will be shown on the button
--------------	--

Parameters

<i>id</i>	- Id champion in database
-----------	---------------------------

Parameters

<i>n</i>	- Name champion
----------	-----------------

Parameters

<i>l</i>	- Level champion
----------	------------------

Parameters

<i>e</i>	- Experience champion
----------	-----------------------

Parameters

<i>m</i>	- Money champion
----------	------------------

Parameters

<i>he</i>	- Health champion
-----------	-------------------

Parameters

<i>st</i>	- Strength champion
-----------	---------------------

Parameters

<i>de</i>	- Dexerity Champion
-----------	-------------------------------------

Parameters

<i>ma</i>	- Magic champion
-----------	------------------

Parameters

<i>def</i>	- Defence champion
------------	--------------------

Parameters

<i>lu</i>	- Lucky champion
-----------	------------------

4.5.3 Member Function Documentation

4.5.3.1 addDefence()

```
void Champion::addDefence (  
    int def ) [inline]
```

Function add Health.

Parameters

<i>hp</i>	- Value to addition to health
-----------	-------------------------------

4.5.3.2 addDexterity()

```
void Champion::addDexterity (
    int dex ) [inline]
```

Function add Dexterity.

Parameters

<i>hp</i>	- Value to addition to Dexterity/param>
-----------	---

4.5.3.3 addExperience()

```
void Champion::addExperience (
    int exp )
```

Function add Experience.

Parameters

<i>hp</i>	- Value to addition to experience
-----------	-----------------------------------

4.5.3.4 addHealth()

```
void Champion::addHealth (
    int hp ) [inline]
```

Function add Health.

Parameters

<i>hp</i>	- Value to addition to health
-----------	-------------------------------

4.5.3.5 addLuck()

```
void Champion::addLuck (
    int luc ) [inline]
```

Function add Lucky.

Parameters

<i>hp</i>	- Value to addition to lucky
-----------	------------------------------

4.5.3.6 addMagic()

```
void Champion::addMagic (  
    int mag ) [inline]
```

Function add Magic.

Parameters

<i>hp</i>	- Value to addition to magic
-----------	------------------------------

4.5.3.7 addMoney()

```
void Champion::addMoney (  
    int mon ) [inline]
```

Function add Money.

Parameters

<i>hp</i>	- Value to addition to Money
-----------	------------------------------

4.5.3.8 addStrength()

```
void Champion::addStrength (  
    int stren ) [inline]
```

Function add Strength.

Parameters

<i>hp</i>	- Value to addition to strength
-----------	---------------------------------

4.5.3.9 attack()

```
virtual int Champion::attack ( ) [inline], [virtual]
```

Virtual function attack.

Reimplemented in [Warrior](#), [Monster](#), [Archer](#), and [Wizzard](#).

4.5.3.10 block()

```
virtual int Champion::block ( ) [inline], [virtual]
```

Virtual function block.

Reimplemented in [Warrior](#), [Monster](#), and [Archer](#).

4.5.3.11 clone()

```
virtual Champion* Champion::clone ( ) const [pure virtual]
```

Virtual method clone copy pointer.

Implemented in [Monster](#), [Archer](#), [Wizzard](#), and [Warrior](#).

4.5.3.12 getDefence()

```
int Champion::getDefence ( ) [inline]
```

Get Defence.

Returns

Experience

4.5.3.13 getDexterity()

```
int Champion::getDexterity ( ) [inline]
```

Get Dexterity.

Returns

Dexterity

4.5.3.14 getExperince()

```
int Champion::getExperince ( ) [inline]
```

Get Experience.

Returns

Experience

4.5.3.15 getHealth()

```
int Champion::getHealth ( ) const [inline]
```

Function get Health champion.

4.5.3.16 getId()

```
int Champion::getId ( ) [inline]
```

Get Id.

Returns

Id champion

4.5.3.17 getLevel()

```
int Champion::getLevel ( ) [inline]
```

Get Level.

Returns

Level

4.5.3.18 getLuck()

```
int Champion::getLuck ( ) [inline]
```

Get Lucky.

Returns

Lucky

4.5.3.19 getMagic()

```
int Champion::getMagic ( ) [inline]
```

Get Magic.

Returns

Magic

4.5.3.20 getMoney()

```
int Champion::getMoney ( ) [inline]
```

Get Money.

Returns

Money

4.5.3.21 getName()

```
std::string Champion::getName ( ) [inline]
```

Function get Name champion.

4.5.3.22 getStrength()

```
int Champion::getStrength ( ) [inline]
```

Get Strength.

Returns

Strength

4.5.3.23 heal()

```
int Champion::heal ( )
```

Function return Helath the value to heal the hero.

4.5.3.24 setHealth()

```
void Champion::setHealth (
    int hp ) [inline]
```

Function set Health.

Parameters

<i>hp</i>	- Helth champion
-----------	------------------

4.5.3.25 takeMoney()

```
void Champion::takeMoney (
    double mon )
```

Function add Money.

Parameters

<i>hp</i>	- Value to addition to money
-----------	------------------------------

4.5.4 Friends And Related Function Documentation

4.5.4.1 operator<<

```
std::ostream& operator<< (  
    std::ostream & out,  
    const Champion & p ) [friend]
```

Showing a value represent champion.

4.5.5 Member Data Documentation

4.5.5.1 defence

```
int Champion::defence [protected]
```

4.5.5.2 dexterity

```
int Champion::dexterity [protected]
```

4.5.5.3 experience

```
unsigned int Champion::experience [protected]
```

4.5.5.4 health

```
int Champion::health [protected]
```

4.5.5.5 id_champion

```
const unsigned int Champion::id\_champion [protected]
```

4.5.5.6 level

```
unsigned int Champion::level [protected]
```

4.5.5.7 luck

```
int Champion::luck [protected]
```

4.5.5.8 magic

```
int Champion::magic [protected]
```

4.5.5.9 money

```
double Champion::money [protected]
```

4.5.5.10 name

```
const std::string Champion::name [protected]
```

4.5.5.11 strength

```
int Champion::strength [protected]
```

4.6 Create_Window Class Reference

Class window support create user.

```
#include <Create_Window.h>
```

Public Member Functions

- [Create_Window](#) (int idUse)
Constructor window.
- [~Create_Window](#) ()
Destructor window.
- void [Start](#) ()
Function start showing window with user creation.

Public Attributes

- [Window_Base](#) window
Base window class use to create window.
- Sprite [Logo_Button](#)
Button to validation adding user.
- Font [arial](#)
Used font.
- Text [Login](#)
Display entered login.
- Text [Password](#)
Display entered password.
- string [login](#)
Entered login.
- string [password](#)
Entered password.
- [DBcontroler](#) dbcontroler
Data base controller.
- RectangleShape [rectangle_user](#)
Mouse click field to enter user login.
- RectangleShape [rectangle_pass](#)
Mouse click field to enter user password.
- RectangleShape [rectangle_create](#)
Mouse click field to validate the data.
- int [flaga](#)
Flag checking which field is clicked.

4.6.1 Detailed Description

Class window support create user.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Create_Window()

```
Create_Window::Create_Window (
    int idUse )
```

Constructor window.

Parameters

<i>idUse</i>	user's id
--------------	-----------

4.6.2.2 ~Create_Window()

```
Create_Window::~~Create_Window ( )
```

Deconstructor window.

4.6.3 Member Function Documentation

4.6.3.1 Start()

```
void Create_Window::Start ( )
```

Function start showing window with user creation.

4.6.4 Member Data Documentation

4.6.4.1 arial

```
Font Create_Window::arial
```

Used font.

4.6.4.2 dbcontroler

```
DBcontroler Create_Window::dbcontroler
```

Data base controller.

4.6.4.3 flaga

```
int Create_Window::flaga
```

Flag checking which field is clicked.

4.6.4.4 Login

```
Text Create_Window::Login
```

Display entered login.

4.6.4.5 login

```
string Create_Window::login
```

Entered login.

4.6.4.6 Logo_Button

```
Sprite Create_Window::Logo_Button
```

[Button](#) to validation adding user.

4.6.4.7 Password

```
Text Create_Window::Password
```

Display entered password.

4.6.4.8 password

```
string Create_Window::password
```

Entered password.

4.6.4.9 rectangle_create

RectangleShape Create_Window::rectangle_create

Mouse click field to validate the data.

4.6.4.10 rectangle_pass

RectangleShape Create_Window::rectangle_pass

Mouse click field to enter user password.

4.6.4.11 rectangle_user

RectangleShape Create_Window::rectangle_user

Mouse click field to enter user login.

4.6.4.12 window

Window_Base Create_Window::window

Base window class use to create window.

4.7 DBcontroler Class Reference

Class responsible for database operation.

```
#include <DBcontroler.h>
```

Public Member Functions

- [DBcontroler](#) ()
Constructor establishing a connection to the database.
- [~DBcontroler](#) ()
Destructor disconnection to the database.
- bool [checkUser](#) (int id)
Function check the user exists in the database.
- bool [checkUser](#) (std::string login)
Function check the user exists in the database.
- void [updateChampion](#) ([Champion](#) *champ)
Update function for data downloaded from player and amending the values in the database.
- void [addChampion](#) ([Champion](#) *champ, int class_hero, int idUser)
Function adding a new character to the database.
- void [addHeroes](#) (int idUser)
Function to assign the created champion to the user.
- void [deleteHeroes](#) (int idUser, int idChampion)
function removing to player's character.
- void [deleteHeroe](#) (int idUser)
function removing to player's character.
- bool [addUser](#) (std::string login, std::string password)
Function adding a new user.
- MYSQL_ROW [getUser](#) (std::string login, std::string password)
Function finding user id.
- MYSQL_RES * [getChampions](#) (int id)
Function finding all player's champions.
- MYSQL_ROW [getProf](#) (int id)
Function finds what class the characters is.
- [Champion](#) * [getChampion](#) (int id)
Function creates a new character

Parameters

id	- champion's id
----	-----------------

Returns

Pointer to new character

- MYSQL_RES * [randomChampion](#) (int id)
Function finding 3 random character different to given id

4.7.1 Detailed Description

Class responsible for database operation.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 DBcontroler()

```
DBcontroler::DBcontroler ( )
```

Constructor establishing a connection to the database.

4.7.2.2 ~DBcontroler()

```
DBcontroler::~~DBcontroler ( )
```

Destructor disconnection to the database.

4.7.3 Member Function Documentation

4.7.3.1 addChampion()

```
void DBcontroler::addChampion (
    Champion * champ,
    int class_hero,
    int idUser )
```

Function adding a new character to the database.

Parameters

<i>champ</i>	- Champion class object with generated basic statistics
<i>class_hero</i>	- player's chosen character class
<i>idUser</i>	- player's id

4.7.3.2 addHeroes()

```
void DBcontroler::addHeroes (
    int idUser )
```

Function to assign the created champion to the user.

Parameters

<i>idUser</i>	- player's id
---------------	---------------

4.7.3.3 addUser()

```
bool DBcontroller::addUser (
    std::string login,
    std::string password )
```

Function adding a new user.

Parameters

<i>login</i>	- player's login
<i>password</i>	- player's password

Returns

confirm adding new user - true or false

4.7.3.4 checkUser() [1/2]

```
bool DBcontroller::checkUser (
    int id )
```

Function check the user exists in the database.

Parameters

<i>id</i>	- id user
-----------	-----------

Returns

true or false

4.7.3.5 checkUser() [2/2]

```
bool DBcontroller::checkUser (
    std::string login )
```

Function check the user exists in the database.

Parameters

<i>login</i>	- login given by user
--------------	-----------------------

Returns

true or false

4.7.3.6 deleteHeroe()

```
void DBcontroler::deleteHeroe (
    int idUser )
```

function removing to player's character.

Parameters

<i>idUser</i>	- player's id
---------------	---------------

4.7.3.7 deleteHeroes()

```
void DBcontroler::deleteHeroes (
    int idUser,
    int idChampion )
```

function removing to player's character.

Parameters

<i>idUser</i>	- player's id
<i>idChampion</i>	- player's character id

4.7.3.8 getChampion()

```
Champion * DBcontroler::getChampion (
    int id )
```

Function creates a new character

Parameters

<i>id</i>	- champion's id
-----------	-----------------

Returns

Pointer to new character

4.7.3.9 getChampions()

```
MYSQL_RES * DBcontroler::getChampions (
    int id )
```

Function finding all player's champions.

Parameters

<i>id</i>	- user's id
-----------	-------------

Returns

Rows with all player's champions

4.7.3.10 getProf()

```
MYSQL_ROW DBcontroler::getProf (
    int id )
```

Function finds what class the characters is.

Parameters

<i>id</i>	- champion's id
-----------	-----------------

Returns

Row with id proffesion

4.7.3.11 getUser()

```
MYSQL_ROW DBcontroler::getUser (
    std::string login,
    std::string password )
```

Function finding user id.

Parameters

<i>login</i>	- user's login
<i>password</i>	- user's password

Returns

Row with user's id

4.7.3.12 randomChampion()

```
MYSQL_RES * DBcontroler::randomChampion (
    int id )
```

Function finding 3 random character different to given id

Parameters

<i>id</i>	- character's id
-----------	------------------

Returns

Rows with champions data

4.7.3.13 updateChampion()

```
void DBcontroler::updateChampion (
    Champion * champ )
```

Update function for data downloaded from player and amending the values in the database.

Parameters

<i>champ</i>	- player's object
--------------	-------------------

4.8 ErrorShow_Window Class Reference

Classe represnr window to show a error messge.

```
#include <ErrorShow_Window.h>
```

Public Member Functions

- [ErrorShow_Window](#) ()
Constructor object [ErrorShow_Window](#).
- [~ErrorShow_Window](#) ()
Destructor object [ErrorShow_Window](#).
- void [Start](#) (string err)
Function Start to show graphic interface.

4.8.1 Detailed Description

Classe represenr window to show a error messge.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 ErrorShow_Window()

```
ErrorShow_Window::ErrorShow_Window ( )
```

Constructor object [ErrorShow_Window](#).

4.8.2.2 ~ErrorShow_Window()

```
ErrorShow_Window::~~ErrorShow_Window ( )
```

Destructor object [ErrorShow_Window](#).

4.8.3 Member Function Documentation

4.8.3.1 Start()

```
void ErrorShow_Window::Start (
    string err )
```

Function Start to show graphic interface.

4.9 Fight Class Reference

Class support fighting between player vs monster or player vs player.

```
#include <Fight.h>
```

Public Member Functions

- [Fight](#) ()
Default constructor class [Fight](#).
- [Fight](#) ([Champion](#) *c1, [Champion](#) *c2)
Constructor Class [Fight](#)

Parameters

c1	- pointer to first character
c2	- pointer of second character

Returns

- [~Fight](#) ()
Destructor class [Fight](#).
- void [heal](#) (const [Champion](#) *const champ)
Function set health.
- void [startFight](#) ()
Function starts fight between characters.
- string [getChamp1](#) ()
Get health from first character.
- string [getChamp2](#) ()
Get health from second character.

Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [Fight](#) &p)

4.9.1 Detailed Description

Class support fighting between player vs monster or player vs player.

4.9.2 Constructor & Destructor Documentation**4.9.2.1 Fight() [1/2]**

```
Fight::Fight ( ) [inline]
```

Default constructor class [Fight](#).

4.9.2.2 Fight() [2/2]

```
Fight::Fight (
    Champion * c1,
    Champion * c2 ) [inline]
```

Constructor Class [Fight](#)

Parameters

<i>c1</i>	- pointer to first character
<i>c2</i>	- pointer of second character

Returns**4.9.2.3 ~Fight()**

```
Fight::~~Fight ( ) [inline]
```

Destructor class [Fight](#).

4.9.3 Member Function Documentation**4.9.3.1 getChamp1()**

```
string Fight::getChamp1 ( ) [inline]
```

Get health from first character.

Returns

Health as String

4.9.3.2 getChamp2()

```
string Fight::getChamp2 ( ) [inline]
```

Get health from second character.

Returns

Health as String

4.9.3.3 heal()

```
void Fight::heal (
    const Champion *const champ )
```

Function set health.

Parameters

<i>champ</i>	- Basic helth champion
--------------	------------------------

4.9.3.4 startFight()

```
void Fight::startFigft ( )
```

Function starts fight between characters.

4.9.4 Friends And Related Function Documentation**4.9.4.1 operator<<**

```
std::ostream& operator<< (
    std::ostream & out,
    const Fight & p ) [friend]
```

4.10 Login_Window Class Reference

Class window support login user.

```
#include <Login_Window.h>
```

Public Member Functions

- [Login_Window](#) ()
Constructor window.
- [~Login_Window](#) ()
Destructor window.
- void [Start](#) ()
Function start showing window with user creation.

Public Attributes

- [Window_Base window](#)
Base window class use to create window.
- Sprite [Logo_Button](#)
Button to validation login user.
- Font [arial](#)
Used font.
- Text [Login](#)
Display entered login.
- Text [Password](#)
Display entered password.
- string [login](#)
Entered login.
- string [password](#)
Entered password.
- [DBcontroler dbcontroler](#)
Data base controller.
- [RectangleShape rectangle_log](#)
Mouse click field to validate the data login.
- [RectangleShape rectangle_user](#)
Mouse click field to enter user login.
- [RectangleShape rectangle_pass](#)
Mouse click field to enter user password.
- [RectangleShape rectangle_create](#)
Mouse click field to go rejestracion window.
- int [flaga](#)
Flag checking which field is clicked.

4.10.1 Detailed Description

Class window support login user.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Login_Window()

```
Login_Window::Login_Window ( )
```

Constructor window.

4.10.2.2 ~Login_Window()

```
Login_Window::~Login_Window ( )
```

Deconstructor window.

4.10.3 Member Function Documentation

4.10.3.1 Start()

```
void Login_Window::Start ( )
```

Function start showing window with user creation.

4.10.4 Member Data Documentation

4.10.4.1 arial

```
Font Login_Window::arial
```

Used font.

4.10.4.2 dbcontroler

```
DBcontroler Login_Window::dbcontroler
```

Data base controller.

4.10.4.3 flaga

```
int Login_Window::flaga
```

Flag checking which field is clicked.

4.10.4.4 Login

`Text Login_Window::Login`

Display entered login.

4.10.4.5 login

`string Login_Window::login`

Entered login.

4.10.4.6 Logo_Button

`Sprite Login_Window::Logo_Button`

[Button](#) to validation login user.

4.10.4.7 Password

`Text Login_Window::Password`

Display entered password.

4.10.4.8 password

`string Login_Window::password`

Entered password.

4.10.4.9 rectangle_create

`RectangleShape Login_Window::rectangle_create`

Mouse click field to go rejestracion window.

4.10.4.10 rectangle_log

```
RectangleShape Login_Window::rectangle_log
```

Mouse click field to validate the data login.

4.10.4.11 rectangle_pass

```
RectangleShape Login_Window::rectangle_pass
```

Mouse click field to enter user password.

4.10.4.12 rectangle_user

```
RectangleShape Login_Window::rectangle_user
```

Mouse click field to enter user login.

4.10.4.13 window

```
Window_Base Login_Window::window
```

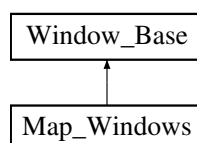
Base window class use to create window.

4.11 Map_Windows Class Reference

Class use to show graphic city and supporting all activities Inherits a [Window_Base](#).

```
#include <Map_Windows.h>
```

Inheritance diagram for Map_Windows:



Public Member Functions

- [Map_Windows](#) (String [title](#), int [width](#), int [height](#), String [backgroundPath](#), [Champion](#) *champ, int [idUser](#))
Constructor Class [Map_Windows](#).
- [Map_Windows](#) ()
Default constructor Class [Map_Windows](#).
- [~Map_Windows](#) ()
Destructor [Map_Windows](#).
- void [Start](#) ()
Function start showing a graphic interface.
- void [setPosition_figure](#) (int, int)
Function to set position character after back to a other activity.

Public Attributes

- [Window_Base](#) [window](#)
Object base class use to create graphic window interface.
- [View_Item](#) [viewItem_key_shop](#)
Object class [View_Item](#) use to show some graphics image.
- [View_Item](#) [viewItem_key_arena](#)
Object class [View_Item](#) use to show some graphics image.
- [View_Item](#) [viewItem_key_rest](#)
Object class [View_Item](#) use to show some graphics image.
- [View_Item](#) [viewItem_figure](#)
Object class [View_Item](#) use to show some graphics image.
- sf::Clock [clock](#)
Object helping to create animations.
- IntRect [rectSourceSprite](#)
Object helping to create animations.
- DBcontroler [dbcontroler](#)
Object use to connect a database.
- RectangleShape [col_l_u_1](#)
Object use to collision on a some object.
- RectangleShape [col_l_u_2](#)
Object use to collision on a some object.
- RectangleShape [col_r_u_1](#)
Object use to collision on a some object.
- RectangleShape [col_r_u_2](#)
Object use to collision on a some object.
- RectangleShape [col_l_d_1](#)
Object use to collision on a some object.
- RectangleShape [col_l_d_2](#)
Object use to collision on a some object.
- RectangleShape [col_r_d_1](#)
Object use to collision on a some object.
- RectangleShape [col_r_d_2](#)
Object use to collision on a some object.
- RectangleShape [col_r_d_3](#)
Object use to collision on a some object.

Additional Inherited Members

4.11.1 Detailed Description

Class use to show graphic city and supporting all activities Inherits a [Window_Base](#).

4.11.2 Constructor & Destructor Documentation

4.11.2.1 Map_Windows() [1/2]

```
Map_Windows::Map_Windows (
    String title,
    int width,
    int height,
    String backroundPath,
    Champion * champ,
    int idUser )
```

Constructor Class [Map_Windows](#).

Parameters

<i>title</i>	- Title a windows
<i>width</i>	- Width window
<i>height</i>	- Height window
<i>champ</i>	-Object type Champion use to refresh value
<i>idUser</i>	- Id User with a database

4.11.2.2 Map_Windows() [2/2]

```
Map_Windows::Map_Windows ( )
```

Default constructor Class [Map_Windows](#).

4.11.2.3 ~Map_Windows()

```
Map_Windows::~~Map_Windows ( )
```

Destructor [Map_Windows](#).

4.11.3 Member Function Documentation

4.11.3.1 setPosition_figure()

```
void Map_Windows::setPosition_figure (
    int x,
    int y )
```

Function to set position character after back to a other activity.

4.11.3.2 Start()

```
void Map_Windows::Start ( )
```

Function start showing a graphic interface.

4.11.4 Member Data Documentation

4.11.4.1 clock

```
sf::Clock Map_Windows::clock
```

Object helping to create animations.

4.11.4.2 col_l_d_1

```
RectangleShape Map_Windows::col_l_d_1
```

Object use to collision on a some object.

4.11.4.3 col_l_d_2

```
RectangleShape Map_Windows::col_l_d_2
```

Object use to collision on a some object.

4.11.4.4 col_l_u_1

RectangleShape Map_Windows::col_l_u_1

Object use to collision on a some object.

4.11.4.5 col_l_u_2

RectangleShape Map_Windows::col_l_u_2

Object use to collision on a some object.

4.11.4.6 col_r_d_1

RectangleShape Map_Windows::col_r_d_1

Object use to collision on a some object.

4.11.4.7 col_r_d_2

RectangleShape Map_Windows::col_r_d_2

Object use to collision on a some object.

4.11.4.8 col_r_d_3

RectangleShape Map_Windows::col_r_d_3

Object use to collision on a some object.

4.11.4.9 col_r_u_1

RectangleShape Map_Windows::col_r_u_1

Object use to collision on a some object.

4.11.4.10 `col_r_u_2`

`RectangleShape Map_Windows::col_r_u_2`

Object use to collision on a some object.

4.11.4.11 `dbcontroler`

`DBcontroler Map_Windows::dbcontroler`

Object use to connect a database.

4.11.4.12 `rectSourceSprite`

`IntRect Map_Windows::rectSourceSprite`

Object helping to create animations.

4.11.4.13 `viewItem_figure`

`View_Item Map_Windows::viewItem_figure`

Object class `View_Item` use to show some graphics image.

4.11.4.14 `viewItem_key_arena`

`View_Item Map_Windows::viewItem_key_arena`

Object class `View_Item` use to show some graphics image.

4.11.4.15 `viewItem_key_rest`

`View_Item Map_Windows::viewItem_key_rest`

Object class `View_Item` use to show some graphics image.

4.11.4.16 viewItem_key_shop

`View_Item` `Map_Windows::viewItem_key_shop`

Object class `View_Item` use to show some graphics image.

4.11.4.17 window

`Window_Base` `Map_Windows::window`

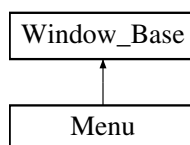
Object base class use to create graphic window interface.

4.12 Menu Class Reference

Class `Menu` represent window `Menu` inherit `Window_Base`.

```
#include <Menu.h>
```

Inheritance diagram for Menu:



Public Member Functions

- `Menu ()`
Defalut Constructor `Menu`.
- `Menu (String, int, int, String, int idUser)`
Constructor `Menu`.
- `~Menu ()`
Destructor `Menu`.
- `void Start ()`
Function start showing a graphic interface.

Public Attributes

- Sprite [backgroundTable](#)
Object Sprite use to show table.
- Texture [backgroundTab](#)
Object Texture use to load image.
- [Window_Base](#) [window](#)
Object a base window class use to create window.
- string [title](#)
Values use to keep name widow.
- int [width](#)
Width windows.
- int [height](#)
Height window.
- Texture [background](#)
Texture use to load background image.
- Sprite [backgroundImage](#)
Sprite use to showing a background.
- [Button](#) [buttonStart](#)
Object buttton to create and show button.
- [Button](#) [buttonExit](#)
Object buttton to create and show button.
- [DBcontroler](#) [dbcontroler](#)
Object class DBController use to connect a database.
- [MYSQL_RES](#) * [res](#)
Object use to hold a database get value.
- [MYSQL_ROW](#) [row](#)
Object hold a row witch data get with database.
- [Font](#) [arial](#)
Value Font to load a font.
- [Text](#) ** [Text_Data](#)
Dynamic value Text to show a chammpion in table .
- int [choose](#)
Value to hold a choose.
- int [size](#)
Value hold count a champion get a database.
- [RectangleShape](#) * [rectangle_choose](#)
Object use to check who row table is click.
- [RectangleShape](#) [add_b](#)
Object use to check if button add is click.
- [RectangleShape](#) [delete_b](#)
Object use to check if button delete is click.

Additional Inherited Members

4.12.1 Detailed Description

Class [Menu](#) represent window [Menu](#) inherit [Window_Base](#).

4.12.2 Constructor & Destructor Documentation

4.12.2.1 Menu() [1/2]

```
Menu::Menu ( )
```

Default Constructor [Menu](#).

4.12.2.2 Menu() [2/2]

```
Menu::Menu (
    String title,
    int width,
    int height,
    String backroundPath,
    int idUser )
```

Constructor [Menu](#).

Parameters

<i>String</i>	- Title window
<i>int</i>	- Wifth window
<i>int</i>	- Height window
<i>String</i>	- Path to backgrand image
<i>idUser</i>	- Value hold in User a database

4.12.2.3 ~Menu()

```
Menu::~~Menu ( )
```

Destructor [Menu](#).

4.12.3 Member Function Documentation

4.12.3.1 Start()

```
void Menu::Start ( )
```

Function start showing a graphic interface.

4.12.4 Member Data Documentation

4.12.4.1 add_b

`RectangleShape Menu::add_b`

Object use to check if button add is click.

4.12.4.2 arial

`Font Menu::arial`

Value Font to load a font.

4.12.4.3 background

`Texture Menu::background`

Texture use to load background image.

4.12.4.4 backgroundTab

`Texture Menu::backgroundTab`

Object Texture use to load image.

4.12.4.5 backgroundImage

`Sprite Menu::backgroundImage`

Sprite use to showing a background.

4.12.4.6 backgroundTable

`Sprite Menu::backgroundTable`

Object Sprite use to show table.

4.12.4.7 buttonExit

`Button Menu::buttonExit`

Object button to create and show button.

4.12.4.8 buttonStart

`Button Menu::buttonStart`

Object button to create and show button.

4.12.4.9 choose

`int Menu::choose`

Value to hold a choose.

4.12.4.10 dbcontroler

`DBcontroler Menu::dbcontroler`

Object class DBController use to connect a database.

4.12.4.11 delete_b

`RectangleShape Menu::delete_b`

Object use to check if button delete is click.

4.12.4.12 height

```
int Menu::height
```

Height window.

4.12.4.13 rectangle_choose

```
RectangleShape* Menu::rectangle_choose
```

Object use to check who row table is click.

4.12.4.14 res

```
MYSQL_RES* Menu::res
```

Object use to hold a database get value.

4.12.4.15 row

```
MYSQL_ROW Menu::row
```

Object hold a row witch data get with database.

4.12.4.16 size

```
int Menu::size
```

Value hold count a champion get a database.

4.12.4.17 Text_Data

```
Text** Menu::Text_Data
```

Dynamic value Text to show a chammpion in table .

4.12.4.18 title

```
string Menu::title
```

Values use to keep name widow.

4.12.4.19 width

```
int Menu::width
```

Width windows.

4.12.4.20 window

```
Window_Base Menu::window
```

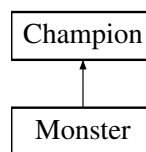
Object a base window class use to create window.

4.13 Monster Class Reference

Class [Monster](#) Inherits a champion.

```
#include <Monster.h>
```

Inheritance diagram for Monster:

**Public Member Functions**

- [Monster](#) (const std::string n, unsigned int l, int he, int st, int de, int ma, int def, int lu)
Constructor [Monster](#) cause a champion constructor.
- [Monster](#) ([Champion](#) *const champ)
Constructor [Monster](#) copying constructor.
- virtual int [attack](#) () override
Virtual function attack.
- virtual int [block](#) () override
Virtual function block.
- virtual [Monster](#) * [clone](#) () const
Virtual function clone to clone object.

Additional Inherited Members

4.13.1 Detailed Description

Class [Monster](#) Inherits a champion.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 [Monster\(\)](#) [1/2]

```
Monster::Monster (
    const std::string n,
    unsigned int l,
    int he,
    int st,
    int de,
    int ma,
    int def,
    int lu ) [inline]
```

Constructor [Monster](#) cause a champion constructor.

Parameters

<i>n</i>	- Name champion
<i>l</i>	- Level champion
<i>he</i>	- Health champion
<i>st</i>	- Strength champion
<i>de</i>	- Dexterity Champion
<i>ma</i>	- Magic champion
<i>def</i>	- Defence champion
<i>lu</i>	- Lucky champion

4.13.2.2 [Monster\(\)](#) [2/2]

```
Monster::Monster (
    Champion *const champ ) [inline]
```

Constructor [Monster](#) copying constructor.

Parameters

<i>champ</i>	- Pointer object const champ
--------------	------------------------------

4.13.3 Member Function Documentation

4.13.3.1 attack()

```
int Monster::attack ( ) [override], [virtual]
```

Virtual function attack.

Returns

value attack

Reimplemented from [Champion](#).

4.13.3.2 block()

```
int Monster::block ( ) [override], [virtual]
```

Virtual function block.

Returns

value block

Reimplemented from [Champion](#).

4.13.3.3 clone()

```
virtual Monster* Monster::clone ( ) const [inline], [virtual]
```

Virtual function clone to clone object.

Returns

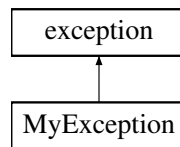
Pointer object [Archer](#)

Implements [Champion](#).

4.14 MyException Struct Reference

Struct to service exception.

Inheritance diagram for MyException:



Public Member Functions

- [MyException](#) (string err)
Constructor Struct [MyException](#)

Parameters

err	- Message error
-----	-----------------

- const string [what](#) () throw ()
Show message error.

Public Attributes

- string [error](#)
Text error message.

4.14.1 Detailed Description

Struct to service exception.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 MyException()

```
MyException::MyException (
    string err ) [inline]
```

Constructor Struct [MyException](#)

Parameters

err	- Message error
-----	-----------------

4.14.3 Member Function Documentation

4.14.3.1 what()

```
const string MyException::what ( ) throw ( )    [inline]
```

Show message error.

Returns

message error

4.14.4 Member Data Documentation

4.14.4.1 error

```
string MyException::error
```

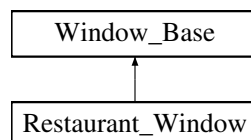
Text error message.

4.15 Restaurant_Window Class Reference

Class restaurant to representation graphic Window object restaurant inherit [Window_Base](#).

```
#include <Restaurant_Window.h>
```

Inheritance diagram for Restaurant_Window:



Public Member Functions

- [Restaurant_Window](#) ([Champion](#) *champ, int idUse)
Constructor [Restaurant_Window](#).
- [~Restaurant_Window](#) ()
Destructor.
- void [Start](#) ()
Function start showing graphic interface.

Public Attributes

- Font [arial](#)
Variable aria lto load a font.
- Text ** [Text_Data](#)
Value dyamic 2D array Text use to hold a text.
- Sprite [backgroundTable](#)
Variable Sprite use to show a table.
- Texture [backgroundTab](#)
Variable Texture use to load a image.
- DBcontroler [dbcontroler](#)
Object use to connect a database.
- MYSQL_ROW [row](#)
Object temporary use to hold a row with database value.
- MYSQL_RES * [res](#)
Variable use to hold data get a database.
- RectangleShape * [rectangle_choose](#)
Dynamic array use to check who row is click.
- int [choose](#)
Value hold a choose .
- int [size](#)
Variable hold a cout data get with database.
- Button [buttonStart](#)
Object class [Button](#).

Additional Inherited Members

4.15.1 Detailed Description

Class restaurant to representation graphic Window object restaurant inherit [Window_Base](#).

4.15.2 Constructor & Destructor Documentation

4.15.2.1 Restaurant_Window()

```
Restaurant_Window::Restaurant_Window (
    Champion * champ,
    int idUse )
```

Constructor [Restaurant_Window](#).

Parameters

<i>champ</i>	- Pointet object Champion
<i>idUse</i>	- Value use to hold id user

4.15.2.2 ~Restaurant_Window()

```
Restaurant_Window::~~Restaurant_Window ( )
```

Destructor.

4.15.3 Member Function Documentation

4.15.3.1 Start()

```
void Restaurant_Window::Start ( )
```

Function start showing graphic interface.

4.15.4 Member Data Documentation

4.15.4.1 arial

```
Font Restaurant_Window::arial
```

Variable aria lto load a font.

4.15.4.2 backgroundTab

```
Texture Restaurant_Window::backgroundTab
```

Variable Texture use to load a image.

4.15.4.3 backgroundTable

```
Sprite Restaurant_Window::backgroundTable
```

Variable Sprite use to show a table.

4.15.4.4 buttonStart

`Button Restaurant_Window::buttonStart`

Object class `Button`.

4.15.4.5 choose

`int Restaurant_Window::choose`

Value hold a choose .

4.15.4.6 dbcontroler

`DBcontroler Restaurant_Window::dbcontroler`

Object use to connect a database.

4.15.4.7 rectangle_choose

`RectangleShape* Restaurant_Window::rectangle_choose`

Dynamic array use to check who row is click.

4.15.4.8 res

`MYSQL_RES* Restaurant_Window::res`

Variable use to hold data get a database.

4.15.4.9 row

`MYSQL_ROW Restaurant_Window::row`

Object temporary use to hold a row with database value.

4.15.4.10 size

```
int Restaurant_Window::size
```

Variable hold a cout data get with database.

4.15.4.11 Text_Data

```
Text** Restaurant_Window::Text_Data
```

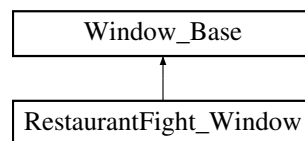
Value dyamic 2D array Text use to hold a text.

4.16 RestaurantFight_Window Class Reference

Class to representation graphic Window object [Fight](#) between players.

```
#include <RestaurantFight_Window.h>
```

Inheritance diagram for RestaurantFight_Window:



Public Member Functions

- [RestaurantFight_Window](#) ([Champion](#) *champ, [Champion](#) *opon, int id)

Constructor [RestaurantFight_Window](#)

Parameters

champ	- Player's character
opon	- Selected opponent
id	- User's id

- [~RestaurantFight_Window](#) ()

Destructor Class [RestaurantFight_Window](#).

- void [Start](#) ()

Function start showing [RestaurantFight_Window](#) and fight between players.

Additional Inherited Members

4.16.1 Detailed Description

Class to representation graphic Window object [Fight](#) between players.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 RestaurantFight_Window()

```
RestaurantFight_Window::RestaurantFight_Window (
    Champion * champ,
    Champion * opon,
    int id )
```

Constructor [RestaurantFight_Window](#)

Parameters

<i>champ</i>	- Player's character
<i>opon</i>	- Selected opponent
<i>id</i>	- User's id

4.16.2.2 ~RestaurantFight_Window()

```
RestaurantFight_Window::~~RestaurantFight_Window ( )
```

Destructor Class [RestaurantFight_Window](#).

4.16.3 Member Function Documentation

4.16.3.1 Start()

```
void RestaurantFight_Window::Start ( )
```

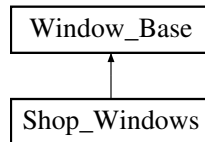
Function start showing [RestaurantFight_Window](#) and fight between players.

4.17 Shop_Windows Class Reference

Class shop to representation graphic Window object shop.

```
#include <Shop_Windows.h>
```

Inheritance diagram for Shop_Windows:



Public Member Functions

- [Shop_Windows](#) ([Champion](#) *champ, int idUse)
Constructor Class [Shop_Windows](#).
- [~Shop_Windows](#) ()
Destructor [Shop_Windows](#).
- void [Start](#) ()
Function start showing window.
- bool [moneycheck](#) (int money)
Function checks if the character has enough cash.

Additional Inherited Members

4.17.1 Detailed Description

Class shop to representation graphic Window object shop.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 Shop_Windows()

```
Shop_Windows::Shop_Windows (
    Champion * champ,
    int idUse )
```

Constructor Class [Shop_Windows](#).

Parameters

<i>champ</i>	- character witch enter to window
<i>idUse</i>	- user's id

4.17.2.2 ~Shop_Windows()

```
Shop_Windows::~~Shop_Windows ( )
```

Destructor [Shop_Windows](#).

4.17.3 Member Function Documentation

4.17.3.1 moneycheck()

```
bool Shop_Windows::moneycheck (
    int money )
```

Function checks if the character has enough cash.

Parameters

<i>money</i>	- the money held by character
--------------	-------------------------------

Returns

true or false

4.17.3.2 Start()

```
void Shop_Windows::Start ( )
```

Function start showing window.

4.18 View_Item Class Reference

Class use to representation some graphics.

```
#include <View_Item.h>
```

Public Member Functions

- [View_Item](#) ()
Default constructor Class [View_Item](#).
- [View_Item](#) (String Path, int x, int y)
Constructor Class [View_Item](#).
- [~View_Item](#) ()
Default constructor Class [View_Item](#).
- Sprite & [getImage](#) ()
Function return object to show graphic.
- void [set_Image](#) (String Path, int x, int y)
Function to set new image.

Public Attributes

- Sprite [Image](#)
Image as Sprite.
- Texture [item](#)
Image as Texture.

4.18.1 Detailed Description

Class use to representation some graphics.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 View_Item() [1/2]

```
View_Item::View_Item ( )
```

Default constructor Class [View_Item](#).

4.18.2.2 View_Item() [2/2]

```
View_Item::View_Item (
    String Path,
    int x,
    int y )
```

Constructor Class [View_Item](#).

Parameters

<i>Path</i>	- Path to image
<i>x</i>	- Width position
<i>y</i>	- Height position

4.18.2.3 ~View_Item()

```
View_Item::~View_Item ( )
```

Default constructor Class [View_Item](#).

4.18.3 Member Function Documentation**4.18.3.1 getImage()**

```
Sprite & View_Item::getImage ( )
```

Function return object to show graphic.

4.18.3.2 set_Image()

```
void View_Item::set_Image (
    String Path,
    int x,
    int y )
```

Function to set new image.

Parameters

<i>Path</i>	- Path to new image
<i>x</i>	- Width position
<i>y</i>	- Height position

4.18.4 Member Data Documentation

4.18.4.1 Image

```
Sprite View_Item::Image
```

Image as Sprite.

4.18.4.2 item

```
Texture View_Item::item
```

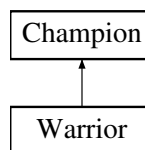
Image as Texture.

4.19 Warrior Class Reference

Class [Warrior](#) Inherits a champion.

```
#include <Warrior.h>
```

Inheritance diagram for Warrior:



Public Member Functions

- [Warrior](#) (const unsigned id, const std::string n, unsigned int l, unsigned int e, double m, int he, int st, int de, int ma, int def, int lu)
Constructor [Warrior](#) cause a champion constructor.
- virtual [Warrior](#) * [clone](#) () const
Virtual function clone to clone object.
- virtual int [attack](#) () override
Virtual function attack.
- virtual int [block](#) () override
Virtual function block.

Additional Inherited Members

4.19.1 Detailed Description

Class [Warrior](#) Inherits a champion.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 Warrior()

```
Warrior::Warrior (
    const unsigned id,
    const std::string n,
    unsigned int l,
    unsigned int e,
    double m,
    int he,
    int st,
    int de,
    int ma,
    int def,
    int lu ) [inline]
```

Constructor [Warrior](#) cause a champion constructor.

Parameters

<i>id</i>	- Id champion in database
<i>n</i>	- Name champion
<i>l</i>	- Level champion
<i>e</i>	- Experience champion
<i>m</i>	- Money champion
<i>he</i>	- Health champion
<i>st</i>	- Strength champion
<i>de</i>	- Dexerity Champion
<i>ma</i>	- Magic champion
<i>def</i>	- Defence champion
<i>lu</i>	- Lucky champion

4.19.3 Member Function Documentation

4.19.3.1 attack()

```
int Warrior::attack ( ) [override], [virtual]
```

Virtual function attack.

Returns

value attack

Reimplemented from [Champion](#).

4.19.3.2 block()

```
int Warrior::block ( ) [override], [virtual]
```

Virtual function block.

Returns

value block

Reimplemented from [Champion](#).

4.19.3.3 clone()

```
virtual Warrior\* Warrior::clone ( ) const [inline], [virtual]
```

Virtual function clone to clone object.

Returns

Pointer object [Warrior](#)

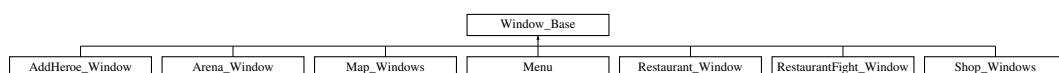
Implements [Champion](#).

4.20 Window_Base Class Reference

Base class use to make graphic interface.

```
#include <Window_Base.h>
```

Inheritance diagram for Window_Base:



Public Member Functions

- [Window_Base](#) (String [title](#), int [width](#), int [height](#), String [backgroundPath](#))
Constructor Class [Window_Base](#).
- [Window_Base](#) ()
Default Constructor Class [Window_Base](#).
- [~Window_Base](#) ()
Destructor [Window_Base](#).
- [RenderWindow](#) & [getWindows](#) ()
Function to get make window.

Public Attributes

- Sprite [backgroundImage](#)
Basic background image.

Protected Attributes

- Texture [background](#)
- RenderWindow [window](#)
- string [title](#)
- int [width](#)
- int [height](#)

4.20.1 Detailed Description

Base class use to make graphic interface.

4.20.2 Constructor & Destructor Documentation

4.20.2.1 Window_Base() [1/2]

```
Window_Base::Window_Base (
    String title,
    int width,
    int height,
    String backgroundColor )
```

Constructor Class [Window_Base](#).

Parameters

<i>title</i>	- Title of window
<i>width</i>	- Width window
<i>height</i>	- Height window
<i>backgroundColor</i>	- Path to basic background image

4.20.2.2 Window_Base() [2/2]

```
Window_Base::Window_Base ( )
```

Default Constructor Class [Window_Base](#).

4.20.2.3 ~Window_Base()

```
Window_Base::~~Window_Base ( )
```

Destructor [Window_Base](#).

4.20.3 Member Function Documentation

4.20.3.1 getWindows()

```
RenderWindow & Window_Base::getWindows ( )
```

Function to get make window.

4.20.4 Member Data Documentation

4.20.4.1 background

```
Texture Window_Base::background [protected]
```

4.20.4.2 backgroundImage

```
Sprite Window_Base::backgroundImage
```

Basic background image.

4.20.4.3 height

```
int Window_Base::height [protected]
```

4.20.4.4 title

```
string Window_Base::title [protected]
```

4.20.4.5 width

```
int Window_Base::width [protected]
```

4.20.4.6 window

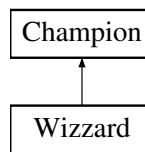
```
RenderWindow Window_Base::window [protected]
```

4.21 Wizzard Class Reference

Class [Wizzard](#) Inherits a champion.

```
#include <Wizzard.h>
```

Inheritance diagram for Wizzard:



Public Member Functions

- [Wizzard](#) (const unsigned id, const std::string n, unsigned int l, unsigned int e, double m, int he, int st, int de, int ma, int def, int lu)
Constructor [Archer](#) cause a champion constructor.
- virtual int [attack](#) () override
Virtual function attack.
- virtual [Wizzard](#) * [clone](#) () const
Virtual function clone to clone object.

Additional Inherited Members

4.21.1 Detailed Description

Class [Wizzard](#) Inherits a champion.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 Wizzard()

```
Wizzard::Wizzard (
    const unsigned id,
    const std::string n,
    unsigned int l,
    unsigned int e,
    double m,
    int he,
    int st,
    int de,
    int ma,
    int def,
    int lu ) [inline]
```

Constructor [Archer](#) cause a champion constructor.

Parameters

<i>id</i>	- Id champion in database
<i>n</i>	- Name champion
<i>l</i>	- Level champion
<i>e</i>	- Experience champion
<i>m</i>	- Money champion
<i>he</i>	- Health champion
<i>st</i>	- Strength champion
<i>de</i>	- Dexerity Champion
<i>ma</i>	- Magic champion
<i>def</i>	- Defence champion
<i>lu</i>	- Lucky champion

4.21.3 Member Function Documentation

4.21.3.1 attack()

```
int Wizzard::attack ( ) [override], [virtual]
```

Virtual function attack.

Returns

value attack

Reimplemented from [Champion](#).

4.21.3.2 clone()

```
virtual Wizzard\* Wizzard::clone ( ) const [inline], [virtual]
```

Virtual function clone to clone object.

Returns

Pointer object [Archer](#)

Implements [Champion](#).

Chapter 5

File Documentation

5.1 AddHeroe_Window.cpp File Reference

```
#include "AddHeroe_Window.h"
#include "Menu.h"
#include "DBcontroler.h"
#include "Champion.h"
#include "Archer.h"
#include "Warrior.h"
#include "Wizzard.h"
```

5.2 AddHeroe_Window.h File Reference

```
#include "Window_Base.h"
#include <SFML/Graphics.hpp>
#include <string>
#include "DBcontroler.h"
#include "Login_Window.h"
#include <iostream>
```

Classes

- class [AddHeroe_Window](#)
Class window support add character to user.

5.3 Archer.cpp File Reference

```
#include "Archer.h"
#include <random>
```

5.4 Archer.h File Reference

```
#include "Champion.h"
```

Classes

- class [Archer](#)
Class [Archer](#) Inherits a champion.

5.5 Arena_Window.cpp File Reference

```
#include "Arena_Window.h"  
#include "Map_Windows.h"  
#include "Fight.h"  
#include "DBcontroler.h"  
#include <thread>  
#include "Monster.h"  
#include <random>
```

Functions

- void [setText](#) (sf::Text *t1, string s1, sf::Text *t2, string s2)

5.5.1 Function Documentation

5.5.1.1 setText()

```
void setText (  
    sf::Text * t1,  
    string s1,  
    sf::Text * t2,  
    string s2 )
```

5.6 Arena_Window.h File Reference

```
#include "Window_Base.h"  
#include "View_Item.h"  
#include "Champion.h"
```

Classes

- class [Arena_Window](#)

Class arena to representation graphic Window object Arena.

5.7 Button.cpp File Reference

```
#include "Button.h"
#include <iostream>
```

5.8 Button.h File Reference

```
#include <string>
#include <SFML/Graphics.hpp>
```

Classes

- class [Button](#)

Class to creat object button.

5.9 Champion.cpp File Reference

```
#include "Champion.h"
```

Functions

- `std::ostream & operator<< (std::ostream &out, const Champion &p)`

5.9.1 Function Documentation

5.9.1.1 `operator<<()`

```
std::ostream& operator<< (
    std::ostream & out,
    const Champion & p )
```

5.10 Champion.h File Reference

```
#include <iostream>
```

Classes

- class [Champion](#)
Class [Champion](#) use to create character.

5.11 Create_Window.cpp File Reference

```
#include "Create_Window.h"  
#include "Login_Window.h"  
#include "Menu.h"  
#include <iostream>
```

5.12 Create_Window.h File Reference

```
#include "Menu.h"  
#include <iostream>
```

Classes

- class [Create_Window](#)
Class window support create user.

5.13 DBcontroler.cpp File Reference

```
#include "DBcontroler.h"  
#include <string>  
#include "Warrior.h"  
#include "Archer.h"  
#include "Wizzard.h"  
#include "ThrowLoad.cpp"
```

5.14 DBcontroler.h File Reference

```
#include <mysql.h>  
#include <iostream>  
#include "Champion.h"
```

Classes

- class [DBcontroler](#)

Class responsible for database operation.

5.15 ErrorShow_Window.cpp File Reference

```
#include "ErrorShow_Window.h"
```

5.16 ErrorShow_Window.h File Reference

```
#include "Window_Base.h"
```

Classes

- class [ErrorShow_Window](#)

Classe represent window to show a error messge.

5.17 Fight.cpp File Reference

```
#include "Fight.h"  
#include <Windows.h>
```

Functions

- `std::ostream & operator<< (std::ostream &out, const Fight &p)`

5.17.1 Function Documentation

5.17.1.1 `operator<<()`

```
std::ostream& operator<< (  
    std::ostream & out,  
    const Fight & p )
```

5.18 Fight.h File Reference

```
#include <iostream>
#include "Champion.h"
#include "string"
```

Classes

- class [Fight](#)

Class support fighting between player vs monster or player vs player.

5.19 Login_Window.cpp File Reference

```
#include "Login_Window.h"
#include "Menu.h"
#include "Create_Window.h"
#include <iostream>
#include <SFML/Graphics.hpp>
#include "DBcontroler.h"
#include "Champion.h"
#include <mysql.h>
#include "ThrowLoad.cpp"
```

5.20 Login_Window.h File Reference

```
#include <SFML/Graphics.hpp>
#include "Window_Base.h"
#include "View_Item.h"
#include "DBcontroler.h"
```

Classes

- class [Login_Window](#)

Class window support login user.

5.21 Map_Windows.cpp File Reference

```
#include "Map_Windows.h"
#include <iostream>
#include "Menu.h"
#include "Arena_Window.h"
#include "Restaurant_Window.h"
#include "Shop_Windows.h"
```

5.22 Map_Windows.h File Reference

```
#include <SFML/Graphics.hpp>
#include "Window_Base.h"
#include "View_Item.h"
#include "Champion.h"
#include "DBcontroler.h"
```

Classes

- class [Map_Windows](#)

Class use to show graphic city and supporting all activities Inherits a [Window_Base](#).

5.23 Menu.cpp File Reference

```
#include "Menu.h"
#include "Map_Windows.h"
#include <SFML/Graphics.hpp>
#include "ThrowLoad.cpp"
#include "DBcontroler.h"
#include "AddHeroe_Window.h"
```

5.24 Menu.h File Reference

```
#include "Button.h"
#include "Window_Base.h"
#include <SFML/Graphics.hpp>
#include <string>
#include "Champion.h"
#include "DBcontroler.h"
```

Classes

- class [Menu](#)

Class [Menu](#) represent window [Menu](#) inherit [Window_Base](#).

5.25 Monster.cpp File Reference

```
#include "Monster.h"
#include <random>
```

5.26 Monster.h File Reference

```
#include "Champion.h"
```

Classes

- class [Monster](#)

Class [Monster](#) Inherits a champion.

5.27 Restaurant_Window.cpp File Reference

```
#include "Restaurant_Window.h"  
#include "Map_Windows.h"  
#include "ThrowLoad.cpp"  
#include "DBcontroler.h"  
#include "RestaurantFight_Window.h"
```

5.28 Restaurant_Window.h File Reference

```
#include "Window_Base.h"  
#include "Champion.h"  
#include "View_Item.h"  
#include "DBcontroler.h"  
#include "Button.h"
```

Classes

- class [Restaurant_Window](#)

Class restaurant to representation graphic Window object restaurant inherit [Window_Base](#).

5.29 RestaurantFight_Window.h File Reference

```
#include "Window_Base.h"  
#include "Champion.h"  
#include "View_Item.h"  
#include "Fight.h"  
#include "DBcontroler.h"
```

Classes

- class [RestaurantFight_Window](#)

Class to representation graphic Window object [Fight](#) between players.

5.30 RestaurantFight_Window.cpp File Reference

```
#include "RestaurantFight_Window.h"
#include "Map_Windows.h"
#include "Fight.h"
#include "DBcontroler.h"
#include <thread>
#include "Restaurant_Window.h"
```

Functions

- void [setText1](#) (sf::Text *t1, string s1, sf::Text *t2, string s2)

5.30.1 Function Documentation

5.30.1.1 [setText1\(\)](#)

```
void setText1 (
    sf::Text * t1,
    string s1,
    sf::Text * t2,
    string s2 )
```

5.31 Shop_Windows.cpp File Reference

```
#include "Shop_Windows.h"
#include "Champion.h"
#include <string>
```

5.32 Shop_Windows.h File Reference

```
#include "Window_Base.h"
#include "Map_Windows.h"
#include "Champion.h"
```

Classes

- class [Shop_Windows](#)
Class shop to representation graphic Window object shop.

5.33 Source.cpp File Reference

```
#include <SFML/Graphics.hpp>
#include "Menu.h"
#include "Login_Window.h"
#include "ThrowLoad.cpp"
#include <iostream>
#include "Window_Base.h"
#include "ErrorShow_Window.h"
```

Functions

- int [main](#) ()

5.33.1 Function Documentation

5.33.1.1 main()

```
int main ( )
```

5.34 ThrowLoad.cpp File Reference

```
#include <iostream>
#include <string>
```

Classes

- struct [MyException](#)
Struct to service exception.

5.35 View_Item.cpp File Reference

```
#include "View_Item.h"
#include <SFML/Graphics.hpp>
#include "ThrowLoad.cpp"
```

5.36 View_Item.h File Reference

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

Classes

- class [View_Item](#)

Class use to representation some graphics.

5.37 Warrior.cpp File Reference

```
#include "Warrior.h"
#include <random>
```

5.38 Warrior.h File Reference

```
#include "Champion.h"
```

Classes

- class [Warrior](#)

Class [Warrior](#) Inherits a champion.

5.39 Window_Base.cpp File Reference

```
#include "Window_Base.h"
#include "ThrowLoad.cpp"
#include <SFML/Graphics.hpp>
#include <string>
```

5.40 Window_Base.h File Reference

```
#include <SFML/Graphics.hpp>
#include <string>
```

Classes

- class [Window_Base](#)

Base class use to make graphic interface.

5.41 Wizzard.cpp File Reference

```
#include "Wizzard.h"  
#include <random>
```

5.42 Wizzard.h File Reference

```
#include "Champion.h"
```

Classes

- class [Wizzard](#)

Class [Wizzard](#) Inherits a champion.

Index

- ~AddHeroe_Window
 - AddHeroe_Window, 8
- ~Arena_Window
 - Arena_Window, 14
- ~Button
 - Button, 18
- ~Create_Window
 - Create_Window, 35
- ~DBcontroler
 - DBcontroler, 39
- ~ErrorShow_Window
 - ErrorShow_Window, 44
- ~Fight
 - Fight, 46
- ~Login_Window
 - Login_Window, 48
- ~Map_Windows
 - Map_Windows, 53
- ~Menu
 - Menu, 59
- ~RestaurantFight_Window
 - RestaurantFight_Window, 72
- ~Restaurant_Window
 - Restaurant_Window, 69
- ~Shop_Windows
 - Shop_Windows, 74
- ~View_Item
 - View_Item, 76
- ~Window_Base
 - Window_Base, 80
- add_b
 - Menu, 60
- addChampion
 - DBcontroler, 39
- addDefence
 - Champion, 24
- addDexterity
 - Champion, 25
- addExperience
 - Champion, 25
- addHealth
 - Champion, 25
- AddHeroe_Window, 7
 - ~AddHeroe_Window, 8
 - AddHeroe_Window, 8
 - arial, 9
 - choose, 9
 - dbcontroler, 9
 - flaga, 9
 - Logo_Button, 9
 - Name, 10
 - name, 10
 - profession, 10
 - rectangle_add, 10
 - rectangle_choose, 10
 - rectangle_name, 10
 - Start, 9
 - window, 11
- AddHeroe_Window.cpp, 85
- AddHeroe_Window.h, 85
- addHeroes
 - DBcontroler, 39
- addLuck
 - Champion, 25
- addMagic
 - Champion, 27
- addMoney
 - Champion, 27
- addStrength
 - Champion, 27
- addUser
 - DBcontroler, 39
- Archer, 11
 - Archer, 12
 - attack, 12
 - block, 12
 - clone, 13
- Archer.cpp, 85
- Archer.h, 86
- Arena_Window, 13
 - ~Arena_Window, 14
 - Arena_Window, 14
 - getClickHeal, 15
 - randMonster, 15
 - Start, 15
- Arena_Window.cpp, 86
 - setText, 86
- Arena_Window.h, 86
- arial
 - AddHeroe_Window, 9
 - Create_Window, 35
 - Login_Window, 49
 - Menu, 60
 - Restaurant_Window, 69
- attack
 - Archer, 12
 - Champion, 27
 - Monster, 65

- Warrior, 78
- Wizzard, 83
- background
 - Menu, 60
 - Window_Base, 81
- backgroundTab
 - Menu, 60
 - Restaurant_Window, 69
- backgroundImage
 - Menu, 60
 - Window_Base, 81
- backgroundTable
 - Menu, 60
 - Restaurant_Window, 69
- block
 - Archer, 12
 - Champion, 28
 - Monster, 65
 - Warrior, 78
- Button, 15
 - ~Button, 18
 - Button, 18
 - button, 20
 - font, 20
 - getButton, 18
 - getText, 19
 - setButton, 19
 - setText, 19
 - text, 20
 - write, 20
- button
 - Button, 20
- Button.cpp, 87
- Button.h, 87
- buttonExit
 - Menu, 61
- buttonStart
 - Menu, 61
 - Restaurant_Window, 69
- Champion, 20
 - addDefence, 24
 - addDexterity, 25
 - addExperience, 25
 - addHealth, 25
 - addLuck, 25
 - addMagic, 27
 - addMoney, 27
 - addStrength, 27
 - attack, 27
 - block, 28
 - Champion, 22
 - clone, 28
 - defence, 32
 - dexterity, 32
 - experience, 32
 - getDefence, 28
 - getDexterity, 28
 - getExperience, 28
 - getHealth, 29
 - getId, 29
 - getLevel, 29
 - getLuck, 29
 - getMagic, 30
 - getMoney, 30
 - getName, 30
 - getStrength, 30
 - heal, 31
 - health, 32
 - id_champion, 32
 - level, 32
 - luck, 33
 - magic, 33
 - money, 33
 - name, 33
 - operator<<, 32
 - setHealth, 31
 - strength, 33
 - takeMoney, 31
- Champion.cpp, 87
 - operator<<, 87
- Champion.h, 88
- checkUser
 - DBcontroler, 40
- choose
 - AddHeros_Window, 9
 - Menu, 61
 - Restaurant_Window, 70
- clock
 - Map_Windows, 54
- clone
 - Archer, 13
 - Champion, 28
 - Monster, 65
 - Warrior, 79
 - Wizzard, 83
- col_l_d_1
 - Map_Windows, 54
- col_l_d_2
 - Map_Windows, 54
- col_l_u_1
 - Map_Windows, 54
- col_l_u_2
 - Map_Windows, 55
- col_r_d_1
 - Map_Windows, 55
- col_r_d_2
 - Map_Windows, 55
- col_r_d_3
 - Map_Windows, 55
- col_r_u_1
 - Map_Windows, 55
- col_r_u_2
 - Map_Windows, 55
- Create_Window, 33
 - ~Create_Window, 35

- arial, 35
- Create_Window, 34
- dbcontroller, 35
- flaga, 35
- Login, 36
- login, 36
- Logo_Button, 36
- Password, 36
- password, 36
- rectangle_create, 36
- rectangle_pass, 37
- rectangle_user, 37
- Start, 35
- window, 37
- Create_Window.cpp, 88
- Create_Window.h, 88
- DBcontroller, 37
 - ~DBcontroller, 39
 - addChampion, 39
 - addHeroes, 39
 - addUser, 39
 - checkUser, 40
 - DBcontroller, 38
 - deleteHeroe, 41
 - deleteHeroes, 41
 - getChampion, 41
 - getChampions, 42
 - getProf, 42
 - getUser, 42
 - randomChampion, 43
 - updateChampion, 43
- dbcontroller
 - AddHeroe_Window, 9
 - Create_Window, 35
 - Login_Window, 49
 - Map_Windows, 56
 - Menu, 61
 - Restaurant_Window, 70
- DBcontroller.cpp, 88
- DBcontroller.h, 88
- defence
 - Champion, 32
- delete_b
 - Menu, 61
- deleteHeroe
 - DBcontroller, 41
- deleteHeroes
 - DBcontroller, 41
- dexterity
 - Champion, 32
- error
 - MyException, 67
- ErrorShow_Window, 43
 - ~ErrorShow_Window, 44
 - ErrorShow_Window, 44
 - Start, 44
- ErrorShow_Window.cpp, 89
- ErrorShow_Window.h, 89
- experience
 - Champion, 32
- Fight, 44
 - ~Fight, 46
 - Fight, 45
 - getChamp1, 46
 - getChamp2, 46
 - heal, 46
 - operator<<, 47
 - startFight, 47
- Fight.cpp, 89
 - operator<<, 89
- Fight.h, 90
- flaga
 - AddHeroe_Window, 9
 - Create_Window, 35
 - Login_Window, 49
- font
 - Button, 20
- getButton
 - Button, 18
- getChamp1
 - Fight, 46
- getChamp2
 - Fight, 46
- getChampion
 - DBcontroller, 41
- getChampions
 - DBcontroller, 42
- getClickHeal
 - Arena_Window, 15
- getDefence
 - Champion, 28
- getDexterity
 - Champion, 28
- getExperince
 - Champion, 28
- getHealth
 - Champion, 29
- getId
 - Champion, 29
- getImage
 - View_Item, 76
- getLevel
 - Champion, 29
- getLuck
 - Champion, 29
- getMagic
 - Champion, 30
- getMoney
 - Champion, 30
- getName
 - Champion, 30
- getProf
 - DBcontroller, 42
- getStrength

- Champion, 30
- getText
 - Button, 19
- getUser
 - DBcontroler, 42
- getWindows
 - Window_Base, 81
- heal
 - Champion, 31
 - Fight, 46
- health
 - Champion, 32
- height
 - Menu, 61
 - Window_Base, 81
- id_champion
 - Champion, 32
- Image
 - View_Item, 76
- item
 - View_Item, 77
- level
 - Champion, 32
- Login
 - Create_Window, 36
 - Login_Window, 49
- login
 - Create_Window, 36
 - Login_Window, 50
- Login_Window, 47
 - ~Login_Window, 48
 - arial, 49
 - dbcontroler, 49
 - flaga, 49
 - Login, 49
 - login, 50
 - Login_Window, 48
 - Logo_Button, 50
 - Password, 50
 - password, 50
 - rectangle_create, 50
 - rectangle_log, 50
 - rectangle_pass, 51
 - rectangle_user, 51
 - Start, 49
 - window, 51
- Login_Window.cpp, 90
- Login_Window.h, 90
- Logo_Button
 - AddHroe_Window, 9
 - Create_Window, 36
 - Login_Window, 50
- luck
 - Champion, 33
- magic
 - Champion, 33
- main
 - Source.cpp, 94
- Map_Windows, 51
 - ~Map_Windows, 53
 - clock, 54
 - col_l_d_1, 54
 - col_l_d_2, 54
 - col_l_u_1, 54
 - col_l_u_2, 55
 - col_r_d_1, 55
 - col_r_d_2, 55
 - col_r_d_3, 55
 - col_r_u_1, 55
 - col_r_u_2, 55
 - dbcontroler, 56
 - Map_Windows, 53
 - rectSourceSprite, 56
 - setPosition_figure, 54
 - Start, 54
 - viewItem_figure, 56
 - viewItem_key_arena, 56
 - viewItem_key_rest, 56
 - viewItem_key_shop, 56
 - window, 57
- Map_Windows.cpp, 90
- Map_Windows.h, 91
- Menu, 57
 - ~Menu, 59
 - add_b, 60
 - arial, 60
 - background, 60
 - backgroundTab, 60
 - backgroundImage, 60
 - backgroundTable, 60
 - buttonExit, 61
 - buttonStart, 61
 - choose, 61
 - dbcontroler, 61
 - delete_b, 61
 - height, 61
 - Menu, 59
 - rectangle_choose, 62
 - res, 62
 - row, 62
 - size, 62
 - Start, 59
 - Text_Data, 62
 - title, 62
 - width, 63
 - window, 63
- Menu.cpp, 91
- Menu.h, 91
- money
 - Champion, 33
- moneycheck
 - Shop_Windows, 74
- Monster, 63

- attack, 65
 - block, 65
 - clone, 65
 - Monster, 64
- Monster.cpp, 91
- Monster.h, 92
- MyException, 66
 - error, 67
 - MyException, 66
 - what, 67
- Name
 - AddHeroe_Window, 10
- name
 - AddHeroe_Window, 10
 - Champion, 33
- operator<<
 - Champion, 32
 - Champion.cpp, 87
 - Fight, 47
 - Fight.cpp, 89
- Password
 - Create_Window, 36
 - Login_Window, 50
- password
 - Create_Window, 36
 - Login_Window, 50
- profession
 - AddHeroe_Window, 10
- randMonster
 - Arena_Window, 15
- randomChampion
 - DBcontroler, 43
- rectangle_add
 - AddHeroe_Window, 10
- rectangle_choose
 - AddHeroe_Window, 10
 - Menu, 62
 - Restaurant_Window, 70
- rectangle_create
 - Create_Window, 36
 - Login_Window, 50
- rectangle_log
 - Login_Window, 50
- rectangle_name
 - AddHeroe_Window, 10
- rectangle_pass
 - Create_Window, 37
 - Login_Window, 51
- rectangle_user
 - Create_Window, 37
 - Login_Window, 51
- rectSourceSprite
 - Map_Windows, 56
- res
 - Menu, 62
 - Restaurant_Window, 70
- Restaurant_Window, 67
 - ~Restaurant_Window, 69
 - arial, 69
 - backgroundTab, 69
 - backgroundTable, 69
 - buttonStart, 69
 - choose, 70
 - dbcontroler, 70
 - rectangle_choose, 70
 - res, 70
 - Restaurant_Window, 68
 - row, 70
 - size, 70
 - Start, 69
 - Text_Data, 71
- Restaurant_Window.cpp, 92
- Restaurant_Window.h, 92
- RestaurantFight_Window, 71
 - ~RestaurantFight_Window, 72
 - RestaurantFight_Window, 72
 - Start, 72
- RestaurantFight_Window.h, 92
- ResteaurantFight_Window.cpp, 93
 - setText1, 93
- row
 - Menu, 62
 - Restaurant_Window, 70
- set_Image
 - View_Item, 76
- setButton
 - Button, 19
- setHealth
 - Champion, 31
- setPosition_figure
 - Map_Windows, 54
- setText
 - Arena_Window.cpp, 86
 - Button, 19
- setText1
 - ResteaurantFight_Window.cpp, 93
- Shop_Windows, 73
 - ~Shop_Windows, 74
 - moneycheck, 74
 - Shop_Windows, 73
 - Start, 74
- Shop_Windows.cpp, 93
- Shop_Windows.h, 93
- size
 - Menu, 62
 - Restaurant_Window, 70
- Source.cpp, 94
 - main, 94
- Start
 - AddHeroe_Window, 9
 - Arena_Window, 15
 - Create_Window, 35
 - ErrorShow_Window, 44

- Login_Window, [49](#)
 - Map_Windows, [54](#)
 - Menu, [59](#)
 - Restaurant_Window, [69](#)
 - RestaurantFight_Window, [72](#)
 - Shop_Windows, [74](#)
- startFight
 - Fight, [47](#)
- strength
 - Champion, [33](#)
- takeMoney
 - Champion, [31](#)
- text
 - Button, [20](#)
- Text_Data
 - Menu, [62](#)
 - Restaurant_Window, [71](#)
- ThrowLoad.cpp, [94](#)
- title
 - Menu, [62](#)
 - Window_Base, [81](#)
- updateChampion
 - DBcontroler, [43](#)
- View_Item, [74](#)
 - ~View_Item, [76](#)
 - getImage, [76](#)
 - Image, [76](#)
 - item, [77](#)
 - set_Image, [76](#)
 - View_Item, [75](#)
- View_Item.cpp, [94](#)
- View_Item.h, [94](#)
- viewItem_figure
 - Map_Windows, [56](#)
- viewItem_key_arena
 - Map_Windows, [56](#)
- viewItem_key_rest
 - Map_Windows, [56](#)
- viewItem_key_shop
 - Map_Windows, [56](#)
- Warrior, [77](#)
 - attack, [78](#)
 - block, [78](#)
 - clone, [79](#)
 - Warrior, [78](#)
- Warrior.cpp, [95](#)
- Warrior.h, [95](#)
- what
 - MyException, [67](#)
- width
 - Menu, [63](#)
 - Window_Base, [81](#)
- window
 - AddHeroe_Window, [11](#)
 - Create_Window, [37](#)
 - Login_Window, [51](#)
 - Map_Windows, [57](#)
 - Menu, [63](#)
 - Window_Base, [82](#)
- Window_Base, [79](#)
 - ~Window_Base, [80](#)
 - background, [81](#)
 - backgroundImage, [81](#)
 - getWindows, [81](#)
 - height, [81](#)
 - title, [81](#)
 - width, [81](#)
 - window, [82](#)
 - Window_Base, [80](#)
- Window_Base.cpp, [95](#)
- Window_Base.h, [95](#)
- Wizzard, [82](#)
 - attack, [83](#)
 - clone, [83](#)
 - Wizzard, [82](#)
- Wizzard.cpp, [96](#)
- Wizzard.h, [96](#)
- write
 - Button, [20](#)