

# Arcade

0.1

Generated by Doxygen 1.9.1



<b>1 Class Index</b>	<b>1</b>
1.1 Class List . . . . .	1
<b>2 File Index</b>	<b>3</b>
2.1 File List . . . . .	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 IGame Class Reference . . . . .	5
3.1.1 Constructor & Destructor Documentation . . . . .	5
3.1.1.1 ~IGame() . . . . .	5
3.1.2 Member Function Documentation . . . . .	5
3.1.2.1 checkCollision() . . . . .	6
3.1.2.2 gameKeys() . . . . .	6
3.1.2.3 gameLoop() . . . . .	6
3.1.2.4 gameRestart() . . . . .	6
3.1.2.5 getAssets() . . . . .	7
3.1.2.6 getElements() . . . . .	7
3.1.2.7 getId() . . . . .	7
3.1.2.8 getMap() . . . . .	7
3.1.2.9 getScore() . . . . .	8
3.1.2.10 isAlive() . . . . .	8
3.1.2.11 loadMap() . . . . .	8
3.1.2.12 parseMap() . . . . .	8
3.2 IGraphic Class Reference . . . . .	9
3.2.1 Constructor & Destructor Documentation . . . . .	9
3.2.1.1 ~IGraphic() . . . . .	9
3.2.2 Member Function Documentation . . . . .	9
3.2.2.1 assetsLoad() . . . . .	9
3.2.2.2 findTuple() . . . . .	10
3.2.2.3 getKey() . . . . .	10
3.2.2.4 getSize() . . . . .	10
3.2.2.5 isRunning() . . . . .	11
3.2.2.6 printElements() . . . . .	11
3.2.2.7 printMap() . . . . .	11
3.2.2.8 printText() . . . . .	12
3.2.2.9 windowClear() . . . . .	12
3.2.2.10 windowClose() . . . . .	12
3.2.2.11 windowRefresh() . . . . .	13
<b>4 File Documentation</b>	<b>15</b>
4.1 IGame.hpp File Reference . . . . .	15
4.2 IGraphic.hpp File Reference . . . . .	15
<b>Index</b>	<b>17</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<b>IGame</b>	.....	5
<b>IGraphic</b>	.....	9



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<b>IGame.hpp</b>	15
<b>IGraphic.hpp</b>	15





## Chapter 3

# Class Documentation

### 3.1 IGame Class Reference

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

#### Public Member Functions

- virtual **~IGame** ()=default
- virtual void **gameLoop** ()=0
- virtual void **gameKeys** (int)=0
- virtual void **gameRestart** ()=0
- virtual size\_t **getScore** ()=0
- virtual bool **isAlive** ()=0
- virtual void **loadMap** ()=0
- virtual void **parseMap** ()=0
- virtual std::vector< std::string > **getMap** ()=0
- virtual std::vector< std::tuple< char, char, std::string > > **getAssets** ()=0
- virtual std::vector< std::pair< char, pos\_t > > **getElements** ()=0
- virtual void **checkCollision** ()=0
- virtual std::string **getId** ()=0

#### 3.1.1 Constructor & Destructor Documentation

##### 3.1.1.1 ~IGame()

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

#### 3.1.2 Member Function Documentation

### 3.1.2.1 checkCollision()

```
virtual void IGame::checkCollision ( ) [pure virtual]
```

Function who will check if the player run into wall, ennemies, ...

#### Returns

Nothing

### 3.1.2.2 gameKeys()

```
virtual void IGame::gameKeys (
    int ) [pure virtual]
```

Function for manage the key pressed

#### Parameters

<i>int</i>	ASCII of the key pressed
------------	--------------------------

#### Returns

Nothing

### 3.1.2.3 gameLoop()

```
virtual void IGame::gameLoop ( ) [pure virtual]
```

Function for loop through the game

#### Returns

Nothing

### 3.1.2.4 gameRestart()

```
virtual void IGame::gameRestart ( ) [pure virtual]
```

Function for restart the game from the beginning

#### Returns

Nothing

### 3.1.2.5 getAssets()

```
virtual std::vector<std::tuple<char, char, std::string> > IGame::getAssets ( ) [pure virtual]
```

Return a vector with all the info for the assets

#### Returns

vector of <ASCII in map, ASCII to print, path to image>

### 3.1.2.6 getElements()

```
virtual std::vector<std::pair<char, pos_t> > IGame::getElements ( ) [pure virtual]
```

Function for get the specific element to print

#### Returns

vector of <pair<ASCII to print, position to print>>

### 3.1.2.7 getId()

```
virtual std::string IGame::getId ( ) [pure virtual]
```

Function who return Id of the game

#### Returns

std::string ID of the game

### 3.1.2.8 getMap()

```
virtual std::vector<std::string> IGame::getMap ( ) [pure virtual]
```

Return the map

#### Returns

std::vector<std::string> map in ASCII

### 3.1.2.9 `getScore()`

```
virtual size_t IGame::getScore ( ) [pure virtual]
```

Function who return the score

#### Returns

size\_t Actual score

### 3.1.2.10 `isAlive()`

```
virtual bool IGame::isAlive ( ) [pure virtual]
```

Return the state of the player

#### Returns

true if alive, false if dead

### 3.1.2.11 `loadMap()`

```
virtual void IGame::loadMap ( ) [pure virtual]
```

Function for load the map directly in a class variable

#### Returns

Nothing

### 3.1.2.12 `parseMap()`

```
virtual void IGame::parseMap ( ) [pure virtual]
```

Function for parse the map and create enemies or player if you encounter the right ascii character

#### Returns

Nothing

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

- **IGame.hpp**

## 3.2 IGraphic Class Reference

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

### Public Member Functions

- virtual **~IGraphic** ()=default
- virtual void **assetsLoad** (std::vector< std::tuple< char, char, std::string >>)=0
- virtual void **windowRefresh** ()=0
- virtual void **windowClear** ()=0
- virtual void **windowClose** ()=0
- virtual bool **isRunning** ()=0
- virtual int **getKey** ()=0
- virtual int **getSize** ()=0
- virtual void **printText** (std::string, pos\_t)=0
- virtual void **printMap** (std::vector< std::string >, pos\_t)=0
- virtual void **printElements** (std::vector< std::pair< char, pos\_t >>, pos\_t)=0
- virtual int **findTuple** (char)=0

### 3.2.1 Constructor & Destructor Documentation

#### 3.2.1.1 ~IGraphic()

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

### 3.2.2 Member Function Documentation

#### 3.2.2.1 assetsLoad()

```
virtual void IGraphic::assetsLoad (
    std::vector< std::tuple< char, char, std::string >> ) [pure virtual]
```

Function for load the assets (in ASCII or path for image)

#### Parameters

<i>std::vector&lt; std::tuple&lt; c, d, std::string &gt;&gt;</i>	
<i>c</i>	Char in the ASCII map
<i>d</i>	Char to print if in ASCII
<i>std::string</i>	path to image

**Returns**

nothing

**3.2.2.2 findTuple()**

```
virtual int IGraphic::findTuple (
    char ) [pure virtual]
```

Function for find specific ID in a list of assets

**Parameters**

<i>char</i>	ID
-------------	----

**Returns**

Position of this ID in assets list

**3.2.2.3 getKey()**

```
virtual int IGraphic::getKey ( ) [pure virtual]
```

Get the ASCII code of the key pressed

**Returns**

ASCII of the key pressed

**3.2.2.4 getSize()**

```
virtual int IGraphic::getSize ( ) [pure virtual]
```

Get the window width

**Returns**

window's width

### 3.2.2.5 isRunning()

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

Return the window state as a boolean

#### Returns

true if open, false if closed

### 3.2.2.6 printElements()

```
virtual void IGraphic::printElements (
    std::vector< std::pair< char, pos_t >> ,
    pos_t ) [pure virtual]
```

Function for print a text on the window at precise coordinate

#### Parameters

<i>std::vector&lt; std::pair&lt;char,pos&gt;&gt;</i>	
<i>char</i>	ID of the element
<i>pos</i>	Position for the element
<i>pos_t</i>	Corner top left of the start to print

#### Returns

Nothing

### 3.2.2.7 printMap()

```
virtual void IGraphic::printMap (
    std::vector< std::string > ,
    pos_t ) [pure virtual]
```

Function for print map

#### Parameters

<i>std::string</i>	Path to map
<i>pos_t</i>	Corner top left of the start to print

**Returns**

Nothing

**3.2.2.8 printText()**

```
virtual void IGraphic::printText (
    std::string ,
    pos_t ) [pure virtual]
```

Function for print a text on the window at precisse coordonate

**Parameters**

<i>std::string</i>	String to print
<i>pos_t</i>	Position of the text

**Returns**

Nothing

**3.2.2.9 windowClear()**

```
virtual void IGraphic::windowClear ( ) [pure virtual]
```

Function for clear the window

**Returns**

nothing

**3.2.2.10 windowClose()**

```
virtual void IGraphic::windowClose ( ) [pure virtual]
```

Function for close the window

**Returns**

nothing



### 3.2.2.11 windowRefresh()

```
virtual void IGraphic::windowRefresh ( ) [pure virtual]
```

Function for refresh the window

#### Returns

nothing

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

- **IGraphic.hpp**



## Chapter 4

# File Documentation

### 4.1 IGame.hpp File Reference

```
#include <vector>
#include <string>
#include <tuple>
#include "Pos.hpp"
Include dependency graph for IGame.hpp:
```

### 4.2 IGraphic.hpp File Reference

```
#include <vector>
#include <tuple>
#include <string>
#include "../games/Pos.hpp"
Include dependency graph for IGraphic.hpp:
```

### Classes

- class **IGraphic**



# Index

- ~IGame
  - IGame, 5
- ~IGraphic
  - IGraphic, 9
- assetsLoad
  - IGraphic, 9
- checkCollision
  - IGame, 5
- findTuple
  - IGraphic, 10
- gameKeys
  - IGame, 6
- gameLoop
  - IGame, 6
- gameRestart
  - IGame, 6
- getAssets
  - IGame, 6
- getElements
  - IGame, 7
- getId
  - IGame, 7
- getKey
  - IGraphic, 10
- getMap
  - IGame, 7
- getScore
  - IGame, 7
- getSize
  - IGraphic, 10
- IGame, 5
  - ~IGame, 5
  - checkCollision, 5
  - gameKeys, 6
  - gameLoop, 6
  - gameRestart, 6
  - getAssets, 6
  - getElements, 7
  - getId, 7
  - getMap, 7
  - getScore, 7
  - isAlive, 8
  - loadMap, 8
  - parseMap, 8
- IGame.hpp, 15
- IGraphic, 9
  - ~IGraphic, 9
  - assetsLoad, 9
  - findTuple, 10
  - getKey, 10
  - getSize, 10
  - isRunning, 10
  - printElements, 11
  - printMap, 11
  - printText, 12
  - windowClear, 12
  - windowClose, 12
  - windowRefresh, 12
- IGraphic.hpp, 15
- isAlive
  - IGame, 8
- isRunning
  - IGraphic, 10
- loadMap
  - IGame, 8
- parseMap
  - IGame, 8
- printElements
  - IGraphic, 11
- printMap
  - IGraphic, 11
- printText
  - IGraphic, 12
- windowClear
  - IGraphic, 12
- windowClose
  - IGraphic, 12
- windowRefresh
  - IGraphic, 12