

# GAME LIBRARY “IGAMEMODULE” DOCUMENTATION

**void IGameModule::reset()**

Reset the game.

**bool IGameModule::loadFromFile(const std::string &filepath)**

Load highscores from **filepath** and return wether it worked or not.

**Parameters**

**filepath** the filepath to load highscores

**Return**

True if it worked  
False if it not worked

**bool IGameModule::loadFromFile()**

Load highscores and return wether it worked or not.

**Return**

True if it worked  
False if it not worked

**bool IGameModule::saveToFile(const std::string &filepath) const**

Save highscores to file return wether it worked or not.

**Parameters**

**filepath** the filepath to save highscores

**Return**

True if it worked  
False if it not worked

**bool IGameModule::saveToFile() const**

Load highscores and return wether it worked or not.

**Return**

True if it worked  
False if it not worked

```
void IGameModule::setPlayerName(const std::string &name)
```

Set the player's name.

```
std::pair<std::string, int> IGameModule::getScore() const
```

Get the current score.

**Return**

The score and with player name

```
std::vector<std::pair<std::string, int>> IGameModule::getBestScore() const
```

Get the 16 best scores.

**Return**

The 16 best scores with them player name

## HANDLE GAME

```
void IGameModule::update(const IDisplayModule &lib)
```

Update the game.

**Parameters**

**lib** the library for the display

```
void IGameModule::render(IDisplayModule &lib) const
```

Render the game.

**Parameters**

**lib** the library for the display

```
std::string &IGameModule::getLibName() const
```

Get the library name.

**Return**

The name of the library

**OUR ENTRY POINT TO INSTANTIATE THE LIBRARY:**  
**The symbol is “createLib”.**

```
extern "C" std::unique_ptr<IGameModule> createLib(void)
{
    return std::make_unique<"gameClass">();
}
```