IGame (../games/IGame.hpp)

The IGame interfaces is the base of all games.

Read more to learn how to implement new games.

Add a new game

To create a new game you must generate a dynamic library which at least contains two functions:

- init_game_lib
- get_lib_name

The first step to create a new game would be to create your own class that inherits from `IGame` and therefore would have at least these three methods:

- IGame::launch
- IGame::stop
- IGame::getScore

/!\ Your dynamic library (your game) must be placed at the root of the `./games` directory in order to be recognized by the Core /!

init_game_lib (../games/init_game_lib.hpp)

```
extern "C" std::unique_ptr<arcade::IGame> init_game_lib(arcade::ICore& core);
```

> This function instantiates `std::unique_ptr<arcade::IGame>`, which is basically you game. Notice that it takes a reference to the `Core` (or rather `ICore`) as parameter for your game to have access to its methods.

get_lib_name (../deps/get_lib_name.hpp)

```
extern "C" std::string get_lib_name();
```

> This function return the name of your game as an `std::string`.

IGame::launch

```
void IGame::launch();
```

> Starts your game. **This method must return once `IGame::stop` is called**

IGame::stop

```
void IGame::stop();
```

Stops the game loop if it was running

IGame::getScore

```
unsigned long IGame::getScore() const;
```

Returns the current game score.