# ILibGraph (../lib/ILibGraph.hpp)

The ILibGraph interfaces is the base of all graphic libraries.

Read more to learn how to implement new games.

## Add a new graphic library

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

- init_graph_lib
- get_lib_name

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

- ILibGraph::loadResource[...]
- ILibGraph::resetResource
- ILibGraph::getKeyboardEvents
- ILibGraph::displayImage
- ILibGraph::displayText
- ILibGraph::playAudio
- ILibGraph::stopAudio
- ILibGraph::clear
- ILibGraph::render

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

### init_graph_lib (../lib/init_graph_lib.hpp)

```
extern "C" std::unique_ptr<arcade::ILibGraph> init_graph_lib();
```

> This function instantiates `std::unique_ptr<arcade::ILibGraph>`, which is basically you gfx library.

### 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`.

### ILibGraph::loadResource[...]

```
void loadResourceAudio(int id, std::string filepath);
void loadResourceFont(int id, std::string filepath);
void loadResourceImage(int id, std::string filepathGraph, std::string
filepathAscii);
```

> These methods loads different types of resources for them to be referred by their ids' later. If an id is already taken the existing resource will be replaced. If a file cannot be loaded, an exception must be thrown.

### ILibGraph::resetResource

```
void resetResource();
```

> Erases internal resource lists.

### ILibGraph::getKeyboardEvents

```
void getKeyboardEvents(std::vector<KeyState> &keysGame, std::vector<KeyState>
&keysCore)
```

> Takes two vectors of KeyState as parameters (see [KeyState (./KeyState.md)](./KeyState.md)) which represents the keys you need to check the state of. The method will then set the state of every key (true/false) in the vector depending on the latest keyboard inputs.

### ILibGraph::displayImage

```
void displayImage(int id, int posX, int posY);
void displayImage(int id, double posX, double posY);
```

> Draws the image pointed by `id` at `posX` and `posY` positions.

### ILibGraph::displayText

```
void displayText(int id, int posX, int posY, std::string const &text);
```

> Draws `text` with the font pointed by `id` at `posX` and `posY` positions.

### ILibGraph::playAudio

```
void playAudio(int id, bool repeat = false);
```

> Plays the audio pointed by `id` and sets it to repeat or not depending on the value of `repeat`. If the audio was already playing it restarts it.

## ILibGraph::stopAudio

```
void stopAudio(int id);
```

Stops the audio pointed by `id` from playing.

## ILibGraph::clear

```
void clear();
```

Clears the window.

## ILibGraph::render

```
void render();
```

Renders what was drawn on the window.