

How to implement new graphical libraries or game libraries

Game library:

Create an instance of a class deriving from game::IGame interface.

Functions to implement:

```
void run(std::unique_ptr<graphical::IGraphical> &lib, game::event::Events &event);
```

Main function of the game. It represents a frame to display: handles events, moves characters, e.t.c...

```
void loadGameObjects(std::unique_ptr<graphical::IGraphical> &lib);
```

Creates all necessary objects to run the game: a map, a character, e.t.c...

```
void setGraphicalLib(graphical::IGraphical &lib);
```

Reloads objects from the game with the new library to be used.

Any additional function to handle your game is optional.

Graphical library:

Create an instance of a class deriving from graphical::IGraphical interface.

Functions to implement:

```
void pollEvent(game::event::Events &event);
```

Retrieves occuring events and fills the parameter event with its specific features.

```
std::unique_ptr<graphical::IObject> createObject(const std::string &filepath);
```

Creates an IObject instance from a config file at path **filepath**.

```
void reloadSprite(std::unique_ptr<graphical::IObject> &sprite);
```

Recreates **sprite** with the same specific features (position, e.t.c...).

(Self explanatory):

```
void windowClear(void);
```

```
void windowOpen(void);
```

```
void windowClose(void);
```

```
void windowDisplay(void);
```

```
void drawObject(std::unique_ptr<IObject> &sprite);
```

Any additional function to handle your graphical library is optional.