

Add a graphic library

development

To design your own functional graphic library on our Arcade, you must comply with the following guidelines:

The main class of your library must be inherited from the `IDisplayModule` interface. East must therefore reuse all of its methods.

here is a description of the methods of the `IDisplayModule` interface:

- `isOpen ()`: returns the status of the window to display.
- `initWindow ()`: initialize your graphics library, the same task must be done in the constructor of the main class
- `destroyWindow ()`: deletes and cleans up your graphics library, the same task must be performed by the destructor of the main class.
- `drawEntity (Entity)`: receives an `Entity` class with a type and a position. You must display the entity on the window at the given position according to its type (if `PLAYER`: display `image_player`, etc ...)
- `eventListener ()`: listens to keyboard and mouse input. returns a value corresponding to an index of the `KeyBind` enum
- `displayMenu (MenuInfo)`: receives the `MenuInfo` class and must be able to display all of its information in the form of a menu.
- `oneCycleDisplay ()`: must display on the window the buffer accumulated and modified by the `drawEntity` method.
- `oneCycleClear ()`: cleans the window to allow the next display cycle to be clean.

your library should be in a folder located in the **lib** folder

Compilation

The library must be compiled with a Makefile.

in the sources necessary for compilation, you must include the path to the `MenuInfo` class located in the sources folder located at the root of the project.

The name of your graphic library must follow the following standard:

```
---  
lib_arcade_ [name].so  
---
```

Here is an example of Makefile for your graphics library:

```
---
LIBSRC= ../../src/MenuInfo.cpp \
        src/LibName.cpp \
        src/MenuName.cpp
LIBOBJ= $(LIBSRC:.cpp=.o)
LIBNAME= lib_arcade_Name.so
CXXFLAGS= -std=c++11 -I../../include -Iinclude -Wall -Wextra -Wshadow -fPIC
LIBFLAGS= -llibflag -llibflag2
all:    $(LIBOBJ)
        g++ --shared -fPIC -o $(LIBNAME) $(LIBOBJ) $(LIBFLAGS)
        mv $(LIBNAME) ..
clean:
        rm -rf $(LIBOBJ)
fclean: clean
        rm -rf $(LIBNAME)
re: fclean all
.PHONY: all clean fclean re
---
```