Add a game library

development

To develop your own game library compatible with our arcade, here are some standards to respect in order to avoid any compatibility problem

Your game library must inherit the IGameModule interface.

here is an explanation of the methods necessary for the proper functioning of your game library:

- receiveEvent (KeyBind): receives the events perceived by the graphic library in the form of an index of the KeyBind enum. from this, you can configure specific actions according to each event.
- getEntities (): returns a vector filled with entities that the game wants to display. these entities will then go to the graphics library to be displayed on the window. an Entity is represented by the Entity class and must contain a type and a position.
- oneCycleLoop (): performs the cyclic actions of your game, on a cycle.
- getScore (): returns the score obtained by the player during the game in the form of an Int.

In order to develop your game with a more advanced and generic system, we have made a customizable ECS System available to you. If you use it, you will have to include its files in the compilation of your game library. For more informations about the ECS system, follow this link

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

Compilation

The library must be compiled with a Makefile.

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

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

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

Here is an example of Makefile for your games library:

```
---
LIBSRC= ../../src/Entity.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
---
```