ARCADE

All added library must come with their own Makefile. Game library must be compiled in the games rule of the root Makefile, graphic library must be compiled in the graphicals rule of the root Makefile.

All library must come with the following extern function :

```
extern "C" <interface>  *start()
{
   return new <name>;
}
```

Adding a game library :

Every game library must inherite from the AGame abstract class.

Constructor must initialize the value of the std::vector<std::vector<int>> _board attribute of AGame class and take no argument;

void update(void) change value of the _board attribute at each call.

void action(void), void left(void), void right(void), void up(void), void down(void) : Each call to those must dictate the next change applied with an update( ) call. No change to the _board attribute must be applied during the call of this function.

Adding a graphic library :

Every graphic library must inherite from the Alib abstract class.

Constructor must take no argument.

int open(std::vector<std::vector<int>>) take as parameter the game _board attribute and initialize all necessary sub-library and attribute. Return value should be COMMAND_CONTINUE or COMMAND_ERROR in case of error.

int run(std::vector<std::string>, std::vector<std::vector<int>>, int) take as parameter the list of graphic library, the game _board attribute and the game _score attribute. This method must call in turn the method printLib, printScore, printName and displayGame and return COMMAND_CONTINUE if no keyboard input was given by the user or instead the corresponding COMMAND constant must be returned.

void printLib(std::vector<std::string>), printScore(int), printName(void) and displayGame(std::vector<std::vector<int>>), displayPause() must display respectively : the name of the graphics library, the score of the running game, the name of the player, the actual game and a Pause message instead of the game window.