

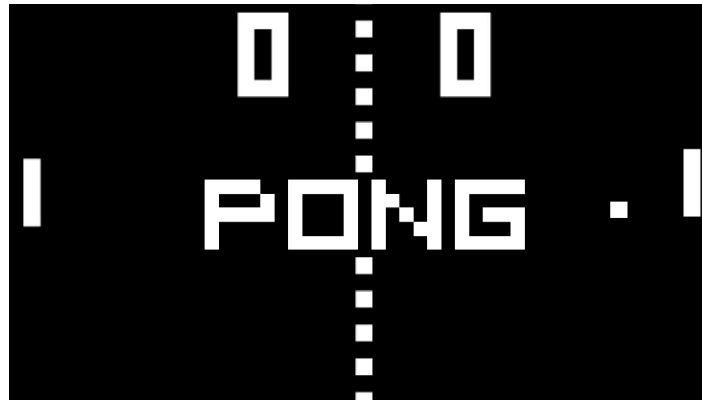


# B4 - Object-Oriented Programming

B-OOP-400

## Arcade

Documentation



# Arcade

binary name: arcade  
repository name: arcade  
language: everything working on "the dump"  
build tool: no need here



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).



Arcade is a gaming platform that lets the user choose a game to play and keeps a register of player scores.

The elements of our gaming platform, our graphics libraries and our games dynamic libraries were loaded at run-time.

Each GUI available for the program are shared libraries that are loaded at run-time.



Graphic libraries needs to contain the following functions:

A function that defines the entry point of the library. This function must return a pointer to a class that inherits from IDisplayModule.

```
Terminal
~/B-00P-400>
extern "C"
IDisplayModule *entryPoint()
{
    return new MyDisplayModule();
}
```

This function initializes the window.

```
Terminal
~/B-00P-400>
void Library::InitWindow()
{
}
```

This function closes the window.

```
Terminal
~/B-00P-400>
void Library::FiniWindow()
{
}
```

This function displays objects on the screen.

```
Terminal
~/B-00P-400>
void Library::displayObjects(std::map<int, std::pair<Enum::ObjectType,
std::pair<int, int>>> _ObjectData)
{
}
```

This function displays the score on the screen.



```
Terminal
~/B-00P-400>
void Library::displayScore(int _Score, int x, int y)
{
}
}
```

This function displays a text on the screen.

```
Terminal
~/B-00P-400>
void Library::displayText(std::string _String, std::pair<int, int> _Pos,
Enum::Color FrontFont, Enum::Color BackFont)
{
}
}
```

This function gets the type of the library.

```
Terminal
~/B-00P-400>
Enum::libType Library::GetLibType()
{
}
}
```

This function gets the type of the library.

```
Terminal
~/B-00P-400>
std::pair<int, int> LibrarySFML::GetWindowSize()
{
}
}
```

This function gets user input.

```
Terminal
~/B-00P-400>
char Library::getUserInput()
{
}
```

This function displays drawn objects on the screen.

```
Terminal
~/B-00P-400>
void Library::display()
{
}
```