

# CPP\_Arcade

## Launcher

The executable called *the launcher* can be launched with the command below, from the root of the repository.

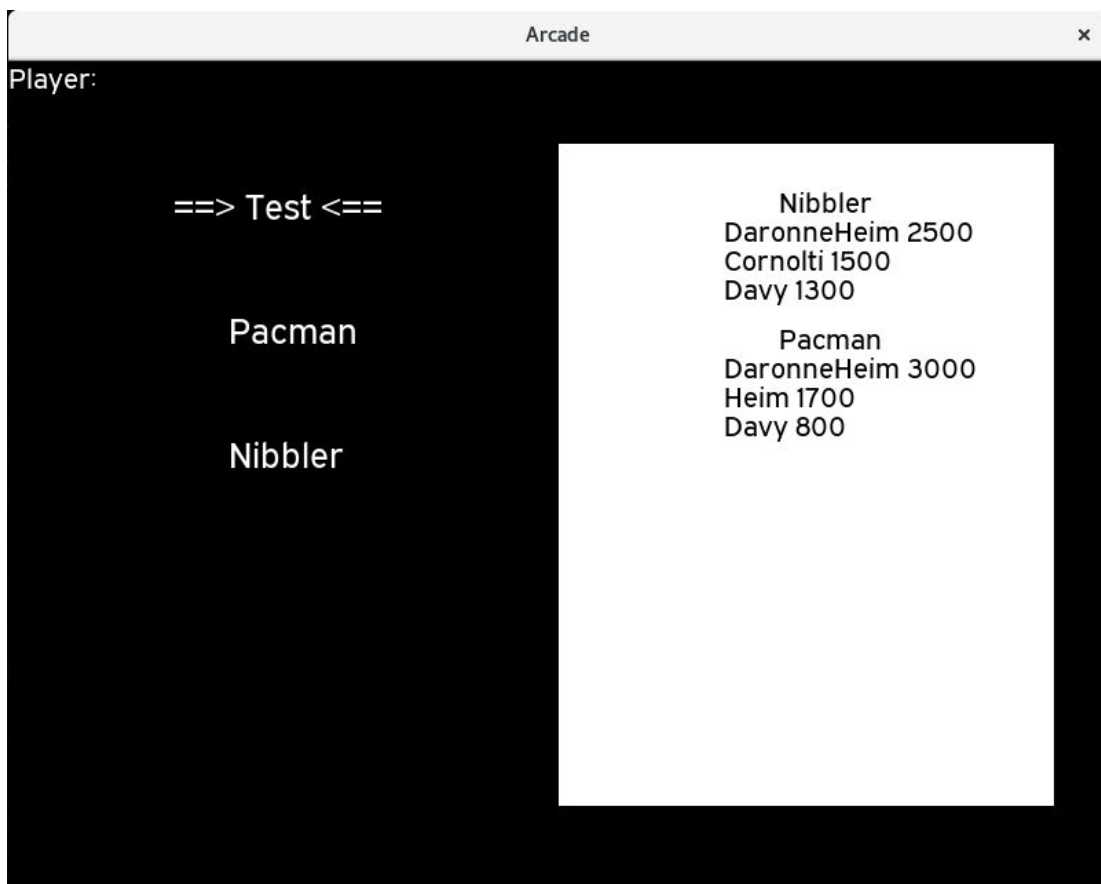
```
./arcade [graphical_lib], [graphical_lib]
```

being an external graphical library you want to load to display menu and games.

## Menu

Menu is the first screen to be displayed by the launcher. Thanks to the menu, you can choose your game and graphical libraries before playing.

Simply press on *UP* and *DOWN* arrow to change the game and *RIGHT* and *LEFT* arrow to change the graphical library. These keys will have the same functions at anytime.



The menu is divided into two boxes. The first one holds the game list, meanwhile the second one holds the scoreboard.

## Game list

The game list is read by the launcher in the folder `games` as well as the graphical libraries in `lib`.

## Scoreboard

Scoreboard is displayed beside the game list in the menu. It contains, for each game, the three best people in the world with their points. The scoreboard is wrote in the `scores` file with the following template:

```
<game name>:<player name> <score>:<player name> <score>:<player name>
<score>
```

## Control

The launcher introduces five key bindings. *UP* and *DOWN* are used to dynamically change the game, as well as *RIGHT* and *LEFT* to dynamically change the graphical library and *ESCAPE* to quit the game or the launcher.

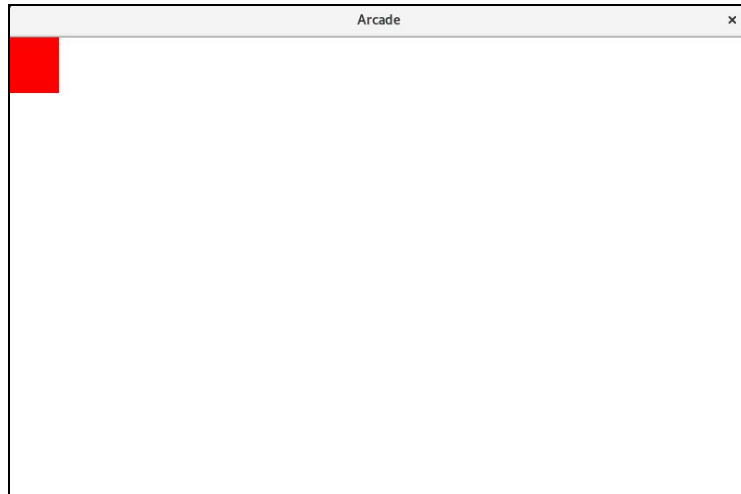
## Graphical libraries

Graphical libraries are stored in the `lib` directory. Each game has a folder named `<game name>` and a binary named `lib_arcade_<game name>.so`.

# Games

## Test Game

### Screenshot



### History

*Test Game* has no history. It's a really bad and boring game.

### Goal

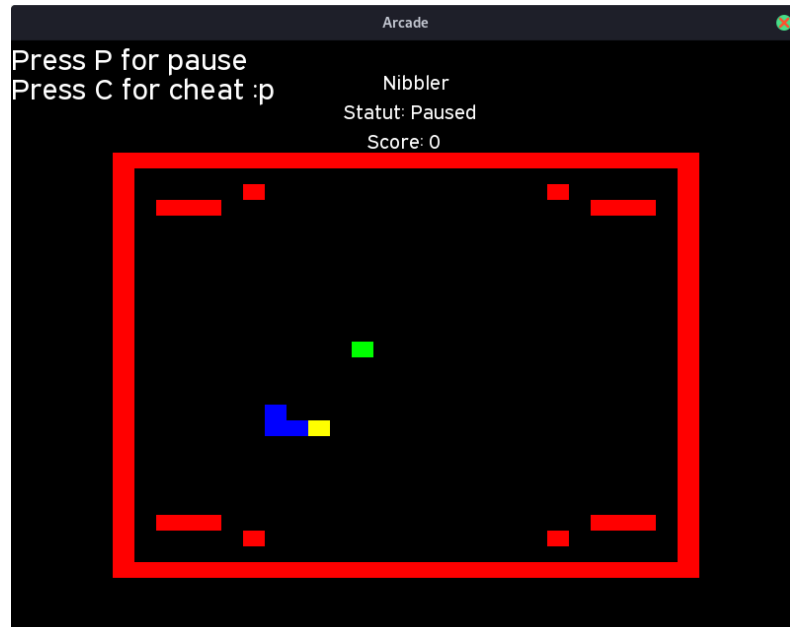
The goal is to show an example to create games with our specifications. *Test Game* only allows to move a player represented by a simple red square. Thanks to this game, you can learn how to manage a player and a map, and how to use the scoreboard or the scaling.

### Control

Controls are as simple as breathing. ZQSD to move the square, that's it.

# Nibbler

## Screenshot



## History

*Nibbler* is an arcade game released in 1982 by Chicago-based developer Rock-Ola, in which players navigate a long snake through an enclosed maze consuming food along the way, while the length of the snake increases with each object consumed. Home versions were produced in 1983 by Datasoft for the Atari 8-bit and the Apple II.

## Goal

A controlled snake continuously navigating into a enclosed maze and eating all foods in its way meanwhile it grows in length as the game progresses. By eating food, your snake increase by one and the frame is reduced by 1 (starting as 150ms). The goal is to make the highest score, simple. Don't hit walls and don't bite your own tail, that makes you loose.

## Control

ZQSD to move the snake

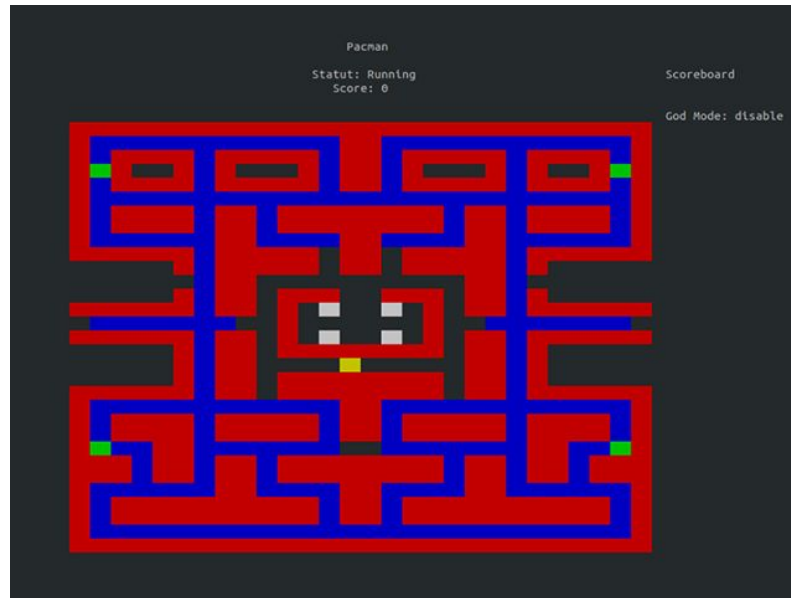
P to pause the game

R to restart the game

C to activate the GOD(BEN)MODE which means you can't die. It's a cheat but don't worries, you won't get VAC Ban for that. In any case we have no anti-cheat so we can't detect it. Remember that great powers lead to great responsibilities.

# Pacman (alpha)

## Screenshot



## History

*Pacman* is an arcade game developed by Namco and first released in Japan in May 1980. It was created by Japanese video game designer Toru Iwatani. Immensely popular from its original release to the present day, *Pac-Man* is considered one of the classics of the medium, and an icon of 1980s popular culture.

## Goal

The goal of pacman is to “eat” all the dot named “pac-dot” (blue square in the screenshot) in the maze of each stage. When a pac-dot is eaten you earn point to reach records (if you win at the end of the 256 original stages). In the mazes there are some special dot named “pac-gum” (green square in the screenshot) that allow pacman to eat the 4 chasing ghost that want to kill pacman.

## Control

ZQSD to move inside the maze

R to reset the game

G to reach the always godmode (switch mode: press to enable, press again to disable)

# Specification

Protocol specification is available in the source code as well as in the `Spec_DOC.pdf` and `Spec_UML.pdf` files. Take a look at the *Test Game* to create a simple game.

Remember that a public Github repository is available on <https://github.com/EPITECH-Strasbourg-2021/Cpp-Arcade-Spec/> with all the necessary documentation. Feel free to open issues for any problem.