

# Documentation Arcade

Arcade is a gaming platform: a program that lets the user choose a game to play and keeps a register of player scores. To be able to deal with the elements of your gaming platform at run-time, graphic libraries and games are implemented as dynamic libraries, loaded at run-time.

Each GUI available for the program must be used as a shared library that will be loaded and used dynamically by the main program.

The architecture is this.

- A Core class orchestrate all the logic of dynamic libraries, run-time changes, ...
- A IGame represent the game.
- A IGraphic represent an graphic library.

# Core

This class manage the game and the graphic library.

## - Core()

Core::Core()

Constructor for class Core.

Search for all « .so » in directories « ./games » and « ./lib » and load it in two vector<std::string>.

## - run()

int Core::run(std::string lib)

Parameters:

lib: path to default graphic lib.

Loop for the game in run():

while:

loop for menu

or

loop for game

- loop()

bool Core::loop(IGame &render)

Parameters:

IGame menu or IGame game.

Loop for IGame:

while:

    handleEvent

    handleUpdate

    handleRender

# IGame

## - handleEvent()

`void IGame::handleEvent(const std::string &name)`

Parameters:

name: The name of the event.

## - handleUpdate()

`void IGame::handleRender(IGraphicRenderer &renderer) const`

Parameters:

renderer: An IgraphicRenderer to interact with the display library.

## - handleRender()

`void IGame::handleUpdate(int elepsedTime)`

Parameters:

elepsedTime: Time elepsed since the last update.

# IGraphic

## - handleEvent()

std::string IGraphic::handleEvent()

This function return an event in string format.

Event	Description	Game	Key (examples, can be changed)
quit	quit the arcade		suppr or window close
next_game	next game		e
prev_game	previous game		c
next_graphic	next graphic lib		a
prev_graphic	previous graphic lib		w
menu	go back to the menu		esc
restart	restart the game		r
enter	enter event	X	enter
space	space event	X	space
left	left event	X	arrow or q
right	right event	X	arrow or d
up	up event	X	arrow or z
down	down event	X	arrow or s