

## Library for an object-oriented programming in C

- ✓ The language used in this project is C
- ✓ The project was in a group of three so we used **git** in order to manage the project and work efficiently.
- ✓ The objective of the project was to implement a library for an object-oriented programming in C. This library will help to create and configure inter-actor such as windows, labels, button etc. We also integrate in the library a geometry manager and events manager.
- ✓ At the end we manage to implement all the functions required in the subject, we manage to display the windows with all its components. However the windows are not too much reactive because in each click on the window we draw all the component. In order to solve this problem, we modify the events managers and the functions that help to draw the components. The new functions draw only the part of the window that changed before the click and after the click. And as we didn't had time to finish the modification before the deadline of the project, we created two directories, the first (named **projet**) hold the old methods events manger. And the second (named **projetSansLag**) hold the new events managers.
- ✓ In this project I worked on displaying graphical elements. Precisely I have implemented the functions :
  - button\_drawfunc, button\_setdefaultsfunc (you can find them in **src/ei\_button.c**)
  - frame\_drawfunc, and frame\_setdefaultsfunc (you can find them in **src/ei\_frame.c**)
  - ei\_place (you can find it in **src/ei\_geometrymanager.c**)
  - all the functions in **src/ei\_pofinet.c**
  - toplevel\_drawfunc and toplevel\_setdefaultsfunc (you can find them in **src/ei\_toplevel.c**)
  - ei\_frame\_configure, ei\_button\_configure, ei\_toplevel\_configure (you can find them in **src/ei\_widget.c**)

## compilation and execution

to compile you can use the following command in the directory **projet** or **projetSansLag** :

**warning** : after launching a program you may need to kill it with the command **xkill** for example.

- ✓ **make frame** → to create a frame. For the execution you could use **./frame**
- ✓ **make button** → to have an example of button. Use **./button** to execute
- ✓ **make hello\_world** → to have an example of button with a text on it. Use **./hello\_world** to execute
- ✓ **make minesweeper** → to have example of a game. Use **./minesweeper** to execute
- ✓ **make puzzle** → to create a puzzle. Use **./puzzle**
- ✓ **make clean** → to clean all the files generate by the compilation.