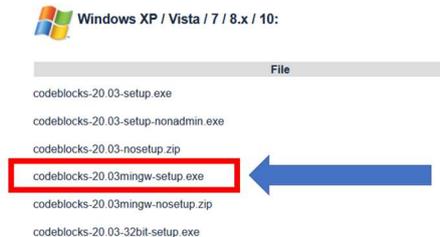


## Etape 1 : Codeblocks (Uniquement Windows et Linux)

Codeblocks est déjà installé sur les ordinateurs de l'université.

Pour installer codeblocks sur votre ordinateur personnel, sous Windows, installer le à partir du lien suivant : <http://www.codeblocks.org/downloads>

Puis "Download the binary release", puis **codeblocks-20.03mingw-setup.exe**



Sous Linux, il faut ajouter un repository (source de paquet) pour avoir la version 17 puis installer codeblocks. Entrer les commandes suivantes

```
sudo add-apt-repository ppa:pasgui
sudo apt update
sudo apt install codeblocks
```

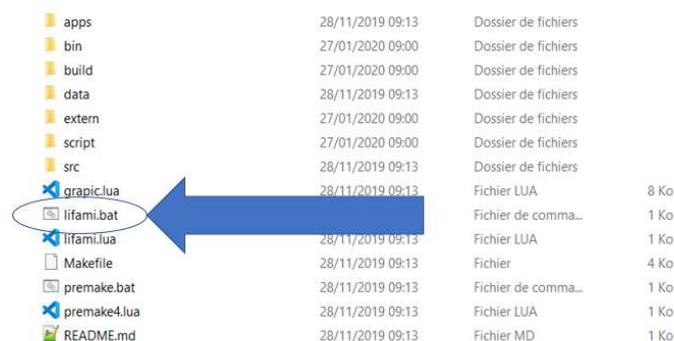
## Etape 1 : sous MacOS installer XCode et mettre à jour la version de l'OS

## Etape 2 : GrAPiC : GRaphics for Algo/Prog In C/C++

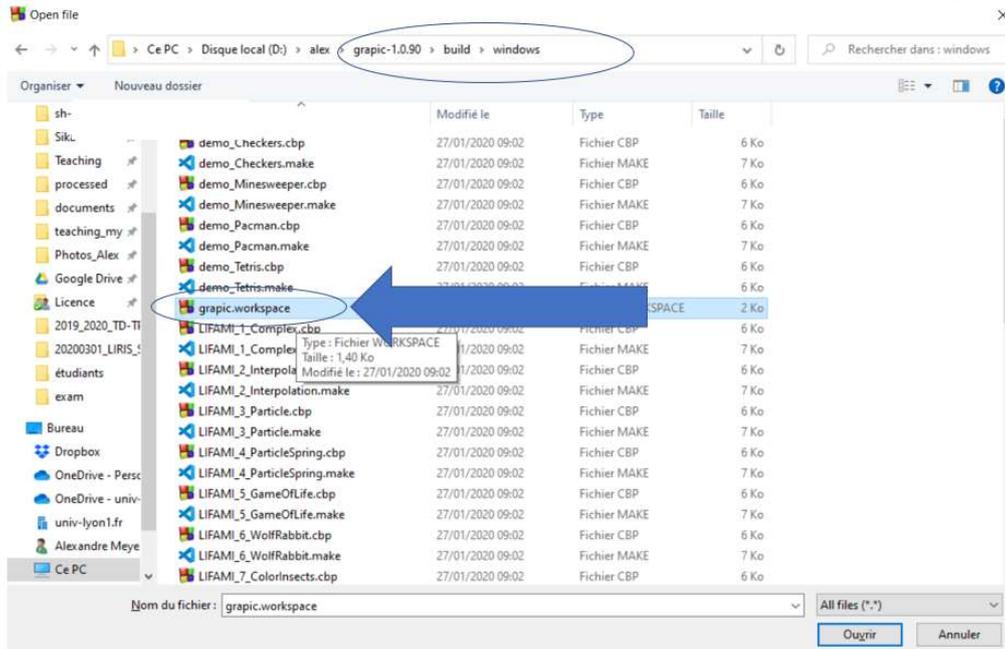
GRAPIC est une librairie facilitant l'affichage graphique pour des programmes en C/C++.

<http://licence-info.univ-lyon1.fr/grapic>

1. Récupérer le fichier .zip qui correspond à votre OS.
2. Décompresser l'archive (.zip ou tgz) dans votre répertoire de travail. **Si vous travaillez sur un ordinateur de l'université sous Windows ne mettez surtout pas vos fichiers dans les répertoires spéciaux de Windows (Mes Documents, Bureau, etc.). Mettez grapic simplement dans W:\**
3. WINDOWS : faire un double clic sur le script *lifami.bat* qui créer tous les projets.  
LINUX ou MacOS : make lifami (dans un terminal)



4. Windows : ouvrir avec Codeblocks le fichier *build/windows/grapic.workspace*



Linux : ouvrir avec Codeblocks le fichier *buid/linux/grapic.workspace*

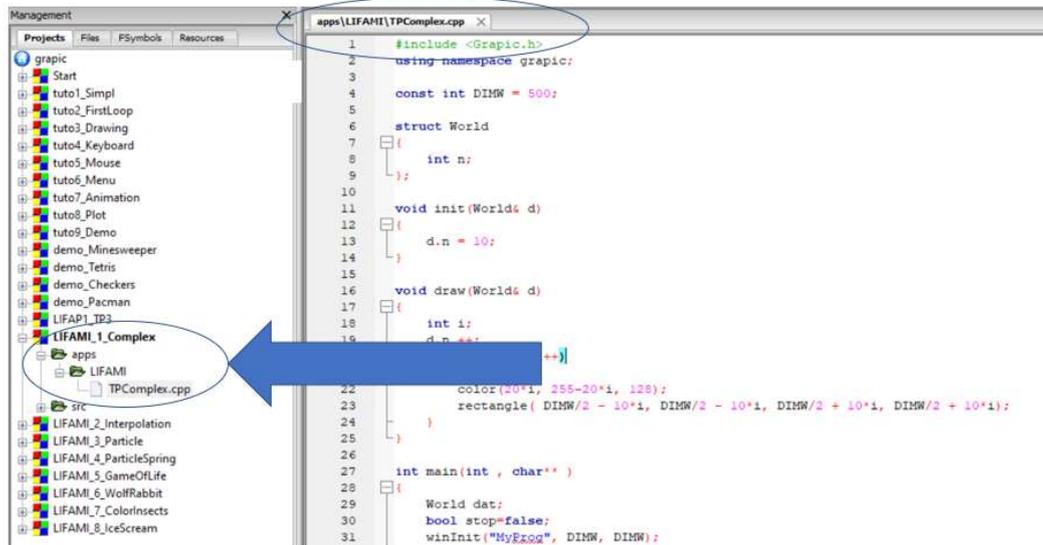
MacOS : ouvrir les fichiers de projet du répertoire *buid/macos* avec XCode

5. Avec Codeblocks, le projet compilé et exécuté est celui en gras.

Un double clic sur un projet rend celui-ci actif.

Faire un double clic sur le projet *LIFAMI\_1\_Complex*

Le fichier source est *apps/LIFAMI/TPComplex.cpp*



Les problèmes courants avec Grapic sont décrits sur la page de Grapic tout à la fin dans la rubrique « Current problems »

Votre programme va souvent ressembler au programme de la page suivante. Vous renommerez la structure *Data* en fonction du problème à résoudre.

```
#include <Grapic.h>
```

```

using namespace grapic;
const int DIMW = 500;

struct Data          // Le type de vos données : A RENOMMER
{
    int n;
};

void init(Data& d)    // Initialise les données
{
    d.n = 10;
}

void update(Data& d)  // Met à jour les données
{
    d.n++;
}

void draw(Data& d)    // Affiche les données
{
    int i;
    for(i=0;i<d.n;i++)
    {
        color(20*i, 255-20*i, 128);
        rectangle( i*5, i*5, (i+1)*5, (i+1)*5 );
    }
}

int main(int , char** )
{
    Data dat;
    Menu m;
    bool stop=false;
    winInit("LIFAMI", DIMW, DIMW);
    init(dat);
    backgroundColor( 100, 50, 200 );
    menu_add( m, "Question 1");
    menu_add( m, "Question 2");
    menu_add( m, "Question 3");
    menu_add( m, "Question 4");

    while( !stop )
    {
        winClear();
        switch(menu_select(m))
        {
            case 0 : dat.n = 5; break;
            case 1 : dat.n = 15; break;
            case 2 : dat.n = 10; break;
            case 3 : dat.n = 20; break;
        }
        draw(dat);
        update(dat);
        menu_draw(m, 5,5, 100, 102);
        stop = winDisplay();
    }
    winQuit();
    return 0;
}

```