

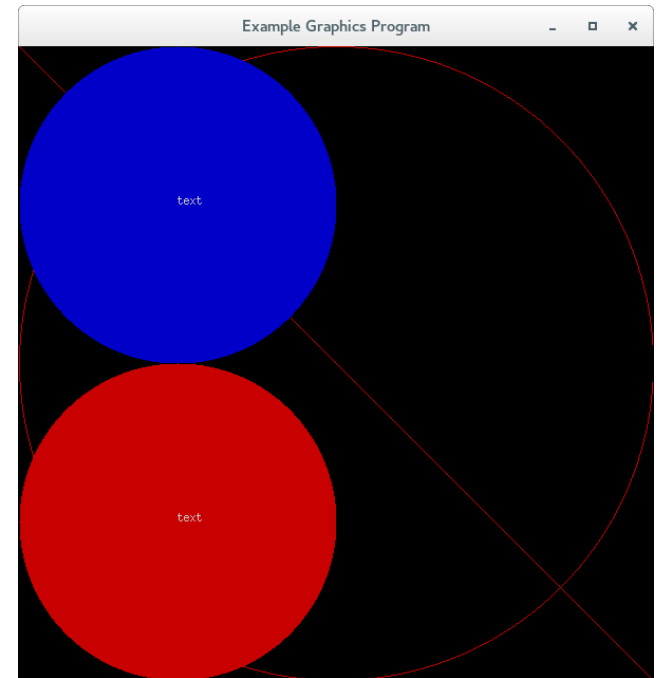
GIT

Comment récupérer les sujets des projets de programmation ?

```
git clone https://gitlab.ensimag.fr/toussaij/projets1a_2024.git
```

gfx : librairie graphique 2D

- Développeurs : Douglas Thain, Jean-Christophe Toussaint
- Plateforme : X11 (Unix)
- librairie contenant 2 fichiers : gfx.h et gfx.c
- tracé d'objets graphiques élémentaires :
points, lignes, rectangles, cercles



gfx: librairie graphique 2D : exemple

- Placez-vous dans le repertoire graph1 :

```
cd ~/projets1A_2023/graph1
```

Continuation ligne
suivante

- Compilez manuellement le projet :

```
gcc -I. -I/usr/X11R6/include main.c gfx.c -o graph1 \  
-L/usr/X11R6/lib -lX11 -lXext -lm
```

Pour executer le programme compile :

```
./graph1
```

Inconvénients de cette approche :

- projet contenu dans un seul fichier « main.c »
⇒ mal adapté pour un gros projet
- commande de compilation difficile à saisir

Projet graph2 multifichiers

On décompose le projet en routines élémentaires

gedit main.h &

```
#ifndef MAIN_H
#define MAIN_H
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <errno.h>
```

```
#include <gfx.h>
```

```
void InitGraph(int taille);
void Draw();
void Animate();
```

```
/* fenetre graphique */
#define L 600
#define FPS 50
#define RECT_WIDTH 300
#define RECT_HEIGHT 300
#endif
```

Fichiers « header » (includes)
des librairies de base de C
Plus celles de la librairie gfx

Déclaration et affectation de
certaines constantes

Pour chaque fonction, on crée
un fichier source :

- InitGraph.c
- Draw.c
- Animate.c
- main.c (programme principal)

Compilation à l'aide de Makefile : automate

make clean; make all
./graph2

Make lance l'interprétation
du fichier « Makefile »

```
# Makefile du Projet: graph2
#
CC      = gcc -ggdb
OBJ     = main.o InitGraph.o Draw.o Animate.o gfx.o
LIBS    = -L/usr/X11R6/lib -lX11 -lXext -lm

INCS    = -I. -I/usr/X11R6/include -I$(HOME)/extralib/include
BIN     = graph2

all:    $(BIN)

clean:
        rm -f $(OBJ) $(BIN)

$(BIN): $(OBJ)
        $(CC) $(INCS) $(OBJ) -o $(BIN) $(LIBS)

.c.o:
        $(CC) $(INCS) -c $<
```

make clean : efface les fichiers
*.o et l'exécutable graph2
make all : crée l'exécutable graph2

```
#include "main.h"
```

fichier « InitGraph.c »

```
void InitGraph(int taille)
{
/* cree une fenetre graphique carree taille = nombre de pixels */
gfx_open(taille, taille,"graph2");
}
```

```
#include <main.h>
```

fichier « Draw.c »

```
void Draw() {
gfx_color(255, 255, 255);
gfx_clear();

/* carre en rouge */
gfx_color(255, 0, 0);
gfx_filled_rectangle(0, 0, L/2, L/2);

/* carre plein en bleu */
gfx_color(0, 0, 255);
gfx_filled_rectangle(L/2, L/2, L/2, L/2);

/* disque en vert */
gfx_color(0, 255, 0);
double Rayon=L/10.;
gfx_filled_circle(L/2, L/2, Rayon);
```

```
/* affichage */
gfx_swap_buffers();

printf("Appuyer sur la touche q\n");
for(;;) {
    char c = gfx_wait();
    if(c=='q') break;
}
}
```

```
#include "main.h"
```

```
void Animate() {  
    int N=100;  
    double Rayon=L/10.;  
    int n;  
  
    for (n=0; n<N; n++) {  
        int x=(int)((double)n/(double)N * L);  
        int y=x;  
  
        gfx_color(255, 255, 255);  
        gfx_clear();  
  
        /* OBLIGATOIRE : capture des evenements */  
        int quit=gfx_event_waiting();  
  
        /* disque en vert */  
        gfx_color(0, 255, 0);  
        gfx_filled_circle(x, y, Rayon);  
        gfx_swap_buffers();  
        usleep(1000*1000/FPS);  
    }  
}
```

fichier « Animate.c »

```
printf("Appuyer encore  
      sur la touche q\n");  
for(;;) {  
    char c = gfx_wait();  
    if(c=='q') break;  
}
```

```
#include "main.h"           fichier « main.c »

int main(void)
{
// creation de la fenetre graphique
    InitGraph(L);

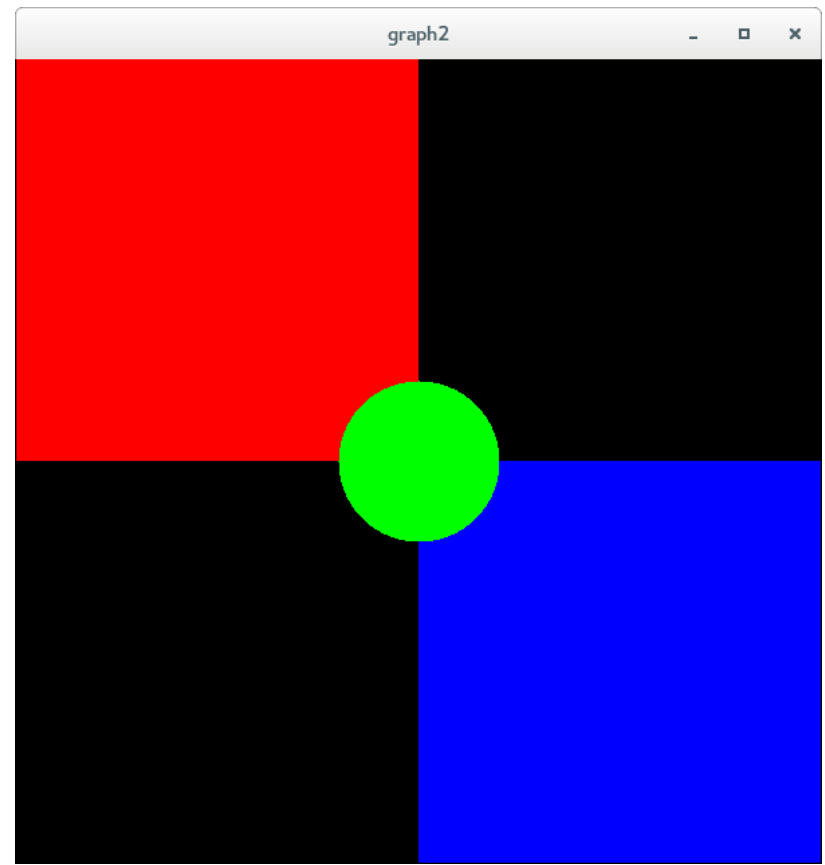
    Draw();

    Animate();

    Rebound();

// fermeture fenetre
    gfx_quit();

    return 0;
}
```



make clean : efface les fichiers *.o (objets binaires)
make all : compile chaque fichier *.c et crée pour
chaque fichier .c un objet binaire .o puis
lie le tout avec les librairies ⇒
exécutable « graph2 »
./graph2 : exécution

Projets

1. Maxwell : simulation d'un gaz de sphères dures en interaction
2. Méthode du simplexe : recherche du minimum local
d'une fonction à plusieurs variables
3. Mouvement chaotique d'un pendule pesant
4. Dynamique d'un satellite ou d'une sonde spatiale
dans le système solaire
5. Modèle d'Ising 2D : transition de phase ferro-paramagnétique
6. Algorithmes génétiques : recherche des minima globaux
d'une fonction à plusieurs variables
7. A la recherche de l'observatoire James Webb