

# LIFAP1 : ALGORITHMIQUE ET PROGRAMMATION IMPÉRATIVE, INITIATION

1

COURS 6 : Programmation graphique

# OBJECTIFS DE LA SÉANCE

- Commencer à faire de la programmation graphique
  - Créer des fenêtres pour dessiner dedans
  - Dessiner des formes de bases
  - Faire évoluer ces formes dans l'espace
  - ...
- Apprendre à utiliser une bibliothèque
  - Savoir comment l'installer
  - Trouver les fonctions qu'elle propose

# PLAN

- Qu'est-ce qu'une bibliothèque ?
- Dessiner
  - sur une console
  - « graphiquement »
- GrAPic : Graphics for Algo/Prog in C/C++
  - Les formes de bases
  - Les couleurs
  - Les chaînes de caractères
  - Les images
  - Les évènements
- Et en TP
  - Comment l'installer
  - Créer un nouveau projet

# LES LIBRAIRIES / BIBLIOTHÈQUES

- 1 bibliothèque est
  - Ensemble de fonctions programmées par d'autres
  - Mises à disposition pour tous
- Constituée
  - Du code source des fonctions
  - D'une documentation listant les fonctions et décrivant les paramètres d'entrée et de sortie
- Beaucoup de bibliothèques disponibles
  - Dans chacun des langages de programmation
  - En C/C++ : stdlib, math, iostream...

# PLAN

- Qu'est-ce qu'une bibliothèque ?
- Dessiner
  - sur une console
  - « graphiquement »
- GrAPic : Graphics for Algo/Prog in C/C++
  - Les formes de bases
  - Les couleurs
  - Les chaînes de caractères
  - Les images
  - Les évènements
- Et en TP
  - Comment l'installer
  - Créer un nouveau projet

# DESSINER DANS LA CONSOLE

- Jusque là pour dessiner, nous avons utilisé la console

Quelles sont les dimensions du rectangle : longueur ?	Résultat :
4	****
Largeur	****
3	****

Longueur du rectangle ?	Résultat :
6	*****
Largeur du rectangle	* *
4	* *
	*****

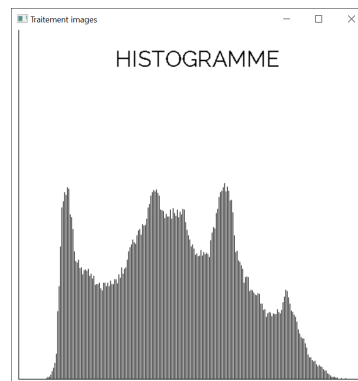
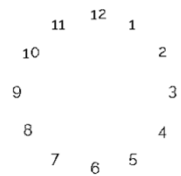
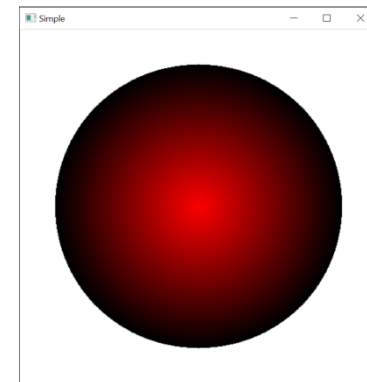
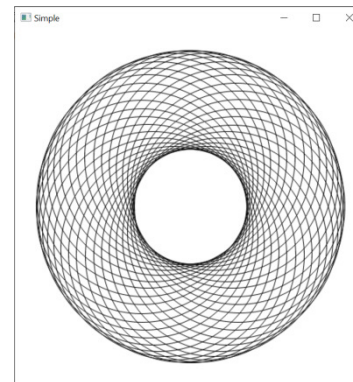
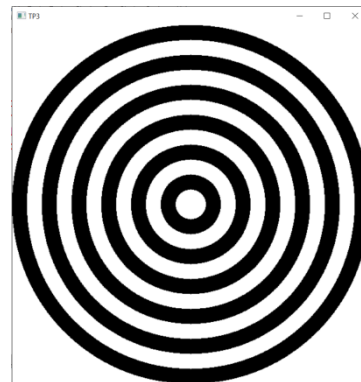
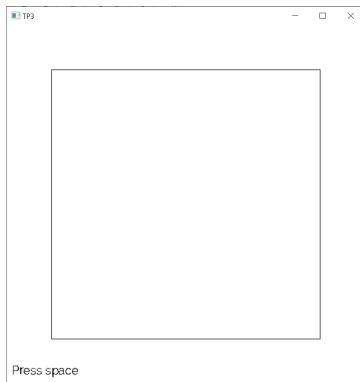
Quelle est la hauteur du triangle ?	Résultat :
4	*
	**
	***
	****

```
"C:\professionnel\Enseignement\LIF1\Automne 10-11\TP\TP8\demineur.exe"
3 - - 0 - - - - -
4 - - - 2 - - - - -
5 - - - - - 2 - - -
6 - - - - - - - - -
7 - 2 - - - - - - -
8 0 - - - - - - - -
9 0 1 - - - - - 1
Nombre de cases Ó trouver : 54
données les coordonnées de la case a sonder
8
2
1 2 3 4 5 6 7 8 9
1 1 - - 0 0 - - - -
2 - 1 0 0 0 0 - - -
3 - - 0 - - - - - -
4 - - - 2 - - - - -
5 - - - - - 2 - - -
6 - - - - - - - - -
7 - 2 - - - - - - -
8 0 1 - - - - - - -
9 0 1 - - - - - 1
Nombre de cases Ó trouver : 53
données les coordonnées de la case a sonder
```

## DESSINER DANS LA CONSOLE

- C'est contraignant
  - On commence toujours en haut à gauche
  - Il faut dessiner ligne par ligne
  - On ne change pas de couleur
  - On ne peut pas « lisser » les dessins
  - ...

# NOUS ON VOUDRAIT FAIRE



A window titled 'Minesweeper' showing a 10x10 grid. The grid contains numbers and mines. A 'PERDU' message is overlaid on the bottom left.

0	0	0	0	0	0	0	0	1	
1	1	1	0	0	0	0	0	1	
		1	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	0	1		1	0	0	0	0



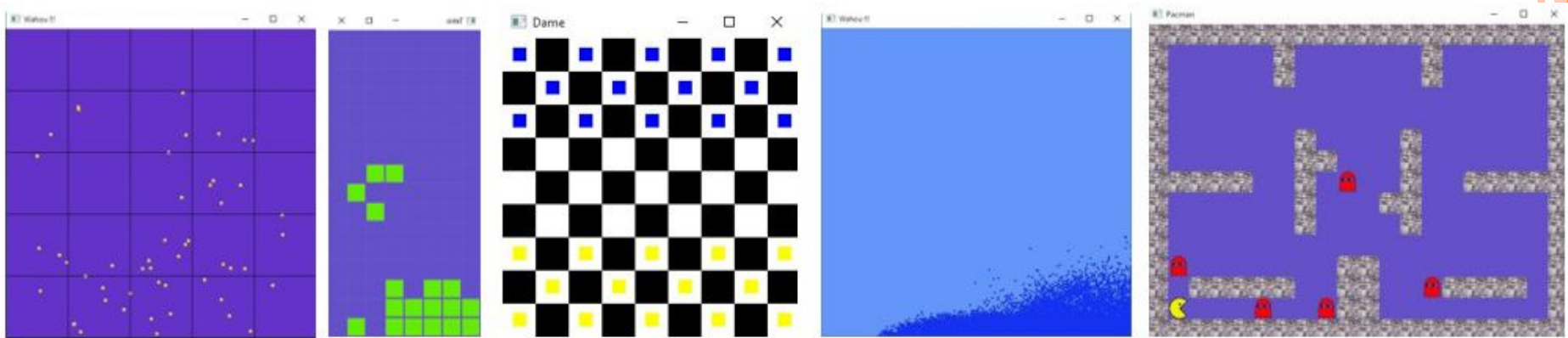
## POUR CELA IL FAUT

- Créer une fenêtre graphique
- Savoir dessiner dedans
- Savoir gérer les évènements clavier et souris
- ...
  
- Donc connaître / savoir programmer beaucoup de choses...  
ou utiliser une bibliothèque pour nous simplifier le travail !

# PLAN

- Qu'est-ce qu'une bibliothèque ?
- Dessiner
  - sur une console
  - « graphiquement »
- GrAPic : Graphics for Algo/Prog in C/C++
  - Les formes de bases
  - Les couleurs
  - Les chaînes de caractères
  - Les images
  - Les événements
- Et en TP
  - Comment l'installer
  - Créer un nouveau projet

# GRAPIC : GRAPHICS FOR ALGO/PROG IN C/C++



- Bibliothèque pour ajouter des fonctionnalités graphiques à votre code C / C ++
- Fournit des fonctions graphiques faciles à utiliser

<http://liris.cnrs.fr/alexandre.meyer/grapic/>

# CRÉER UNE FENETRE ET L' AFFICHER

```
int main(int , char** )
{
    bool stop=false;

    winInit("LIFAP1 TP3", 500, 500);
    backgroundColor( 100, 50, 200 );

    while( !stop )
    {
        winClear();
        draw();
        stop = winDisplay();
    }

    winQuit();
    return 0;
}
```

**winInit** : crée un fenêtre avec un nom et une taille donnée

**backgroundColor** : indique la couleur de cette fenêtre

**winClear** : vide l'affichage

**winDisplay** : rafraichit l'affichage

**winQuit** : ferme la fenêtre

# CRÉER UNE FENETRE ET L'AFFICHER

```
int main(int , char** )
{
    bool stop=false;

    winInit("LIFAP1 TP3", 500, 500);
    backgroundColor( 100, 50, 200 );

    while( !stop )
    {
        winClear();
        draw();
        stop = winDisplay();
    }

    winQuit();
    return 0;
}
```

## winDisplay :

- Rafraichit l'affichage
- Boucle TANTQUE pour attendre une action de l'utilisateur
- Si l'utilisateur clique sur la croix pour fermer de la fenêtre, alors la fonction renvoie 'stop'

## draw :

- Fonction dans laquelle nous allons mettre ce que nous voulons dessiner...

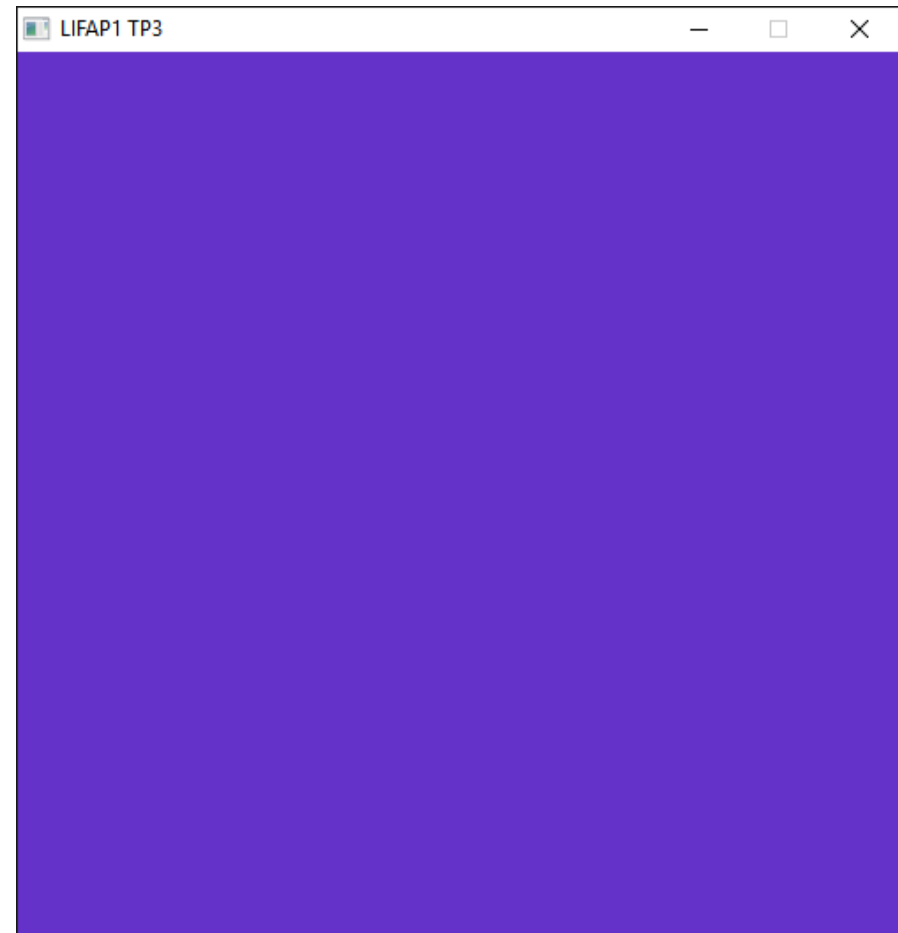
# CRÉER UNE FENETRE ET L'AFFICHER

```
int main(int , char** )
{
    bool stop=false;

    winInit("LIFAP1 TP3", 500, 500);
    backgroundColor( 100, 50, 200 );

    while( !stop )
    {
        winClear();
        draw();
        stop = winDisplay();
    }

    winQuit();
    return 0;
}
```



!!! coordonnées x=0 ; y=0 !!!

## LES FORMES DE BASES

- Une ligne entre le point (x1, y1) et le point (x2, y2)
  - `void line (int x1, int y1, int x2, int y2)`
- Un rectangle (vide ou plein) entre le point (xmin,ymin) et le point (xmax,ymax)
  - `void rectangle (int xmin, int ymin, int xmax, int ymax)`
  - `void rectangleFill (int xmin, int ymin, int xmax, int ymax)`
- Un cercle (vide ou plein) de centre (xc, yc) et de rayon r
  - `void circle (int xc, int yc, int r)`
  - `void circleFill (int xc, int yc, int r)`

## PAR EXEMPLE

```
// Si on définit la taille de  
// notre fenetre à 500*500
```

```
const int DIMW = 500;
```

```
// On peut dessiner ainsi :
```

```
void draw()
```

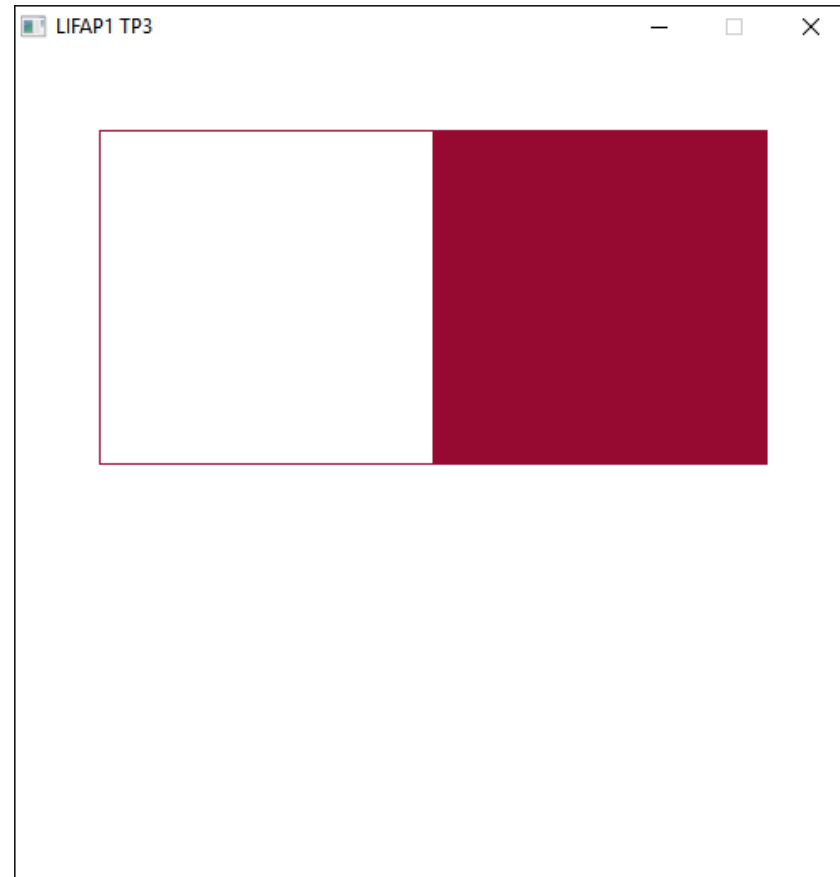
```
{
```

```
    color( 150, 10, 50);
```

```
    rectangle( DIMW/2 - 200, DIMW/2, DIMW/2 , DIMW/2 + 200 );
```

```
    rectangleFill( DIMW/2 , DIMW/2, DIMW/2 + 200 , DIMW/2 + 200 );
```

```
}
```





## LES COULEURS

- Une couleur est définie par ses composantes Rouge-Verte-Bleue
  - Chaque composante est un nombre entre 0 et 255
  - Par exemple :
    - Noir = 0 0 0
    - Blanc = 255 255 255
    - Rouge = 255 0 0
- La couleur du pinceau
  - `void color (int rouge, int vert, int bleu)`
- La couleur du fond de la fenêtre
  - `void backgroundColor (int rouge, int vert, int bleu)`

# Les chaînes de caractères

- Ecrire un texte commençant aux coordonnées (x,y)

- `print( int x, int y, const char *txt);`

- `print( 10, 20, "Bonjour");`

- `print( int x, int y, float nb);`

- `print( 10, 20, 3.14);`

- Définir la taille de la police d'écriture

- `void fontSize (int s)`

# LES IMAGES

```
struct Data{
    Image im;
};

void init(Data& d)
{
    d.im = image("data/grapic.bmp", true,
                255, 255, 255, 255);
}

void draw(Data& d)
{
    image_draw(d.im, 0, 255);
}
```

Elles sont stockées dans une structure particulière 'Data'

**image(...)** : permet de charger l'image se trouvant à l'endroit 'data/grapic.bmp'

**image\_draw(...)** : permet d'afficher l'image aux coordonnées (x, y)

# LES ÉVÈNEMENTS : LA SOURIS

- Savoir quand l'utilisateur a cliqué avec la souris
  - `bool isMousePressed (int button)`
  - Il faut préciser quel bouton on regarde :  
DL\_BUTTON\_LEFT / DL\_BUTTON\_RIGHT
- Connaître la position de la souris quand l'utilisateur clique
  - `void mousePos (int &x, int &y)`

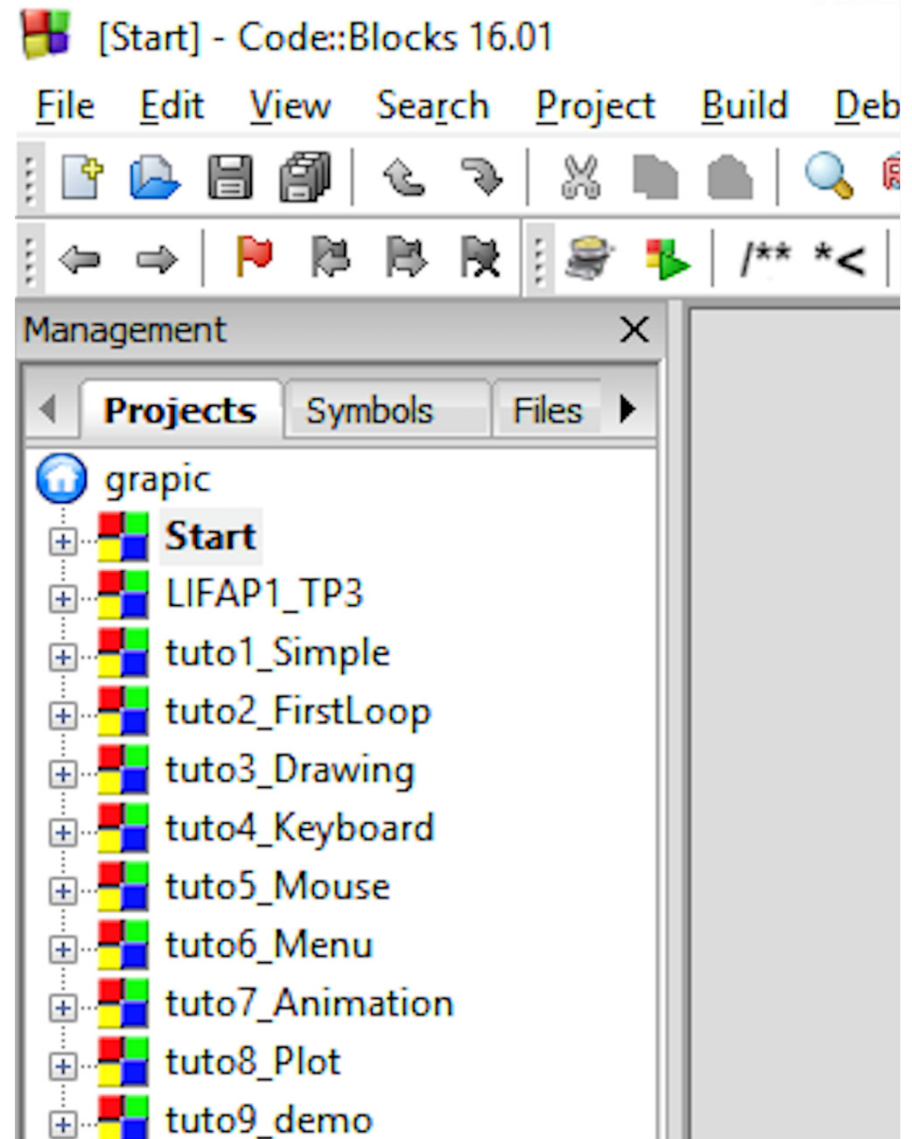
```
int x,y;
mousePos(x,y); // sauve la position de la souris
if ((isMousePressed(SDL_BUTTON_LEFT)) // si l'utilisateur a cliqué
    && (x>10) && (y>10) && (x<50) && (y<50)) // sur le bouton gauche
    // et que la souris se trouve
    // dans cet endroit de l'écran
{ ... } // alors ...
```

# PLAN

- Qu'est-ce qu'une bibliothèque ?
- Dessiner
  - sur une console
  - « graphiquement »
- GrAPic : Graphics for Algo/Prog in C/C++
  - Les formes de bases
  - Les couleurs
  - Les chaînes de caractères
  - Les images
  - Les événements
- Et en TP
  - Comment l'installer
  - Créer un nouveau projet

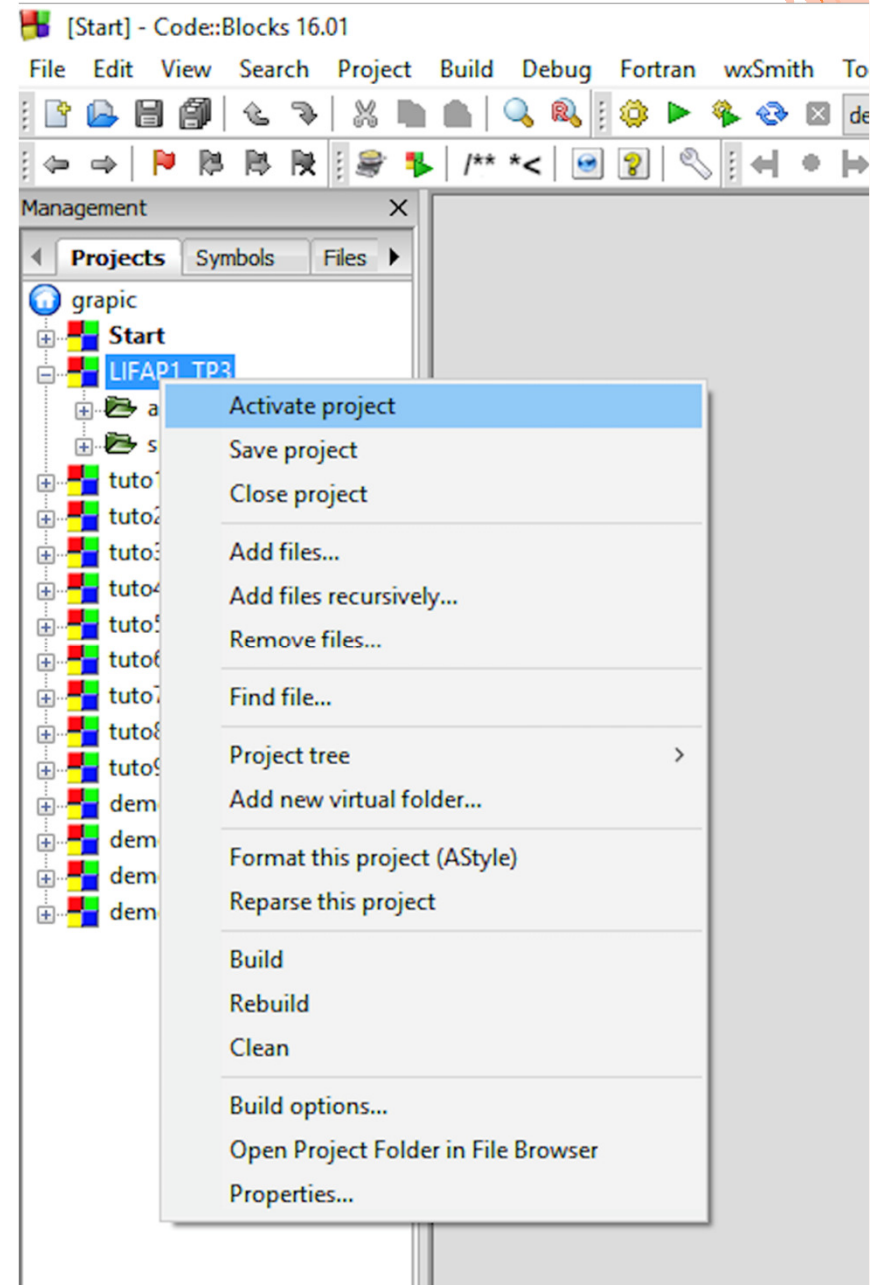
# COMMENT INSTALLER GRAPIC ?

- Si vous travaillez de chez vous, il faut installer CodeBlocks
- Version codeblocks-16.01mingw-setup.exe
- Télécharger GrAPic
  - <http://liris.cnrs.fr/alexandre.meyer/grapic/html/index.html>
- Le décompresser
- Ouvrir le fichier Grapic/build/windows/**grapic\_workspace** avec Codeblocks
- Vous avez accès à tous les projets de l'archive



## COMMENT ACTIVER UN PROJET

- Pour choisir quel projet compiler et exécuter, il faut l'activer
- Vous pouvez ouvrir et modifier plusieurs projet en même temps
- Mais un seul est actif et donc compilable / executable à la fois
- Les sources des projets
  - Dans **grapic/apps**
- Les executables des projets
  - Dans **grapic/bin**



# Comment créer un nouveau projet

- Sous Windows (en TP)
  - Copier le dossier “apps/start” dans un nouveau dossier “apps/MYPROJECT”
    - Remplacer MYPROJECT par le nom que vous voulez....
  - Renommer “apps/MYPROJECT/main\_start.cpp” en “main\_MYPROJECT.cpp”
  - Editer le fichier “Grapic/premake4.lua” et ajouter à la fin “make\_project( "MYPROJECT", "apps/MYPROJECT/main\_MYPROJECT.cpp" )”
  - Exécuter le script 'premake.bat'
- Sous Linux et Mac : regarder la documentation sur le site...





# COMMENT TROUVER UNE FONCTION

- Regarder la documentation
- Specifications des fonctions,
- Explication de leur fonctionnement, exemples d'utilisation



## GrAPiC

Graphics for Algo/Prog in C/C++

Main Page **Doc**

File List

- GrAPiC
  - Introduction
  - Download
  - ▶ How to start ?
  - ▶ Tutorials
  - ▶ Examples
  - Doc
    - File List
      - src
        - documentations.h
        - ▶ Grapic\_func.h

```
bool grapic::isMousePressed (int button)
return true if the mouse button 'button' is pressed

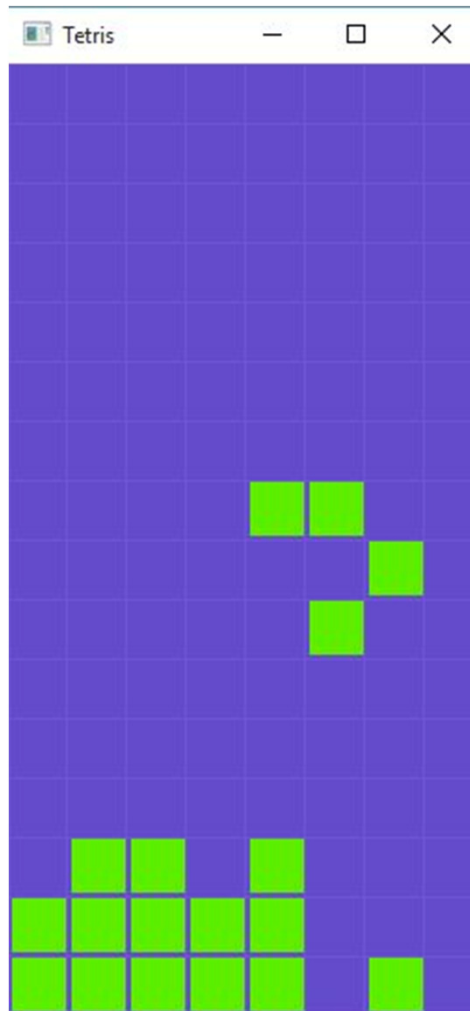
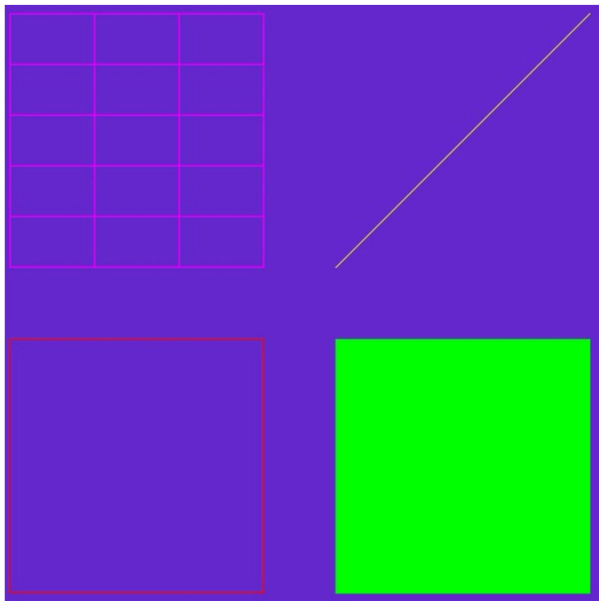
if (isMousePressed(SDL_BUTTON_LEFT)) { ... } // test if the left button of the mouse is pressed

void grapic::mousePos (int &x, int &y)
After this function (x,y) store the mouse position

int x,y;
mousePos(x,y);
if ((isMousePressed(SDL_BUTTON_LEFT)) && (x>10) && (y>10) && (x<50) && (y<50)) { ... } // if the user clicks if the square (10,10)(50,50)
```

# Comment s'entraîner...

- Faire les tutoriels

A screenshot of a website navigation menu. The menu is titled "Main Page" and "Doc". It contains a list of links: "GrAPiC", "Introduction", "Download", "How to start?", "Tutorials", "Tutorial 1: Simple program", "Tutorial 2: First Loop", "Tutorial 3: Drawing", "Tutorial 4: Keyboard", "Tutorial 5: Mouse", "Tutorial 6: Menu", "Tutorial 7: Animation", "Tutoria 8: Plot/Graph", "Tutoria 9: Demo", "Examples", "Tetris", "Pacman", and "Doc".

- ▼ GrAPiC
  - Introduction
  - Download
  - ▶ How to start ?
- ▼ Tutorials
  - Tutorial 1: Simple program
  - Tutorial 2: First Loop
  - Tutorial 3: Drawing
  - Tutorial 4: Keyboard
  - Tutorial 5: Mouse
  - Tutorial 6: Menu
  - Tutorial 7: Animation
  - Tutoria 8: Plot/Graph
  - Tutoria 9: Demo
- ▼ Examples
  - Tetris
  - Pacman
- ▶ Doc

A VOUS DE JOUER !

