

# TP1 - Utiliser SFML

M2103 - Programmation orientée objets

Département Informatique - IUT de Bordeaux - février 2016

**Objectifs** Pendant cette séance vous écrirez vos premiers programmes qui manipuleront des objets. Ceci vous familiarisera aussi avec

- les appels de la bibliothèque SFML.
- les éléments d'une application graphique : boucle d'interaction, traduction des événements en actions modifiant l'état, affichage à partir de l'état, ...)
- la documentation en ligne : <http://www.sfml-dev.org/documentation/2.0/>

**Note : Chargement d'un projet QT Creator depuis une archive :**

1. Extraire le répertoire du projet à l'endroit voulu
2. Vérifier qu'il n'y a pas de fichier avec le suffixe .user. Le supprimer au besoin
3. Sous QtCreator, ouvrir le fichier .pro.

**Note : Copier un projet QT Creator sous un autre nom :**

1. Copier le répertoire sous le nom voulu
2. Dans la copie, renommer le fichier .pro
3. Supprimer le fichier .user

**Note : Pour créer un nouveau projet SFML :**

1. lancez QT Creator
2. cliquez sur Nouveau Projet / Projet non QT / Projet C++
3. éditez le fichier .pro du projet pour y **ajouter les deux lignes**

```
QMAKE_CXXFLAGS += -std=c++11 -Wall -Wextra
LIBS            += -lsfml-graphics -lsfml-window -lsfml-system
```

## 1 Animation

Reprendre le programme de démonstration pour faire tourner le carré sur lui-même.

Vous trouverez l'archive dans ~/Bibliotheque/M2103-P00/TP1/ExempleRondCarre.zip

- ajoutez une variable `angle` qui sera incrémentée à chaque tour de la boucle principale, entre le traitement des événements et le réaffichage ;
- servez-vous en pour définir la rotation du rectangle avant son affichage

**Indication** : `carre.setRotation(angle); // en degrés`

## 2 Anneaux olympiques

Écrire un programme qui dessine les 5 anneaux olympiques

1. Commencez par faire tracer un anneau : cercle transparent avec un entourage de la couleur voulue. Voir la doc pour `setFillColor`, `setOutlineColor`, `setOutlineThickness` et `sf::Color::Transparent`.
2. Au besoin, cherchez sur Internet l'ordre des couleurs. Et faites un dessin sur le papier pour fixer, au moins approximativement, les coordonnées de ce que vous voulez dessiner.

### 3 Damier

1. Afficher un damier 10 x 10 (cases noires et blanches, fond gris). Remarque : vous n'utiliserez que deux formes, que vous dessinerez plusieurs fois.
2. Ajoutez un rond rouge (au départ en haut à gauche) qui bougera de case en case (flèches `sf::Keyboard::Up` etc.). Vous utiliserez deux variables pour les numéros de ligne et colonne (entre 0 et 9).
3. Amélioration esthétique (en fin de séance si il vous reste du temps) ajoutez des coordonnées (de A, B, C ... horizontalement, de 1, 2, 3... verticalement en partant du bas).

### 4 Pendule

Faire afficher une pendule, avec aiguilles des heures, des minutes et des secondes.

- Les aiguilles sont 3 rectangles de tailles et couleurs différentes.
- Les fonctions système `time` et `localtime` permettent de récupérer le temps courant, et le décomposer dans une structure avec l'heure, les minutes, les secondes mais aussi le jour, le mois, l'année etc. Voir exemple plus loin

#### Feuille de route

1. Afficher l'heure dans un coin, sous forme de texte, permet de vérifier qu'on a bien réussi à la récupérer.
2. Faites ensuite dessiner l'aiguille des secondes. Il est plus facile de vérifier qu'elle tourne dans le bon sens...
3. Les aiguilles des heures et minutes ne devraient pas poser de problème ensuite

#### Récupération du temps, exemple

```
#include <iostream>
#include <ctime>

using namespace std;

int main(int argc, char **argv)
{
    time_t t = time(NULL);
    struct tm *temps = localtime( & t);

    cout << "Il est "      << temps->tm_hour
         << " heures et " << temps->tm_min
         << " minutes."   << endl;

    return 0;
}
```