

Découverte de Processing

Résumé

Le but ici est de "prendre en main" l'environnement de développement `Processing` pour la programmation.

`Processing` est une plateforme de développement pour `Java` simple à utiliser. Il en existe d'autres, la plus célèbre étant `eclipse` : vous l'utiliserez plus tard dans ce semestre. Pendant les deux premières semaines, vous allez découvrir les fondements de la programmation : nous utiliserons `Processing` pendant cette période. L'objet de cette séance est simplement de s'assurer que `Processing` est bien opérationnel.

Exercice 1 : installation et test opérationnel)

Télécharger la version de `Processing` pour votre système : <https://processing.org/download/>.

Lancer l'application : un onglet vide apparaît. Il s'agit de l'onglet destiné à contenir le code de votre projet. Choisir comme nom de projet : "Exercice1" (menu : Fichier/Enregistrer sous).

Ecrire cette ligne :

```
println("Cela marche !");
```

Appuyer sur le bouton "play" : une fenêtre graphique (vide) doit apparaître et le message "Cela marche" doit s'inscrire dans le terminal (bandeau noir du bas).

Exercice 2 : découverte de l'interface

Il est maintenant temps de regarder de plus près l'environnement de développement. Pour cela, suivez les instructions de ce tutoriel (qui porte sur une version antérieure) :

<https://fr.flossmanuals.net/processing/les-bases-de-processing/>
(uniquement la partie : "Les bases de processing")

Exercice 3 : quelques compléments (auto-complétion, débogueur)

Par défaut, une fonctionnalité pourtant fort pratique n'est pas activée : l'auto-complétion des commandes avec "ctrl-espace". Dans le menu "Fichier/Préférences", cocher la case "auto-complétion du code".

Tester ensuite son fonctionnement en refaisant la saisie d'un des exemples de l'exercice précédent. Par la suite, employez-vous à utiliser de manière systématique l'auto-complétion : le raccourci "ctrl-espace" est utilisé par la plupart des environnements de développement (NetBeans, eclipse ...).

Nous allons maintenant utiliser le débogueur : activez-le en appuyant sur le symbole de l'insecte. Une nouvelle fenêtre "Variables" doit apparaître.

Ecrire le programme suivant :

```
println("cela marche");  
println("cela continue a marcher");  
println("il va falloir penser a arreter maintenant");  
println("on s'arrete");
```

1. cliquer sur le numéro de ligne "2" : un losange le remplace. Vous venez de placer un "point d'arrêt" (breakpoint) à cet endroit du programme. Lancer le programme et observer en regardant le contenu du terminal que celui-ci s'est arrêté (en fait mis en pause) juste avant celle-ci.
2. appuyer sur "continuer". Le programme reprend son exécution (jusqu'au point d'arrêt suivant, mais il n'y en a pas). Vérifier que le terminal contient maintenant tous les affichages.
3. démarrer à nouveau le programme. Lorsqu'il est en pause sur la ligne "2". Reprendre l'exécution en utilisant "step". Le programme exécute une instruction puis se remet en pause. C'est un moyen très commode de faire une exécution pas-à-pas.
4. enlever le point d'arrêt en ligne 2 et en mettre en lignes 1 et 3. Refaites ensuite des exécutions contrôlées par le débogueur comme ci-dessus.

Intéressons-nous maintenant à la fenêtre "Variables". Celle-ci affiche l'état des variables du programme (cf prochaine séance). Processing utilise un certain nombre de variables dites "globales" (i.e. qui existent tout au long de l'exécution du programme). Déplier l'item Processing de la fenêtre "Variables" pour voir l'état de celles-ci.

Voici ce qu'elles signifient :

- `width`, `height` : dimensions de la fenêtre graphique ;
- `mouseX`, `mouseY` : coordonnées du pointeur de souris ;
- `pmouseX`, `pmouseY` : coordonnées antérieures du pointeur de souris ;
- `key`, `keyCode` : touche appuyée ;
- `keyPressed` : indique si une touche est appuyée ;
- `focused` : indique si la fenêtre graphique a le focus ;
- `frameRate` : nombre d'images par seconde ;
- `frameCount` : nombre d'images qui ont été affichées.