

1. introduction

Processing (<http://processing.org>) est un langage (une **surcouche** du langage Java en fait) développé par Ben Fry et Casey Reas (MIT) dédié à la programmation créative. Il permet entre autres de créer des images (fixes ou animées) et interagir avec elles. Processing permet aussi de manipuler, générer des textes, des formes vectorielles, des images en trois dimensions (en utilisant OpenGL), du son, ... et plus généralement tout !

Processing peut enfin être étendu à l'aide de bibliothèques ou grâce à des projets « *proches* » comme Arduino (<http://www.arduino.cc>) et Wiring (<http://wiring.org.co>) pour la gestion de capteurs

Il existe de multiples versions de Processing dont « p5.js » (<http://p5js.org>), « processing JS » (<http://processingjs.org>) écrits en javascript, Processing Python (<http://py.processing.org>).



Figure 1 : Exemple de sketch Processing

Processing s'appuie sur le langage Java, ce qui lui permet de fonctionner sur les systèmes d'exploitation Windows, MacOS X et Linux (dont Raspbian (Raspberry Pi) et Android avec le mode idoine proposé avec Processing 3). Ce langage est une alternative gratuite et open-source à des logiciels auteur tels qu'Adobe Flash et Microsoft Silverlight. Malgré certaines lacunes qui en font un logiciel de production inadapté à de nombreux cas, Processing a de nombreux atouts dont notamment sa simplicité d'usage.

2. installer Processing

Nota : Supposons que Processing 3.3.2 soit installé à sur le répertoire **C:/langages**

Si ce n'était pas le cas, la première chose à faire est de se rendre à l'adresse <https://www.processing.org/download/?processing> et télécharger une des versions proposées (version stable 3.3.2 du 25 avril 2017 en 32 ou 64 bits ou version précédente 2.2.1 en 32 ou 64 bits du 19 mai 2014).

Une fois téléchargée, décompressez l'archive dans le répertoire de votre choix et lancez l'exécutable « *processing* ».

La première chose à faire est de déterminer le répertoire où seront sauvés vos programmes (cf. *File | Preferences*). Choisissez l'emplacement dans le champ « *sketchbook location* » et appuyez sur ok (cf. Figure 2).

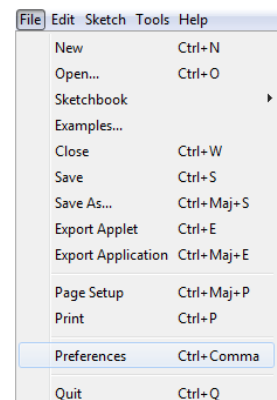


Figure 2 : Menu

3. mon premier sketch

Il est possible d'utiliser Processing de différentes manières dont notamment en mode « *script* », et en mode « *continu* ».

Le mode « *continu* » demande d'implémenter au moins deux fonctions : **setup()** qui initialise les variables et **draw()**, boucle d'affichage de données. Cette boucle permet d'afficher des animations graphiques complexes, réagir des événements synchrones provenant d'actions de l'utilisateur ou d'événements systèmes.

Par convention, les mots réservés du langage sont affichés en *bleu*, *vert* et *orange* dans l'IDE (cf. Figure 3).

```

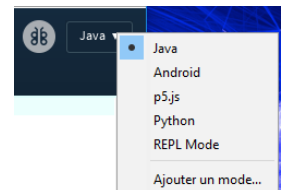
primitives_graphiques ▼
1 /**
2 Utiliser des primitives graphiques dans un script Processing
3
4 */
5
6 void setup() { // Initialiser le sketch
7   size(400,400); // créer une fenêtre de taille 800x600
8
9 }
10
11 void draw() { // boucle infinie de dessin
12   background(0); // redessiner la fenêtre en noir - 0 -> noir, 255 -> blanc
13   noStroke(); // ne pas dessiner de contour pour les objets
14
15   // dessiner une ellipse verte
16   fill(0,255,0); // vert (Red/Green/Blue)
17   ellipse(200,250, 100,100);
18
19   // dessiner un rectangle bleu
20   fill(0,0,255);
21   rect(50,50,100,150);
22
23   // dessiner un triangle rouge (avec transparence)
24   fill(255,0,0,120);
25   triangle(20,20, 300,150, 80, 300);
26 }
27
28

```

Figure 3 : mon premier sketch

Les possibilités du langage Processing sont quasi-infinies notamment avec la possibilité d'utiliser la programmation orientée-objet, d'ajouter des bibliothèques externes et d'en écrire soi-même ! Les quelques exercices ci-après vous donneront un aperçu de ce qui peut être fait en quelques lignes de code.

L'IDE Processing propose plusieurs modes : Java (par défaut) mais aussi Android (ADB doit être installé), p5.js, Python et REPL (Read Eval Print Loop soit sous forme de « *shell* »). Il suffit de choisir le mode (une installation peut être requise) que vous souhaitez sur le bouton à droite de l'IDE.



4. exercices

4.1 des dessins

- Créer une composition graphique en couleur (avec composante de transparence) utilisant au moins un cercle, un triangle et un rectangle

4.2 utiliser les entrées

- Utiliser une touche du clavier (ESPACE par exemple) pour modifier la couleur de remplissage d'un triangle affiché dans la fenêtre
- Dessiner deux formes qui réagissent différemment suivant les actions de la souris (sur l'événement *Press* et *Release* par exemple)

4.3 gérer du texte

- Afficher votre citation favorite avec votre police de caractères préférée
- Utiliser deux polices de caractères différentes pour simuler un dialogue entre 2 utilisateurs (phrases justifiées à gauche pour l'un et à droite pour l'autre)

4.4 gérer des images


- Afficher deux images avec une teinte différente
- Charger un fichier **png** avec un bit de transparence et créer une composition graphique en superposant les couches

4.5. capteurs

4.5.1 webcam

Installer au préalable la librairie externe *Video* (Menu **Outils** | **Ajouter un outil**, onglet **Libraries**)

✔ **Video** | GStreamer-based video library for Processing.

 **The Processing Foundation**

A partir des exemples, écrire un programme qui permet de démarrer la caméra en appuyant sur la touche « *espace* » et l'éteindre de la même manière (mettre à la place une photo de chat 😊)

4.5.2 interaction tangible à partir de la webcam

A partir du projet **PTopCode** (équivalent « *light* » à ARToolkit) ,
<http://www.irit.fr/~Philippe.Truillet/ens/ens/processing/PTopCode.zip>, faire bouger à droite et à gauche un objet affiché à l'écran en présentant un code TopCode.

4.5.3 Arduino

Utiliser au préalable l'IDE Arduino (<http://www.arduino.cc>) sur votre machine. Nous supposons que l'IDE est installé préalablement sur C : / langages

Télécharger l'exemple ici :

<http://www.irit.fr/~Philippe.Truillet/ens/ens/processing/arduino.zip>

Compiler et téléverser le code **capteur.ino** sur le module arduino branché sur le port série. Brancher une led infrarouge sur le pin Analogique **A0** et **GND**. (le code va lire la valeur du capteur et l'écrire sur le port série)

Exécuter le code Processing.

Modifier le code de telle manière que la valeur du capteur récupérée soit affichée sous forme de barre verticale de 300 pixels maximum (si la valeur récupérée est « *1023*»)

4.5 classes objet

Télécharger le projet **Gestion_Objets**

(http://www.irit.fr/~Philippe.Truillet/ens/ens/processing/Gestion_Objets.zip) et modifier le code de telle manière que l'objet change de couleur quand on clique dessus et revienne à sa couleur initiale quand on le relâche (penser à utiliser les événements *MouseDragged()*, *MousePressed()* et *MouseReleased()*).

Composer maintenant une liste de 3 d'objets gérés de la même manière (utiliser une *ArrayList* pour ce faire).

5. adresses utiles

- **Processing** : <http://www.processing.org>
- **P5.js** : <http://p5js.org>
- **studioSketchpad** : sri-upssitech-p5.sketchpad.cc

- **Référence** : <http://processing.org/reference>
- **Learning Processing** : <http://www.learningprocessing.com>
- **Hello Processing** : <http://hello.processing.org>

- **Librairies** : <https://processing.org/reference/libraries/>