

TP 1 – Initiation à Processing.org Ph. Truillet Novembre 2019 – v. 1.0



1. introduction

Processing (<http://processing.org>) est un langage (en réalité, une **surcouche** du langage Java) développé par Ben Fry et Casey Reas (MIT) et dédié à la programmation créative.

Il permet entre autres de manipuler des images (fixes ou animées) et interagir avec elles. Processing permet aussi de générer des textes, des formes vectorielles, des images en trois dimensions (en utilisant OpenGL), du son et plus généralement ... **tout ce que vous souhaitez** !

Processing peut aussi être étendu à l'aide de bibliothèques (fichiers JAR – **J**ava **A**rchive) ou grâce à des projets « proches » comme Arduino (projet directement dérivé de Processing) (<http://www.arduino.cc>) et Wiring (<http://wiring.org.co>) pour la gestion de capteurs physiques.

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

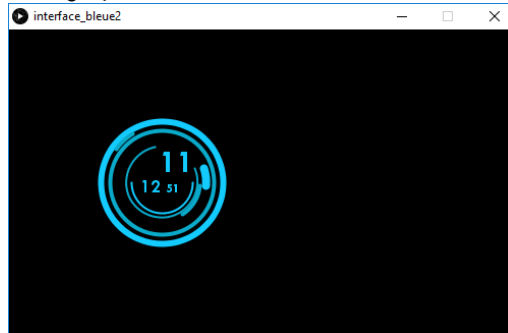


Figure 1 : Exemple de sketch écrit en Processing

Processing dans sa version originale s'appuie sur le langage Java, ce qui lui permet de fonctionner sur les systèmes d'exploitation Windows, MacOS X et Linux (et donc aussi sur Raspbian pour Raspberry Pi) et **Android** avec le mode idoine proposé avec Processing 3).

Malgré encore quelques petites lacunes (de plus en plus corrigées), Processing a de nombreux atouts dont notamment sa simplicité d'usage qui en fait un langage presque idéal pour le prototypage rapide !

2. installer Processing

Nota : Nous supposons dans la suite du document que l'IDE (Integrated Development Environment) Processing 3.5.3 est installé dans 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 3.5.3 du 3 février 2019 en 32 ou 64 bits).

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

Déterminer ensuite le répertoire où seront sauves vos programmes (*sketches*) (cf. *File* | *Preferences*), choisissez l'emplacement dans le champ « *sketchbook location* » et appuyez sur ok (cf. Figure 2).

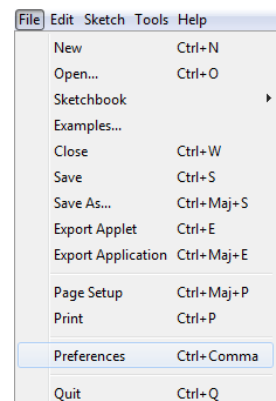


Figure 2 : Menu

Nota : Cette étape est primordiale pour la suite ; c'est dans ce répertoire que toutes bibliothèques externes seront téléchargées et vos « sketches » sauves.

3. mon premier sketch

Il est possible d'utiliser Processing de différentes manières dont notamment en mode « *script* » (suites d'actions exécutées qu'une seule fois), et en mode « *continu* » (avec une boucle infinie – fonction **draw**).

Le mode « *continu* » demande d'implémenter au moins deux fonctions : **setup()** qui initialise les variables (depuis Processing, on peut en plus utiliser la fonction **settings()**) et **draw()**, boucle d'affichage de données (*mainLoop*). Cette boucle permet d'afficher des animations graphiques complexes, réagir à des événements asynchrones 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).

Pour ce premier sketch, recopiez le code ci-après (vous trouverez le fichier source ici →

https://github.com/truillet/upssitech/blob/master/SRI/1A/Code/primitives_graphiques.zip) et lancez le script.

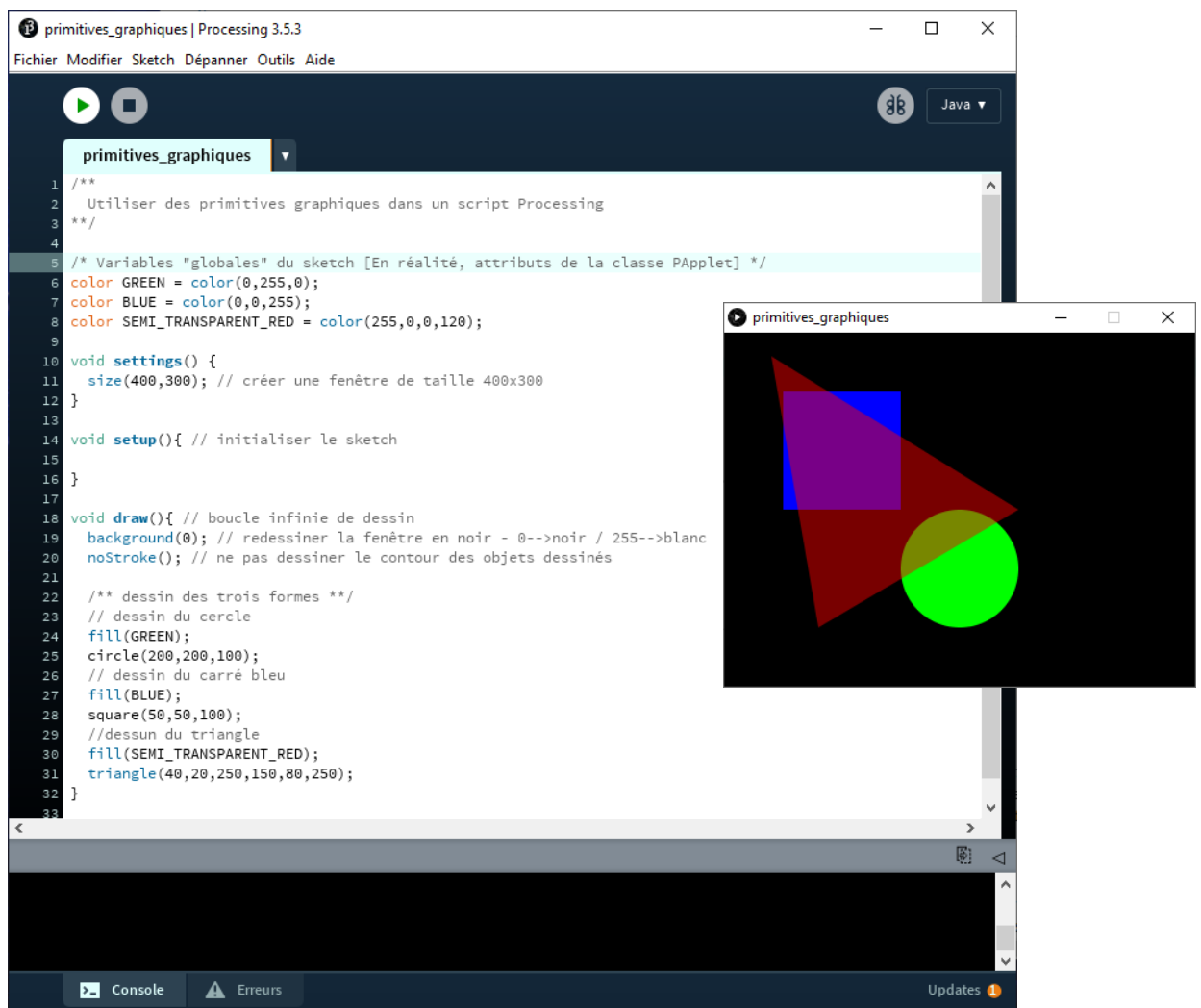
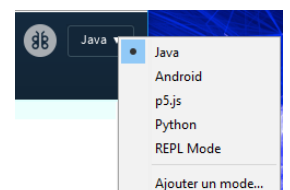


Figure 3 : mon premier sketch

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 !

Enfin, l'IDE Processing propose plusieurs modes : Java (par défaut) mais aussi Android (ADB doit être installé), p5.js, Python, R 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.



Les premiers exercices ci-après vous donnerons un aperçu de ce qui peut être fait en quelques lignes de code.

4. formes & images

4.1 manipuler des formes

- Créer une nouvelle composition graphique en couleur (incluant de la transparence) utilisant au moins un morceau de cercle (arc), une ligne et un quadrilatère.
- Composer un tableau simplifié de type « Mondrian » (père du néo-plasticisme) ou Sophie Taeuber-Arp

Voir des exemples ici : <https://www.wikiart.org/en/piet-mondrian>
et là : <https://www.wikiart.org/en/sophie-taeuber-arp>

4.2 manipuler des images

- Afficher deux images ayant une teinte différente.
- Charger un fichier **png** avec un bit de transparence et créer une composition graphique en superposant les couches.
- Faire défiler de droite à gauche votre composition graphique.

4.3 utiliser de la typographie

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

5. un peu d'interaction

5.1 utiliser les entrées « classiques »

- Utiliser les flèches du clavier (droite, gauche, haut et bas) pour modifier la position d'un triangle affiché dans la fenêtre
- Dessiner un personnage qui réagit suivant les actions de la souris.

5.2 manipuler des films vidéo

Télécharger une vidéo sur Youtube au format mp4 (vous pouvez utiliser « vlc portable » ou par exemple le site <https://youtube-mp4.download/fr> pour la sauver)

Charger le projet **videos**

(<https://github.com/truillet/upssitech/blob/master/SRI/1A/Code/videos.zip>) et modifier le code de telle manière que vous puissiez gérer le rembobinage, la pause et le démarrage de la vidéo avec les touches du clavier.

6. ... et de mise en réseau

6.1 un flux RSS

Charger le projet **RSS** (<https://github.com/truillet/upssitech/blob/master/SRI/1A/Code/rss.zip>), l'installer et l'ouvrir.

Exécuter le code.

Modifier le code de telle manière à télécharger le flux RSS (**R**eally **S**imple **I**ndication) du journal « le Monde » et afficher les « Unes » dans des cercles quand l'utilisateur clique sur le titre du journal.

7 capturer l'instant ...

7.1 webcam

Installer au préalable la librairie **Video** disponible dans le menu *Outils* | *Ajouter un outil...* puis onglet *Libraries*.

Télécharger le projet webcam

(<https://github.com/truillet/upssitech/blob/master/SRI/1A/Code/webcam.zip>)

Modifier le projet de telle manière qu'en appuyant sur la barre espace, on puisse modifier les effets à appliquer sur le flux vidéo (on souhaite notamment avoir comme effet : mode niveaux de gris et inverse vidéo)

7.2 “je suis ... ton père ?!”

Installer au préalable la librairie **OpenCV** (<https://github.com/atduskgreg/opencv-processing/releases>) disponible dans le menu *Outils* | *Ajouter un outil...* puis onglet *Libraries*.

Modifier l'exemple *LiveCam* fourni par **OpenCV** et remplacer chaque figure détectée par une webcam par le masque de Dark Vador (https://github.com/truillet/upssitech/blob/master/SRI/1A/Code/darth_vader.png)

8. adresses utiles

- **Processing** : <http://www.processing.org>
- **P5.js** : <http://p5js.org>
- **Référence** : <http://processing.org/reference>
- **Learning Processing** : <http://www.learningprocessing.com>
- **Hello Processing** : <http://hello.processing.org>
- **Support de cours** :
https://github.com/truillet/upssitech/blob/master/SRI/1A/Cours/C_processing.org_2.2.pdf
- **Librairies** : <https://processing.org/reference/libraries/>