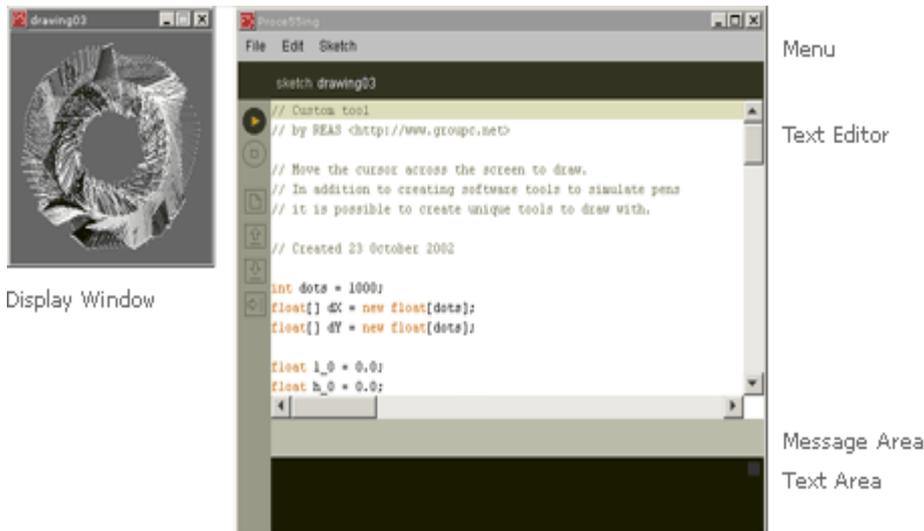


Environnement de Programmation Processing.

Nous avons conçu un éditeur simple et utilisable pour écrire et exécuter des programmes. L'environnement Processing (Integrated Development Environment ou IDE) dispose d'un éditeur de texte et d'une fenêtre pour visualiser les programmes. Quand on clique sur le bouton « RUN » le programme est compilé, la fenêtre de visualisation s'ouvre et le logiciel s'exécute à l'intérieur. La barre d'outils permet d'accéder aux fonctions d'exécution et d'arrêt d'un programme, de création, d'ouverture, de sauvegarde et d'exportation d'un nouveau croquis. D'autres commandes sont accessibles à partir des menus. La barre de messages informe de l'enregistrement ou du chargement et indique la nature et la position des erreurs lors de la compilation. La zone de débogage affiche des messages et peut recevoir du texte grâce aux instructions `print()` et `println()`.



La barre d'outils

La barre d'outils permet d'accéder aux six commandes de base de Processing : Run, Stop, New, Open, Save, Export.



Run : exécute le code (compile le code, ouvre la fenêtre de visualisation et exécute le programme à l'intérieur)



Stop : arrête l'exécution d'un programme, mais ne ferme pas la fenêtre de visualisation.



New : Crée un nouveau croquis. Dans Processing, les projets sont appelés « études » ou « croquis » (sketches en anglais).



Open : Sélectionne et charge un croquis existant. Un menu déroulant s'ouvre et vous pouvez choisir à partir de votre sketchbook « carnet de croquis », des exemples, ou encore vous pouvez ouvrir un projet situé n'importe où dans votre ordinateur ou sur le réseau.



Save : Enregistre le croquis courant dans le carnet de croquis de Processing (Sketchbook). Si vous voulez donner un autre nom au croquis, vous pouvez choisir « Save As » (enregistrer sous) du menu « File ».



Export : exporte le croquis courant dans le carnet de croquis comme Applet JAVA intégré dans une page HTML. Le dossier contenant le fichier est ouvert. Ouvrez le fichier `index.html` dans votre navigateur pour exécuter l'Applet. Il y a plus d'informations concernant la fonction d'exportation plus bas.

Menus

Des commandes additionnelles se trouvent dans les 5 menus « File », « Edit », « Sketch », « Tools » et « Help ». Ces menus sont contextuels, ce qui signifie que seuls les éléments utiles pour le projet en cours sont accessibles.

File (Fichier)

New (Ctrl+N) (Nouveau)

Crée un nouveau projet, dont le nom est la date courante suivi d'un indice (une lettre) : « sketch_AAAMMJJa » puis « sketch_AAAMMJJb », etc.

Open (Ctrl+O) (Charger)

Permet d'ouvrir un projet situé n'importe où dans l'ordinateur ou sur le réseau, dans le carnet de croquis, ou encore un exemple.

Save (Ctrl+S) (Enregistrer)

Enregistre le croquis en cours dans son état actuel.

Save as... (Ctrl+Shift+S) (Enregistrer sous)

Enregistre le croquis en cours avec la possibilité de lui donner un autre nom. Ainsi la version précédente du croquis n'est pas écrasée.

Export (Ctrl+E) (Exporter)

Par défaut, exporte le croquis courant dans le carnet de croquis comme Applet JAVA intégré dans une page HTML. Le dossier contenant le fichier est ouvert. Ouvrez le fichier index.html dans votre navigateur pour exécuter l'Applet. Il y a plus d'informations concernant la fonction d'exportation plus bas.

Export Application (Ctrl+Shift+E) (Exporter l'application)

Exporte comme application JAVA indépendante. Ouvre le répertoire contenant les fichiers exportés. (commande encore en développement)

Preferences (Accéder aux Préférences de Processing)

Vous permet de changer la manière dont processing fonctionne. (Pour l'instant non intégrée complètement)

Quit (Ctrl+Q) (Quitter)

Quitte l'environnement Processing et clos toutes les fenêtres

Edit (Édition)

Le menu Edit fournit un ensemble de commandes pour l'édition des fichiers Processing.

Undo (Ctrl+Z) (Annuler)

Annule la dernière commande ou la dernière modification. Pour revenir à l'état précédent, choisir « Redo » (répéter).

Redo (Ctrl+Y) (répéter)

Revient à l'état avant la dernière commande « Undo ». cette commande n'est disponible que dans le cas où la commande « Undo » a déjà été utilisée.

Cut (Ctrl+X) (Couper)

Supprime et copie la sélection vers le Presse-Papier (Une mémoire de texte).

Copy (Ctrl+C) (Copier)

Copie la sélection vers le Presse-Papier.

Paste (Ctrl+V) (Coller)

Insère le contenu du Presse-Papier à la position du curseur et remplace la sélection courante.

Select All (Ctrl+A) (Tout sélectionner)

Sélectionne tout le texte du fichier ouvert.

Find (Ctrl+F) (Rechercher)

Recherche les occurrences d'une chaîne de caractères dans le texte du fichier ouvert et propose de le remplacer par un texte différent.

Find Next (Ctrl+G) (Poursuivre la Recherche)

Cherche la prochaine occurrence de la même chaîne de caractères.

Sketch(Croquis)

Run (Ctrl+R)(Exécuter)

Exécute le code (compile, ouvre la fenêtre de visualisation et exécute le programme à l'intérieur).

Present (Ctrl+Shift+R)(Présenter)

Exécute le code au milieu de l'écran avec un fond neutre. Cliquez sur le bouton Stop en bas à gauche pour quitter le mode présentation.

Stop(Arrêter)

Si le programme est en train d'être exécuté, arrête l'exécution. Les programmes écrits en mode Basique ou qui utilisent la structure draw() sont arrêtés juste après avoir dessiné.

Add File(Ajouter un Fichier)

Ouvre une fenêtre pour ajouter une image, une police de caractères ou tout autre fichier dans le dossier « data » du répertoire du croquis en cours.

Import Library(Importer une bibliothèque)

Ajoute la ligne de commande 'import' nécessaire au début du fichier. Par exemple, en choisissant Sketch >> Import Library >> video ajoute la ligne "import processing.lib.net.*;" au début du fichier en cours. Ces lignes de commandes sont nécessaires pour utiliser une bibliothèque spécifique. (Cette fonction n'est pas intégrée avant la version 70 de Processing)

Show Sketch Folder (Voir le dossier du Croquis)

Ouvre le répertoire du croquis courant.

Tools (Outils)(Ce menu sera disponible à partir de Processing version 70)

Auto Format (Formatage automatique)

Tente de formater le code dans une mise en page plus facile à lire. Cette fonction était appelée « Beautify ».

Create Font... (Créer la Police de Caractères)

Convertit une police de caractères dans le format de Polices de processing et l'ajoute au croquis courant. Ouvre une boîte de dialogue qui propose de paramétrer la police, sa taille, si elle est anti-aliased ou non, et si tous les caractères doivent être générés. Si l'option « All Characters »(Tous les caractères) est cochée, les caractères non-anglais comme ü et Å et les accents sont générés mais la police augmente alors en taille. La quantité de mémoire nécessaire à la police est aussi déterminée par la taille en pixels choisie. Les polices Processing sont des textures, donc des polices plus grandes nécessitent plus de données-image.

Help (Aide)

Reference (Documentation)

Ouvre la documentation dans le Navigateur Web par défaut. Cette page inclut la documentation du langage, de l'environnement de programmation (le document que vous êtes en train de lire), des bibliothèques courantes et la comparaison avec d'autres langages.

Find in Reference (Ctrl+Shift+F)(Trouver dans la Documentation)

Sélectionner une commande dans votre programme et cliquez sur "Find in Reference"pour ouvrir la page de documentation la concernant.

Visit Processing.org (Ctrl+5)(Visitez Processing.org)

Ouvre la page de garde du site Processing.org dans le Navigateur Web par défaut.

About Processing (À propos de Processing)

Ouvre une page regroupant les informations minimum sur le logiciel Processing.

Sketchbook (le Carnet de Croquis)

Les projets dans Processing sont appelés des « sketches » (Croquis). Chaque croquis possède son propre répertoire (dossier) à l'intérieur duquel se trouve le fichier principal du programme qui porte le même nom que le croquis. Par exemple si le nom du croquis est « Sketch_123 », le répertoire du croquis sera nommé « Sketch_123 » et le fichier principal « Sketch_123.pde »

Les Croquis nécessitent d'autres dossiers qui contiennent les fichiers supplémentaires (images...) ou des bibliothèques. Quand une police de caractère ou une image est ajoutée au croquis, avec la commande « Add File... » du menu « Sketch », un dossier « data » est créé. Toutes les images, les polices, et autres fichiers de données/média doivent figurer dans ce dossier. Les Bibliothèques additionnelles doivent se trouver dans un dossier nommé « code ». Quand un Croquis est exporté, tous les fichiers situés dans les dossiers « data » et « code » sont exportés avec lui au sein d'un seul et même fichier avec l'extension « .jar » portant le même nom que le croquis. Par exemple si le croquis s'appelle « Sketch_123 », le fichier exporté s'appellera « Sketch_123.jar ».

Les Croquis sont tous stockés dans le répertoire de Processing, qui peut être situé dans différents endroits de votre ordinateur ou du réseau selon que vous utilisez Windows, Linux ou Mac OSX, et selon vos Préférences. Pour déterminer ce répertoire, éditez les préférences de Processing.

Exporting (Exportation)

L'exportation crée une version du programme qui peut s'exécuter dans un navigateur Web. Quand le code est exporté à partir de l'éditeur, il est transformé en code JAVA et ensuite compilé vers un Applet JAVA. Quand un projet est exporté, plusieurs fichiers sont écrits dans un dossier nommé « applet » créé alors dans le répertoire du croquis. Le dossier « applet » contient les fichiers suivants :

index.html

Fichier HTML dans lequel sont insérés l'applet, un lien vers le code source et un autre vers le site Processing.org. Vous pouvez ouvrir ce fichier dans votre navigateur à condition que la Machine virtuelle java soit installée pour ce navigateur (ce qui est pratiquement toujours le cas)

Sketch_123.jar

Archive JAVA qui contient tous les fichiers nécessaires au croquis pour s'exécuter. Elle inclut les classes de Processing, celles créées pour le croquis, ainsi que les fichiers de média (comme les images) si elles font partie du croquis.

Sketch_123.java

Fichier JAVA généré par le pré-processeur à partir du fichier PDE. C'est le fichier qui est compilé comme Applet par Jikes, le compilateur JAVA utilisé avec Processing.

Sketch_123.pde

Le fichier original contenant le code du programme. On y accède à partir de la page index.html.

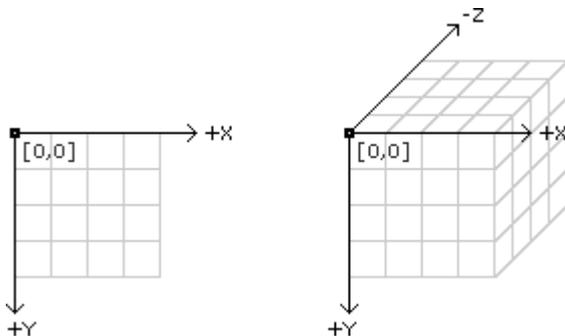
Chaque fois que le croquis est exporté, tous les fichiers ci-dessus sont réécrits. Toute modification faite entre temps dans la page index.html sont alors perdues.

Quand un croquis est exporté, tous les fichiers situés dans les dossiers « data » et « code » sont compactés dans une seule archive JAR. Les images et les bibliothèques inutiles doivent être enlevées de ces dossiers pour conserver un fichier de taille minimum. Par exemple, s'il existe un grand nombre d'images inutilisées dans le dossier « data », elles vont être ajoutées à l'archive JAR, augmentant inutilement la taille du programme. Les archives JAR peuvent être ouvertes avec des Logiciels comme Winzip, Stuffit, Winrar,... Ainsi on peut voir et extraire les fichiers.

Dans le futur, il sera possible d'exporter des applications. Ceci permettra aux programmes de fonctionner indépendamment de l'environnement Processing, en plein écran et d'accéder aux données refusées aux applets (comme par exemple charger des images directement à partir d'Internet).

Coordonnées (Coordonnées)

Processing utilise un système de coordonnées cartésiennes dont l'origine est le coin supérieur gauche de la fenêtre. Si votre programme fait 320 pixels de large par 240 pixels de haut, les coordonnées [0, 0] sont celles du pixel en haut à gauche, et les coordonnées [320, 240] sont celles du pixel situé le plus bas à droite. Le dernier pixel visible dans le coin en bas à droite a pour coordonnées [319, 239] parce que les pixels sont dessinés à droite et en dessous de leurs coordonnées.



Processing peut aussi simuler un univers à trois dimensions. À la surface de l'image, la coordonnée z est zéro, avec des valeurs négatives de z s'enfonçant dans l'espace. En dessinant dans un univers de simulation 3D, la « caméra » (le point de vue) est placée au milieu de l'écran.

Programming Modes (Modes de Programmation)

Processing permet de programmer à trois niveaux de complexité : Le Mode Statique, Le Mode Continu et le Mode Java. Les Novices en programmation doivent plutôt commencer avec le Mode Statique, pour se familiariser avec les notions de coordonnées, les variables, les boucles avant de passer aux Modes intégrant les structures draw() et loop().

Static Mode (Mode Statique)

Ce mode est utilisé pour dessiner des images statiques et apprendre les fondamentaux de la programmation. De simples lignes de code ont une représentation directe sur l'écran. L'exemple qui suit dessine un rectangle jaune à l'écran.

```
size(200, 200);
background(255);
noStroke();
fill(255, 204, 0);
rect(30, 20, 50, 50);
```

Continuous Mode (Mode Continu)

Ce mode utilise une structure setup() qui est exécutée une fois quand le programme débute. Écrivez les parties mobiles du programme dans la partie loop() ou d'autres structures comme mousePressed(). Dans ce mode, il est possible d'inventer de nouvelles fonctions et classes.

Cet exemple dessine quatre cercles sur l'écran et utilise une fonction créée intitulée cercles(). La fonction cercles() ne fait pas partie intégrante du langage de processing, mais a été définie spécialement pour cet exemple. La structure draw() est jouée une seule fois après setup().

```
void setup()
{
  size(200, 200);
  noStroke();
  background(255);
  fill(0, 102, 153, 204);
  smooth();
}

void draw()
{
```

```

    cercles(40, 80);
    cercles(90, 70);
}

void cercles(int x, int y) {
    ellipse(x, y, 50, 50);
    ellipse(x+20, y+20, 60, 60);
}

```

Cet exemple dessine des rectangles qui suivent la position de la souris (positions stockées dans les variables-système mouseX et mouseY). La partie loop() s'exécute indéfiniment jusqu'à ce que le programme soit arrêté, ceci permettant les mouvements et les interactions.

```

void setup()
{
    size(200, 200);
    rectMode(CENTER);
    noStroke();
    fill(0, 102, 153, 204);
}
void loop()
{
    background(255);
    rect(width-mouseX, height-mouseY, 50, 50);
    rect(mouseX, mouseY, 50, 50);
}

```

Java Mode (Mode Java)(*Non disponible avant Processing Version 70*)

Ce mode sera le plus flexible, et permettra d'écrire des programmes JAVA complets avec Processing. Écrire en Mode Java supprime les limites dues aux bibliothèques liées à Processing et donne accès au langage JAVA dans sa totalité.

```

public class MyDemo extends PApplet {
    void setup()
    {
        size(200, 200);

        rectMode(CENTER);
        noStroke();
        fill(0, 102, 153, 204);
    }

    void loop()
    {
        background(255);
        rect(width-mouseX, height-mouseY, 50, 50);
        rect(mouseX, mouseY, 50, 50);
    }
}

```