



ON VA CODER !

Introduction à la programmation graphique

Titouan Milon tmilon@student.42.fr

42 Staff pedago@staff.42.fr

Tuteurs tuteurs@student.42.fr

Résumé: les Pamplemousses Ananas, ces Immondices, ont Mangé mon Tacos...

Table des matières

I	Préambule	2
II	Introduction	4
III	Consignes générales	6
III.1	Consignes	6
IV	Fonctionnalités de base	8
IV.1	Palier 1 - On commence en douceur	8
IV.2	Palier 2 - Un peu plus d'interactivité et de complexité	8
IV.3	Palier 3 - Chroma Squad	8
IV.4	Palier 4 - On passe aux choses sérieuses	9
V	Bonus - Palier free for all	10



Chapitre I

Préambule



Félicitations ! Vous venez de sortir de l'académie des vilains, votre master Maître de donjon maléfique option extorquage de sucettes aux enfants en bas-âge en poche. Vos efforts acharnés vous ont permis de décrocher une bourse suffisamment élevée pour vous payer votre premier château des ténèbres (vous avez loué les éclairs pendant une semaine pour faire bonne impression sur le voisinage, mais vous tenez quand même à ménager votre budget).

Cependant...

Une fois arrivé sur les lieux, y a comme quelque chose qui cloche...



Votre château est désespérément vide de sbires pour s'occuper des basses besognes à votre place ! Heureusement, vous n'êtes pas sans ressources et quelques recherches sur le d4rkn3t vous procurent rapidement avec de quoi construire votre propre PAIMT (Programme Avancé d'Invocation de Monstres au travers des Tracés)

A vous de tirer parti de cet outil qui vous permettra d'ériger rapidement une armée digne de ce nom avec la magie la plus dangereuse et arcanique qui soit : la TECHNOLOGIE.

Bien sûr, si les gobelins et autres elfes noirs ne vous enchantent guère, rien ne vous empêche de partir sur une armée de robots futuristes, de clones de Marge Simpson, d'anomalies lovecraftiennes ou plus effrayant, des Gnomes des Forêts du Nord unijambistes dansant à la pleine lune au milieu de douze statuettes enroulées dans du jambon.

Bon courage, et n'oubliez pas de tricher sur votre voisin si vous rencontrez la moindre difficulté ;)

TL;DR : on va refaire un paint basique.

Chapitre II

Introduction

Bien que la majorité des projets à 42 ou durant la piscine se fassent en C, aujourd'hui nous allons nous tourner vers Processing, une librairie java qui permet de faire du dessin informatique de manière propre et rapide.

Pourquoi Processing me demanderez-vous ?

Voici comment tracer une ligne entre un point start et un point finish en C (version courte) :

```
static void    img_put_pixel(t_mlxunit *unit, t_point point, t_data data)
{
    if ((point.ry * data.window_w) + point.rx < data.window_w * data.window_h
        && ((point.ry * data.window_w) + point.rx) < data.window_w * (point.ry + 1)
        && ((point.ry * data.window_w) + point.rx) > data.window_w * point.ry
        && point.ry >= 0 && point.rx >= 0
        && ((point.ry * data.window_w) + point.rx >= 0))
        unit->picstr[point.ry * data.window_w + point.rx] = point.color;
}

static void    bresenham(t_data data, t_mlxunit *unit, t_point start, t_point finish)
{
    int distx;
    int disty;
    int err;
    int tmp_err;

    distx = abs(finish.rx - start.rx);
    disty = abs(finish.ry - start.ry);
    err = (distx > disty ? distx : -disty) / 2;
    while (start.rx != finish.rx || start.ry != finish.ry)
    {
        img_put_pixel(unit, start, data);
        tmp_err = err;
        if (tmp_err > -distx)
        {
            err -= disty;
            start.rx += (start.rx < finish.rx ? 1 : -1);
        }
        if (tmp_err < disty)
        {
            err += distx;
            start.ry += (start.ry < finish.ry ? 1 : -1);
        }
    }
    img_put_pixel(unit, start, data);
}
```

Voici comment tracer une ligne entre un point start et un point finish et de couleur col sous Processing (version longue) :

```
void drawLine(int startx, int starty, int finishx, int finishy, color col)
{
    stroke(col);
    line(startx, starty, finishx, finishy);
}
```

Voilà pourquoi :)

Chapitre III

Consignes générales

III.1 Consignes

Vous aurez face à vous deux fenêtres : "Demo_technique.pde", qui contient tous les éléments qui vous seront nécessaires à la réalisation des premiers paliers, et "PAIMT.pde", qui sera l'endroit où vous écrirez votre propre programme. Prenez le temps d'expérimenter avec la démo afin de comprendre son fonctionnement. Un programme se lance en appuyant sur le bouton en haut à gauche.

N'ayez pas peur de casser le programme, à partir du moment où vous faites toutes vos manipulations dans un fichier à part, vous pourrez toujours revenir en arrière. On a même des backups au cas où vous effaceriez/modifieriez la démo technique par inadvertance.



Les exercices sont répartis par paliers, mais vous n'êtes pas obligés de passer immédiatement au suivant quand vous finissez un exercice : si vous voulez rajouter des choses avant, ou partir dans une direction différente du sujet, ce n'est pas grave, l'important est que vous vous amusiez.

Voici quelques tips pour vous aider à partir du bon pied :

- Vous apercevrez deux onglets, un contenant *fannexe*. Ne vous en souciez pas pour l'instant, il contient des fonctions pour vous faciliter la tâche. Vous pouvez y jeter un oeil si vous avez envie, mais vous risquez de vous embrouiller plus qu'autre chose.
- Voir tout le code d'un coup peut être déroutant. Copier coller le code de la démo technique petit à petit dans votre propre programme et voir quelles fonctions apparaissent peut être une bonne approche pour comprendre son fonctionnement.
- Les valeurs mouseX, mouseY, pmouseX et pmouseY sont capitales pour un bon nombre de chose. Remplacez certaines valeurs avec elles et voyez ce qui se passe. Arrivez-vous à voir la différence entre pmouse et mouse ?
- Essayez de mettre // devant la commande background et relancez la démo technique. Que remarquez-vous ?

- 3 paramètres de couleur (red, green, blue) vous ont été donnés pour vous aider à démarrer. Si vous changez les 3 nombres mis au départ, que se passe-t-il? Vous pouvez vous aider de [ce site](#) pour choisir vos couleurs.



N'hésitez pas à demander de l'aide aux étudiants encadrant l'atelier, ils sont là pour vous !

Chapitre IV

Fonctionnalités de base

IV.1 Palier 1 - On commence en douceur

- Commencez par trouver comment tracer un trait lorsque vous maintenez un bouton de la souris enfoncé.
- Une fois que vous avez trouvé, dessinez donc un monstre et émerveillez-vous devant les miracles de la technologie moderne.



Mots pouvant être utiles : `drawLine()`, `if`, `mousePressed`, `pmouseX` et `mouseX`, `pmouseY` et `mouseY`

IV.2 Palier 2 - Un peu plus d'interactivité et de complexité

- Essayez de faire en sorte que l'on puisse changer la couleur de votre trait en appuyant sur des touches (par exemple b pour bleu). Ajoutez du rouge, du vert et du bleu, ainsi que deux autres couleurs.
- Permettez également à l'utilisateur d'effacer tout son dessin en appuyant sur une touche. Si vous trouvez comment faire un mode gomme par-dessus le marché, c'est tout bénéf.



Mots pouvant être utiles : `background()`, `if`, `key`, `color`

IV.3 Palier 3 - Chroma Squad

- Rajoutez la fonction `drawSquares()` dans votre code et faites en sorte que l'on puisse changer la couleur du trait en cliquant sur les carrés de couleur correspondante.

- Créez deux autres carrés supplémentaires pour les couleurs que vous venez d'ajouter. Ils doivent être cliquables eux aussi.
- Trouvez un moyen d'augmenter / diminuer la taille de votre pinceau avec les touches + / - du clavier. Attention, si votre taille de pinceau est négative, votre programme plantera. A vous de faire en sorte que cela n'arrive pas.



Mots pouvant être utiles : `int`, `strokeWeight()`,
`mouseIsInBox()`, `key`, `&&`

IV.4 Palier 4 - On passe aux choses sérieuses

- On va faire un pinceau multicolore. Il vous faudra pour cela vous servir de la fonction `random`. Elle n'est pas présente dans la démo technique, vous devrez donc découvrir son fonctionnement par vous-même sur la documentation en ligne.
- On peut également changer la forme du pinceau. Faites en sorte que l'on puisse avoir une brosse en forme de fleur et une brosse en forme d'étoile. La première est une forme que vous pouvez réaliser vous-même, l'autre une image déjà présente dans les fichiers qu'il vous revient d'exploiter.
- Enfin, encore plus dur : l'étoile est une image déjà importée dans le programme, à vous maintenant de découvrir comment vous pouvez ajouter les vôtres pour gribouiller dessus.



Choses pouvant être utiles : [La documentation de Processing](#),
`boolean`, `random()`, `PImage`, `loadImage()`, `image()`

Chapitre V

Bonus - Palier free for all

Premièrement, félicitation pour être arrivé jusque-là. Vous commencez à avoir une bonne compréhension des notions de base en programmation. S'il vous reste du temps, vous pouvez continuer à améliorer votre PAIMT en vous inspirant des suggestions suivantes :

- Avoir un pinceau symétrique, un peu comme sur [cette vidéo](#) (en moins complexe, bien sûr)
- Avoir un pinceau qui mélange les couleurs : par exemple, ajouter du bleu sur du rouge avec ce pinceau fera du magenta. (get et colorLerp)
- Appliquer des filtres comme le noir et blanc ou le sépia sur vos images.
- Normalement, vous n'avez pas d'option pour dessiner avec une étoile multicolore. Il ne tient qu'à vous de changer cela.
- Tout ce qui vous passe par la tête, franchement.



Bien sûr, rien ne vous empêche de commencer à faire un autre programme comme un jeu si vous en avez envie ! La communauté Processing est remplie d'artistes et de créations dont vous pouvez vous inspirer pour la suite. Voilà quelques liens à explorer :

- [Le site officiel de Processing](#)
- [Une galerie de programmes Processing](#)
- [Tutoriel OpenClassroom pour faire un Pong en Processing](#)
- [Les tutoriels de Coding Train \(Si vous voulez quelque chose de vraiment costaud\)](#)