

Édition 2022



DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET

LES
TROPHÉES NSI



NOM DU PROJET : Robot traceur

> PRÉSENTATION GÉNÉRALE :

L'objectif du projet était de piloter un stylo qui écrit au tableau par de la programmation en langage Python. Nous nous sommes servis d'un microcontrôleur programmable en langage Python pour piloter un stylo sur un tableau blanc. Le robot est composé de deux moteurs pas à pas qui actionnent des courroies crantées, ce qui nous permet d'être très précis sur le déplacement du stylo. Afin de faire en sorte que celui-ci écrive ou non sur le tableau, nous utilisons un servo moteur qui éloigne le stylo du tableau. Les trois moteurs précédents sont pilotés grâce à nos fonctions en langage Python. La plus grosse problématique a été que si l'on tire sur une courroie autrement dit qu'un seul moteur est activé, le stylo traçait un arc de cercle. De plus, l'arc de cercle dépend de l'emplacement du stylo. Dans un premier temps l'objectif était de tracer un trait horizontal et un vertical afin de s'assurer d'avoir un contrôle véritable de notre robot. Ce qui nous permet par la suite de dessiner les figures que l'on souhaite ainsi que des images.

> ORGANISATION DU TRAVAIL :

Julien et Etan se sont occupés de la partie du contrôle du stylo. Julien a également proposé un tracé d'une autre figure. Agathe s'est occupée du tracé de la fractale, plus précisément du flocon de Koch avec l'aide de Etan. Quant à Jules et Valentin, ils se sont occupés du traitement d'image. Julien s'est occupé du montage vidéo.

Après une présentation du projet en classe et deux heures de travail sur la décomposition d'un problème complexe en fonctions, nous avons pris en main le simulateur. Nous avons eu une fréquence de travail dans notre établissement scolaire qui était de trois heures par semaines le mercredi après-midi pendant un mois. De plus nous sommes venus un samedi matin lors des portes ouvertes afin de nous regrouper librement pour travailler. De plus, nous avons chacun travaillé de nombreuses heures en dehors de l'établissement. Pour communiquer nos avancés, nous avons pu utiliser des outils de travail tel que des mails et les réseaux sociaux.

LES ÉTAPES DU PROJET :

- 1) prise en main du projet
- 2) divisions des tâches au sein de l'équipe
- 3) pris en main du simulateur
- 4) algorithmes + recherche mathématique et algorithmique : tracés de traits droits, fractale, traitement d'image
- 5) trait droit avec des tests réguliers
 - création de fonction permettant de connaître la longueur de courroies
 - fonction permettant d'aller au point initial au point final et inversement
 - division du trajet en morceaux afin d'être plus précis

6) flocon de Koch

- la première fonction attribue à deux points A et B trois autres points X, Y, Z et renvoie tous les points sous forme de liste [A, X, Y, Z], sans renvoyer volontairement le B
- la seconde permet de créer une liste de points beaucoup plus grande qui correspondent aux trois côtés du triangle itérant le nombre de fois voulu la première fonction
- tracer le flocon en utilisant la première fonction

7) traitement d'image

- utilisation de la fonction « size » et de la bibliothèque « PIL » pour reprendre les dimensions des images et en créer une nouvelle de la même taille
- définition de l'intensité des couleurs avec les variables R, G, B
- faire une moyenne de ces valeurs
- découpage de l'image en carrés de 10 pixels
- définir quatre seuils d'intensité de couleurs pour chaque carré (de blanc à coloré totalement)
- tracé de lignes verticales en fonctions du seuil d'intensité sur la photo
- tracé d'une nouvelle image composée uniquement des traits verticaux.

Explications du cœur du projet :

Nous avons en premier créé une fonction qui renvoie les longueurs des deux courroies, que l'on nommera A et B, en connaissant les coordonnées x et y du point initial. Nous avons ensuite créé une fonction permettant de faire l'inverse. Nous avons utilisé le théorème de Pythagore pour ces deux fonctions.

```
def xy_to_ab(x,y):
    a=sqrt(x**2+y**2)
    b=sqrt((m-x)**2+y**2)
    return a,b

def ab_to_xy(a,b):
    x=((a**2)-(b**2)+(m**2))/(2*m)
    y=sqrt(a**2-x**2)
    return x,y
```

La fonction suivante a pour but de tracer un trait droit entre deux points en calculant la différence de longueurs de A et B. Néanmoins, la modification des longueurs se faisait une à une, créant ainsi deux traits au lieu d'un seul pour aller au point voulu.

Pour résoudre ce problème nous avons donc dû modifier la fonction dont le principe est le suivant :

- Elle prend en argument les coordonnées x et y du point où l'on veut aller
- Elle utilise la fonction « xy_to_ab » pour connaître les longueurs des courroies
- Elle récupère également les coordonnées x et y, et a et b, du point où l'on se situe
- Elle découpe ensuite le trajet minimal en tout petit morceaux, en calculant pour chacun point ses coordonnées et en traçant entre chaque point consécutif grâce à la différence de longueurs des courroies. On obtient donc de toutes petites imperfections invisibles à l'œil nu.
- Nous obtenons ensuite un trait droit que l'on peut désormais utiliser pour la suite du projet



```
def deplacement_droit(xf,yf):
    n=1
    af,bf=xy_to_ab(xf,yf)
    aдебут=bot.left_motor.get_length()
    bдебут=bot.right_motor.get_length()
    xдебут,yдебут=ab_to_xy(aдебут,bдебут)
    for i in range(1000):
        ai=bot.left_motor.get_length()
        bi=bot.right_motor.get_length()
        xi,yi=ab_to_xy(ai,bi)
        xm=xдебут+(xf-xдебут)*0.001*n
        ym=yдебут+(yf-yдебут)*0.001*n
        am,bm=xy_to_ab(xm,ym)
        diff_a=am-ai
        diff_b=bm-bi
        if diff_a >0:
            valeur_a= True
        else:
            valeur_a= False
        if diff_b >0:
            valeur_b= True
        else:
            valeur_b= False
        courroie_g= abs(round(diff_a/bot.left_motor.get_step_length()))
        courroie_d= abs(round(diff_b/bot.right_motor.get_step_length()))
        for i in range(courroie_g):
            bot.left_motor.step(valeur_a)
        for i in range(courroie_d):
            bot.right_motor.step(valeur_b)
    n=n+1
```

FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Pour mettre au point les programmes, de nombreuses solutions ont été apportées au projet telles que la division des problèmes à résoudre en sous-problèmes et l'écriture algorithmiques dans un premier temps. Il y a également de nombreuses vérifications sur un simulateur que notre professeur a créé ou bien pour terminer la réflexion au sein du groupe afin de régler et améliorer le résultat. Le rendu graphique a compliqué à la mise en place de tests unitaires.

Difficultés rencontrées

Pour les traits droits

Pour les traits droits, nos principales difficultés ont été des problèmes mathématiques car le travail en coordonnées x et y nous paraissait moins complexe que cela l'était réellement. De plus comprendre le pilotage du stylo et comment situer son emplacement lors de sa progression pour à chaque fois définir le nouveau point final. Par conséquent nous avons dû changer notre manière de réfléchir pour palier à ce problème ?

Fractales

Pour les fractales, nous étions partis à l'origine sur le triangle de Sierpiński. Cependant au vu de la difficulté, notre professeur nous a dit que nous étions un peu trop ambitieux et nous ferions mieux de commencer par le flocon de Koch qui serait plus accessible. La principale difficulté rencontrée pour le flocon est mathématique. En effet, nous n'avons pas travaillé

sur les rotations vectorielles. Cela nous a demandé de nous renseigner et comprendre comment cela fonctionnait.

Traitement d'image

Nous avons rencontré plusieurs difficultés dans ce projet notamment au moment où l'on devait parcourir l'image afin de trouver comment remplir les pixels mais nous avons finalement trouvée une boucle qui permettait de découper toute l'image et de récupérer toutes les données sur les couleurs.

Temps

Cependant une des grandes difficultés supplémentaires au projet était le manque de temps. Malgré cela, nous avons réussi à faire tout ce que nous souhaitions. Nous avons donc réussi à faire des traits droits, qui nous ont par la suite permis de dessiner des figures géométriques, le flocon de Koch, et de pouvoir reproduire des figures ainsi que tracer des images.

OUVERTURE :

Nous pourrions améliorer la vitesse de déplacement dans les traits droits lorsque la distance serait suffisamment longue. De plus, afin d'améliorer la précision du traitement d'image, nous aurions pu utiliser différents algorithmes de traitement. Ensuite, un module turtle aurait pu être créé pour faciliter l'utilisation et ne pas avoir à utiliser les coordonnées x et y comme nous l'avons fait. Cela aidera également à diffuser une partie du projet.

Ce projet permet une diffusion à différentes échelles. Tout d'abord, nous pouvons le présenter aux élèves de Sciences Numériques et Technologiques de notre lycée puis le mettre sur le site du lycée et pour finir, sur le site académique de Numérique et Sciences Informatiques. Par conséquent, tout cela permettrait de valoriser la spécialité Numérique et Sciences Informatiques.

Nous sommes satisfaits du résultat car nous avons eu une bonne organisation dès le lancement du projet. Cependant, nous aurions aimé avoir plus de temps. Pour cela, une solution aurait été de s'y prendre en avance.

DOCUMENTATION

Le microcontrôleur est une carte Pyboard. Le seul module Python utilisé est PIL. Un simulateur (bot.py) a été fourni par l'enseignante afin de pouvoir programmer sans la carte. En attendant que Julien et Etan finissent les premières fonctions de tracé, un autre simulateur a été proposé par l'enseignante utilisant le module pygame. Le module math est nécessaire pour certaines figures (sin, cos...).

Pour tester nos programmes dans le simulateur, il suffit de déposer le fichier bot.py dans le même dossier que nos programmes et de les exécuter.