



École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. (33)2-47-36-14-14
Fax (33)2-47-36-14-22
www.polytech.univ-tours.fr

Rapport

Projet de Programmation et Génie Logiciel

Modélisation d'un spirographe à disques

Auteur(s)

Adam NASSIRI

[adam.nassiri@etu.univ-tours.fr]

Florian BETHENCOURT

[florian.bethencourt@etu.univ-tours.fr]

Encadrant(s)

Christophe LENTE

[christophe.lente@univ-tours.fr]

Polytech Tours

Département Informatique

Version du 17 Janvier 2022

Table des matières

Introduction

1 Conception

1.1 Conception mathématique

1.1.1 Définitions

1.1.2 Prérequis mathématiques

1.1.3 Coordonnées polaires du spirographe

1.1.4 Autres outils mathématiques

1.2 Conception logicielle

1.2.2 Principe de modélisation

1.2.3 Classes utilisées

2 Développement

2.1 Outils de développement utilisés

2.2 Classes principales

2.3 Distribution des tâches

2.4 Méthode d'affichage

2.5 Méthode de paramétrage du spirographe

2.6 Interactions utilisateur/machine

3 Résultat

3.1 Etat actuel du projet

3.2 Points notables

3.3 Difficultés rencontrées

3.4 Améliorations possibles

3.4.1 Généralisation du spirographe

3.4.2 Coloration par l'utilisateur

3.4.3 Interface graphique interactive

Conclusion

Bibliographie et liens utiles

Introduction

Ce projet a pour but de modéliser fidèlement un spirographe, en utilisant son modèle mathématique.

Qu'est-ce qu'un spirographe?

Un spirographe est un ensemble de roues dentées (aussi appelées disques dans ce rapport) dont on se sert pour dessiner des formes cycliques variées (Figure 1)

Une roue est statique, et une autre peut rouler de sorte à ce que les deux s'engrènent entre elles. On plante ensuite un crayon dans cette dernière pour produire des dessins guidés par l'engrenage des deux roues.

Figure 1:

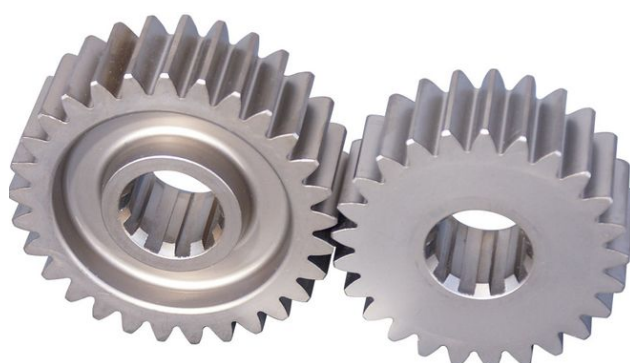


Le but de ce projet est de modéliser un spirographe, plus précisément on veut pouvoir dessiner avec fidélité la courbe dessinée par un spirographe composé d'un disque statique, et un disque roulant à l'intérieur (Figure 2) ou à l'extérieur (Figure 3) du disque statique.

Figure 2:



Figure 3:



Les spirographes sont des jouets très intéressants et les formes qu'ils permettent de créer peuvent être aussi complexes qu'élégantes.

Nous verrons qu'on peut relativement facilement les analyser et les définir en fonction de divers paramètres.

Nous nous proposons donc de percer le secret de ce jouet qui fait la passerelle entre mathématiques et art visuel.

Chapitre 1

Conception

1.1.1 Définitions

Plusieurs termes plus ou moins techniques sont à connaître pour pleinement saisir les analyses fournies dans ce rapport:

Une épitrochoïde est la courbe obtenue en suivant le tracé d'un crayon planté dans un disque qui tourne à l'extérieur du disque statique.

Une hypotrochoïde est la courbe obtenue en suivant le tracé d'un crayon planté dans un disque qui tourne à l'intérieur du disque statique.

Une épicycloïde est une épitrochoïde obtenue avec la pointe de crayon au bord du disque roulant.

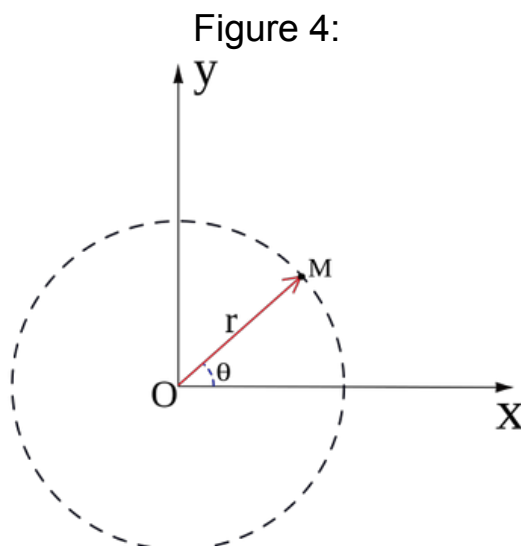
Une hypocycloïde est une hypotrochoïde obtenue avec la pointe de crayon au bord du disque roulant.

1.1.2 Prérequis mathématiques

Étant donné le caractère circulaire du spirographe, nous utiliserons les coordonnées polaires pour déterminer la position des disques et de la pointe du crayon.

En coordonnées polaires, on se repère en utilisant r et θ au lieu de x et y , ou r est en unités de longueur et θ est en radians.

Supposons un point M , r_M correspond à la distance entre M et l'origine O , et θ_M correspond à l'angle entre la droite OM et l'axe des abscisses. Exemple Figure 4.



On peut faire le rapport entre les coordonnées polaires et cartésiennes, on a les relations suivantes:

$$x_M = r_M \cos(\theta_M)$$

$$y_M = r_M \sin(\theta_M)$$

1.1.3 Coordonnées polaires du spirographe

Dans le cas d'un spirographe, nous irons une étape plus loin: nous avons deux disques, il nous faut donc logiquement deux sets de coordonnées polaires.

On place les points suivants:

O, origine du premier repère polaire, au centre du disque statique.

A, point de contact entre les deux disques.

B, origine du second repère polaire, au centre du disque roulant.

M, représente la pointe du crayon qui dessine ici une épitrochoïde.

R1 est le rayon du disque statique

R2 est le rayon du disque roulant

r est la distance $\|BM\|$

φ est l'angle entre la droite BM et l'axe x

on a donc en vecteurs:

$$\vec{OM} = \vec{OA} + \vec{AB} + \vec{BM}$$

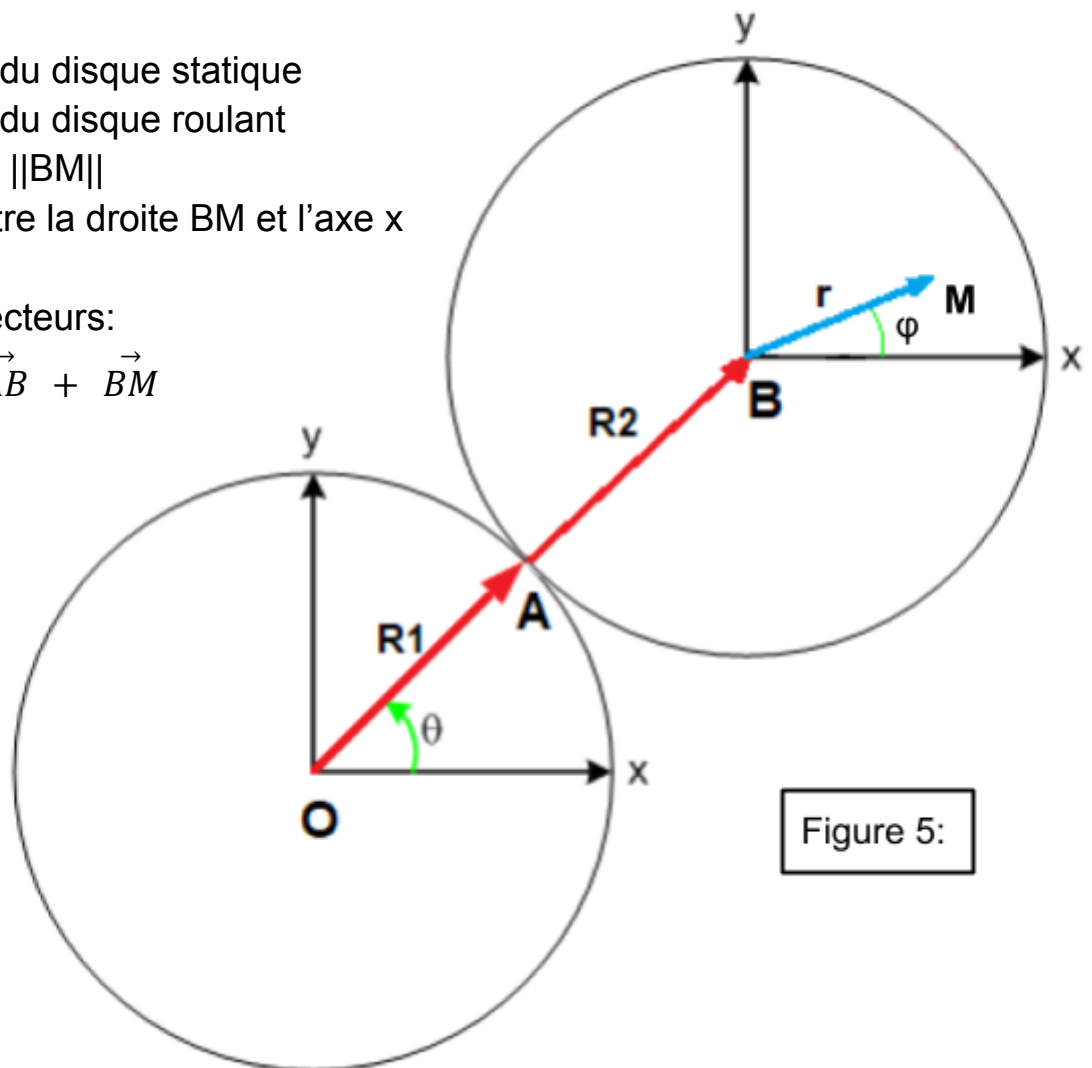


Figure 5:

Déduisons à présent les coordonnées du point M.

en coordonnées polaires on a:

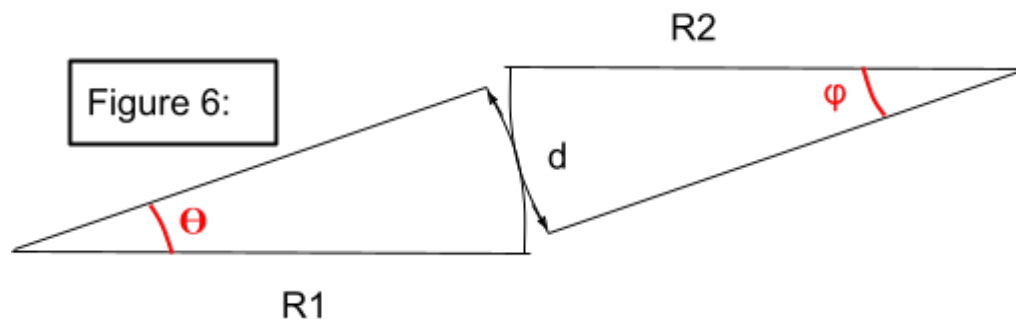
$$\vec{OM}_x = (R1 + R2) \cos(\Theta) + r \cos(\Theta + \varphi)$$

$$\vec{OM}_y = (R1 + R2) \sin(\Theta) + r \sin(\Theta + \varphi)$$

Dans la troisième composante on a $\Theta + \varphi$ car le disque roulant roule d'un angle φ et d'un angle Θ car il tourne autour du premier, la trajectoire du centre du disque roulant n'est pas droite.

Il est possible de mettre en évidence une relation entre Θ et φ comme suit:

Quand le disque roulant roule sur une distance d , le point de contact A décrit un arc de cercle d'une longueur d et de rayon $R1$



Sur la partie gauche du schéma **Figure 6** on peut voir les portions des disques concernés par le roulement.

On connaît la formule de la longueur d'un arc de cercle en fonction de son rayon et son angle, on peut alors poser les deux relations:

$$d = R1 * \Theta$$

$$d = R2 * \varphi$$

$$\Rightarrow R1 * \Theta = R2 * \varphi$$

La relation entre Θ et φ est donc:

$$\varphi = \Theta * \frac{R1}{R2}$$

La définition de OM devient alors:

$$\vec{OM}_x = (R1 + R2) \cos(\Theta) + r \cos(\Theta + \frac{R1}{R2} \Theta)$$

$$\vec{OM}_y = (R1 + R2) \sin(\Theta) + r \sin(\Theta + \frac{R1}{R2} \Theta)$$

Les équations paramétriques établies, nous avons toutes les clés en main pour commencer la modélisation de spirographe.

1.1.4 Autres outils mathématiques

Un aspect qu'il faut considérer est le suivant: comment sait-on quand un spirographe a complété un cycle? (nous entendons par "terminer le cycle" que la courbe est fermée).

Visuellement il est facile de déterminer si une courbe est achevée ou pas, mais cela ne tient pas mathématiquement, en outre nous utiliserons un peu d'arithmétique modulaire pour arriver à définir ce "point d'arrêt" du spirographe.

Physiquement, pour compléter le motif, il faut que le disque tournant roule sur une distance qui soit divisible par sa circonférence.

Mathématiquement, la distance nécessaire à la complétion du motif est la distance la plus petite qui respecte la relation:

$$d \bmod (2 * \pi * R2) = 0$$

d étant la distance sur laquelle le disque doit rouler au total, soit la distance parcourue par le point de contact entre les deux disques.

On a donc cette relation, or dans notre cas on sait que la distance d sera un multiple de la circonférence du disque statique (on effectue un nombre entier de tours), on peut donc dire que:

$$d = T * (2 * \pi * R1) \text{ avec } T \in \mathbb{N}^*$$

la relation précédente devient alors:

$$\Rightarrow (T * (2 * \pi * R1)) \bmod (2 * \pi * R2) = 0$$

T représente le nombre de tours que le disque roulant doit faire autour du disque statique pour compléter la trochoïde.

Pour déterminer le nombre de tours nécessaires on peut donc incrémenter T jusqu'à ce que le résultat de l'opération donne 0, on sait alors qu'il nous faut T tours pour fermer la courbe du spirographe.

Nous savons maintenant dessiner la courbe, et quand nous arrêter de dessiner.

Chapitre 2

Développement

2.1 Outils de développement utilisés

Le sujet spécifie un développement en C++, c'est pourquoi nous avons choisi l'IDE Code::Blocks. Nous avons tous les deux de l'expérience sur cet IDE particulier, et notre encadrant étant lui-même utilisateur le préférerait à d'autres solutions possibles comme Visual Studio par exemple.

Nous avons structuré l'avancée du projet grâce à Git. Nos précédents projets nous ont apporté toute l'expérience nécessaire pour utiliser efficacement cet outil de versioning, qui a rendu l'implémentation de fonctionnalités plus faciles pendant le développement.

Pour ce qui est des bibliothèques utilisées, nous avons fait le choix d'implémenter le Spirographe à l'aide de SFML, librairie open-source permettant de créer des interfaces graphiques simples, souvent utilisée pour la création de jeux vidéo et autres applications interactives.

2.2 Classes principales

La classe maîtresse de ce projet est la classe Spirograph, c'est l'objet Spirographe qui met à jour la position des disques roulants en fonction du temps et qui vérifie si la condition de fin est atteinte (si on a fini le motif).

La classe Disc représente les engrenages du spirographe, et possède des méthodes pour définir son mouvement autour et à l'intérieur d'un autre Disc.

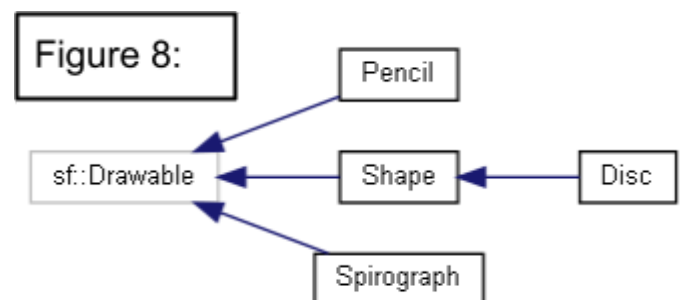
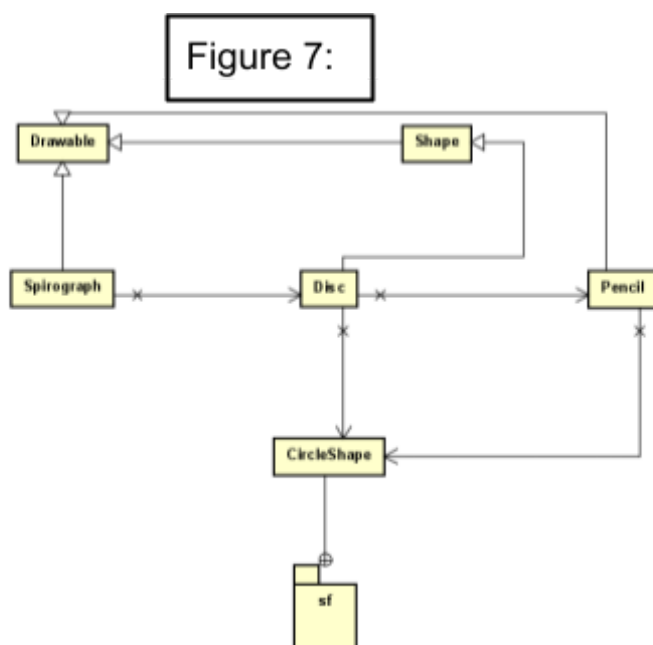
La classe Pencil représente la pointe d'un crayon posée sur la feuille, on peut lui assigner une couleur, et attacher un Pencil à un Disc pour simuler le crayon planté dans la roue dentée roulante du spirographe.

Nous avons porté une attention particulière à la réutilisabilité de ces classes, elles peuvent être utilisées facilement avec la librairie graphique SFML

On peut par exemple imaginer se servir de la classe Disc pour modéliser une cycloïde dessinée en roulant sur une droite plutôt qu'autour d'un autre Disc, assigner un gradient de couleur à un Pencil plutôt qu'une couleur solide, ou encore définir des fonctions de rotation autour d'une autre forme qu'un disque.

Un objet Spirograph contient des objets Disc et Pencil, d'un point de vue programmation cela permet de créer des spirographes aux caractéristiques différentes simplement en remplaçant les disques et crayons contenus.

Diagramme de classes généré avec Astah **Figure 7**, et diagramme d'héritage **Figure 8**.



2.3 Distribution des tâches

Dans la phase de conception mathématique, Adam s'est occupé de poser toutes les équations décrivant le mouvement d'un Spirographe, dans le but de quantifier l'évolution mathématique que nous devons reproduire.

Une fois la logique mathématique bien saisie, il ne fut pas compliqué de déterminer les classes que nous aurions besoin d'implémenter.

Étonnamment, il a été plutôt rapide de voir un Spirographe en construction même s'il était imparfait.

A ce stade, il a fallu corriger les imprécisions de calcul qui, accumulées, faussaient la trajectoire qui s'écartait de celle initialement attendue, et effectuer divers tests sur l'initialisation.

Après une avancée conséquente, le prochain défi était d'implémenter un Spirographe où la forme centrale n'était plus un disque mais une barre arrondie. Une fois de plus Adam s'est occupé des équations paramétriques de ce nouveau mouvement, pendant que Florian s'est attelé à l'automatisation de l'initialisation, jusque-là assez peu pratique, grâce à un Parser de fichier texte.

Les dernières modifications apportées ont permis de mieux structurer le projet en vue d'une réutilisation grâce à la classe Shape, et de corriger les erreurs de décalage grâce à une méthode de détection de fin de motif.

Enfin des fonctionnalités comme le screenshot et le changement de vitesse ont été apportées afin d'améliorer l'interaction utilisateur.

2.4 Méthode d'affichage

L'affichage du spirographe se fait via une fenêtre graphique aux dimensions 960 par 960 pixels.

Pour commencer le programme initialise tous les pixels de l'écran avec la couleur noire.

Puis à chaque frame (unité de temps unitaire du programme), on efface tout ce qui est actuellement dessiné sur l'écran, et on affiche le spirographe dans sa nouvelle position (après avoir mis à jour la position des disques et crayons)

Finalement on affiche la courbe en colorant à chaque frame le pixel correspondant aux coordonnées du crayon.

2.5 Méthode de paramétrage du spirographe

L'initialisation du spirographe s'effectue grâce à un fichier texte (resources/init.txt). On y précise le nombre de disques, puis le rayon de chacun suivi du type de roulement :

- “i” ou “e” pour interne ou externe au disque précédent
- “t” ou “c” roulement dans le sens trigonométrique ou horaire.

Enfin le nombre de crayons sur le dernier disque, avec chacun leur distance du centre du disque.

Exemple :

<i>Number_of_discs 2</i> <i>Disc 100 e c</i> <i>Disc 50 e t</i> <i>Number_of_pencils 1</i> <i>Pencil 80</i>

À noter que les paramètres de roulement du premier disque n’ont pas d’impact sur la forme du spirographe étant donné qu’il est statique.

Ici: “e” signifie “exterior”, on dessine donc une épitrochoïde, on pourrait entrer “i” (“i” signifie “interior”) pour dessiner une hypotrochoïde à la place.

“c” signifie “clockwise”, et “t” signifie “trigonometric”, le 2ème disque roule donc dans le sens trigonométrique.

Une classe Parser dotée d’une méthode statique M_Assert() se charge de vérifier la nomenclature du fichier, et d’écrire un message spécifique à l’utilisateur en cas d’erreur.

Pour chaque ligne lue d’init.txt, le constructeur du Spirographe va initialiser ses données membres au fur et à mesure en faisant appel à M_Assert() dès que nécessaire.

2.6 Interactions Utilisateur-Machine

Actuellement des interactions basiques sont implémentées telles qu’indiquées dans l’invité de commandes quand on lance le programme, l’utilisateur peut notamment:

- appuyer sur echap pour quitter le programme
- appuyer sur la barre espace pour afficher ou pas la structure du spirographe
- appuyer sur les flèches gauche et droite pour ralentir ou accélérer la rotation
- appuyer sur S pour prendre une capture d’écran

Chapitre 3

Résultat

3.1 Etat actuel du projet

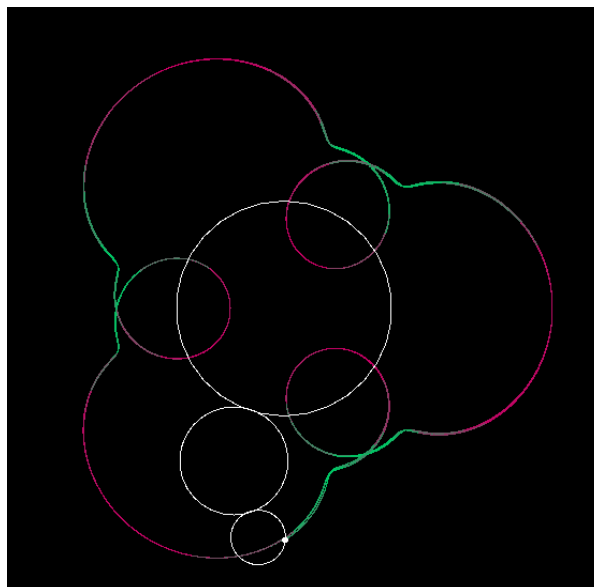
Dans son état actuel le software développé permet de simuler fidèlement un spirographe composé de 2 disques, le disque roulant peut se trouver à l'intérieur comme à l'extérieur et peut tourner dans le sens horaire ou trigonométrique sans difficulté.

L'utilisateur peut facilement définir les dimensions de son propre spirographe pour indiquer des conditions initiales au programme. Nous avons pensé notre code pour permettre les extensions et modifications, cela nous paraissait important pour éventuellement pouvoir compléter le projet avec d'autres formes que les disques, un contrôle des couleurs par l'utilisateur, etc.

Nous pensons avoir atteint l'objectif principal du projet.

3.2 Points notables

La fonctionnalité dont nous sommes particulièrement fiers était à la base involontaire: l'utilisateur peut définir un nombre de disques supérieur à 2 pour dessiner des formes parfois extrêmement complexes, le spirographe se transforme alors en une sorte de "machine à dessiner de Fourier", comme ici avec 3 disques de dimensions 100,50,25, tous en rotation externe dans le sens horaire



3.3 Difficultés rencontrées

L'un des principaux défis que nous avons tenté de relever est l'implémentation de formes autres que les disques, et bien que nous comprenions le principe mathématique et algorithmique, il nous fallait revoir l'architecture de la classe Spirograph pour implémenter par exemple le fait d'avoir un rectangle au centre au lieu d'un disque statique.

3.4 Améliorations possibles

Nous voudrions apporter dans le futur plusieurs améliorations à des facettes variées du projet, concernant l'ergonomie d'utilisation et l'évolution fonctionnelle du programme.

3.4.1 Généralisation du spirographe

Cette amélioration a pour but de faire en sorte qu'un disque puisse tourner autour de n'importe quelle forme en 2 dimensions, et que l'utilisateur puisse, à terme, définir ses propres formes pour les mettre au centre du spirographe et générer les motifs correspondants.

3.4.2 Coloration par l'utilisateur

Actuellement, il faut aller modifier les valeurs des couleurs dans le code pour obtenir de beaux dégradés de couleurs qui suivent l'évolution des angles du spirographe, nous aimerions rendre cette fonctionnalité disponible plus facilement à l'utilisateur, via une interface ou un paramètre entré dans le fichier texte par exemple.

3.4.3 Interface graphique interactive

La dernière facette du projet que nous aimerions améliorer est l'interface homme-machine, en effet actuellement, une fois le fichier texte écrit et le programme lancé, il est impossible de modifier les dimensions, les couleurs, et autres paramètres du spirographe.

L'utilisateur est contraint de quitter, modifier le fichier d'entrée, et relancer le programme.

Nous avons pour idée de rendre la modification "en live" possible avec une interface présente pendant la phase de dessin.

Conclusion

Ce projet a mis à l'épreuve notre capacité à planifier notre code, et nous a forcé à nous préparer longuement avant même d'entamer la programmation, c'était un exercice extrêmement instructif.

A titre personnel nous sommes heureux d'avoir pu conduire un projet visuel, cela nous a rendu particulièrement fiers du résultat, et à mesure que nous ajoutions des fonctionnalités, nous avons découvert toute la complexité derrière les mouvements pourtant intuitifs des fameuses roues dentées d'un spirographe.

L'attrait pour le côté graphique du projet a vite été éclipsé par l'envie d'en apprendre plus, d'ajouter de nouvelles fonctionnalités, de rendre le programme plus confortable à utiliser...

Nous espérons que ce projet pourra encore être approfondi dans le futur pour atteindre le but fixé à l'origine: un spirographe avec une forme centrale quelconque.

Bibliographie et liens utiles

- Page wikipedia sur les trochoïdes
<https://fr.wikipedia.org/wiki/Trocho%C3%AFde>
- Page wikipedia sur les Hypotrochoïdes
<https://fr.wikipedia.org/wiki/Hypotrocho%C3%AFde>
- Explication de ma méthode de détermination du point d'arrêt
<http://www.exo.net/~pauld/activities/spirograph/Spirograph.html>
(voir "Modular arithmetic what is the remainder?")
- Pour aller plus loin, une vidéo d'explication sur les séries de Fourier, une approche complexe plutôt que polaire
https://www.youtube.com/watch?v=r6sGWTCMz2k&ab_channel=Blue1Brown
- répertoire github de notre projet:
https://github.com/AdamWizard/Projet_Spirograph