Compte rendu projet tech spirographe

Antoine Dumont, Julien Chantail et Papis Diop

11/12/2024

Contents

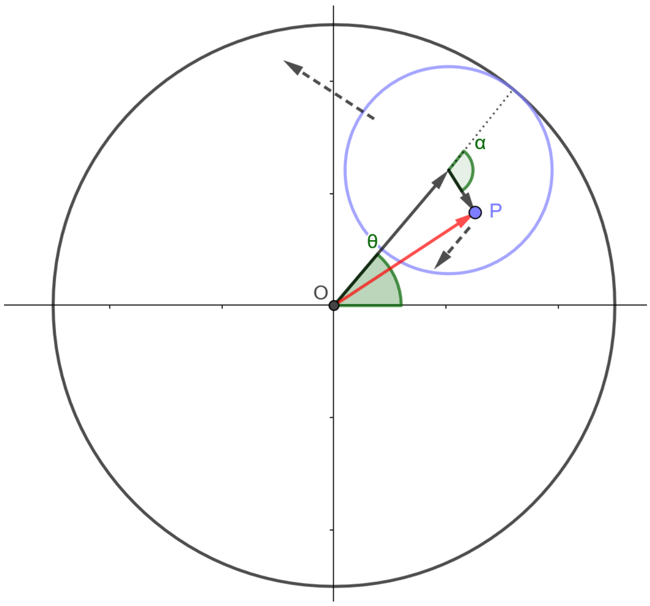
1. Introduction : explication du projet

2. Parties Mathématiques

Description et objectifs du projet

Le Spirograph est un jeu pour enfant des années 1960, permettant de dessiner des rosaces en un seul trait, sans lever le crayon. L'impression 3D par robocasting permet d'imprimer des pâtes formulées de divers matériaux (chocolat, terre de potier...) ayant la consistance de dentifrice au moment de l'impression. L'objet imprimé durcit après impression par séchage et/ou cuisson. Ce type d'impression 3D est particulièrement adapté à l'impression de motifs en continu. Ce projet consiste à développer un logiciel permettant de créer des motifs de Spirograph et de le fichier .gcode correspondant permettant d'imprimer ces motifs en robocasting

Définir le problème



définition des notations

- Soit O l'origine du spirographe
- Soit R le rayon du grand cercle
- Soit r le rayon du petit cercle
- Soit C le centre du petit cercle
- Soit θ l'angle entre l'axe Ox et (OC)
- Soit P le point où le crayon est en contact avec la feuille
- Soit p la distante CP
- Soit α le complémentaire de l'angle entre (OC) et (CP)

formulation du problème

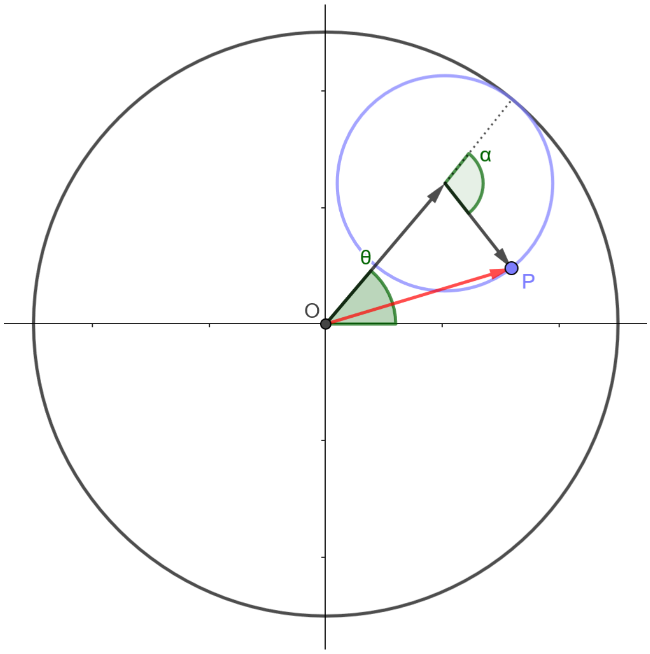
Le but du problème est donc de trouver les coordonnées de P en fonction de R, r et θ.

R, et r sont des données du problème tandis que θ représente l'état d'avancement du dessin.

résolution d'un problème plus simple

cas particulier

On va d'abord s'intéresser au cas particulier où P se situe à l'extrémité du disque de centre C et de rayon r :



démonstration

trouver les coordonnées de $P = (x, y)$ revient à trouver les coordonées du vecteur $\vec{OP}$.

Or :

$$\vec{OP} = \vec{OC} + \vec{CP}$$

Plaçons nous en coordonnées cylindrique :

$$OC = (R - r)$$

Donc :

$$\vec{OC} = \begin{pmatrix} (R-r)\cos(\theta) \\ (R-r)\sin(\theta) \end{pmatrix}$$

et :

$$\vec{CP} = \begin{pmatrix} r\cos(\theta+\alpha) \\ r\sin(\theta+\alpha) \end{pmatrix}$$

Or :

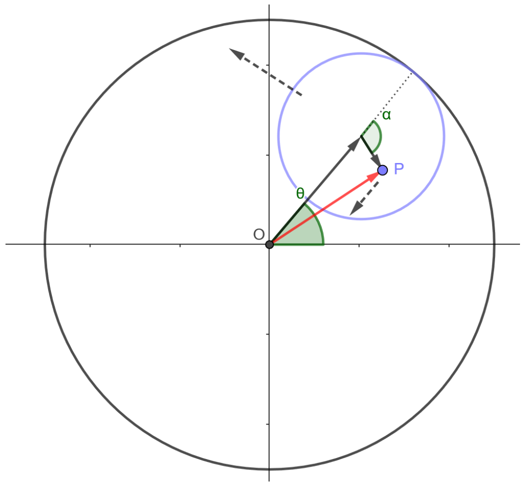
$$\alpha = -\frac{R}{r}\theta$$

D'où :

$$\vec{OP} = \begin{pmatrix} (R-r)\cos(\theta) + r\cos(\theta+\alpha) \\ \\ (R-r)\sin(\theta) + r\sin(\theta+\alpha) \end{pmatrix}$$

$$\vec{OP} = \begin{pmatrix} (R-r)\cos(\theta) + r\cos(\theta - \frac{R}{r}\theta) \\ \\ (R-r)\sin(\theta) + r\sin(\theta - \frac{R}{r}\theta) \end{pmatrix}$$

trouver la solution au problème

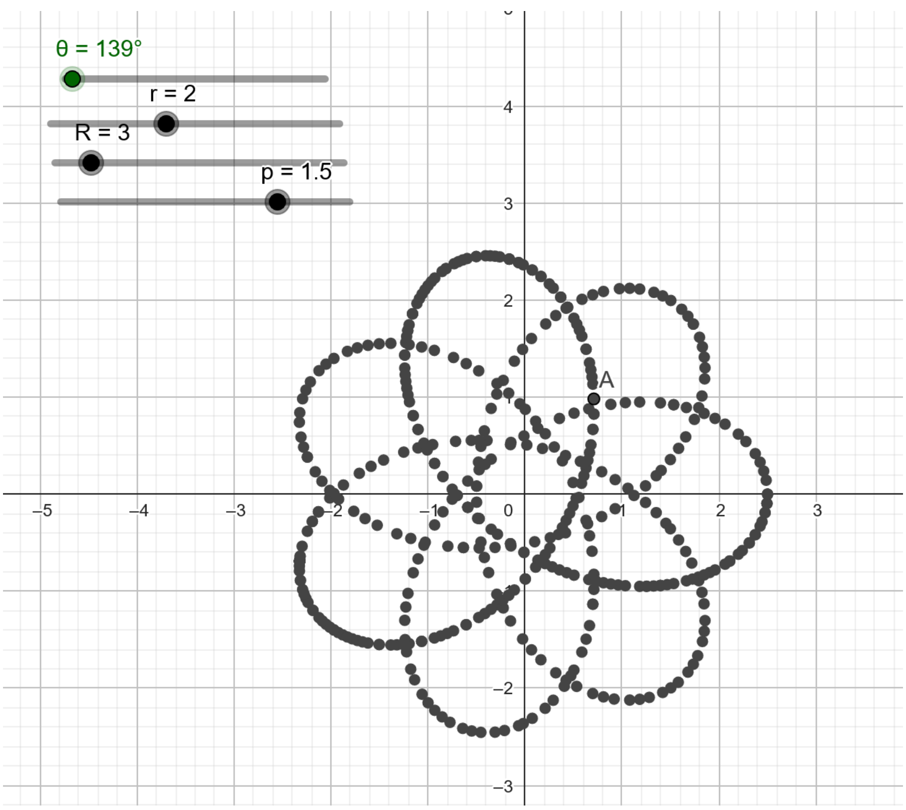


du cas particulier au cas général

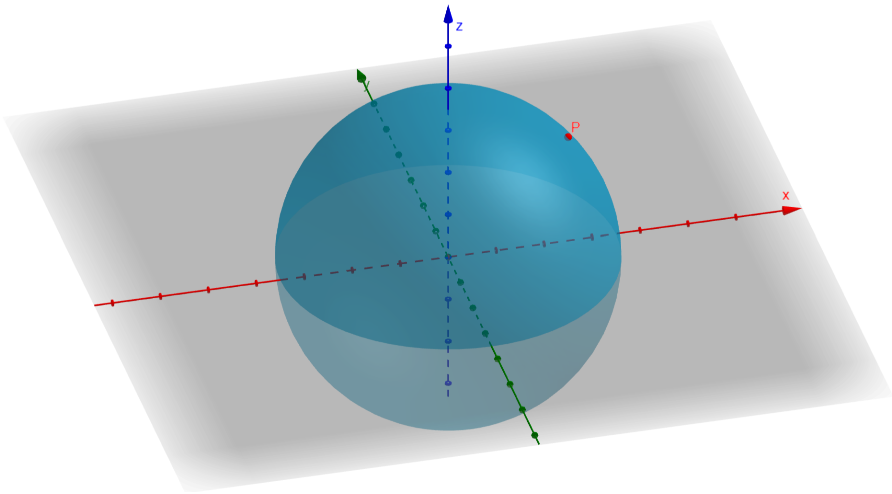
Il suffit ici d'adapter :
la distance CP n'est plus r mais p.
L'équation devient donc :

$$\vec{OP} = \begin{pmatrix} (R-r)\cos(\theta) + p\cos(\theta - \frac{R}{r}\theta) \\ \\ (R-r)\sin(\theta) + p\sin(\theta - \frac{R}{r}\theta) \end{pmatrix}$$

test de l'équation sur Géogébra



passage à la demi-sphère



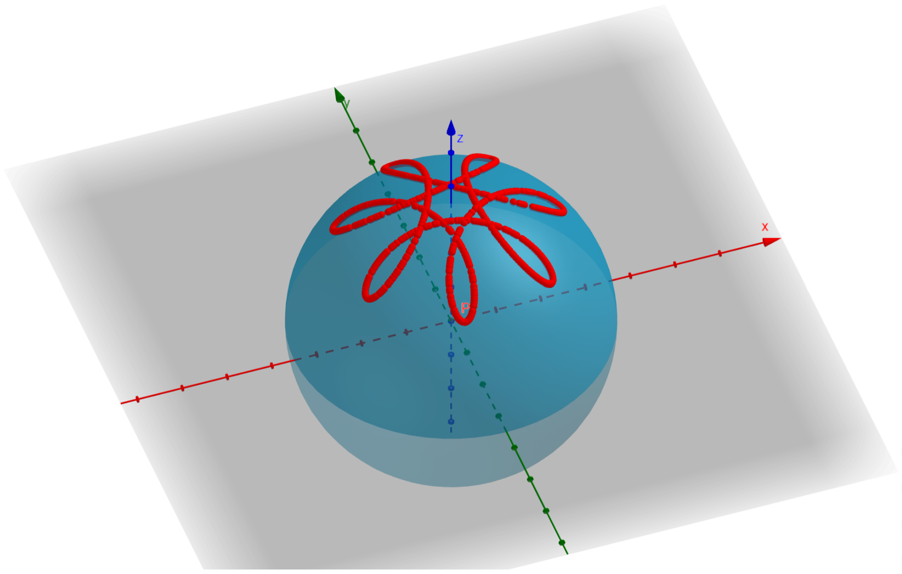
On ajoute une inconnue : z
Et on ajoute une équation : $x^2+y^2+z^2=R^2$

Donc

$$\begin{array}{lc} & x^2+y^2+z^2=R^2 \\ \iff & z^2=R^2-(x^2+y^2) \\ \iff & z=\sqrt{R^2-(x^2+y^2)} \\ \iff & z=\sqrt{R^2-\left(\left((R-r)\cos(\theta)+p\cos(\theta-\frac{R}{r}\theta)\right)^2+\left((R-r)\sin(\theta)+p\sin(\theta-\frac{R}{r}\theta)\right)^2\right)} \end{array}$$

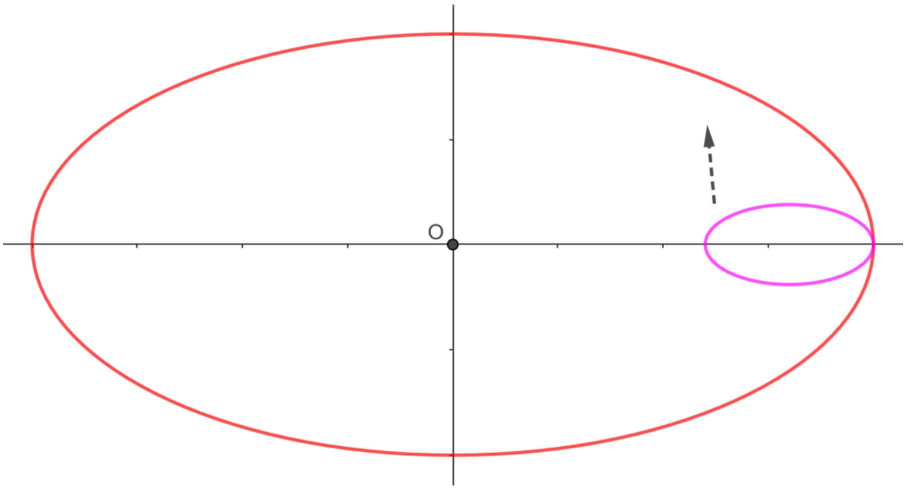
D'où :

$$\vec{OP}=\begin{pmatrix} (R-r)\cos(\theta)+p\cos(\theta-\frac{R}{r}\theta) \\ (R-r)\sin(\theta)+p\sin(\theta-\frac{R}{r}\theta) \\ \sqrt{R^2-\left(\left((R-r)\cos(\theta)+p\cos(\theta-\frac{R}{r}\theta)\right)^2+\left((R-r)\sin(\theta)+p\sin(\theta-\frac{R}{r}\theta)\right)^2\right)} \end{pmatrix}$$

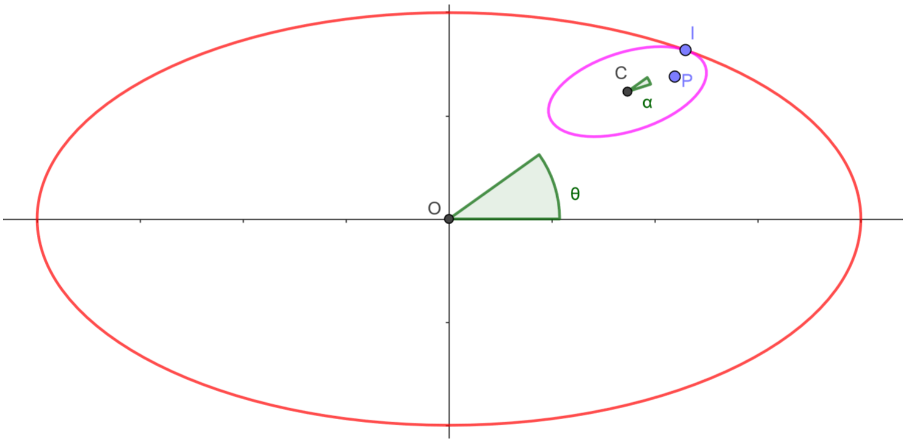


passage à l'ellipse, introduction

schéma initial

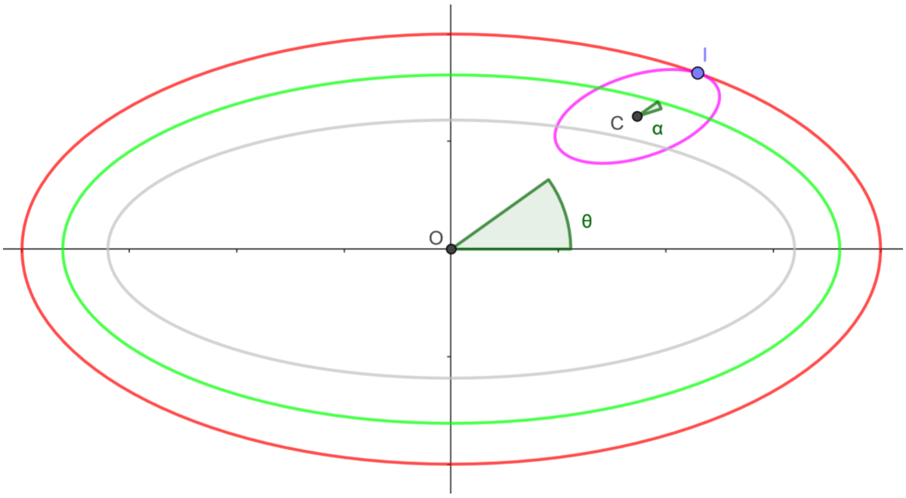


définition des notations



- Soit A le demi grand axe de l'ellipse rouge
- Soit B le demi petit axe de l'ellipse rouge
- Soit L le périmètre de l'ellipse rouge
- Soit a le demi grand axe de l'ellipse violette
- Soit b le demi petit axe de l'ellipse violette
- Soit l le périmètre de l'ellipse violette
- Soit C le centre de l'ellipse violette
- Soit I le point d'intersection des deux ellipse
- Soit P le point du tracé
- Soit θ l'angle entre l'axe Ox et (OC)
- Soit α l'angle entre la droite (CI) et la droite (CP).
- Soit p la distance CP
  Attention : ici α a fait plus d'un tour et donc vaut plus que 360

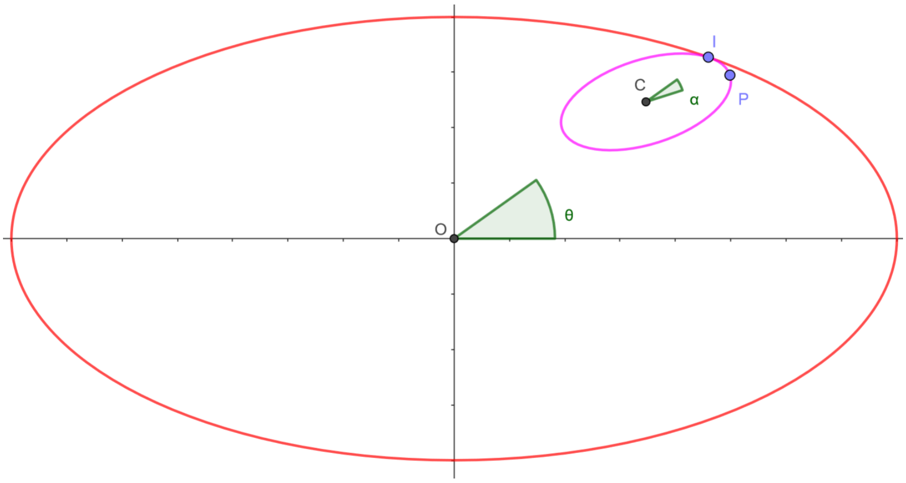
formulation du problème et observations



C est contenu entre l'ellipse verte (de demi grand axe $A-a$ et de demi petit axe $B-a$, et l'ellipse grise (de demi grand axe $A-b$ et de demi petit axe $B-b$.
On va d'abord déterminer sa trajectoire

Centre de la petite ellipse

Soit (x, y) les coordonnées de C dans le repère $(O, \vec{x}, \vec{y})$



On a :

$$\vec{OC} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} OC\cos(\theta) \\ OC\sin(\theta) \end{pmatrix}$$

Il suffit donc de trouver OC :
O, C et I sont toujours alignés, donc

$$OC = OI - CI$$

Les coordonnées de I s'écrivant en cylindrique

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} OI\cos(\theta) \\ OI\sin(\theta) \end{pmatrix}$$

Et I vérifie l'équation de l'ellipse rouge donc :

$$
\begin{aligned}
1 = \frac{x_i^2}{A^2} + \frac{y_i^2}{B^2} \quad &\Longleftrightarrow \quad 1 = \frac{OI^2 \cos^2(\theta)}{A^2} + \frac{OI^2 \sin^2(\theta)}{B^2} \\
&\Longleftrightarrow \quad \frac{1}{OI^2} = \frac{\cos^2(\theta)}{A^2} + \frac{\sin^2(\theta)}{B^2} \\
&\Longleftrightarrow \quad \frac{1}{OI^2} = \frac{B^2 \cos^2(\theta + A^2 \sin^2(\theta))}{A^2 B^2} \\
&\Longleftrightarrow \quad OI^2 = \frac{A^2 B^2}{B^2 \cos^2(\theta) + A^2 \sin^2(\theta)} \\
&\Longleftrightarrow \quad OI = \frac{A \times B}{\sqrt{B^2 \cos^2(\theta) + A^2 \sin^2(\theta)}}
\end{aligned}
$$

Et de la même façon on peut calculer la distance IC : On considère C comme étant l'origine, et on se place dans la petite ellipse. Le point P ici sur la figure sert à repérer le grand axe de l'ellipse, que l'on prend comme axe des abscisses. (le petit axe étant celui des ordonnées)

Les coordonnées de I dans ce repère sont :

$$
\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} CI \cos(\alpha) \\ CI \sin(\alpha) \end{pmatrix}
$$

Donc par le même calcul que juste avant on trouve :

$$
CI = \frac{a \times b}{\sqrt{b^2 \cos^2(\alpha) + a^2 \sin^2(\alpha)}}
$$

Hors, de la même façon que pour les cercles, α est lié à θ par la relation :

$$
\alpha = -\frac{L}{l}\theta
$$

Avec L et l les périmètres des deux ellipses, valant :

$$
L = 4 \int_0^{\frac{\pi}{2}} \sqrt{A^2 \sin^2(t) + B^2 \cos^2(t)} dt \text{ et } l = 4 \int_0^{\frac{\pi}{2}} \sqrt{a^2 \sin^2(t) + b^2 \cos^2(t)} dt
$$

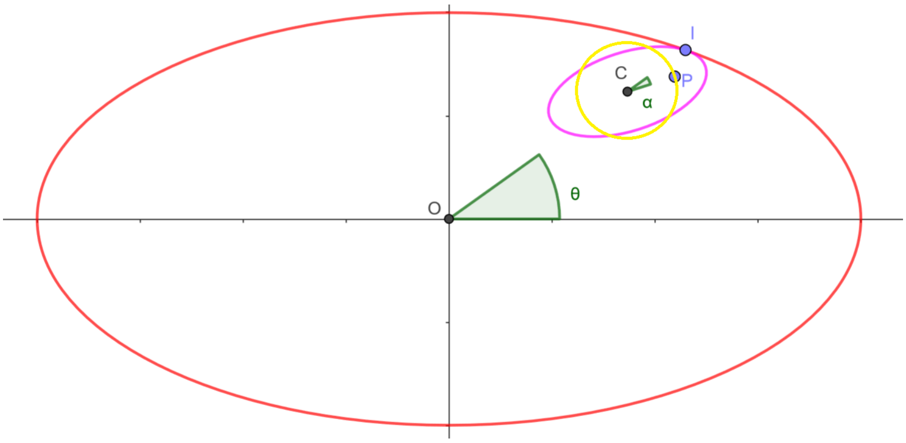
d'où :

$$
CI = \frac{a \times b}{\sqrt{b^2 \cos^2 \left(-\frac{4 \int_0^{\frac{\pi}{2}} \sqrt{A^2 \sin^2(t) + B^2 \cos^2(t)} dt}{4 \int_0^{\frac{\pi}{2}} \sqrt{a^2 \sin^2(t) + b^2 \cos^2(t)} dt} \theta \right) + a^2 \sin^2 \left(-\frac{4 \int_0^{\frac{\pi}{2}} \sqrt{A^2 \sin^2(t) + B^2 \cos^2(t)} dt}{\int_0^{\frac{\pi}{2}} \sqrt{a^2 \sin^2(t) + b^2 \cos^2(t)} dt} \theta \right)}}
$$

On a donc les coordonnées du point C :

$$\begin{aligned}\vec{OC} &= \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} OC\cos(\theta) \\ OC\sin(\theta) \end{pmatrix} \\ &= \begin{pmatrix} (OI - CI)\cos(\theta) \\ (OI - CI)\sin(\theta) \end{pmatrix}\end{aligned}$$

Solution du problème, coordonées du point C

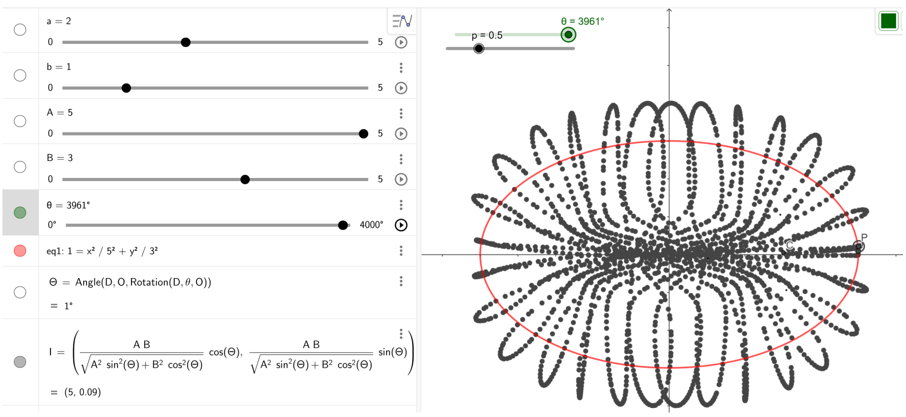


Ici on se rend compte que P décrit un cercle autour de C. Il suffit donc de reprendre les calculs du problème précédent, et d'adapter le α

Donc :

$$\begin{aligned}\vec{OP} &= \vec{OC} + \vec{CP} \\ &= \begin{pmatrix} (OI - CI)\cos(\theta) \\ \\ (OI - CI)\sin(\theta) \end{pmatrix} + \begin{pmatrix} p\cos(\theta - \alpha) \\ \\ p\sin(\theta - \alpha) \end{pmatrix} \\ &= \begin{pmatrix} \left(\dfrac{A \times B}{\sqrt{B^2\cos^2(\theta) + A^2\sin^2(\theta)}} - \dfrac{a \times b}{\sqrt{b^2\cos^2(\alpha) + a^2\sin^2(\alpha)}}\right)\cos(\theta) \\ \left(\dfrac{A \times B}{\sqrt{B^2\cos^2(\theta) + A^2\sin^2(\theta)}} - \dfrac{a \times b}{\sqrt{b^2\cos^2(\alpha) + a^2\sin^2(\alpha)}}\right)\sin(\theta) \end{pmatrix} + \begin{pmatrix} p\cos(\theta - \frac{L}{l}\theta) \\ \\ p\sin(\theta - \frac{L}{l}\theta) \end{pmatrix}\end{aligned}$$

test de l'équation sur Géogébra

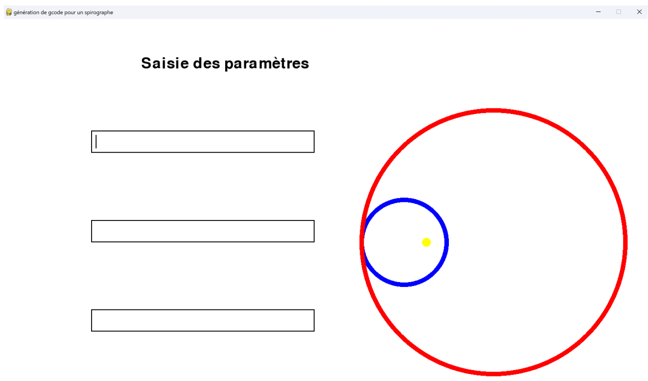


3. Interface

Outil d'affichage et de calcul

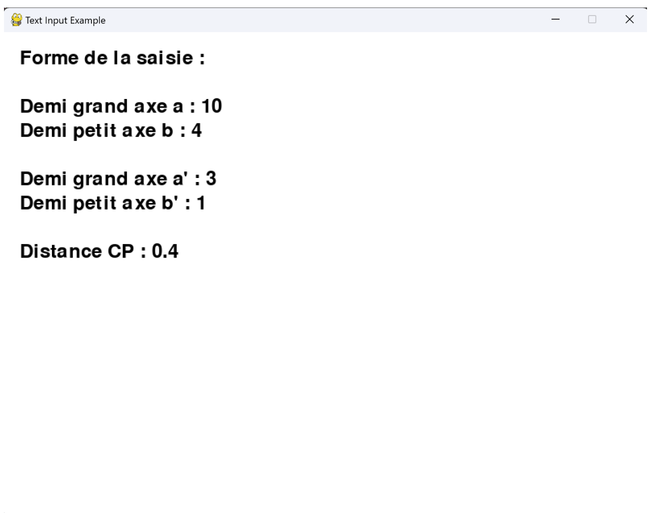
Nous utiliserons pour ce projet un programme python. Ce programme sera capable de receuillir les données rentrées par l'utilisateur via une interface facile d'utilisation. Il affichera le résultat de façon visuel, et produira un fichier GCODE capable d'être lu par une imprimante 3D. La librairie utilisée pour l'affichage est pygame.

Idée, maquette



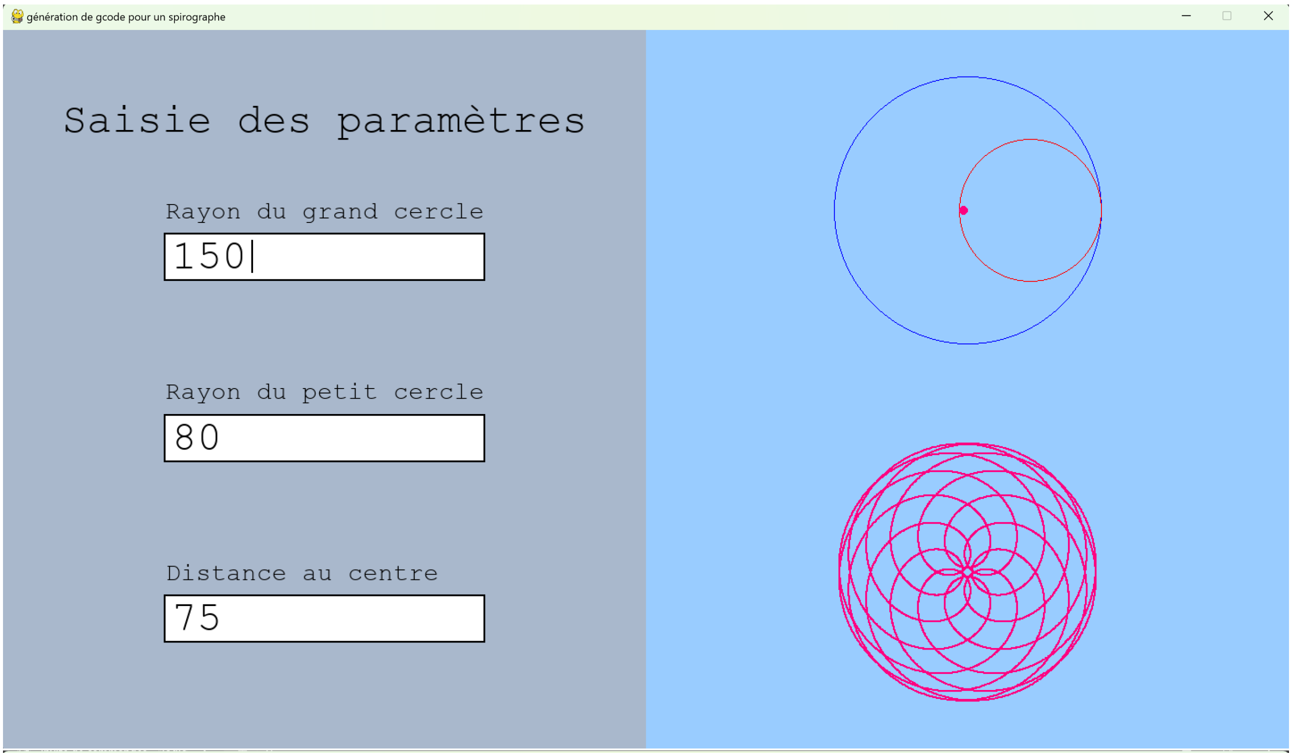
Le but est de permettre à l'utilisateur de saisir les différents paramètres. D'afficher la taille des ellipses/cercles en temps réel. D'afficher également le rendu final

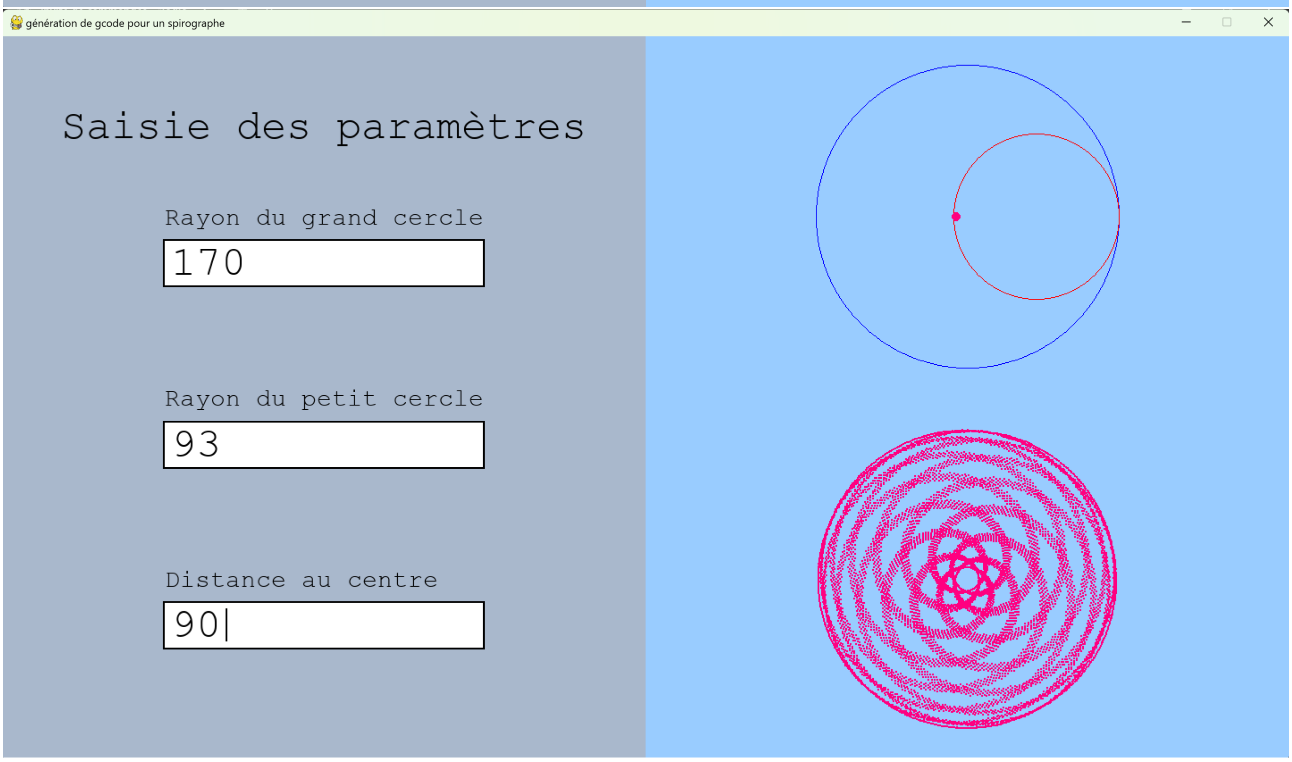
saisie des paramètres



Avec pygame on est capable d'afficher les données rentrées par l'utilisateur

rendu intermédiaire

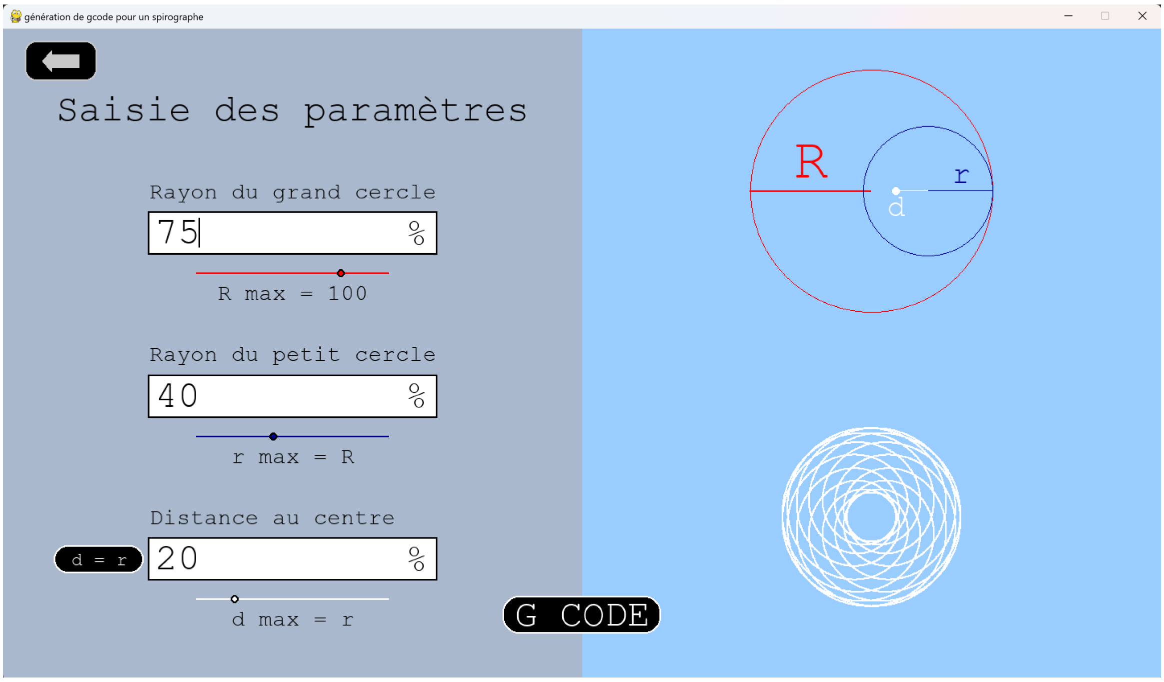


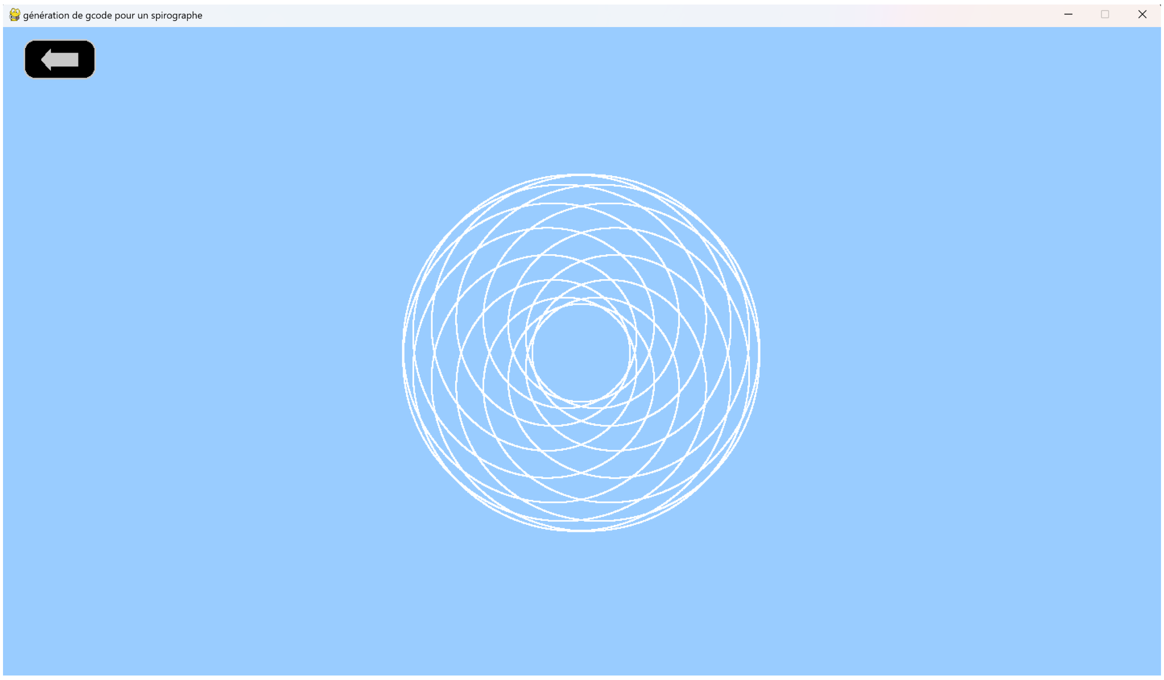


Fonctionalités :

- prend tout l'espace de l'écran
- les 3 champs sont clickables et modifiables
- apparition instantanée du rendu lors de l'appui sur la touche entrée
- au dessus la taille des 2 cercles et du point
- en dessous le rendu final
- le petit cercle ne peut pas dépasser le grand cercle.
- le point ne peut pas sortir du petit cercle.

Rendu final 2D

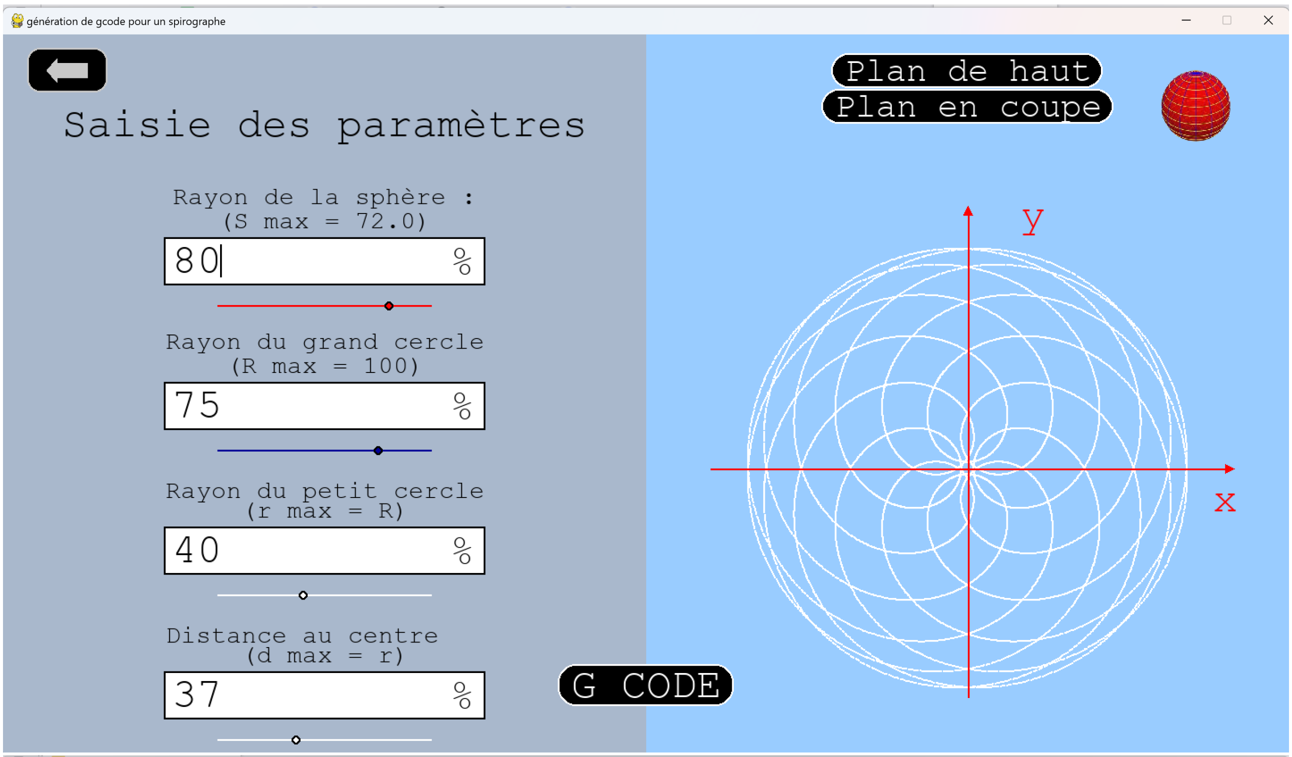


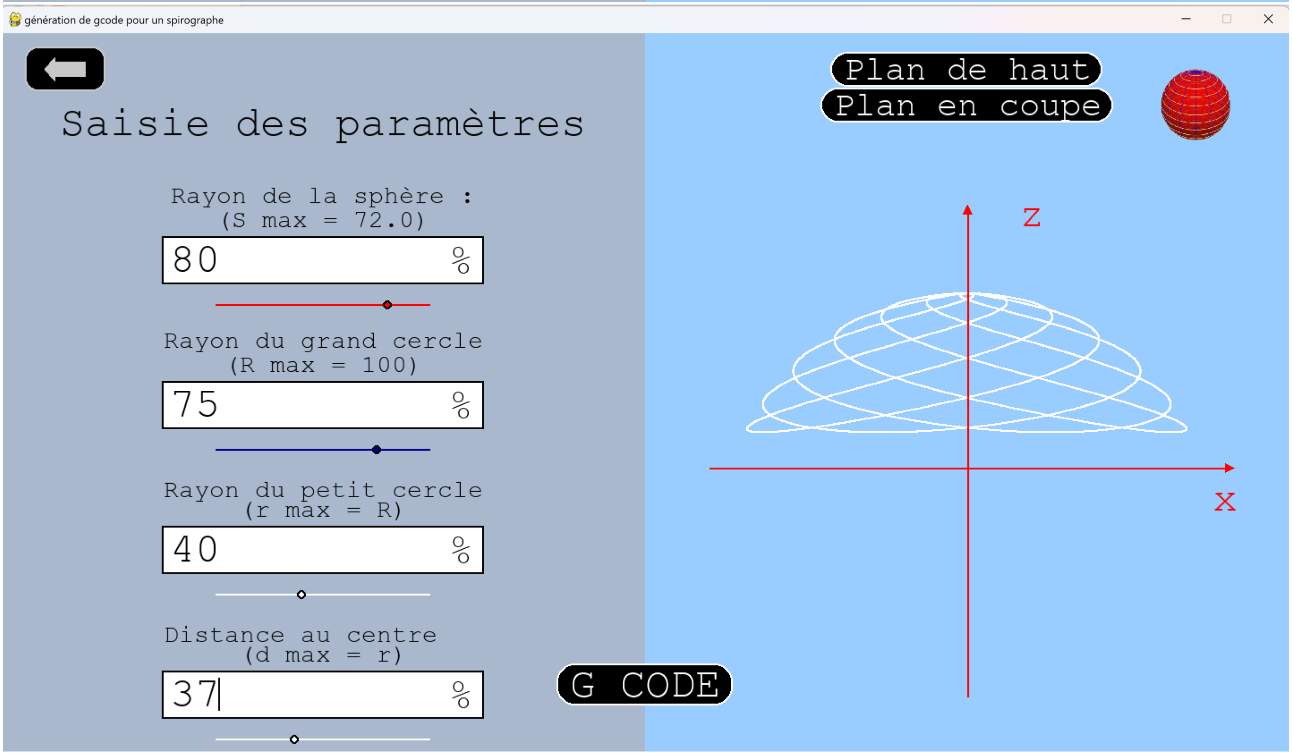


Fonctionalités :

- l'écran s'adapate à la taille de n'importe quel écran et ne déforme rien
- bouton retour
- normalisation des valeurs (sur 100 -¿ %)
- bouton G CODE pour accéder à l'interface de génération de G CODE
- Bouton retour pour changer de menu
- Zoom possible sur le rendu ou le schéma
- possibilité d'utiliser des curseurs pour changer les valeurs
- changement du rendu en temps réel
- Définition des variables sur le schéma

Interface pour la 3D





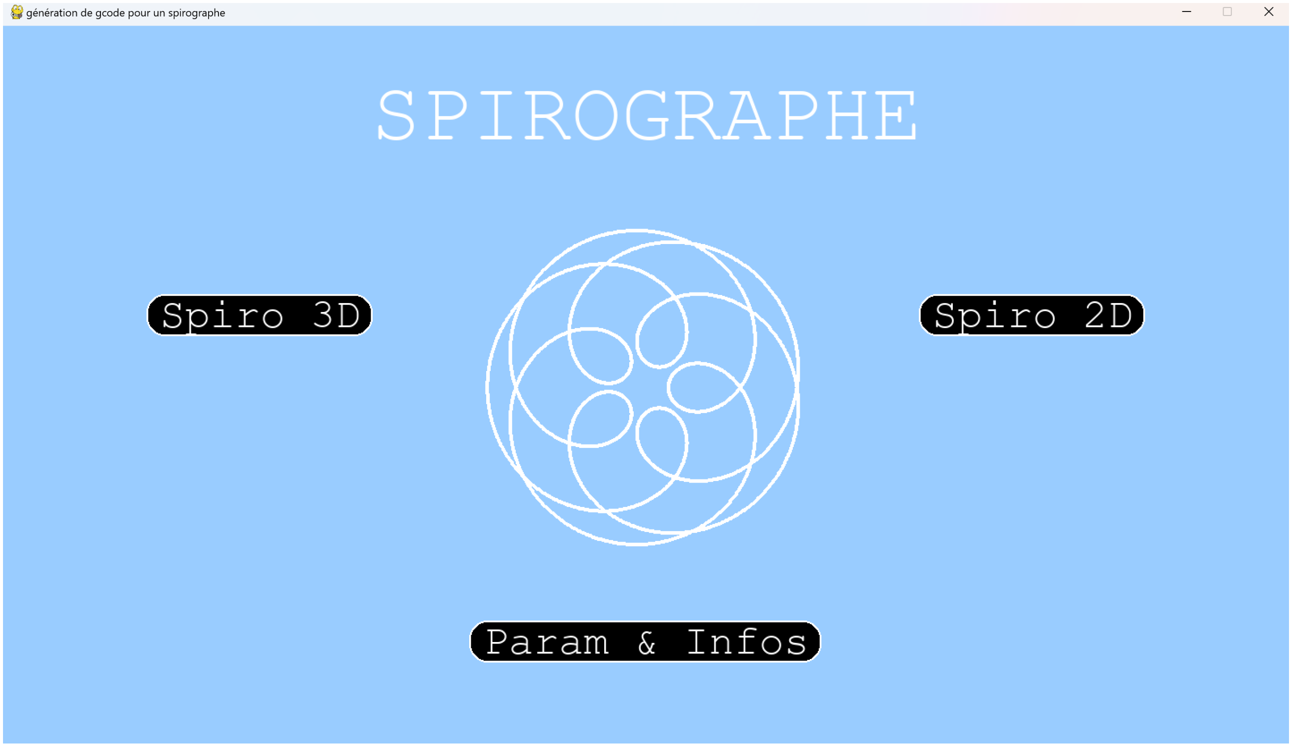
Fonctionalités :

- Ajout d'un champ pour définir le rayon S de la demi-sphère sur laquelle on projette le motif
- mise à jour dynamique de la valeur maximale de S à partir de laquelle on a un rendu correct
- bouton pour afficher une vue de dessus et une vue en coupe du motif paramétré
- image de sphère pour indiquer à l'utilisateur que le schéma va se déployer sur une sphère

Autres menus

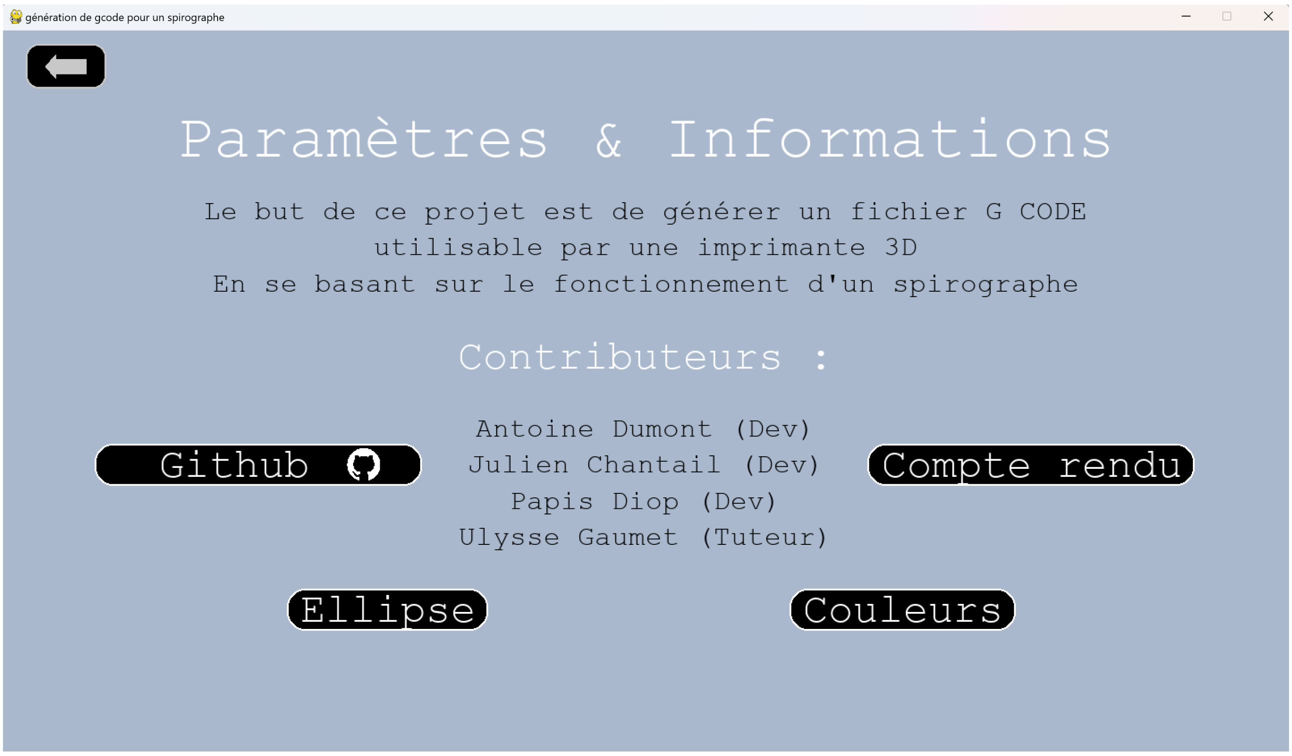
Menu Principal

Menu pour choisir le mode, ou pour aller dans les paramètres



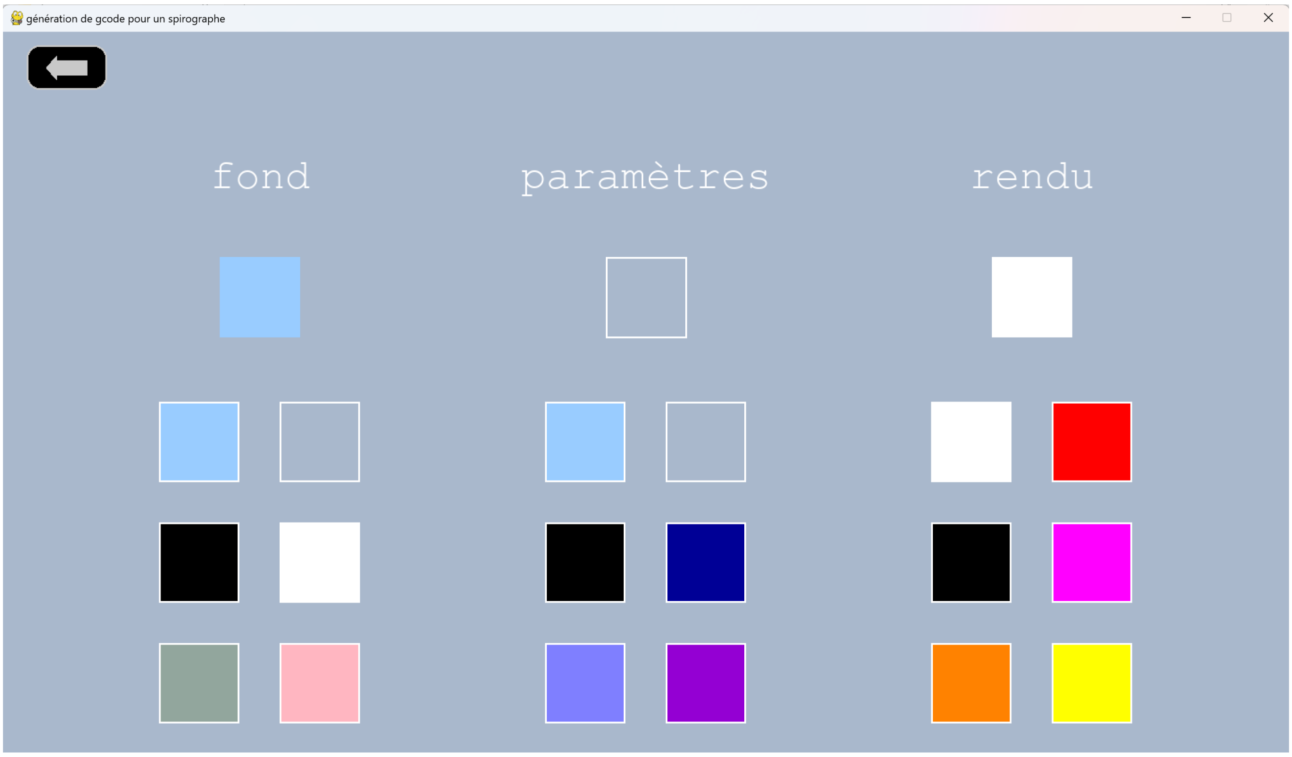
Menu paramètres et informations

Ce menu permet d'accéder à ce rapport, au code source en libre accès sur github ainsi qu'à l'interface couleurs. Le menu ellipse lui est un vestige de nos essais mathématiques.

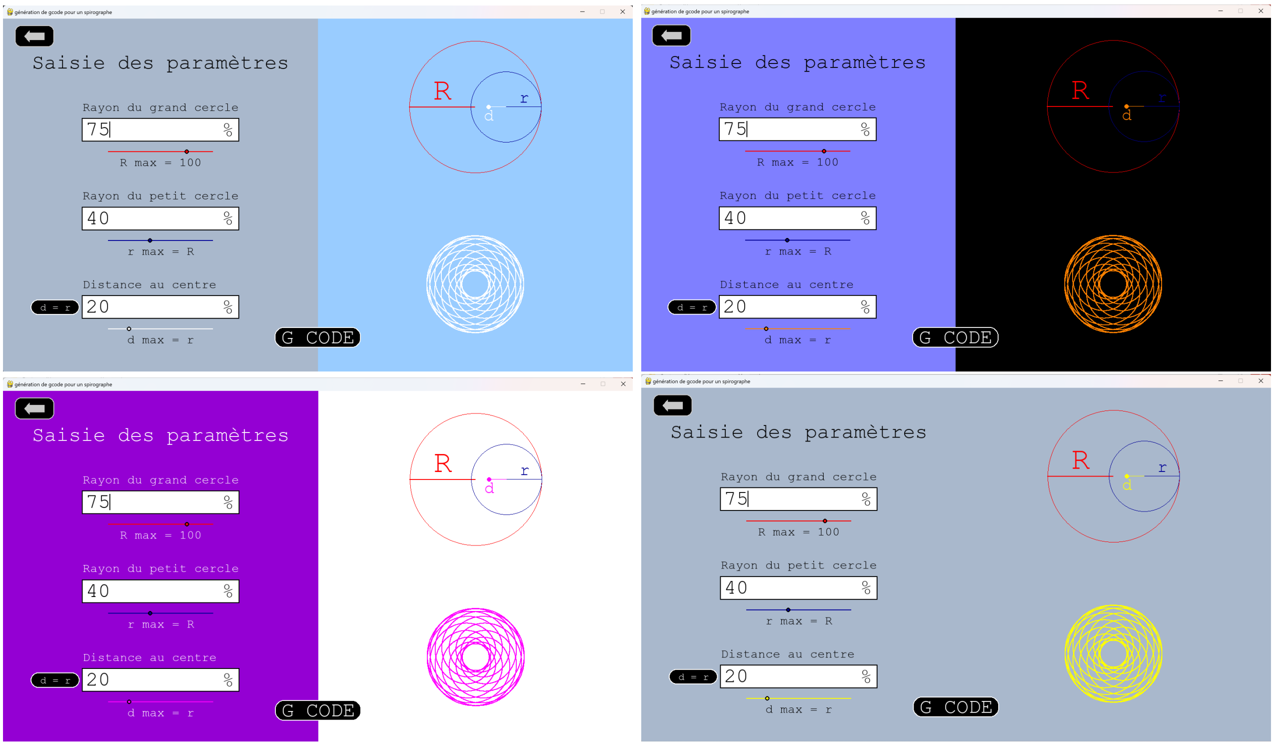


Menu Couleurs

Ce menu permet de personnaliser l'interface Les données sont stockées dans un fichier json, et garde en mémoire les préférences de l'utilisateur



Différentes configurations :



4. G CODE

G CODE : qu'est-ce que c'est ?

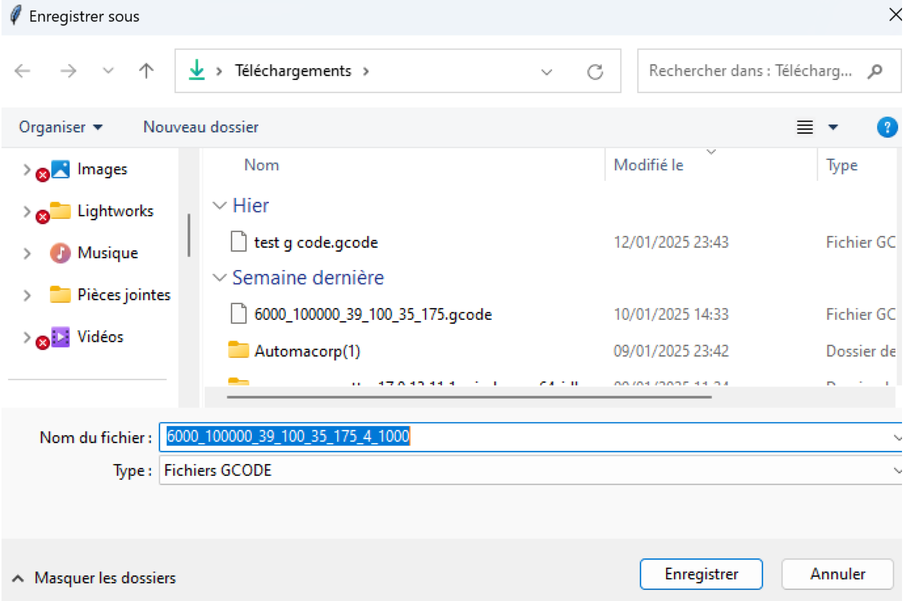
Le G CODE est le langage compris par une imprimante 3D pour construire une pièce. il ressemble à ça :

```
; Début du fichier G-code
G21 ; Unités en millimètres
G90 ; Mode de positionnement absolu
G28 ; Aller à l'origine
G92 E0 ; Réinitialiser l'extrusion
G1 Z0.2 F300 ; Début à la hauteur de couche initiale
G0 X96.000000 Y0.000000 Z0.000000 ; Déplacement initial
G1 X95.736216 Y0.378012 Z0.172242 E0.0246
G1 X94.946616 Y0.771725 Z0.683705 E0.0756
G1 X93.636438 Y1.196632 Z1.518877 E0.1561
G1 X91.814367 Y1.667817 Z2.652791 E0.2660
G1 X89.492471 Y2.199749 Z4.052258 E0.4041
G1 X86.686108 Y2.806095 Z5.677395 E0.5691
G1 X83.413806 Y3.499527 Z7.483318 E0.7592
G1 X79.697127 Y4.291548 Z9.421863 E0.9725
G1 X75.560501 Y5.192330 Z11.443225 E1.2070
G1 X71.031041 Y6.210557 Z13.497435 E1.4609
G1 X66.138337 Y7.353295 Z15.535644 E1.7320
G1 X60.914234 Y8.625868 Z17.511187 E2.0184
G1 X55.392591 Y10.031759 Z19.380449 E2.3182
G1 X49.609027 Y11.572522 Z21.103540 E2.6296
G1 X43.600655 Y13.247723 Z22.644810 E2.9509
G1 X37.405803 Y15.054893 Z23.973239 E3.2803
G1 X31.063731 Y16.989508 Z25.062719 E3.6163
G1 X24.614340 Y19.044987 Z25.892256 E3.9573
G1 X18.097878 Y21.212712 Z26.446108 E4.3018
G1 X11.554644 Y23.482068 Z26.713881 E4.6483
G1 X5.024696 Y25.840510 Z26.690583 E4.9954
G1 X-1.452438 Y28.273646 Z26.376646 E5.3417
G1 X-7.838045 Y30.765335 Z25.777929 E5.6858
```

Nous avons utilisé un site en ligne pour vérifier que notre code généré était correct :

implémentation

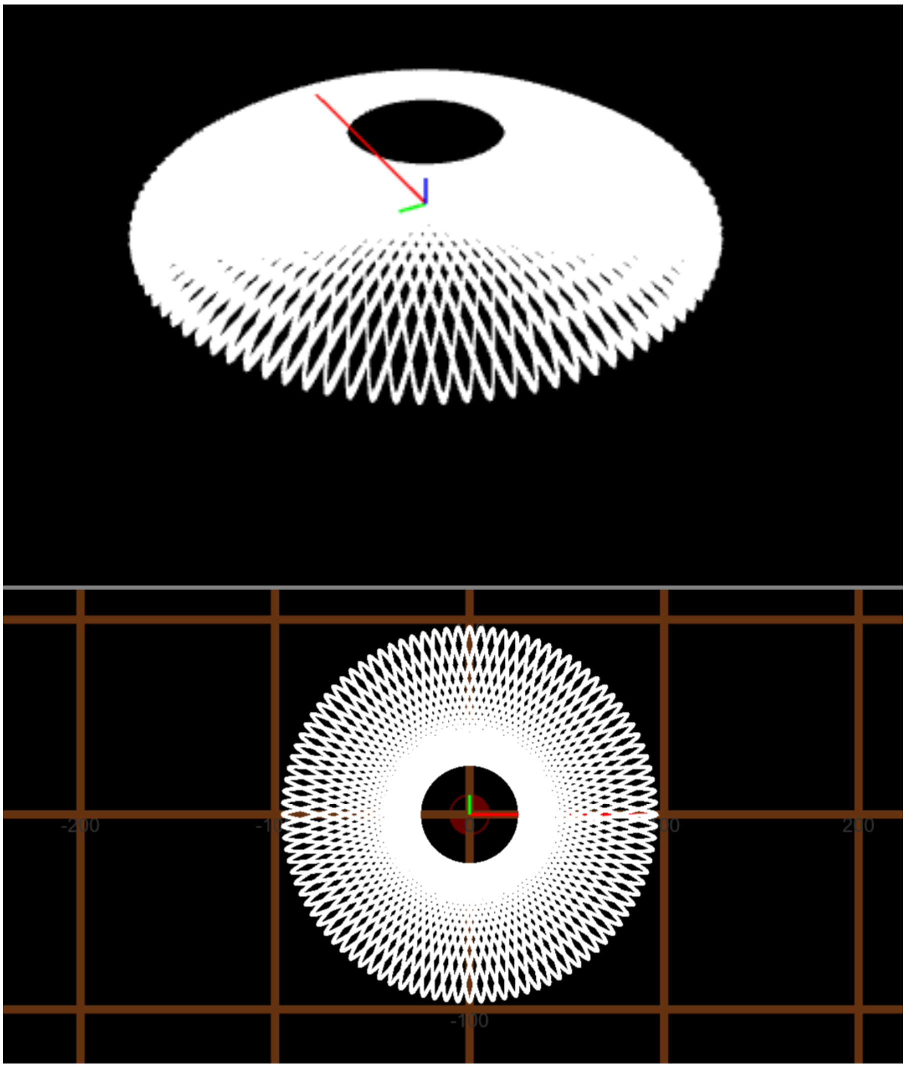
à l'aide de la bibliothèque tkinter de Python, le logiciel peut demander à l'utilisateur où il souhaite enregistrer son code :



Les calculs mathématiques nous permettent d'avoir les coordonnées cartésiennes des points du motifs, c'est donc facile de transformer ça en coordonnées G CODE, en format absolu.

test du g code

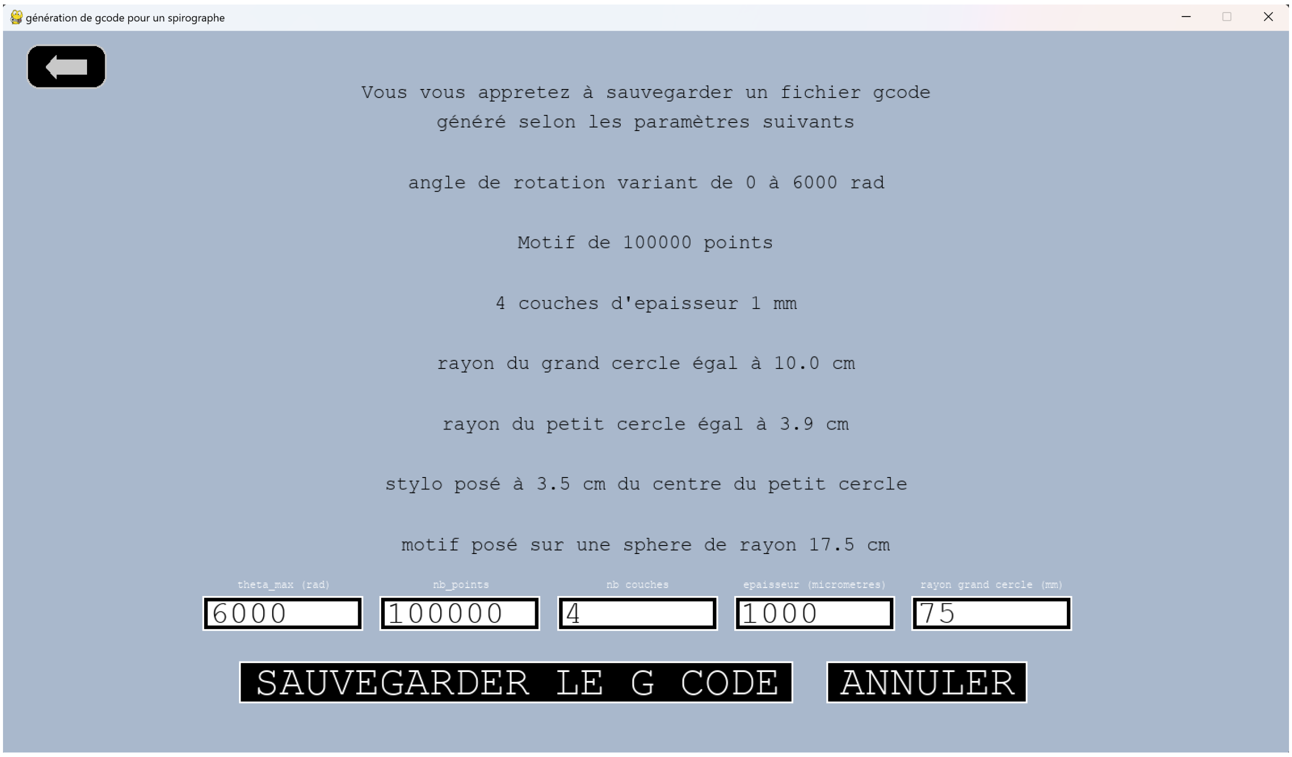
Nous pouvons donc tester et admirer nos motifs :



interface de paramétrage

Afin de permettre à l'utilisateur de paramétrer comme il le souhaite son motif, une interface de paramétrage lui permet de remplir les données suivantes :

- θ max : le nombre de tour que va faire notre spirographe. Plus le motif est compliqué, plus il faut un θ max grand pour avoir l'entièreté du motif. Un θ max inférieur permet également de dessiner un motif plus simple
- nb_points : le nombre de point dont sera composé chaque couche
- nb couches : le nombre de couches qui se superposeront pour former la pièce
- epaisseur (μ m) : l'épaisseur d'une couche (cela correspond entre la différence de position z d'un point du motif sur 2 couches adjacentes
- rayon grand cercle (sur lequel sont indexés proportionnellement tous les autres paramètres choisis par l'utilisateur précedemment



5. Exécution

Le but de ce logiciel est de pouvoir être utilisé partout, même sur un ordinateur qui ne dispose pas de python, et encore moins des librairies complexes que nous avons utilisé telles que pygame

Nous avons donc créé à partir du code python un fichier exécutable, grâce à la commande :

```
pyinstaller
    --onefile --add-data "ressources/github.png;ressources"
    --add-data "ressources/sphere.png;ressources"
    --add-data "ressources/project_tech__spirographe-4.pdf;ressources"
    "logiciel spirographe.py"
```

```
logiciel spirographe.exe
1 fichier(s)       26 731 807 octets
```

Le logiciel, à chaque exécution, va chercher les informations suivantes dans un fichier json "données/data.json" que l'application crée la première fois quelle est lancée.

- les 3 couleurs
- les paramètres du menu 2D
- les paramètres du menu 3D