

1. INTRO

L'objectif de ce projet est de dessiner des formes dans une fenêtre grâce à la bibliothèque graphique SFML, grâce à une série de mots clés. Nous verrons donc dans un premier temps ces mots clés, puis les résultats que l'on peut obtenir en les utilisant voire en les combinant. Après cela, il y aura une rapide explication du code derrière ces affichages.

2. COMMANDES

1.1 ENTREES CONSOLES

J'ai créé des fonctions de base acceptant certains paramètres dans un ordre interchangeable. Ces fonctions permettent de tracer des formes complexes et intéressantes.

❖ DRAW

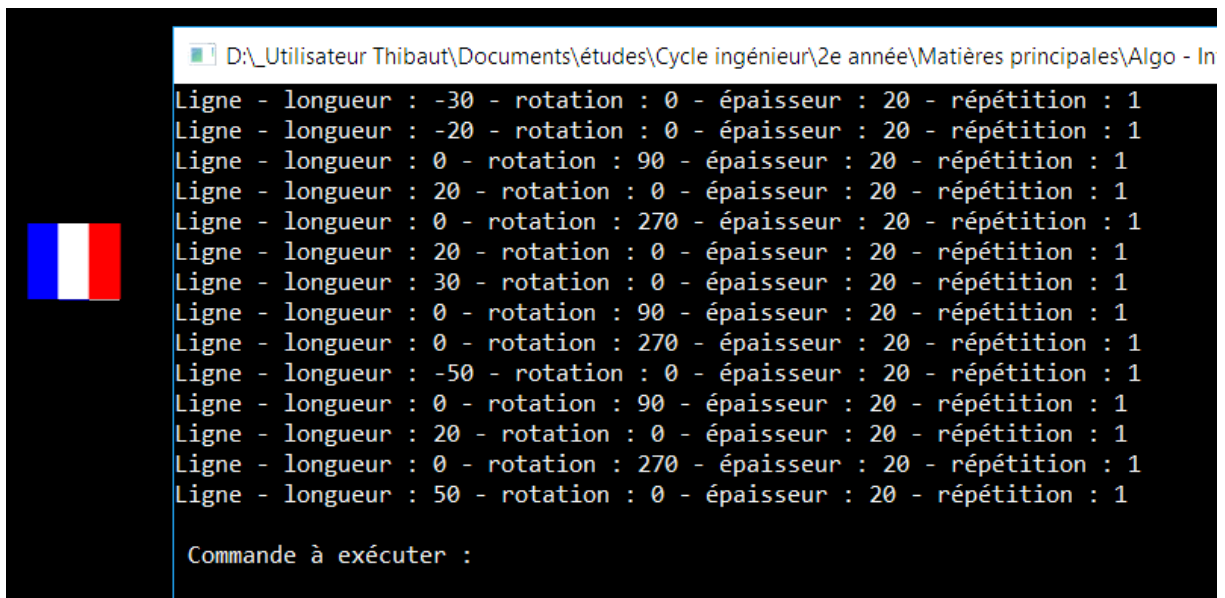
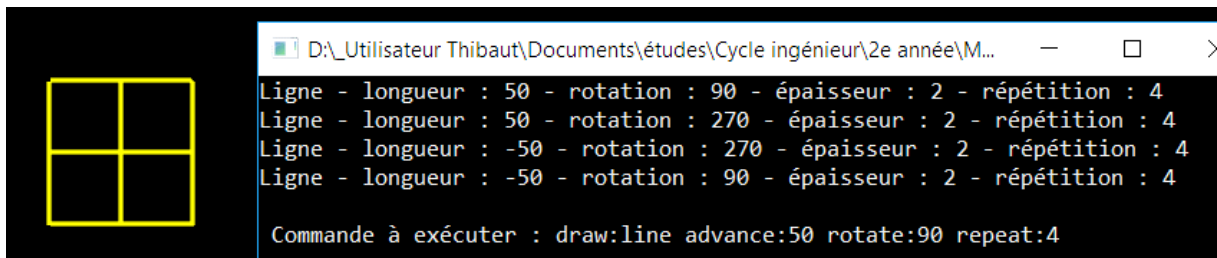
Ensemble	Description	Commandes utilisables	Valeur par défaut	Description des commandes
Line	Trace une ligne simple	<ul style="list-style-type: none">▪ advance : réel▪ rotate : réel▪ repeat : entier positif▪ thickness : entier positif▪ color : nom (red, green...)▪ color : entier-entier-entier	<ul style="list-style-type: none">▪ 0▪ 0▪ 1▪ 2▪ yellow	<ul style="list-style-type: none">▪ Déplacement en pixels▪ Rotation après tracé▪ Nombre de reproduction▪ Epaisseur de ce trait▪ Nouvelle couleur de tracé▪ Code couleur RGB
Spiral	Trace une forme ronde exacte	<ul style="list-style-type: none">▪ radius : réel positif▪ revolution : réel▪ variation : réel▪ wrap : booléen▪ repeat▪ color	<ul style="list-style-type: none">▪ 0▪ 1▪ 1▪ false▪ 1▪ yellow	<ul style="list-style-type: none">▪ Rayon de la spirale à tracer▪ Nombre de révolution (%)▪ Evolution du rayon▪ Resserre la spirale
Mosaic	Trace une forme ronde spéciale	<ul style="list-style-type: none">▪ radius▪ revolution▪ variation▪ wrap▪ repeat▪ color	<ul style="list-style-type: none">▪ 0▪ 1▪ 1▪ false▪ 1▪ yellow	
Fractal	Trace une fractale	<ul style="list-style-type: none">▪ name : mot▪ iterate : entier positif▪ advance	<ul style="list-style-type: none">▪ error▪ 0▪ 0	<ul style="list-style-type: none">▪ Nom de la fractale▪ Profondeur récursive

❖ CURSOR

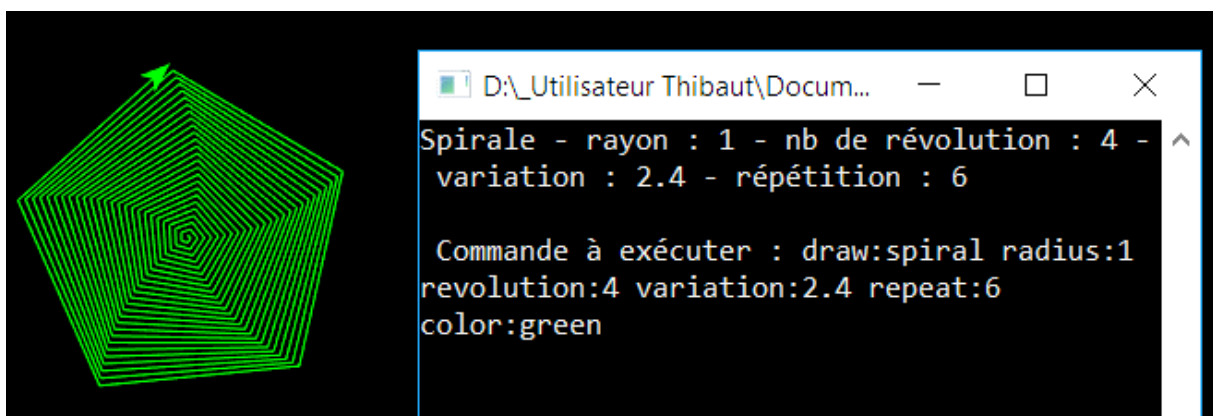
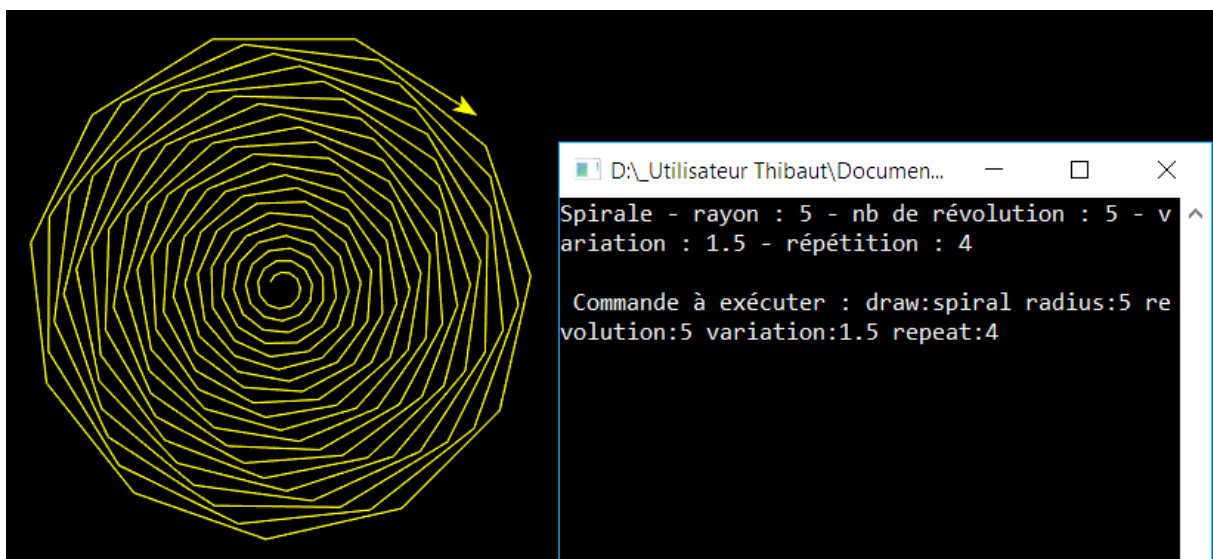
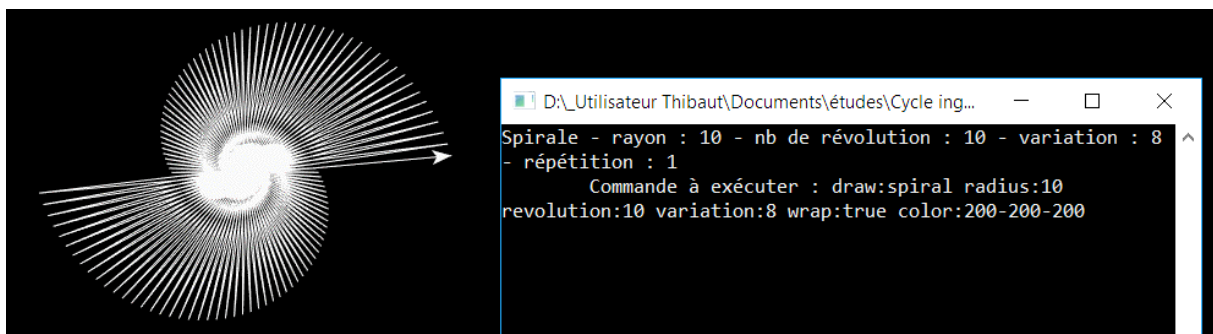
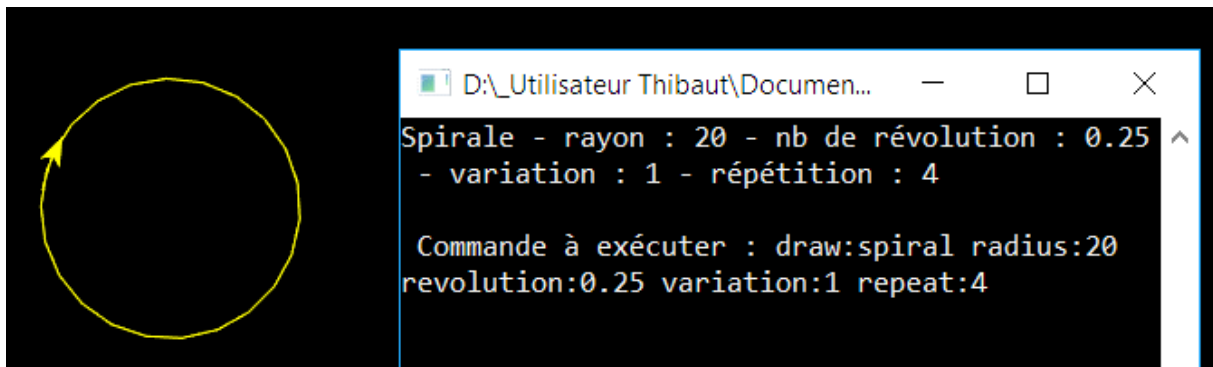
Ensemble	Description	Commandes utilisables	Valeur par défaut	Description des commandes
Move	Etat actuel du curseur	<ul style="list-style-type: none"> rotate : réel xCoord : réel yCoord : réel 	<ul style="list-style-type: none"> 0 0 0 	<ul style="list-style-type: none"> Rotation relative du curseur Déplacement relatif en largeur Déplacement relatif en hauteur
Style	Affichage du curseur	<ul style="list-style-type: none"> visible : show / hide clearLast : booléen clear : booléen thickness : entier positif color 	<ul style="list-style-type: none"> show false false 2 yellow 	<ul style="list-style-type: none"> Affichage du curseur Supprime dernière forme Supprime toutes les formes Epaisseur du trait (line / fractal)

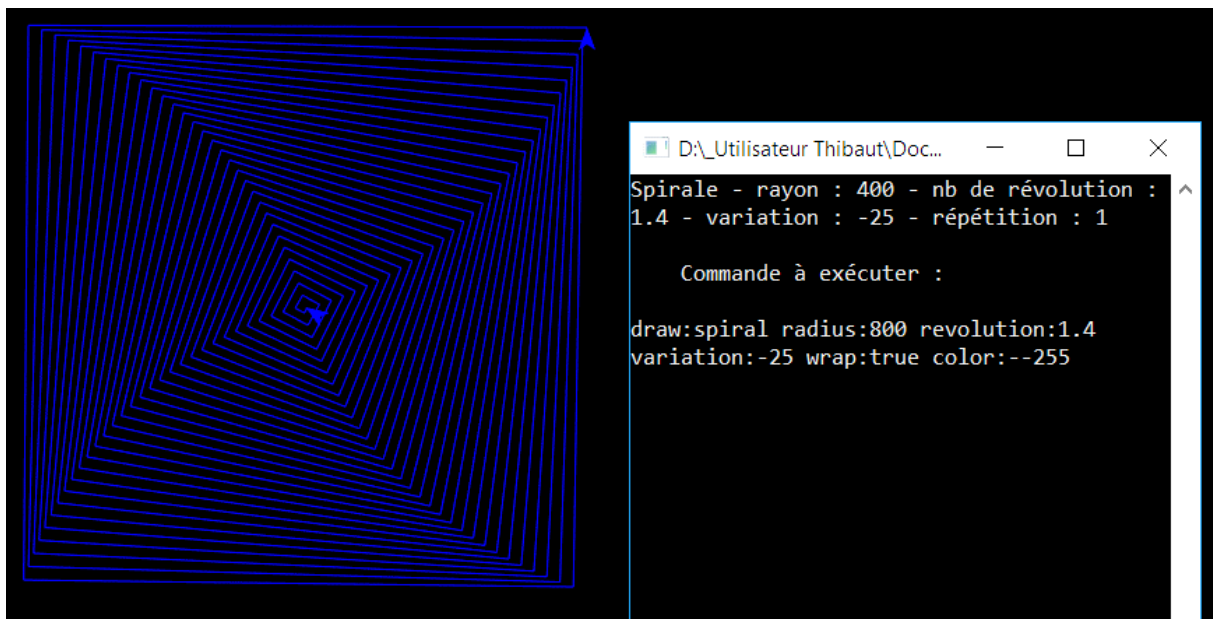
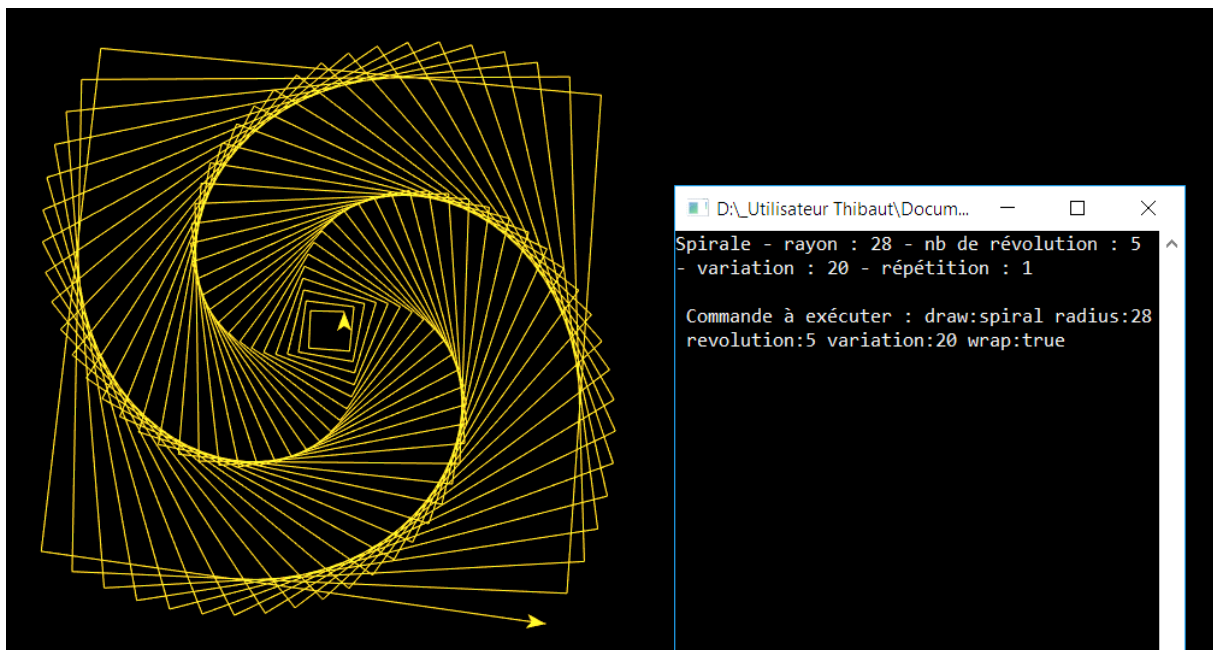
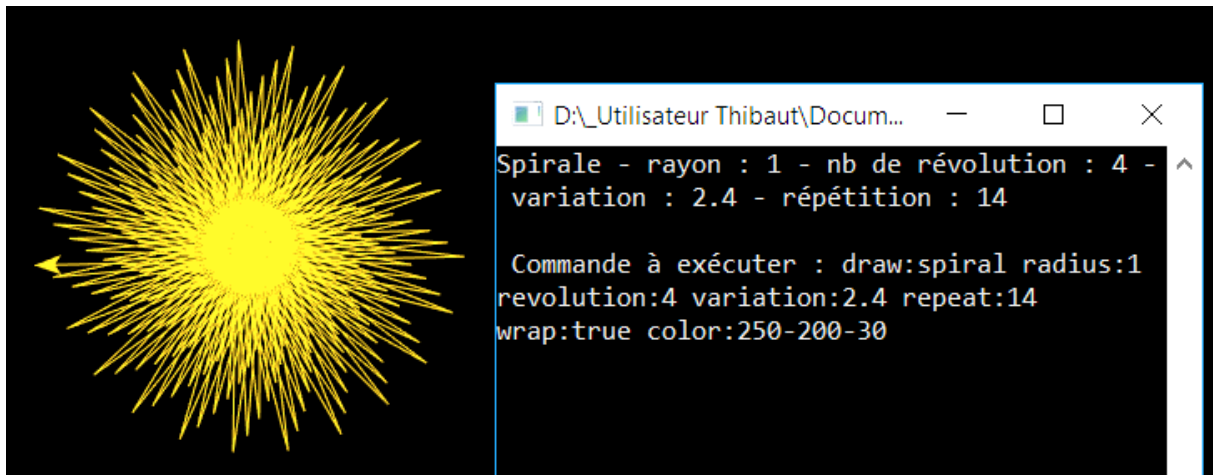
1.2 EXEMPLES ET ILLUSTRATIONS

❖ **Lignes** : simple, basique. Travailler sur l'épaisseur permet de faire du « pixelArt ».

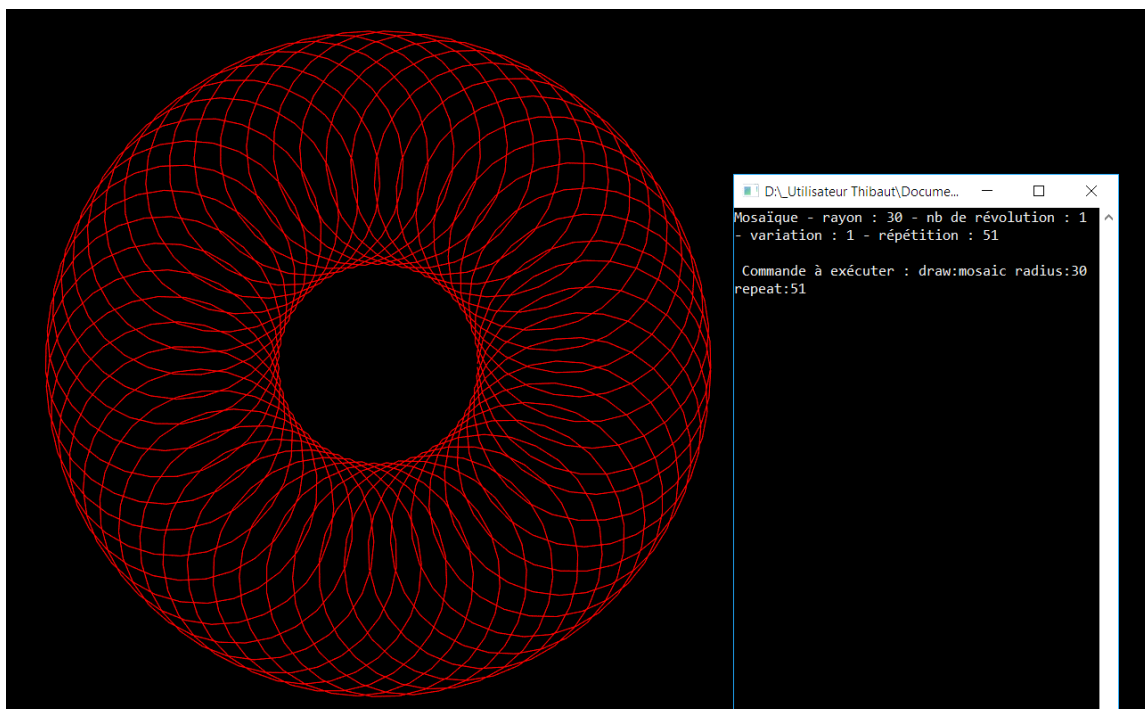
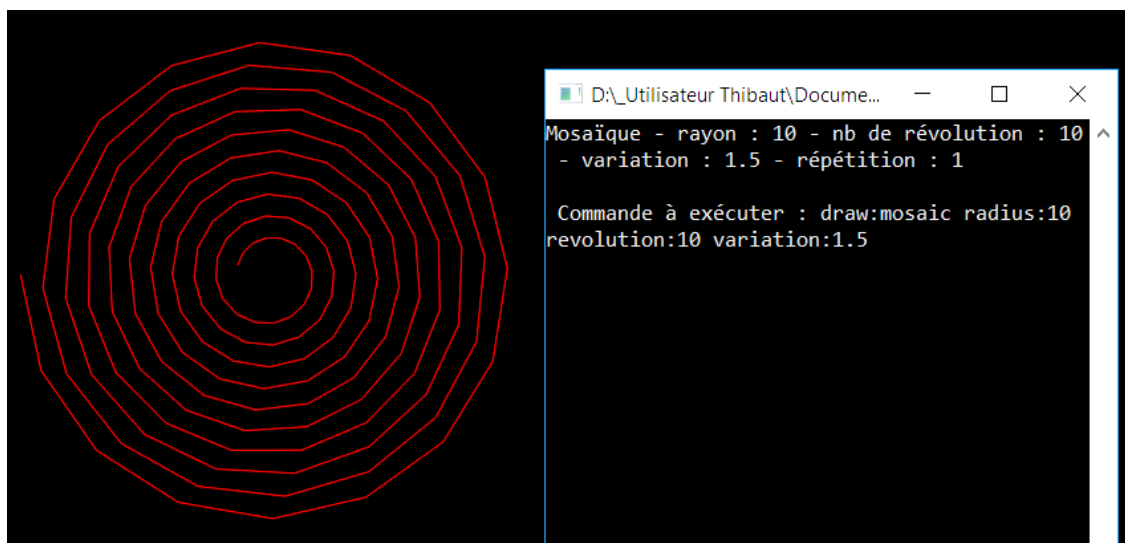
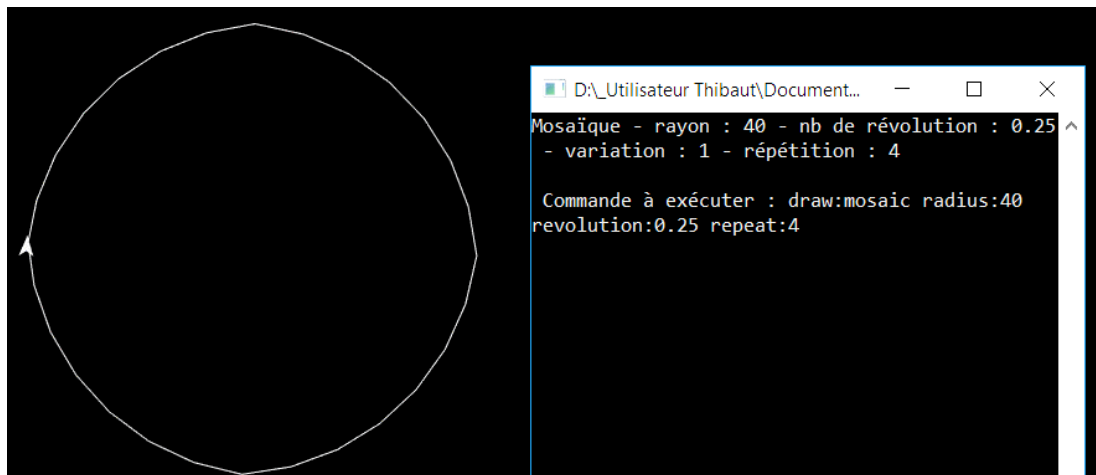


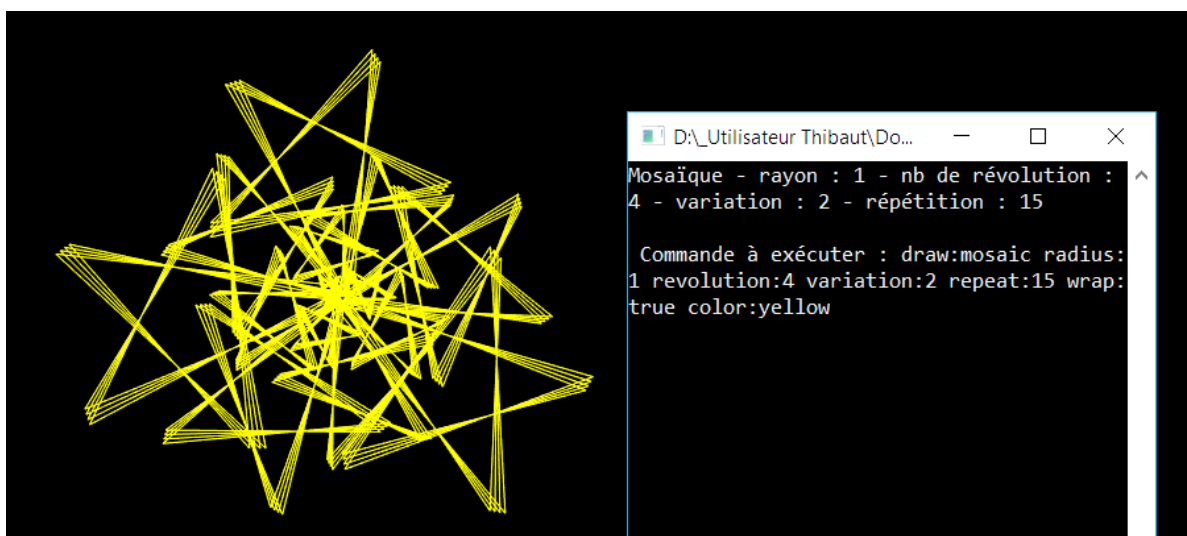
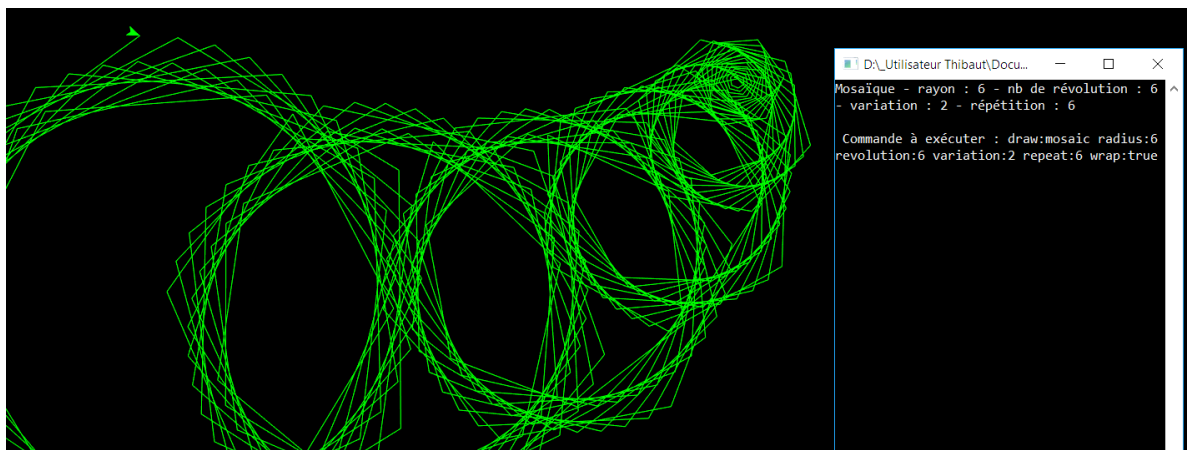
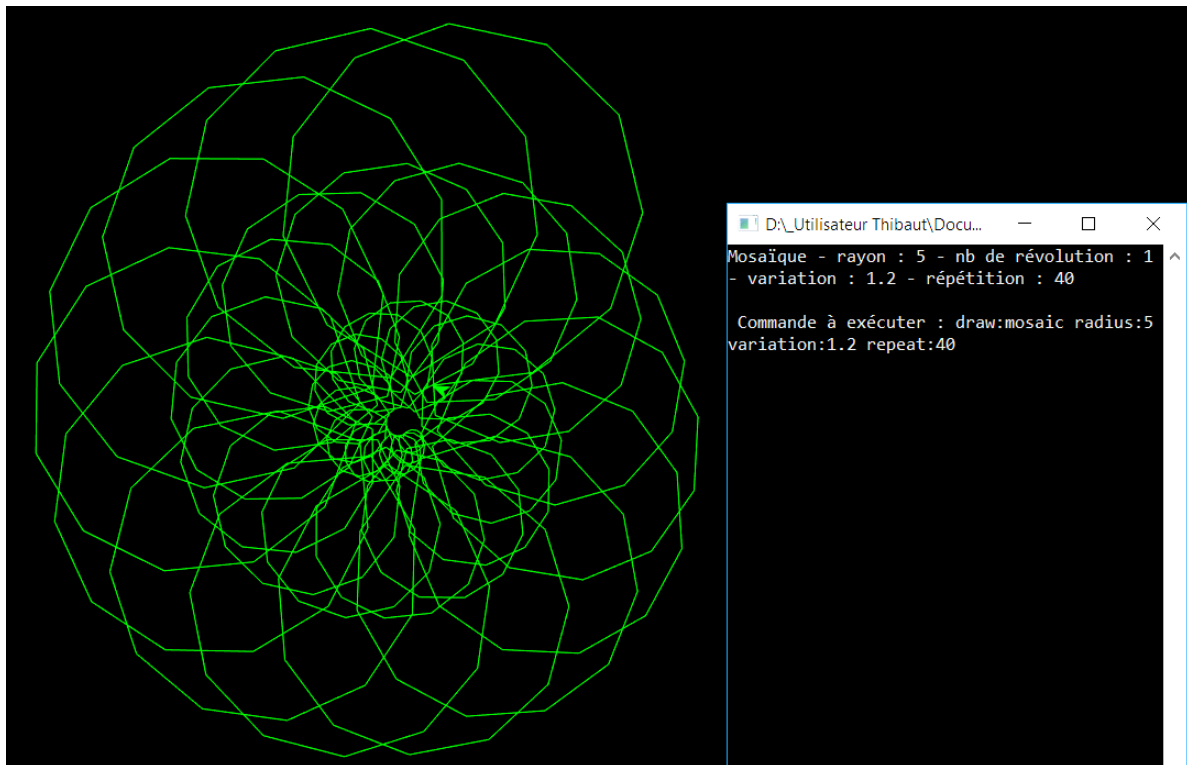
❖ **Spirales** : généralement simples. Obtenir une forme « spéciale » reste compliquée.

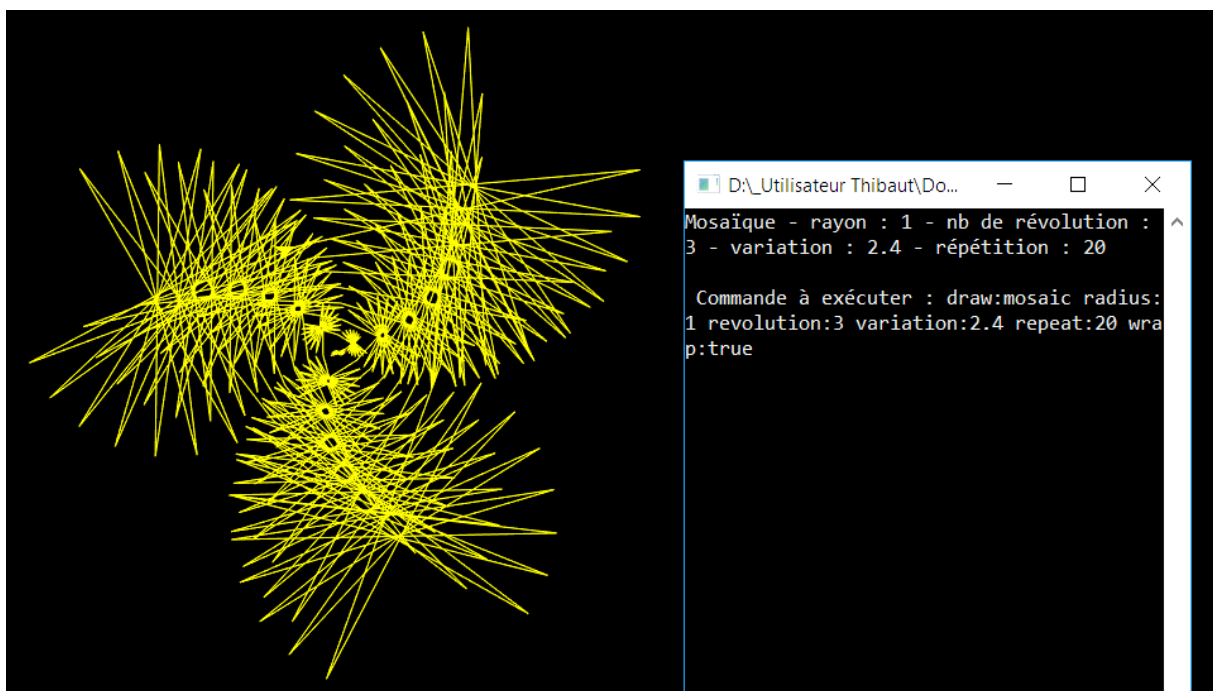
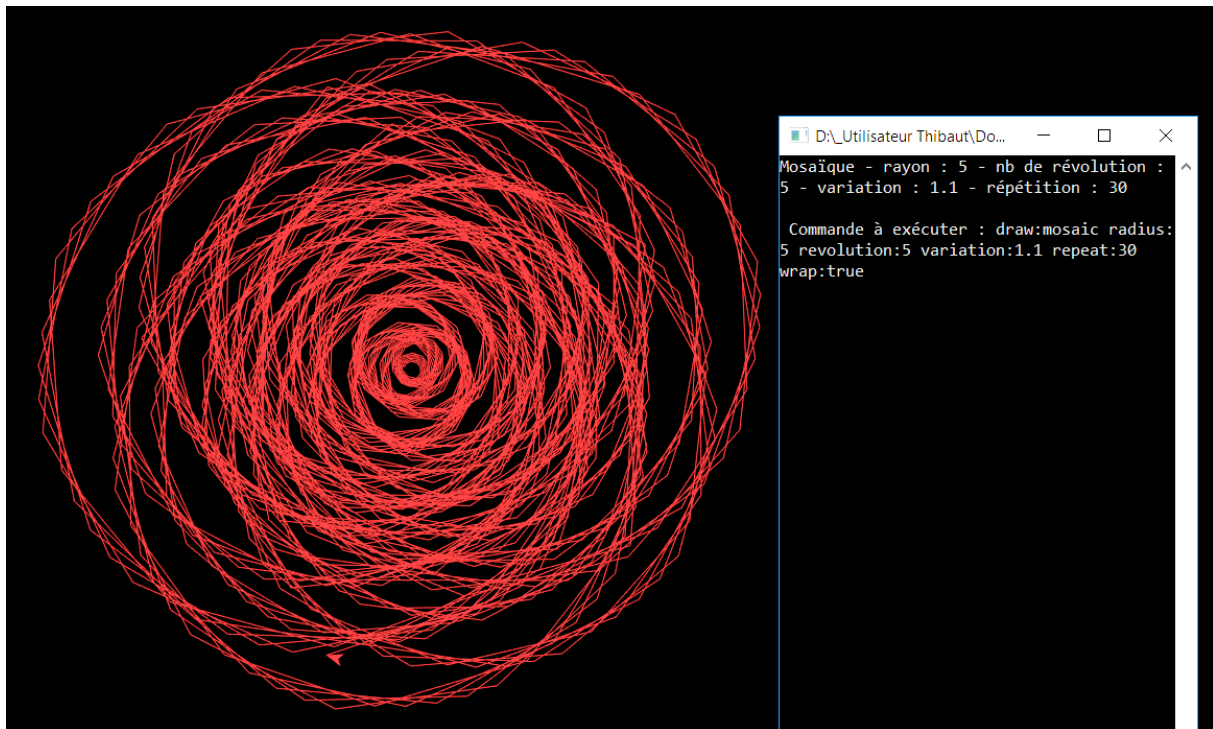




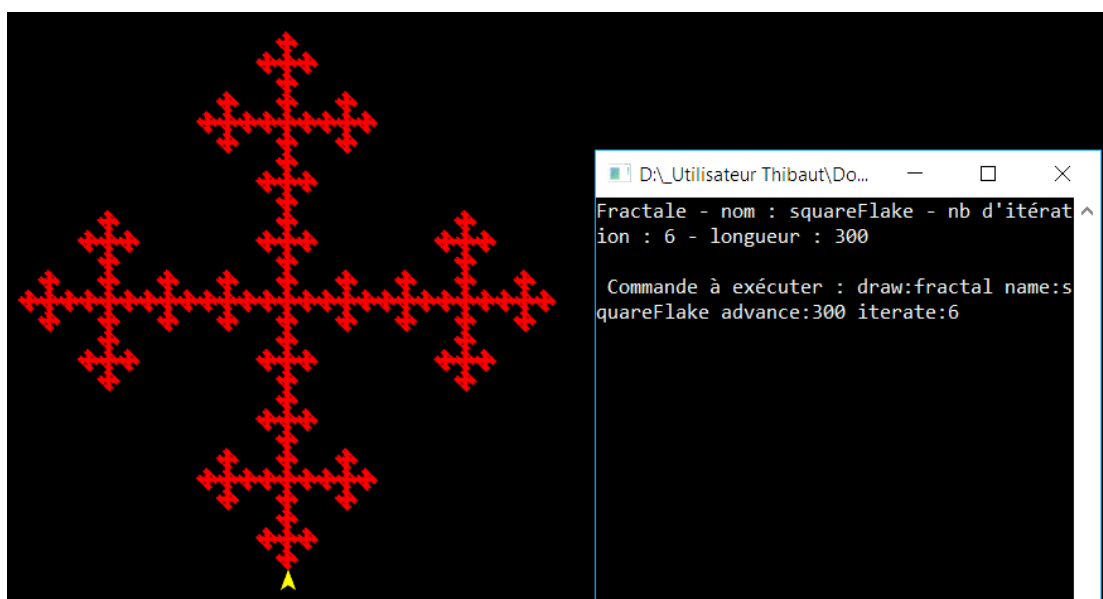
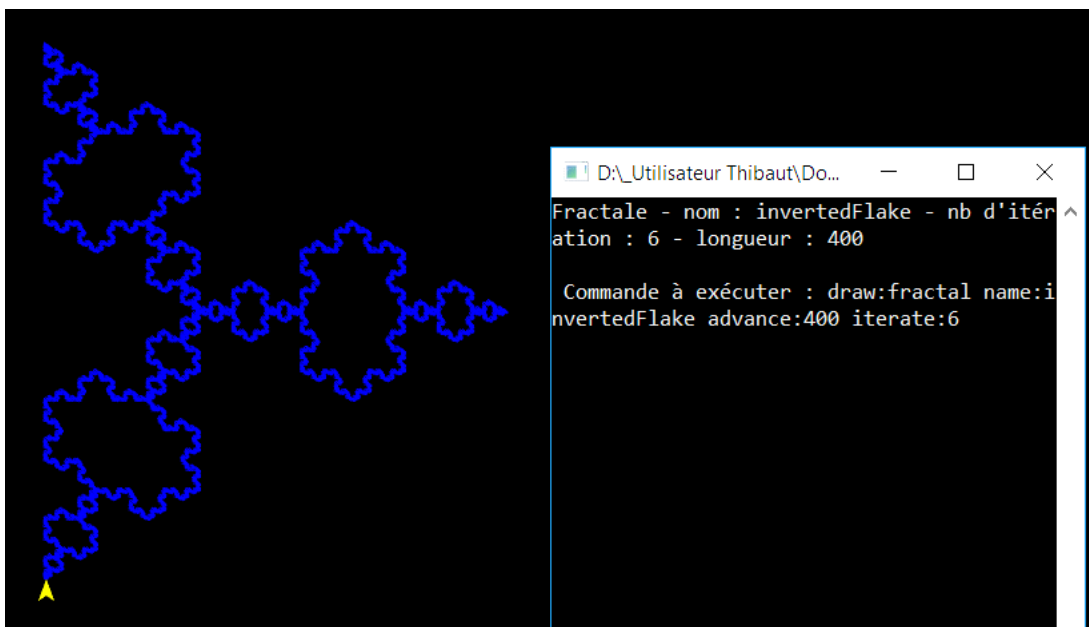
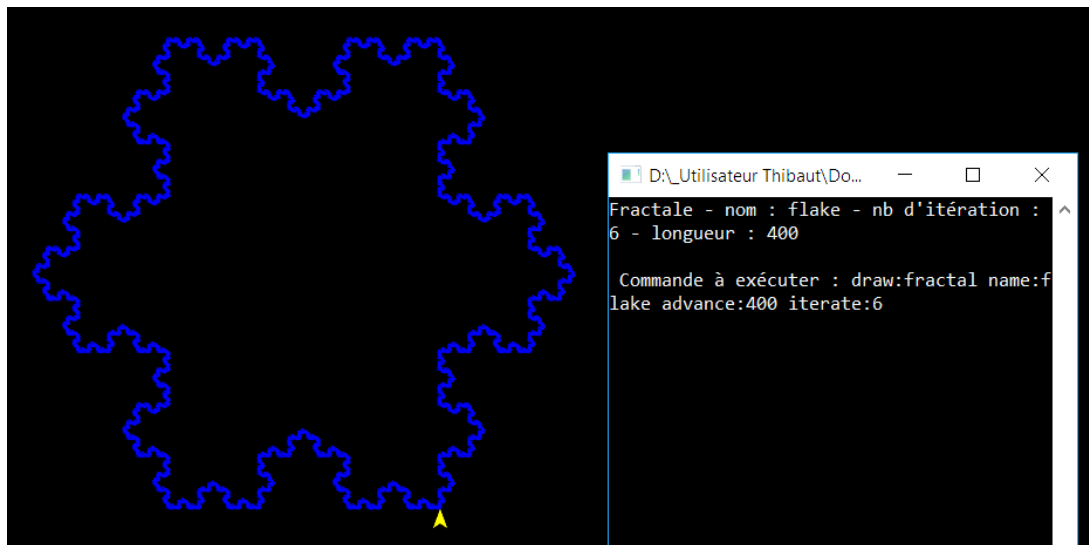
❖ **Mosaïques** : spirales dont le léger décalage génère des formes « spéciales » plus facilement.

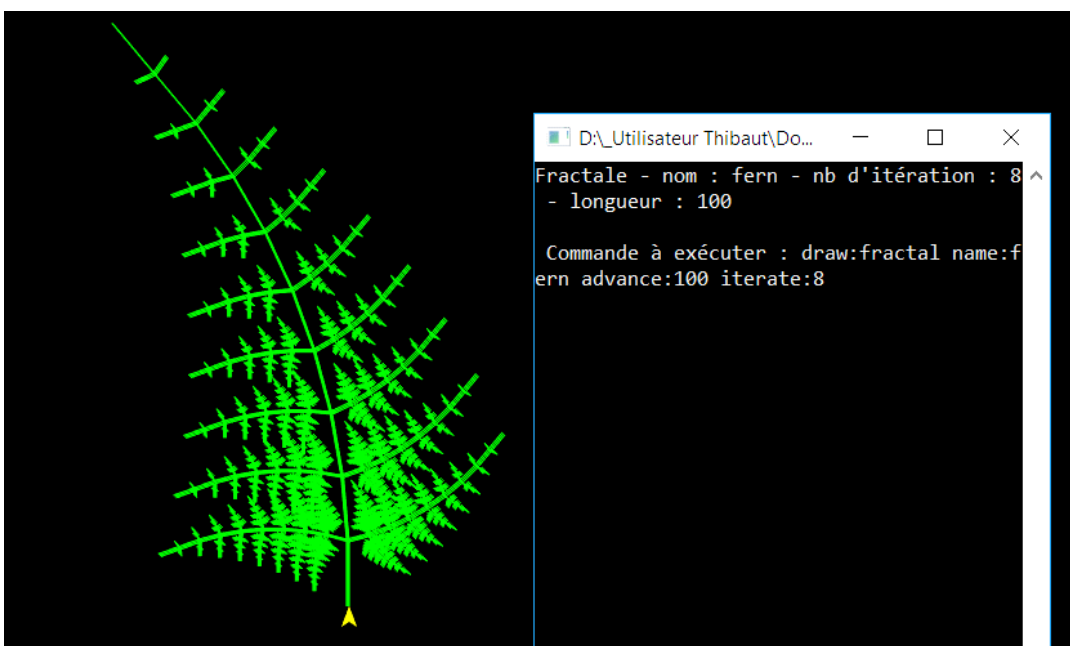
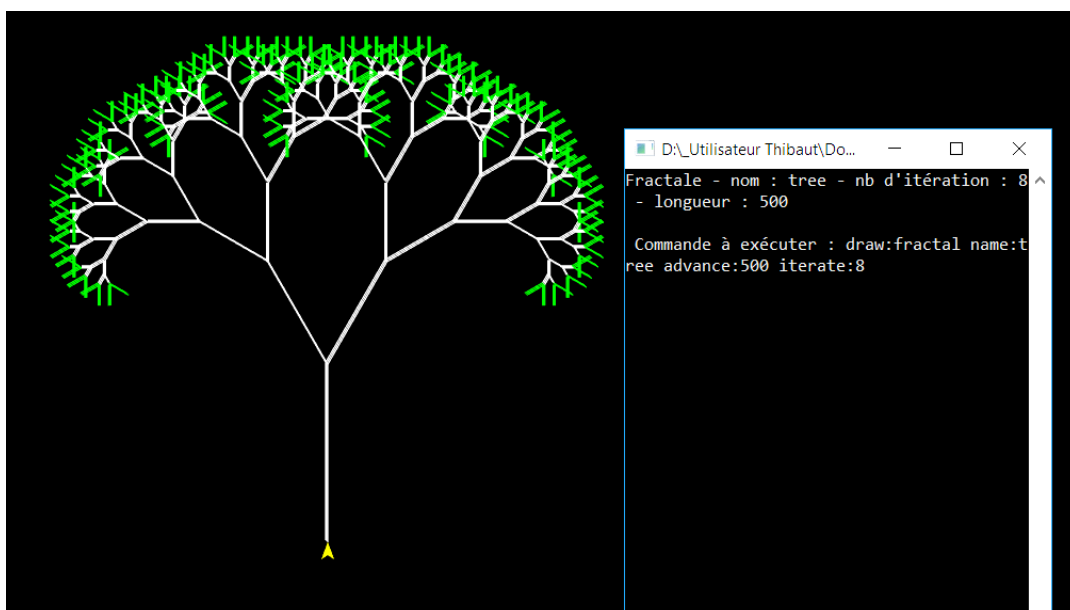
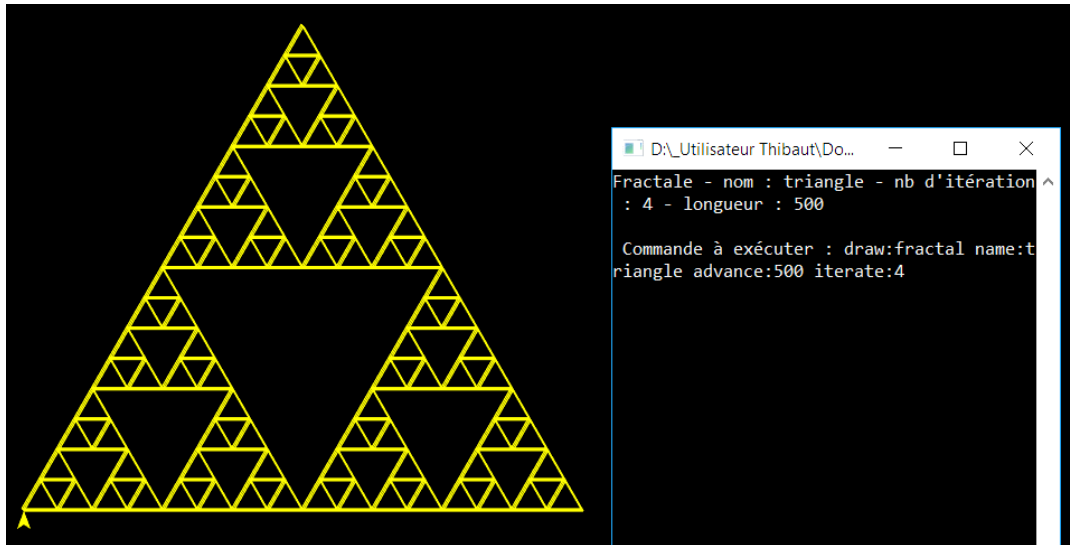






❖ **Fractales** : Complexes mais impressionnantes.





Ensemble de commandes à tester, en partie illustrées précédemment :

❖ Lignes :

draw:line advance:90 rotate:90 repeat:4	=> carré (pour damier, utilise toute les combinaisons de positif et négatif)
draw:line advance:20 rotate:20 repeat:18	=> cercle (peu pratique)

❖ Spirales :

Continuité :

draw:spiral radius:40 revolution:0.25 variation:1 repeat:4 => maîtrise (tester les combinaisons de positif et négatif pour l'éprouver)

Prolongement :

draw:spiral radius:60	revolution:20	variation:-1.68		=> spirale intérieure
draw:spiral radius:5	revolution:5	variation:1.5	repeat:10	=> spirale "mosaïque"
draw:spiral radius:5	revolution:5	variation:1.5	repeat:10 wrap:true	=> spirale "mosaïque" 3D
draw:spiral radius:1	revolution:4	variation:2.4	repeat:20	=> pentagone
draw:spiral radius:1	revolution:4	variation:2.4	repeat:20 wrap:true	=> mini soleil
draw:spiral radius:30	revolution:10	variation:8	wrap:true	=> étoile / lotus
draw:spiral radius:28	revolution:10	variation:20	wrap:true	=> escalier
draw:spiral radius:800	revolution:1.4	variation:-25	wrap:true	=> carré plein

❖ Mosaïques :

Continuité :

draw:mosaic radius:40	revolution:0.25	repeat:4	=> léger décalages (but différent)
draw:mosaic radius:10	revolution:20	variation:1.5	=> spirale
draw:mosaic radius:10	revolution:40	variation:1.8	wrap:true => spirale "mosaïque"
draw:mosaic radius:30	revolution:10	variation:8	wrap:true => étoile / lotus
draw:mosaic radius:40	revolution:10	variation:8	wrap:true => escalier
draw:mosaic radius:800	revolution:1.4	variation:-25	wrap:true => carré plein

Prolongement :

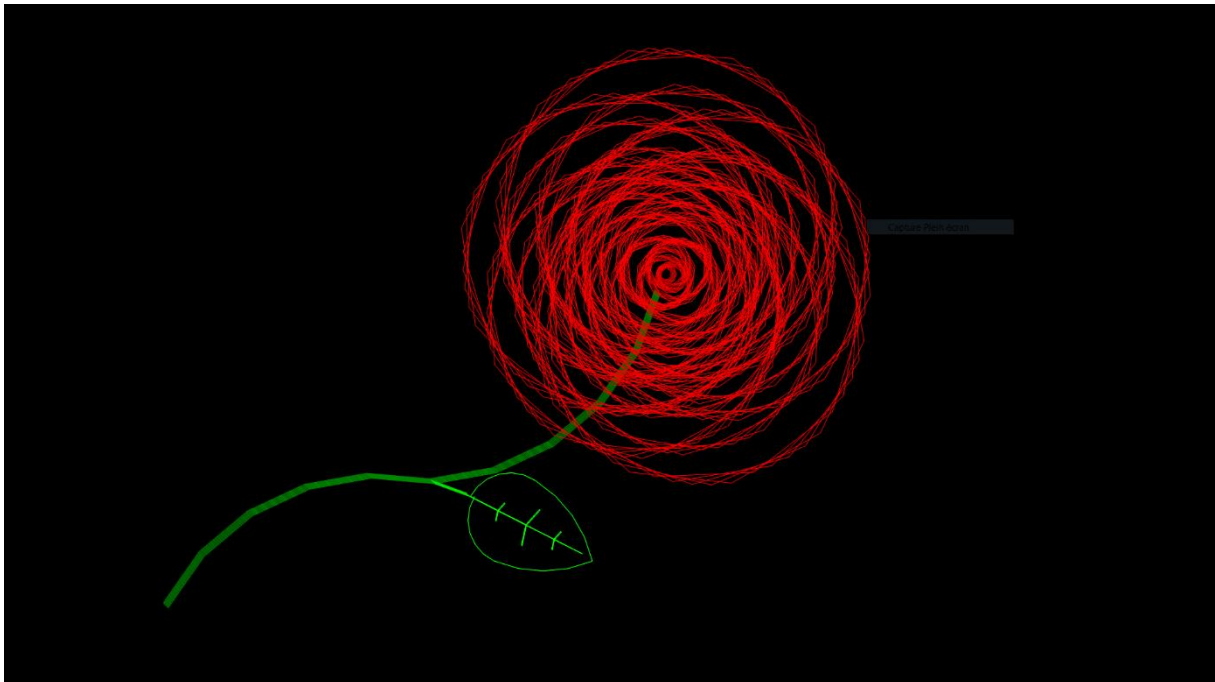
draw:mosaic radius:36	repeat:66	=> rosace		
draw:mosaic radius:30	repeat:51	=> tore		
draw:mosaic radius:5	revolution:5	variation:1.5	repeat:10	=> tourbillon
draw:mosaic radius:6	revolution:6	variation:2	repeat:6	wrap:true => tourbillon 2
draw:mosaic radius:5	revolution:5	variation:1.5	repeat:10	wrap:true => fleur épaisse
draw:mosaic radius:5	variation:1.2	repeat:40	=> escargot	
draw:mosaic radius:5	revolution:5	variation:1.1	repeat:30	=> fleur escargot
draw:mosaic radius:5	revolution:5	variation:1.1	repeat:30	wrap:true => rose
draw:mosaic radius:5	variation:1.2	repeat:40	wrap:true	=> fleur "triangle"
draw:mosaic radius:10	revolution:30	variation:2.4	wrap:true	=> fleur multitude petites pétales
draw:mosaic radius:6	revolution:6	variation:2.1	repeat:15	wrap:true => petite tornade
draw:mosaic radius:6	revolution:6	variation:2.4	repeat:20	wrap:true => moyenne tornade
draw:mosaic radius:10	revolution:6	variation:2.4	repeat:20	wrap:true => grosse tornade
draw:mosaic radius:1	revolution:6	variation:2.4	repeat:14	=> fleur pétales pentagonales
draw:mosaic radius:1	revolution:6	variation:2.4	repeat:14	wrap:true => arc étoilé
draw:mosaic radius:1	revolution:4	variation:2	repeat:15	wrap:true => étoiles condensées
draw:mosaic radius:1	revolution:3	variation:2.4	repeat:50	=> tourbillon de pétales pentagonales
draw:mosaic radius:1	revolution:3	variation:2.4	repeat:50	wrap:true => tourbillon d'étoiles
draw:mosaic radius:20	revolution:6	variation:12	repeat:4	wrap:true => constellation

❖ Fractales :

Prologement :

draw:fractal name:flake	advance:400	iterate:4
draw:fractal name:invertedFlake	advance:400	iterate:4
draw:fractal name:squareFlake	advance:300	iterate:4
draw:fractal name:triangle	advance:500	iterate:4
draw:fractal name:tree	advance:500	iterate:8
draw:fractal name:fern	advance:100	iterate:8

Il est aussi possible de mixer les fonctions de base pour obtenir de beaux résultats :



```

Ligne - longueur : 100 - rotation : 15 - épaisseur : 8 - répétition : 5
Ligne - longueur : 100 - rotation : 345 - épaisseur : 8 - répétition : 5
Ligne - longueur : 0 - rotation : 15 - épaisseur : 2 - répétition : 1
Ligne - longueur : -100 - rotation : 15 - épaisseur : 8 - répétition : 5
Ligne - longueur : -60 - rotation : 187 - épaisseur : 6 - répétition : 1
Fractale - nom : fern - nb d'itération : 0 - longueur : 100
Ligne - longueur : 100 - rotation : 0 - épaisseur : 2 - répétition : 1
Fractale - nom : fern - nb d'itération : 0 - longueur : 100
Ligne - longueur : -100 - rotation : 0 - épaisseur : 2 - répétition : 1
Fractale - nom : fern - nb d'itération : 0 - longueur : 200
Ligne - longueur : 0 - rotation : 263 - épaisseur : 2 - répétition : 1
Spirale - rayon : 20 - nb de révolution : 0.25 - variation : 1 - répétition : 1
Spirale - rayon : 40 - nb de révolution : 0.1 - variation : 1 - répétition : 1
Ligne - longueur : 0 - rotation : 80 - épaisseur : 2 - répétition : 1
Spirale - rayon : 40 - nb de révolution : 0.12 - variation : 1 - répétition : 1
Spirale - rayon : 20 - nb de révolution : 0.25 - variation : 1 - répétition : 1
Mosaïque - rayon : 5 - nb de révolution : 5 - variation : 1.1 - répétition : 30

```

cursor:style	visible:hide color:-100- thickness:8
cursor:move	rotate:110 xCoord:80 yCoord:-80
draw:line	advance:100 rotate:15 repeat:5
draw:line	advance:100 rotate:-15 repeat:5
draw:line	rotate:15
draw:line	advance:-100 rotate:15 repeat:5
draw:line	rotate:10
draw:line	advance:-60 rotate:187 thickness:6 color:-180-
draw:fractal	name:fern advance:100
draw:line	advance:100
draw:fractal	name:fern advance:100
draw:line	advance:-100
draw:fractal	name:fern advance:200
draw:line	rotate:-97 color:green
draw:spiral	radius:20 revolution:.25
draw:spiral	radius:40 revolution:.1
draw:line	rotate:80
draw:spiral	radius:40 revolution:.12
draw:spiral	radius:20 revolution:.25
cursor:move	xCoord:306 yCoord:-360 color:200--
draw:mosaic	radius:5 revolution:5 variation:1.1 repeat:30 wrap:true

3. PROGRAMMATION

3.1 EXPLICATION DES CLASSES

Le programme contient relativement peu de classes, ce qui donne une architecture simple à appréhender et maîtriser :

- ❖ DrawCursor : Classe gérant le curseur : position d'origine, déplacement, couleur...
- ❖ AsyncGetline : Classe trouvée sur internet pour opérer des lectures de flux non bloquants, grâce à un thread parallèle. Cette classe m'affranchit de la lecture en fichier et me permet une interaction avec l'utilisateur plus forte.
- ❖ DrawIO : Classe dédiée aux flux (entrants). Conserve notamment les entrées utilisateur.
- ❖ Draw : Classe de calcul et d'impression des formes. Exécute entrées reçues puis affiche le résultat.

3.2 FIERTES ET REGRETS

Je regrette de n'avoir pas eu le temps d'appliquer toutes mes idées. J'ai perdu un certain temps en développant une architecture plus complexe en cours de route, sans parvenir aux mêmes résultats qu'avec celle-ci. Ce temps perdu aurait pu être utilisé pour faire fonctionner la refonte du système d'entrées, plus sécurisé et optimisé, ou encore opérer des enregistrements en fichier de macros – les x dernières entrées -, réutilisable ensuite. De même, j'aurais aimé intégrer un mécanisme de variables. Enfin, j'avais aperçu la possibilité d'enregistrer le contenu de la fenêtre en fichier. L'idée me plaisait beaucoup, mais faute de temps, elle n'a pas été implémentée.

Cependant, pouvoir interagir directement avec l'utilisateur via la console est un parti pris que je suis fier de pouvoir proposer. L'interaction avec l'utilisateur est largement renforcée, puisqu'elle est dynamique. Même si ce n'est pas directement mon code, j'ai pu appréhender un mécanisme puissant.

Aussi, pouvoir proposer des fractales est une jolie réussite pour moi. En effet, le temps de réflexion nécessaire au développement de ces formes n'est pas anodin, mais le résultat est à la hauteur du temps dépensé.

Enfin, je suis fier d'avoir développé une map associant un string à un pointeur de fonction. Ce mécanisme m'évite une série de « if else » et offre sans doute un gain de performance. Cependant, je n'ai pas pu pousser le concept au bout et y intégrer directement les méthodes de classe.

3.3 PISTES D'AMELIORATION

- ✓ Enregistrer des captures de la fenêtre
- ✓ (Re)sécuriser et optimiser les entées
- ✓ Compter la présence de mots clés à valeur booléenne comme « true »
- ✓ Ajouter une commande « help » qui afficherait les commandes et leur valeur par défaut
- ✓ Réintégrer l'ensemble de commandes cursor :origin, modifiant les valeurs du point de départ du curseur et gérer des animations (rotation) sans s'appuyer sur le temps de calcul des objets
- ✓ Faire fonctionner l'ensemble « modify », permettant de modifier les objets
- ✓ Réaliser des lectures et écritures en fichier, supportés par un gestionnaire de fichier
- ✓ Implémenter des variables aux lignes de commandes
- ✓ Intégrer l'interaction avec l'utilisateur à la fenêtre SFML