



Le musée des fractales

Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

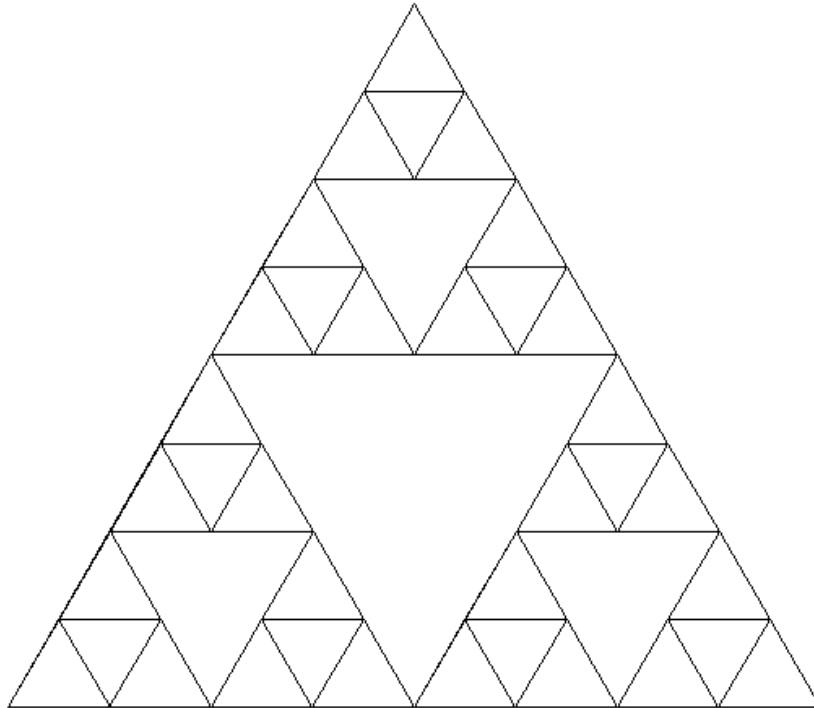
Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à info@trophees-nsi.fr.

NOM DU PROJET : LE MUSEE DES FRACTALES

> PRÉSENTATION GÉNÉRALE :

L'idée générale est d'utiliser la récursivité que nous avons abordé dans le cadre d'un chapitre du programme de Terminale NSI.

La notion de fractale nous est venue naturellement grâce à des illustrations de nos fiches de cours, en effet elles permettent de représenter clairement la notion de récursivité.



Notre problématique : déterminer quelles fractales utiliser qui soient belles et représentatives, savoir utiliser des outils adéquats de Python pour réaliser la partie graphique et au final réaliser un projet suffisamment construit et intéressant pour être en compétition avec d'autres équipes.

Nous avons donc plutôt opté pour plusieurs fractales (**le flocon de Von Koch**, **l'arbre récursif**, **le triangle de Sierpiński**, et **le carré récursif**) présentées dans une interface graphique intuitive à utiliser.

L'objectif était de créer une interface graphique pour l'utilisateur dans laquelle il peut choisir une fractale, de voir le rendu en observant le processus de tracé de la fractale, et ce à des vitesses et avec un nombre d'itérations paramétrables.

En même temps cela pouvait nous permettre de découvrir plus en profondeur la programmation en équipe, la programmation GUI ainsi que la récursivité qui est une notion très intéressante à nos yeux.

> ORGANISATION DU TRAVAIL :

Présentation de l'équipe et répartition des tâches :

Le groupe est constitué de Thibault LAVAL, Ugo MALLETT et Enzo CHETAH.
Dans ce groupe de 3 membres, nous n'avons malheureusement pas de fille avec nous.

Voici principalement la manière dont nous nous sommes réparti les tâches :

- Thibault a géré la programmation des fonctions récursives permettant de tracer les fractales.
- Ugo a réalisé l'illustration et a travaillé sur l'organisation
- Enzo s'est occupé de l'interface graphique et de la gestion des événements (clics et actions des boutons) et de la cohérence de fonctionnement du programme

Organisation et outils utilisé :

Pour réaliser ce projet, nous avons travaillé à la maison mais aussi pendant des séances de NSI. Notre professeur nous accompagnait et cadrait notre travail.

Nous avons essentiellement communiqué sur Instagram via un groupe prévu pour l'occasion.

Les réunions en présentiel se faisaient pendant les séances de NSI car nous ne sommes pas dans la même classe de Terminale.

Les programmes ont été réalisés sur EduPython et les dessins ont été réalisé grâce à l'application IBIS PAINT sur IPAD.

Pour les fichiers Python et autres documents, nous avons pu les échanger par internet et en classe sur un disque de partage.

➤ LES ÉTAPES DU PROJET :

Voici les principales étapes pour la réalisation de notre projet (beaucoup ayant été travaillées en parallèle) :

- Etape 1 : lecture des consignes et du cahier des charges du projet
- Etape 2 : réflexion et choix de plusieurs idées de projets
- Etape 3 : choix du projet
- Etape 4 : établir un calendrier et répartition des tâches
- Etape 5 : élaboration de l'image
- Etape 6 : choix des motifs et réalisation des fonctions de tracé de fractales
- Etape 7 : réalisation de la partie graphique de Python
- Etape 8 : mise en commun
- Etape 9 : compléter le document descriptif
- Etape 10 : réaliser les prises de vues vidéos
- Etape 11 : réaliser la capsule vidéo (professeur)

Pour la partie Python :

- Etape 1 : Faire une interface graphique pouvant accueillir une fenêtre Turtle
- Etape 2 : Implémenter les fonctions récursives
- Etape 3 : Implémentation de bouton permettant de lancer les fonctions
- Etape 4 : Ajouter des options pour mieux contrôler le processus de création de la fractales
- Etape 5 : Ajout de la fenêtre d'information
- Etape 6 : Ajout des icônes sur les boutons permettant de choisir les fractales
- Etape 7 : Implémentation de l'illustration du projet dans la fenêtre d'info
- Etape 8 : Correction des bugs
- Etape 9 : Projet Terminé

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

5 avril 2022

L'interface graphique est globalement constituée, la positions des boutons ne sont pas réfléchit et ne sont illustrée que grâce à du texte. Seul la fonction Flocon de Koch est introduit dans le programme pour effectuer des tests d'affichage et de fonctionnement des boutons. Les fonctions permettant de créer d'autres fractales ne sont pas encore introduites dans le programme.

10 avril 2022

L'interface est plus finalisée, les boutons sont plus organisés mais aucun à part celui du Flocon de Koch n'est lié à une fonction, la fonction « dessin » n'a pas encore de choix multiple à faire. L'idée de pouvoir changer la vitesse et le nombre d'itérations est en cours de création. Des images sont ajoutées aux boutons pour un rendu plus esthétique.

14 avril 2022

L'interface graphique est retravaillée pour que les boutons soient plus logiquement positionnés pour l'utilisateur. La notion de valeur par défaut est mentionnée dans le groupe Instagram de l'équipe et est travaillé. Les boutons sont maintenant reliés à des fonctions permettant de les afficher sur l'interface. Un bug est repéré, les fonctions ne s'arrêtent pas lorsque le bouton Clear est utilisé. L'affichage des valeurs de vitesse et d'itérations est introduit dans le programme.

18 avril 2022

Une nouvelle fenêtre d'information est créée, l'image du groupe ne s'y affiche pas. La variable « etat » est créée afin d'éviter de lancer plusieurs fois le processus de création d'une fractale et pour arrêter une fonction en cours.

30 avril 2022

Tout fonctionne correctement, le bug de l'image dans la nouvelle fenêtre à été réglé grâce à la méthode `TopLevel()` de Tkinter. Deux règles ont été ajoutées, il est impossible de cliquer plusieurs fois sur le bouton Start après le premier lancement (afin d'éviter de dessiner plusieurs fois la même figure simultanément), après la figure complètement finie, il est possible d'appuyer directement sur Start pour lancer une nouvelle création (la fonction « dessin » appel la fonction « clear » pour effacer l'interface Turtle). Le programme est maintenant terminé.

Afin de détecter la présence de bugs, tous les options et règles étais tester à chaque teste. Tous les membres du groupe ont eu accès au fichier du programme afin qu'à chaque mise à jour il soit testé par l'ensemble de l'équipe. Ainsi, les anciens bugs étaient vérifiés et de nous étaient détectés.

Aucunes grandes difficultés n'ont été vraiment dérangeante, la documentation tkinter est à disposition en ligne et de nombreuses réponses à des problèmes ont été trouvable sur des forums tel que « Stack Overflow ». Les fractales ont été la seule difficulté due au temps de réflexion nécessaire afin de trouver une manière récursive de les tracés.

Globalement nous avons manqué de temps pour réaliser notre projet.

> OUVERTURE :

a. Idées d'améliorations

Voici quelques idées pour améliorer notre programme qui nous semblent intéressantes :

- ajouter d'autres motifs de fractales
- améliorer le rendu visuel global : ajouter des couleurs dans le tracé des fractales et insérer des images
- notre programme comporte environ 250 lignes de codes ce qui est raisonnable
Dans le cas d'un programme plus lourd et/ou dans un souci de clarté du code et d'utilisation des fonctions de tracé des motifs par d'autres programmes, on peut créer des programmes annexes qui comporteraient les tracés des fractales. C'est ce qu'on appelle la programmation modulaire.

b. Stratégie de diffusion

Voici quelques idées de média de diffusion afin de toucher un large public :

- il nous semble déjà important de faire une diffusion au sein de notre propre lycée via son site internet et son agenda Pronote que la grande majorité des élèves consultent
- à plus grande échelle, Pronote au niveau national pourrait diffuser au travers d'une bannière à thème sur le numérique (actuellement ce sont les Echechs)
- on pourrait diffuser une information de notre projet Trophée NSI sur les réseaux sociaux afin de diriger des personnes vers un site internet afin qu'ils puissent découvrir librement ce programme.

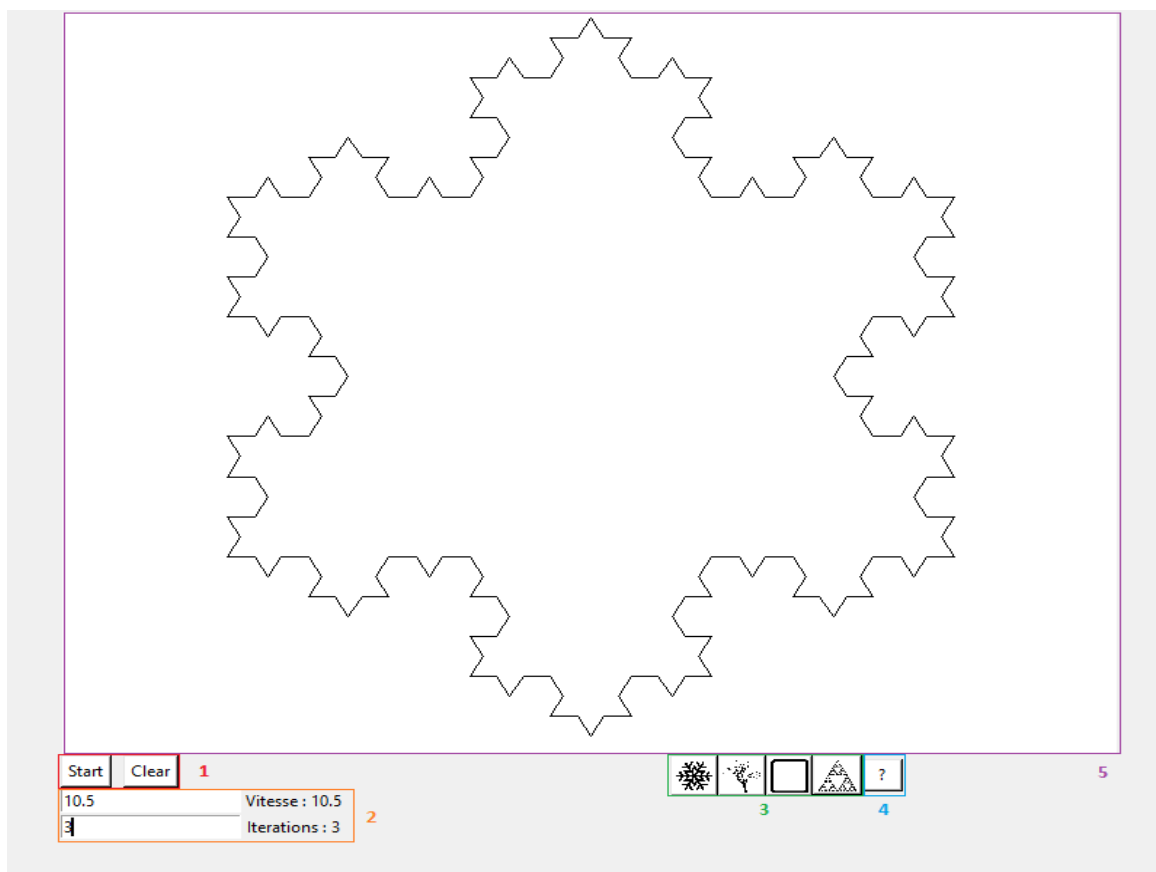
c. Analyse critique du résultat

Il nous a manqué du temps pour réaliser ce projet, nous aurions probablement travaillé un peu différemment dans un autre contexte.

En particulier, nous aurions pu établir une feuille de route initiale plus cohérente, réaliser des scripts Python plus clairs et optimisés et proposer un programme avec plus de fonctions.

DOCUMENTATION

Voici l'interface graphique du programme de tracés des fractales.



Dans la majeure partie de la fenêtre se trouve l'espace dédié au tracé de la figure et on peut découvrir ses fonctionnalités :

1 Lorsque l'interface graphique est lancée, le bouton **Start** lance le tracé d'une fractale (par défaut, le flocon de Koch). Le bouton **Clear** permet d'effacer et/ou arrêter une fractales en cours de création.

2 Deux **entrées** permettant de choisir, la vitesse de tracé ainsi que le nombre d'itérations souhaité. Celles-ci sont accompagnées **d'un label** à leur droite afin de voir la vitesse et le nombre d'itération précédemment choisis.

3 Dans le menu en bas à droite se trouvent **des boutons permettant de choisir le motif d'une fractale** ainsi qu'un **4 bouton avec un « ? »** qui donne un descriptif de ce qu'est une fractale ainsi que l'illustration du projet.

Lors de l'exécution du programme, les paramètres Tkinter (interface, boutons, entrées) puis les paramètres par défaut (**statue de vitesse et itération, fractale par défaut**, la tortue est cachée pour que le processus de création soit plus esthétique).

Les paramètres sont ensuite initialisés en fonction de l'action de l'utilisateur sur l'interface graphique et dessine la fractale demandée.

Le bouton Start appelle la fonction « dessin » qui trace une fractale avec Turtle, la fractale est choisie grâce à la variable « choix_fractales » qui se modifie avec **les boutons de sélections (3)**, La fonction s'assure que la zone de dessin soit vide en appelant la fonction « clear », définit l'« etat » du programme comme « en marche », définit un nombre d'itération puis l'envoie dans les paramètres de la fonction qui trace une fractale et initialise la vitesse avec « init_vitesse ».

```
def dessin():
    """Demarre le dessins choisi"""
    global etat
    if etat == False: # empeche les bugs de répétition
        clear() # S'assure que le canvas soit nettoyer
        etat = True
        fract = int(choix_fractales.get())
        init_vitesse() # Met a jours la vitesse de la turtle

    if fract == 1: 
    elif fract == 2: 
    elif fract == 3: 
    elif fract==4: 
```

Le bouton Clear appelle la fonction « clear ». Cette fonction permet d'effacer la fractale dessinée par la Turtle et de définir l'état du programme comme « à l'arrêt » en modifiant la variable « etat ».

```
def clear():
    """Efface le dessins"""
    global etat
    etat = False # Reinitialise l'etat du programme
    t.reset()
    t.hideturtle() # Supprime l'affichage du curseur de la turtle
```


La première entrée vitesse permet de rentrer une variable qui est récupérée ensuite dans la fonction « init_vitesse » afin d'initialiser une vitesse à la Turtle, si aucune vitesse n'est entrée alors la vitesse par défaut est initialisée à 5. **Le label vitesse** est ensuite modifié afin d'afficher la vitesse utilisée.

La seconde entrée itération permet de rentrer une variable d'itération qui est ensuite récupérée lors de l'appel de la fonction « dessin » afin d'attribuer un nombre d'itération. Si aucune itération n'est définie alors une itération par défaut est attribué grâce au paramètre « default », si le paramètre « default » n'est pas utilisé alors la variable par défaut est 0. **Le label itération** est ensuite modifié afin d'afficher le nombre d'itérations de la fractale.

```
def init_vitesse():
    """initialise tous les paramtre lié à la vitesse """
    if str(entrer_vitesse.get()) != "":
        t.speed(float(entrer_vitesse.get()))
        statu_vitesseVar.set("Vitesse : {}".format(entrer_vitesse.get())) # Met à jour le label de statu de vitesse
    else:
        t.speed(5)
        statu_vitesseVar.set("Vitesse : {}".format(5))
def init_iteration(default: "Variable d'itérations par défaut"):
    """initialise tous les paramtre lié à l'itération"""
    if str(entrer_iteration.get()) != "":
        nb_iterations = int(round(float(entrer_iteration.get()),0)) # arondie la variable entrée pour en faire un type int
        statu_iterationVar.set("Iterations : {}".format(nb_iterations)) # Afficher le statu de l'iteration
        return nb_iterations
    else:
        statu_iterationVar.set("Iterations : {}".format(default)) # Variable par défaut
        return default
```

Les boutons de choix ont tous un numéro attribué qui est enregistré dans la variable « choix_fractales ». Chaque numéro représente une fractale est permet ensuite de choisir la fractale dans la fonction « dessin ».

```
bouton_flocon = Button(frm_menus_gauche, text= "Flocon de Koch", command= lambda: choix_fractales.set(1), bg='white', image=imgFlocon)
bouton_arbre = Button(frm_menus_gauche, text= "Arbre", command= lambda: choix_fractales.set(2), bg='white', image=imgArbre)
bouton_carre = Button(frm_menus_gauche, text= "Carre recursive", command= lambda: choix_fractales.set(3), bg='white', image=imgCarre)
bouton_triangle = Button(frm_menus_gauche, text= "Sierpinski", command= lambda: choix_fractales.set(4), bg='white', image=imgSerpinski)
```

Des images sont ajoutées aux boutons pour un aperçu du dessin.

```
imgFlocon = PhotoImage(file = "assets/flocon.png", master=root) # Importe une image
imgFlocon = imgFlocon.subsample(95, 95) # Rogner l'image
```

Le bouton « ? » créer une second fenêtre Tkinter dans un autre programme, celle-ci affiche des informations sur les fractales ainsi que l'image d'illustration de projet.

```
def infoScreen():
    temp = Toplevel()
    temp.geometry("800x800")
    temp.title("INFOS")

    label_titre = Label(temp, text="Qu'est ce qu'une fractale ?", font=("Arial", bold, 18))
    label_info = Label(temp,
        text=
        """
        Le mot fractale désigne des figures auto-similaires c'est-à-dire des figures dont le motif
        C'est le mathématicien français Benoît Manderlbrot qui est à l'origine du terme fractale (
        Les fractales existe dans la nature à l'image des feuilles de fougère ou du chou de Romane
        Mais on les trouve également dans les vaisseaux sanguins, les poumons, les bordure côtière
        Parmi les images fractales célèbres, on peut citer le flocon de Koch, le triangle de Serpi
        En image de synthèse, les fracatles sont utilisées pour la génération de décor.
        Ici, le tracé des fractales utilise la récursivité.
        Cliquer sur les différents boutons pour tracer un arbre, un carré, le flocon de Koch et le
        """)
    img = PhotoImage(file="assets/projet.png")

    label_titre.grid(row=0)
    label_info.grid(row=1)
    lbl = Label(temp, image=img).grid(row=2)

    temp.mainloop()
```

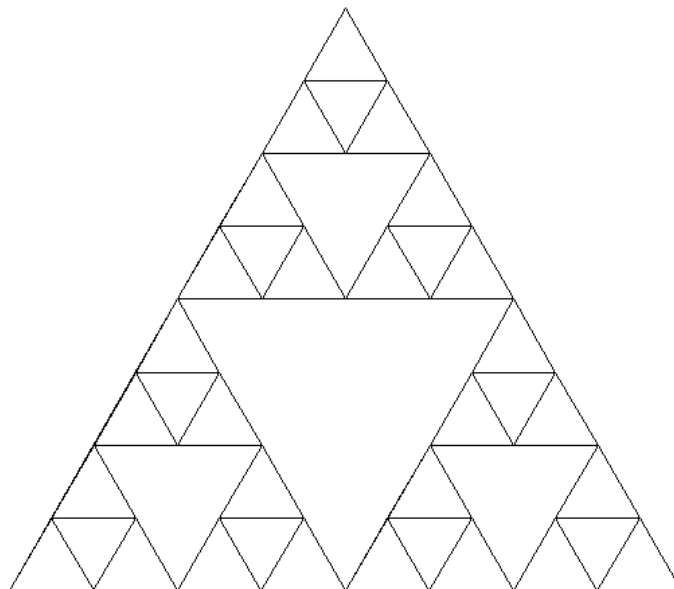


La fenêtre d'affichage un écran pouvant utiliser la méthode Turtle dans un canvas tkinter afin de pouvoir faire interagir tkinter et Turtle. Le canvas est initialisé en début de programme et l'écran de Turtle juste ensuite avec « TurtleScreen ». L'utilisation de RawTurtle permet ensuite l'utilisation de la tortue dans l'écran de Turtle.

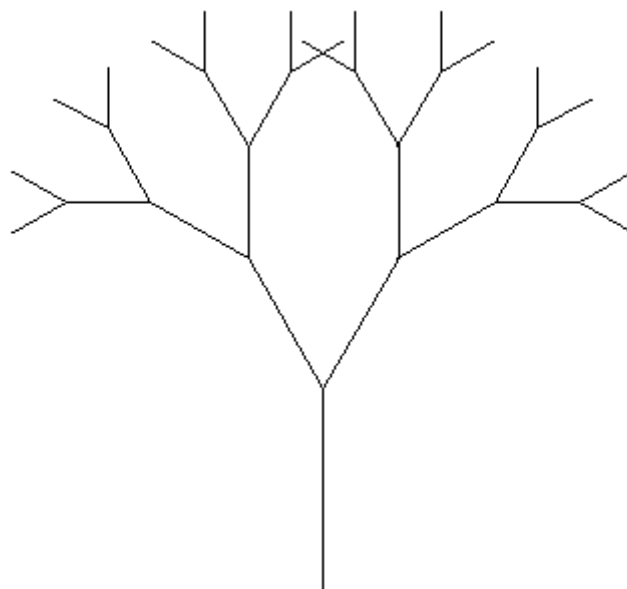
```
canvas = Canvas(root, width=725, height=595) # Canvas de dessin
canvas.grid(row = 0, columnspan=2, padx=37)

screen = TurtleScreen(canvas) # Met en place un écran turtle dans le canvas de tkinter
t = RawTurtle(canvas) # Turtle dans le canvas
```

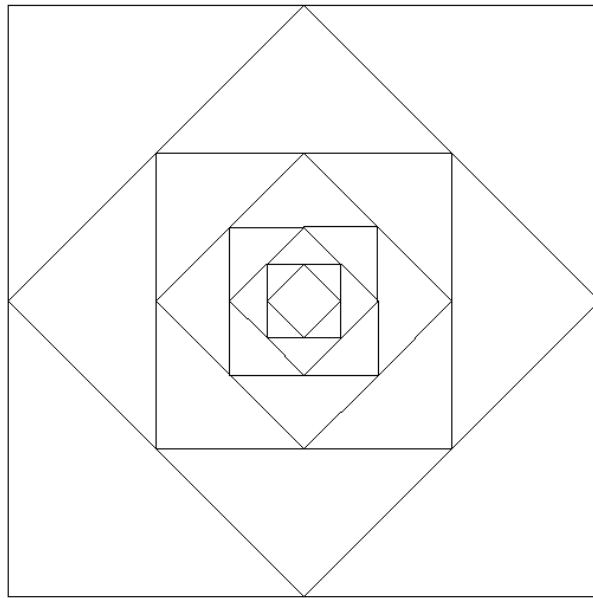
Voici quelques captures d'écran de réalisations :



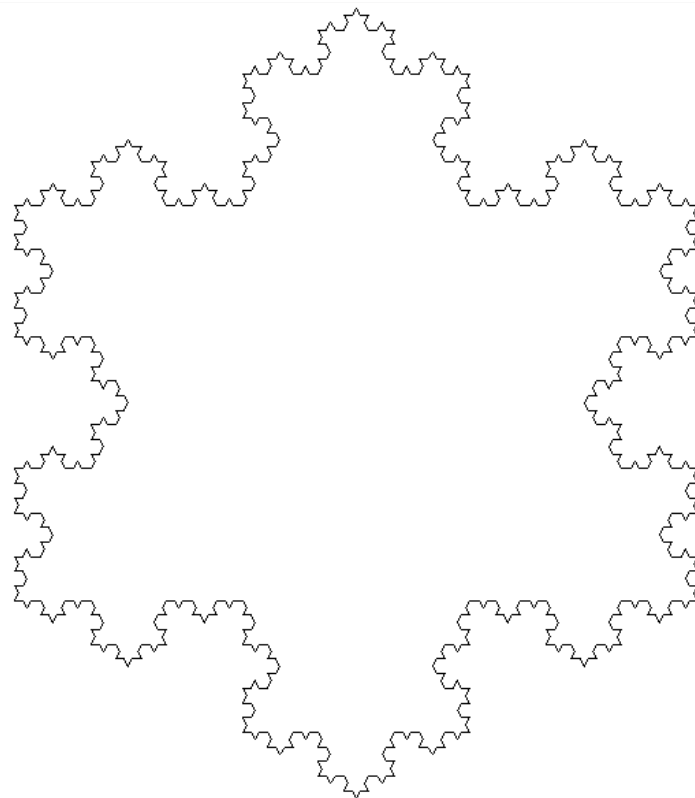
Triangle de Sierpinsky avec 3 itérations



Arbre avec 5 itérations

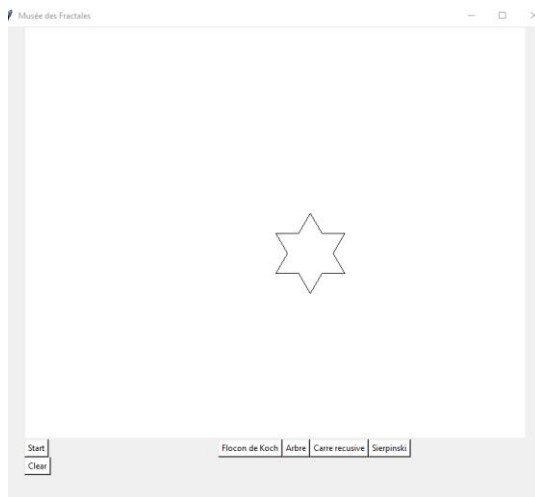


Carrés avec 7 itérations

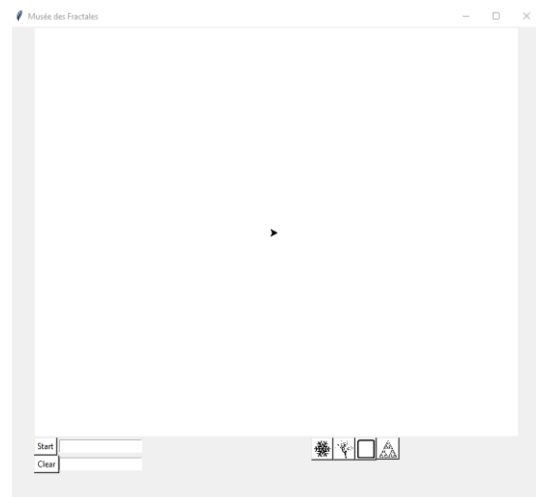


Flocon de Von Koch avec 4 itérations

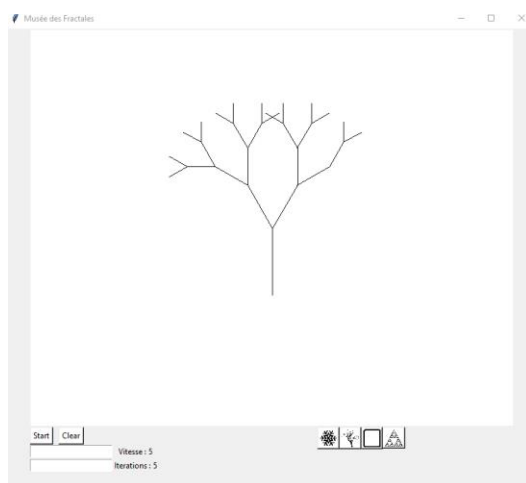
Captures d'écran du fonctionnement et de l'opérationnalité :



Problème d'affichage des boutons



Ajout des fenêtres de saisie et des boutons



Repositionnement des boutons et ajout de textes



Problème de reset d'écran