

|  |  |
| --- | --- |
|  | **Édition 2023**  **DOSSIER DE CANDIDATURE**  **PRÉSENTATION DU PROJET** |

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**](https://trophees-nsi.fr/preparer-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**](mailto:info@trophees-nsi.fr).

**NOM DU PROJET : Projekt-Fractal**

**> PRÉSENTATION GÉNÉRALE :**

**Projekt-Fractal** est un générateur de fractales qui **n’utilise pas la récursivité**.

Nous utilisons principalement la bibliothèque **Turtle** pour le dessin des fractales, et la bibliothèque **Tkinter** pour l’interface graphique. Nous proposons des fractales en formes **d’Arbres** ou de **Polygones**, chacune avec des paramètres modifiables pour assurer une liberté la plus totale dans la génération.

Une **fractale** est une figure appartenant au domaine des mathématiques théoriques, car elle touche au domaine de **l’infini**. C’est une figure possédant un motif se répétant à l’infini, c’est-à-dire que si on zoomait sur un endroit de la fractale, on pourrait presque voir la figure complète, et ça à l’infini. On dit que c’est une figure auto similaire.

Le concept de fractales existe depuis très longtemps, mais le mot « fractal » est apparu en 1975, inventé par Benoit Mandelbrot.

Quelques objets dans la nature possèdent une forme similaire à une fractale, tel que le chou romanesco :

Pour que notre programme de prenne pas un temps « infini » à se compléter, nous proposons un paramètre qui est le nombre de générations de la fractale. Plus ce paramètre est haut, plus la fractale prendra de temps à se compléter.

L’idée de départ est venue grâce à notre professeur de NSI (expliqué dans les étapes). Mais notre professeur nous a rapidement prévenu que ce projet serait compliqué à faire sans récursivité, mais on s’est dit que plus il y avait de défi, plus on s’amuserait, donc on a décidé de commencer.

Finalement, ne pas avoir utilisé la récursivité a peut-être été une grosse difficulté en plus, mais ça nous a permis de mieux comprendre comment le programme fonctionnait, et nous a permis d’inventer des nouvelles possibilités de fractales, et des paramètres possibles à modifier, ce qui nous a permis **d’enrichir beaucoup** notre programme.

**> ORGANISATION DU TRAVAIL :**

Nous avons eu beaucoup de mal à nous organiser au début, car on voulait au départ se mettre en groupe de 5, mais au fur et à mesure du temps, deux de nos membres ont dû se retirer pour diverses raisons, nous avons donc dû nous replier sur rester en groupe de 3 (Merci Luca et Justine pour le support moral).

**Répartition des rôles :**

* Thibaud Berthou : s’occupe de toute l’interface grâce à la bibliothèque Tkinter (travail immense, force à toi Thibaud)
* Maxence Bayart : s’occupe des fractales Arbre, Arbre avec angles, Polygone, Flocon de Koch spécial et Polygone avec motif. S’occupe aussi du document de présentation (salut, c’est moi qui écris ça, désolé pour les fautes…), et de rassembler les différentes documentations techniques. A aidé aussi à l’écriture du script de la vidéo.
* Gabriel Houdayer-Kerrand : s’occupe de la vidéo et de la fractale Arbre réglable

On se retrouvait presque tous les jours à l’école, on discutait très souvent du projet, et on en discutait presque à chaque cours de NSI. On travaillait souvent le soir chez nous (entre les parties d’Overwatch et les sessions de Sea Of Thieves), et on discutait et s’échangeait nos avancées et problèmes rencontrés le soir par Discord. Nos fichiers étaient échangés par Discord aussi. On a finalement utilisé GitHub pour le stockage des fichiers à la fin.

**LES ÉTAPES DU PROJET :**

L’idée :

Nous avions déjà créé notre groupe après que notre professeur de NSI nous avait déjà parlé du trophée, mais il nous manquait une idée, ce qui a été résolu lorsque notre professeur de NSI nous a présenté un livre contenant des idées de mini-projets à faire, et on a aperçu un mini-projet sur les fractales utilisant la récursivité (quelque-chose de très simple, environ 20 lignes de programmes). Moi (Maxence) qui a toujours trouvé le sujet des fractales intriguant, me suis dit qu’on pourrait faire un projet de générateur de fractale personnalisée, puis on a, au fur et à mesure, apporté des idées pour étoffer le concept, et le projet est né.

Le codage :

Très peu de temps après l’idée, je (Maxence) me suis directement mis à coder, parce que je trouvais ça très intéressant, et en moins d’une semaine, (après beaucoup de galère), j’ai réussi à développer la première fractale, l’Arbre. Je me suis tout de suite amusé à modifier des paramètres et à effectuer des recherches pour trouver ce que je pouvais faire avec ça.

En voyant ce que j’avais fait, Thibaud s’est rapidement motivé à créer une interface, et après avoir effectué plusieurs recherches de son côté et après avoir bien galéré aussi a réussi à nous créer un début d’interface qui fonctionne.

Gabriel a ensuite commencé à coder et nous a donné la fractale de l’Arbre réglable.

J’ai ensuite développé les fractales de Polygones.

On a aussi pris l’habitude de documenter chaque fonction après le développement pour faciliter la compréhension des fonctions par les autres membres (et aussi pour éviter d’oublier ce que fait un paramètre).

La vidéo, la documentation et la présentation papier :

Thibaud a ensuite continué d’améliorer son interface et à la documenter, pendant que Gabriel et moi (Maxence), nous avons travaillé sur la présentation et la vidéo.

**> FONCTIONNEMENT ET OPÉRATIONNALITÉ :**

**Avancement du projet :**

Nous avons déjà majoritairement terminé notre projet, mais quelques idées que nous avions n’ont jamais réussi à aboutir.

Ce qui est terminé :

* Une interface avec :
  + Des paramètres modifiables
  + Une sélection de différentes fractales
  + Un bouton valider et effacer
  + Un système d’erreurs fonctionnel
  + Un bouton d’aide
  + Une fenêtre de dessin des fractales
  + Des presets
* Des fractales, plus majoritairement :
  + Arbre
  + Polygone
  + Arbre réglable

Le reste des fractales sont des modifications de ces 3 fractales majeures.

Ce qui est commencé mais qui n’as jamais abouti (par manque de temps ou de motivation), ou ce qui était prévu au départ mais qui n’as jamais été fait :

* Calcul de temps (les fonctions sont créées, mais pas implémentés dans l’interface, elles peuvent être toujours utilisés avec une console python)
* Fractale du carré de Sierpinsky ou du triangle de Sierpinsky
* Ensemble de Mandelbrot ou de Julia

**Débogage :**

Nous n’avons pas eu beaucoup besoin de faire beaucoup de débogage, car nous travaillions très souvent avec les fractales et l’interface (par exemple lors de l’élaboration des presets), donc on trouvait assez facilement des bugs, parce-qu’on cherchait toujours à trouver les limites du programme.

Mais ce qui était plus important, c’était l’optimisation, pour faire en sorte que notre programme prenne moins de lignes et qu’il soit plus clair (il fait toujours environ 2000 lignes donc…).

**Difficultés rencontrées et solutions apportés :**

Nous avons rencontré plusieurs difficultés, majeures étant :

* La difficulté de la récursivité. En effet, les fractales sont des objets récursifs de base, donc nous avons dû trouver des manières pour la remplacer, car ne pas utiliser la récursivité était un de nos buts. Pour les fonctions de type Arbres, nous avons utilisé 2 listes, une contenant les coordonnées et une contenant l’orientation, et on balayait la liste pour récupérer les informations et continuer les branches. Pour les fonctions de type Polygones, on a utilisé une liste y contenant des instances de « k » entre les angles que devra suivre la fractale, puis on balayait la liste en remplaçant tous les « k » dans la liste par la liste y, en rassemblant tous les nombres des listes dans une même liste grâce à la fonction déplier. On répète cette étape pour le nombre de générations demandées, puis on supprime tous les « k » en trop. (« k » est une lettre choisie aléatoirement, uniquement parce qu’elle n’était pas utilisée autre-part dans le programme).
* Les problèmes avec le changement d’interface pour chaque fractales. Lors du changement de fractale, le programme doit supprimer tous les widgets présents sur l’interface, et on avait des problèmes avec ça. Mais ce problème a été résolue grâce à la fonction suppression\_widget, qui balaye tous les widgets présents sur l’interface (au lieu d’interagir avec chacun des widgets nommés), et les supprime.
* Nous avons eu un dernier problème, qui est la taille de l’écran. En effet, lorsque Thibaud travaillait sur l’interface, il avait un écran plus grand que la moyenne pour les ordinateurs portables. Donc quand il m’a envoyé (à Maxence) son programme de l’interface, on s’est rendu compte que la fenêtre était trop grande pour mon écran, et que plusieurs éléments n’apparaissaient pas. Thibaud a donc modifié son programme pour qu’il calcule la taille de l’écran en pixels, et que les éléments de son interface soient adaptés par rapport à cette taille.

**> OUVERTURE :**

**Idées d’améliorations :**

Plusieurs pistes d’améliorations sont possibles, du coté des fractales et de l’interface. Pour améliorer notre programme, on pourrait rajouter plusieurs fractales, telles que le carré ou le triangle de Sierpinsky, l’ensemble de Mandelbrot, ou par exemple la courbe du C (qui est déjà possible dans une console, mais pas avec l’interface, il faut utiliser des angles de pi/3 et 2pi/3 en radians dans la fonction arbre avec angles, et utiliser un rapport de 1/sqrt(2), exemple de courbe du C grâce à la console en dessous).

Une image contenant diagramme

Description générée automatiquement

On pourrait améliorer notre programme en implémentant les fonctions qui permettent de calculer le temps, ou d’utiliser les radians dans les angles.

On pensait aussi pouvoir peut-être implémenter une barre de progression de la fractale pour celles avec beaucoup de générations.

On pourrait implémenter des images dans la fenêtre d’aide permettant de mieux comprendre certains des paramètres plus compliqués.

On pourrait aussi implémenter un bouton permettant de sauvegarder l’image de la fractale sur son ordinateur.

On pourrait aussi donner une option de décider ou commencera le dessin, ou de régler la taille du canvas de dessin, pour permettre d’afficher la totalité de la fractale pour les plus grandes.

On pourrait aussi tenter d’implémenter les couleurs pour la totalité des fractales.

On pourrait enfin aussi ajouter la possibilité aux utilisateurs de sauvegarder leur propre presets, et utiliser un fichier JSON pour les stocker localement.

**Stratégie de diffusion possible :**

Plusieurs stratégies de diffusions sont possibles. Thibaud et moi (Maxence), on est administrateur d’une petite communauté Discord, on pourrait leur proposer d’essayer notre programme. On a aussi un ami qui possède une chaine YouTube (bien que pas très grande), on pourrait lui proposer de faire une petite vidéo dessus pour le présenter. On peut aussi demander à notre professeur de NSI si on pourrait le présenter à notre classe, et potentiellement aux autres classes et niveaux de l’établissement.

On pourrait aussi créer par exemple un site internet faisant la promotion de notre projet.

**Analyse critique du résultat :**

Je pense que si c’était à refaire, on aurait gardé la règle de ne pas utiliser la récursivité, car ça a permis de beaucoup enrichir notre programme, mais on aurait commencé le projet beaucoup plus tôt. On aurait aussi essayé de mieux s’organiser dès le début.

On aurait aussi dès le début dessiné un plan exact de ce que l’on voulait faire au final (parce que là on est un peu allé au feeling).

**DOCUMENTATION**

**Spécifications fonctionnelles et techniques**

Avoir installé **Python 3.8.10**, on n’a pas testé avec d’autres versions, donc on ne sait pas si ça fonctionne avec autre-chose que cette version.

Nous utilisons les modules :

* Turtle
* Tkinter
* Random
* Math
* Time

Tous ces modules sauf le module turtle devraient déjà être compris avec Python 3.8.10. Pour installer le module turtle, dans un terminal, écrire la commande :



Vous pouvez ensuite récupérer le fichier sur GitHub, puis naviguer dans le terminal pour arriver au fichier téléchargé.

Vous pouvez ensuite exécuter le programme grâce à la commande :

py projekt\_fractal.py

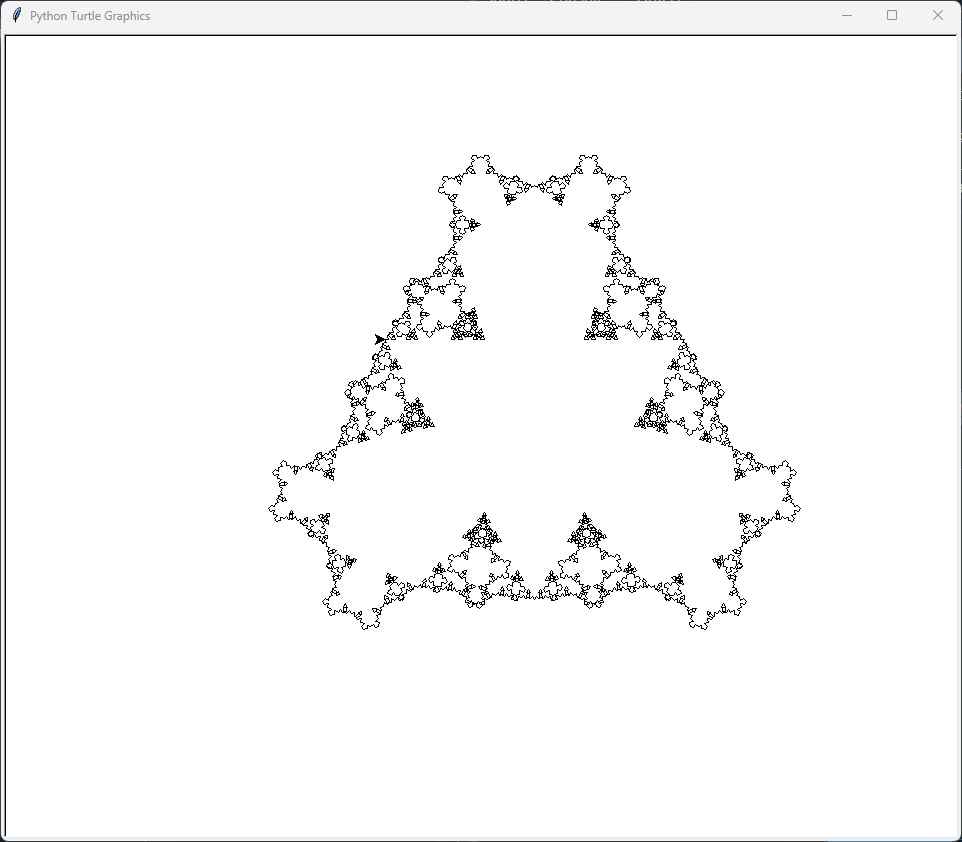
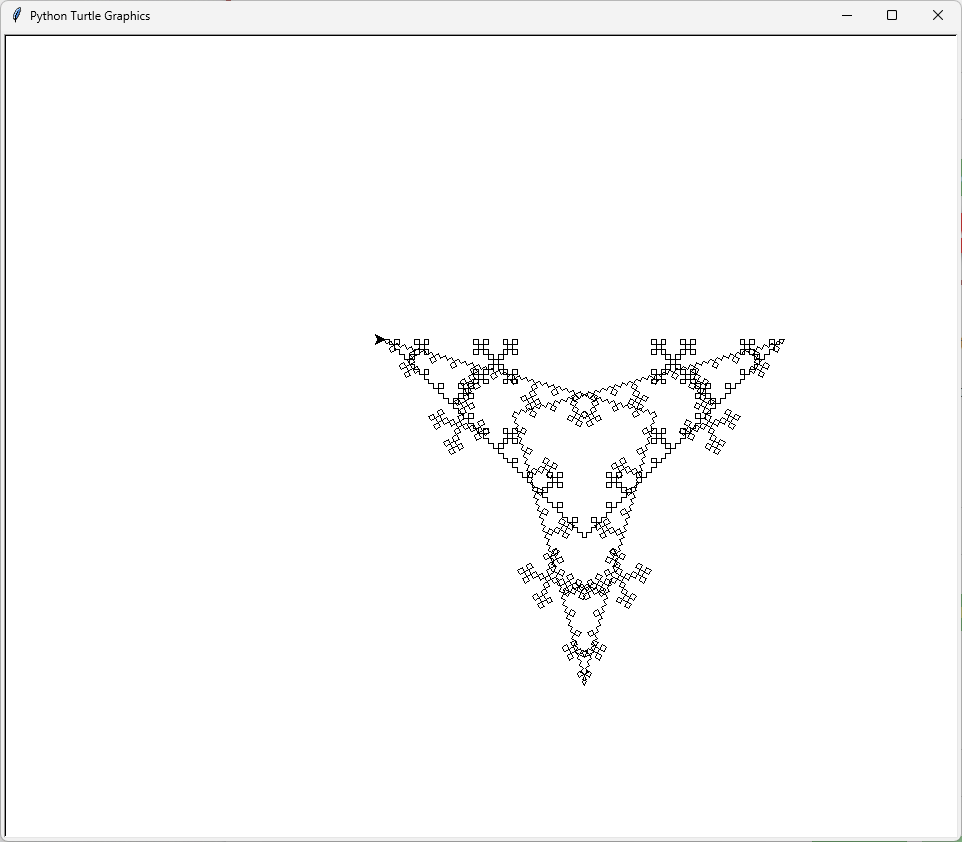
Notre programme est composé intégralement en langage python.

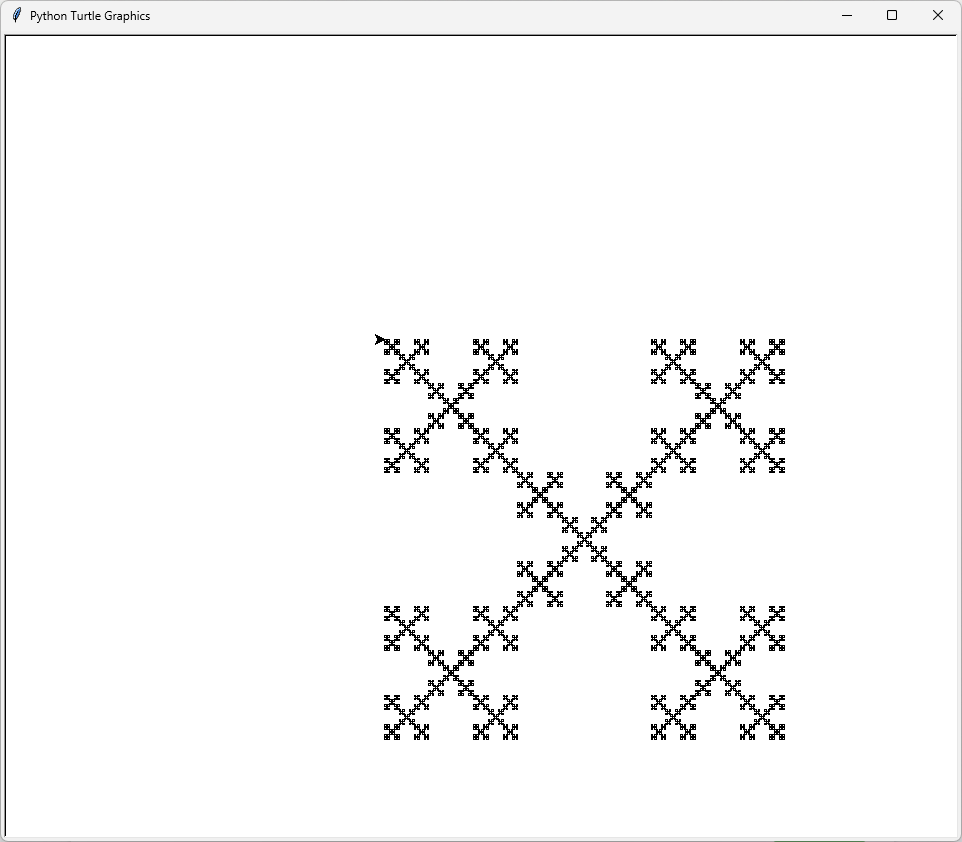
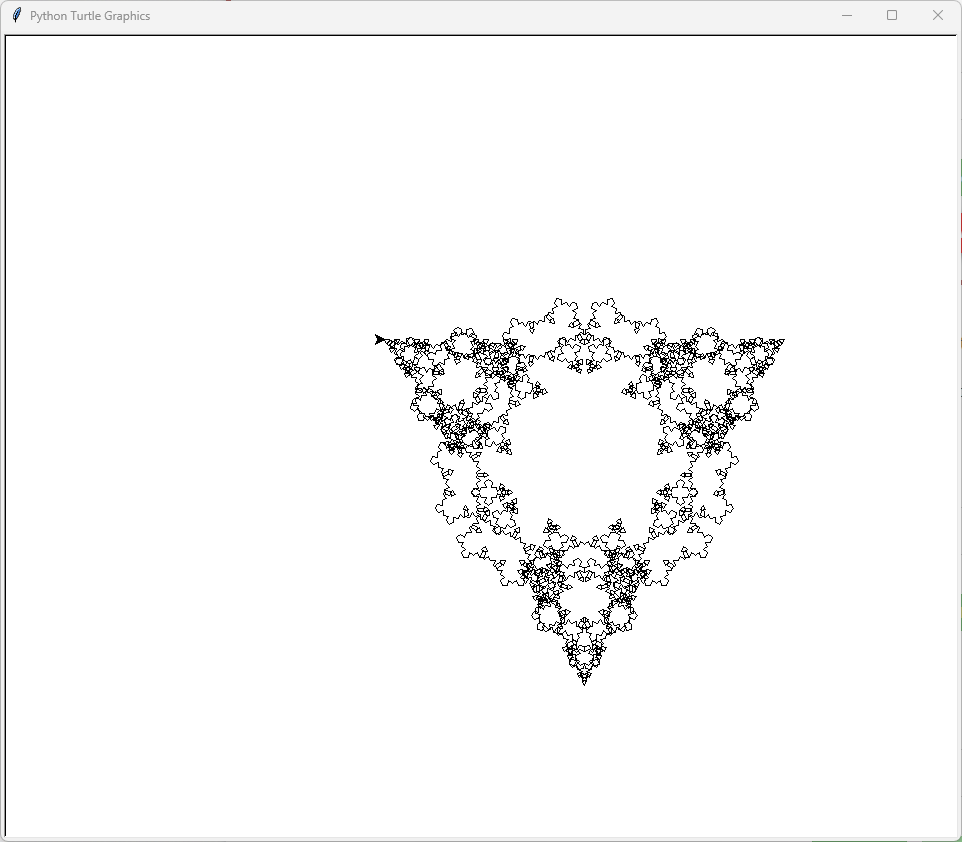
Nous avons utilisé le site <https://mathcurve.com/fractals/fractals.shtml> pour certaines des idées de fractales et de paramètres.

**Images :**

Des fractales obtenues avant la création d’une interface :

Une image contenant diagramme

Description générée automatiquementUne image contenant diagramme

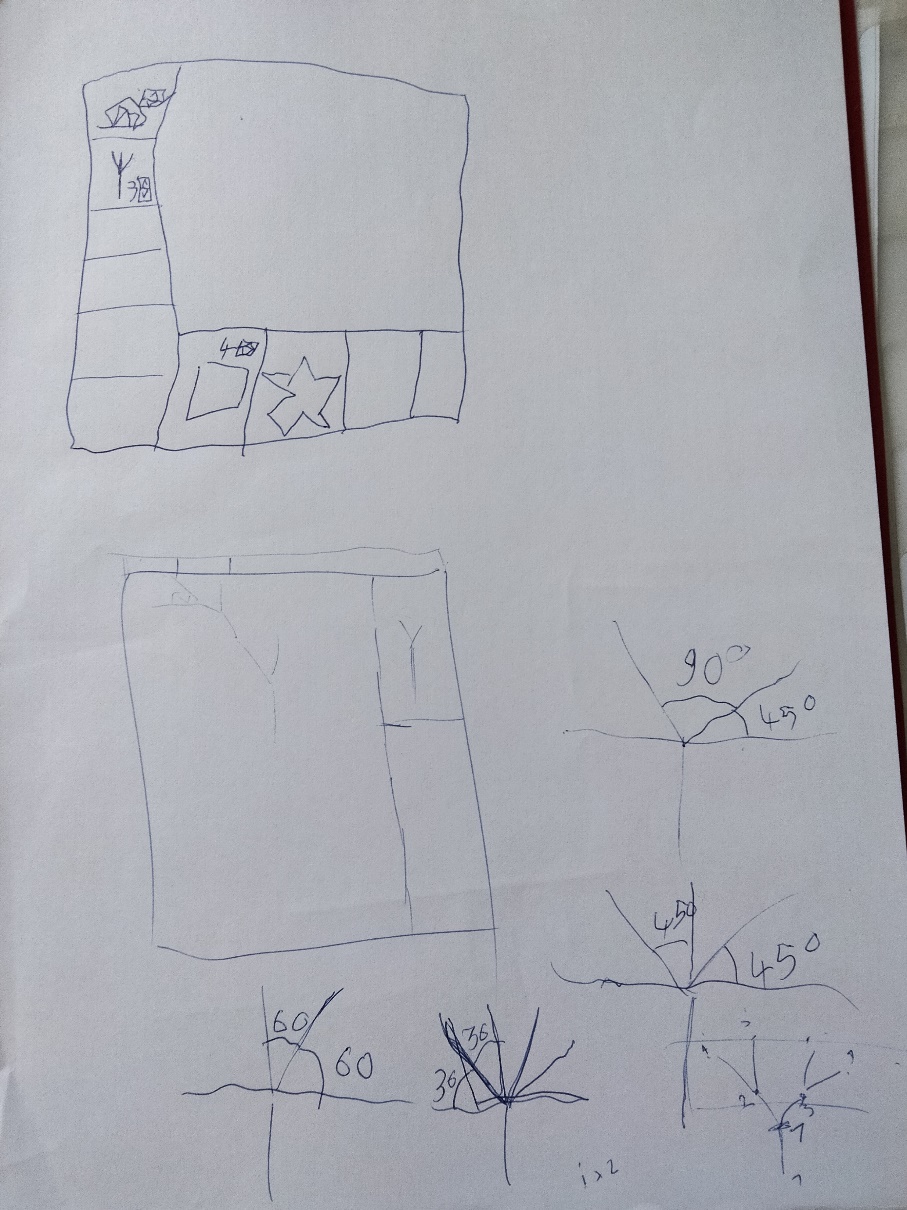
Description générée automatiquement

Petit secret :

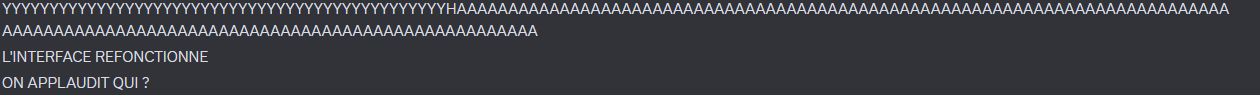
Pour faire la fractale PROJEKT de l’affiche : utiliser la fonction arbre réglable

* Nombre de générations : 2
* Epaisseur : 0
* Taille de la plus petite branche : 2
* Orientation des branches : 45
* Règle du curseur : [++fR]--fR++fR++fR[++fR]--fR[++fR]--fR++fR++fR[++fR---fR]--fR++fR--fR--fR[fR--fR--fR]++fR++fR--fR--fR[fR]++fR[[++fR--fR]--fR++fR]fRfR[[[--fR]-fR]+fR]++fR--fRfR[--fRfR[--fR]++fR]fR--fR++fR

Schémas hasardeux lors de l’idée :



Thibaud qui est content quand son programme fonctionne



Essai d’explication envoyé sur discord : fait sur paint

Une image contenant polygone

Description générée automatiquement

Exemple de vote pour les presets

Une image contenant diagramme

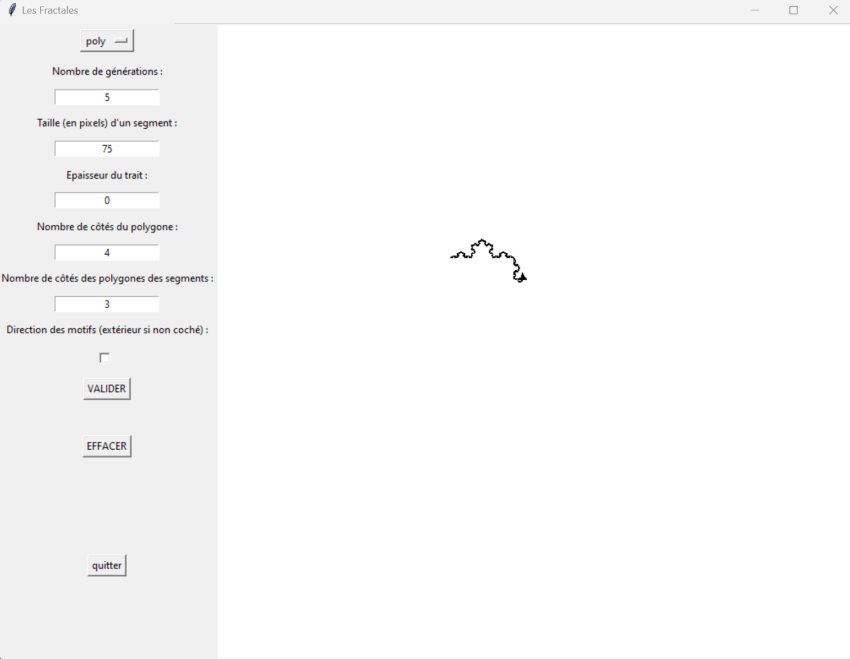
Description générée automatiquement

Des règles que l’on s’était imposé pour la création de fractales :

Une image contenant texte

Description générée automatiquement

Version très antérieure de l’interface



Interface actuelle :

