

Liste des fonctionnalités imposés :

- Lecture d'un fichier externe (type « exemples/ ») formaté.
- Interprétation de ce fichier en un L-système, type **system**.
- Interprétation des chaînes générées à partir du système.
 - Interprétation à la volée des chaînes parenthésées afin d'éviter d'utiliser excessivement de la mémoire(possibilité pour commencer d'interpréter ces chaînes en suites de commande.
- Faire une partie graphique simple, mais fonctionnel. Ne pas se perdre dans du superficiel, tel que des boutons, etc..
 - L'affichage du système pourrait se faire dans un premier temps uniquement par le résultat final d'une itération.
 - Par la suite, il faudrait que les itérations du système soit affichés comme une suite d'image, pour que l'utilisateur puisse voir visuellement la construction d'un fractal.(soit itération par itération).
- Il faut gérer l'échelle de l'affichage pour un système.
 - Dans un premier temps, l'échelle pourrait être entrée manuellement.
 - Par la suite, il faudrait faire un calcul raisonnable et automatique des bornes de l'échelle pour la partie graphique du système.

Extensions possibles :

Quelques extensions proposés par le sujet que je trouve(en tant que matthieu) intéressante. C'est donc exhaustif. La sauvegarde des L-Systèmes. A partir d'une image de L-Système, créer ce système sous la forme des fichiers textes de « exemples/ ».

Ajout des couleurs au fractal, afin d'apporter un dégradé de couleur.

Points importants à ne pas négliger dans l'implémentation du projet :

- Indentation systématique du code
- 80 caractères max. par ligne
- Donner un nom clair et parlant pour chaque fonction et chaque variable
- Commenter le code de façon claire et concise, régulièrement, donc un commentaire par fonction la décrivant en 2-3 lignes, entrée-sortie. Commenter dans les fonctions pour les lignes complexes.
- Ne pas se répéter dans le code.