

Presentation L systeme

Interpréteur de systèmes de Lindenmeyer

Autheur :
Bedjou Celina
Djeha Wassim
Belkacemi Yani
Kaced Louheb

Université de Caen Normandie

Table des matieres

- ① Introduction
- ② Objectifs
- ③ Éléments technique
 - Moteur de réécriture
 - Générateur du rendu 2d
 - Générateur du rendu 3d
 - Fonctionnalités
- ④ Utilisation des fichiers pour les dessins 2D
- ⑤ Conclusion

Introduction

Objectifs

- Générer des structures complexes à partir de règles simples .
- Modéliser la croissance de plantes .
- Explorer des fractales et des structures auto-similaires.
- Simuler des phénomènes naturels .
- Développer des algorithmes d'optimisation .

Pourquoi ce projet?

- Curiosité scientifique .
- Créativité artistique .

Un peu D'Histoire

- Aristid Lindenmayer .
- Dans les années 1960.

Grammaire formelle

- Un alphabet V ; $V=\{A,B\}$.
 - Ensemble de constants S ; $S=\{-,+,[,],..\}$.
 - Un axiome de départ w ; $w=\{A\}$.
 - Ensemble de règles P ; $P=\{A:AB,B:A\}$.
- Notation : $G=\{V,S,w,P\}$

Exemple avec des Constantes

$$\begin{array}{c} A = A+B \\ B = B-A \\ \text{Axiom} = A \\ \updownarrow \\ A + B \\ \swarrow \searrow \\ A + B + B - A \end{array}$$

Exemple détaillée

Alphabet: A, B

Axiome: A

Règles: $A \rightarrow AB$

Règles: $B \rightarrow A$

Constantes: *Vide*

Itérations: 6

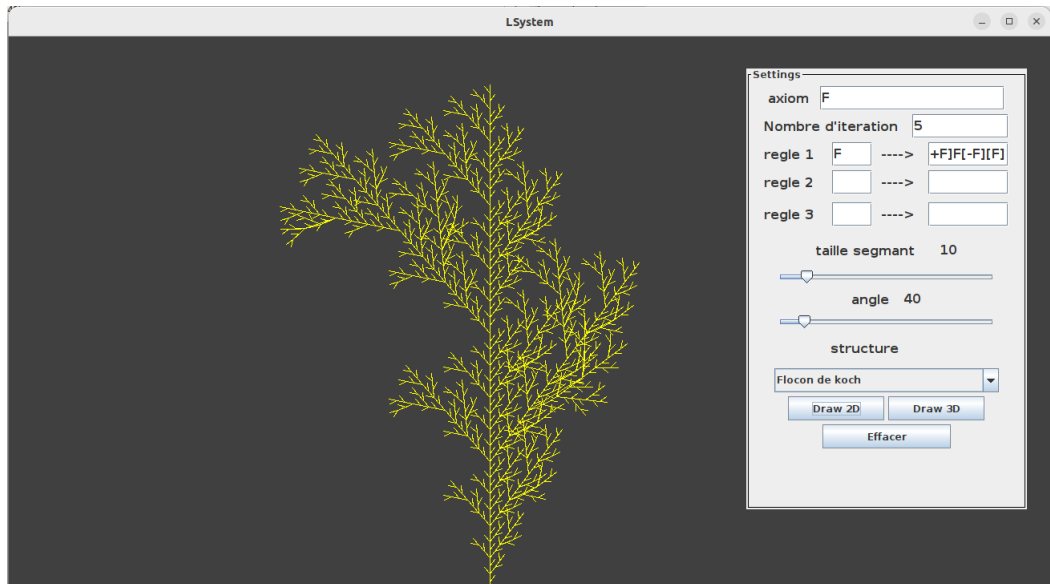
Exemple détaillée

- * $n = 0$, A
- * $n = 1$, AB
- * $n = 2$, AB A
- * $n = 3$, AB A AB
- * $n = 4$, AB A AB AB A
- * $n = 5$, AB A AB AB A AB A AB
- * $n = 6$, AB A AB AB A AB A AB AB A AB AB A

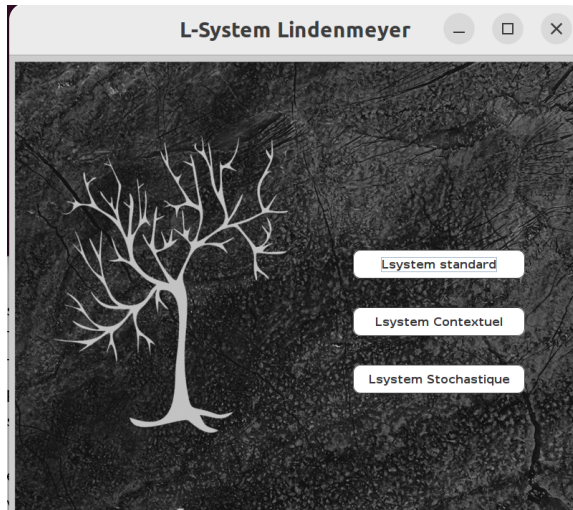
Génération des L-systèmes standard

- Une Classe Lsystem .
- Une fonction simulation().
- Une fonction rotation 2D.
- Une fonction ClaculCoordinate().
- Dessiner les points avec drawLine().

.

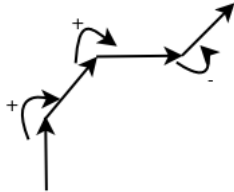


Interface réaliséé



Rendu 2d

- Le moteur de rendu graphique 2d se base sur une rotation sur les axes X et Y selon l'opérateur de rotation.
- Nous pouvons voir à travers ce schéma la génération du rendu 2d.



Rendu 2D

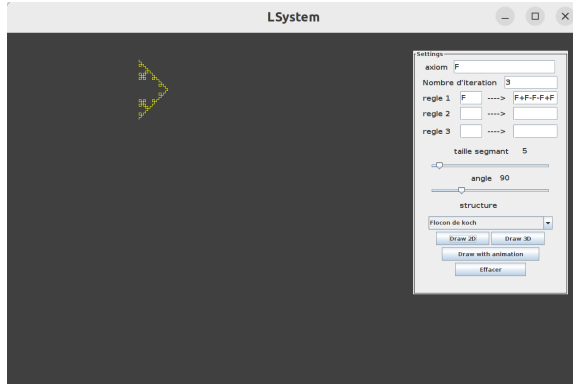


Figure: Rendu 2D

Rendu 2D

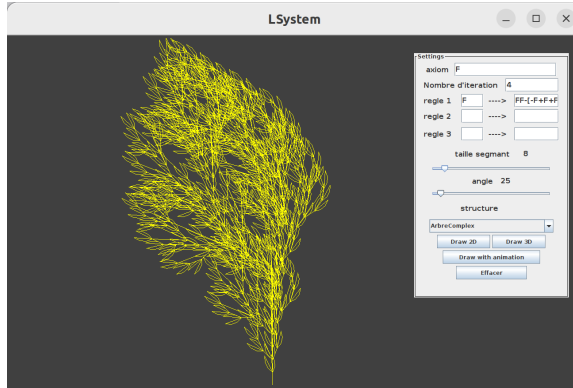
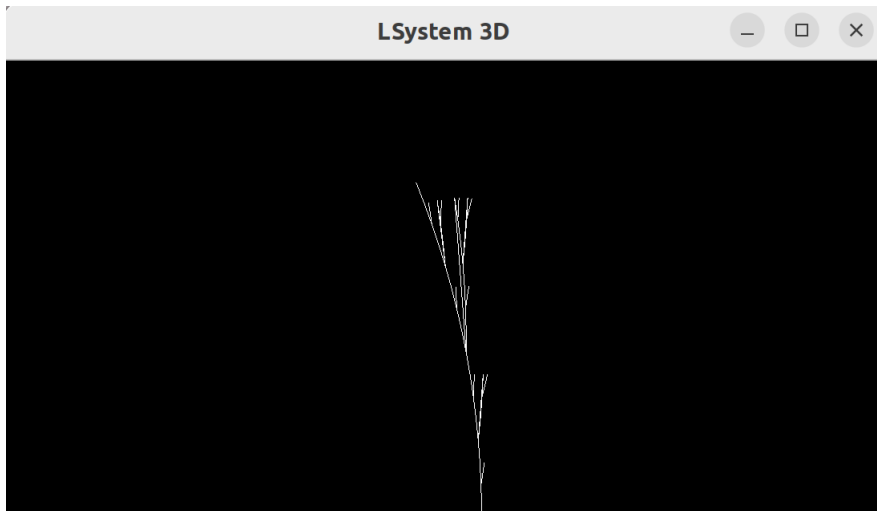


Figure: Rendu 2D

Rendu 3d

- Le moteur de rendu graphique 3D se base sur une représentation en trois dimensions des structures générées par les L-systèmes, avec une rotation sur les axes X, Y et Z selon les opérateurs de rotation.

Rendu 3d



Rendu 3D

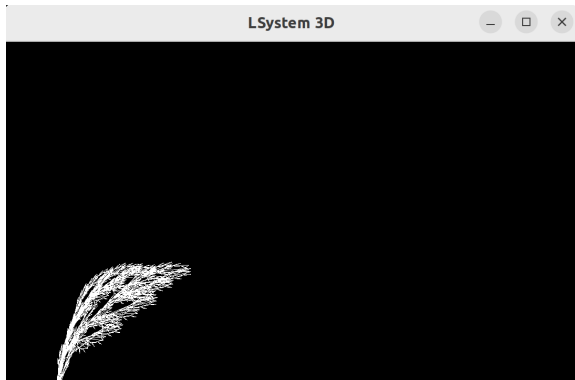
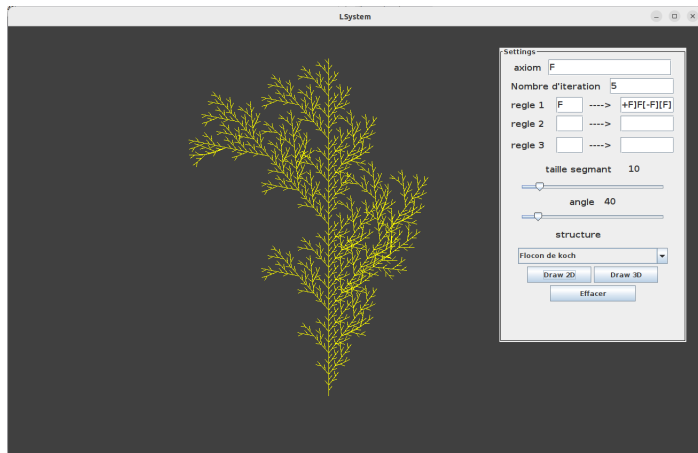


Figure: Rendu 3D

Génération des L-systèmes contextuels

- Les L-systèmes contextuels sont des L-systèmes où les règles de développement des symboles dépendent du contexte. Cela permet de générer des formes plus complexes et réalistes, mais nécessite des règles de développement plus complexes.

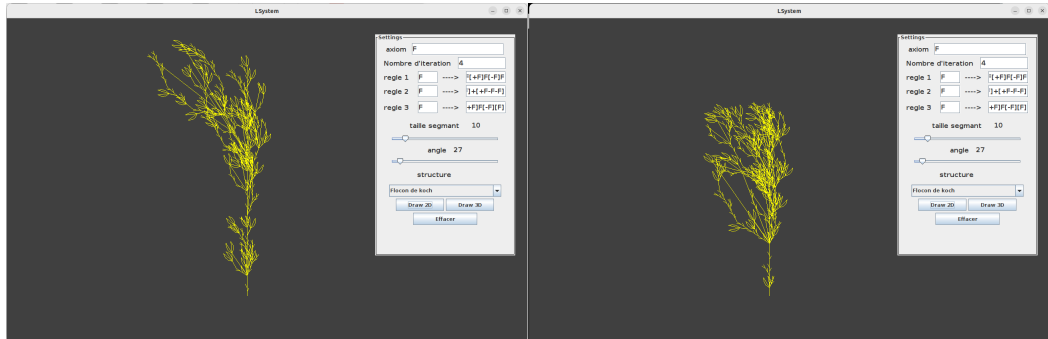
Génération des L-systèmes contextuels



Génération des L-systèmes stochastiques

- Les L-systèmes stochastiques utilisent des probabilités pour déterminer les règles de développement des symboles, permettant ainsi de produire une variété de formes à partir d'un même L-système de base.

Génération des L-systèmes stochastiques



Interface utilisateur + Gestion des erreurs

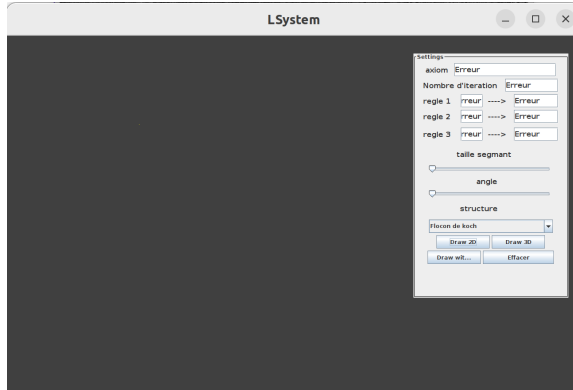


Figure: Gérer les exceptions



Animation de L-systèmes

- Visualiser comment les L-systèmes évoluent à mesure qu'ils se développent .
- Voir comment les motifs se répètent à différentes échelles

Timer

```
private int currentLineIndex = 0, nbIteration = 0;
//Créez un objet Timer avec une période de rafraîchissement de 1000 ms
private Timer timer = new Timer(1000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        //Incrémentez l'index de ligne actuel
        currentLineIndex++;
        //Redessinez la zone de dessin
        repaint();
        //Vérifiez si toutes les lignes ont été dessinées
        if (currentLineIndex > nbIteration) {
            //Arrêtez le timer
            setDoitDessiner(doitDessiner:false);
            timer.stop();
        }
        if(currentLineIndex ==0) {
            timer.start();
        }
    }
});
```

Figure: Timer

Timer

```
if (doitDessiner) {
    Alphabet a = new Alphabet(this.listAlphabet);
    LSystem l = new LSystem(a, this.rule, this.axiome, this.nbIteration, this.tailleSegment, this.angle);
    Interpretation inter = new Interpretation();
    //Calculez les coordonnées pour la première itération
    resultas = inter.calculateCoordinatesWithIndex(l, currentLineIndex);
    //Démarrez le timer
    timer.start();
}

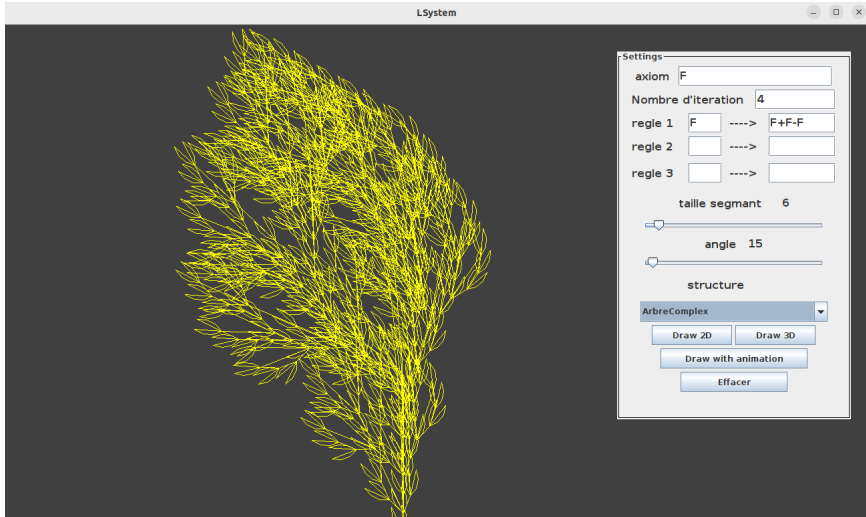
if (resultas != null) {
    //Dessinez toutes les lignes jusqu'à l'index de ligne actuel
    for (int i = 0; i < resultas.size()-1; i++) {
        g.setColor(this.couleur);
        g.drawLine((int) resultas.get(i).getX(), (int) resultas.get(i).getY(), (int) resultas.get(i + 1).getX(),
            (int) resultas.get(i + 1).getY());
    }
}
```

Figure: Timer

Utilisation des fichiers pour les dessins 2D

- Utilisation d'une fonction pour sauvegarder les points.
- Stockage des points dans des fichiers.
- Récupération des points pour le dessin.
- Exemple de fonctionnement.

Utilisation de fichiers pour les dessins 2D



Conclusion

Merci de votre attention.

Questions?