

Présentation du projet LSystem

Augustin L.
Luka K.
Maël F.
Paul M.

Université de Caen Normandie
L2 Info
Groupe 4B

May 4, 2023

- 1 Présentation du projet
 - Description générale du projet
 - Objectif du projet
- 2 Présentation des différentes interfaces
 - Le menu principale
 - En console
 - Interface 2D
 - Interface 3D
- 3 Les algorithmes importants
 - Le modèle
 - Les dessins
 - Les threads
- 4 Les tests
- 5 Conclusion

Les objectifs du projet

- **Développement d'un LSystem**
- Comprendre ce que sont les Systemes de Lindenmayer
- Projet codé à l'aide de contrats pour contrer les erreurs
- Développer un interpréteur de LSystem
- Développer une première interface en console
- Développer une interface 2D
- Développer une interface 3D

Exemple de rendu finale en 2D/3D

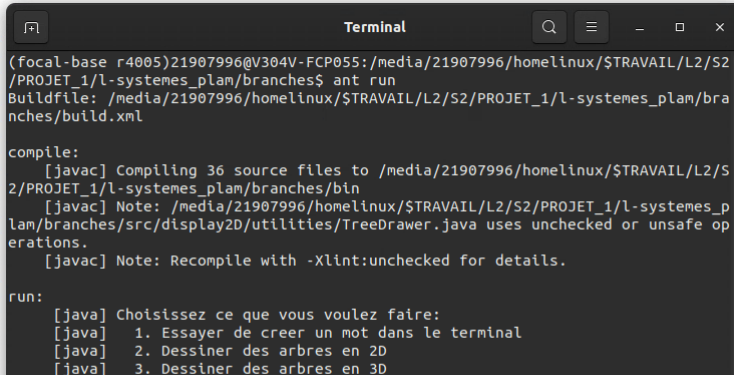


Figure: Exemple d'arbre en 2D et 3D



Figure: source:<https://en.wikipedia.org/wiki/L-system>

Le menu principale



```
(focal-base r4005)21907996@V304V-FCP055:/media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches$ ant run
Buildfile: /media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches/build.xml

compile:
[javac] Compiling 36 source files to /media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches/bin
[javac] Note: /media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches/src/display2D/utilities/TreeDrawer.java uses unchecked or unsafe operations.
[javac] Note: Recompile with -Xlint:unchecked for details.

run:
[java] Choisissez ce que vous voulez faire:
[java] 1. Essayer de creer un mot dans le terminal
[java] 2. Dessiner des arbres en 2D
[java] 3. Dessiner des arbres en 3D
```

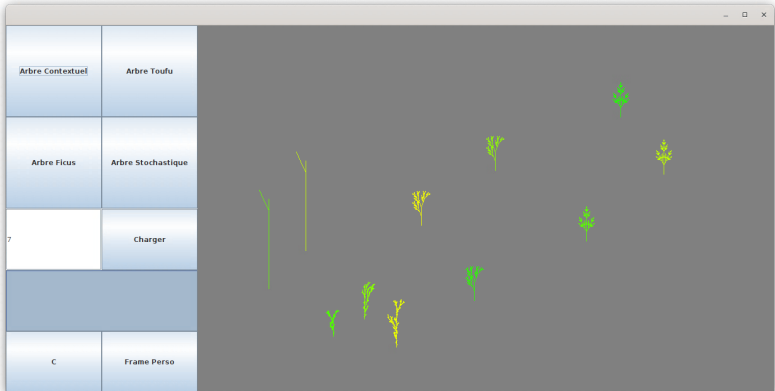
Interpréteur en console

```
Terminal
1
  [java] Commencez par entrer les règles que va suivre votre mot: (Appuyez sur
entrée sans écrire pour passer l'étape)
  [java] Ecrire une règle de la forme: ?->??? ou ?<?>?->???
a->ab
  [java] Ecrire une règle de la forme: ?->??? ou ?<?>?->???
b->bac
  [java] Ecrire une règle de la forme: ?->??? ou ?<?>?->???
c->d
  [java] Ecrire une règle de la forme: ?->??? ou ?<?>?->???

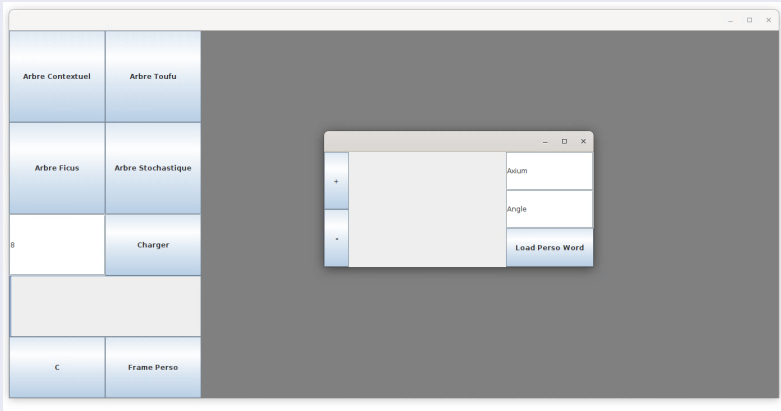
  [java] Maintenant entrez les lettres ignorées par l'amélioration du mot sous
les règles contextuelles
  [java] Ecrire une lettre ignorée
e
  [java] Ecrire une lettre ignorée

  [java] Entrez l'axiom de votre mot:
a
  [java] Entrez le nombre de modification que doit subir votre mot (<20 et >0)
:
6
  [java] Voici votre mot:
  [java] abbaabacabdbacabdabbacdbacabdabbacdbacabacabddbacabdabbacdbacabacab
ddabbacbacabdbacabdabbacdd
```

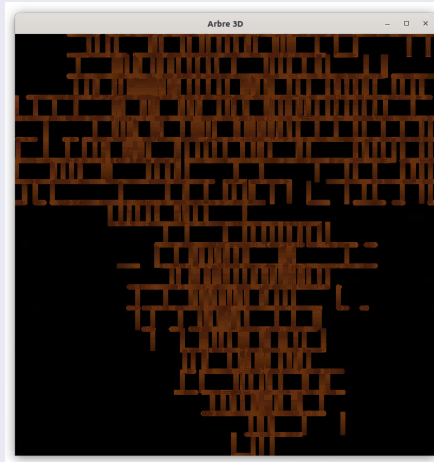
Interpréteur en 2D



Interface de personnalisation



Interpréteur en 3D



Evolution d'un mot

```
fonction readAndApplyRulesToWord(entier n, bar de chargement)
    met la bar de chargement à 0%
    récupère le mot qui se rapproche le plus de la modification voulue
    si le mot n'est pas enregistré l'ajoute dans la backUp
    sinon modifie automatiquement le mot avec sa backUp
    boucle de n itération
        on avance la bar de chargement
        création d'un mot temporaire vide(motTmp)
        boucle parcourant chaque lettre du mot
            modification des lettres precedentes et suivantes
            vérification pour les règles contextuelles
            modification de motTmp selon la règle de la lettre
        le mot de base récupère le mot temporaire
        ajout du mot dans la backUp
    rempli la bar de chargement
```

Figure: L'algorithme readAndApply

Dessin en 2D

```
fonction paintTree2D()  
  boucle parcourant le mot  
    crée une liste de Double (lastCo)  
    ajoute le point de départ x/y de l'arbre dans lastCo avec un angle à 0  
    si le motif = "forward"  
      lance la fonction forward qui trace un trait suivant la position x/y point de départ avec l'angle donné  
      met l'ArrayList lastCo au point d'arrivée du trait avec son angle  
    si le motif = "memorisedPose"  
      sauvegarde la position x/y actuel du point  
    si le motif = "backMemorisedPose"  
      met les valeurs x/y de lastCo à celle de la dernière sauvegarde  
    si le motif = "turnLeft"  
      met la valeur de l'angle de lastCo à elle même - la valeur d'angle de l'arbre  
    si le motif = "turnRight"  
      met la valeur de l'angle de lastCo à elle même + la valeur d'angle de l'arbre
```

Figure: Algorithme permettant de dessiner un arbre en 2D

Dessin en 3D

```
fonction paintTree3D()  
    création des positions x/y/z et des angles x/y/z et les mets à 0  
    boucle parcourant le mot  
        si le motif = "forward"  
            crée un cylindre à partir des positions x/y/z avec les angles de l'arbre  
            modifie les variables x/y/z à la position d'arrivée  
        si le motif = "turnLeft"  
            met la valeur de la variable y à elle + la valeur d'angle de l'arbre  
        si le motif = "turnRight"  
            met la valeur de la variable y à elle - la valeur d'angle de l'arbre  
        si le motif = "pitchDown"  
            met la valeur de la variable x à elle + la valeur d'angle de l'arbre  
        si le motif = "pitchUp"  
            met la valeur de la variable x à elle - la valeur d'angle de l'arbre  
        si le motif = "rollLeft"  
            met la valeur de la variable z à elle + la valeur d'angle de l'arbre  
        si le motif = "rollRight"  
            met la valeur de la variable z à elle - la valeur d'angle de l'arbre  
        si le motif = "turnAround"  
            met la valeur de la variable y à elle + 180  
        si le motif = "savePose"  
            sauvegarde la position x/y/z actuel du point  
        si le motif = "savePose"  
            met les valeurs x/y/z du point à celle de la dernière sauvegarde
```

Figure: Algorithme permettant de dessiner un arbre en 3D

Thread de la frame de personnalisation

```
/**
 * Cette méthode est l'action a faire lorsque le bouton est cliqué.
 * @param e l'action performé
 * @ensures thread != null
 */
@Override
public void actionPerformed(ActionEvent e){
    if(this.getInterface2d().getInterfacePerso().isVisible() == false){
        Thread thread = new Thread(this);
        thread.start();
    }
}

/**
 * Cette méthode est l'action lancé grace au thread .
 * @ensures this.getInterface2d().isVisible() == true
 */
@Override
public void run(){
    this.getInterface2d().afficheFramePerso();
}
```

Thread du chargement d'un arbre

```
/**
 * Cette méthode est le actionPerformed lorsque l'on clique sur le bouton qui lance le thread de chargement du mot.
 * @param e argument obligatoire de la méthode actionPerformed
 */
@Override
public void actionPerformed(ActionEvent e){
    if(this.getInterface2d().getWord() != null){
        Thread thread = new Thread(this);
        thread.start();
    }
}

/**
 * Cette méthode est le run du thread qui est lancé depuis le actionPerformed du bouton qui charge un wordGraph.
 */
@Override
public void run(){
    this.getInterface2d().setWordGraph(this.rulesGraphics, taille.setOccu());
}
```

Figure: Thread permettant de charger un arbre

Exemple de code de nos tests

```
//Initialisation de la class
RuleNormal ruleTest1 = new RuleNormal(letter:"A", rule:"AA");
RuleNormal ruleTest2 = new RuleNormal(letter:"B");
RuleNormal ruleTest3 = new RuleNormal(letter:"C", rule:"CC");

//Test getKey()
assert ruleTest1.getKey().equals("A") : "Test failed: getKeys()";
assert ruleTest2.getKey().equals("B") : "Test failed: getKeys()";
assert ruleTest3.getKey().equals("C") : "Test failed: getKeys()";
```

Résultats de nos tests

```
Terminal
/PROJET_1/l-systemes_plam/branches$ ant test
Buildfile: /media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches/build.xml

compile:
[javac] Compiling 36 source files to /media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches/bin
[javac] Note: /media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches/src/display2D/utilities/TreeDrawer.java uses unchecked or unsafe operations.
[javac] Note: Recompile with -Xlint:unchecked for details.

test:
[java] RuleNormal: All tests passed.
[java] RuleStochastic: All tests passed.
[java] RuleContextuel: All tests passed.
[java] RulesMap: All tests passed.
[java] Word: All tests passed.
[java] WordBackUp: All tests passed.

BUILD SUCCESSFUL
Total time: 2 seconds
(focal-base r4005)21907996@V304V-FCP055: /media/21907996/homelinux/$TRAVAIL/L2/S2/PROJET_1/l-systemes_plam/branches$
```


Les objectifs du projet

- Les problèmes rencontrés et les points à améliorer
- Notre ressenti sur le projet