

ANALYSEUR DE LIVRE DONT VOUS ETES LE HERO (LDVELH)

KAMGANG KENMOE Miguel Jordan
SISSOKO Dioukou Moussa
ABOGOUNRIN Ayath
ALHAZZAA Laith

UE : Conception logicielle 1

Université de Caen Normandie

18 avril 2024

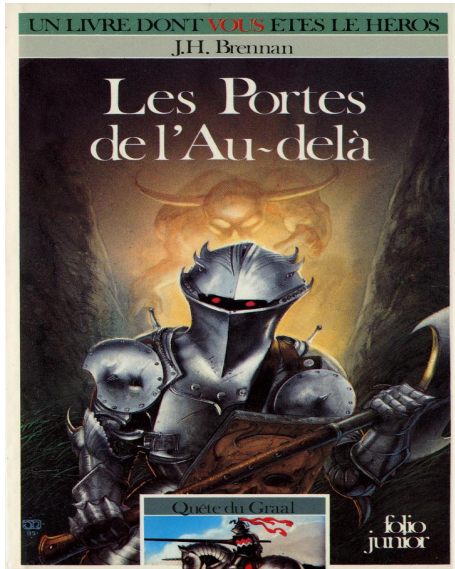
- 1 • Introduction
- 2 • Objectifs du projet
- 3 • Fonctionnalités implémentées
- 4 • Éléments techniques
- 5 • Conclusion

Introduction

Choix du projet

- Notre intérêt pour les livres dont vous êtes le héros.
- Apprendre la notion des graphes.

Introduction



- C'est quoi un livre dont vous êtes le héros ?
- C'est quoi un analyseur de livre ?

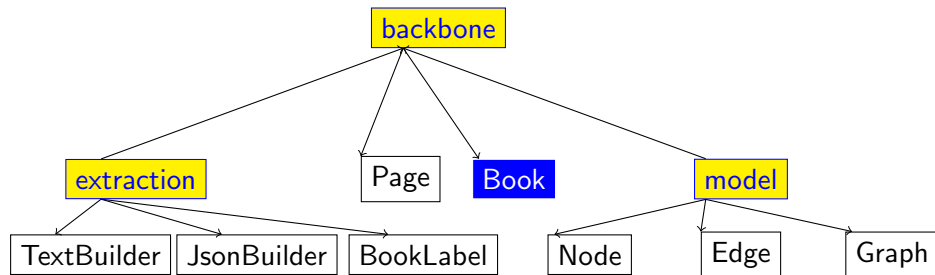
Objectifs du projet

Description détaillée

- Obtenir un graphe à partir d'un fichier JSON ou d'un fichier TXT.
- Afficher un graphe à l'aide d'un algorithme de force.
- Implémenter différents algorithmes pour analyser le graphe.
- Étendre le modèle du graphe.

Fonctionnalités implémentées

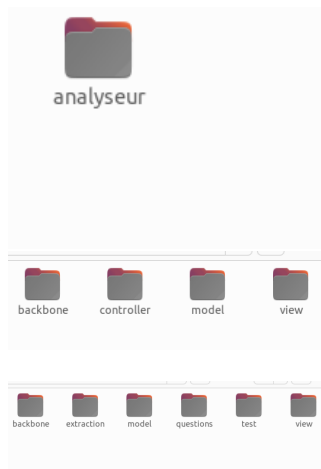
Descriptions des fonctionnalités



Fonctionnalités implémentées

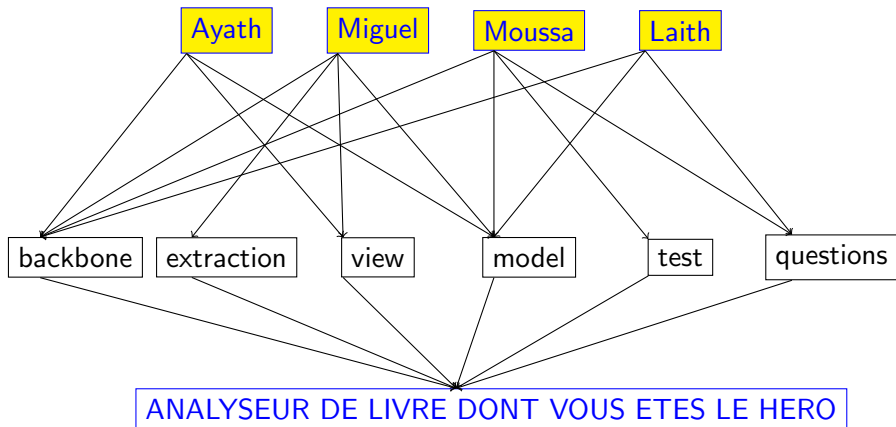
Description des paquetages

- ① Racine
- ② Système MVC
- ③ Notre graphe est fixe et l'inutilité de controller
- ④ Le changement et son utilité pour notre graphe



Éléments techniques

Répartition des tâches



Dans ce projet nous avons exploré plusieurs algorithmes tels que :

- Fruchterman Reingold
- Kamada-Kawai
- Dijkstra
- Breath First Search
- Depth First Search Directed Acyclic Graphs
- Tri topologique

Éléments techniques

Pseudocode de l'algorithme de Fruchterman-Reingold (Partie 1)

Entrées : Largeur W , Longueur L , Ensemble de Sommets V , Ensemble d'Arêtes E , Nombre d'itérations

Sorties : Positions finales des sommets

```
1  $area \leftarrow W \times L$ 
2  $G \leftarrow (V, E)$ 
3  $k \leftarrow \sqrt{\frac{area}{|V|}}$ 
4 pour  $i \leftarrow 1$  à  $iterations$  faire
5     pour  $v$  in  $V$  faire
6          $v.\text{disp} \leftarrow (0, 0)$ 
7         pour  $u$  in  $V$  faire
8             si  $u \neq v$  alors
9                  $\delta \leftarrow v.\text{pos} - u.\text{pos}$ 
10                 $v.\text{disp} \leftarrow v.\text{disp} + \frac{\delta}{|\delta|} \times \text{fr}(|\delta|)$ 
11            fin
12        fin
13    fin
```

Éléments techniques

Pseudocode de l'algorithme de Fruchterman-Reingold (Partie 2)

Entrées : Largeur W , Longueur L , Ensemble de Sommets V , Ensemble d'Arêtes E , Nombre d'itérations

Sorties : Positions finales des sommets

```
1 pour  $e$  in  $E$  faire
2    $\delta \leftarrow e.v.pos - e.u.pos$ 
3    $e.v.disp \leftarrow e.v.disp - \frac{\delta}{|\delta|} \times fa(|\delta|)$ 
4    $e.u.disp \leftarrow e.u.disp + \frac{\delta}{|\delta|} \times fa(|\delta|)$ 
5 fin
```

Éléments techniques

Pseudocode de l'algorithme de Fruchterman-Reingold (Partie 3)

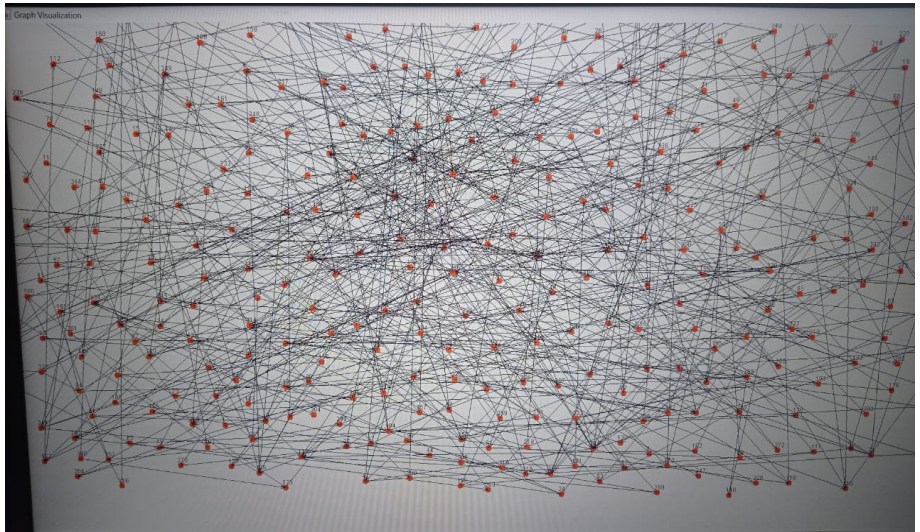
Entrées : Largeur W , Longueur L , Ensemble de Sommets V , Ensemble d'Arêtes E , Nombre d'itérations

Sorties : Positions finales des sommets

```
1 pour  $v$  in  $V$  faire  
2    $v.pos \leftarrow v.pos + \frac{v.disp}{|v.disp|} \times \min(v.disp, t)$   
3    $v.pos.x \leftarrow \min(\frac{W}{2}, \max(-\frac{W}{2}, v.pos.x))$   
4    $v.pos.y \leftarrow \min(\frac{L}{2}, \max(-\frac{L}{2}, v.pos.y))$   
5 fin  
6  $t \leftarrow cool(t)$ 
```

Éléments techniques

Affichage du graphe sans le facteur de refroidissement



A présent montrons à quoi ressemble le projet une fois exécuté, à savoir que son exécution est divisée en trois :

- Affichage du graphe dans une interface graphique : `java -jar Graph.jar`
- Affichage dans la console des éléments d'analyse du graphe : `java -jar Analyse.jar`
- Affichage du résultat des Tests effectués sur le graphe : `java -jar Test.jar`

Conclusion

Récapitulatif des fonctionnalités principales

Récapitulatif

Conclusion

Propositions d'améliorations

- Implémentation de l'algorithme de Dijkstra afin de voir la plus rapide parmi les deux et d'opter pour ça.
- Revoir l'affichage de notre graphe pour avoir le même graphe à chaque fois qu'on lance le main, ne plus avoir des dispositions aléatoires des nœuds.
- Ajouter les inventaires.