
The Maze

— Mael Pena - Verner Boisson - 2020 —

Sommaire

- Présentation de l'équipe
- Présentation du projet
- Présentation des outils utilisés
- La base de donnée
- L'API
- La base de donnée
- L'IHM d'édition
- L'IHM de résolution
- L'algorithme

Présentation de l'équipe

Verner Boisson

Bachelor 3 à Ynov



Mael Pena

Bachelor 3 à Ynov



Objectif du projet

- Faire une base de donnée afin de stocker les labyrinthes.
- Faire une API qui permet de communiquer avec cette base de donnée.
- Faire une interface homme-machine qui permet de créer et éditer un labyrinthe.
- Faire une deuxième interface homme-machine qui permet de résoudre un labyrinthe.

Les outils utilisé - BDD

Base de donnée choisi : Mongo

- Base de donnée orientée document.
- Documenté
- Technologie déjà utilisée

ORM choisi : Mongoose

- Technologie déjà utilisée



mongoose

Les outils utilisé - API

NodeJS : API

- Création Web-Serveur accessible

Dépendances utilisées :

- Express
- Mongoose
- Method-override



Les outils utilisé - IHMs



JAVA :

Langage Orienté Objet

-
- Librairie Java utilisé :
 - Swing (gestion de l'interface)
 - HttpURLConnection (communication avec l'API)
 - Awt (gestion layout, grille du labyrinthe, evenement de la souris)

La base de donnée

Les documents sont dans le format :

```
{  
  id: {type: Number, required: true, unique: true},  
  title: {type:String},  
  author: {type:String},  
  maze: [[{type:String}]],  
  createdAt: {type: Date, default: new Date()},  
  timer: {type: Number},  
  movement: {type: Number},  
}
```


L'API

Fonctionnalités (CRUD) :

- Lire un labyrinthe
- Poster un labyrinthe
- Editer un labyrinthe
- Supprimer un labyrinthe

L'interface d'édition

- Fonctionnalités :
 - Choisir la taille
 - Positionner/supprimer une entité
 - Générer un labyrinthe aléatoire
 - Enregistrer le labyrinthe
 - Modifier un labyrinthe
 - Supprimer un labyrinthe
 - Mettre un titre
 - Mettre un auteur

L'interface de resolution

Fonctionnalités :

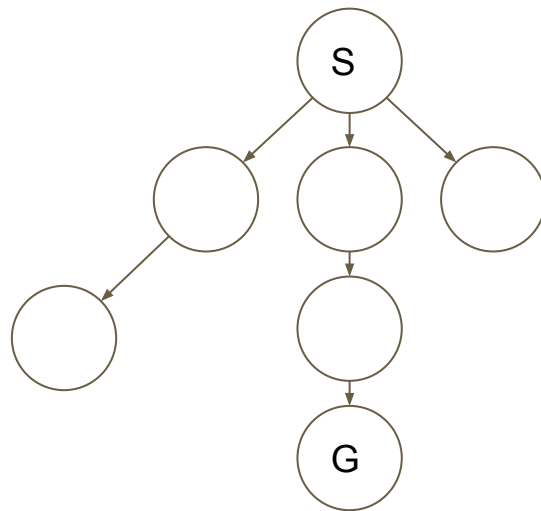
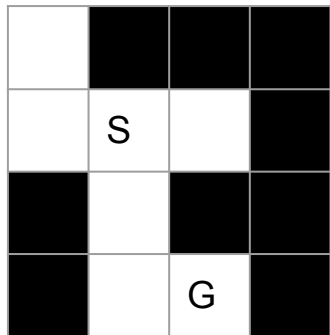
- Lire un labyrinthe
- Choisir un labyrinthe
- Rafraîchir la liste de labyrinthe
- Lancer la resolution
- Enregistrer le résultat
- Récupère tous les labyrinthes
- Arrêter la résolution
- Effacer la résolution

L'algorithme

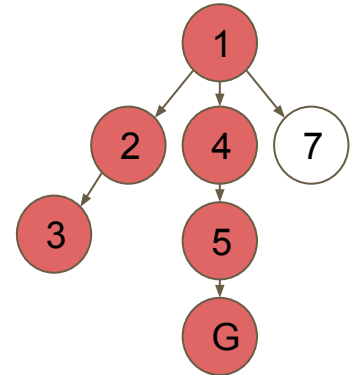
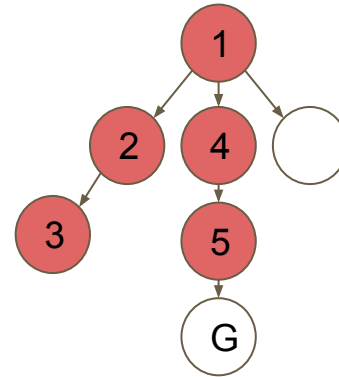
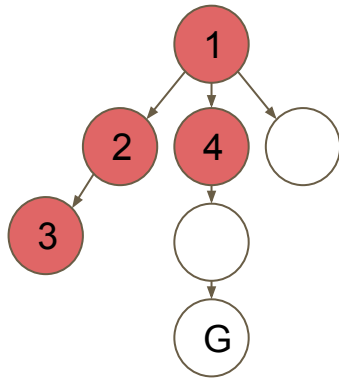
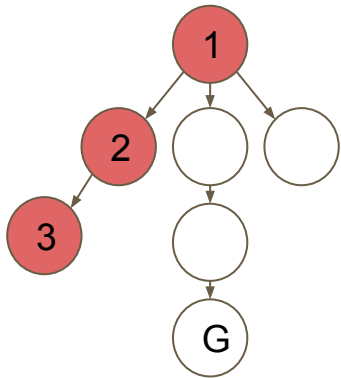
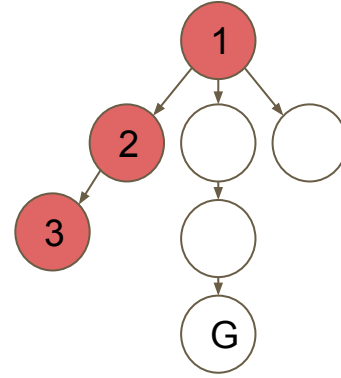
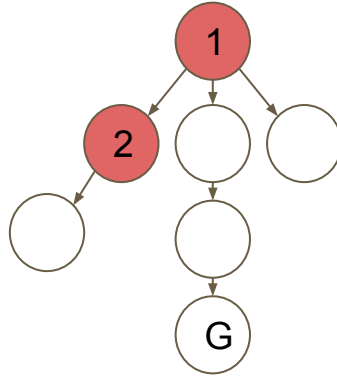
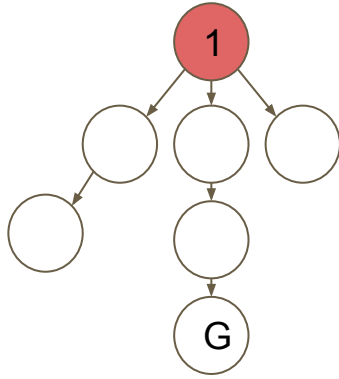
Choix de l'algorithme de résolution : Parcours en profondeur (DFS)

- Algorithme adapté à la résolution de Labyrinthe
- Algorithme adapté à la génération aléatoire de Labyrinthe

Représentation du Labyrinthe



Principe de l'algorithme de parcours en profondeur



Conclusion

Point à améliorer :

- L'organisation
- Choix de plusieurs algorithmes de résolution
- Ajout de fonctionnalité dans la résolution

Compétences acquises :

- Algorithmie
- Graphe
- Récursivité
- Threading

On vous remercie de votre écoute