

## **Cahier des charges : Logik Ville**

### **1) Présentation du projet**

Le projet consiste à implanter le jeu de société Logik Ville pour pouvoir jouer au travers d'une application sur ordinateur. L'interface sera abordable et compréhensible pour les utilisateurs.

Logik Ville est un jeu accessible pour les enfants dès 4 ans pour les niveaux de base et dès 10 ans pour les plus compliqués.

C'est un jeu de logique contenant 84 énigmes : il faut trouver dans quelle maison se trouvent les habitants et les animaux, une seule réponse est admissible.

Le jeu de base se joue entre 3 et 5 maisons. Chaque maison contient au moins un personnage et peut contenir, selon la difficulté, un animal en plus. On peut avoir 3 à 8 contraintes agissant sur une même carte. Une contrainte agit sur une ou deux entités. Une entité étant un animal ou un personnage.

Nous retrouvons 7 types de contraintes différentes :

- On impose la maison d'une entité
- On interdit à une entité d'habiter dans une maison
- On impose que 2 entités soient voisines
- On interdit que 2 entités soient voisines
- On impose la position d'une entité par rapport à une autre
- On impose qu'un personnage et un animal vivent dans la même maison
- On interdit qu'un personnage et un animal vivent dans la même maison

### **2) Contraintes**

Pour implanter le jeu, nous réaliserons une interface graphique 'belle' et accessible qui doit être utilisée à la souris.

Les solutions pour chaque énigme doivent pouvoir être calculées automatiquement via un solveur qui doit être optimisé et efficace en temps.

L'interface sera personnalisable (par exemple : les images seront modifiables, ...) via les dossiers ou via l'application.

Le code sera organisé sous l'architecture MVC.

Il n'y a pour l'heure aucun état de projet existant, le projet devra être fidèle au jeu de plateau. Pour réaliser cette application, nous avons à notre disposition un Git, lieu de dépôt et partage de code pour faciliter nos échanges.

### **3) Fonctionnalités potentielles :**

Le jeu sera réalisé en java.

Nous donnerons la possibilité à l'utilisateur d'ajouter des cartes selon ses contraintes tout en affichant les solutions disponibles ou seulement le nombre s'il y en a encore beaucoup. S'il n'y a pas de solution, la contrainte problématique sera affichée.

Il sera possible potentiellement d'ajouter de nouveaux personnages et animaux mais le nombre de l'un doit être égal au nombre de l'autre.

Un site web présentant l'application sera également disponible, les règles du jeu y seront expliquées.

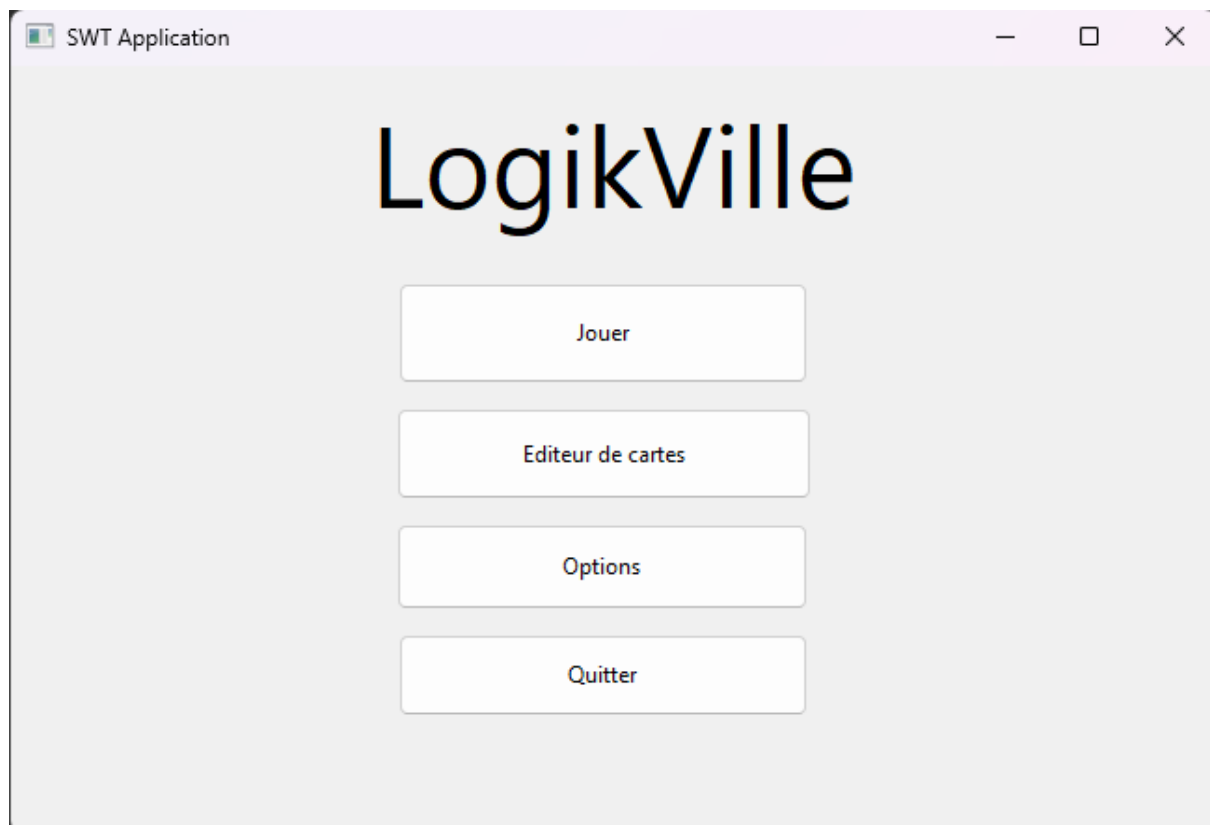
Il sera possible de charger un thème de jeu (un univers particulier sur lequel jouer).

2 niveaux bonus : une course contre la montre où le but est de faire le plus de niveaux possibles dans le temps imparti | chronométrer le temps mis pour chaque niveau avec un record personnel – le temps mis pour une liste aléatoire (ceci est un ajout optionnel).

#### **4) Concept graphique**

Afin de donner un visuel sur le rendu possible, nous avons réalisé quelques concepts graphiques. Ces visuels sont cependant voués à évoluer lors de la réalisation.

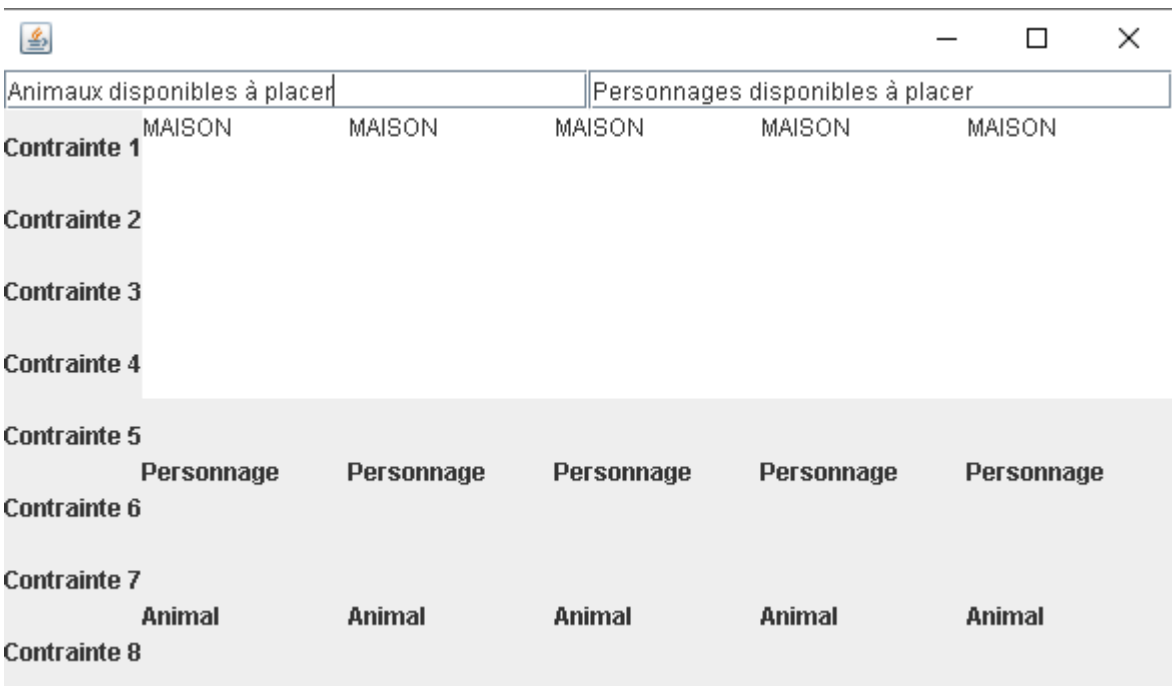
Écran d'accueil :



Choix de niveau :



Jouer :



Editeur de carte:

Carte

Choix du nombre de maisons :

3 >

Souhaitez-vous jouer avec des compagnons

Non >

Suivant >

Ajouter le niveau

Carte

5 maisons

Animaux : Oui

Ajouter le niveau

Contrainte 1

>

Maçon >

Maison 1 >

Ajouter

Annuler <-

Solutions

58 solutions

5 maisons

Animaux : Non

Maçon est dans la maison 1

Plongeur est dans la maison à côté de Pompier

Pompier est dans la maison 2

Ajouter le niveau

Contrainte 1

>

Maçon

>

Maison 1

>

Ajouter

Annuler <-

Solutions

2 solutions

Maçon dans maison 1

Pompier dans maison 2

Plongeur maison 3

Policier dans maison 4

Cuisinier dans maison 5

Maçon dans maison 1

Pompier dans maison 2

Plongeur maison 3

Cuisinier dans maison 4

Policier dans maison 5

Menu d'options :

Options

Plein écran

☐

Résolution :

Automatique

Charger un thème






classic

Créateur de thèmes

Appliquer

Retour

Créateur de thèmes :

				
Texte	Texte	Texte	Texte	Texte
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

				
Texte	Texte	Texte	Texte	Texte
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Sélectionner un thème	Thème <input type="text"/>
<input type="button" value="Charger le thème"/>	
<input type="button" value="Supprimer le thème"/>	

Entrez le nom du thème : <input type="text"/>
Si vide, le thème chargé sera modifié

<input type="button" value="Réinitialiser"/>	<input type="button" value="Créer le thème"/>	<input type="button" value="Retour"/>
----------------------------------------------	-----------------------------------------------	---------------------------------------

## **5) Répartition des tâches**

Pour mener à bien ce projet, nous devons établir avec l'ensemble des membres un ensemble des classes nécessaires et des besoins pour mener à bien vue et modèle de l'application.

Pour ce faire, nous avons également réparti nos rôles de la façon suivante :

Florian : Développements de classes/ modèle

Axelle : Développements de classes/ modèle / Interfaces graphiques

Etienne : Interfaces graphiques