



SORBONNE UNIVERSITÉ / CFA DES SCIENCES

Projet tuteuré Quoridor

Auteur : Le groupe "Trollidor"

Devamadushan THEVARANCHAN

Tamer ABOU KARROUM

Yoan LE NEVEZ

Rémi RAJARATNAM

Chargé de TD: Sébastien TIXEUIL

Résumé

Le projet s'inscrit dans le cadre de l'UE projet tuteuré LU3IN117 visant à développer un logiciel ludique et stratégique inspiré du jeu de plateau Quoridor. Ce projet offre l'opportunité de combiner plusieurs compétences techniques, notamment la conception d'interfaces graphiques modernes, et l'implémentation d'une intelligence artificielle avancée.

1.1 But du Jeu

Le but fondamental du jeu Quoridor est de traverser le plateau en déplaçant son pion depuis sa position de départ jusqu'à atteindre la ligne opposée. Chaque joueur commence la partie positionnée sur une ligne spécifique (par exemple, l'un en haut et l'autre en bas dans une partie à deux joueurs) et doit trouver le chemin le plus rapide et stratégique pour parvenir à l'objectif.

Chaque tour, le joueur peut choisir de déplacer son pion d'une case adjacente (horizontalement ou verticalement) pour progresser vers la ligne d'arrivée. Dans certaines situations, lorsqu'un adversaire bloque directement le chemin, le pion peut effectuer un saut pour contourner l'obstacle.

En plus de déplacer son pion, le joueur dispose d'un nombre limité de murs qu'il peut placer sur le plateau. Ces murs servent à créer des barrières qui ralentissent la progression de l'adversaire. Le challenge consiste à bien doser entre l'avancement personnel et le blocage de l'itinéraire adverse, sans compromettre son propre parcours.

La limitation du nombre de murs impose une gestion stratégique de ces ressources. Placer un mur au mauvais moment ou dans une mauvaise position peut se retourner contre le joueur, car il pourrait se retrouver lui-même bloqué.

Avant chaque coup, le joueur doit analyser le plateau, anticiper les mouvements adverses et envisager plusieurs itinéraires possibles pour atteindre la ligne opposée. La capacité à prévoir plusieurs coups à l'avance est cruciale pour se dégager des impasses.

L'état du plateau évolue constamment en raison du placement des murs. Ainsi, même une stratégie soigneusement planifiée doit pouvoir être adaptée en fonction des actions de l'adversaire. L'anticipation et la réactivité sont des atouts majeurs pour réussir.

Une règle essentielle du jeu est que, même avec le placement de murs, il doit toujours rester au moins un chemin viable pour chaque joueur. Cette contrainte garantit que le jeu reste dynamique et accessible, tout en forçant les joueurs à optimiser l'utilisation de leurs murs.

Synthèse

En résumé, le but du jeu Quoridor est de parvenir le premier à atteindre la ligne opposée du plateau, tout en utilisant intelligemment le déplacement du pion et le placement des murs pour contrer son adversaire. Chaque partie est une véritable épreuve de réflexion et de tactique.

1.2 Options retenues

Afin de concevoir le jeu du Quoridor nous avons choisis deux options.

Une interface graphique moderne

L'interface utilisateur est développée en **JavaFX**, une technologie adaptée aux applications graphiques interactives en Java. Ce choix est motivé par plusieurs facteurs :

Expérience et maîtrise de l'équipe : Tous les membres du groupe ont déjà utilisé Java dont JavaFX, ce qui réduit la courbe d'apprentissage et optimise le temps de développement.

Ergonomie et accessibilité : L'objectif est d'offrir une interface fluide, intuitive et agréable à utiliser, où les joueurs peuvent interagir aisément avec le plateau et les options de jeu.

Flexibilité et modularité : JavaFX permet une séparation claire entre la logique et l'affichage grâce à l'utilisation de fichiers FXML, facilitant ainsi la maintenance et l'amélioration future de l'interface.

Jouer contre un ordinateur intelligent

L'un des axes majeurs du projet est l'intégration d'une intelligence artificielle pour offrir un mode de jeu contre l'ordinateur. L'IA sera développée en Python.

Richesse des bibliothèques IA : Python est reconnu pour sa vaste gamme de bibliothèques spécialisées en intelligence artificielle et en optimisation de décisions, facilitant l'implémentation d'un adversaire performant.

Flexibilité et rapidité de développement : Grâce à sa syntaxe claire et concise, Python permet de tester rapidement différentes approches stratégiques sans nécessiter une lourde implémentation. *Expérience et expertise de l'équipe* : Plusieurs membres du groupe maîtrisent déjà Python et ses outils d'IA, ce qui permet une intégration efficace de l'intelligence artificielle sans difficulté technique majeure.

Adaptabilité et évolutivité : L'utilisation de Python pour l'IA permet de futurs développements basés sur l'apprentissage automatique, offrant ainsi la possibilité d'adapter l'adversaire virtuel au niveau des joueurs en fonction de leurs performances.

1.2.1 Architecture et Technologies Adoptées

Nous avons conçu une architecture modulaire permettant une séparation claire des différentes couches du projet :

Backend: moteur de jeu en Java Le cœur du jeu repose sur un backend développé en Java, qui gère :

La logique du jeu et l'application des règles définies

La gestion des mouvements et des positions des entités (joueurs et murs)

La validation et la vérification des placements de murs

Ce backend assure un fonctionnement fiable du jeu en vérifiant en temps réel la validité des actions effectuées par le joueur ou l'IA.

Frontend JavaFX pour l'affichage L'affichage du jeu est géré par JavaFX permettant d'implémenter l'interface et d'interagir avec la logique du jeu en backend.

Gestion des modules et dépendances Le projet sera organisé en modules et utilisera Maven pour la gestion des dépendances (définies dans **pom.xml**).

1.2.2 Perspectives d'amélioration

Une amélioration envisageable serait le déploiement via Docker pour simplifier la distribution et la gestion des versions de l'application.

En résumé, nous avons opté pour une combinaison de **Java** pour le backend et la logique de jeu car tous les membres ont déjà utilisé ce langage, Python l'intelligence artificielle, et JavaFX pour une interface moderne et réactive.

1.3 Le groupe

Notre équipe se compose de plusieurs membres aux compétences complémentaires. Bien que les noms et rôles définitifs soient à confirmer, nous envisageons la répartition suivante :

Développement de l'intelligence artificielle (Python)

Devamadushan THEVARANCHAN

Tamer ABOU KARROUM

Développement de l'interface graphique (JavaFX)

Yoan LE NEVEZ

Rémi RAJARATNAM

1.3.1 Mini-CV

Voici nos minis CV avec les projets et expériences pertinentes vis-à-vis du projet se trouvent ci-dessous.

Note: Les rôles pourront être ajustés en fonction de l'évolution du projet.

1.4 Spécification fonctionnelle

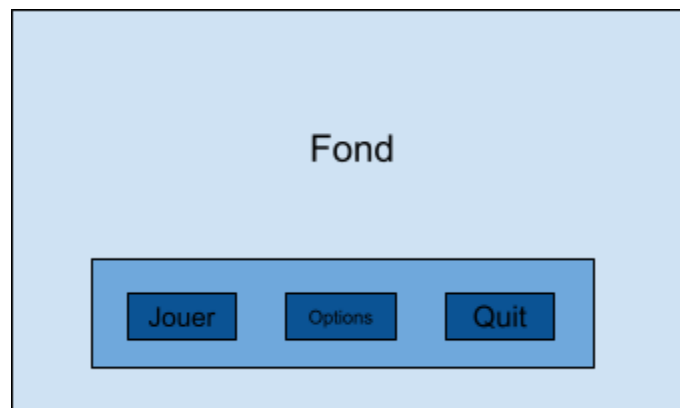
L'application **Trolidor** est un jeu de stratégie au tour par tour inspiré de *Quoridor*, avec une interface moderne, minimaliste et immersive.

Elle est conçue pour offrir une **expérience de jeu fluide**, adaptable à différents contextes (1v1, IA, jeu à 4 joueurs).

Toutes les interactions sont accessibles via une interface **cliquable et graphique** construite avec JavaFX.

Menu principal

À l'ouverture de l'application, un menu principal est affiché avec les boutons suivants :



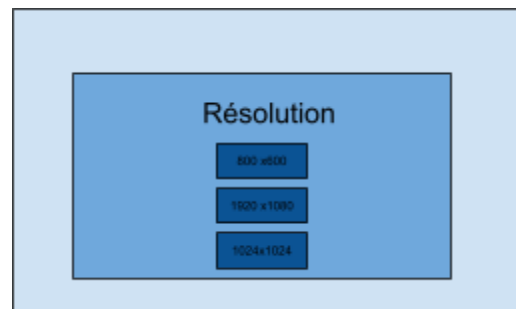
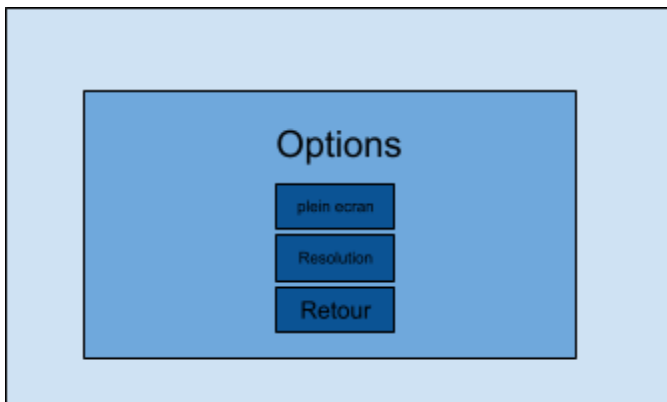
Jouer

- Ouvre une **fenêtre de configuration de partie** contenant les paramètres suivants :
 - **Nombre de joueurs** : 2 ou 4.
 - **Type de partie** :
 - Joueur vs IA
 - Joueur vs Joueurs en local
 - **Nombre de murs par joueur** : champ personnalisable (ex : 10, 12, 5...).
 - **Bouton "Lancer la partie"** :
 - Vérifie les champs saisis,
 - Initialise le plateau et les joueurs (avec IA ou non),
 - Démarre la partie.

The diagram illustrates the layout of the game configuration window. It features a light blue background with several dark blue rectangular buttons. At the top, a button is labeled 'Cb de joueur (2 a 4)' and 'Bot ou jr'. Below it is a button labeled 'Cb de mur'. At the bottom, there is a container box containing two buttons: 'Lancer' on the left and 'Retour' on the right.

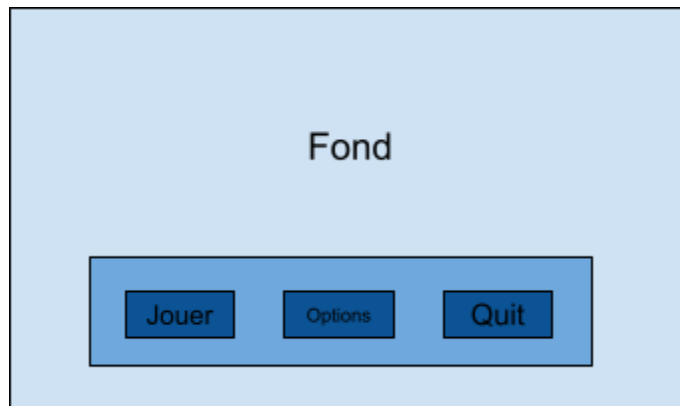
Options

- Ouvre un sous-menu avec :
 - **Choix de résolution** :
 - 800x600 (
 - 1024x1024 (par défaut)
 - 1920x1080
 - **Mode plein écran** : checkbox
 - **Bouton "Retour"** : revient au menu principal



Quitter

- Ferme l'application proprement



2. Paramétrage de la partie

Lorsqu'on appuie sur "**Jouer**", un **écran intermédiaire** permet de configurer la session :

Paramètre	Type
Nombre de joueurs	2 ou 4
Adversaire	IA ou Humain
Murs par joueur	Champ numérique
Lancer la partie	Bouton

3. Interface de jeu

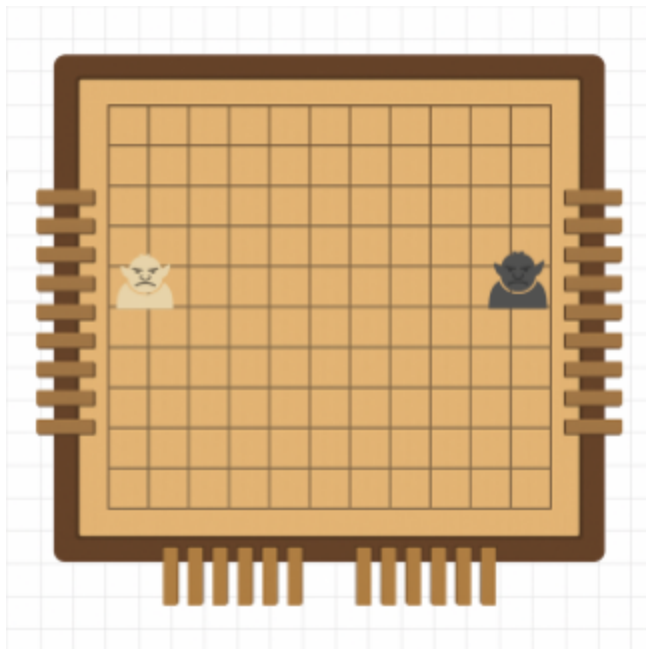
Une fois la partie lancée, l'utilisateur voit :

- Le **plateau 9x9** (ou adapté selon le mode),
- Les **pions des joueurs** représentés par des images (trolls),
- Les **emplacements de murs disponibles** (représentés par "+"),
- Une **zone d'infos** : joueur courant, nombre de murs restants.



Interaction :

- Les **cases valides** pour un déplacement sont **colorées en vert**.
- Le joueur clique sur une **case valide** pour déplacer son pion.
- Il peut cliquer sur un emplacement pour **placer un mur**, après avoir choisi l'orientation (**Horizontal** ou **Vertical**).



4. Intelligence Artificielle (a definir)

L'IA joue automatiquement à son tour :

- *Elle choisit un déplacement ou un mur de façon stratégique (ou aléatoire au début).*
- *Elle respecte les règles du jeu : pas de blocage total.*

6. Gestion des erreurs et exceptions

Tous les cas d'erreur sont anticipés avec des **alertes bloquantes ou informatives** :

Situation	Comportement attendu
Joueur tente de poser un mur alors qu'il en a 0	Alerte : <i>"Vous n'avez plus de murs disponibles."</i>
Placement d'un mur sur un emplacement existant	Alerte : <i>"Un mur est déjà présent à cet endroit." ou ignoré</i>
Le mur bloque totalement le passage	Alerte : <i>"Chaque joueur doit pouvoir atteindre sa cible."</i>
Fichier FXML, CSS ou image manquant	Message console + comportement de secours
Mauvaise saisie dans les options	Alerte : <i>"Veuillez entrer une valeur numérique valide."</i>
Clic sur case invalide	Ignoré, aucune action, aucune exception

Résumé des comportements utilisateur

Action utilisateur	Résultat
Clique sur Jouer	Ouvre une fenêtre de configuration
Clique sur Lancer partie	Démarre la partie selon les choix
Clique sur Options	Ouvre les paramètres d'affichage
Clique sur Résolution	Change dynamiquement la taille de la fenêtre
Clique sur Retour	Revient au menu principal
Place un mur invalide	Message bloquant
Termine la partie	Affiche une fenêtre "Victoire" avec options "Rejouer" ou "Menu principal"