Projet IHM R2.02

Rapport final

Groupe 11

HEINRY Noan

Mai 2025

Sommaire

1. Structure de l'Application

```
src
└── main
    ├── java
    │   ├── module-info.java
    │   └── com.example.tictactoe
    │       ├── FinPartieController.java
    │       ├── JeuController.java
    │       ├── MainJar.java
    │       ├── Morpion.java
    │       ├── MorpionController.java
    │       ├── NomJoueurController.java
    │       ├── ScoreManager.java
    │       ├── ScoresController.java
    │       └── TopBarController.java10
    │
    └── resources.com.example.tictactoe
        ├── Accueil.fxml
        ├── FinPartie.fxml
        ├── JeuMorpion.fxml
        ├── NomJoueur.fxml
        ├── Scores.fxml
        └── TopBar.fxml
    │
    ├── fonts
    │   ├── lithos.ttf
    │
    ├── Image
    │   ├── BoutonJeu.png
    │   ├── BoutonJeuHover.png
    │   ├── BoutonQuitter.png
    │   ├── BoutonQuitterHover.png
    │   ├── BtnAcceuil.png
    │   ├── BtnAcceuilHover.png
    │   ├── BtnRegles.png
    │   ├── BtnReglesHover.png
    │   ├── BtnReset.png
    │   ├── BtnResetHover.png
    │   ├── Gold_Coin.png
    │   ├── LogoMorpHalla.png
    │   ├── LogoMorphallaTaskBar.png
    │   ├── MC_Coin.png
    │   └── TourDe.png
    ├── sounds
    │   └── WelcomeToBrawlhalla.wav
    │
    └── styles
        └── style.css
```

Choix de Conception :

- ➔ Utilisation du **modèle MVC** : un contrôleur par vue pour une meilleure organisation.
- ➔ Centralisation des ressources (images, sons, polices) pour simplifier la gestion et l'organisation du projet.
- ➔ Persistance des scores grâce à la classe Properties, stockée dans un fichier externe afin de conserver les scores entre les sessions.
- ➔ Utilisation de **FXML et CSS** pour la stylisation de l'application
- ➔ Conception de l'ensemble des assets pour un rendu qui me correspond (boutons, logos, couleurs)

2. Fonctionnalités :

Fonctionnalités Obligatoires Présentes :

- ➔ **Un titre sur chaque fenêtre** : Présent avec la méthode stage.setTitle("...").
- ➔ **Plusieurs fenêtres (modales et non modales)** :

  Fenêtres : Accueil, Choix des noms, Jeu, Fin de partie, Scores.

  Fenêtre de fin de partie modale (bloque l'interaction avec la fenêtre de jeu), fenêtre pour les règles .
- ➔ **Barre de menu permettant de réinitialiser, voir les règles et quitter :**

  Réinitialiser le jeu : ✅ Présent.

  Quitter l'application : ✅ Présent.

  Voir les règles : ✅ Présent**.**
- ➔ **Nomination des deux joueurs :** Interface de saisie des noms via NomJoueur.fxml.
- ➔ **Choix du joueur qui commence (J1, J2 ou aléatoire)** : Disponible dans l'écran de choix des noms.
- ➔ **Zone de jeu avec 9 cases (3x3)** : Implémentée avec des boutons dans un GridPane.
- ➔ **Affichage des symboles O et X et alternance automatique des tours** :

  Présent, avec images personnalisées à la place des simples O/X.
- ➔ **Indication claire du joueur dont c'est le tour** :

  L'icône du joueur dont c'est le tour est affichée à droite de la grille de jeu.
- ➔ **Fin de jeu automatique avec indication du vainqueur ou match nul** :

  Détection de victoire et affichage d'une fenêtre de fin de partie avec le résultat qui bloque l'interaction avec la fenêtre de jeu..
- ➔ **Interface pour rejouer ou quitter l'application après la fin du jeu** :
- ➔ **Tableau de scores avec nombre de victoires de chaque joueur** :

  Implémenté via la classe Properties dans un fichier externe.

Fonctionnalités Supplémentaires Implémentées :

- ➔ Lecture d'un son d'accueil à l'ouverture de l'application.
- ➔ Thème graphique personnalisé avec images et polices dédiées.
- ➔ Icône personnalisée pour la fenêtre d'application.
- ➔ Système de persistance des scores entre les sessions via un fichier .properties.
- ➔ Ajout d'une barre de navigation réutilisable (TopBar) sur toutes les fenêtres.
- ➔ Animation des éléments graphiques (hover sur les boutons, icônes dynamiques).
- ➔ Saisie et affichage des images personnalisées pour remplacer les symboles O et X.

Fonctionnalités non implémentées :

- ➔ **Mode de jeu contre une Intelligence Artificielle (IA) :**
  Je n'ai pas implémenté de mode IA pour deux raisons principales. D'une part, je ne souhaitais pas consacrer trop de temps à cette fonctionnalité afin de me concentrer sur les autres aspects du projet. D'autre part, j'avais déjà réalisé une IA pour le Morpion en Python par le passé, et je ne trouvais donc pas cette partie particulièrement intéressante à refaire dans ce projet.

- ➔ **Montrer l'alignement gagnant (surbrillance des cases, barre, etc.).**

- ➔ **Permettre aux joueurs de personnaliser leur symbole (choix d'image ou de texte personnalisé).**

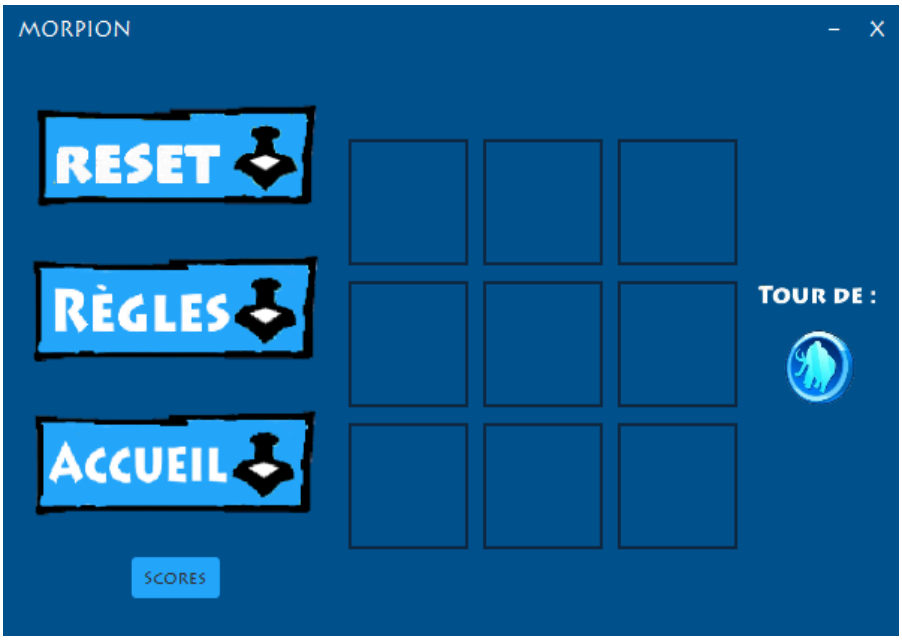
- ➔ **Permettre de choisir la taille de la grille pour des variantes du Morpion.**

3. Vues FXML :

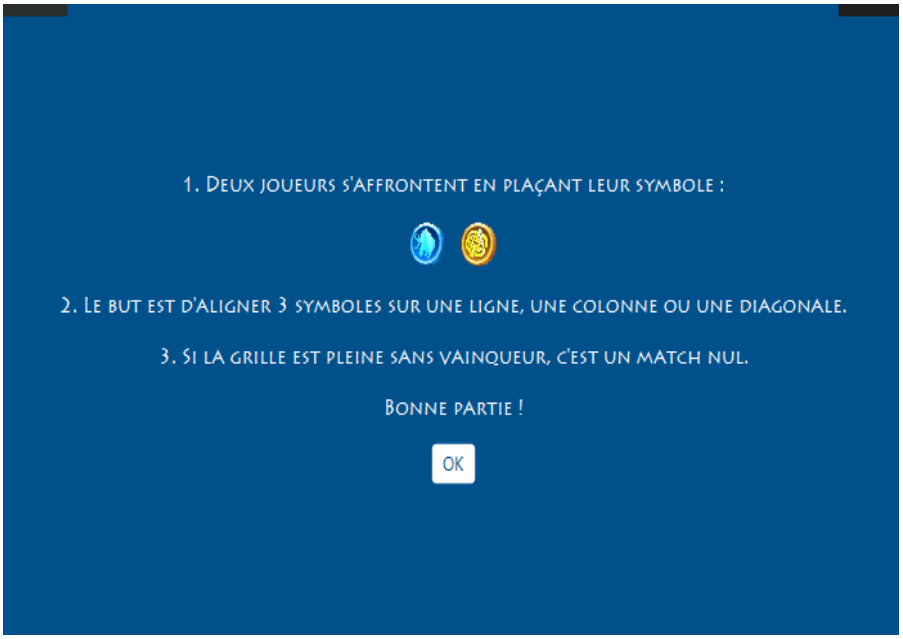
Accueil :

Choix du nom des joueurs et de l'ordre de jeu :

Page de jeu:

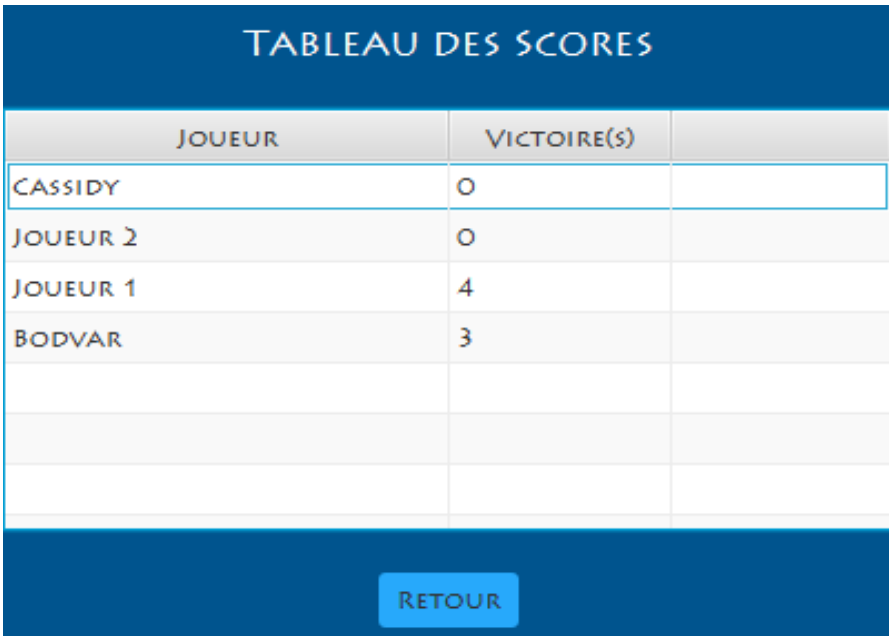


Page des règles:



Ecran de fin de partie:

Page du tableau des scores :



4. Difficultés

Exécution du .jar et Configuration Java :

Au départ, j'ai tenté d'exécuter mon fichier .jar avec Java 8, mais j'ai rapidement rencontré une erreur de type UnsupportedClassVersionError. Après vérification, j'ai compris que cette erreur est liée à la différence entre la version de Java utilisée pour compiler le projet (JDK 23.0.1) et celle utilisée pour l'exécuter (Java 8).

Suite à cette recherche, j'ai découvert qu'un .jar compilé avec Java 23 ne peut pas être exécuté avec une version de Java plus ancienne. Pour résoudre ce problème, voici la démarche que j'ai réalisé pour exécuter le .jar sur un autre ordinateur :

Étapes pour exécuter le .jar :

1. Installer Java JDK 23 ou supérieur :
   Télécharger et installer le JDK depuis le site officiel d'Oracle ou OpenJDK.
2. Si la version indiquée est inférieure à 23, il faut mettre à jour le PATH et la variable JAVA_HOME pour pointer vers le répertoire du JDK 23.

Exécuter le .jar :
Une fois la bonne version configurée, se rendre dans le dossier contenant le fichier .jar et lancer :
java -jar TicTacToe.jar

Cas où plusieurs versions de Java sont installées :
Si Java 8 est toujours prioritaire dans le système, il est possible de spécifier directement le chemin vers la bonne version :
"C:\Program Files\Java\jdk-23.0.1\bin\java.exe" -jar TicTacToe.jar

Problème de son avec JavaFX Media non supporté en version 23 :

➔ Solution : Utilisation de la librairie standard javax.sound.sampled pour lire les fichiers .wav.