

Documentation Technique

Projet Jeu de Curling sur Table

Contents

1	Présentation du Projet	2
2	Installation et Configuration	2
3	Organisation des Packages	5
4	Logique du Jeu	7

1 Présentation du Projet

Jeu de Curling sur Table

Description : Ce projet vise à développer un jeu de curling sur table combinant des éléments physiques et une interface numérique pour une expérience interactive et éducative.

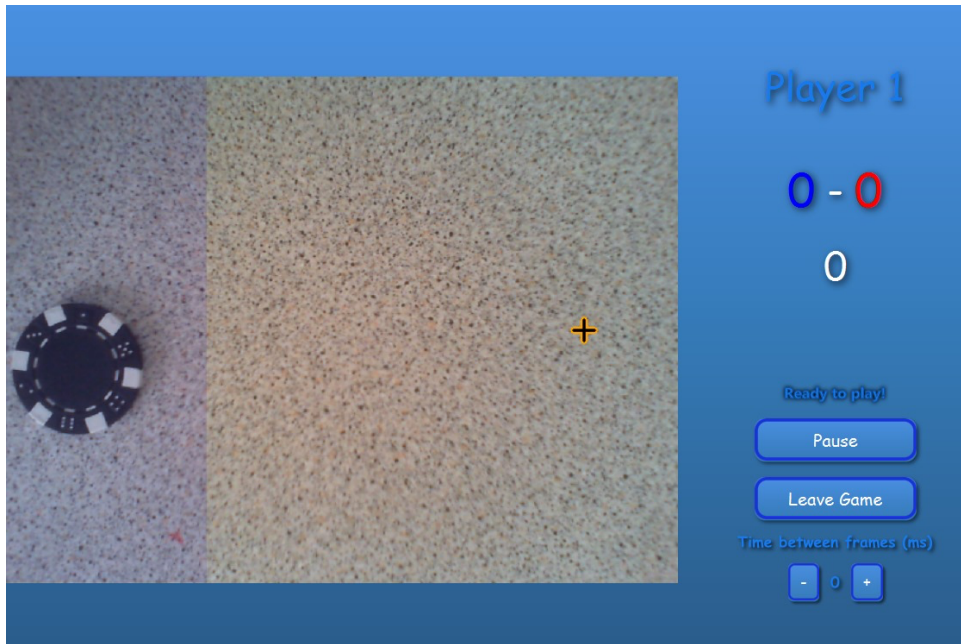


Figure 1: Exemple d'interface utilisateur du jeu.

2 Installation et Configuration

Pour Windows 10/11 64x :

1. **Installer le plug-in e(fx)clipse** depuis le marketplace d'Eclipse.
2. **Créer un projet JavaFX :** Ouvrez Eclipse et créez un nouveau projet JavaFX.
3. **Convertir en projet Maven :** Faites un clic droit sur le projet, puis choisissez Configure -> Convertir en projet Maven.
4. **Modifier le fichier pom.xml :** Ajoutez les dépendances suivantes :

```
<dependencies>
  <!-- OpenCV -->
  <dependency>
    <groupId>org.openpnp</groupId>
    <artifactId>opencv</artifactId>
    <version>4.5.1-2</version>
  </dependency>

  <!-- JavaFX -->
  <dependency>
```

```

        <groupId>org.openjfx</groupId>
        <artifactId>javafx-controls</artifactId>
        <version>20</version>
    </dependency>
    <dependency>
        <groupId>org.openjfx</groupId>
        <artifactId>javafx-fxml</artifactId>
        <version>20</version>
    </dependency>
</dependencies>

```

5. **Modifier le fichier module-info.java** : Ajoutez les lignes suivantes :

```

requires opencv;
requires java.desktop;

```

6. **Télécharger OpenCV** : Téléchargez la version 4.5.1 Windows 64x depuis le site officiel : <https://opencv.org/releases/>. Placez le dossier dans le dossier racine du projet.
7. **Configurer les arguments VM** : Dans Eclipse, faites un clic droit sur Main.java, sélectionnez Run As -> Run Configurations, et ajoutez :

```

-Djava.library.path=C:\opencv\opencv\build\java\x64

```

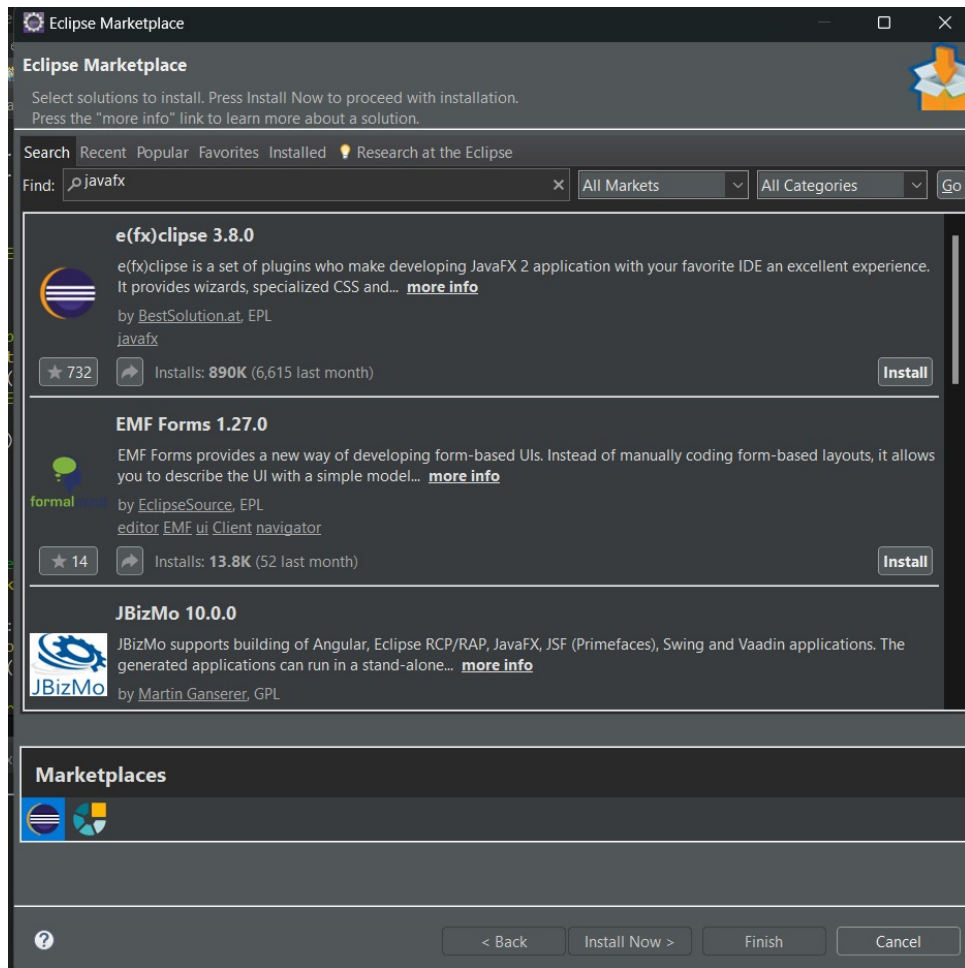


Figure 2: Installation du plug-in e(fx)clipse dans Eclipse.

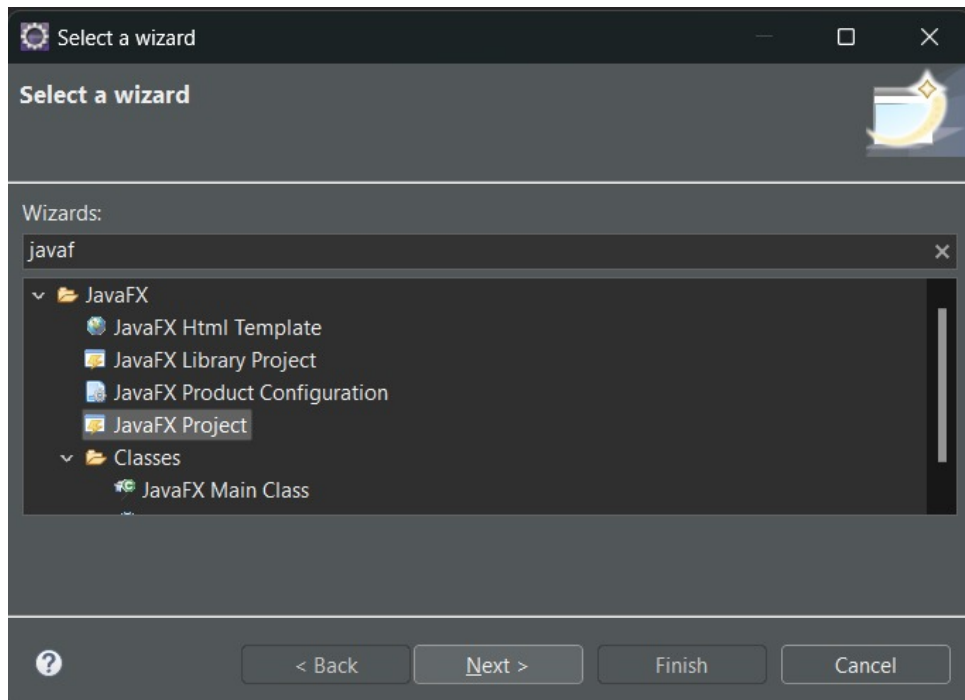


Figure 3: Création d'un projet JavaFX.

3 Organisation des Packages

Description des Packages :

- **ui** : Gère l'interface utilisateur (fichiers FXML et contrôleurs).
 - inGame-view.fxml : définissant l'interface graphique
 - inGameController : Contrôleur lié à inGame-view.fxml, gère les interactions utilisateur sur l'interface.
 - mainMenu-view.fxml : définissant l'interface graphique du menu principal.
 - mainMenuController : Contrôleur lié à mainMenu-view.fxml, gère les actions utilisateur dans le menu principal
- **gameLogic** : Implémente la logique de jeu.
 - gameLogic : Contient la logique principale du jeu (gestion des manches, alternance des joueurs, calcul des scores, etc.).
- **ImageProcessing** : Gère le traitement d'images.
 - ImageProcessing : Classe principale pour le traitement d'images, incluant :
 Capture des images via la webcam.
 Détection des jetons à l'aide de la Transformée de Hough.
 Calcul des distances des jetons par rapport à la cible.

- ImageProcessingResult: Classe permettant de stocker et traiter les résultats des algorithmes de traitement d'image (coordonnées des jetons, tailles, etc.).
- application : Contient la classe principale Main.
 - Launcher : Classe de lancement initiale, charge les configurations nécessaires avant le démarrage de l'application principale.
 - Main : Classe principale de l'application, initialise les packages et lance l'interface utilisateur.

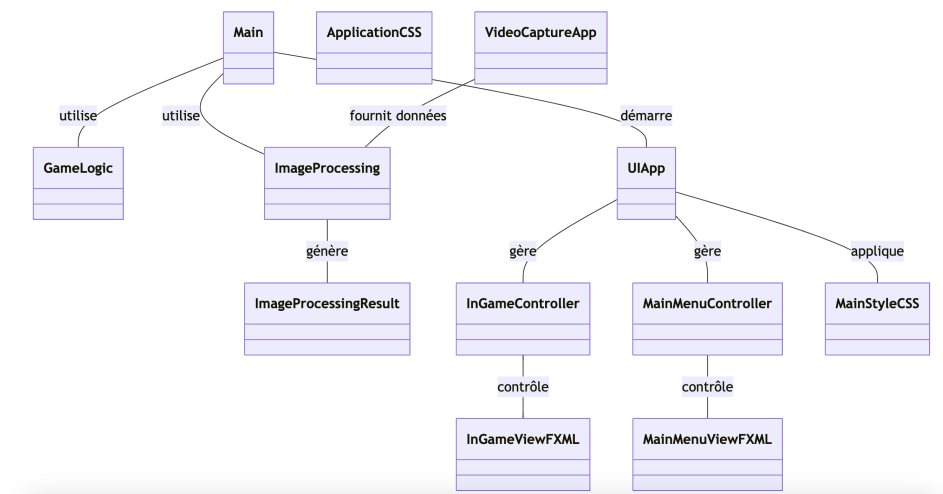


Figure 4: Diagramme UML simplifié des packages.

4 Logique du Jeu

- **Alternance des joueurs** : Le jeu alterne automatiquement entre les joueurs.
- **Utilisation d'un seul jeton** : Les joueurs se partagent un unique jeton.
- **Calcul des scores** : Basé sur la distance du jeton au centre de la cible.
- **Gestion des collisions** : Si un jeton en frappe un autre de son adversaire, ce dernier est retiré du jeu.
- **Durée de la partie** : Les scores sont mis à jour à la fin de chaque manche. La partie se termine lorsque l'un des joueur atteint 13 points .