



Document Technique V1.0

SAE 2.01

réalisé par le Olivencia Eliot, Veron Victor, Larrose Thomas

Groupe : 4B

fait le 14/06/2025

*Groupe 2A SAE Développement d'application . n Informatique [informatique] BUT Blagnac à Toulouse , 2024.
Français .tel-0XXXXXXX*

En remerciant tous nos professeurs qui nous ont pour la plupart apporté une quelconque aide que ce soit par les Travaux Dirigés, ainsi que les Travaux Pratiques et Cours en Amphithéâtre et de cette manière apporté du savoir afin de pouvoir réaliser cette Étude d'opportunité .

SOMMAIRE

Table des matières

Document Technique V1.0.....	3
1. Présentation de l'Application.....	3
1.1. Qui fait quoi ? (Rôles et Infos Clés).....	3
1.2. Comment ça marche ? (Architecture).....	3
1.3. Vue d'ensemble des fonctions (Diagramme de Cas d'Utilisation Global).....	4
2. Ce qui compose l'application (Les modules).....	5
2.1. Les "Modèles" (fr.tournois.model).....	5
2.2. Les "DAO" (Pour parler à la base de données - fr.tournois.dao).....	5
2.3. Les "Contrôleurs" (Ce qui gère l'écran - fr.tournois.ui.controller).....	5
2.4. La "Sécurité" (fr.tournois.security).....	6
2.5. Les "Utilitaires graphiques" (fr.tournois.ui.util).....	6
3. Les fonctionnalités qu'on a faites (Version 1).....	6
3.1. Gérer les Joueurs (ADMIN).....	6
3.2. Gérer les Équipes (ORGANISATEUR).....	7
3.3. Gérer les Jeux (ORGANISATEUR).....	8
3.4. Associer Joueurs et Équipes (ORGANISATEUR).....	9
4. Comment on a travaillé et ce qu'on a appris.....	10
5. Bilan et Prochaines étapes.....	10

Document Technique V1.0

1. Présentation de l'Application

Notre projet, "LiveTournois", est une application qui aide à organiser des tournois de jeux vidéo. Elle est conçue pour que différentes personnes (administrateurs, organisateurs...) puissent l'utiliser selon leurs tâches. Pour ce projet, notre équipe a développé la Version 1 (V1) de certaines parties importantes, en nous basant sur une première version (V0) que nos professeurs nous ont donnée.

1.1. Qui fait quoi ? (Rôles et Infos Clés)

L'application a plusieurs types d'utilisateurs et gère plusieurs "choses" (entités) :

- **Les rôles :**
 - **ADMIN** : C'est la personne qui a tous les droits, elle gère les utilisateurs du système et les joueurs.
 - **ORGANISATEUR** : Cette personne peut créer et gérer les tournois, les équipes, les jeux, et dire quel joueur va dans quelle équipe.
 - **JOUEUR** : C'est le participant au tournoi, dans une équipe.
 - **STAFF** : Aide l'organisateur pendant le tournoi.
- **Les informations gérées (entités)** : On gère des Utilisateurs, des Joueurs, des Equipes, des Jeux, des Tournois, des Staffs, des Affectations (pour le staff) et des Inscriptions (pour les équipes aux tournois).

1.2. Comment ça marche ? (Architecture)

Notre application est construite selon le modèle **MVC (Modèle-Vue-Contrôleur)**, ce qui est courant avec JavaFX.

- **Le Modèle** : C'est là où sont toutes les données et la logique de l'application (ce sont nos classes dans `fr.tournois.model.*`).
- **La Vue** : C'est ce que l'utilisateur voit, l'interface graphique (les fichiers `.fxml`).

- **Le Contrôleur** : Il fait le lien entre le Modèle et la Vue. C'est lui qui réagit quand l'utilisateur clique et qui met à jour l'affichage (nos classes dans fr.tournois.ui.controller.*).

Pour communiquer avec la base de données, on utilise des objets spéciaux appelés **DAO (Data Access Object)**. Ils savent comment enregistrer, lire, modifier ou supprimer des informations.

1.3. Vue d'ensemble des fonctions (Diagramme de Cas d'Utilisation Global)

Version 1 - Gestion des Équipes et Jeux

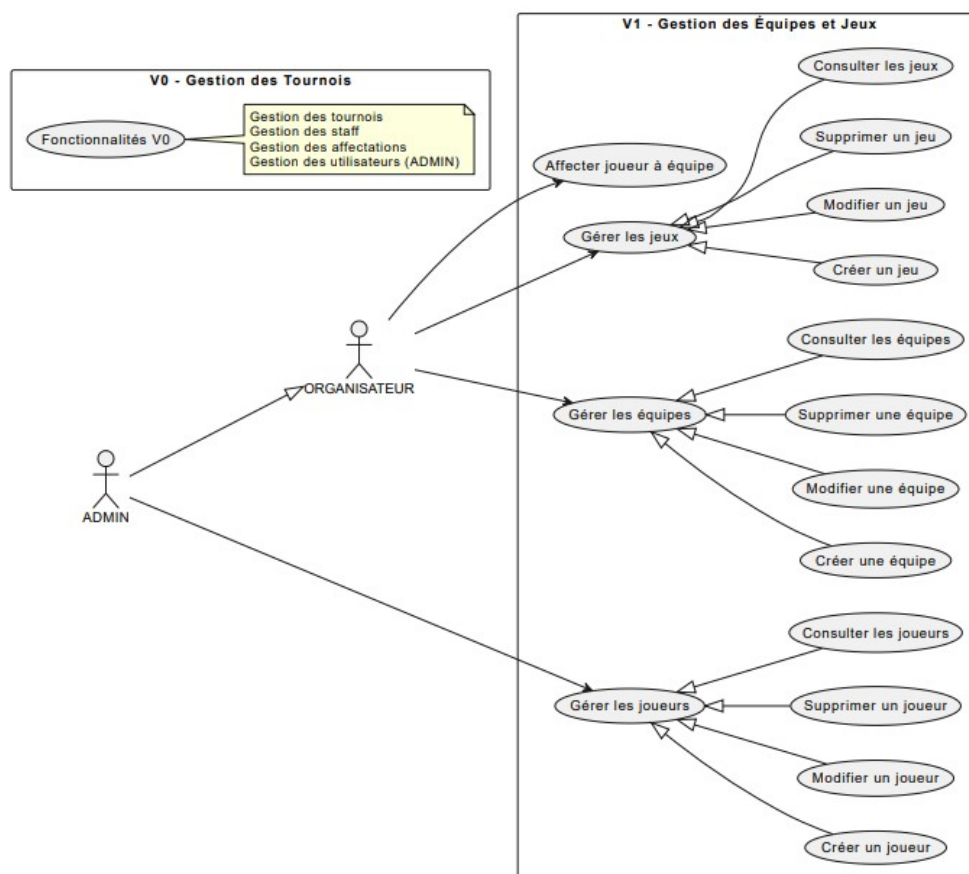


Diagramme de cas d'utilisation global de la V1 comprenant les fonctionnalités souhaitées

Ce schéma montre toutes les grandes fonctions de l'application et qui peut les utiliser. Il présente l'ensemble du projet, y compris les parties que les professeurs nous ont données (V0) et celles que nous avons faites (V1).

- **ADMIN** : Peut gérer les comptes des Utilisateurs et les Joueurs.

- **ORGANISATEUR** : Peut gérer les Tournois, les Jeux, les Équipes, et attribuer les Joueurs aux Équipes.
- **JOUEUR** : Est un participant aux Tournois.
- **STAFF** : Est affecté à des Tournois par l'Organisateur.

1.4. Conventions d'Interface

Nous avons appliqué des conventions pour nos interfaces utilisateur :

- **Organisation standard** : Les écrans de gestion (Joueurs, Équipes, Jeux) affichent une liste principale (TableView) et des champs/boutons pour les opérations courantes (ajouter, modifier, supprimer).
- **Fenêtres d'édition modales** : Les formulaires d'ajout/modification s'ouvrent dans de petites fenêtres séparées (dialogue modal) avec des boutons "Valider" et "Annuler".
- **Validation et retours utilisateur** : Les saisies sont systématiquement validées. En cas d'erreur, des messages clairs sont affichés via des alertes (AlertUtils).

2. Ce qui compose l'application (Les modules)

Ici, on explique les grandes parties du code.

2.1. Les "Modèles" (fr.tournois.model)

Ce dossier contient toutes les classes qui représentent les "choses" que l'application gère : par exemple, Joueur.java, Equipe.java, Jeu.java, Tournoi.java, etc. Elles décrivent ce qu'est un joueur (son nom, son pseudo...) et comment ces éléments sont liés.

2.2. Les "DAO" (Pour parler à la base de données - fr.tournois.dao)

Ces classes sont faites pour sauvegarder et récupérer les informations de la base de données. Pour notre partie V1, nous avons surtout travaillé avec :

- JoueurDAO.java : Gère tout ce qui concerne les joueurs en base de données.
- EquipeDAO.java : Gère les équipes dans la base de données.
- JeuDAO.java : Gère les jeux vidéo dans la base de données.
- *D'autres DAOs comme TournoiDAO, UtilisateurDAO, StaffDAO... ont été fournis par les professeurs pour les fonctions de la V0, et nous les utilisons si besoin.*

2.3. Les "Contrôleurs" (Ce qui gère l'écran - fr.tournois.ui.controller)

Ces classes font le lien entre ce que l'utilisateur voit et ce qui se passe dans le programme. **Les contrôleurs que notre groupe a développés (V1) :**

Auteur : Olivia Eliot, Veron Victor, Larrose Thomas

- GererJoueursController.java : L'écran où l'ADMIN gère les joueurs (liste, ajout, modification, suppression).
- EquipeController.java : L'écran principal pour voir et gérer les équipes.
- nouvelleEquipeController.java : Le petit écran pour ajouter ou modifier les détails d'une équipe.
- JeuManagementController.java : L'écran principal pour voir et gérer les jeux (liste, recherche, filtre).
- JeuEditorController.java : Le petit écran pour ajouter ou modifier les détails d'un jeu.
- GestionJoueurEquipeController.java : L'écran spécial pour dire quel joueur fait partie de quelle équipe.

Les contrôleurs fournis par les professeurs (V0) :

- UtilisateursManagementController.java, UtilisateurEditorController.java, UtilisateursViewerController.java (pour la gestion des utilisateurs).
- TournoisManagementController.java, TournoiEditorController.java, InscriptionEditorController.java, InscriptionIntegrationController.java, AffectationEditorController.java (pour la gestion des tournois, staffs et inscriptions).
- LoginController.java : Gère la connexion des utilisateurs.
- AppMainFrameController.java : Le contrôleur de la fenêtre principale de l'application, qui contient les menus.
- ADeveloperController.java : Un contrôleur "en attente" pour des fonctions futures.

2.4. La "Sécurité" (fr.tournois.security)

Ce module a été fourni par les professeurs. Il contient :

- PasswordHasher.java : Sert à sécuriser les mots de passe.
- SecurityContext.java : Garde en mémoire qui est connecté et ce qu'il a le droit de faire.

2.5. Les "Utilitaires graphiques" (fr.tournois.ui.util)

Ce module, aussi fourni, regroupe des petites fonctions pratiques pour l'interface :

- AlertUtils.java : Pour afficher facilement des messages d'alerte.
- DialogUtils.java : Pour bien positionner les boîtes de dialogue sur l'écran.

3. Les fonctionnalités qu'on a faites (Version 1)

Notre équipe s'est concentrée sur le développement de ces fonctionnalités pour la V1 du projet, en partant de la base (V0) fournie.

3.1. Gérer les Joueurs (ADMIN)

Qui a fait : Olivencia Eliot

Cette fonctionnalité permet à un ADMIN de centraliser et de manipuler les fiches des participants. L'interface offre une table interactive pour consulter l'ensemble des joueurs enregistrés. Un formulaire dédié permet d'ajouter, modifier ou supprimer des joueurs, en assurant l'intégrité des données.

- **Contrôleur utilisé :** GererJoueursController.java

Gestion des Joueurs

Gestion des Joueurs

ID:

Pseudo:

Nom:

Prénom:

Date de naissance:

Créer

Modifier

Supprimer

Effacer

ID	Pseudo	Nom	Prénom	Date de naissance
1	Wolf	Martin	Thomas	1999-03-15
2	Omega	Müller	Jan	2001-05-20
3	Flash	Garcia	Carlos	1998-09-10
4	Thunder	Smith	John	1996-12-05
5	Eagle	Rossi	Marco	1999-07-22

Rechercher:

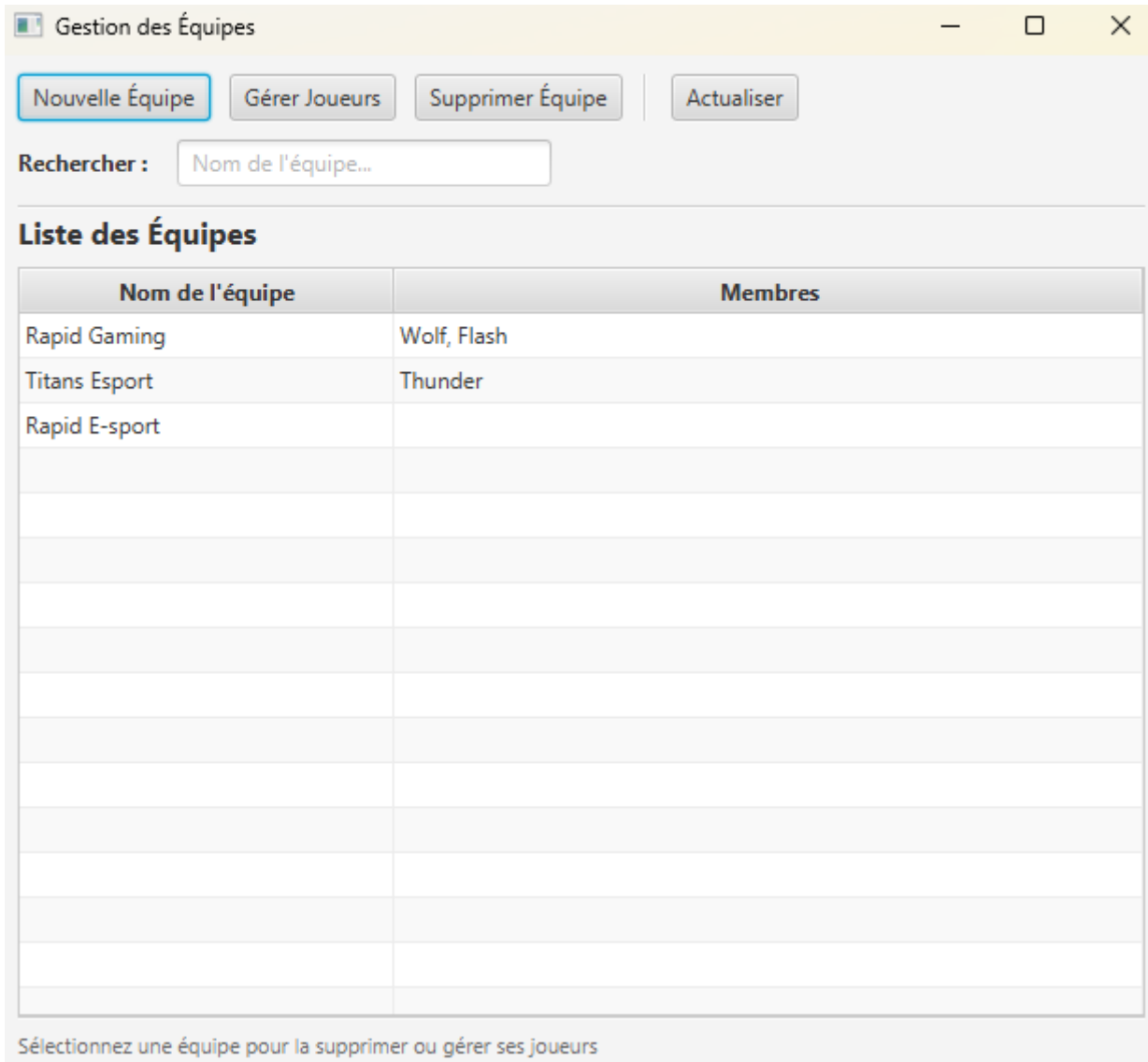
Un écran montrant la liste des joueurs et les champs pour en ajouter/modifier un.

3.2. Gérer les Équipes (ORGANISATEUR)

Qui a fait : Larrose Thomas

Cette fonctionnalité couvre la gestion complète des équipes pour les tournois. L'ORGANISATEUR peut consulter la liste des équipes via une vue tabulaire. Un dialogue modal est utilisé pour ajouter, modifier ou supprimer une équipe, permettant de renseigner tous ses détails (nom, tag, date de création, etc.).

- **Contrôleurs utilisés :** EquipeController.java, nouvelleEquipeController.java



Nom de l'équipe	Membres
Rapid Gaming	Wolf, Flash
Titans Esport	Thunder
Rapid E-sport	

Sélectionnez une équipe pour la supprimer ou gérer ses joueurs

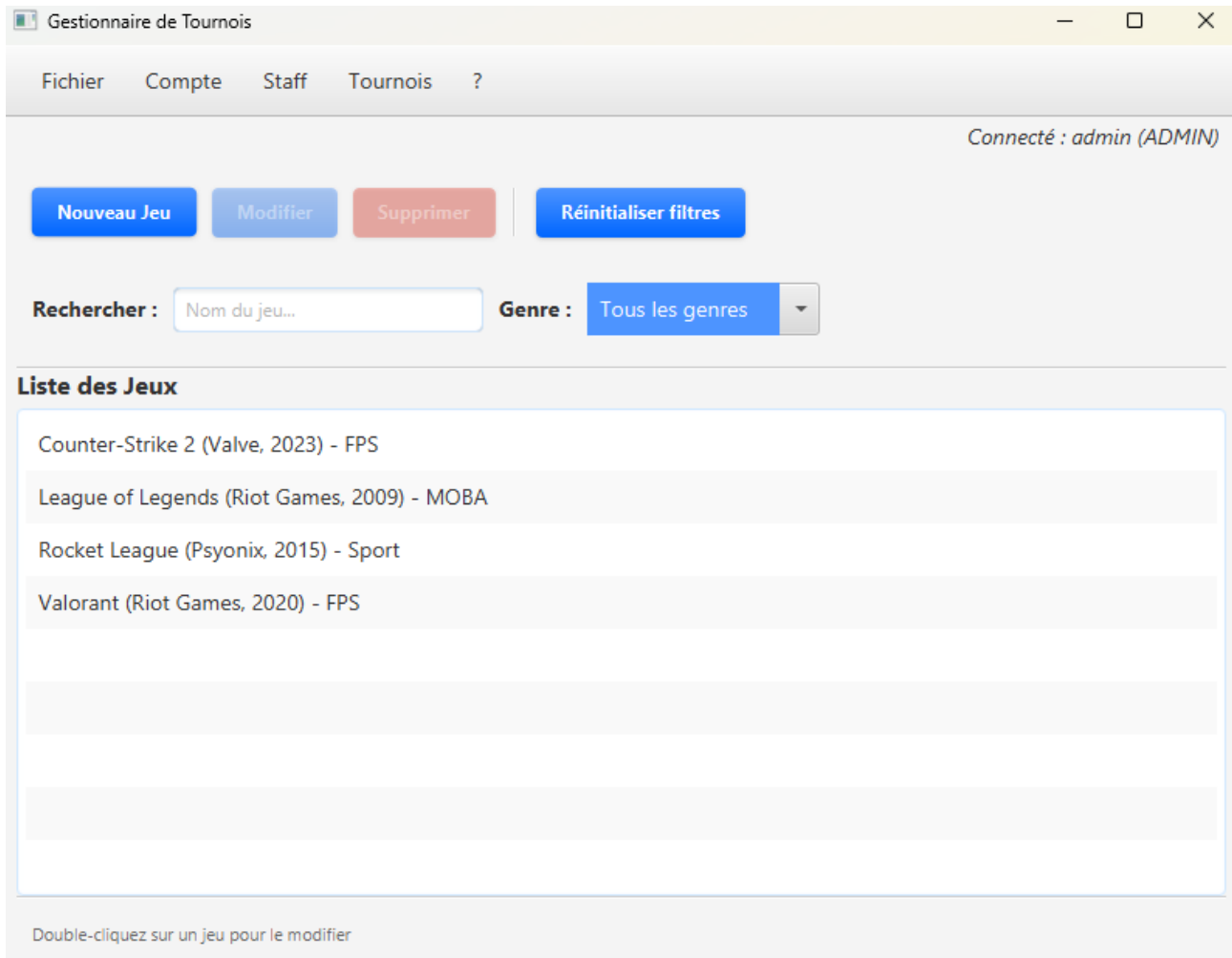
Un écran montrant la liste des équipes.

3.3. Gérer les Jeux (ORGANISATEUR)

Qui a fait : Veron Victor

Cette fonctionnalité permet à l'ORGANISATEUR de maintenir la liste des jeux vidéo supportés. L'interface présente une liste des jeux et inclut des options de recherche par nom et de filtrage par genre. Un formulaire d'édition via dialogue modal permet d'ajouter, modifier ou supprimer un jeu avec ses attributs (nom, éditeur, année de sortie, genre, description).

- **Contrôleurs utilisés :** JeuManagementController.java, JeuEditorController.java



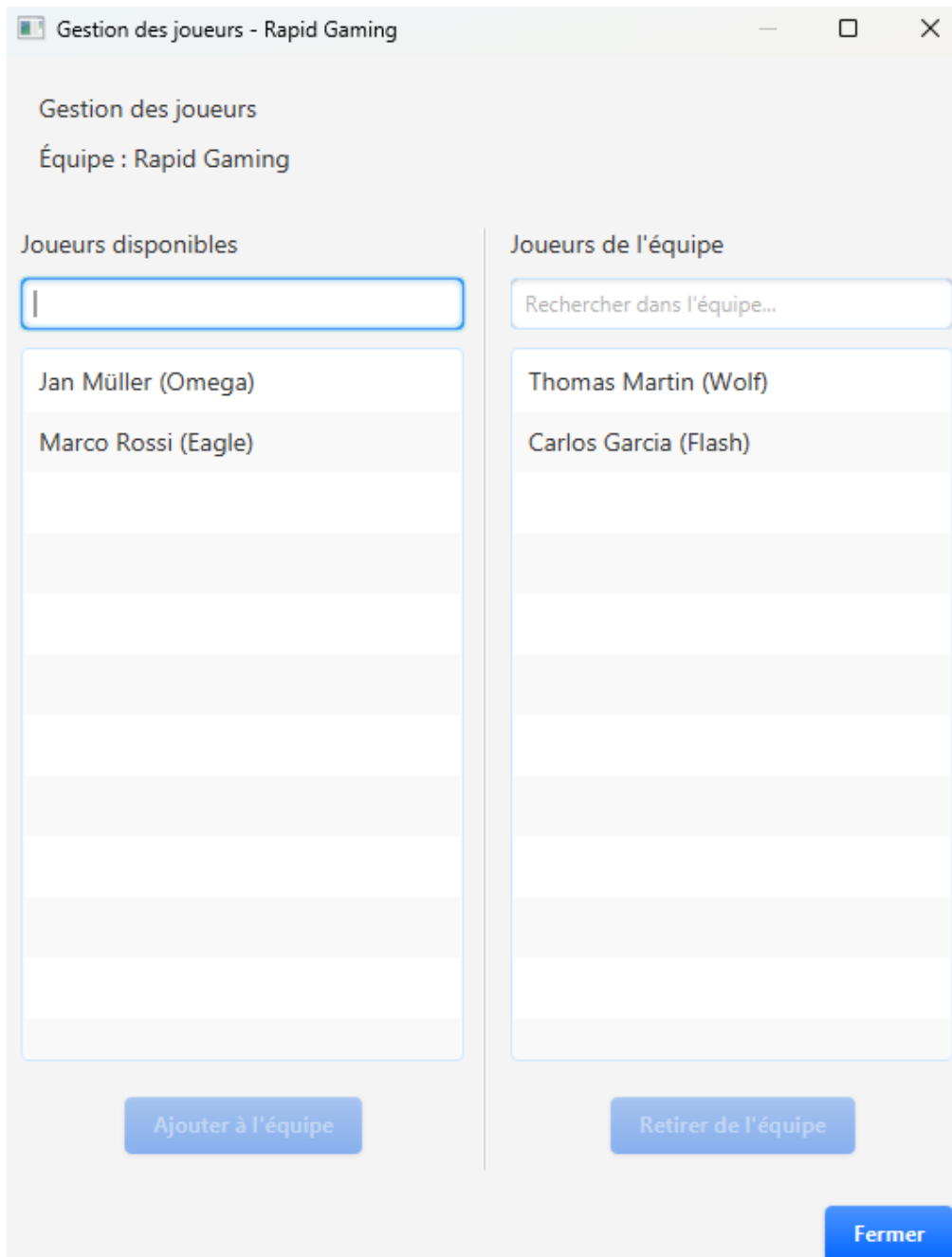
Un écran montrant la liste des jeux, avec les options de recherche ou filtre.

3.4. Associer Joueurs et Équipes (ORGANISATEUR)

Qui a fait : Thomas Larrose

Cette fonctionnalité est cruciale pour la préparation des tournois, car elle permet à l'ORGANISATEUR de former les équipes. L'interface est conçue avec deux listes distinctes (joueurs libres / joueurs de l'équipe sélectionnée), permettant de déplacer les joueurs d'une liste à l'autre pour une composition d'équipe flexible et intuitive.

- **Contrôleur utilisé :** GestionJoueurEquipeController.java



L'écran où l'on peut choisir des joueurs et les mettre dans une équipe (souvent avec deux listes côte à côte).

4. Comment on a travaillé et ce qu'on a appris

On a bossé à trois sur ces fonctionnalités de la V1. Voici comment on s'est partagé le travail et les choses importantes qu'on a apprises :

Fonctionnalité / Qui l'a

Apprentissages Clés (Ce qu'on a appris)

Module	fait	
Gestion des Joueurs (V1)	T. Larrose	DAO, Contrôleur (GererJoueursController), Validation (champs obligatoires), Interface Utilisateur (CRUD).
Gestion des Équipes (V1)	T. Larrose	DAO, Contrôleurs (EquipeController, nouvelleEquipeController), Validation des données, Interface Utilisateur (CRUD).
Gestion des Jeux (V1)	V. Veron	DAO, Contrôleurs (JeuManagementController, JeuEditorController), Validation (dates, textes), Interface Utilisateur (recherche/filtre).
Affectation Joueurs/Équipes (V1)	T. Larrose	Logique d'association entre entités, Contrôleur (GestionJoueurEquipeController), Conception d'une interface spécifique.

5. Bilan et Prochaines étapes

La Version 1 (V1) de "LiveTournois" a posé les bases importantes pour gérer les joueurs, les équipes, les jeux, et leurs liens. Ce projet a été super pour mettre en pratique ce qu'on a vu en cours : l'architecture MVC, comment utiliser une base de données avec les DAO, et créer des interfaces avec JavaFX.

Nous avons surtout rencontré des défis avec la gestion des interactions entre les différentes parties du programme et la validation des informations pour éviter les erreurs. On a aussi appris l'importance d'organiser proprement notre code et de bien travailler en équipe.

Un apprentissage clé a été la Validation des Données (AC3). Nous avons appris à vérifier la justesse et la complétude des informations saisies par l'utilisateur (champs obligatoires, formats corrects, cohérence des dates...). C'est essentiel pour éviter les erreurs, maintenir la qualité des données de l'application et offrir une bonne expérience à l'utilisateur grâce à des messages d'erreur clairs.

Pour les prochaines versions de l'application, on pourrait imaginer développer toutes les fonctions pour les tournois (inscription, gestion des matchs), améliorer la sécurité pour tous les types d'utilisateurs, et ajouter des fonctionnalités pour les joueurs eux-mêmes ou pour le staff pendant un tournoi.