

# Plan de test

## 1. Introduction

L'Objectif de ce plan de test sera de valider la partie fonctionnalité.

En termes de périmètre, la couverture s'étendra à la création, la gestion, l'exécution et le suivi des résultats des tournois.

Notre scope comprend des fonctionnalités essentielles à les générations de tournois :

- La création des équipes à partir de joueurs
- Les créations de tournois

Quant à notre out of scope, il comprend la fonctionnalité qui permet de charger des fichiers afin de générer des équipes ou de charger des équipes prêtes à concourir.

Dans le but de nous assurer que notre programme répond aux fonctionnalités attendues, nous allons mettre en place différents types de test :

- Test Unitaires : Dans le but de tester les parties importantes de lié à la logique de notre code
- Test d'intégration : Ici notre souhait est de nous assurer que l'ajout de nouvelles fonctionnalités ne viennent pas casser le code existant
- Tests Fonctionnels : Pour vérifier chaque fonctionnalité contre ses spécifications.
- etc.

## 2. Environnement de Test

Le projet étant dans une phase de prototype. Nous exécuterons les tests sur nos propre machine. Il n'est pas nécessaire d'automatiser les tests aujourd'hui ou de prévoir des environnements de déploiement

## 3. Analyse de risque

# Plan de test

En nous basant sur le scope défini, une première analyse de risque.

## 1. Analyse de risque du générateur d'équipe:

### 1.1. Risque de mauvaise répartition des joueurs dans les équipes:

Description : Si l'algorithme de génération d'équipes ne répartit pas équitablement les joueurs selon leurs compétences ou si la répartition est déséquilibrée, cela peut compromettre l'équité du tournoi.

Impact : Cela pourrait entraîner un désavantage compétitif et affecter négativement l'expérience des participants.

Probabilité : Moyenne

Criticité : Élevée

### 1.2. Risque d'erreur dans l'ajout de nouveaux joueurs aux équipes:

Description : Des bugs dans la fonction `addPlayerToTeam` pourraient permettre l'ajout de joueurs au-delà de la limite prévue ou échouer à les ajouter correctement.

Impact : Cela peut causer des déséquilibres dans le nombre de joueurs par équipe et affecter le déroulement du tournoi.

Probabilité : Faible

Criticité : Moyenne

### 1.3. Risque que les équipes ne soient pas générées:

Description : Un échec dans la méthode `generateTeams` due à une erreur de logique ou un problème de données d'entrée pourrait empêcher la création d'équipes.

Impact : Empêcherait le début du tournoi et pourrait causer des retards importants.

Probabilité : Moyenne, particulièrement si de nouveaux scénarios de données n'ont pas été testés.

Criticité : Élevée, car cela stopperai..

### 1.4. Risque lié à la manipulation des données des joueurs:

Description : Des erreurs dans la gestion des données des joueurs pourraient mener à la perte de données importantes ou à leur corruption.

Impact : Perte de confiance des utilisateurs et potentielles complications juridiques.

# Plan de test

Probabilité : Faible, avec un bon système de gestion de base de données et de sauvegardes.

Criticité : Élevée, car cela affecte l'intégrité des données utilisateur.

## 1.5. Risque de performance sous charge élevée:

Description : Le système pourrait ne pas être capable de gérer une charge élevée lors de la génération d'équipes si de nombreux tournois sont créés simultanément.

Impact : Temps de réponse lent et potentiellement des défaillances du service.

Probabilité : Moyen, dépend de la capacité de l'infrastructure actuelle.

Criticité : Élevée, en particulier lors de grands événements.

## 2. Analyse de risque du générateur du tournoi:

### 2.1. Génération incorrecte des poules :

Description : Des erreurs dans la répartition des équipes peuvent survenir lors de l'exécution de la fonction `generatePoules`.

Impact : Cela peut entraîner un déséquilibre dans les compétitions et remettre en question l'équité du tournoi.

Niveau de risque : Moyen

Criticité : Élevée

### 2.2. Simulation imprécise des matches des poules :

Description : La méthode `simulatePoulesMatches` peut fournir des résultats de match non représentatifs des capacités réelles des équipes.

Impact : Cela pourrait conduire à l'avancement inapproprié d'équipes moins qualifiées, nuisant à l'intégrité du tournoi.

Niveau de risque : Moyen

Criticité : Moyenne

### 2.3. Erreurs dans la génération des étapes finales :

Description : Des bugs dans `generateFinalStages` peuvent causer des erreurs dans la création des matches des phases finales du tournoi.

Impact : Une mauvaise configuration des étapes finales peut diminuer la qualité du tournoi et l'expérience des participants.

Niveau de risque : Moyen

# Plan de test

Criticité : Élevée

## 2.4. Échec de la génération complète du tournoi :

Description : Si `generateTournament` échoue, aucun tournoi complet ne peut être produit.

Impact : Cela pourrait entraîner l'annulation ou des retards significatifs dans le tournoi, affectant la réputation et la satisfaction des utilisateurs.

Niveau de risque : Moyen

Criticité : Très élevée

## 2.5. Élimination incorrecte des équipes :

Description : La fonction `eliminateTeam` pourrait ne pas fonctionner comme prévu, retirant incorrectement une équipe ou affectant les données des autres équipes.

Impact : Erreur dans l'élimination peut entraîner des confusions et des erreurs dans l'organisation des matchs suivants.

Niveau de risque : Faible

Criticité : Moyenne

## 2.6. Acquisition erronée des équipes qualifiées :

Description : La méthode `getQualifiedTeams` pourrait échouer à identifier correctement les équipes qualifiées après les matches de poule.

Impact : Qualification incorrecte des équipes, entraînant des problèmes dans la progression du tournoi.

Niveau de risque : Moyen

Criticité : Élevée