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

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

* **Risque de non-fonctionnement des fonctionnalités de base :** Si les fonctionnalités de base telles que la création des équipes ou la génération des tournois ne fonctionnent pas correctement, cela compromettrait l'ensemble de l'application et entraînerait une mauvaise expérience utilisateur.
* **Risque de mauvaise gestion des données :** Si le système ne gère pas correctement les données des équipes et des tournois, cela pourrait entraîner des résultats incorrects ou incohérents lors de la simulation des matchs ou du classement des équipes.
* **Risque de performances insuffisantes :** Si le système prend trop de temps à générer des tournois ou à simuler des matchs, cela pourrait entraîner une mauvaise expérience utilisateur et limiter l'évolutivité de l'application.
* **Risque de non-intégration des nouvelles fonctionnalités :** Si de nouvelles fonctionnalités ne s'intègrent pas correctement avec le code existant, cela pourrait entraîner des bugs ou des comportements inattendus dans l'application.