Cahier des Charges pour un Programme Python d'Organisation de Tournoi de Beach-Volley en Mélé-Démélé

1. Introduction

Ce cahier des charges décrit les exigences pour le développement d'un programme Python permettant d'organiser un tournoi de beach-volley par équipe de deux joueurs en mode "mêlé-démêlé". Le programme doit permettre de définir le nombre de joueurs et le nombre de matchs par joueur, puis générer une planification des matchs en respectant ces paramètres.

Fonctionnement du mêlé-démêlé :

- Les matchs se font en **2 contre 2** (format classique du beach volley).
- À chaque match, les **équipes sont tirées au sort** différemment.
- Les joueurs changent **à la fois de partenaire et d'adversaires** à chaque match.
- L'objectif est que **chaque joueur joue le même nombre de matchs**, et si possible, avec un maximum de personnes différentes.

Exemple de planification des matchs avec 10 joueurs (J1 à J10) et 4 matchs par joueur :

- Chaque match engage **4 joueurs**.
- Si chaque joueur joue **4 matchs**, on a un total de **10 joueurs × 4 matchs = 40 participations**.
- Chaque match comptant pour **4 participations**, il faut donc organiser 10 matchs.

Match	Équipe 1	Équipe 2
1	J1 & J2	J3 & J4
2	J5 & J6	J7 & J8
3	J9 & J10	J1 & J3
4	J2 & J4	J5 & J7
5	J6 & J8	J9 & J1
6	J2 & J5	J3 & J10
7	J4 & J6	J7 & J9
8	J8 & J1	J2 & J10
9	J3 & J5	J4 & J9
10	J6 & J10	J7 & J8

2. Objectifs

- Développer un programme **interactif et convivial**.
- Permettre à l'utilisateur de **spécifier le nombre de joueurs et le nombre de matchs par joueur.**
- Générer une **répartition équitable des joueurs dans les équipes** pour chaque match.
- **S'assurer que chaque joueur participe au nombre de matchs spécifié** et avec une variété de partenaires et d'adversaires.

3. Fonctionnalités

3.1 Interface Utilisateur

- **Entrée Utilisateur :**
  - Nombre total de joueurs (8 à 12 joueurs).
  - Nombre de matchs par joueur (4 à 8 matchs).
- **Sortie :**
  - Liste des matchs avec les équipes pour chaque match.

3.2 Génération des Matchs

- **Création d'Équipes :**
  - Former des équipes de 2 joueurs.
  - S'assurer que chaque joueur joue avec différents coéquipiers et contre différents adversaires à chaque match.
- **Planification :**
  - Générer un planning où chaque joueur participe exactement au nombre de matchs spécifié.

3.3 Validation et Vérification

- Vérifier que chaque joueur joue le nombre de matchs requis.
- S'assurer que les équipes sont variées pour maximiser les combinaisons de partenaires et adversaires.

4. Contraintes Techniques

- **Langage de Programmation :** Python 3.x
- **Librairies :**
  - Utilisation de bibliothèques standard (ex: itertools, random).
  - Possibilité d'utiliser des bibliothèques supplémentaires si justifié (ex: pandas pour la manipulation des données).
- **Compatibilité :**
  - Le programme doit être compatible avec les systèmes d'exploitation principaux (Windows, macOS, Linux).

5. Scénarios d'Utilisation

5.1 Cas de Base

1. L'utilisateur lance le programme.

2. Le programme demande le nombre de joueurs.
3. Le programme demande le nombre de matchs par joueur.
4. Le programme génère et affiche la planification des matchs.

5.2 Cas d'Erreur

- **Nombre de Joueurs ou de Matchs Invalides :**
  - Si le nombre de joueurs ou de matchs est invalide (ex: nombre impair de joueurs), le programme doit afficher un message d'erreur et redemander l'entrée.
- **Incompatibilité :**
  - Si le nombre de joueurs et de matchs ne permet pas une répartition équitable, le programme doit notifier l'utilisateur et suggérer des ajustements.

6. Exigences de Performance

- Le programme doit être capable de générer la planification en moins de 3 secondes pour un cas avec 10 joueurs.
- Utilisation de la mémoire optimisée pour traiter efficacement jusqu'à 10 joueurs.

7. Tests et Validation

- **Tests Unitaires :**
  - Tester les fonctions de base (ex: génération d'équipes, vérification des matchs).
- **Tests Fonctionnels :**
  - Vérifier le bon fonctionnement de l'interface utilisateur.
  - Valider la génération correcte et équitable des matchs pour différents nombres de joueurs et de matchs.