



# Jeu du Yahtzee (Yam's)

---

PROJET SOCKET EN PYTHON – SEMESTRE 7

# Cahier des Charges

## 1. Contexte du projet

Le projet consiste à développer une application de jeu de dés, le Yahtzee, utilisant des sockets en Python. Il s'agit d'un projet académique dans le cadre du M1 R&T / 2A STRI. L'application doit permettre à plusieurs joueurs de participer à des parties en ligne, avec des fonctionnalités progressives à travers trois versions : jeu simple, multi-parties et chat intégré.

## 2. Objectifs

L'objectif principal est de concevoir et développer une application Yahtzee fonctionnelle en trois versions :

- **Version 1 (Yam's simple)** : Une partie unique avec gestion des lancers de dés et des scores.
- **Version 2 (Yam's multi-parties)** : Possibilité de jouer à plusieurs parties simultanément.
- **Version 3 (Chat)** : Intégration d'un système de messagerie entre les joueurs.

## 3. Fonctionnalités requises

### 3.1. Version 1 : Yam's simple

- Serveur central pour gérer les parties.
- Lancement et relancement des dés.
- Calcul des points pour chaque joueur.
- Affichage du tableau des scores à la fin de la partie.

### 3.2. Version 2 : Yam's multi-parties

- Gestion de plusieurs parties simultanées.
- Possibilité pour les joueurs de rejoindre une partie non démarrée.
- Élimination d'un joueur s'il quitte une partie en cours.

### 3.3. Version 3 : Chat

- Système de discussion en temps réel pour les joueurs d'une même partie.
- Intégration du chat via le serveur.

## 4. Contraintes techniques

- **Langage de programmation** : Python.
- **Bibliothèques autorisées** : Utilisation des bibliothèques standard Python pour la gestion des sockets et des threads.

- **Interdiction** : Reprise de code existant ou de solutions d'IA sans citation claire. Aucun plagiat n'est toléré.
- **Gestion réseau** : Utilisation de sockets TCP pour la communication entre le serveur et les clients.
- **Architecture client-serveur** : Un serveur central gère l'ensemble des parties et des connexions client.

## 5. Livrables

Les livrables attendus à chaque échéance incluent :

- **Rapport de conception** : Diagrammes UML (cas d'utilisation, séquence, classes) et explication détaillée des choix techniques.
- **Code source** : Fichiers Python commentés.
- **Vidéo de démonstration** : Courte vidéo montrant le fonctionnement de l'application.

### 5.1. Échéances pour alternants

- **Version 1** : 22/11/2024.
- **Version 2** : 10/01/2025.
- **Version 3** : 24/01/2025.

## 6. Critères d'évaluation

L'évaluation sera basée sur :

- La qualité technique du code (lisibilité, modularité, gestion des erreurs).
- Le respect des délais de livraison.
- L'ergonomie et l'aspect général de l'application.
- La qualité de la documentation et de la conception UML.
- La conformité aux fonctionnalités demandées.