

# Documentation : Course de Chevaux

## 1. Documentation Fonctionnelle

### Présentation

Ce programme est un jeu de course de chevaux en mode console (CLI) où deux joueurs s'affrontent en temps réel. Le but est d'être le premier à franchir la ligne d'arrivée en appuyant de manière répétée sur une touche spécifique du clavier.

### Règles du Jeu

- **Nombre de joueurs** : Le jeu est conçu exclusivement pour 2 joueurs.
- **Objectif** : Atteindre la distance définie par le parcours choisi avant son adversaire.
- **Contrôles** : Chaque joueur définit son pseudonyme et une touche unique de l'alphabet (minuscule) pour faire avancer son cheval.

### Parcours disponibles

L'utilisateur peut choisir parmi quatre types de distances :

- **Test** : 10 mètres.
- **Plaine** : 100 mètres.
- **L'Hippodrome Olympique** : 150 mètres.
- **Route Arc-en-ciel** : 200 mètres.

### Déroulement d'une partie

1. **Configuration** : Les joueurs saisissent leurs noms et leurs touches de commande respectives.
2. **Choix de la map** : Sélection de la distance de la course.
3. **Compte à rebours** : Un décompte de 3 secondes lance le départ.
4. **Course** : Les joueurs appuient sur leur touche. La console affiche la progression en mètres pour chaque joueur avec une couleur distincte (bleu pour le joueur 1, rouge pour le joueur 2).
5. **Résultats** : Le vainqueur et le perdant sont affichés avec leur temps final en secondes.

## 2. Documentation Technique

### Architecture du Code

Le programme utilise la programmation orientée objet (POO) et le **multithreading** pour gérer les entrées clavier simultanées des deux joueurs.

#### La Classe Player

Héritant de `threading.Thread`, cette classe gère l'instance de chaque coureur :

- **Attributs** : position, name, commande, parcours, time, color, et design (fixé à l'emoji 🐘).
- **Méthode run()** : Boucle principale du thread qui utilise la bibliothèque `keyboard` pour détecter l'appui sur la touche assignée. À chaque appui correct, la position augmente de 1 jusqu'à atteindre la distance du parcours.

### Dépendances et Environnement

Le projet a été développé dans un environnement virtuel Python 3.12. Il nécessite les bibliothèques suivantes :

- `time`, `sys`, `os` : Gestion du temps, du système et de la console.
- `keyboard` : Capture des événements clavier en temps réel.
- `colorama` : Gestion des couleurs et de la mise en forme dans le terminal.
- `threading` : Gestion de l'exécution simultanée des joueurs.

### Fonctions Clés

- **main()** : Point d'entrée gérant le menu principal (Jouer, Règles, Quitter).
- **play()** : Orchestre l'initialisation des objets `Player`, démarre les threads (`start()`) et attend leur fin (`join()`) avant de traiter les scores.
- **names()** / **commande()** : Fonctions de saisie avec vérification des caractères autorisés.
- **check\_pseudonymes\_mm()** / **check\_commandes\_players\_2\_mm()** : Fonctions de sécurité pour s'assurer que les deux joueurs n'utilisent pas le même nom ou la même touche.