

# Projet Cambiecolo | Rendu du 17/12/2021

---

Groupe projet : Tristan Devin et Antoine Piron 3TC2

## Conception générale

Pour la réalisation de ce projet nous avons décidé de partir sur un système implémentant un programme **serveur** et un programme **client**.

Le serveur va correspondre au processus principal et se chargera de déclencher une partie en lançant un processus **jeu**.

Le client va lui correspondre au processus joueur qui va interragir avec le serveur.

## Déroulement d'une interaction client-serveur

Au début de multiples clients se connectent à notre serveur. Ces derniers se voit proposés différents ID de parties et choisissent de se connecter à l'une d'entre-elle. Côté serveur, chaque client sera identifié par un unique ID également afin de pouvoir interragir facilement avec lui. On pourra imaginer par exemple un dictionnaire avec pour clé les IDs clients (uniques et généré automatiquement lors d'une connection)

Une fois connecté le client est en attente du début de partie qui peut-être déclenché de 2 manières différentes :

- Un client demande un début de partie, tous les joueurs reçoivent une demande de confirmation pour lancer celle- ci et si tous le monde confirme la partie démarre.
- Au bout d'un certains *time out* et d'un nombre minimal de joueur (3) une proposition automatique de début de partie est envoyée.

La partie se déroule alors comme prévu dans les règles. Une fois la partie finie on demande à chaque participant s'il veut rejouer. Si oui on le renvoie dans la selection des parties, sinon on ferme sa connection (on pourra éventuellement lui renvoyer ses statistiques : nombres de parties, nombres de victoires).

## Implémentations des différents éléments qui permettent le bon déroulement de la partie

Tout d'abord nous allons profiter de l'aspect **POO** de python.

Effectivement nous allons pouvoir créer une classe qui gère la main de chaque joueur avec le type et le nombre de carte. On pourra l'imaginer de la sorte :

```
import enum

#On déclare d'abord une enum qui permet d'associer un type de carte à un
entier
class cardType(enum.Enum):
    airplane = 1
    car = 2
    train = 3
    bike = 4
    shoes = 5
```

```
class Hand:
    #De base on a une main vide
    myHand = [0,0,0,0,0]

    #Un constructeur de base qui affecte une main à un joueur
    #La distribution et la composition des mains seront géré par le
    processus jeu
    def __init__(self, hand):
        self.myHand = hand

    #Un surcharge de la méthode __str__ pour afficher la main du joueur
    def __str__(self):
        outstr = ""
        for i in range(self.myHand):
            outStr += "Carte %s : %s |"%(i, Animal(self.myHand[i]).name)
        return(outstr)
```

Il s'agit simplement d'un exemple montrant l'enum pour les cartes et comment la main serait affichée dans la console.

De plus l'imposition de l'utilisation de la *shared memory* pour les offres courantes va nécessiter l'utilisation de **mutex** pour gérer les accès concurrent des différents processus joueurs.

## Plan d'implémentation

L'implémentation de notre programme va se dérouler en plusieurs étapes :

1. Création d'un programme client et d'un programme serveur.
2. Création des assignations auto d'identifiants pour chaque connexion et rangement dans un dictionnaire.
3. Création de "l'enveloppe" du processus jeux → il va simplement s'agir de lancer un processus qui ne communique qu'avec un certains nombre restreint de clients assignés à cette partie.
4. Création de la couche de communication inter-processus. Il s'agit de l'interaction entre les joueurs mais également entre les joueurs et le serveur.
5. Finalement on ajoutera les règles du Cambiecolo pour pouvoir jouer une partie !