

Compte rendu

⇒ Cadre du projet

But du Jeu

Le but est de se débarrasser de tous ses dominos ou d'empêcher l'adversaire de jouer.

Règles de Base Mise en place

Les dominos sont placés face cachée et mélangés. Chaque joueur pioche un nombre déterminé de dominos :

À 2 joueurs: 7 dominos chacun.

À 3 ou 4 joueurs: 5 dominos chacun.

Les dominos restants forment la "pioche", qui est mise de côté face cachée.

Début du jeu

Le joueur qui commence est déterminé aléatoirement et pose le premier domino joué.

Fonctionnalitée

Un mode Joueur contre Joueur(s) et un mode Joueur(s) contre Bot(s) est disponible.

L'affichage du jeu est en console.

⚠ Pour que le code marche ⚠

Lancer dans une console main.py et la première fois inscrire: `pip install -r requirements.txt`

⇒ Structure de données:

Le projet est entièrement codé en Programmation Orientée Objet (POO) python.

Méthode / class importé de bibliothèque utilisée

- La méthode `randint` importée de `random` cela nous sert à choisir un nom pour un bot aléatoirement depuis une liste.
- La méthode `deque` importée de `collections` que nous utilisons pour comme file représentant notre chaîne de dominos.
- La class `Console` importée de `rich.console` (`rich` est une bibliothèque à installer) qui permet de mettre de la couleur dans la console

- La méthode `is_pressed` importée de `Keyboard` (`Keyboard` est une bibliothèque à installer) sert à obtenir l'input utilisateur quand il veut se déplacer dans un menu.
- La méthode `sleep` importée depuis la bibliothèque `Time` sert à mettre un temps d'arrêt dans l'affichage notamment quand les bots jouent.
- La méthode `system` importée depuis `os` permet de clear la console en fonction du système d'exploitation (Linux ou Windows)
- La méthode `shutil` qui sert à récupérer la largeur de la console

Fonctionnement global du programme

- Le plateau est une structure de données de type `file`, permettant d'ajouter un domino à droite ou à gauche en fonction des dominos déjà posés et du domino que le joueur souhaite jouer depuis son deck.
- Le deck des participants est une liste que l'on append quand on pioche et que l'on pop quand on joue.
- Les participants sont stockés dans une liste que l'on fait bouclé grâce du tour (un `int`) auquel on joue ce qui permet de faire jouer un participant un par un.

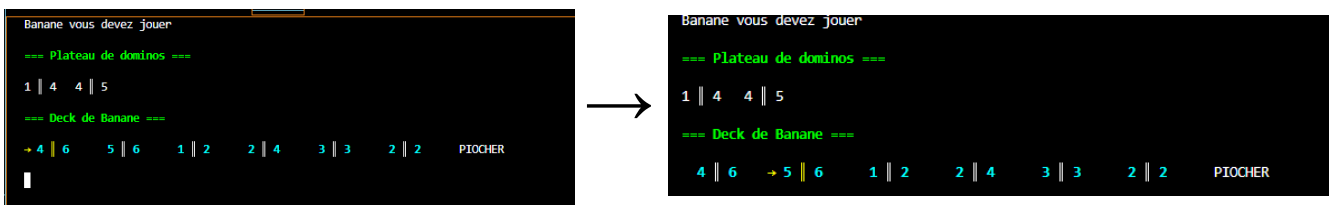
Pour plus détail consulter le fichier `html` fournie avec ce fichier.

⇒ Découpage du code:

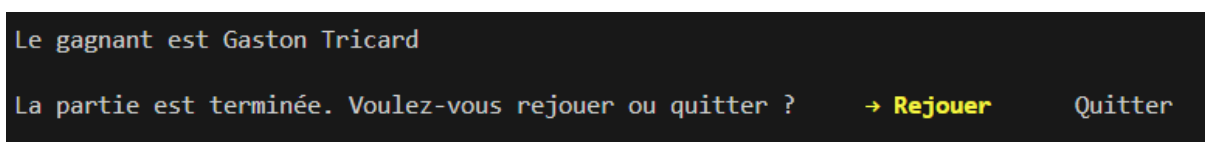
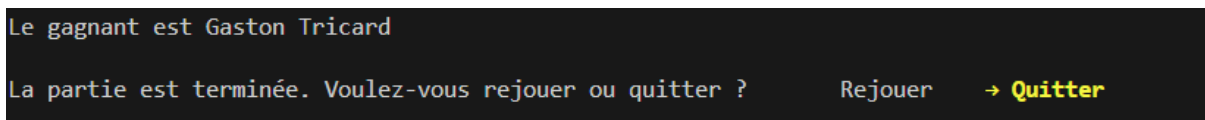
Voir le `HTML` de `pdoc` fourni avec le compte rendu et le code

⇒ Test:

Ici on remarque après avoir appuyé sur flèche droite le curseur s'est bien déplacé (`plateau_et_deck()` et `input()` ✓)



On voit bien que le menu de fin marche bien (`plateau_et_deck()` et `fin()` ✓)



(Si rejouée est choisie) ↓

```
Bienvenue dans un jeu de domino.  
Combien de joueurs veulent participer ? (1-4)  
>>> 
```

↓ (Si Quitter est choisie)

```
Merci d'avoir joué à notre jeu !!!  
PS C:\Users\tchab\Desktop\Programatio  
PS C:\Users\tchab\Desktop\Programatio
```

On voit que le bot a joué un domino correct (jouerBot() ✓)

```
=== Plateau de dominos ===  
5 | 6 6 | 1  
  
=== Deck de Jouer Test ===  
0 | 3 0 | 6 2 | 2 → 1 | 1 2 | 4 1 | 5 PIOCHER
```



```
Gaston Tricard a joué (1, 3)  
  
=== Plateau de dominos ===  
5 || 6 6 || 1 1 || 1 1 || 3
```

Ici on voit que on peut placer le domino à droite et à gauche (plateau_et_deck() ✓)

```
=== Plateau de dominos ===  
4 || 6 6 || 1 1 || 5  
  
Où voulez-vous placer ce domino: 4 || 5 ? → À droite À gauche Annuler
```

Ici on voit bien que le plateau a changé quand le joueur a joué (jouer() ✓)

```
Banane vous devez jouer  
  
=== Plateau de dominos ===  
6 | 4 4 | 4 4 | 2 2 | 2 2 | 6 6 | 1 1 | 3 3 | 5 5 | 0 0 | 2 2 | 1  
  
=== Deck de Banane ===  
→ 5 | 6 3 | 3 0 | 1 PIOCHER
```



```
=== Plateau de dominos ===  
5 | 6 6 | 4 4 | 4 4 | 2 2 | 2 2 | 6 6 | 1 1 | 3 3 | 5 5 | 0 0 | 2 2 | 1
```

Ici on voit bien que après que le joueur ait choisi son domino il est placé sur le plateau et on passe à un autre joueur (partie() et jouer() ✓)

```
=== Plateau de dominos ===  
6 | 6  
  
=== Deck de Joueur 1 ===  
2 | 2 → 0 | 0 4 | 6 4 | 4 3 | 4 5 | 6 2 | 5 PIOCHER
```



```
=== Plateau de dominos ===  
6 | 6 6 | 4  
  
=== Deck de Joueur 2 ===  
→ 3 | 6 4 | 5 5 | 5 0 | 5 2 | 4 0 | 2 PIOCHER
```

Pour tous les autres tests, voir notre fichier python des tests unitaire (test.py)