

Compte-rendu projet jeu du président

(Mathieu LE, Tomi QYSHKOLLARI et Sami MELHAOUI)

Notre objectif lors de ce projet a été de recréer sur Python un jeu de cartes assez connu : “Le Président” (aussi nommé “Trou du cul”). Le but est de déposer toutes ses cartes en premier (jeu de 52 cartes). Après chaque nouvelle partie, un titre est attribué à chaque joueur selon leur classement. Cela leur donne des avantages ou désavantages pour la prochaine partie. Pour notre jeu, l'utilisateur affronte des ordinateurs (on parlera aussi de “bot”, venant de l'anglais): les adversaires sont des algorithmes.

Lors du projet, nos attendus et objectifs ont été modifiés. Assez simple : après nous être rendus compte que 52 (le nombre de cartes) n'était pas un multiple de 6 (le nombre de joueurs initialement prévus), nous nous sommes rabattus sur 4 joueurs (3 bots+l'utilisateur). Mais nous avons fait face à de nombreux gros problèmes qui ont retardé l'avancement du projet : cela a entraîné un manque de temps et des éléments que nous voulons ajouter, qui ne verront jamais le jour (liste non exhaustive des éléments qui auraient mérité d'être ajoutés : bouton qui modifie le temps d'attente des ordinateurs, onglet aide sur le plateau de jeu, etc.)

Tout d'abord, nous nous sommes rendus compte que comme vous l'aviez dit, **faire un projet à trois est loin d'être une chose simple**. Tout le monde travaillait plus ou moins de son côté, principalement en raison du fait que tout le monde dans le groupe n'a pas forcément le même niveau en programmation. Il s'est ainsi avéré plus difficile que prévu de répartir équitablement les tâches et de mettre en commun les programmes de chacun. De ce constat, nous pensions que séparer les tâches entre l'affichage Tkinter et le cœur du jeu était une bonne idée. Ces 2 tâches ont été réalisées séparément, le cœur du jeu utilisait ainsi la console pour afficher les éléments du jeu (cela devait être temporaire). Mais nous nous sommes alors retrouvés à convertir un programme dont l'affichage est sur console, à un programme qui fonctionne sur Tkinter. Naïvement, nous pensions qu'il suffisait de remplacer l'affichage de la console en affichage Tkinter. Or cela ne marche pas comme cela : Tkinter a sa propre manière de fonctionner. Il a donc fallu adapter le programme pour le faire marcher sur Tkinter. Nous avons réussi, mais à quel prix ! Le temps perdu aurait pu être évité en réalisant tout le projet pour Tkinter. Voici l'affichage console contre l'affichage Tkinter:

```
"IDLE Shell 3.10.6"
File Edit Shell Debug Options Window Help
Python 3.10.6 (main, Nov 14 2022, 16:10:14) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: /home/mathieu/Documents/Programmes/Projet NSI/ProjetNSI_01.07.py ===
DISTRIBUTION DES CARTES
Mains actuelles : [['4 ca', '4 tr', '6 pi', '8 co', '9 co', '10ca', '10pi', '11
co', '12tr', '13ca', '13co', '15tr', '15co'], ['3 ca', '4 co', '4 pi', '5 tr', '
6 ca', '7 tr', '7 pi', '9 pi', '11pi', '12ca', '12pi', '13pi', '15ca'], ['3 co',
'5 ca', '6 co', '7 ca', '7 co', '8 ca', '8 tr', '9 ca', '10co', '12co', '13tr',
'14co', '14pi'], ['3 tr', '3 pi', '5 co', '5 pi', '6 tr', '8 pi', '9 tr', '10tr',
'11ca', '11tr', '14ca', '14tr', '15pi']]
Les cartes que vous possédez sont : ['4 ca', '4 tr', '6 pi', '8 co', '9 co', '10
ca', '10pi', '11co', '12tr', '13ca', '13co', '15tr', '15co']
Vous êtes le joueur 0
Votre main est []
La pile est []
Entrez la carte que vous voulez jouer
OU Entrez la carte que vous avez sélectionnée une nouvelle fois pour l'enlever
OU taper TERMINER quand vous voulez envoyer les cartes
OU taper PASSER pour passer votre tour]
```



Ensuite, ce projet était **trop ambitieux**. On avait annoncé beaucoup de choses, qui paraissent simples à programmer, mais qui ne le sont pas. Rien que la distribution des cartes a pris un peu plus de deux heures. Nous avons en plus voulu utiliser Pygame, pour ajouter des animations. Mais la bibliothèque est dure à maîtriser et nous nous sommes rabattus sur Tkinter seulement. Son apprentissage n'est pas non plus facile, mais plus facile que Pygame. Bien que nous soyons trois, le travail d'équipe ne compense pas la charge que nous nous étions imposés (vu au-dessus). Par conséquent, chaque élément ajouté prenait du temps à être écrit. Les heures de programmation à la maison ont été plus grandes que celles en classe. Nous n'avons également pas réalisé pleinement tout ce que nous avons annoncé : principalement le comportement des ordinateurs dont on aimerait l'améliorer, et le plateau de jeu où l'on souhaitait un affichage meilleur.

En outre, cette ambition se conjugue à la programmation qui était **difficile** pour certains car ce projet de programmation était leur premier. Ainsi, les emprunts à Internet n'ont pas été de refus pour combler les lacunes en programmation. Cet outil nous a servi à connaître les fonctions de tkinter dès qu'on en avait besoin. On s'est aussi inspiré de programme, voire copié des bouts de programme lorsque nous étions perdus (merci Chat GPT pour l'aide sur les tableaux et les noms). C'est surtout l'affichage qui en a bénéficié : Tkinter est un module que nous ne connaissions pas. Le tableau a notamment été pris sur Internet. Il s'est avéré qu'en raison de l'utilisation d'une classe spécifique de tkinter (la classe Table), la réalisation du tableau nous a pris plus de temps que prévu. Sa réalisation a été copiée sur Internet pour ce tableau mais celle-ci ne marchait tout d'abord pas. Il a ensuite fallu l'aide d'un camarade pour arriver à ce programme de tableau ressemblant à ça...

```
# Interface de fin
# Définition du Tableau de fin
# Définition de la classe
class Table:
    def __init__(self, root):
        for i in range(total_rows):
            for j in range(total_columns):
                self.e = Label(root, width=20, fg='blue',
                               text = lst[i][j],bg="white",
                               font=('Arial', 16, 'bold'))
                self.e.grid(row=i + 1, column=j)
        #Contenu des cellules
lst = [('Classement', 'Nom', 'Rang'),
       ('1er', nom_joueur[classement.index(1)], 'Président'),
       ('2e', nom_joueur[classement.index(2)], 'Vice-Président'),
       ('3e', nom_joueur[classement.index(3)], 'Vice Trou du cul'),
       ('4e', nom_joueur[classement.index(4)], 'Trou du cul')]
        #Nombre de colonnes et lignes
total_rows = len(lst)
total_columns = len(lst[0])

        # Fenêtre de fin
resultat = Toplevel(fenetre)
resultat.geometry("750x300")
resultat.config(background='#DEB887')
resultat.resizable(width=False, height=False)

        # Affichage du tableau
t = Table(resultat)
```

...Afin d'arriver à un tableau comme ceci:

Classement	Nom	Rang
1er	(Vous)	Président
2e	Alexis	Vice-Président
3e	Julie	Vice Trou du cul
4e	Yuken	Trou du cul

Voulez-vous recommencer ?

REJOUER!

NON

Enfin, nous avons eu beaucoup de mal à **optimiser notre programme**. Le manque d'expérience en Tkinter et en programmation en général et de temps en sont les causes. Dans l'ensemble du programme, les conditions sont très nombreuses, ce qui ne nous semble pas optimisées. Il y a aussi beaucoup de "global", qui ne sont pas très propres mais qui ont permis de gagner un temps considérable dans la conversion du cœur du jeu sur Tkinter. Des copies de bouts de nos programmes ont également été réalisées car les fonctions ne marchaient pas (pour afficher les images correctement). Un exemple de la programmation de la fenêtre d'aide (et sûrement d'autres éléments) pouvait sûrement être améliorée. Ce problème d'optimisation nous a fait perdre énormément de temps à un moment où nous avons perdu tout espoir d'optimisation et effectué un programme (non fonctionnel) de 5000 lignes environ avec Pygame, car nous ne savions pas comment afficher les mains (cartes que possèdent les joueurs) en fonction d'une liste contenant les cartes. Cependant, nous avons trouvé une solution plus optimisée pour par exemple afficher les images de cartes au début de la partie. Ainsi, au lieu d'écrire une montage de if/elif:

```

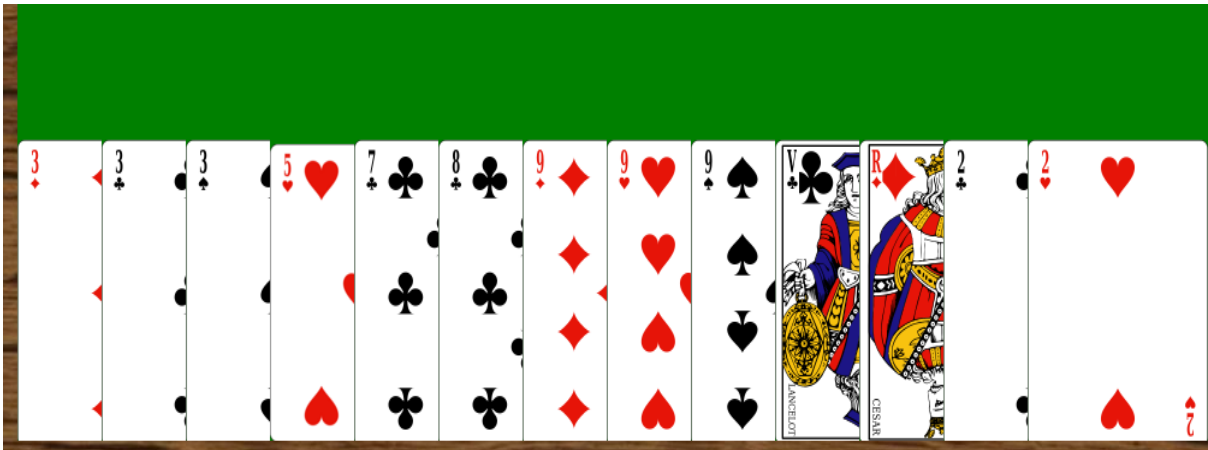
4666 elif mains[3][10] == "Dix Pique":
4667     ecran_de_jeu.blit(imageN, (1700, 750))
4668 elif mains[3][10] == "Valet Trèfle":
4669     ecran_de_jeu.blit(imageO, (1700, 750))
4670 elif mains[3][10] == "Valet Carreau":
4671     ecran_de_jeu.blit(imageP, (1700, 750))
4672 elif mains[3][10] == "Valet Cœur":
4673     ecran_de_jeu.blit(imageQ, (1700, 750))
4674 elif mains[3][10] == "Valet Pique":
4675     ecran_de_jeu.blit(imageR, (1700, 750))
4676 elif mains[3][10] == "Dame Trèfle":
4677     ecran_de_jeu.blit(imageS, (1700, 750))
4678 elif mains[3][10] == "Dame Carreau":
4679     ecran_de_jeu.blit(imageT, (1700, 750))
4680 elif mains[3][10] == "Dame Cœur":
4681     ecran_de_jeu.blit(imageU, (1700, 750))
4682 elif mains[3][10] == "Dame Pique":
4683     ecran_de_jeu.blit(imageV, (1700, 750))
4684 elif mains[3][10] == "Roi Trèfle":
4685     ecran_de_jeu.blit(imageW, (1700, 750))
4686 elif mains[3][10] == "Roi Carreau":
4687     ecran_de_jeu.blit(imageX, (1700, 750))
4688 elif mains[3][10] == "Roi Cœur":
4689     ecran_de_jeu.blit(imageY, (1700, 750))
4690 elif mains[3][10] == "Roi Pique":
4691     ecran_de_jeu.blit(imageZ, (1700, 750))
4692 elif mains[3][10] == "As Trèfle":
4693     ecran_de_jeu.blit(imagea, (1700, 750))
4694 elif mains[3][10] == "As Carreau":
4695     ecran_de_jeu.blit(imageb, (1700, 750))
4696 elif mains[3][10] == "As Cœur":
4697     ecran_de_jeu.blit(imagec, (1700, 750))
4698 elif mains[3][10] == "As Pique":
4699     ecran_de_jeu.blit(imaged, (1700, 750))
4700 elif mains[3][10] == "Deux Trèfle":
4701     ecran_de_jeu.blit(imagee, (1700, 750))
4702 elif mains[3][10] == "Deux Carreau":
4703     ecran_de_jeu.blit(imagef, (1700, 750))

```

nous avons créé une boucle pour afficher les images de cartes de chaque joueur et utilisé des listes contenant les images des cartes, puis afficher les bonnes cartes :

```
# afficher les cartes du joueur 0
for k in mains[0]:
    # Création et placement de la carte non retournée
    # Canvas
    can = Canvas(fenetre, width=150, height=313, bg="green", highlightthickness=0)
    canvas_mains[0].append(can)
    # Image
    image_deouff = can.create_image(150 / 2, 213 / 2 + 100, image=image_cartes[cartes.index(k)])
    image_mains[0].append(image_deouff)
    # Placement
    can.place(x=450 + 70 * mains[0].index(k), y=650)
```

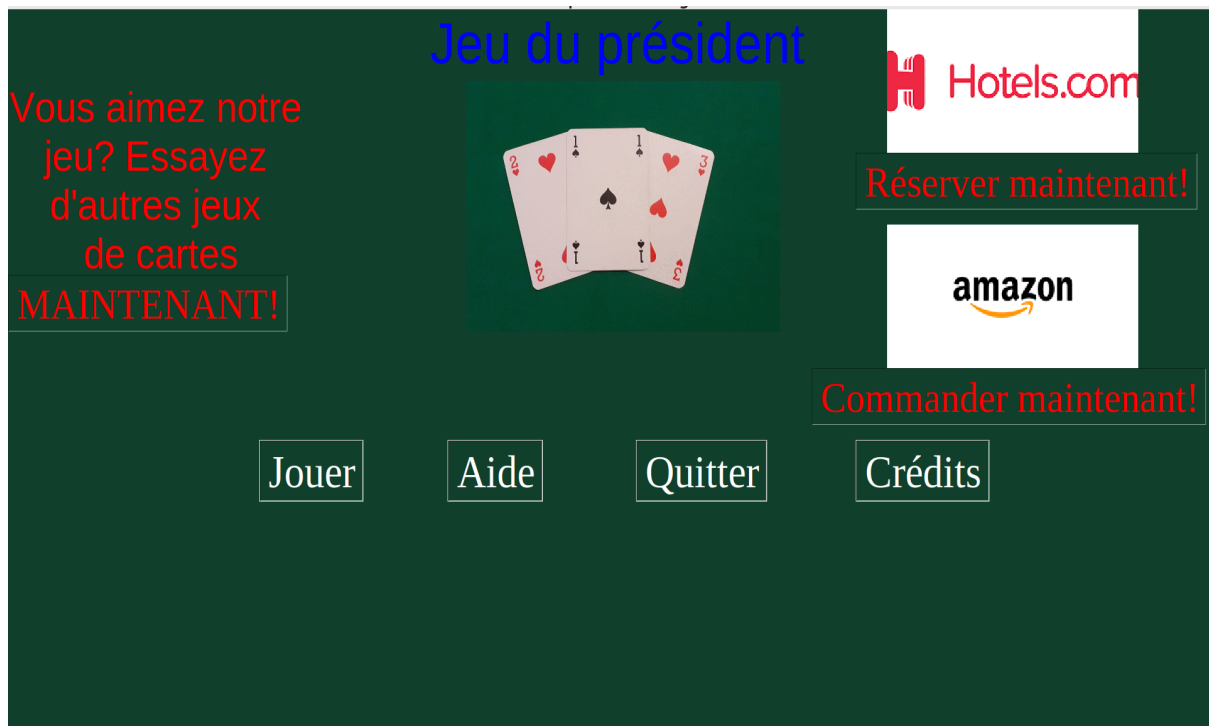
Et ce, pour le même résultat:



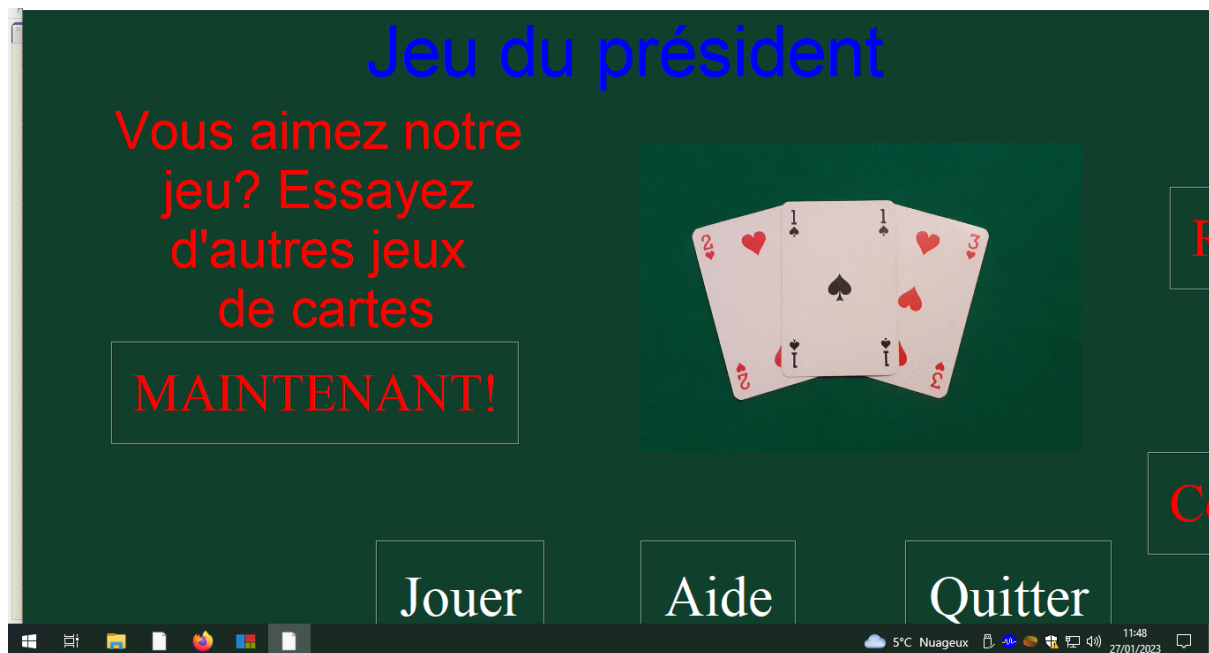
Un autre exemple est la représentation des cartes, des mains des joueurs, des tours, du classement qui est assez complexe. L'index et l'élément de la liste sont liés : l'index indique à quel joueur appartient l'élément. Mais on se demande encore si celle-ci n'a pas handicapé notre projet. On se retrouve parfois avec ce type de ligne:

```
canvas_mains_retournees[joueur].remove(canvas_mains_retournees[joueur][mains[joueur].index(carte_enleve)])
```

NB: nous nous sommes rendus compte (beaucoup trop tard) que la position des composants changeait en fonction des ordinateurs: ainsi, alors que sur un ordinateur tout est correctement placé (ordinateur du lycée sous Linux Mint, résolution à 16:9, échelle du moniteur à 100%) :



l'image ne s'affiche pas correctement chez l'ordinateur de quelqu'un d'autre, pour les mêmes coordonnées (il faudrait placer et dimensionner tous les widgets en fonction de la résolution de l'ordinateur, mais il était trop tard pour faire une telle modification) :



- En résumé, voici **les différentes étapes de notre projet pour l'interface Tkinter**:
- Apprentissage des bases de Tkinter et programmation de l'écran d'accueil
 - Affichage des images de cartes face cachée pour les bots, face visible pour le joueur réel
 - Visuel de la fin du jeu et du passement de cartes entre Président/Trou du cul et Vice-Président/Vice-Trou du cul.

- Conversion du déroulement du jeu du coeur du jeu vers Tkinter
 - Dans le même temps **pour le coeur du jeu:**
- Distribution des cartes
- Etablissement du classement
- Sélection et échange des cartes lors d'une nouvelle partie
- Déroulement de la partie selon ses règles traditionnelles
- Programmation des bots et de la manière dont ils joueront

Les problèmes énoncés plus tôt ne sont qu'une infime partie des problèmes que nous avons eu (qui ne sont pas intéressants à présenter).

Bien que notre jeu soit loin d'être parfait et qu'il possède sans doute plusieurs bugs que nous n'avons pas pu identifier, nous sommes tous contents de l'effort que nous y avons mis. Voici une évolution du projet (assez grossière):

Distribution des cartes

```

joueur 0 : ['05 T', '05 Co', '05 P', '07 Co', '08 T', '09 Co', '09 P', '11 Ca', '11 Co', '11 P', '13 P', '15 T', '15 P']
joueur 1 : ['03 Ca', '03 P', '06 Ca', '06 Co', '08 Ca', '08 P', '09 Ca', '10 Ca', '12 Co', '12 P', '13 Ca', '14 Co', '15 Ca']
joueur 2 : ['03 T', '04 Ca', '04 P', '06 T', '07 Ca', '07 T', '10 T', '12 Ca', '13 T', '13 Co', '14 Ca', '14 T', '15 Co']
joueur 3 : ['03 Co', '04 T', '04 Co', '05 Ca', '06 P', '07 P', '08 Co', '09 T', '10 Co', '10 P', '11 T', '12 T', '14 P']
  
```

Réalisation du classement

```

Le joueur 0 a terminé par un 2...., il finira dernier

Mains des joueurs en fin de partie :
joueur 0 : []
joueur 1 : ['15 Ca']
joueur 2 : ['09 T']
joueur 3 : []
Premier classement: (classement pour les joueurs ayant terminés avant l'utilisateur)
joueur 0 : 1 ème
joueur 1 : 4 ème
joueur 2 : 4 ème
joueur 3 : 2 ème
Classement intermédiaire: (prise en compte du nombre de cartes et prise en compte des égalités)
joueur 0 : 1 ème
joueur 1 : 3 ème
joueur 2 : 4 ème
joueur 3 : 2 ème
Classement final: (si quelqu'un a joué un 2 en dernière carte)
joueur 0 : 4 ème
joueur 1 : 2 ème
joueur 2 : 3 ème
joueur 3 : 1 ème
  
```

Affichage du menu

The screenshot shows a dark green menu for a game titled "Jeu du président". On the left, there is a promotional message: "Vous aimez notre jeu? Essayez d'autres jeux de cartes MAINTENANT!". In the center, there is an image of four playing cards: the 2 of hearts, the 3 of spades, the 4 of hearts, and the 5 of hearts. On the right, there are two advertisement banners: one for Hotels.com with the text "Réservé maintenant!" and one for Amazon with the text "Commander maintenant!". At the bottom, there are four buttons: "Jouer", "Aide", "Quitter", and "Crédits".

Sélection des cartes sur console

```
Votre main est []
La pile est [['04 T']]
Entrez la carte que vous voulez jouer
OU Entrez la carte que vous avez sélectionnée une nouvelle fois pour l'enlever
OU taper TERMINEE quand vous voulez envoyer les cartes
OU taper PASSER pour passer votre tour:03 T

La carte sélectionnée est trop faible
Votre main est []
La pile est [['04 T']]
```

Affichage d'un plateau de jeu

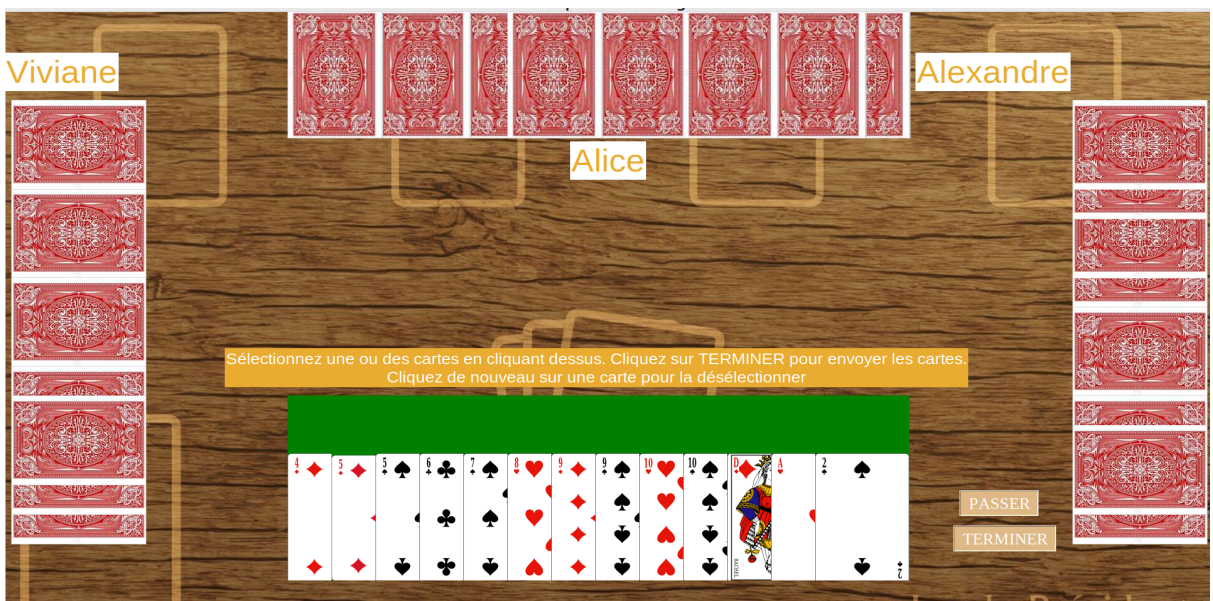


Tableau de fin

Classement	Nom	Rang
1er	(Vous)	Président
2e	Alexis	Vice-Président
3e	Julie	Vice Trou du cul
4e	Yuken	Trou du cul

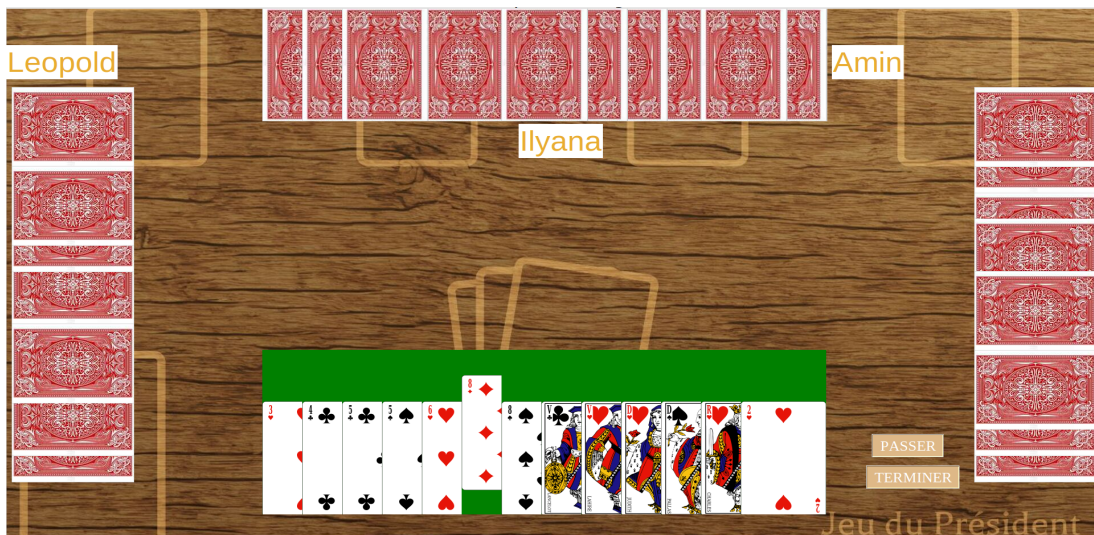
Voulez-vous recommencer ?

REJOUER!
NON

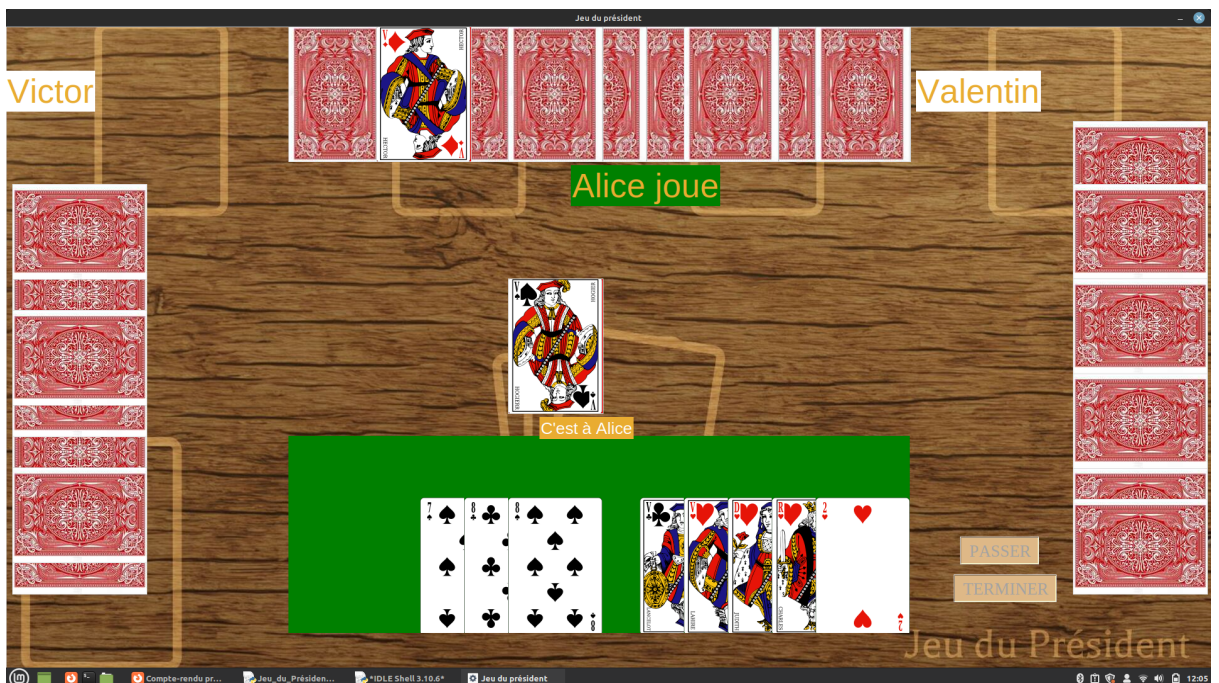
Fonctionnement de l'ordinateur

```
La pile est [['04 T']]
Les cartes que vous possédez sont : ['03 Co', '03 P', '05 Co', '08 Ca', '08 T', '09 Co', '10 Ca', '10 P', '11 Co', '12 Ca', '12 T', '12 P', '14 T']
Vous êtes le joueur 1
La pile est [['04 T'], ['05 Co']]
Les cartes que vous possédez sont : ['04 Ca', '04 Co', '06 Ca', '07 Co', '09 P', '10 T', '10 Co', '11 Ca', '11 P', '12 Co', '13 Ca', '13 Co', '14 Ca']
Vous êtes le joueur 2
La pile est [['04 T'], ['05 Co'], ['06 Ca']]
Les cartes que vous possédez sont : ['03 Ca', '03 T', '05 Ca', '05 T', '06 Co', '07 Ca', '08 Co', '08 P', '09 Ca', '14 Co', '15 T', '15 Co', '15 P']
Vous êtes le joueur 3
La pile est [['04 T'], ['05 Co'], ['06 Ca'], ['06 Co']]
Vous êtes le joueur 0
Vous vous êtes fait sauté votre tour
```

Sélection des cartes sur Tkinter



Déplacement des images



Passement des cartes entre joueurs

