



Projet Python : Jeu de la Bataille

**Présenté par :
Hugo et Aounou**

Sommaire :

- -présentation du jeu
- -présentation algorithmme
- -La fonction clé du code
- -déroulement de la partie de bataille
- -présentation répartition des parties réalisées dans le code
- -bilan

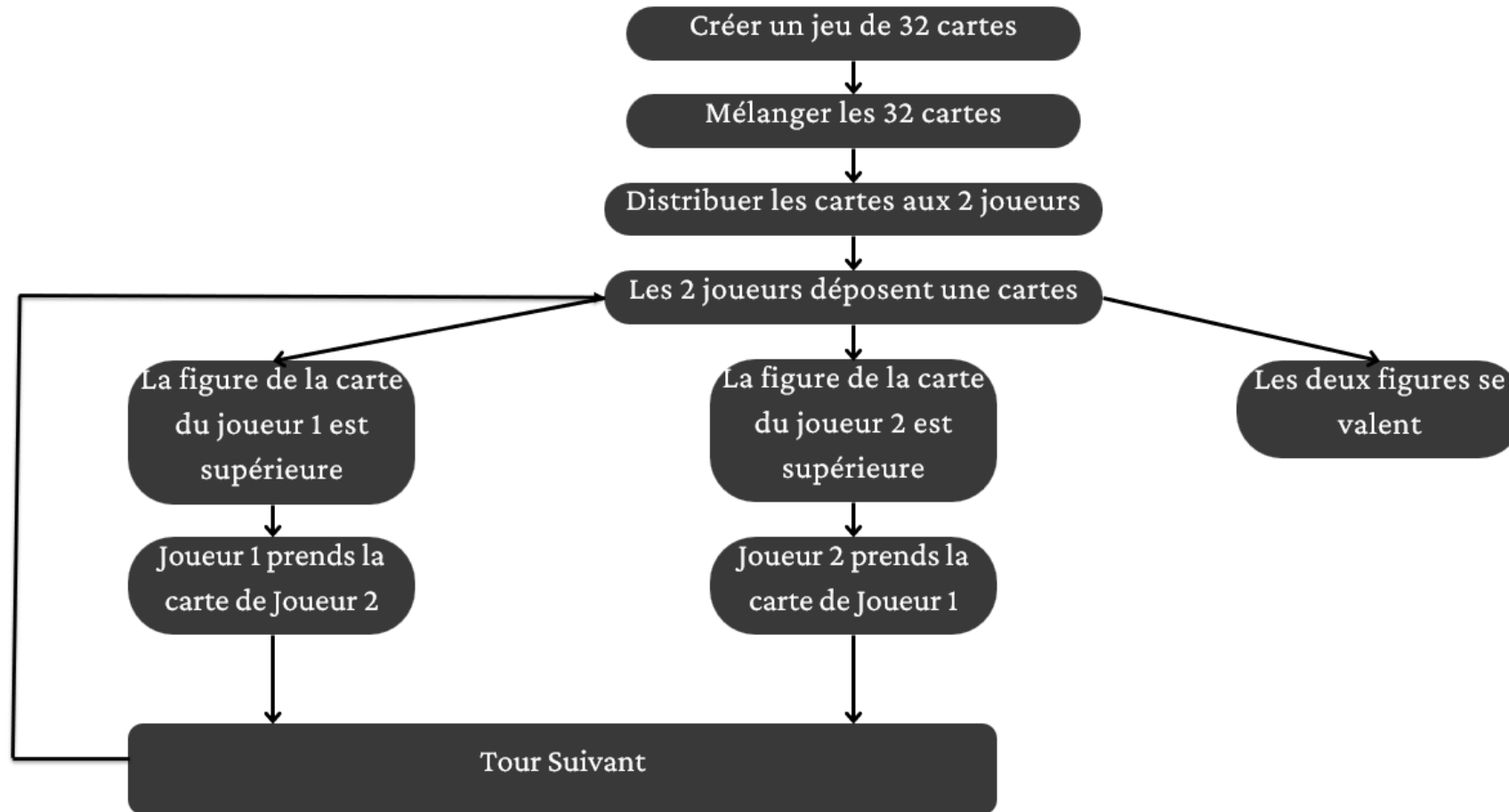


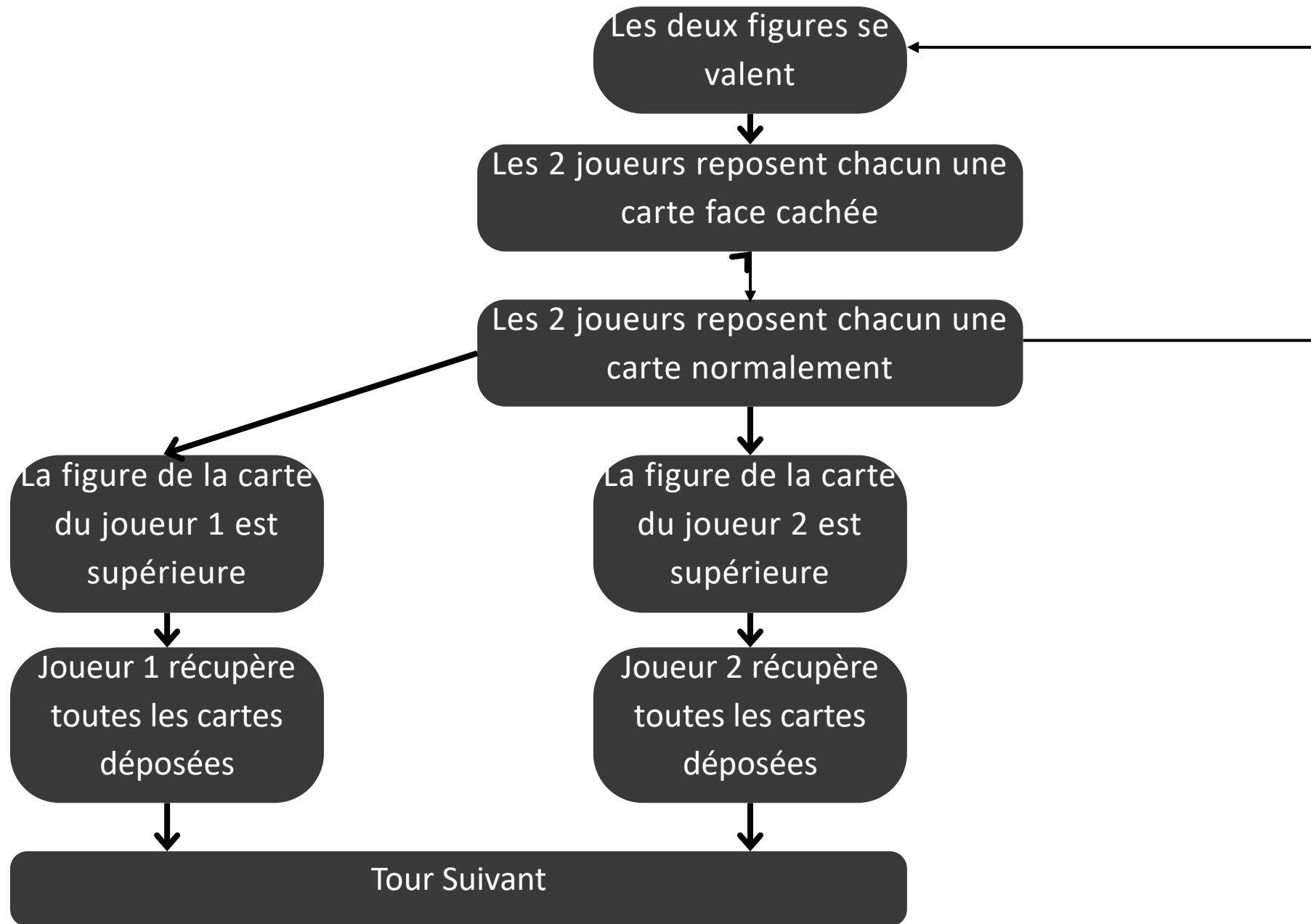
Le Jeu de la Bataille !

Règles du jeu

- Celui qui tire dans le paquet la carte la plus forte est le donneur.
- Le donneur bat les cartes, et fait « couper » le paquet à son partenaire. Il distribue les cartes une à une, en commençant par son partenaire, faces cachées, jusqu'à la dernière. Les joueurs rassemblent leurs cartes en un paquet, faces cachées.
- L'autre joueur retourne sur la table la carte du dessus de son paquet (celle reçue en dernier), puis le donneur en fait autant.
- Celui qui a la plus forte carte prend les 2 cartes et les met sous son paquet.
- Si les 2 joueurs posent des cartes de même valeur, on dit qu'il y a « bataille ». Chacun doit alors recouvrir sa première carte d'une carte face cachée, puis d'une autre face visible.

Algorithme





Fonction clé du code

```
def GérerTour():
    i=1
    l=[]
    nouveauTour=True
    while len(j1)>0 and len(j2)>0 and i<200:
#Au cas où si on sort d'une bataille
        if nouveauTour:
            print("Tour n°",i)
            Carte1,Carte2=DéposerCartes()
#stock les valeurs des upplets dans des variables
            Figure1,Couleur1=Carte1
            Figure2,Couleur2=Carte2
#Affectation de valeurs pour Valets, Dame, Roi et As comme c'est des chaines
#de caractères
            if Figure1=="Valet":
                Figure1=11
            elif Figure1=="Dame":
                Figure1=12
            elif Figure1=="Roi":
                Figure1=13
            elif Figure1=="As":
                Figure1=14
            if Figure2=="Valet":
                Figure2=11
            elif Figure2=="Dame":
                Figure2=12
            elif Figure2=="Roi":
                Figure2=13
            elif Figure2=="As":
                Figure2=14

            if Figure1 > Figure2 :
                j1.append(Carte1)
                j1.append(Carte2)
                for elem in l:
                    j1.append(elem)
                l=[]
                nouveauTour=True
            elif Figure1 < Figure2 :
                j2.append(Carte1)
                j2.append(Carte2)
                for iteration in l:
                    j2.append(iteration)
                l=[]
                nouveauTour=True
            else:
#Cartes de même valeurs: Bataille
                l.append(Carte1)
                l.append(Carte2)
                print("Bataille !")
                bataille(l)
                nouveauTour=False
            if nouveauTour:
                i+=1
                print()
                print("Main de J1:",j1)
                print("Main de J2:",j2)
                print()
            if len(j1)<=len(j2):
                print("Le joueur 2 a gagné")
            else:
                print("Le joueur 1 a gagné")
```

Répartition des parties réalisées dans le code:

Partie Hugo seul :

```
from random import *
from collections import deque

def Creer_Cartes():
    liste_motif=['\u2660','\u2666','\u2665','\u2663']
    carte=[]
    for c in range(0,4):
        for f in range(7,15):
            if f==11:
                carte.append(("Valet",liste_motif[c]))
            elif f==12:
                carte.append(("Dame",liste_motif[c]))
            elif f==13:
                carte.append(("Roi",liste_motif[c]))
            elif f==14:
                carte.append(("As",liste_motif[c]))
            else:
                carte.append((f,liste_motif[c]))
    print(carte)
    return carte

cartes=Creer_Cartes()
```

```
def Battre_Carte(liste):
    shuffle(liste)

Battre_Carte(cartes)

def distribuer(cartes):
    joueur1=cartes[0:16]
    print("joueur1=",joueur1)
    print()
    joueur2=cartes[16:32]
    print("joueur2=",joueur2,"\n")
    return joueur1, joueur2

joueur1,joueur2=distribuer(cartes)
```

Répartition des parties réalisées dans le code:

Partie Aounou seul :

```
def bataille(l):  
    if len(j1)!=0 and len(j2)!=0:  
        Carte1=j1.popleft()  
        print("Carte cachée du joueur 1:",Carte1)  
        l.append(Carte1)  
        Carte2=j2.popleft()  
        print("Carte cachée du joueur 2:",Carte2)  
        l.append(Carte2)  
    #On met les cartes cachées dans l  
    else:  
        i=200  
    #Pour arrêter la boucle while de GérerTour
```

```
def RemplirFile(pile):  
    file=deque(pile)  
    return file  
  
print()  
j1=RemplirFile(joueur1)  
j2=RemplirFile(joueur2)  
  
def DéposerCartes():  
    CarteDeJ1=j1.popleft()  
    CarteDeJ2=j2.popleft()  
    print("Carte de J1:",CarteDeJ1)  
    print("Carte de J2:",CarteDeJ2)  
    return CarteDeJ1,CarteDeJ2
```


Répartition des parties réalisées dans le code:

Partie codage en duo :

```
def GérerTour():
    i=1
    l=[]
    nouveauTour=True
    while len(j1)>0 and len(j2)>0 and i<200:
#Au cas où si on sort d'une bataille
        if nouveauTour:
            print("Tour n°",i)
            Carte1,Carte2=DéposerCartes()
#stock les valeurs des upplets dans des variables
            Figure1,Couleur1=Carte1
            Figure2,Couleur2=Carte2
#Affectation de valeurs pour Valets, Dame, Roi et As comme c'est des chaines de caractères
            if Figure1=="Valet":
                Figure1=11
            elif Figure1=="Dame":
                Figure1=12
            elif Figure1=="Roi":
                Figure1=13
            elif Figure1=="As":
                Figure1=14
            if Figure2=="Valet":
                Figure2=11
            elif Figure2=="Dame":
                Figure2=12
            elif Figure2=="Roi":
                Figure2=13
            elif Figure2=="As":
                Figure2=14
#On compare les valeurs des figures de chaque cartes, on ajoute également les éléments de l au cas où si on sort d'une bataille
            if Figure1 > Figure2:
```

```
                if Figure1 > Figure2 :
                    j1.append(Carte1)
                    j1.append(Carte2)
                    for elem in l:
                        j1.append(elem)
                    l=[]
                    nouveauTour=True
                elif Figure1 < Figure2 :
                    j2.append(Carte1)
                    j2.append(Carte2)
                    for iteration in l:
                        j2.append(iteration)
                    l=[]
                    nouveauTour=True
                else:
#Cartes de même valeurs: Bataille
                    l.append(Carte1)
                    l.append(Carte2)
                    print("Bataille !")
                    bataille(l)
                    nouveauTour=False
            if nouveauTour:
#Si bataille, on annule l'itération et on refait pareil que d'habitude
                i+=1
                print()
                print("Main de J1:",j1)
                print("Main de J2:",j2)
                print()
            if len(j1)<=len(j2):
                print("Le joueur 2 a gagné")
            else:
                print("Le joueur 1 a gagné")

GérerTour()
```

BILAN

