

Simulation d'un jeu en Python

a. Avis sur le joueur gagnant.

b. Programme sans fonction

```
from lycee import *
seuil=input("Saisir la différence des écarts de 0 à 99")

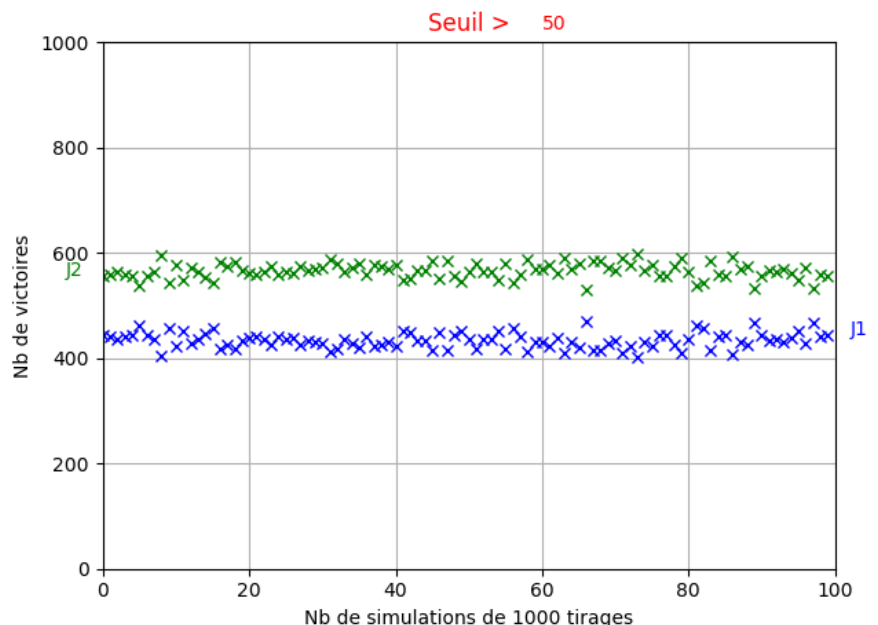
##Création de la liste des abscisses
abscisses=[]
for i in range(100):
    abscisses.append(i)                                #abscisses est la liste [0,1,2,...,99]

##Création de la liste des ordonnées
ordonnees_J1=[]
ordonnees_J2=[]
for j in range(100):
    nb_vict_J1 = 0
    nb_vict_J2 = 0
    for i in range(1000):
        alea_J1=randint(1,75)
        alea_J2=randint(26,100)
        if alea_J2-alea_J1 > int(seuil):
            nb_vict_J1 = nb_vict_J1 + 1
        elif alea_J1 > alea_J2:
            nb_vict_J1 = nb_vict_J1 + 1
        else:
            nb_vict_J2 = nb_vict_J2 + 1
    ordonnees_J1.append(nb_vict_J1)                    #liste du nb de victoires du joueur 1
    ordonnees_J2.append(nb_vict_J2)

##Partie graphique
repere.clf()                                           #Efface le graphique
repere.axis([0,100,0,1000])                          #Bornes de la fenêtre
repere.grid(True)                                     #Affiche la grille
repere.xlabel("Nb de simulations de 1000 tirages")   #Affiche la légende en abscisses
repere.ylabel("Nb de victoires")
repere.text(102,ordonnees_J1[0],"J1",color="blue")   #Affiche le texte « J1 » en bleu
repere.text(-5,ordonnees_J2[0],"J2",color="green")   #Affiche le texte « J2 » en vert
repere.title("Seuil >",color="red")                 #Affiche le titre
repere.text(60,1025,seuil,color="red")
repere.plot(abscisses,ordonnees_J1,"bx")             #Trace le nuage de points du joueur 1
repere.plot(abscisses,ordonnees_J2,"gx")
repere.show()                                         #Affiche le graphique
```

c. Cent simulations de 1000 tirages ont donné les résultats ci-contre donnés sous forme de graphique.

On peut estimer que le joueur 2 est gagnant avec une fréquence d'environ 0,58.



Simulation d'un jeu en Python

d. Programme avec fonction

```
from lycee import *

##Fonction des simulations de tirages et des tracés
def simul(seuil):
    ##Création de la liste des abscisses
    abscisses=[]
    for i in range(100):
        abscisses.append(i)                    #abscisses est la liste [0,1,2,...,99]

    ##Création de la liste des ordonnées
    ordonnees_J1,ordonnees_J2=[],[]
    for j in range(100):
        nb_vict_J1 = 0
        nb_vict_J2 = 0
        for i in range(1000):
            alea_J1=randint(1,75)
            alea_J2=randint(25,100)
            if alea_J2-alea_J1 > int(seuil) or alea_J1 > alea_J2:
                nb_vict_J1 = nb_vict_J1 + 1
            else:
                nb_vict_J2 = nb_vict_J2 + 1
        ordonnees_J1.append(nb_vict_J1) #liste du nb de victoires du joueur 1
        ordonnees_J2.append(nb_vict_J2)

    ##Réglage de la fenêtre graphique et tracés
    repere.axis([0,100,0,1000])                #Bornes de la fenêtre
    repere.grid(True)                          #Affiche la grille
    repere.xlabel("Nb de simulations de 1000 tirages") #Affiche la légende en abscisses
    repere.ylabel("Nb de victoires")
    repere.text(102,ordonnees_J1[0],"J1",color="blue") #Affiche le texte "J1" en bleu
    repere.text(-7,ordonnees_J2[0],"J2",color="orange")
    repere.title(seuil,color="red")             #Affiche le titre
    repere.plot(abscisses,ordonnees_J1)         #Trace le nuage de points du joueur 1
    repere.plot(abscisses,ordonnees_J2)

##Lancement des simulations et des tracés graphiques
repere.subplot(221)    #Positionne le 1er tracé dans la partie 1 du carré 2x2
simul(34)              #Lance la simulation et le tracé pour le seuil 34
repere.subplot(222)
simul(43)
repere.subplot(223)
simul(50)
repere.subplot(224)
simul(55)
repere.show()          #Affiche le graphique
```

Simulation d'un jeu en Python

Les simulations permettent d'estimer l'écart $N_2 - N_1$ à 43 pour que le jeu soit équitable.

