2.2 Etude de jeu de cartes :

1. Cartes vertes : 10

Cartes Violettes : 10

Cartes Oranges : 4

Le jeu a généré en tout 24 cartes

2. Les cartes Oranges n’ont aucune valeur c’est-à-dire 0.

3. La couleur verte est représentée par un 1.

4. Les cartes vertes ont pour valeurs : 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9 ; 10

Les cartes violettes ont pour valeurs : 20 ; 40 ; 60 ; 80 ; 100 ; 20 ; 40 ; 60 ; 80 ; 100.

5. Le 5 vert est représenté comme la carte qui a le plus de valeur.

6. Le début et la fin de la liste générée par la fonction crea\_jeu() :

[[1, 1], [1, 2], [1, 3], [1, 4], [1, 5], [1, 6], [1, 7], [1, 8], [1, 9], [1, 10], [0, 20], [0, 40], [0, 60], [0, 80], [0, 100], [0, 20], [0, 40], [0, 60], [0, 80], [0, 100], [2, 0], [2, 0], [2, 0], [2, 0]]

3.2 qui gagne le Pli ?:

def pli(carte1,carte2):  
 *"""carte1 et carte2:listes de la forme[Couleur,Valeur]"""* """retourne une liste:carte ayant remporté le pli"""  
 """retourne None en cas d'egalite"""  
 if carte1==carte2:  
 return(None)  
 elif carte1[0]>carte2[0]:  
 return(carte1)  
 elif carte1[0]<carte2[0]:  
 return(carte2)  
 elif carte1[0]==carte2[0]:  
 if carte1[1]>carte2[1]:  
 return(carte1)  
 elif carte1[1]<carte2[1]:  
 return(carte2)  
 gagnant = carte1 or carte2  
 if carte1 < carte2:  
 gagnant = carte2  
 else:  
 gagnant = carte1  
  
 return gagnant  
print(pli([1, 150], [1, 100]))

3.3 Elaborer la stratégie du jeu

3.3.1 la stratégie du robot

3.3.2 Algorithme en language naturel

Fonction jouer une carte de niveau 0 :

from random import choice  
def JouerUneCarte(ListeCartes):  
 return choice(ListeCartes)  
print(JouerUneCarte([[1, 2], [0, 3], [1, 5], [1, 2], [0,1]]))

Fonction jouer une carte de niveau 1 :