# -\*- coding: cp1252 -\*-  
#  
  
###############=============================##############  
###############============Créée============##############  
###############=============Par=============##############  
###############============Elnabo===========##############  
###############=============12/09===========##############  
###############=============================##############  
  
#On importe les fonctions necessaires  
  
from Tkinter import \*  
from random import randrange  
  
#Fonction la plus compliquée permettant le déplacement du serpent  
def deplacement():  
 global a,b,z,y,lu,lv,score,serpent,j,m  
 c=len(serpent)  
 c=c-1  
 #Chaque carré reprend la coordonnée du précédent dans la liste (serpent)  
 while c!=0 :  
 lu[c]=lu[c-1]  
 lv[c]=lv[c-1]  
 c+=-1  
 #On change les coordonées du premier carré  
 lu[0] += a  
 lv[0] += b  
 c=0  
 #On applique les nouvelles coordonnées aux carrés correspondant  
 while c!=len(serpent):  
 can.coords(serpent[c],lu[c],lv[c],lu[c]+10,lv[c]+10)  
 c+=1  
 c=1  
 #Si les coordonnées du premier carré sont égales à celle d'un autre le jeu s'arrêtera  
 while c!=len(serpent):  
 if lu[c]==lu[0] and lv[c]==lv[0]:  
 j=1  
 score = 'Perdu avec ' + str(score\*10)  
 scores.set(score)  
 break  
 c+=1  
 #Si le serpent est mord un coté il ressort de l'autre  
 #La valeur 'd' sert à empecher un bug empechant la transfert du serpent de l'autre coté du canvevas  
 d=1  
 if lu[0]==200:  
 lu[0],d=10,0  
 if lu[0]==0 and d==1:  
 lu[0]=200  
 if lv[0]==200:  
 lv[0],d=10,0  
 if lv[0]==0 and d==1:  
 lv[0]=200  
 d=0  
 #Si le carré de tête recoupe le cercle, le score augmente et un nouveau cercle apparait aléatoirement  
 if z-7<=lu[0]<=z+7 and y-7<=lv[0]<=y+7:  
 score+=1  
 scores.set(str(score\*10))  
 bestiole()  
 if j!=1 and m!=1:  
 fen.after(100,deplacement)  
  
#Cette fonction crée un cercle de coordonée multiple de 10 pour éviter que le cercle soit partiellement coupé par le serpent  
  
def bestiole():  
 global z,y,n,lu,lv,serpent,a,b  
 z=randrange(2,18)  
 y=randrange(2,18)  
 z = z\*10  
 y = y\*10  
 can.coords(cercle,z,y,z+5,y+5)  
 #On ajoute un carré hors du canevas (pour allèger le code) qui se rajoutera à la suite  
 serpents = can.create\_rectangle(300,300,310,310,fill='green')  
 serpent.append(serpents)  
 lu.append(lu[n]+12+a)  
 lv.append(lv[n]+12+b)  
 n+=1  
  
#Ces quatres fonctions permettent le déplacement dans quatres directions du serpent  
#Grace aux modifications successives des coordonées du premier carrée grave au valeur a et b  
#La valeur s permet de ne pas accelerer la vitesse du serpent ou à modifier ca direction  
#en appuyant successivement sur Haut/Bas/Gauche/Droite  
  
def gauche(event):  
 global a,b,s  
 a=-10  
 b=0  
 if s==0:  
 s=1  
 deplacement()  
  
def droite(event):  
 global a,b,s  
 a=10  
 b=0  
 if s==0:  
 s=1  
 deplacement()  
   
def haut(event):  
 global a,b,s  
 a=0  
 b=-10  
 if s==0:  
 s=1  
 deplacement()  
  
   
def bas(event):  
 global a,b,s  
 a=0  
 b=10  
 if s==0:  
 s=1  
 deplacement()  
  
#Cette fonction permet d'arrêter le serpent  
   
def pause(event):  
 global j,a,b,m,enpause  
 t=0  
 if a==b:  
 t=1  
 if j!=1:  
 #Affichage ou Effacage du texte 'PAUSE'  
 #Et arrêt du serpent  
 if m!=1:  
 m=1  
 can.coords(enpause,100,100)  
 else:  
 m=0  
 can.coords(enpause,300,300)  
 if t!=1:  
 deplacement()  
  
#Cette fonction réinitialise toutes les valeurs et recréée le serpent de base ainsi que le premier repas  
  
def recommencer(event):  
 global z,y,lu,lv,score,serpent,j,m,s,n,a,b,cercle  
 if j!=1:  
 print 'Le suicide est puni'  
 can.delete(ALL)  
 s=score=j=m=a=b=0  
 z=y=50  
 lu,lv,serpent = [100,112],[100,112],[]  
 n=1  
 tete = can.create\_rectangle(100,100,110,110,fill='dark green')  
 carre = can.create\_rectangle(112,100,122,110,fill='green')  
 cercle = can.create\_oval(z,y,z+5,y+5,fill='red')  
 serpent.append(tete)  
 serpent.append(carre)  
 scores.set('0')  
  
#On définit les valeurs initiales  
  
s=score=j=m=t=a=b=0  
z=y=50  
lu,lv,serpent = [100,112],[100,112],[]  
n=1  
  
print ' '\*35 + 'Les fleches pour bouger'  
print ' '\*35 + 'P pour mettre/enlever la pause'  
print ' '\*35 + 'Entree pour recommencer, attention au suicide'  
  
#On crée un canevas tout gris  
  
fen = Tk()  
can = Canvas(fen,width = 200, height = 200 , bg = 'gray')  
can.grid(row=1,column=0,columnspan=3)  
  
enpause=can.create\_text(300,300,text="PAUSE")  
  
#On crée la base du serpent ainsi que le premier repas  
  
tete = can.create\_rectangle(100,100,110,110,fill='dark green')  
carre = can.create\_rectangle(112,100,122,110,fill='green')  
cercle = can.create\_oval(z,y,z+5,y+5,fill='red')  
  
serpent.append(tete)  
serpent.append(carre)  
  
#On crée les commandes au clavier  
  
can.bind\_all('<Up>', haut)  
can.bind\_all('<Down>', bas)  
can.bind\_all('<Left>', gauche)  
can.bind\_all('<Right>', droite)  
can.bind\_all('<Return>',recommencer)  
can.bind\_all('p',pause)  
  
#L'affichage du score  
  
Label(fen, text='Score: ').grid(row=0,column=0)  
  
scores = StringVar()  
Score = Entry(fen, textvariable=scores)  
Score.grid(row=0,column=1)  
scores.set('0')  
  
fen.mainloop()