

# SR01-TP3

## Programmation Python

*Nhat-Minh Doan & François Marès*

L'objectif de ce TP est d'implémenter le jeu de vie en utilisant une interface graphique (GUI pour *graphical user interface*) réalisée grâce au package Tkinter.

## 1 Le jeu de la vie

Le jeu de la vie que l'on a implémenté évolue sur un damier sous la forme d'une surface torique ; il est représenté par une matrice dont les bords droite et gauche sont reliés entre eux, ainsi que les bords supérieur et inférieur. Chaque case est occupée par une cellule qui peut être soit vivante soit morte.

A chaque génération, l'état de chaque cellule est reconsidéré, en fonction de son contexte déterminé par le nombre de ses voisins vivants. Les règles qui permettent de passer d'une génération à l'autre sont les suivantes :

- Pour une cellule vivante :
  - Si elle a exactement 2 ou 3 voisins vivants, elle survit à la génération suivante.
  - Si elle a au moins 4 cellules voisines vivantes, elle meurt d'étouffement à la génération suivante.
  - Si elle a au plus une cellule voisine vivante, elle meurt d'isolement à la génération suivante.
- Pour une cases vides :
  - Si elle a exactement 3 voisins vivants, une cellule naîtra à la génération suivante.

On implémente ce mécanisme à l'aide des deux fonctions suivantes (*Fonctions*, ligne 41) :

```
#----- modifications du damier
def Genetation():
    global damier
    global damierVoisins

    for i in range(jeuTaille):
        for j in range (jeuTaille):
            damierVoisins[i][j]=nbrVoisins(i,j)

    for i in range(jeuTaille):
        for j in range (jeuTaille):
            nbr_voisins=damierVoisins[i][j]
            if (damier[i][j]==1):
                if (nbr_voisins<=1 or nbr_voisins>=4):
                    damier[i][j]=0
            else:
                if (nbr_voisins==3):
                    damier[i][j]=1
```

```
def nbrVoisins(i,j):
    count=0
    for a in range(i-1,i+2):
        for b in range (j-1,j+2):
            if (a<0):
                a=jeuTaille-1
            elif (a==jeuTaille):
                a=0
            if (b<0):
                b=jeuTaille-1
            elif (b==jeuTaille):
                b=0
            if (damier[a][b]==1):
                count+=1
    if (damier[i][j]==1):
        count-=1
    return count
```

## 2 GUI avec Tkinter

Tkinter permet de créer des objets d'interface, les **widgets**, qui peuvent être disposés les uns par rapport aux autres grâce à la fonction **pack**. En effet tout widget appartient à une **Frame**, ici la fenêtre principale, dans laquelle il est situé.

### 2.1 Fenêtre principale

Création de la fenêtre principale (*GUI*, ligne 156), et de l'espace de dessin du damier à l'intérieur (à gauche) :

```
#----- Fenêtre principale
fenetre = tkinter.Tk()
fenetre.title("SR01 Jeu de la vie")
fenetre.configure(width=jeuWidth+0,height=jeuHeight+200)
fenetre.resizable(width=False,height=False)
#----- Canevas
canvas = tkinter.Canvas(fenetre, width=jeuWidth, height=jeuHeight,bg = "white")
canvas.pack(side=tkinter.LEFT)
```

### 2.2 Boutons

Les boutons sont des widgets qui appellent une fonction spécifiée par l'argument **command**, lorsque l'utilisateur clic dessus.

Création des boutons dans fenêtre principale (*GUI*, ligne 165) :

```
#----- Boutons
tkinter.Button( fenetre,text = 'Lancer',font='helvetica',fg='DeepSkyBlue4',
                bg='light gray',height = 1,width = 15,relief='raised',
                command=LancerEncore).pack(side=tkinter.TOP)

tkinter.Button( fenetre,text = 'Arreter',font='helvetica',fg='DeepSkyBlue4',
                bg='light gray',height = 1,width = 15,relief='raised',
                command = Arreter).pack(side=tkinter.TOP)

tkinter.Button( fenetre,text = 'Initialiser',font='helvetica',fg='DeepSkyBlue4',
                bg='light gray',height = 1,width = 15,relief='raised',
                command = Initialiser).pack(side=tkinter.TOP)

tkinter.Button( fenetre,text = 'A propos',font='helvetica',fg='DeepSkyBlue4',
                bg='light gray',height = 1,width = 15,relief='raised',
                command = Propos).pack(side=tkinter.TOP)

tkinter.Button( fenetre,text = 'Quitter',font='helvetica',fg='DeepSkyBlue4',
                bg='light gray',height = 1,width = 15,relief='raised',
                command = fenetre.destroy).pack(side=tkinter.BOTTOM)
```

### 2.2.1 Fonctions des boutons

Les fonctions appelées sont les suivantes :(*Fonctions*, ligne 89) :

- Création d'une fenêtre contenant une zone de texte pour afficher les informations relatives à l'application :

```
#----- fonctions des boutons
def Propos():
    top=tkinter.Toplevel()
    top.geometry("300x100+500+350")
    top.title("Bienvenue!")
    texte="SR01-TP3\n11/01/2019\nProduit sans licence\nPython
          version 3.7.1/3.5.2 sur Windows 10\n
          Par DOAN Nhat-Minh et MARES François"
    tkinter.Label(top,text=texte).pack()
```

- La fonction Lancer verifie que le damier n'est pas vide et appel la fonction récursive Continuer qui créé la boucle des générations :

```

def Continuer():
    Genetation()
    Afficher()
    if flag>0:
        fenetre.after(jeuVitesse, Continuer)

def Lancer():
    global flag
    if damier == [[0] * jeuTaille for i in range(jeuTaille)]:
        flag=0
    else:
        flag=1
    fenetre.after(jeuVitesse,Continuer)

```

- La fonction Initialiser utilise la package random pour générer une nouvelle matrice :

```

def Initialiser():
    Arreter()
    canvas.delete(damierRectangles)
    global flag
    flag=0
    for i in range(jeuTaille):
        for j in range (jeuTaille):
            k=random.randrange(0,101)
            if k<=jeuProba:
                damier[i][j]=1
            elif k>jeuProba:
                damier[i][j]=0
    Afficher()

```

- La fonction Arreter positionne le flag qui assurait la succession des générations sur 0, et efface l'affichage du damier :

```

def Arreter():
    global flag
    global damierRectangles
    flag=0
    canvas.delete(damierRectangles)

```

## 2.3 Curseurs

Enfin des curseurs permettent de changer les valeurs initiales (`set()`) de la vitesse d'enchaînement des générations, de la taille de la grille, et de la probabilité d'apparition des cellules à l'initialisation (*GUI*, ligne 178) :

```
#----- Curseurs
VitesseScale=tkinter.Scale(fenetre,label='Vitesse',font=('helvetica','10'),
                           fg='DeepSkyBlue4',troughcolor='gray88',
                           orient=tkinter.HORIZONTAL,width='20',from_=1,
                           to=5000, command=Vitesse)
VitesseScale.pack(side=tkinter.BOTTOM)
VitesseScale.set(2500)

VieScale=tkinter.Scale(fenetre,label='% de Vie',font=('helvetica','10'),
                       fg='DeepSkyBlue4',troughcolor='gray88',
                       orient=tkinter.HORIZONTAL,width='20',command=Proba)
VieScale.pack(side=tkinter.BOTTOM)
VieScale.set(50)

TailleScale=tkinter.Scale(fenetre,label='Taille de la grille',font=('helvetica','10'),
                          fg='DeepSkyBlue4',troughcolor='gray88',
                          from_=5, to=100,
                          orient=tkinter.HORIZONTAL,width='20',command=Taille)
TailleScale.pack(side=tkinter.BOTTOM)
TailleScale.set(100)
```

### 2.3.1 Fonctions des curseurs

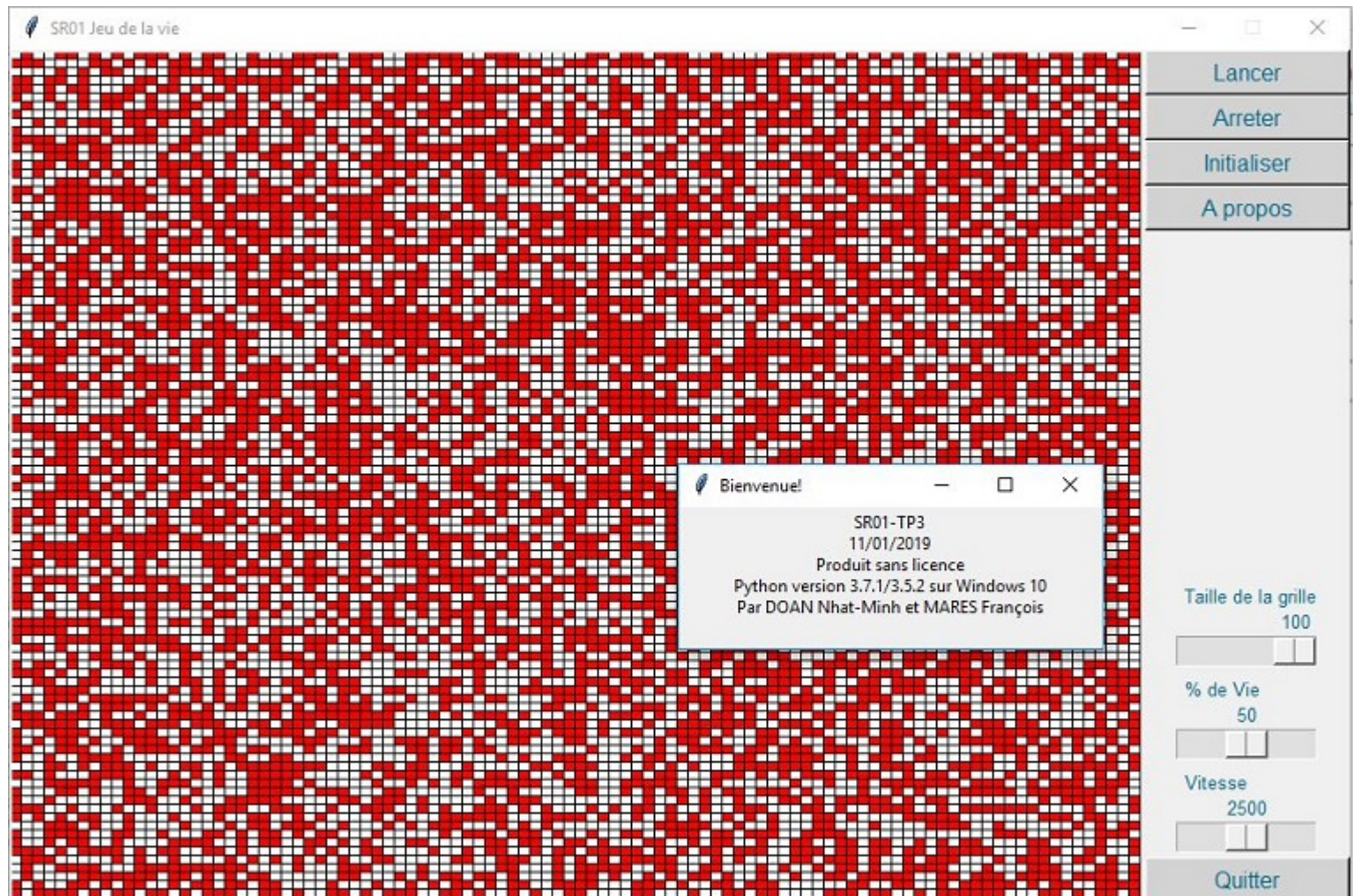
Les curseurs font appel aux fonctions suivantes (*Fonctions*, ligne 131) :

```
#----- fonctions des curseurs
def Taille(val):
    global jeuTaille
    jeuTaille=int(val)

def Proba(val):
    global jeuProba
    jeuProba=int(val)

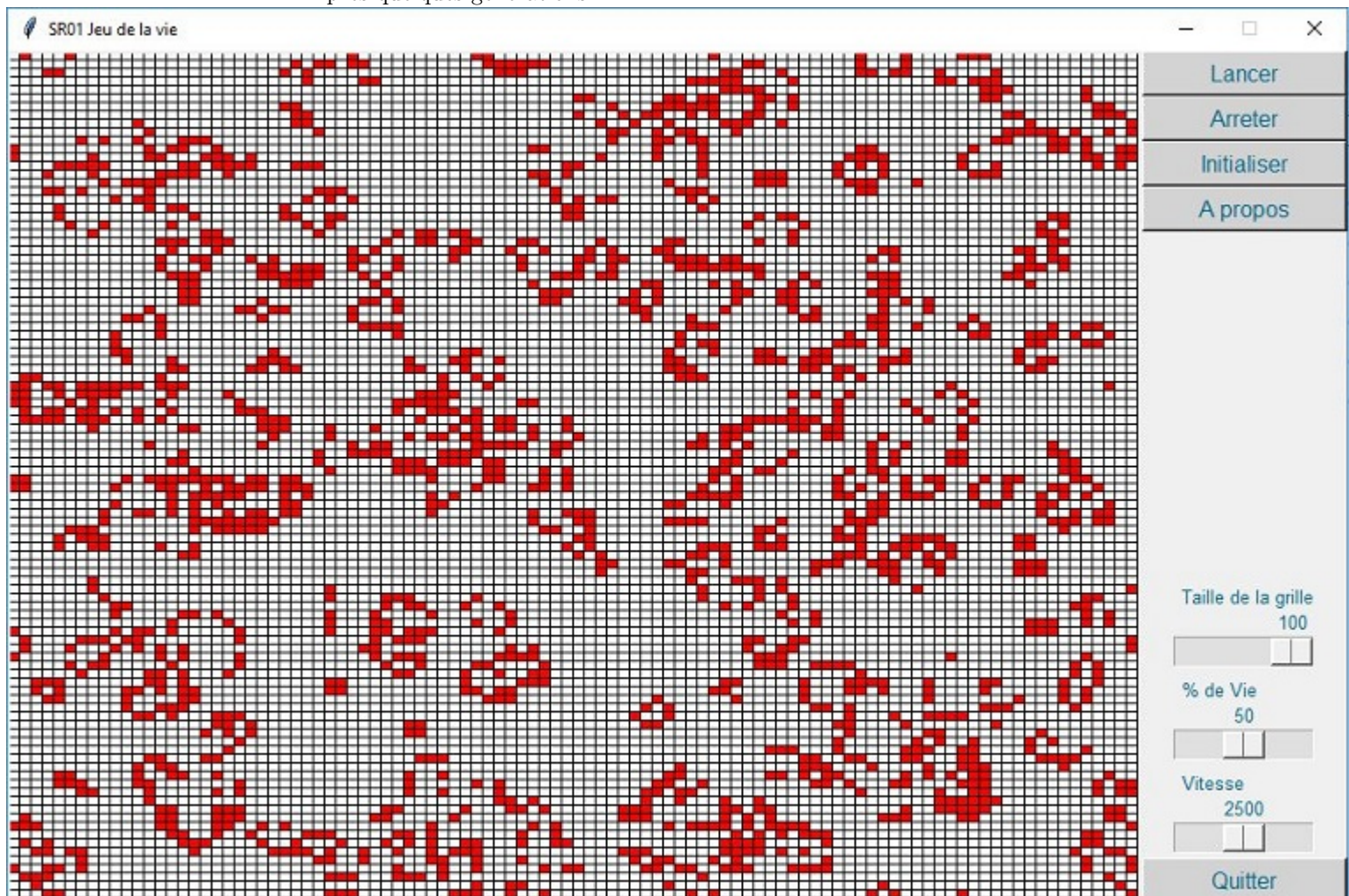
def Vitesse(val):
    global jeuVitesse
    jeuVitesse=5010-int(val)
```

### 3 Rendu





Après quelques générations :



## Table des matières

<b>1</b>	<b>Le jeu de la vie</b>	<b>1</b>
<b>2</b>	<b>GUI avec Tkinter</b>	<b>2</b>
2.1	Fenêtre principale . . . . .	2
2.2	Boutons . . . . .	2
2.2.1	Fonctions des boutons . . . . .	3
2.3	Curseurs . . . . .	4
2.3.1	Fonctions des curseurs . . . . .	5
<b>3</b>	<b>Rendu</b>	<b>6</b>