

Création d'une grille et de configurations d'un système binaire modélisé

```
<sxh python; title : grille_configurations_melange_binaire.py> #!/usr/bin/env python # -*- coding: utf-8 -*- # travail de ML et MP, ba2 chimie 2012-2013 # Création d'une grille et de configurations d'un système binaire modélisé
```

```
from Tkinter import * import math import random
```

```
root = Tk()
```

```
#Fonction appelée lors du clic sur le bouton "Rafraichir" def boutonClick():
```

```
#Effacement de la grille précédente
c.delete(ALL)
#Récupération des informations dans les différents champs
nbLig = int(eLig.get())
nbCol = int(eCol.get())
nb1 = int(e1.get())
#Regénération de la nouvelle grille
draw(nbLig, nbCol, nb1)
```

```
#Sélection nombre de lignes Label(root, text="Nombre de lignes :").pack() eLig = Entry(root)
eLig.pack() eLig.insert(0, "4") #Sélection nombre de colonnes Label(root, text="Nombre de colonnes
:").pack() eCol = Entry(root) eCol.pack() eCol.insert(0, "4") #Sélection nombre de 1 Label(root,
text="Nombre de 1 :").pack() e1 = Entry(root) e1.pack() e1.insert(0, "8")
```

```
#Bouton de génération de la grille avec le nombre de lignes, colonnes et de 1 sélectionnés b =
Button(root, text="Rafraichir la grille", command=buttonClick) b.pack()
```

```
c = Canvas(root, width=600, height=600) c.pack()
```

```
#Fonction de génération de la grille # nbLig: nombre de lignes # nbCol: nombre de colonnes # nb1:
nombre de 1 def draw(nbLig, nbCol, nb1):
```

```
#Taille de la grille
size = nbLig*nbCol
```

```
#Le traçage de la grille démarrera à la coordonnée (start, start)
#ses côtés seront de longueur sideLength
#Le point de fin d'une ligne sera donc de start+sideLength+1 (+1 afin de
voir le contour de la grille)
start = 50
sideLength = 400
end = start+sideLength+1
cellHeight = int(sideLength/nbLig)
cellWidth = int(sideLength/nbCol)
#Tableaux de 1 et de 2
table = [1]*nb1
```

```
table2 = [2]*(size-nb1)
#Fusionner le tableau de 1 avec le tableau de 2 et les réordonner de
manière aléatoire
table.extend(table2)
random.shuffle(table)
```

```
#Dessiner la grille
#Tracer les lignes horizontales
#- Déplacer le y de la coordonnée de départ (start) jusqu'à la coordonnée
de fin (end)
# par intervalles de cellHeight (sideLength/nbLig)
#- Tracer une ligne partant de la coordonnée (start, y) à la coordonnée
(end, y)
for y in range(start, end, cellHeight):
    c.create_line(start, y, end, y, fill="blue")
```

```
#Tracer les lignes verticales
#- Déplacer le x de la coordonnée de départ (start) jusqu'à la coordonnée
de fin (end)
# par intervalles de cellWidth (sideLength/nbCol)
#- Tracer une ligne partant de la coordonnée (x, start) à la coordonnée
(x, end)
for x in range(start, end, cellWidth):
    c.create_line(x, start, x, end, fill="blue")
```

```
#Remplir la grille de 1 et de 2 (présents dans le tableau)
#Le milieu de la lère case se trouve aux coordonnées suivantes:
x = start+int(cellWidth/2)
y = start+int(cellHeight/2)
```

```
#Prendre tous les éléments du tableau un par un
#i contient l'index du tableau (1 -> size)
#val contient l'élément présent dans le tableau (1 ou 2)
for i, val in enumerate(table):
    #Ecrire 1 ou 2 au milieu de la case (coordonnée (w, h))
    c.create_text(x, y, text=str(val))
```

```
#Prendre le modulo (reste de la division entière) de l'index i+1/nbCol
(i+1 le tableau démarre à 0)
#- Si = 0, il faut passer à la ligne suivante et revenir à la première
colonne
#- Sinon, il faut juste passer à la colonne suivante
if (i+1) % nbCol == 0:
    x = start+int(cellWidth/2)
    y = y + cellHeight
else:
    x = x + cellWidth
```

```
#Calcul du nombre d'états possibles
```

```
#N!/N1!(N-N1)!
nbrEtats =
int(math.factorial(size)/(math.factorial(nb1)*math.factorial(size-nb1)))
c.create_text(50, 500, text="Nombre d'états possibles: " + str(nbrEtats),
anchor="w")
```

```
root.mainloop()
```

```
</sxh>
```

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

https://dvillers.umons.ac.be/wiki/teaching:progappchim:grille_configurations_melange_binaire_2013

Last update: **2013/11/29 11:05**

