

Interaction Homme-Machine

Réaliser une application interactive avec
pyqt5

Auteur : Qianqian SHA, Ling Ma

Tuteur de IHM: Monsieur Thomas Pietrzak

Master 1 Informatique

Sommaire

Chapitre 1 Introduction	3
1.1 Introduction du sujet	3
1.2 Le plan de travail	3
1.3 Les outils utilisés	3
1.3.1 Python3	3
1.3.2 Pyqt5	3
Chapitre 2 Développement	4
2.1 Le travail pour l'application	4
2.2 Les difficultés	7
Chapitre 3 Test	7
Chapitre 4 Conclusion	10
Annexe	10

Chapitre 1 Introduction

1.1 Introduction du sujet

Pendant ce semestre, nous avons appris à utiliser pyqt5 pour réaliser une application ou on peut dessiner des cercles. Pour ajouter des fonctions interactives, les utilisateurs peuvent décider de la technique, la densité, la taille, le numéro de cible. Pour tester notre projet, on a demandé à 6 personnes de nous aider à tester notre projet et noter le temps de réponse en fonction de la technique, le temps de réponse en fonction de la technique et de la densité de cible, le temps de réponse en fonction de la technique et la taille de cible et le taux d'erreur en fonction de la technique.

1.2 Le plan de travail

Ce projet concernant l'utilisation de pyqt5 et python3, notre objectif de travail est de créer une application permettant de dessiner les cercles selon les demandes des utilisateurs. Pour apprendre et bien utiliser PyQt5, nous avons réalisé aussi 3 tp avant. Et le dernier tp a été fini en cours.

1.3 Les outils utilisés

1.3.1 Python3

Python est un langage de programmation objet, multi-paradigme et multiplateformes. Python est un langage qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives. On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses extensions destinées au calcul numérique.

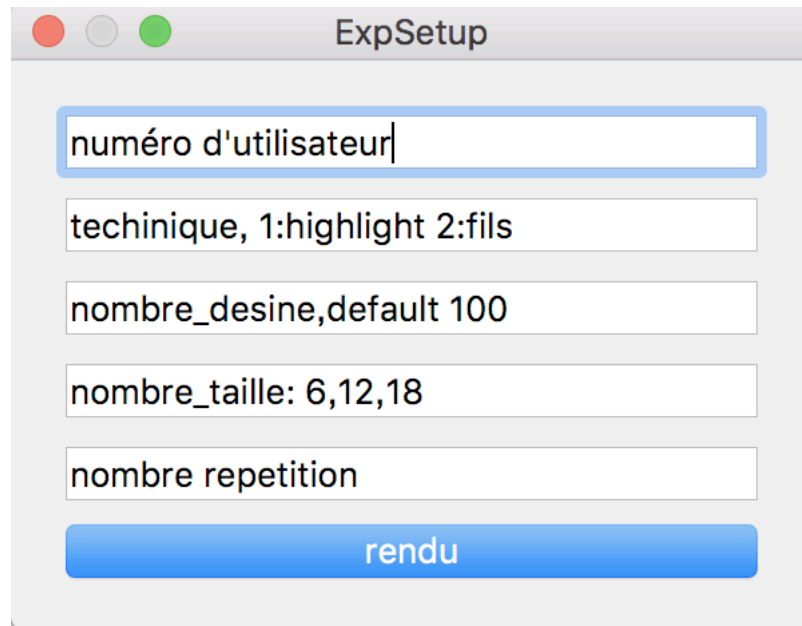
1.3.2 PyQt5

PyQt est un module libre qui permet de lier le langage Python avec la bibliothèque Qt distribué sous deux licences : une commerciale et la GNU GPL. Il permet ainsi de créer des interfaces graphiques en Python. Une extension de QtDesigner (utilitaire graphique de création d'interfaces Qt) permet de générer le code Python d'interfaces graphiques.

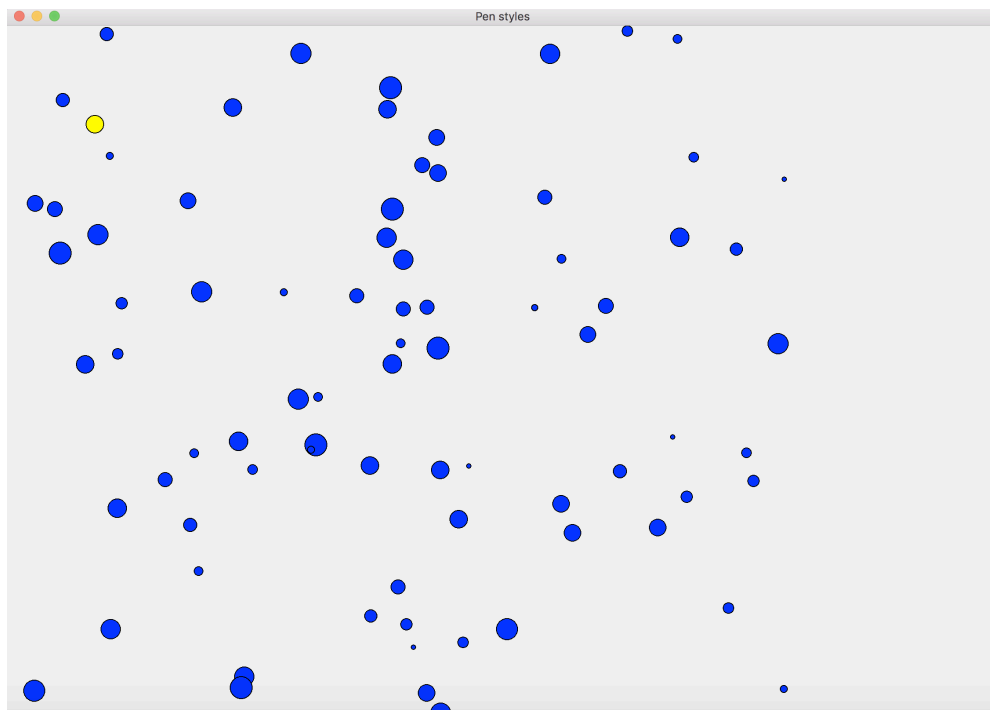
Chapitre 2 Développement

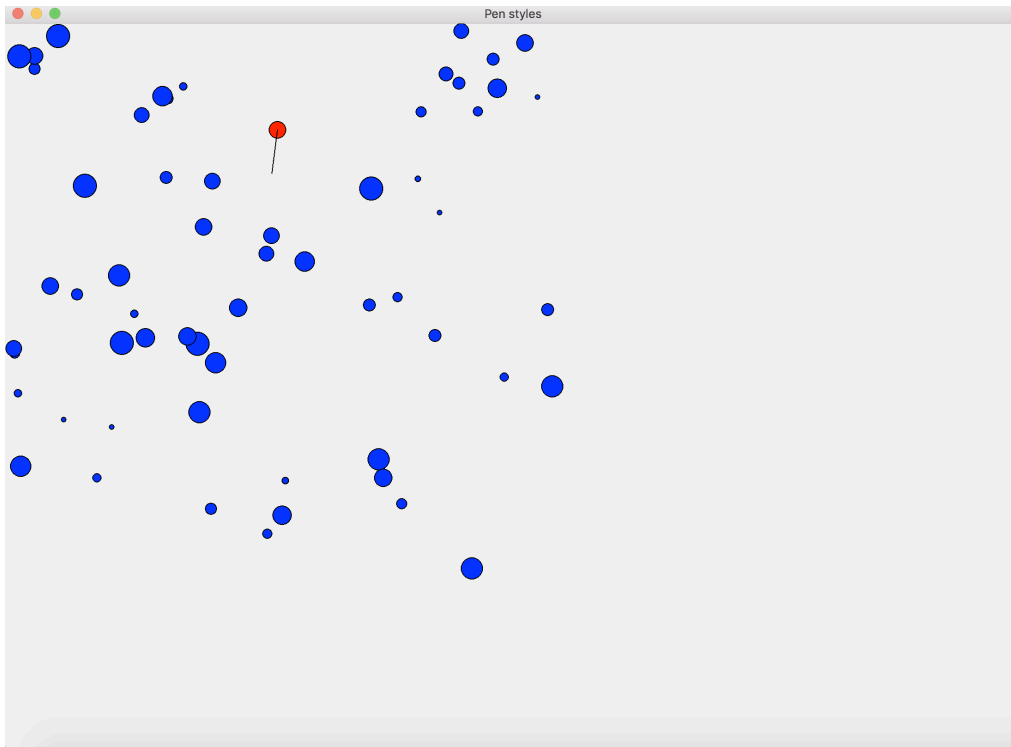
2.1 Le travail pour l'application

Nous avons fait des applications permettant de prendre des données directement depuis un csv généré aléatoirement. Chaque fois que nous avons rencontré des problèmes on demandait à notre professeur pour trouver des solutions et avoir des conseils.

The image shows a window titled "ExpSetup" with a standard macOS-style title bar (red, yellow, green buttons). Inside the window, there are five text input fields stacked vertically. The first field contains the text "numéro d'utilisateur" and is highlighted with a blue border. The second field contains "technique, 1:highlight 2:files". The third field contains "nombre_desine,default 100". The fourth field contains "nombre_taille: 6,12,18". The fifth field contains "nombre repetition". Below these fields is a large blue button with the text "rendu" in white.

L'ecran d'accueil





Les contenus

开始

插入

页面布局

公式

数据

审阅

粘贴

格式

✂

剪切

📄

复制

🔗

格式

SimSun (正文)

12

B

I

U

A1

✖

✓

CSV

```

def mouseMoveEvent(self, event):
    self.pStart = event.pos()
    self.BubbleCursor=event.pos()
    for i in range(0,len(self.list)):
        if math.sqrt(pow((self.list[i].x()+self.list[i].width()/2)-
self.BubbleCursor.x(),2)+pow((self.list[i].y()+self.list[i].width()/2)-self.BubbleCursor.y(),2))-
self.list[i].width()/2<=self.distance:
            self.distance=math.sqrt(pow((self.list[i].x()+self.list[i].width()/2)-
self.BubbleCursor.x(),2)+pow((self.list[i].y()+self.list[i].width()/2)-self.BubbleCursor.y(),2))-
self.list[i].width()/2
            self.cx=self.list[i].x()+self.list[i].width()/2
            self.cy=self.list[i].y()+self.list[i].width()/2
            self.cr=self.list[i].width()
            self.desinne=i
    for i in range(0,len(self.list)):
        ellipse = QRegion(self.list[i],QRegion.Ellipse) # défini une region elliptique
        #if ellipse.contains(event.pos()): # test la position
        if i==self.desinne:
            self.myBools[i]= True

    else:
        self.myBools[i]= False

self.update()

```

Chaque fois quand la souris bouge, on cherche la cible la plus proche du curseur.

```

def paintEvent(self, event):
    file=open('command.csv')
    self.tech=file.readline().strip('\n')

    painter = QPainter(self)
    for i in range(0,len(self.list)):
        if self.myBools[i] :
            painter.setBrush(Qt.red)
        else:
            painter.setBrush(Qt.blue)
        painter.drawEllipse(self.list[i])

    path = QPainterPath()
    if self.BubbleCursor!= None:
        if self.tech=="1":
            painter.setBrush(Qt.yellow)
            painter.drawEllipse(QRect(self.cx-self.cr/2,self.cy-self.cr/2, self.cr, self.cr))
        else:
            path.moveTo(self.BubbleCursor.x(),self.BubbleCursor.y())
            path.lineTo(self.cx,self.cy)
            ##print(self.distance)
            painter.drawPath(path)
    self.tech=None
    self.distance=10000

```

Changer le style de cercle ,desiner le trait entre le cercle et le curseur.

```

self.contenu1=self.contenu1+1
self.LineEdit1.setText("numéro d'utilisateur:"+str(self.contenu1))
# self.contenu1 = self.LineEdit1.text()
self.contenu2 = self.LineEdit2.text()
command=open("command.csv","w")
command.write(self.contenu2)
if self.LineEdit3.text()!='nombre_desine':
    self.contenu3 = int(self.LineEdit3.text())

```

```

else :
    self.contenu3 =100
self.contenu4 = self.LineEdit4.text()
self.contenu5 = self.LineEdit5.text()
self.ajoute_points()
self.cree_csv()
self.s= Example()
self.s.show()

```

Conserver les choix des utilisateurs

```

def cree_csv(self):
self.List_cs=[]
while self.n<self.contenu3:
    self.x_c=random.randint(self.List_choix[self.n][0],self.List_choix[self.n][0]+10)
    self.y_c=random.randint(self.List_choix[self.n][1],self.List_choix[self.n][1]+10)
    if(self.contenu4!="6"and self.contenu4!="12" and self.contenu4!="18"):
        self.r_c=random.randint(5,25)
    else:
        self.r_c=int(self.contenu4)
    self.circle=[self.x_c,self.y_c,self.r_c]
    self.List_cs.append(self.circle)
    self.n=self.n+1
self.n=0
csvfile=open("button.csv","w")
writer = csv.writer(csvfile)
while self.n<self.contenu3:
    writer.writerow([self.List_cs[self.n]])
    self.n=self.n+1
self.n=0

```

Créer les cibles par hasards et les conserver dans un fichier csv.

2.2 Les difficultés

Durant ce projet, nous avons rencontré des difficultés, dans tous les aspects : de la connaissance technologique, à la communication entre membre de l'équipe et avec le professeur, à la gestion du temps. Nous n'avions jamais appris le python avant ce projet, nous avons donc dû rapidement nous adapter.

Chapitre 3 Test

Pour tester notre projet, on a invité 6 personnes pour nous aider à faire des tests, et conserver les résultats dans un tableau. On soite les temps de tous les tests dans terminale.

(1)Ling Ma

Taille Dentité	Default(100)		30	60	90
Default	0.023834	100%	0.023403	0.018650	0.019599

6	0.027026	0.012244	0.019094	0.019481
12	0.021214	0.017925	0.027026	0.016208
18	0.016524	0.026664	0.012244	0.017602

(2)Yiling Wang

Taille Dentité	Default(100)		30	60	90
default	0.017962	100%	0.035623	0.016524	0.023414
6	0.013322		0.023023	0.012452	0.016065
12	0.013452		0.028949	0.010352	0.017025
18	0.023452		0.024523	0.011171	0.008910

(3) Bouita Mathias

Taille Dentité	Default(100)		30	60	90
default	0.033171	100%	0.023571	0.021214	0.017382
6	0.011171		0.012739	0.023346	0.013256
12	0.018606		0.018923	0.021345	0.019485

18	0.021171	0.018293	0.010283	0.017925
----	----------	----------	----------	----------

(4) Arthur Roland

Taille Dentité	Default(100)		30	60	90
default	0.022171	100%	0.023471	0.022930	0.012739
6	0.012171		0.022953	0.022937	0.018923
12	0.017206		0.009010	0.019302	0.018293
18	0.020071		0.012313	0.012783	0.017925

(5) zhongyi fan

Taille Dentité	Default(100)		30	60	90
default	0.019203	100%	0.021242	0.021823	0.019038
6	0.022143		0.021382	0.021923	0.019833
12	0.012098		0.010303	0.019038	0.019834
18	0.020932		0.020293	0.018293	0.015728

(6) Corwin Nolimittometal

Taille Dentité	Default(100)		30	60	90
default	0.027026	100%	0.019094	0.019481	0.020293
6	0.021214		0.017925	0.027026	0.016208
12	0.016524		0.026664	0.012244	0.017602
18	0.023834		100	0.023403	0.018650

Chapitre 4 Conclusion

Ce tp a été très enrichissant pour nous car il nous a permis de découvrir de nouvelles technologies pendant le travail et appris le concept d'IHM. C'est pas un projet facile et il n'y a pas trop de tutoriaux en ligne, on a eu besoin d'apprendre et rechercher par nous-même.

Annexe

Lien git de notre tp : <https://gitlab-etu.fil.univ-lille1.fr/sha/IHM-TP4-MA-SHA>