Rapport de travail de Programmation Concurrente

Jérémy Payet, L3 informatique

11 octobre 2017

Résumé

Un résumé du travail passé sur la programmation des fenêtres en JAVA et en Python.

1 Introduction

Au cours de notre apprentissage en programmation concurrente, nous avons eu le choix entre deux codes à réaliser :puces :

1. Faire un exercice sur les Files dans le langage Java ou Python+ un exercice sur les balles dans l'autre langage ;
2. Réaliser le programme des balles dans les 2 langues ;

J'ai décidé de réaliser le programme des Files, en Java, et celui des balles, en Python, dans le but de ne pas "répéter" la même chose.

2 Programme des Files(JAVA)

2.1 But de l'exercice

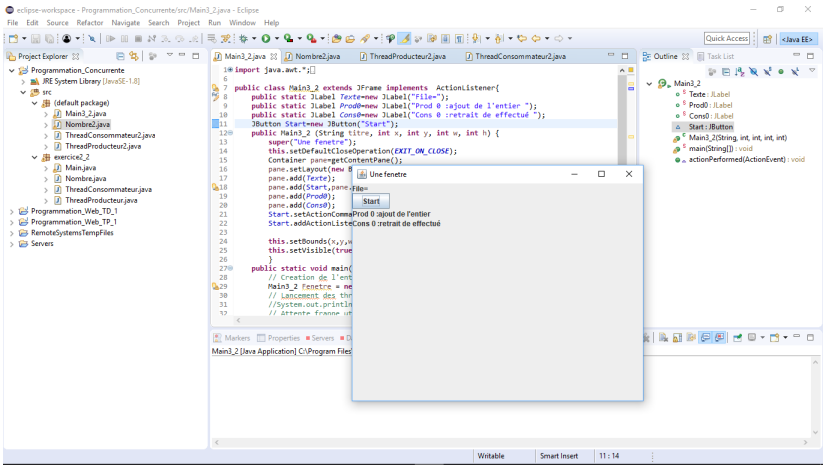
Le but du programme est d'avoir une liste de 20 nombres entiers maximum,compris entre 1 et 100, dont on ajoute ou retire des membres.Tout cela sera affiché dans une fenêtre qui donnera les éléments actuels de la file, et les actions d'ajout et de retrait d'élément.

2.2 Architecture

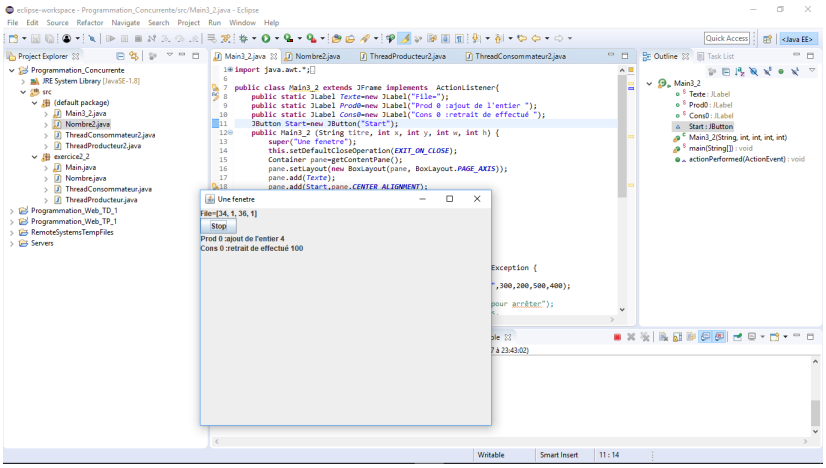
Le programme se divise en plusieurs parties :
— - Un Main qui gère l'action principal et la fenêtre.
— La fenêtre est généré par une JFrame venant de *Swing* [2] et contient des JLabel qui affichent la liste des files, et ce que le consommateur et le producteur ont fait récemment ;
— - Une classe nombre qui donnera les fonctions d'ajout et de retrait et la file d'attente(représenté en ArrayList<Integer>) ;
— - Deux Threads Producteurs et Consommateurs qui vont respectivement ajouter et retirer un nombre dans la file ;

2.3 Captures d'écran

Fenêtre avant le début :



Fenêtre durant l'exécution :



2.4 Points à détailler

Malgré le fonctionnement des fonctions d'ajout et de retrait, je n'ai pas réussi à faire en sorte que l'une des deux fonctions se lancent.Une des difficultés rencontrés a été de transformer le texte dans la fenêtre, qui a été remédié en mettant le Label en public

2.5 Morceau de code

Portion de code d'ajout d'élément(avec les variables qu'ils manipulent) :

```
private int element = 0;
public boolean affichage = true;
public static String S;
public ArrayList<Integer> tab = new ArrayList<Integer>();
public int ajout;
public int retrait;

public synchronized void ajouter() throws InterruptedException {
affichage=Math.random() < 0.5;
while (element>=20) wait();

ajout = 1+ (int) ( Math.random()*100);
```

```
//System.out.println(tab[element+1]);
afficher();
Main3_2.Prod0.setText("Prod 0 :ajout de l'entier "+ajout);
tab.add(ajout);
element=tab.size();
notifyAll();
}
```

3 Programme des balles(Python)

3.1 But de l'exercice

Le but du programme est de créer une fenêtre qui affichera des balles, dont le nombre réglé par deux boutons.En cas de collision entre 2 balles, elles disparaissent en font grimper un score, tandis qu'un chronomètre tourne sans interruption.

3.2 Architecture

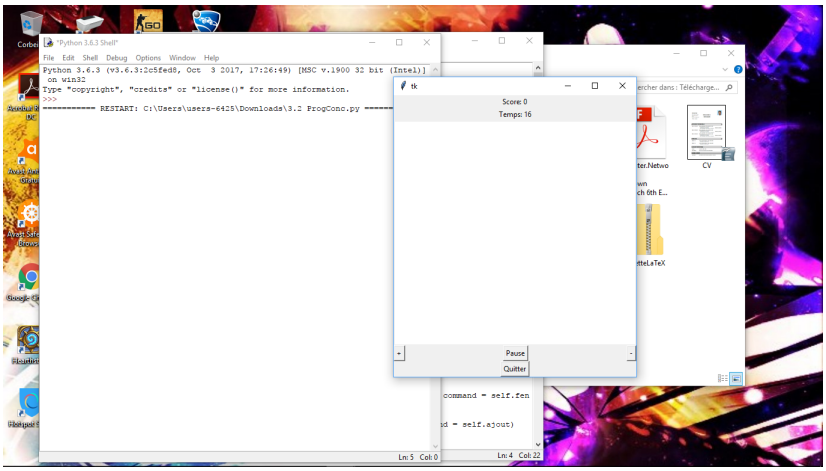
Le code possède 3 classes ayant chacune leurs rôles :

1. - Une classe main, avec un Thread en paramètre, qui définit les éléments de la fenêtre de format *Tkinter* [3] ,et deux fonction pour ajouter ou retirer une balle ;
2. - Une classe balle, avec un Object en paramètre, qui créé la balle qui sera utilisée ;
3. - Une classe calcul, avec un Thread en paramètre, qui calcule les coordonnées que la balle prend ;

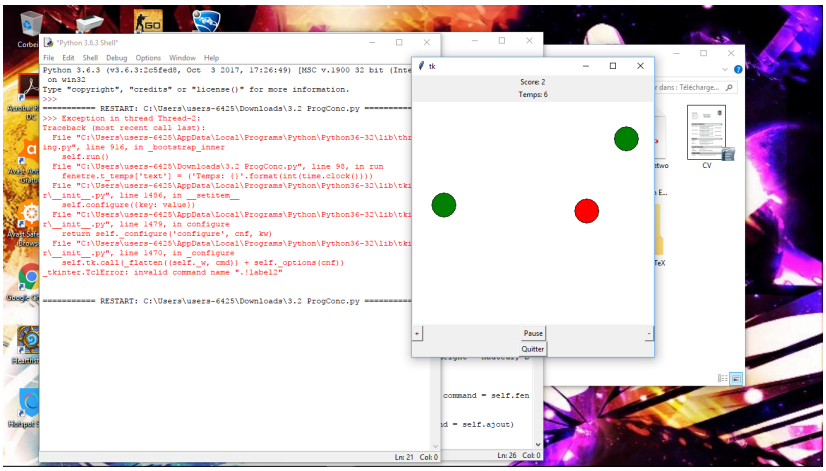
Un programme principal lancera les classes avec .start()

3.3 Captures d'écran

Fenêtre sans les balles :



Fenêtre avec l'ajout des balles :



3.4 Points à détailler

Le bouton Pause est défaillant et ne cause pas l'interruption des déplacement des balles.

3.5 Morceau de code

Portion de code d'ajout de balle :

```
def ajout(self):
      x = randint(taille, largeur-taille)
      y = randint(taille, hauteur-taille)
      couleur = choice(couleurs)
      new = self.canevas.create_oval(x, y, x+taille, y+taille, fill = couleur)
      ball(new, x, y, couleur)
```

4 Conclusions

Travailler sur les 2 programmes a été une expérience enrichissante qui m'a permis d'apprendre un peu plus sur les différentes façons de coder un programme de façon à avoir une interface dynamique, grâce à l'apprentissage en cours et par l'intermédiaire de sites tels qu'*OpenClassRoom* [1]

Références

[1] Openclassrooms. https://openclassrooms.com/courses/tout-sur-le-javascript/.

[2] Swing. https://docs.oracle.com/javase/tutorial/uiswing/index.html.

[3] Tkinter. https://docs.python.org/2/library/tkinter.html.