TP : Exercice 2 :

**Énoncé :**

**1. Exercice sur les structures de contrôle simple if-then-else à une condition**

1) Écrivez un sketch mettant en scène une balle (« balle.png ») chutant verticalement du milieu de la fenêtre et rebondissant sur le sol. À chaque rebond, la balle atteint une altitude de plus en plus basse, jusqu'à devenirimmobile (Vous diviserez par 2, ou bien par 3, la hauteur de 2 rebonds successifs).

2) Jouez un son au moment de la collision avec le sol :

a) Installez la librairie permettant de jouer du son :

- Sketch > importer une librairie > ajouter une librairie :

- Installez la librairie « Sound » de « The Processing Foundation ».

b) Importez la librairie dans votre sketch en écrivant en haut du programme :

import processing.sound.\* ;

c) Pour lire un son :

// chargez le fichier son (qui est dans data/ du répertoire du sketch)

SoundFile sound = new SoundFile(this, dataPath("bounce.mp3"));

// jouez le son

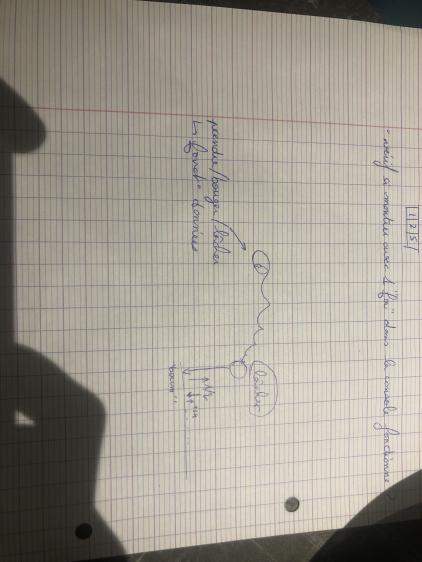
sound.play() ;

3) Ajouter la possibilité de définir la position initiale de la balle à l'aide de la souris, avant de lâcher la balle (« drag and drop »). Utilisez les fonctions «mousePressed», «mouseDragged» and «mouseReleased».

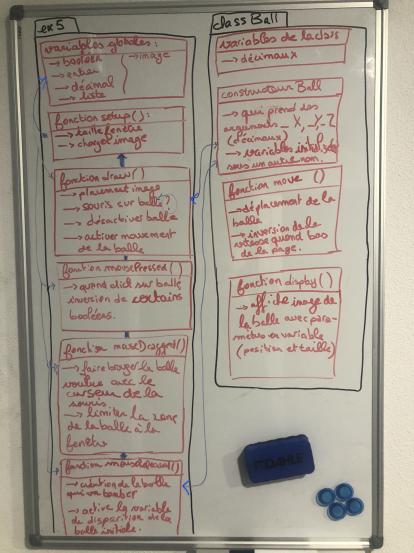
4) Ajoutez une deuxième balle (sans gestion des collisions entre balles).

5) En quoi la définition et l'utilisation d'une classe « Balle » réduit la longueur et la complexité du programme facilitant ainsi l'ajout de plusieurs balles ?

**Brouillon :**



**Diagramme de classe :**



**Exemples de ce qui se passe (graphique et console) :**

Il ne doit rien se passer dans l’interface de la console.

Dans l’interface graphique nous pouvons donc donner des exemples de ce qui se passe :

* La balle en question doit être une image et non un cercle généré par processing ;
* Si l’on maintient clique gauche sur une balle et que l’on bouge la souris, la balle doit sur le pointeur ;
* Si on lâche la souris la balle doit retomber ;
* La balle doit faire un son quand elle rentre en contact avec le sol ;
* La balle doit remonter de la moitié de la distance de descente précédente.

**Capture d’écran du code final :**

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

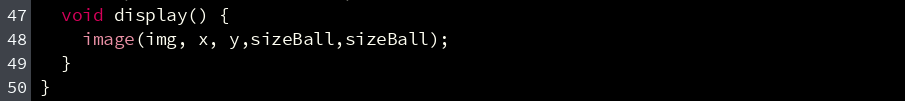
**Il y a aussi une classe Ball :**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

****

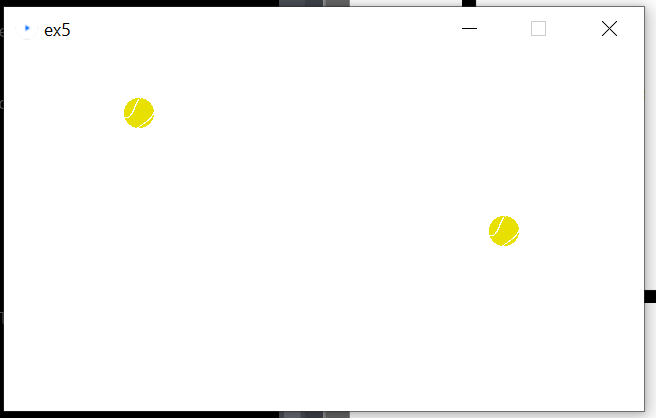
**Capture d’écran du l’exécution (interface graphique) :**

**Au départ :**

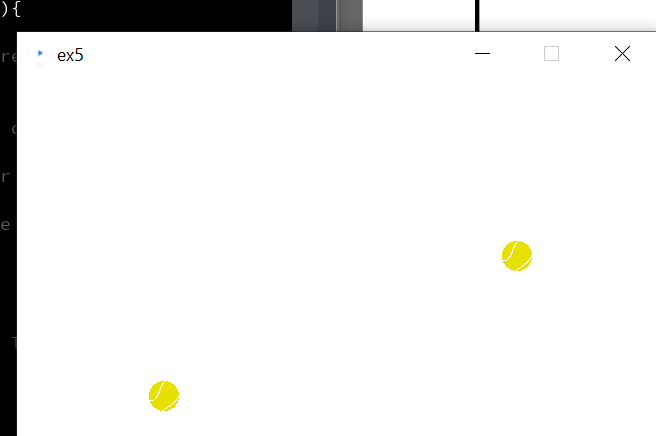
Une image contenant texte, sport athlétique, tennis, capture d’écran

Description générée automatiquement

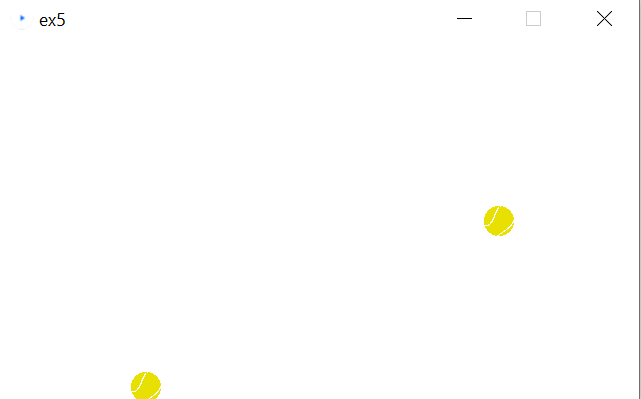
**Puis on sélectionne et fait bouger une balle :**



**La balle tombe et rebondit en faisant du son :**

****

**Et enfin la balle reste au sol (nous pouvons faire de même avec la seconde balle) :**

****

**Avis/difficultés/problèmes rencontrés/piste d’amélioration/remarques :**

* J’ai eu du mal à comprendre comment fonctionnaient les listes et surtout comment elle fonctionnait dans le cas de notre code ;
* Je n’ai pas éprouvé de problème pour que la balle est une allure réelle avec une accélération lors de la chute. Au contraire, faire remonter la balle à ½ de la distance à été plus compliqué, et j’ai dû procéder par des tests successif avec un affichage en console ;
* L’utilisation et l’application du son a aussi été complexe afin d’éviter d’une saturation lors de la stagnation de la balle au bas de l’écran, en plus du décalage de son qui apparait (il est possible qu’il reste une erreur quand les deux balles sont lâchées en même temps mais je n’ai jamais eu ce cas lors de mes tests) ;
* L’implémentation de l’image ne m’a pas posé trop de problème ;
* Cet exercice est selon moi ma plus grande réussite dans cette matière, je suis passé par de nombreux codes et j’ai dû utiliser de nombreuses ressources (votre cours ou bien de la documentation officielle Processing). J’y ai passé beaucoup de temps pour avoir un résultat qui me convient et je pense que je vais poursuivre ce code dans le futur pour y intégrer le lancement des balles par rapport à la vitesse du curseur qui la lance. Je pourrais aussi implémenter la collision de balle et éventuellement un décor à l’arrière-plan.
* Dans ce code l’intérêt d’avoir intégrer créer une class pour les balles est que ça simplifie grandement le code. Ça le réduit aussi en longueur car nous n’avons pas besoin de mettre le code de chaque balles mais une seul code pour deux balles suffit.