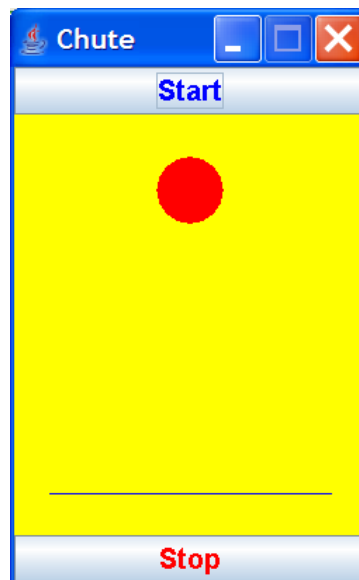


## Travaux pratiques d'informatique N° 20

Le but principal de cette séance est de vous permettre d'améliorer vos connaissances concernant la programmation graphique et la gestion des événements (surtout les événements liés à la souris et au clavier).

1. Concevoir une application graphique autonome qui affiche au coin supérieur gauche de l'écran une fenêtre **JFrame** non redimensionnable appelée "**Chute**" (comme dans la copie d'écran présentée dans la **Figure 1**. ci-dessous).



**Figure 1.**

Dans les régions "Nord" et "Sud" de cette fenêtre se trouvent deux boutons **JButton** "appelés" "**Start**" et "**Stop**", les deux noms étant écrits en bleu et, respectivement, en rouge avec une fonte **Arial** grasse de 20 pixels. Dans la région centrale de couleur jaune et de 250 x 300 pixels, on présente l'animation du mouvement vertical d'une balle rouge qui subit une chute libre suivie par un choc élastique (au contact avec le sol représenté par une ligne bleu) et un rebondissement jusqu'à la hauteur initiale et ainsi de suite.

Tout au début, la balle est figée à la hauteur maximale. Si l'utilisateur clique sur le bouton "**Start**" ou appuie sur la touche "**ENTER**" du clavier à un moment où la balle est immobile, celle-ci reprend son mouvement à partir de la hauteur courante et dans le bon sens (ascendant ou descendant).

Si l'utilisateur clique sur le bouton **"Stop"** ou appuie sur la touche **"ESCAPE"** du clavier durant le mouvement de la balle, celle-ci s'immobilise à la hauteur courante.

Afin de simplifier le problème, on peut considérer que la balle se déplace avec une vitesse constante.

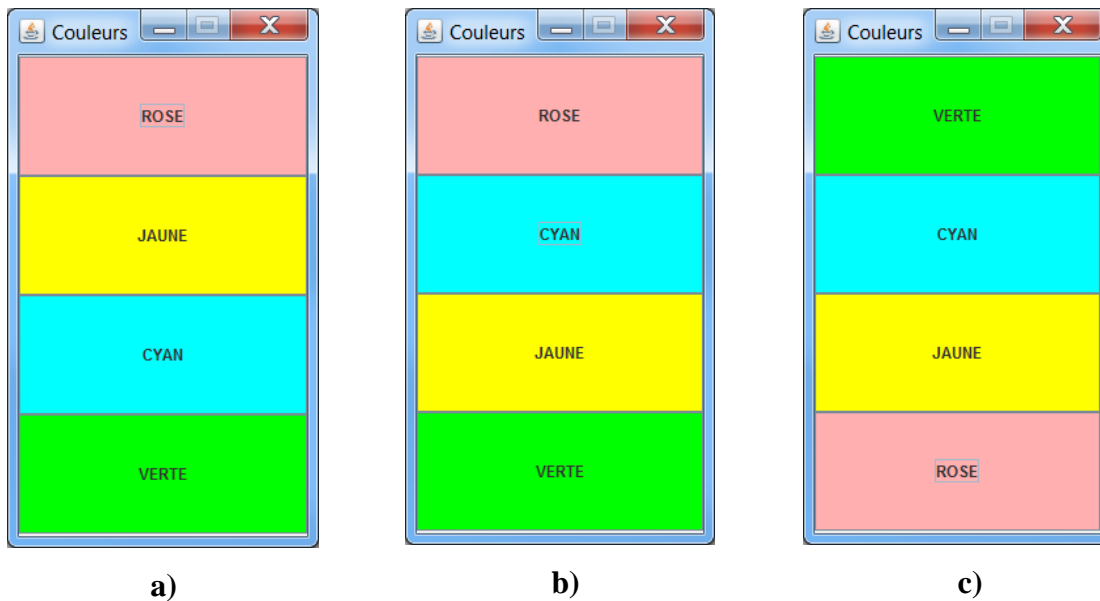
**Indications :**

- écrivez une classe graphique qui hérite de la classe **JFrame** et implémente les interfaces **ActionListener** et **KeyListener** ;
- dans cette classe englobante, écrivez une classe graphique interne qui hérite de la classe **JPanel** et redéfinit la méthode **paint** (afin de dessiner la zone centrale de la fenêtre graphique principale) ;
- dans le constructeur de la classe graphique englobante :
  - créez deux instances de la classe prédéfinie **JButton** et une instance de la classe graphique interne ;
  - créez également une instance de la classe **javax.swing.Timer** qui sera le "moteur" de l'animation ;
- dans la classe graphique englobante :
  - définissez la méthode **ActionPerformed** de sorte qu'elle redessine périodiquement et convenablement le contenu du panneau central et, le cas échéant, démarre ou arrête le moteur de l'animation ;
  - définissez la méthode **keyPressed** de sorte qu'elle démarre ou arrête le moteur de l'animation (quand l'utilisateur appuie sur les touches mentionnées ci-dessus) ;
- écrivez la classe principale qui, dans la méthode **main**, crée une instance de la classe graphique englobante.

**Remarque :** vous pouvez améliorer votre travail en ajoutant la déformation de la balle au contact avec le sol ou en tenant compte du fait que le mouvement de la balle est (rectiligne et) uniformément varié.

**2.** Ecrire un projet Java correspondant à une application autonome qui réalise l'interface graphique utilisateur (GUI) décrite ci-dessous.

Au début de l'exécution du projet, l'interface graphique présentée dans la **Figure 2.a)** doit apparaître dans le coin supérieur gauche de l'écran. Cette GUI correspond à un container de premier niveau muni de quatre boutons swing. Les textes initiaux associés aux quatre boutons correspondent aux couleurs initiales de ces boutons, à savoir : rose, jaune, cyan et verte.



**Figure 2.**

Quand l'utilisateur appuie (avec la souris) sur un des trois premiers boutons (comptés du haut vers le bas), ce bouton échange la couleur et le texte associé avec le bouton placé en-dessous de lui. Par contre, quand l'utilisateur appuie sur le quatrième bouton (placé tout en bas), ce bouton échange la couleur et le texte associé avec le premier bouton (placé tout en haut).

La **Figure 1.b**) montre l'interface graphique obtenue à partir de celle présentée dans la **Figure 1.a**) si l'on clique avec la souris sur le deuxième bouton (ayant la couleur jaune avant le clic). La **Figure 1.c**) montre l'interface graphique obtenue à partir de celle présentée dans la **Figure 1.b**) si l'on clique avec la souris sur le quatrième bouton (ayant la couleur verte avant le clic).

Suivre la démarche indiquée ci-dessous :

- créer un nouveau projet Eclipse nommé **PrTP20Exo2** ;
- ajouter à ce projet un package *cms\_tp20* ;
- ajouter à ce package 2 sources copiées "localement" depuis le bureau virtuel du CMS, à savoir les fichiers **Fenestra.java** et **CP\_TP20Exo2.java** définissant, respectivement, la classe graphique *Fenestra* et la classe "principale" **CP\_TP20Exo2** contenant le méthode *main()* ;
- ouvrir le code source de la classe *Fenestra* et compléter son contenu, en respectant les consignes précisées en commentaires ;
- exécuter le projet et tester le comportement de l'interface graphique résultante.