TP 5: modélisation et simulation.

Bernard Hugueney, Sylvain Lefebvre et Mohamed Sellami

<2017-10-20 ven.>

Contents

1 Objectifs pédagogiques

1.1 Modélisation

- Modéliser un problème et son environement programmatiquement
- Séparer la représentation du calcul

1.2 Programmation modulaire

- Importer et utiliser une librairie externe

1.3 Programmation graphique / Simulation

- Ecrire une boucle d'affichage
- Séparation affichage / modèle

2 Animation

2.1 Importation de la librairie graphique StdDraw

Un fichier JAR (Java ARchive) est un fichier ZIP utilisé pour distribuer un ensemble de **classes** (code Java compilé). Il est possible d'importer les classes définies dans un JAR pour les utiliser dans un autre programme, cela permet aux auteurs d'un programme de le distribuer sous la forme d'un unique fichier de code compilé et réutilisable. L'importation du code s'effectue grâce à l'inclusion du fichier JAR dans ce que l'on appelle le CLASSPATH du programme, c'est à dire la liste des répertoires depuis lesquels il est possible d'importer des classes.

Télécharger la libraire StdLib-package.jar depuis Moodle:

1. Copier le fichier téléchargé dans le répertoire de votre projet TP5.
2. Dans l'interface d'Eclipse, faire un clic-droit sur votre projet "TP5".
3. Dans le menu contextuel, cliquer sur "properties"
4. Dans la nouvelle fenêtre, sélectionner "Java Build Path", puis cliquer sur l'onglet "Libraries"
5. Cliquer sur le bouton "add external JARs", et aller sélectionner la librairie que vous venez de télécharger.

Il est maintenant possible d'utiliser les fonctions définies dans cette librairie.

2.2 Utilisation de la librairie: exemple

Le programme ci-dessous affiche un cercle à l'écran, dans un espace à deux dimensions.

```
1  import edu.princeton.cs.introcs.StdDraw;
2
3  public class TestMove1 {
4      public final static int X_MAX= 64;
5      public final static int Y_MAX= 24;
6      public final static float WIDTH= 0.5f;
7
8      public static void main (String [] args){
9
10         // Défini l'espace de représentation (visible à l'écran)
11         StdDraw.setXscale(-WIDTH, X_MAX+WIDTH);
12         StdDraw.setYscale(-WIDTH, Y_MAX+WIDTH);
13         for( int y= Y_MAX; y >= 0; --y){
14             for(int x= 0; x <= X_MAX; ++x){
15                 // Efface l'ecran et le colore en gris
16                 StdDraw.clear(StdDraw.GRAY);
17                 // Définit la couleur d'affichage actuelle comme rouge
18                 StdDraw.setPenColor(StdDraw.RED);
19                 // Dessine un disque de rayon WIDTH aux coordonnées x,y indiquées.
20                 StdDraw.filledCircle(x, y, WIDTH);
21                 // Affiche le dessin et pause pendant 20 ms
22                 StdDraw.show(20);
23             }
24         }
```

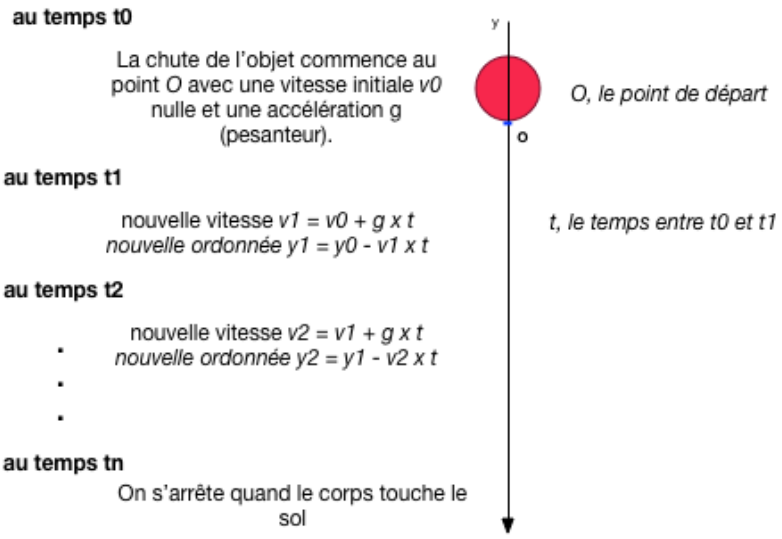
```
25      }
26  }
```

L'espace de coordonnées utilisé est défini par les fonctions StdDraw.setXscale et StdDraw.setYscale qui prennent en paramètre la borne minimale et maximale de chaque dimension. La liste et la description des fonctions disponibles dans la librairie StdDraw est disponible ici: `http://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html`

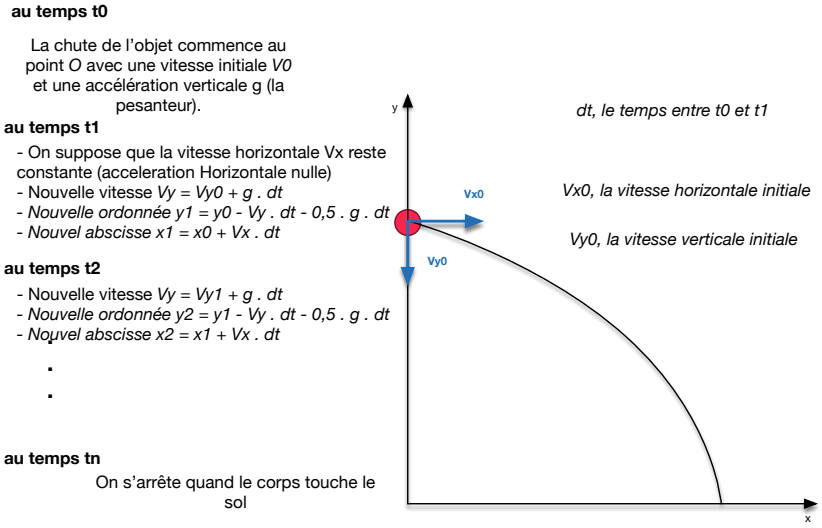
3 Simulation de trajectoire d'un corps pesant

Le but de ce tp est de calculer et dessiner la trajectoire d'une balle rebondissante dans un espace à 2 dimensions, délimité par des murs au dessus, en dessous et à droite et à gauche. Cette simulation se fera PAS A PAS, en temps discret, et donc la position de la balle à l'instant t+1 sera calculée à partir de sa position à l'instant t.

- sans vitesse initiale



- avec vitesse initiale



On rappelle la valeur de la force gravitationnelle $G = 9.81 m.s^{-2}$

3.1 Modélisation

1. Quel type de variable pouvez vous utiliser pour modéliser une balle ? plusieurs balle ?

2. Quel espace de coordonnées allez vous utiliser pour représenter la balle ?

3.2 Utilisation du modèle

1. Ecrire une classe nommée `RebondConsole` qui affiche dans la console les coordonnées de la balle à l'instant t si on la lache depuis le plafond:

   - sans vitesse initiale
   - avec vitesse initiale

2. Prendre en compte :

   - les rebonds
   - le frottement

3.3 Affichage de la balle

En vous basant sur l'exemple ci dessus, écrire un programme qui affiche une balle rebondissante à l'ecran. Pensez à initialiser la grille (avec les méthodes `StdDraw.setXscale` et `StdDraw.setYscale`).

1. Ecrire une classe `RebondAnim` qui a une fonction `main` qui permet d'afficher et d'animer une balle avec la librairie StdDraw.

2. Ecrire une classe `RebondMulti` qui a fonction `main` qui permet d'afficher et d'animer plusieurs balles de couleurs différentes.

3.3.1 Bonus

Gérez les collisions entres les différentes balles.

3.4 Rendu attendu

Déposer sur moodle une archive zippée contenant:

- Un compte rendu au format word décrivant l'algorithme utilisé pour simuler une balle, et celui utilisé pour simuler plusieurs balles.
- Le code java contenant les 3 classes demandées dans les questions précédente avec chacune leur fonction `main`
