

IFT1025 Été 2010 — Devoir 2

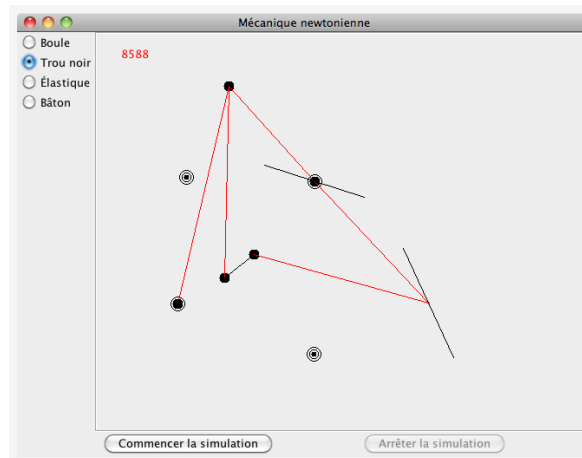
Miklós Csűrös

11 juin 2010

À remettre avant 19 :59 lundi le 28 juin par courriel à csuros@iro.umontreal. . .
Ce TP est destiné à des équipes de 2 ou 3 étudiant/es.

1 Mécanique newtonienne

Dans ce TP, vous avez à implanter une simulation pour modéliser le mouvement d'objets physiques dans la présence de certaines forces. Vous devrez implanter en particulier le mouvement de boules liées par des élastiques. Le logiciel permettra de placer des boules et des élastiques sur un bord à l'aide d'un interface graphique, ainsi que de lancer et d'arrêter la simulation de mouvements des boules selon les lois physiques. Les classes de ce TP doivent être placées dans un package nommé `mechanique`.



L'interface graphique de l'application. Les élastiques tendus sont affichés en rouge, les autres en noir. L'image montre aussi les éléments prédéfinis : trou noir et bâton.

1.1 Modélisation d'un système physique

Le système physique comprend des objets physiques et des sources de forces capturés par les interfaces `PhysicalObject` et `Force`. Un objet physique est

caractérisé par sa position et sa vitesse (en deux dimensions), ainsi que par sa masse. L'ensemble du système est stocké par une instance de `PhysicalSystem`. Cette classe calcule les forces sur chaque objet et les déplacements pendant une intervalle (courte) de temps δ . Le déplacement des objets se fait selon les règles de la mécanique Newtonienne : l'accélération est calculée à partir des forces exercées sur chaque objet.

1.2 Interface graphique

Le système est affiché par un panneau de la classe `SystemDisplay` (sous-classe de `JPanel`). Le panneau est placé au milieu d'un `JFrame` avec des boutons en bas pour lancer/arrêter la simulation, et des boutons à côté (`JRadioButton`) pour le placement des éléments du système. Un élément est affiché s'il implante l'interface `Drawable` avec la seule méthode `draw(Graphics2d g)`.

Le placement des éléments se fait à l'aide des implantations de l'interface `ObjectPlacement`. Cette interface est un écouteur d'événements de souris. Chaque type d'élément (boule, élastique, etc.) a son propre bouton radio — quand ce bouton est sélectionné, tous les événements de souris sont relayés à l'instance de `ObjectPlacement` appropriée.

La simulation est lancée ou arrêtée par les boutons en bas du `JFrame` de l'application. Pour performer la simulation, on utilise un `Timer`. À chaque `ActionEvent` généré par le `Timer`, il faut recalculer les positions des boules et reafficher le bord (`repaint()`). Le `Timer` utilise une petite intervalle δ (10 millisecondes) : une action est générée à de telles intervalles, et les déplacements sont calculés avec le même δ (voir méthode `computeNewPositions(double d)` dans la classe `PhysicalSystem` : attention, l'argument `d` doit être en seconds).

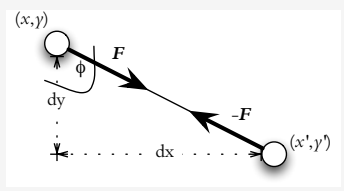
1.3 Modélisation des boules et des élastiques

Vous devez implanter des classes pour représenter des boules (`PhysicalObject`) et des élastiques (`Force`). Les classes implantent aussi l'interface `Drawable`, et doivent être accompagnées par des implantations de l'interface `ObjectPlacement`.

Boule. Une boule est juste un objet physique de masse 1. Elle est affichée comme un cercle rempli de diamètre 10 pixels.

Élastique. Un élastique est une connexion entre deux objets physiques («gauche» et «droit»). Les attributs d'un tel objet sont la longueur neutre ℓ , la rigidité α , et les deux objets de type `PhysicalObject`. La classe pour élastiques implante l'interface `Force` qui déclare la méthode `getAcceleration(PhysicalObject o)`.

(Les rigeurs et les longueurs neutres sont les mêmes pour tous les élastiques : un bon choix pour les constantes est $\ell = 100$, $\alpha = 0.1$.)



The diagram shows two spheres. The left sphere is at coordinates (x, y) and the right sphere is at (x', y') . A force vector \mathbf{F} points from the left sphere towards the right sphere. The displacement components are labeled as dx (horizontal) and dy (vertical). The angle between the force vector and the horizontal axis is ϕ .

La vecteur de force \mathbf{F} est déterminée par la longueur de l'élastique, c'est à dire, la distance entre les deux boules.

$$\|\mathbf{F}\| = \begin{cases} 0 & \text{si } d \leq \ell; \\ \alpha(d - \ell) & \text{si } d > \ell \end{cases},$$

où $d = \sqrt{(x - x')^2 + (y - y')^2}$ et (α, ℓ) caractérisent l'élastique : rigeur et longueur neutre. Pour calculer les coordonnées de la force, calculez d'abord l'angle ϕ (`Math.atan2` avec $dx = x' - x$ et $dy = y' - y$) : $\mathbf{F} = \|\mathbf{F}\| \cdot (\cos \phi, \sin \phi)$.

1.4 Placement de boules et élastiques

L'affichage du bord est par la classe `SystemDisplay` qui est une extension de `JPanel`. On peut placer une boule par le clic de souris sur une partie vide du bord. Pour placer un élastique, il faut sélectionner (par un clic) d'abord la première boule («boule gauche»), ensuite la deuxième boule («boule droite»). Quand une boule est sélectionnée («boule gauche» d'un élastique), il faut afficher une ligne droite entre la boule sélectionnée et le pointeur du souris. Cette ligne est affichée (et suit le mouvement du souris) jusqu'à ce que l'utilisateur choisit la deuxième boule («boule droite») par un clic : elle peut être une boule déjà placée ou une nouvelle boule (clic sur une partie vide). Les deux boules définissent un nouvel élastique à placer sur le bord. L'utilisateur peut annuler le placement d'un élastique par le clic du bouton droit (v. `SwingUtilities.isRightMouseButton`). Utilisez `closestObject` de `PhysicalSystem` pour la sélection d'objets : un clic sélectionne l'objet, si cela se produit dans le rectangle défini par `getWidth` et `getHeight` de l'objet le plus proche.

Le logiciel utilise l'interface `ObjectPlacement` pour le placement d'objets.

1.5 Application

L'application est lancée par la classe `SystemDisplay` :

```
% java -cp Mecanique.jar mecanique.SystemDisplay
```

2 Travail et remise

Une grande partie du code source vous êtes fournie. À l'addition des interfaces, `PhysicalObject`, et `SystemDisplay`, vous trouverez aussi le code pour les objets de type «trou noir» (une `Force`) et «bâton» (un `PhysicalObject`). Le trou noir (classe `BlackHole`) illustre le calcul d'accélération (dans le cas d'un élastique, il faut vérifier si l'argument de **getAcceleration** est l'un des objets gauche ou droit). Le bâton (class `Stick`) illustre le placement d'un bâton par un principe similaire que celui d'un élastique (deux clics pour le placement des bouts du bâton).

Vous devez implanter des classes pour la représentation et placement des élastiques et des boules. Notez qu'il suffit de modifier la méthode **getTypes** de `SystemDisplay` pour incorporer vos classes dans l'applicaton. En particulier, vous devez développer le code pour quatre classes :

- ▶ «boule» : implémente les interfaces `Drawable` et `PhysicalObject`
- ▶ «placement de boule» : implémente l'interface `ObjectPlacement`
- ▶ «élastique» : implémente les interfaces `Drawable` et `Force`
- ▶ «placement d'élastique» : implémente l'interface `ObjectPlacement`

Ne modifiez pas le code fourni sauf dans la méthode **getTypes** de `SystemDisplay` où vous ajoutez vos classes de l'interface `ObjectPlacement`.

Compilez un archive jar nommé `Mecanique.jar` qui contient toutes les classes du package `mecanique`. L'archive doit inclure les sources `.java` et les classes compilées `.class`. Envoyez le fichier comme attachement dans un courriel à `csuros@iro.umontreal...` Votre programme doit être fonctionnel, clair, commenté et original (plagiat=0).

3 Boni

Vous pouvez avoir jusqu'à 15% de boni pour tout autre type d'élément que vous implantez (45% au plus). La créativité de types additionnels fait partie de l'évaluation. Exemples :

- * boules avec charge électrique (positif et négatif) : force de Coulomb
- * molécules avec interaction (force genre van der Waals) : force de grandeur $\|\mathbf{F}\| = \frac{\lambda}{d^t} - \frac{\mu}{d^s}$ avec, disons $t = 5$ et $s = 2$ où d est la distance entre molécules (repulsion quand d est petit et attraction quand d est plus grand ; force négligeable à grandes distances).
- * un ressort : $\|\mathbf{F}\| = \alpha|d - \ell|$ avec repulsion quand $d < \ell$.

Expliquez vos éléments dans un fichier appelé `LisezMoi.pdf` ou `LisezMoi.txt` inclut dans l'archive jar.