Le projectile [si02] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel algoprog Université Haute-Alsace Version 22 mai 2018

Table des matières

Java - Le projectile (Solution)

Mots-Clés Simulation ■
Requis Axiomatique impérative (sauf Fichiers) ■
Optionnel Graphique ■
Difficulté • ○ ○ (30 min) ■

Objectif
Cet exercice détermine puis trace la trajectoire d'un boulet de canon par simulation physique.

1 Le projectile / pgprojectile

1.1 Distance de tir (distanceTir, saisirBoulet)

Distance de tir

Un boulet de canon qui sort avec une vitesse v_0 et une inclinaison d'angle α par rapport à l'horizontale atterrit plus loin (sur terrain plat) à une distance :

$$dist = v_0^2 \sin(2\alpha)/g$$

où $g = 9.80665$ m/s^2 est la constante gravitationnelle.

Définissez la constante `GTERRE=9.81` (accélération de la pesanteur).

Définissez le type `Boulet`, structure contenant la vitesse `v0` (réel) et l'angle d'inclinaison `alpha` (réel) initiaux d'un boulet.

Validez vos définitions avec la solution.

Solution Java

```java
/**
  Accélération de la pesanteur
*/
final static double GTERRE = 9.81;

/**
  Représente un Boulet
*/
public static class Boulet
{
  /// vitesse initiale
  public double v0;
  /// angle d'inclinaison en radians
  public double alpha;
}
```

Écrivez une fonction `distanteTir(b)` qui calcule et renvoie la distance du tir d'un `Boulet b` (voir formule ci-dessus).

Outil Java

L'opération $\sin x$ s'écrit `Math.sin(x)`.

Solution Paramètres

Entrants : Un `Boulet b`

Résultat de la fonction : Un réel

Validez votre fonction avec la solution.

Solution Java

```
/**
  Distance d'un tir
  @param[in] b - un Boulet
  @return la distance du tir de b
*/

public static double distanceTir(final Boulet b)
{
  return b.v0 * b.v0 * Math.sin(2.0 * b.alpha) / GTERRE;
}
```

Écrivez une procédure `saisirBoulet(b)` qui saisit les caractéristiques d'un `Boulet` dans `b`. Affichez l'invite :

```
Angle et vitesse du boulet?
```

Écrivez une procédure `afficherBoulet(b)` qui affiche les caractéristiques (angle et vitesse) d'un `Boulet b`.

Validez vos procédures avec la solution.

Solution Java

```
/**
  Saisie les caractéristiques d'un Boulet
  @param[out] b - un Boulet
*/

public static void saisirBoulet(Boulet b)
{
  Scanner input = new Scanner(System.in);
  input.useLocale(Locale.US);
  System.out.print("Angle et vitesse du boulet? ");
  b.alpha = input.nextDouble();
  b.v0 = input.nextDouble();
}
```

```
/**
  Affiche les données d'un Boulet
  @param[in] b - un Boulet
*/

public static void afficherBoulet(final Boulet b)
{
  System.out.println("==> angle et vitesse du boulet " + b.alpha + " " + b.v0);
}
```

1.2 Mouvement d'un projectile

Un projectile de position $p(t) = \begin{pmatrix} x \\ y \end{pmatrix}$ de vitesse $v(t) = \begin{pmatrix} v_x(t) \\ v_y(t) \end{pmatrix}$ soumis à une accélération $a(t) = \begin{pmatrix} a_x(t) \\ a_y(t) \end{pmatrix}$ pendant un temps δt a pour nouvelle vitesse :

$$v(t + \delta t) = v(t) + a \cdot \delta t$$

et nouvelle position :

$$p(t + \delta t) = p(t) + v(t + \delta t) \cdot \delta t$$

Écrivez une procédure `bougerProjectile(x,y,vx,vy,ax,ay,dt)` qui calcule la **nouvelle** position dans `(x,y)` et la **nouvelle** vitesse dans `(vx,vy)` d'un projectile d'accélération `(ax,ay)` à l'instant `dt` suivant. Tous les paramètres sont des réels.

Écrivez une fonction `distanceSimulee(b,dt)` qui calcule et renvoie la distance du tir d'un projectile lancé avec un `Boulet b` soumis à la force de pesanteur (constante `GTERRE`) de valeur 9.81, le delta temps étant `dt` (réel).

Aide méthodologique

Placez le projectile en $(0, 0)$ de vitesse initiale $(v0 \cos \alpha, v0 \sin \alpha)$, puis bougez le projectile avec l'accélération $(0, -g)$ jusqu'à ce que $y \leq 0$, enfin retournez x.

Outil Java

Les opérations $\sin x$ et $\cos x$ sont définies par `Math.sin(x)` et `Math.cos(x)`.

Validez votre procédure et votre fonction avec la solution.

Solution Java @[pgprojectile.java]

```
/**
  Bouge un projectile
  @param[in,out] x - position x
  @param[in,out] y - position y
  @param[in,out] vx - vitesse x
  @param[in,out] vy - vitesse y
  @param[in] ax - accéleration x
  @param[in] ay - accéleration y
  @param[in] dt - delta temps
*/

public static void bougerProjectile(double[] x, double[] y, double[] vx, double[] vy,
    double ax, double ay, double dt)
```

```
{
  vx[0] += ax * dt;
  vy[0] += ay * dt;
  x[0] += vx[0] * dt;
  y[0] += vy[0] * dt;
}
```

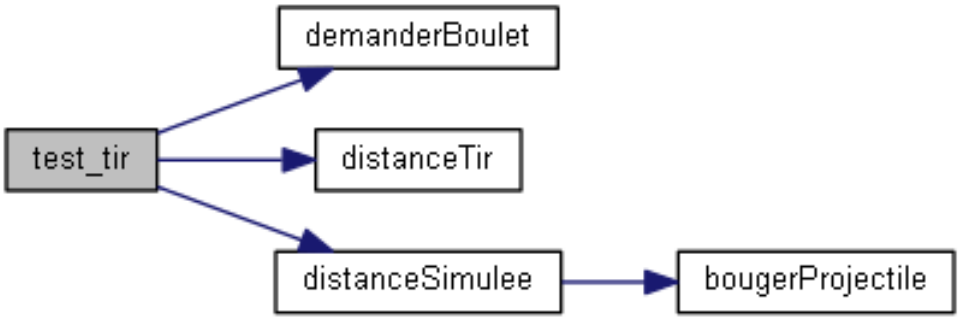
```
/**
  Distance simulée d'un tir
  @param[in] b - un Boulet
  @param[in] dt - delta temps
  @return la distance d'un tir de vitesse v0 d'angle alpha de delta dt
*/

public static double distanceSimulee(final Boulet b, double dt)
{
  double[] x = new double[1], y = new double[1];
  x[0] = 0.0;
  y[0] = 0.0;
  double[] vx = new double[1], vy = new double[1];
  vx[0] = b.v0 * Math.cos(b.alpha);
  vy[0] = b.v0 * Math.sin(b.alpha);

  boolean btir = true;
  while (btir)
  {
    bougerProjectile(x,y,vx,vy,0.0,-GTERRE,dt);
    btir = (y[0] > 0.0);
  }
  return x[0];
}
```

1.3 Test de tir

Écrivez une procédure `test_tir` qui saisit les caractéristiques de tir d'un boulet (angle et vitesse) puis calcule et affiche la distance exacte et les distances simulées à 0.1 et 0.01 du tir.



Testez. Exemple d'exécution :

```
angle et vitesse du boulet? 0.6 50
==> Distance exacte = 237.5227028542
==> Distance simulee 0.1  = 235.2206502465
```

```
==> Distance simulee 0.01 = 237.2839892837
```

Validez votre procédure avec la solution.

Solution Java @[pgprojectile.java]

```java
public static void main(String[] args)
{
  Boulet b = new Boulet();
  saisirBoulet(b);
  System.out.println("==> Distance exacte = " + distanceTir(b));
  System.out.println("==> Distance simulee 0.1  = " + distanceSimulee(b,0.1));
  System.out.println("==> Distance simulee 0.01 = " + distanceSimulee(b,0.01));
}
```

On désire saisir l'angle en degrés.
Que faut-il modifier dans le programme principal ?

Solution simple
On écrit une fonction `degVersRad(alpha)` qui donne l'équivalent en radians d'un angle `alpha` en degrés puis, après la saisie (en degrés), on calcule son équivalent en radians avant de lancer les calculs.

2 Références générales

Comprend [Boudreault-CC1 :c2 :td6] ■