

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 159

Devoir #1 — Automne 2009

Programmation

Devoir à remettre au plus tard le vendredi 18 septembre 2009 à 23h59. La remise devra être faite grâce à la commande *turnin* (voir page 6 pour plus de détails).

Le but de ce travail consiste à implanter la solution d'un problème dont vous recevez la spécification, l'analyse et la conception. L'objectif est de vous familiariser, au niveau programmation, avec la notion de séquence. **Vous devez obligatoirement implanter la conception fournie en annexe en utilisant seulement les notions de programmation vues dans le cours.**

Vous devez me remettre votre programme, sa documentation, trois fichiers de tests ainsi que le formulaire d'estimation de l'effort. Votre programme devra respecter les normes de programmation du département. Les fichiers de tests devront contenir des tests significatifs. La documentation se fera sous forme de commentaires dans le programme en utilisant les normes de *doxygen*.

Sujet : la balistique

Vous devez faire l'analyse, la conception et l'implantation d'un programme qui servira à calculer l'altitude atteinte par un projectile, la distance horizontale qu'il parcourt ainsi que le temps qu'il met à retomber. Le projectile est tiré par un lanceur (personne, fusil, canon, ...) et n'est pas autopropulsé. Nous ne considérons pas dans le calcul la force de friction ni les variations dans la force d'attraction. Comme nous simplifions le problème, nous allons induire une erreur dans notre calcul. Le programme devra donc corriger ses résultats en y additionnant une valeur correspondant à une certaine marge d'erreur spécifiée comme un pourcentage des résultats. Cette marge d'erreur est spécifiée par l'utilisateur.

Le programme prendra en entrée les vitesses verticales et horizontales du projectile (en milles/heure), la force d'attraction (en mètres/seconde²) ainsi que le pourcentage estimée de l'erreur. Le programme devra afficher comme résultats l'altitude et la distance en mètres, ainsi que le temps en années, jours, heures, minutes et secondes.

Afin de pouvoir produire les résultats demandés, voici quelques notions simplifiées de balistique.

Lorsqu'un projectile est lancé, il reçoit initialement une vitesse verticale V_0 ainsi qu'une vitesse horizontale H_0 comme le montre la figure 1. Sur cette figure, A est l'altitude maximale atteinte par le projectile et D est la distance horizontale parcourue par le projectile.

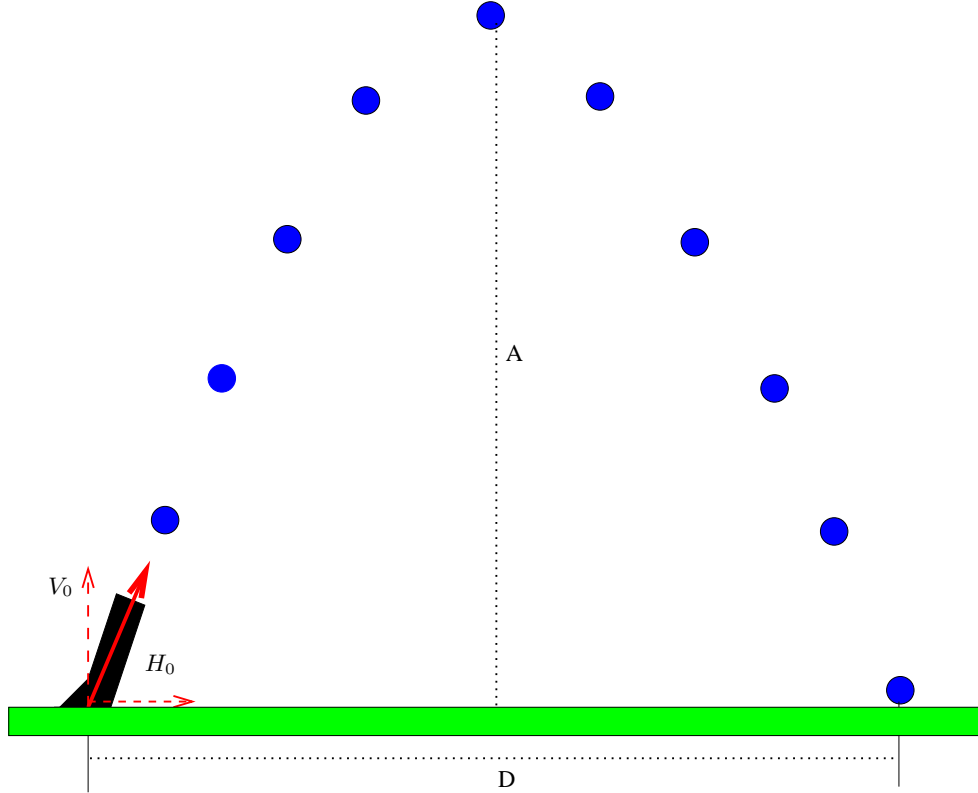


FIG. 1 – Notions de balistique

Sur le plan vertical, le poids est la seule force qui s'exerce sur l'objet. Comme le poids de l'objet est constant, on peut se servir de la seconde loi de Newton pour résoudre le mouvement vertical :

$$-W = F = m \times a = m \times dV_t/dt \quad (1)$$

où W est le poids, m la masse, V_t la vitesse verticale au temps t , t le temps, a l'accélération et F la force externe globale. Comme la direction de la vitesse initiale est vers le haut, le poids est négatif car il exerce une force vers le bas sur le projectile. Si on résout l'équation on obtient :

$$dV_t/dt = -W/m = -g \quad (2)$$

$$V_t = V_0 - gt \quad (3)$$

où g est la force gravitationnelle, V_0 est la vitesse initiale et V_t la vitesse au temps t .
En intégrant l'équation de la vitesse on obtient la position à tout moment :

$$dy/dt = V_t = V_0 - gt \quad (4)$$

$$y = V_0 t - 0,5gt^2 \quad (5)$$

où y est l'altitude du projectile au temps t .

Afin de calculer l'altitude maximale A atteinte par le projectile, il faut d'abord connaître le temps pris par le projectile pour atteindre cette altitude. Pour ce faire on reprend l'équation $V_t = V_0 - gt$ en supposant $V = 0$ (la vitesse verticale devient nulle lorsque le projectile atteint le sommet de sa trajectoire). En isolant t , on obtient $t = V_0/g$.

Un fois que la durée t de la montée du projectile est trouvée, on obtient l'altitude en remplaçant t par V_0/g dans l'équation de l'altitude (équation 5). Nous obtenons donc :

$$y = V_0(V_0/g) - 0,5g(V_0/g)^2 \quad (6)$$

$$y = \frac{V_0^2}{2g} \quad (7)$$

Enfin, pour connaître le temps t_r que prend le projectile pour revenir au sol, on utilise la formule de l'altitude (équation 5) en supposant une altitude $y = 0$:

$$V_0 t = \frac{gt^2}{2} \quad (8)$$

$$t_r = \frac{2V_0}{g} \quad (9)$$

Il est intéressant de remarquer que la vitesse verticale d'impact est la même que la vitesse verticale de départ V_0 et que la durée totale du vol du projectile est le double du temps mis pour atteindre l'altitude maximale.

Finalement, la distance horizontale se calcule en utilisant le temps de retour au sol t_r . La formule est :

$$D = H_0 \times t_r. \quad (10)$$

Informations complémentaires :

- Pour plus d'information sur la balistique, vous pouvez visiter le site suivant :

[http ://www.grc.nasa.gov/WWW/K-12/airplane/ballflight.html](http://www.grc.nasa.gov/WWW/K-12/airplane/ballflight.html)

- Rappel sur les forces gravitationnelles exercées par différentes planètes ou lunes :

Soleil	→ 274,1 m/s^2 (27,9 fois la terre)	Jupiter	→ 25,93 m/s^2 (2,64 fois la terre)
Mercure	→ 3,703 m/s^2 (0,38 fois la terre)	Saturne	→ 11,19 m/s^2 (1,14 fois la terre)
Venus	→ 8.872 m/s^2 (0,903 fois la terre)	Uranus	→ 9,01 m/s^2 (0,92 fois la terre)
Terre	→ 9,8226 m/s^2	Neptune	→ 11,28 m/s^2 (1,148 fois la terre)
Lune	→ 1,625 m/s^2 (0,166 fois la terre)	Pluton	→ 0,610 m/s^2 (0,062 fois la terre)
Mars	→ 3,728 m/s^2 (0,39 fois la terre)		

- On rappelle l'équivalence suivante : 1 kilomètre = 0,62 mille.

1 Analyse

Entrées :

1. (clavier) Vitesse verticale (réel)
2. (clavier) Vitesse horizontale (réel)
3. (clavier) Force d'attraction (réel)
4. (clavier) Erreur en pourcentage (réel)

Sorties

1. (écran) Altitude en mètres (réel)
2. (écran) Distance horizontale (réel)
3. (écran) Temps en mois, jour, heure, minute, seconde (5 entiers)

Constantes :

1. Nombre mille dans un kilomètre = .62
2. Nombre de mètres dans un kilomètre = 1000
3. Nombre de secondes dans une minute = 60
4. Nombre de minutes dans une heure = 60
5. Nombre d'heures dans une journée = 24
6. Nombre de journée dans une année = 364

Formules (Relation entrées/sorties) :

1. Conversion de milles à Kilomètres = nombre de milles ÷ nombre de mille dans kilomètre
2. Conversion de kilomètres à mètres = nombre de kilomètres × nombre de mètres dans un kilomètre
3. Conversion de mètres par heure en m/sec = metre / nbr sec ds min * nbr min ds heure
4. Temps pour retombée = 2 × Vitesse verticale / force gravitationnelle
5. Altitude max = .5 × Vitesse verticale² / force gravitationnelle
6. Distance parcourue = temps pour retombée × vitesse horizontale
7. Résultat corrigé = résultat (temps/altitude/distance) + (résultat × erreur)
8. On trouve les années, jours, heures, minutes, secondes (unités) avec des opérations de division entière (div) et modulo
 - (a) temps unité = temps restant **div** nb de secondes dans unité
 - (b) temps restant = temps restant **modulo** nb de secondes dans unité

2 Conception

2.1 Algorithme

- 1 Lire les entrées
 - 1.1 Lire la vitesse verticale V_0 (réel)
 - 1.2 Lire la vitesse horizontale H_0 (réel)
 - 1.3 Lire force d'attraction g (réel)
 - 1.4 Lire le pourcentage d'erreur e (réel)
- 2 Convertir les vitesse en mètres par secondes
 - 2.1 Transformer vitesse verticale de milles/heure en kilomètres/heure (formule 1)
 - 2.2 Transformer vitesse horizontale de milles/heure en kilomètres/heure (formule 1)
 - 2.3 Transformer vitesse verticale en de kilomètres en mètres (formule 2)
 - 2.4 Transformer vitesse horizontale de kilomètres en mètres (formule 2)
 - 2.5 Transformer vitesse verticale en de mètres/heure en mètres/seconde (formule 3)
 - 2.6 Transformer vitesse horizontale de mètres/heure en mètres/seconde (formule 3)
- 3 Calculer altitude maximale
 - 3.1 Calculer l'altitude (formule 5)
 - 3.2 Ajouter l'erreur (formule 7)
- 4 Calculer temps retombée
 - 4.1 Calculer le temps (formule 4)
 - 4.2 Ajouter l'erreur (formule 7)
- 5 Calculer distance parcourue (formule 6)
- 6 Convertir le temps en années-jours-heures-minutes-secondes
 - 6.1 Calculer le nombre d'années (formule 8a - unité = année)
 - 6.2 Calculer le nombre de secondes restantes (formule 8b - unité = année)
 - 6.3 Calculer le nombre de jours (formule 8a - unité = jour)
 - 6.4 Calculer le nombre de secondes restantes (formule 8b - unité = jour)
 - 6.5 Calculer le nombre d'heures (formule 8a - unité = heure)
 - 6.6 Calculer le nombre de secondes restantes (formule 8b - unité = heure)
 - 6.7 Calculer le nombre de minutes (formule 8a - unité = minute)
 - 6.8 Calculer le nombre de secondes restantes (formule 8b - unité = minute)
- 7 Afficher la distance sur l'écran
 - 7.1 Afficher l'altitude
 - 7.2 Afficher la distance
 - 7.3 Afficher le nombre d'années.
 - 7.4 Afficher le nombre de jours.
 - 7.5 Afficher le nombre d'heures.
 - 7.6 Afficher le nombre de minutes.
 - 7.7 Afficher le nombre de secondes.

3 Soumission

Vous devez me soumettre avant la date limite, à l'aide de la commande

turnin -cift159 -ptp1 tp1.cpp test1 test2 test3 effort.txt

- un fichier portant **OBLIGATOIREMENT** le nom *tp1.cpp* qui contient le programme demandé
- trois fichiers tests portant **OBLIGATOIREMENT** les noms test1, test2 et test3. Les trois tests soumis doivent être significatifs, i.e. qu'ils permettent de tester le plus de cas possibles pour votre application.
- un fichier portant **OBLIGATOIREMENT** le nom **effort.txt** qui est le formulaire d'estimation de l'effort que vous pouvez vous procurer sur le site web du cours. Pour générer un fichier texte, vous n'avez qu'à utiliser le fichier en format chiffrier (.xls) et à copier/coller les données dans un fichier texte (.txt) par l'intermédiaire d'un éditeur. Vous pouvez aussi le sauver en format « .csv » et le renommer en format « .txt ».

Vous pouvez vérifier votre soumission grâce à la commande

turnin -cift159 -ptp1 -v

Étant donné que la correction est partiellement automatisée, une mauvaise soumission entraîne une mauvaise note (possiblement 0). Il n'y a pas de recorection.

Avant de soumettre vos travaux, faites exécuter votre programme dans les mêmes conditions que celles du train de travaux. Pour cela mettre vos données dans un fichier (par exemple de nom donnees.tp1) et vos résultats dans un autre fichier (par exemple de nom resultats.tp1) et faire les commandes :

(clavier)

```
> avec un éditeur de texte (nedit, textedit, ...), créer le fichier donnees.tp1
> g++ tp1.cc -o tp1
> tp1 < donnees.tp1 > resultats.tp1
> less resultats.tp1
```

4 Documentation

La documentation technique sera générée à l'aide de l'outil doxygen et du fichier de configuration présent dans le répertoire du cours /home/public/cours/ift159/ift159.cfg. Celle-ci sera générée automatiquement durant la correction. Pour pouvoir évaluer la documentation que vous créez, vous pouvez soit utiliser l'outil soumettre qui permet générer la documentation qui sera remise, ou bien utiliser la commande

doxygen /home/public/cours/ift159/ift159.cfg

sur le serveur rigel qui fera la même chose. Vous pouvez aussi utiliser l'outil (qui est installé sur les ordinateurs du D4-1017) : doxywizard. Cet outil peut être installé chez vous. Il est compris dans le téléchargement de doxygen à partir du site du cours. Pour cela vous aurez besoin du fichier de configuration pouvant être trouvé sur le site web du cours dans la section outils. Si pour ce TP, la documentation générée par doxygen n'est pas obligatoire, à partir du TP4, c'est celle-ci qui sera principalement prise en compte dans la note. Il est donc conseillé de commencer à regarder comment l'écrire et comment la générer.