ÉCOLE
POLYTECHNIQUE
MONTRÉAL

PHS4700
Devoir #2
Trajectoire d'un ballon de soccer
Présenté à M. Guy Marleau

Matthieu Ouellette-Vachon – 1325531

6 octobre 2009

Introduction

Ce document est produit dans le cadre du devoir #2 du cours « Physiques pour les applications multimédia ». Le but de ce laboratoire est de faire une simulation par ordinateur d'un ballon de soccer à l'aide de la deuxième loi de Newton et des forces qui régissent le mouvement du ballon. La simulation à lieu à l'intérieur d'un terrain de jeu ayant les mêmes dimensions qu'un terrain de soccer officiel. Le terrain possède une ligne centrale délimitant le terrain en deux parties égales. Le terrain possède un but de soccer réglementaire dans la partie inférieur du terrain. La position initiale du ballon se trouve à (0, 0, 0) dans le coin inférieur gauche. Le but de la simulation est de réussir à faire un coup de pied de coin qui terminera sa trajectoire directement dans le but. La vitesse linéaire initiale ainsi que la vitesse angulaire initiale sont demandé à l'utilisateur au démarrage de la simulation.

Problème

Le problème du devoir consiste à résoudre les équations de mouvements afin de réaliser une simulation réaliste au point de vue physique en fonctions des forces exercés sur le ballon. Il y a trois équations différentielles à résoudre, deux pour le mouvement linéaire du ballon et une autre pour le mouvement de rotation de ce dernier.

Équations

Pour cette simulation, le programme utilise une méthode numérique pour faire la simulation. La méthode utilisée est celle de Runge-Kutta d'ordre 4. Le programme peut aussi utiliser la méthode d'Euleur mais dans le programme remis, c'est la méthode de Runge-Kutta qui est utilisée. La décision d'utiliser Runge-Kutta a été prise à cause de sa plus grande tolérance aux erreurs puisque cette méthode est d'ordre 4 tandis que celle d'Euler (tel qu'implémenté) est d'ordre 1. Même si la méthode de Runge-Kutta est plus couteuse en termes de temps (quatre évaluation de fonctions plutôt qu'une seule évaluation), sa précision grandement supérieure en fait un choix judicieux pour les simulations de physique tel que la notre.

Trois forces viennent faire varier la vitesse linéaire du ballon. On retrouve tout d'abord la force gravitationnelle dû à l'attraction de la terre, la force visqueuse provenant du mouvement du ballon dans le fluide (l'air) et finalement la force de Magnus provenant du mouvement rotationnel du ballon dans l'air. Les trois forces sont notées F_1 (Force gravitationnelle), F_2 (Force visqueuse) et F_3 (Force de Magnus). Les forces sont toutes vectorielles.

$$F_1 = -mg$$

$$F_2 = -\frac{\rho C^{vis} A}{2} |\vec{v}|\vec{v}$$

$$F_3 = \frac{\rho C^{M} A}{2} |\vec{v}|^2 \frac{(\vec{w} \times \vec{v})}{|(\vec{w} \times \vec{v})|} \quad avec\ C^M = 0.385 \left(\frac{|\vec{w}|r}{|\vec{v}|}\right)^{0.25}$$

Grâce à ces trois forces et la deuxième loi de Newton, il est maintenant possible d'obtenir l'accélération linéaire du ballon. Cette dernière sera alors utilisée pour résoudre les équations de mouvements.

$$\sum F = m\vec{a}$$

$$\vec{a} = \frac{\sum F}{m}$$

Puisqu'on a maintenant l'accélération linéaire, il est donc aisé d'avoir la vitesse et la position du ballon grâce aux équations différentielles du mouvement.

$$\frac{dv(t)}{dt} = \vec{a} = \frac{\sum F}{m} = \frac{F_1 + F_2 + F_3}{m}$$

$$\frac{dr(t)}{dt} = v(t)$$

Au temps t_0, la vitesse de la balle est donnée par v_0 (fournis par l'utilisateur) et la position par r_0 (0, 0, 0). La résolution de la première équation à l'aide de l'analyse numérique nous permettra d'obtenir la vitesse de la balle au temps t_i. Cette vitesse au temps t_i sera ensuite utilisée pour résoudre la deuxième équation ce qui nous donnera alors la position de la balle au temps t_i. À l'aide de ces valeurs, on détermine si le ballon entre dans le but ou s'il termine sa trajectoire sur le terrain ou à l'extérieur du terrain. Dans le cas où le ballon tombe à l'extérieur du terrain, la position à l'extérieur est donnée (à gauche, à droite, à l'arrière ou à l'avant du terrain).

Pour ce qui est de la rotation du ballon, c'est l'équation différentielle pour les quaternions qui est utilité. On a alors :

$$\frac{dq(t)}{dt} = \frac{q(t) * \ \omega(t)}{2} \ avec \ \omega(t) \ le \ quaternion \ [0, \vec{\omega}]$$

Le quaternion $q(t_0)$ est donné par le quaternion [1, 0, 0, 0] où les composantes sont [w, x, y, z]. Après que le solveur d'équations différentielles est retourné le quaternion au temps t_{i+1}, ce dernier est normalisé pour éviter les problèmes de grossissement du ballon mentionné dans le cadre du cours théorique.

Solution

La solution proposée est un projet Visual Studio 2008. Le fichier de la solution se trouve dans « build/vss9/GameEngine.sln ». Cette solution contient deux projets. Le projet « GameEngine » qui est le cœur du programme de simulation. Il contient le code pour l'affichage de la fenêtre, les solutions numériques, et bien d'autres trucs utiles. La source de ce projet se trouve dans le répertoire « src ». Le projet « Tp2 » contient le code requis pour faire toute la simulation demandée, le code source pour ce projet se trouve dans « src\tp2 ».

Pour exécuter la solution, deux options sont offertes. On peut soit démarrer le projet depuis Visual Studio. Dans ce cas, il faut aller dans les propriétés du projet « Tp2 » et sous l'onglet « Debugging » il faut changer la valeur du champ « Working Directory » pour qu'elle soit assignée à « $(SolutionDir)..\..\bin ». Sinon, à chaque compilation depuis Visual Studio, l'exécutable est copié dans le dossier « bin » à la racine du projet. Il suffit alors de double cliquer sur l'exécutable copier dans ce dossier pour lancer la simulation.

Il est possible de faire pivoter la caméra grâce aux mouvements de la souris : gauche-droite pour faire varier theta et haut-bas pour faire varier phi (coordonnées sphériques). Les flèches directionnelles sont utilisables pour déplacer la caméra selon l'axe des X ou des Y. Finalement, la touche « r » peut être utilisée pour recommencer l'affichage d'une simulation déjà terminée.

Résultats

Voici un tableau récapitulatif montrant différents résultats d'exécutions. Pour chaque ligne on a la position initiale (au haut de la case) et finale (au bas de la case) de la balle, la vitesse initiale et finale (comme pour la position), la vitesse angulaire initiale du ballon, une description de l'état du ballon à la fin de la simulation (voir paragraphe plus bas) et finalement le temps écoulé lors de la trajectoire du ballon de soccer.

L'état du ballon à la fin de la simulation peut être :

- Le ballon est sorti à l'arrière du terrain (Arrière) (plan x = 0)
- Le ballon est sorti à l'avant du terrain (Avant) (plan x = 120)
- Le ballon est sorti à droite du terrain (Droite) (plan y = 90)
- Le ballon est sorti à gauche du terrain (Gauche) (plan y = 0)
- Le ballon entre dans le but (But)
- Le ballon tombe sur le terrain sans entrer dans le but (Terrain)

Position (m)	Vitesse (m/s)	Vitesse Angulaire (rad/s)	État Final	Temps (s)
(0, 0, 0) (90, 43, 0.264)	**(75, 75, 15) (17.2, 0.537, -10.5)**	**(PI, 0, -2*PI)**	**Droite**	**2.72**
(0, 0, 0) (56.2, 36.6, 0)	**(75, 75, 15) (30.7, 11, -7.03)**	**(-PI, 0, -2*PI)**	**Terrain**	**1.22**
(0, 0, 0) (66.5, 120, 7.46)	**(25, 255, 45) (17.6, 11.1, -9.49)**	**(0, 0, -2*PI)**	**Avant**	**2.5**
(0, 0, 0) (55.8, 0, 1.28)	**(175, 25, 20) (65.1, -9.91, -5.86)**	**(0, PI, -2*PI)**	**Arrière**	**0.54**
(0, 0, 0) (0, 27.4, 0.758)	**(15, 225, 15) (-8.67, 129, -1.14)**	**(-PI, PI, 2*PI)**	**Gauche**	**0.159**
(0, 0, 0) (45.5, 0, 0.573)	**(120, 15, 12.5) (54.1, -6.99, -5.36)**	**(0, PI, -2*PI)**	**But**	**0.578**

Conclusion

Premièrement, j'ai des doutes sur la grandeur des valeurs initiales que j'utilise. Dans l'énoncé, il est écrit qu'en moyenne, les ballons de soccers atteignent rarement plus de 130 Km/h ce qui donne 36.111 m/s. Or, mes vitesses initiales sont souvent de 100 m/s et plus pour obtenir une trajectoire correcte. J'ai vérifié à plusieurs reprises les unités de mesures que j'utilise et je n'ai pas détecté de problème à ce niveau. Je ne sais pas donc si c'est valeur sont normales ou si j'ai une erreur à quelque part dans mon implémentation.

Une observation que j'ai pu constatée lors des mes différents essais est que le comportement de la trajectoire pouvait changer substantiellement si je faisais varier le pas de temps de la simulation. En effet, à un certain moment, j'avais des valeurs pour les vitesses linéaire et angulaire me permettant de lancer le ballon dans le but avec un delta t de 0.05 secondes. Quand j'ai changé cette valeur de 0.05 à 0.01 secondes, le ballon était beaucoup plus à gauche (dans la direction des y positifs) et il n'entrait plus du tout dans le but. On voit donc clairement que la variation rapide de la vitesse jumelée à un pas de temps faible entrainait une plus grande erreur. Par exemple, voici les valeurs de la position pour la simulation où le ballon entre dans le but (dernière ligne du tableau ci-dessus) avec des pas de temps différents :

1. Position Finale (34.9, 0, 0.499) avec un pas de temps de 0.1 secondes
2. Position Finale (40.8, 0, 0.548) avec un pas de temps de 0.05 secondes
3. Position Finale (45.5, 0, 0.573) avec un pas de temps de 0.01 secondes
4. Position Finale (46, 0, 0.576) avec un pas de temps de 0.005 secondes

On remarque donc des variations importantes reliées directement à la variation du pas de temps. Les simulations du tableau précédent ont toutes été réalisées avec un pas de temps de 0.01 secondes. Ce pas de temps a été choisit car l'erreur entre 0.01 et 0.005 est assez faible comparativement à 0.05 et 0.01. De plus, le passage de 0.01 à 0.005 entraine une augmentation du nombre de ballon affiché donc j'ai préférer garder 0.01 comme pas de temps dans le but d'afficher moins de ballons afin de mieux voir la rotation de ce dernier au fil des itérations.

Un autre élément qui fait grandement varier la trajectoire du ballon de soccer est la vitesse angulaire initiale. Cette dernière peut faire changer totalement la direction du ballon dépendamment de son sens de rotation. Par exemple, lorsque la rotation est négative sur l'axe des X, le ballon a tendance à aller plus rapidement vers le bas. Au contraire, si la rotation est positive sur le même axe, le ballon fera une monté beaucoup plus prononcée vers le ciel. On peut voir ce phénomène pour la simulation où il tombe sur le terrain et celle où il tombe à droite du terrain. Dans ces deux cas, la seule valeur initiale qui change est le sens de rotation en X. Dans un cas, il tombe au trois de quart de largeur (-PI) tandis que l'autre tombe à l'extérieur (PI).

En conclusion, la simulation semble être bien rendue même s'il subsiste un doute dans mon esprit au niveau de l'ordre de grandeur des valeurs initiales de vitesse linéaire que j'ai utilisées.