Projet informatique S4

1. Introduction

Expliquer le principe (c’est-à-dire : le temps imparti, le déroulement, les étapes, le choix …)

1. Etude physique du phénomène de projection initiale
   1. Principe

De manière similaire aux équations utilisées pour les jets de balles, on étudie physiquement la trajectoire initiale du joueur.

Pour modéliser la trajectoire initiale du jour, nous avons choisi, dans l’étude physique, de lui faire adopter une trajectoire balistique. En faisant l’hypothèse qu’il n’est soumis qu’à son propre poids, et en appliquant le principe fondamental de la dynamique nous parvenons à la relation suivante :

En projetant sur les deux axes du repère on obtient :

En intégrant par rapport au temps, et en prenant en compte les conditions initiales : , on obtient :

En intégrant de nouveau par rapport au temps :

**y=-g/(2\*v02\*cos(apha)2)\*x2+x\*tan(alpha)+y0**

Afin de trouver les valeurs initiales optimales pour notre jeté, nous avons eu recours au logiciel Matlab. Plutôt que de faire les essais directement sur java.

* 1. Simulation

Voici le programme réalisé sur Matlab:

% Programme test Matlab projection du personnage

clear all;

close all;

% initialisations des valeurs

g=10; % accélération de la pesanteur

x0=150 ; %position initiale en x, ici valeur arbitraire, dépendant de la plateforme de départ

y0=250; %position initiale en y, ici valeur arbitraire, dépendant de la plateforme de départ

v0=100; % vitesse initiale du projectile, ici arbitraire, calculable proportionnellement au nombre de clique du joueur

alphad= 45; % angle initiale de projection en degrés

alphar=alphad\*(pi./180);

dmax = 300; % distance maximale parcourue en x pour ne pas excéder la fenêtre

ymax=600; % hauteur maximale atteinte pour ne pas excéder la fenêtre

xmax=(v0.^2)./g\*sin(2\*alphar) % coordonnée en x du maximum selon les valeurs entrées initialement

pas = xmax./500 % pas de calcul, on veut 100 calculs, veiller à ce qu'il soit entier pour les coordonnées sur fenetres !

x= x0:pas:xmax; % discrétisation de la valeur de x

y=(-g/(2\*(v0.^2)\*(cos(alphar)).^2)\*x.^2+x\*tan(alphar))+y0; %équation de la parabole issue de l'application du PFD, calcul de y pour chaque x souhaité

%boucle de tracé

for p=1:pas:length(x)

figure (1)

title('Simulation d''un projectile uniquement soumis à son poid')

trace = plot(x(p),y(p),'.'); % tracé

xlabel('distance (m)'); ylabel('hauteur (m)');

set(trace,'LineWidth',2);

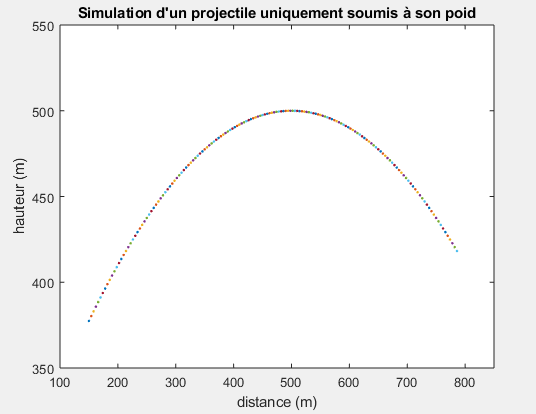
xlim([100 850]); ylim([350 550]);

hold on;

pause(0.001);

end

La courbe obtenue est la suivante :



Il est à noter qu’il faut veiller (à contrario de ce qu’il y a dans la simulation Matlab) à caster les valeurs de x et y comme entières, afin d’obtenir les coordonnées ainsi que d’imposer une fin de course lorsque le personnage atteint une plateforme (condition sur y).

Et le système de coordonnées étant différent sur java, il faut changer l’écriture des équations : il faut « translater » le repère de calcul (dans java, le y est orienté vers le bas), ainsi que d’écrire le fait que l’accélération de pesante ne présente pas de signe négatif (car pesanteur et axe y sont dans le même sens). On obtient alors les équations suivantes, (présentes dans la méthode deplacementInit dans la classe Joueur) :

y(x) = y0+9.8/(2\*v02\*cos(alphar)2)\*(x-x0)2+(x-x0)\*tan(alphar)

avec (x0,y0) la position initiale du joueur.