

Algorithmique et programmation

TP 8 : Jeu des châteaux

1 Objectif

L'objectif du TP est d'implémenter un petit jeu à deux joueurs. Chaque joueur détient un château, avec un canon pointé dans la direction du château adverse. Le but du jeu est de détruire le château de son adversaire. Pour cela, les joueurs tirent à tour de rôle. Avant de tirer, le joueur dont c'est le tour de jouer doit entrer l'angle de tir, ainsi que la force du tir. Le premier joueur qui réussit à toucher le château adverse a gagné.

Pour compliquer un peu les choses, on ajoute le fait qu'il y a du vent (de vitesse constante) qui va avoir une influence sur la trajectoire du boulet de canon. On ajoute également une colline entre les deux châteaux. Il faudra donc tirer par dessus la colline en tenant compte de l'influence du vent.

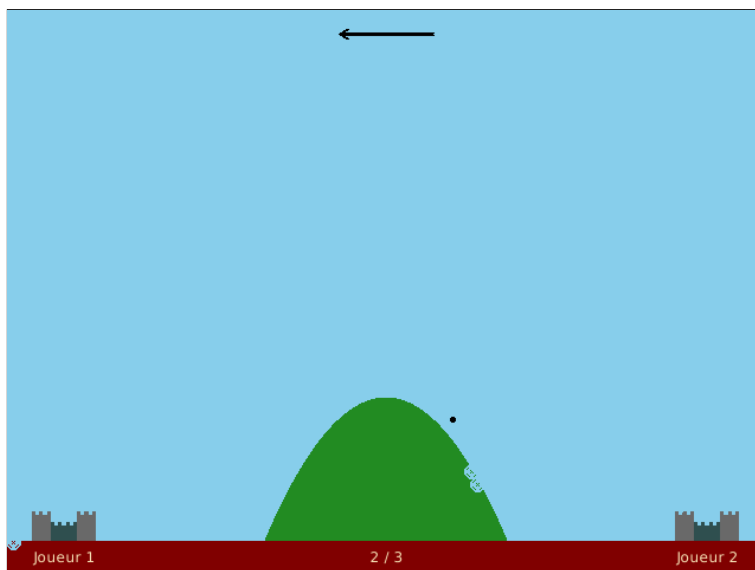


FIGURE 1 – Terrain de jeu

La figure 1 donne un exemple de représentation du jeu. On y voit les deux châteaux, ainsi que la colline. Le vent est symbolisé par la flèche : le vent souffle dans le sens de la flèche, avec une vitesse proportionnelle à la longueur de la flèche. On peut également distinguer des traces d'impact sur la colline, et un boulet de canon en cours de vol.

Une spécification précise de ce qui est attendu, ainsi que diverses indications sont données en section 2. Des extensions possibles au sujet de base sont proposées ensuite en section 3.

2 Spécifications

Une description plus détaillée de ce qui est attendu est donnée dans cette section. On commencera par décrire l'interface (2.1), on donnera ensuite quelques indications sur la modélisation de la colline (2.2) et du vent (2.3). On décrira ensuite la manière de simuler la trajectoire du boulet (2.4), avant d'évoquer les collisions du boulet (2.5) et de conclure sur les paramètres du jeu (2.6).

2.1 Interface

Du point de vue de l'interface, il est attendu d'avoir une représentation graphique du terrain de jeu, faisant apparaître les châteaux, la colline, ainsi qu'une représentation du vent (par exemple avec une flèche, comme sur la figure 1).

Les interactions avec les joueurs, notamment pour la saisie des paramètres de tir, pourront se faire sur la console en utilisant l'entrée et la sortie standard.

La modélisation de la position des châteaux, de la colline et de la trajectoire du boulet se feront dans un repère orthogonal. Afin de simplifier les calculs, il est suggéré de définir un repère propre au terrain de jeu, et de faire les conversions de coordonnées appropriées pour l'affichage (voir pour cela l'exercice 2.4 du TP 7). On pourra par exemple placer les deux châteaux sur l'axe des abscisses de manière symétrique par rapport à l'axe des ordonnées. Les deux châteaux auront donc pour coordonnées $(-p, 0)$ et $(p, 0)$, et la colline sera centrée autour de l'axe des ordonnées.

2.2 Colline

La colline sera modélisée par une simple parabole, définie par sa hauteur et sa largeur à la base. La hauteur et la largeur à la base de la colline seront tirées aléatoirement avant chaque partie. Avec ces paramètres, on peut définir la colline par la zone se trouvant sous la courbe d'équation :

$$y = hauteur \times \left(1 - \left(\frac{2 \times x}{largeur} \right)^2 \right)$$

2.3 Vent

Comme les paramètres de la colline, la vitesse et le sens du vent sera tirée aléatoirement et restera ensuite la même tout au long d'une partie. En posant *ventMax* la vitesse maximale du vent, on choisira la vitesse du vent dans l'intervalle $[-ventMax; ventMax]$. Un vent positif soufflera de gauche à droite, et un vent négatif de droite à gauche.

2.4 Trajectoire du boulet

Pour calculer la trajectoire¹, l'idée est d'évaluer sa position à tous les intervalles de temps δ_t . La position du boulet à un instant t est alors donnée par un couple de coordonnées $(x(t), y(t))$ dans le repère. Pour cela, on part des conditions initiales qui sont : la position initiale du boulet $(x(0), y(0))$, l'angle de tir α et la force du tir v_0 . De plus, on note v_{vent} la vitesse du vent, g la constante de gravitation (9,81) et k le coefficient de frottement de l'air.

1. Le but ici n'est pas de faire une simulation de balistique fidèle à la réalité (les paramètres pourraient être beaucoup plus nombreux, et les équations bien plus compliquées), mais simplement de rendre quelque chose qui ressemble à ce qu'on peut s'imaginer être la réalité.

On commence par définir la vitesse initiale, suivant ses composantes x et y par :

$$\begin{cases} v_x(0) = v_0 \times \cos \alpha \\ v_y(0) = v_0 \times \sin \alpha \end{cases}$$

Ensuite, connaissant la position et la vitesse du boulet à un instant t , on évaluera la position et la vitesse à l'instant $t + \delta_t$ de la manière suivante. On commence par estimer l'accélération à l'instant t à l'aide des formules :

$$\begin{cases} a_x(t) = -k \times v_r(t) \times (v_x(t) - v_{vent}) \\ a_y(t) = -k \times v_r(t) \times v_y(t) - g \end{cases}$$

avec

$$v_r(t) = \sqrt{(v_x(t) - v_{vent})^2 + (v_y(t))^2}$$

et on peut enfin calculer la position du point en $t + \delta_t$:

$$\begin{cases} x(t + \delta_t) = x(t) + v_x(t) \times \delta_t \\ y(t + \delta_t) = y(t) + v_y(t) \times \delta_t \end{cases}$$

ainsi que sa vitesse :

$$\begin{cases} v_x(t + \delta_t) = v_x(t) + a_x(t) \times \delta_t \\ v_y(t + \delta_t) = v_y(t) + a_y(t) \times \delta_t \end{cases}$$

2.5 Collisions

La simulation de la trajectoire du boulet devra s'arrêter lorsque celui-ci rencontre un obstacle. Il y a *a priori* trois sortes d'obstacles :

le sol, lorsque la composante y de la position du boulet devient négative ;

la colline, lorsque le boulet passe *sous* la courbe la définissant ;

un château, lorsque le boulet tombe à proximité de la position du château.

2.6 Paramètres

La description du jeu donnée précédemment dépend d'un certain nombre de paramètres comme : les positions des châteaux, la hauteur et la largeur minimales et maximales de la colline, la vitesse maximale du vent, le coefficient de frottement de l'air k , l'intervalle de temps de la simulation de trajectoire δ_t , etc.

Il est *fortement conseillé* de définir des constantes en début de programme pour tous ces paramètres. Les valeurs pour les paramètres devront être choisies de manière à rendre le jeu le plus attrayant possible.

3 Extensions possibles

Les étudiants les plus motivés pourront ajouter quelques extensions aux spécifications initiales. On peut citer par exemple (la liste n'est pas exhaustive) :

- la représentation des impacts des boulets ;
- un affichage particulier lors de la destruction d'un château ;
- la possibilité de recommencer une partie, avec le décompte des parties gagnées par chaque joueur ;
- un mode où l'ordinateur simulerait un joueur ;
- une forme plus complexe pour la colline.