

Asteroids - Projet PRG2

Théo NAZÉ et Antoine PINSARD

5 Avril 2015

1 Introduction

La majeure partie du projet réside dans la gestion des déplacements des formes dans le plan.

Afin de gérer cela, nous avons décidé d'attribuer à chaque objet (vaisseau, astéroïde, missile), une position (celle de son centre de gravité), une orientation et une vitesse. Ainsi, à chaque étape, nous modifions la position du centre de gravité en fonction de la vitesse et de l'orientation. Puis nous recalculons la position des différents sommets du polygone en fonction de la taille de l'objet, de son centre de gravité et de son orientation. Pour la rotation du vaisseau, il suffit de modifier son orientation.

Se pose ensuite le problème des collisions. Nous avons d'abord pensé à calculer l'intersection entre chaque astéroïde/missile et entre le vaisseau et chaque astéroïde. Afin de simplifier le problème nous avons finalement décidé de calculer la distance entre les deux éléments. Si la distance est inférieure à la somme des rayons des objets, on peut considérer une collision. Cette méthode est moins précise que celle de l'intersection, mais plus simple à mettre en place.

2 Analyse du problème

Le calcul de l'état suivant se découpe en 6 étapes :

- Calcul du nouvel état du vaisseau
- Calcul du nouvel état des missiles
- Calcul du nouvel état des astéroïdes
- Vérification des collisions astéroïde/vaisseau
- Vérification des collisions astéroïde/missile et recalcul de l'état des astéroïdes le cas échéant.
- Vérification de la présence d'astéroïdes

2.1 Calcul du nouvel état du vaisseau

L'accélération n'étant pas implémentée, et la rotation étant traitée directement à l'enfoncement des touches "j" et "l", cette étape est omise.

2.2 Calcul du nouvel état des missiles

Le déplacement d'un missile se fait à l'aide de la formule suivante :

$$new_pos = pos + (\cos(orientation) * vitesse, \sin(orientation) * vitesse)$$

Si la nouvelle position sort du cadre de jeu, le missile est supprimé.

2.3 Calcul du nouvel état des astéroïdes

Le calcul du déplacement d'un astéroïde se fait de la même manière que pour celui d'un missile :

$$new_pos = pos + (\cos(orientation) * vitesse, \sin(orientation) * vitesse)$$

En revanche, la position est calculée modulo (largeur du cadre, hauteur du cadre) de manière à faire réapparaître l'astéroïde de l'autre côté de l'écran lorsqu'il sort d'un côté.

2.4 Vérification des collisions astéroïde/vaisseau

On parcourt la liste des astéroïdes et on vérifie la distance avec le vaisseau. Si la distance est trop faible, le jeu se termine : partie perdue.

2.5 Vérification des collisions astéroïde/missile

On parcourt la liste des missiles. Pour chaque missile on parcourt la liste des astéroïdes. On calcul la distance entre les deux objets. Si elle est trop faible, le missile est supprimé et l'astéroïde éclatée en 2 à 4 astéroïde de taille inférieure. Si la taille était déjà de 1 (la taille minimum), l'astéroïde est simplement détruite.

2.6 Vérification de la présence d'astéroïdes

S'il n'y a plus d'astéroïdes, le jeu s'arrête : la partie est gagnée.

3 Bilan

Nous avons réalisé le fonctionnement basique du jeu. Le vaisseau est de forme triangulaire. Les astéroïdes sont de forme ronde, de différentes couleurs et de différentes tailles. Les missiles sont de forme ronde également, par souci de simplicité.

Le tir de missiles est implémenté mais les problèmes suivants restent à corriger :

- Une fois un astéroïde percuté, le missile continue sa trajectoire
- Seules les collisions avec le dernier missile lancé sont prises en compte.

La collision missile/astéroïde est implémentée.

La collision astéroïde/vaisseau n'est pas implémentée.

4 Exécuter le programme

`./asteroids`

5 Rapport de test

(À compléter *)*

6 Conclusion

La réalisation du projet nous a pris deux jours. Les difficultés que nous avons rencontrés sont le manque de documentation du langage. Très peu de site référencent les bibliothèques et built-ins d'OCaml. Globalement, la seule ressource valable est `caml.inria.fr`. Il est aussi assez difficile d'exploiter l'interpréteur interactif pour effectuer de petits tests et vérifier rapidement des résultats attendus. En effet il n'est pas possible d'utiliser les flèches directionnelles pour se déplacer sur la ligne courante et rappeler des commandes précédemment exécutées.