

# 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. Pour cela, après avoir calculé l'état suivant, il faut vérifier l'intersection entre le vaisseau et chaque astéroïde. Si elle est vide, on ne fait rien. Si elle n'est pas vide, la partie est perdue. Il faut également vérifier l'intersection entre chaque astéroïde et chaque missile. Si l'intersection est vide, on ne fait rien. Si elle n'est pas vide, on supprime le missile et lance le comportement d'explosion de l'astéroïde.

## 2 Analyse du problème

Le calcul de l'état suivant se découpe en 5 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.

### 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 :

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

## 3 Bilan

(\* À compléter \*)

## 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.