

Informatique Graphique : Projet

Ce projet, réalisé dans le cadre du module d'informatique graphique en 4ème année à l'INSA de Rennes, consistait à animer une voiture dans un scène. J'ai choisi de passer la majorité de mon temps sur le moteur physique, et principalement sur les mouvements de la voiture et la détection des collisions. Le projet est aussi composé d'un moteur de rendu composé en travaux pratique, qui contient entre autres la possibilité de charger des fichiers .obj réalisés sous Blender ou encore de la simulation de lumière et matériaux selon le modèle de Phong. Les fonctionnalités de ce moteur de rendu ne seront pas explicités dans ce rapport puisqu'elles le sont déjà dans des rapports précédents. Certaines fonctionnalités telles que les textures, biens qu'implémentées lors des travaux pratiques n'ont pas été ajoutée ici par manque de temps. Une vidéo du résultat est disponible [ici](#).

Partie I : Physique de la voiture

Le principal élément de mon projet est la physique sur les déplacements de la voiture, elle est contrôlable au clavier (à l'aide des flèches) et j'ai essayé d'avoir une physique la plus proche possible du réel. Ma principale source pour cela a été une thèse nommée [*Development of a car physics engine for games*](#).

Les différentes fonctionnalités que permet cet algorithme sont les suivantes :

- Accélération linéaire et angulaire pour des mouvements progressif et moins brusque.
- Effet de dérapage lors des virages à grande vitesse
- Animation réaliste des roues
- Mouvement dynamique de la caméra pour plus d'immersion.

Afin d'avoir un bon milieu entre agréable et réalistes, il m'a fallu régler un bon nombre des constantes présentes dans les formules, principalement les formules de calcul de force. L'algorithme se déroule de la manière suivante :

- On calcule toutes les forces exercées sur la voiture : La force de traction (correspondant aux roues), la force latérale (correspondant à la tendance à déraiper de la voiture), la force de trainée (correspondant à la résistance exercé par l'air sur la voiture), la force de résistance (correspondant aux frottements des roues sur le sol) et enfin la gravité.
- De ces forces on peut, grâce aux lois de Newton, déduire une accélération linéaire et angulaire.
- On intègre cette accélération pour avoir une vitesse puis on intègre à nouveau pour avoir la position et on met à jour le modèle.
- On peut aussi calculer la vitesse de rotation des roues afin de les animer elles aussi.
- Pour un rendu plus agréable j'ai aussi ajouté à la caméra un effet de décalage lors des virages. La caméra est légèrement tournée à droite lors d'un virage à gauche et inversement afin de mieux donner l'impression de rotation.

Partie II : Détection de collisions

L'autre fonctionnalité principale de mon projet est la détection de collision entre « boîte de délimitation orientée » (OBB : *Oriented Bounding Box* en anglais). Dernière se nom se cache simplement un cube déformé qui encapsule nos objets. Tous les objets de la scène sont liés à une de ces boîtes, qui épouse leur forme du mieux qu'elle peut et se déplace avec eux. La détection de collision consiste comme son nom l'indique à détecter lorsque les boîtes de deux objets entrent en contact.

La première partie consiste donc à calculer la forme de cette boîte, pour cela je me contente simplement de récupérer les positions extrêmes de l'objet selon les trois axes. Ces positions correspondront aux sommets de la boîte et nous sommes ainsi sûr que l'intégralité de l'objet est contenue dans celle-ci.



Figure 1 : Les boîtes de collision de la voiture sont représentée en fil de fer

Pour détecter les collisions, j'ai ici utilisé le *Separating Axis Theorem (SAT)*. Le théorème en soit est très simple et dit que s'il est possible de passer un axe quel qu'il soit entre deux objets convexes (comme le sont ici nos boîtes), alors ces objets ne sont pas en collision.

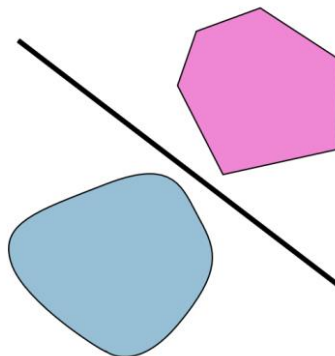


Figure 2 : Représentation 2D du théorème (source Wikipédia)

Le théorème est assez intuitif, principalement en 2D, mais il propose une difficulté : quels axes doit on observer pour s'assurer qu'il y ait bien collision ou non ? Si on utilisait des boîtes non orientées (c'est-à-dire alignées selon les axes X, Y et Z), il suffirait de vérifier selon ces trois axes, mais ici avec des boîtes orientées c'est plus compliqué. D'après la thèse [Dynamic Collision Detection using Oriented Bounding Box](#), il faut un total de 15 axes pour assurer une détection complète des collisions :

- **Les 3 axes de la première boîte** (qui serviront à savoir si un sommet de la première boîte a pénétré la seconde)
- **Les 3 axes de la seconde boîte** (qui serviront à savoir si un sommet de la seconde boîte a pénétré la première)
- **Les 9 axes des produits scalaires des arrêtes** des deux boîtes (qui serviront à savoir si celles-ci sont entrées en collision selon leurs arrêtes).

Pour vérifier si un axe passe entre deux formes, on projette ces deux formes selon cet axe, si les projections n'intersectent pas, l'axe sépare bien les deux formes et inversement. Si un seul de ces axes peut être passé entre les deux formes, alors il n'y a pas collision. Dans mon cas cependant, la voiture tourne uniquement sur l'axe Y, et il est donc impossible d'avoir une collision arrête à arrête, je fais donc uniquement les tests sur les 6 premiers axes afin d'économiser du temps de calcul mais le système peut s'étendre si nécessaire.

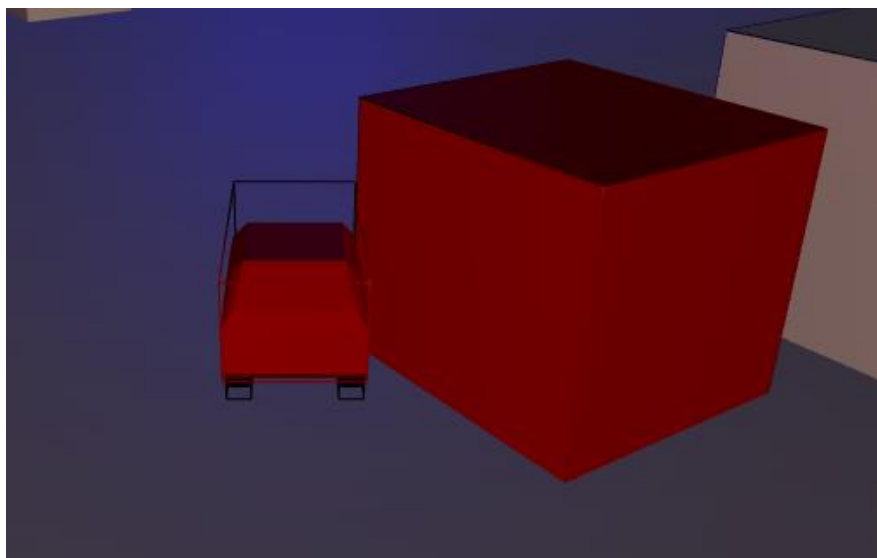


Figure 3 : Exemple de collision, la boîte en collision avec la voiture s'allume en rouge

Je voulais ensuite intégrer de la réaction à la collision, c'est-à-dire faire en sorte que les mouvements de la voiture soient modifiés lorsqu'elle entre en contact avec un des obstacles présents sur la scène mais cela s'est avéré bien plus difficile que prévu. Mon principal problème est que le SAT ne permet pas de savoir selon quelle surface de la boîte il y a eu collision, or il me faudrait savoir la normale de cette surface afin d'y appliquer d'appliquer une force contraire à la voiture suivant cette normale et simuler l'effet d'un mur. J'ai donc abandonné cette idée par manque de temps.