



Projet de synthèse d'images Hovercraft

Imac première année

Abstract

L'objectif de ce projet est de réaliser un petit jeu temps réel permettant d'exploiter vos nouvelles connaissances en OpenGL. Vous travaillerez par binômes.



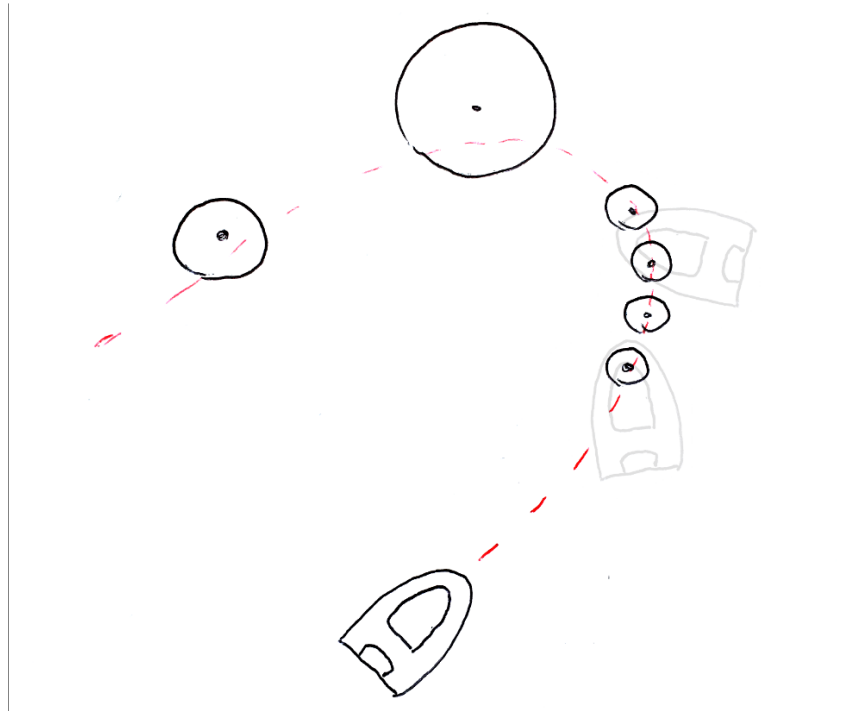
1 Introduction

Il s'agit de réaliser un petit jeu d'aéroglysieur en 2D, vu de dessus. Le but du jeu est de manipuler l'aéroglysieur pour lui faire atteindre un ensemble de balises.

2 Déroulement

Un aéroglisseur est un véhicule amphibie à portance aérostatique et à propulsion aérienne (wikipedia). Contrairement à une voiture, son adhérence est très faible, ce qui ne lui permet pas de prendre des virage très serrés à grande vitesse.

Le jeu demandé affiche une vue de dessus d'un terrain 2D. Ce terrain comporte un ensemble de *check points* matérialisés sous forme de cercle. Le but du jeu est de passer par l'ensemble des *check points* en un temps minimal.



3 Manipulation de l'aéroglisseur

3.1 Manipulation

La manipulation de l'aéroglisseur se fait en changeant son orientation et la force de son moteur. Il n'y a donc en tout que 3 commandes :

- droite ou gauche : change l'orientation de l'aéroglisseur.
- puissance : produit une accélération dans la direction de l'aéroglisseur.

3.2 Physique

La trajectoire de l'aéroglesseur sera calculée à l'aide du principe fondamental de la dynamique :

1. vous appliquez une accélération au véhicule.
2. vous mettez à jour sa vitesse.
3. vous mettez à jour sa position.

C'est à vous de trouver durant laquelle de ces trois phases il faut traiter les forces de frottement. Il ne s'agit pas d'être rigoureux sur le plan scientifique, mais plutôt d'obtenir un résultat ludique.

3.3 Les commandes

L'aéroglesseur devra être piloté au clavier. Les plus motivés pourront faire un mode multijoueurs avec partage du clavier.

4 Le terrain

4.1 Modélisation

Un terrain est défini par les attributs suivants :

- le nombre de *check points*.
- pour chaque *check point* :
 - leur centre
 - leur rayon

4.2 Les collisions

Le passage sur un *check point* est très similaire à un calcul de collision (qui n'aurait pas d'effet sur le véhicule). La traversée d'un *check point* doit se matérialiser par un événement graphique (changement de couleur du *check point*, ou autre).

4.3 Chargement

Les données du terrains seront éditées sur un fichier texte qui sera lu par votre programme avant le lancement du jeu.

4.4 Les bords

Lors du chargement des données, votre programme devra automatiquement ajuster la taille de la carte à la taille de la fenêtre. Cette taille est définie par le plus petit rectangle recouvrant l'ensemble des *check points*. Il est possible de zoomer sur l'aéroglesseur et de le suivre lors de ses déplacements, il faudra alors veiller à trouver une solution pour indiquer au joueur où sont les *check points* non parcourus. Enfin, vous pourrez traiter les bords de l'image comme des bords rigides ou bien simplement les ignorer (c'est juste une question de game design).

5 Options

Pour ceux qui en veulent plus, voici ce que vous pouvez ajouter :

- Vous pouvez charger un terrain sous forme d'image au format `png`. Il ne s'agit pas d'afficher l'image sous forme de texture, mais plutôt de la lire. La couleur d'un pixel correspond en fait à un code de terrain (mer, terre, rocher, ...). Chaque type de terrain est franchissable, ou non, et possède un coefficient de frottement propre, ainsi qu'une couleur ou une texture.
- Ajouter un menu pour choisir son parcours.
- Ajouter la possibilité de contrôler l'aéroglesseur avec une manette (la SDL les reconnaît).
- Vous pouvez rajouter du son avec la `SDL_sound`.
- Donner une identité visuelle à votre jeu.
- Rajouter une IA pour avoir des concurrents (attention, c'est chaud).
- ...

Par contre, veuillez à ne pas commencer à rajouter d'options tant qu'une version basique et jouable du jeu n'est pas réalisée.

6 Le programme

Votre programme sera codé en langage C et devra compiler dans les salles machines de l'université, sous linux. Il sera composé de plusieurs fichiers .c et .h compilés à l'aide d'un `Makefile`. Il devra compiler sans warning (mais avec l'option `-Wall`). Vous rendrez votre projet sous forme d'archive nommée `nom1_nom2_radial.tgz`. Cette archive doit générer un répertoire nommé `nom1_nom2_hovercraft/` contenant votre projet. Votre programme ne comportera aucun `scanf`, les paramètres nécessaires au fonctionnement du programme (s'il y en a) seront transmis en ligne de commande (cf. `argc` et `argv`).

7 Le rapport

Vous fournirez en version électronique un rapport de 5 pages minimum et de 10 pages maximum. **Consacrez suffisamment de temps au rapport car il représente une bonne partie de la note finale.** Essayez de respecter au mieux les directives suivantes :

- **Votre rapport doit commencer par une page listant vos travaux avec les indications suivantes :**
 - éléments demandés et codés qui fonctionnent.
 - éléments demandés et codés qui ne fonctionnent pas.
 - éléments demandés mais pas codés.
 - éléments non demandés (options) et codés qui fonctionnent.
 - éléments non demandés mais pas codés ou qui ne fonctionnent pas, mais pour lesquels vous avez des choses à dire.
- Ne perdez pas de temps à réexpliquer le sujet du projet, l'enseignant le connaît déjà, faites seulement un bref résumé de quelques lignes. De manière plus générale, ne détaillez pas des méthodes déjà expliquées dans l'énoncé à moins que vous les ayez modifiées.
- Un rapport sert surtout à montrer comment vous avez fait face aux problèmes (d'ordre algorithmique/mathématique). Certains problèmes

sont connus (on en parle dans l'énoncé), d'autres sont imprévus. Montrez que vous les avez remarqués et compris. Donnez la liste des solutions à ce problème et indiquez votre choix. Justifiez votre choix (vous avez le droit de dire que c'est la méthode la plus facile à coder).

- Il ne doit figurer aucune ligne de code dans votre rapport. Un rapport n'est pas un listing de votre programme où vous détaillez chaque fonction. Vous devez par contre détailler vos structures de données et mettre du pseudocode pour expliquer vos choix algorithmiques. Il est autorisé d'utiliser des "raccourcis" tels que "`initialiser T à 0`" plutôt que de détailler la boucle faisant la même chose.
- N'hésitez pas à mettre des images dans votre rapport pour illustrer vos propos et vos résultats.

8 Pour finir

Vous pouvez laisser libre cours à votre imagination, toute amélioration sera la bienvenue. Vous trouverez quelques informations complémentaires à l'adresse :

<http://www-igm.univ-mlv.fr/~vnozick/>

Bon courage.