

Travail pratique de programmation concurrente

Travail pratique - Astéroïdes

1 Objectif

Transformer un code séquentiel en code multi-threadé à l'aide des primitives de concurrence POSIX vues en cours.

Pour ce faire, il faut utiliser le code se trouvant sur le repo https://githopia.hesge.ch/orestis.malaspin/concurrent_asteroids_code.

2 Règles du jeu

Ce jeu est une mauvaise version *d'Asteroids*.

Les règles du jeu sont assez simples. Le joueur contrôle un vaisseau qui peut accélérer, tourner à gauche ou à droite, et décélérer à l'aide des flèches du clavier. Il peut également tirer des boulets de canon pour détruire les astéroïdes. Le vaisseau peut être détruit s'il entre en collision avec un astéroïde et perd une vie à chaque destruction. Lorsque son nombre de vies est à zéro la partie se termine. Les boulets de canon permettent de détruire les astéroïdes. Lorsqu'un boulet de canon touche un astéroïde, celui-ci se sépare en deux parties de tailles égales. Un astéroïde peut être détruit deux fois avant de disparaître totalement. Les astéroïdes et le vaisseau sont soumis à la gravité (ils s'attirent) et les astéroïdes se repoussent lorsqu'ils sont trop près les uns des autres.

3 Énoncé

Adapter le code séquentiel fourni en suivant les règles suivantes:

1. Chaque astéroïde doit être dans son propre thread.
2. Lorsqu'un astéroïde est détruit chaque "sous-astéroïde" est dans un nouveau thread séparé.
3. Le vaisseau a son propre thread.
4. Chaque boulet de canon a son propre thread.
5. L'affichage et la gestion du clavier se font dans un seul thread séparé (voir TP précédent).
6. Si le temps le permet (on décidera en fonction de votre avancement), la sauvegarde/chargement de l'état du jeu (pas implémentée dans le code séquentiel) devra se faire dans un thread séparé (plus d'informations dans un futur proche).

3.1 Synchronisation

Il y a plusieurs endroits critiques pour la synchronisation entre les threads. Nous en listons quelques-uns ci-dessous.

3.1.1 L'accélération

La dynamique des astéroïdes à la force de gravité qui dépend de leurs positions respectives. Le vaisseau peut subir une accélération uniquement due aux commandes de l'utilisateur (rotation, accélération, décélération).

Afin d'éviter un accès concurrent (ou d'avoir une gestion supplémentaire de la mise à jour de la position des astéroïdes) il est recommandé d'avoir deux tableaux d'astéroïdes: un tableau contenant les positions avant la mise à jour des forces, et un tableau où les positions sont mises à jour après avoir mis à jour les forces.

La synchronisation des positions se fera ensuite lors de l'affichage des différents objets.

3.1.2 Déplacements

Les boulets de canons, le vaisseau et les astéroïdes se déplacent de façon indépendantes une fois les accélérations mises à jour. Ils doivent se synchroniser avant que la détection des collisions ait lieu. Les collisions peuvent avoir lieu entre les boulets et astéroïdes, et le vaisseau et les astéroïdes. Aucune autre collision n'a lieu.

3.1.3 Les collisions

Si un boulet de canon rencontre un astéroïde, l'astéroïde se sépare en deux sous astéroïdes de taille plus petite. Chaque astéroïde est créé dans son thread séparé. Un astéroïde peut être détruit deux fois, à la troisième il disparaît. S'il n'y a plus d'astéroïdes en jeu, le jeu se termine. Si le vaisseau entre en collision avec un astéroïde, il est détruit. Lorsque son nombre de vie atteint zéro la partie est finie. Au moment où le vaisseau est "recréé" il a une immunité pendant une certaine durée (il devient rouge). Les collisions doivent être toutes résolues avant l'affichage.

3.1.4 Les boulets de canon

Un boulet de canon ne peut être tiré qu'à une fréquence donnée (un certain intervalle de temps doit être attendu avant entre le tir de deux boulets successifs par le vaisseau). Par ailleurs, les boulets disparaissent après avoir parcouru une certaine distance. Il faut bien veiller à détruire les threads associés à ce moment là.

3.1.5 Affichage, entrées/sorties clavier

Pour effectuer l'affichage le jeu doit être dans un état synchronisé. Choisissez l'endroit qui vous semble le plus approprié pour le faire. Les entrées clavier doivent se faire avant la mise à jour de l'accélération pour le vaisseau. Il faut également gérer le mode "pause" du jeu.

3.1.6 Fin de partie

La partie doit se terminer proprement et les threads joints (pas de deadlocks ou de plantage ou de `exit` violent).

3.1.7 Sauvegarde/chargement de partie

Plus d'information si cette partie doit être implémentée.

4 Rendu

On fera un *pull* des repos le 7 juin à 23h59. Votre code doit être donc terminé. Le jeudi 11 juin, vous devrez effectuer une présentation de 20min (10min par personne dans le groupe). Ensuite suivront environ 10min de questions. La présentation se fera sur discord, et il faut que vous ayez préparé des diapositives.

4.1 Remarques

Mettez au propre un schéma de synchronisation dans votre présentation pour avoir une base pour expliquer le fonctionnement de votre programme.