Laboratoire de PCO #4
Simulation de trains

Rémi Ançay & Lucas Charbonnier

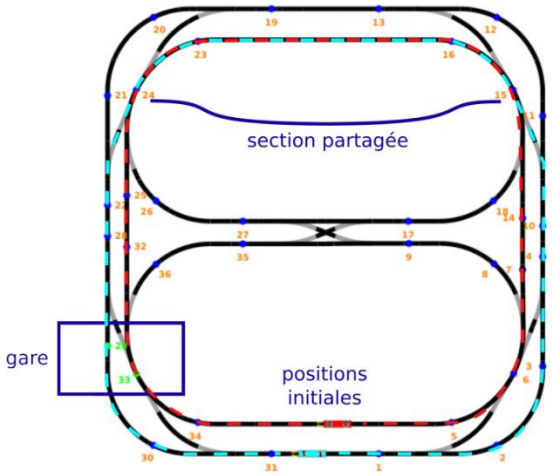
Introduction

Le laboratoire #4 de PCO consistait en l'implémentation de la gestion d'une section partagée de rails au sein d'une simulation de trains. Ce rapport contient le détail de notre implémentation ainsi que nos choix de conception.

Implémentation

Choix d'une maquette et d'un chemin de test

Nous avons choisi la même maquette et le même chemin de test que l'exemple montré dans l'énoncé :



Gestion de la section partagée

La gestion de la section partagée se fait à l'intérieur de la classe Synchro. Les deux trains font appel à la méthode access() pour indiquer le besoin d'accéder à la section. On commence par vérifier quel train est prioritaire (le deuxième train à être parti de la gare a la priorité). Si la section est déjà occupée, le train s'arrête et il attend passivement que la section soit libre.

Lorsque un train sort de la section partagée, il appelle la méthode leave() et donc libérer le sémaphore, ce qui permettra à l'autre train en attente d'accéder à la section.

Priorité des locos

La gestion de la priorité des locomotives se fait à l'aide d'un attribut privé numLocoPrioritaire dans la classe Synchro. Cet attribut est protégé par un mutex mutexPrio et est mis à jour par le deuxième train qui arrive à la gare. Le deuxième train réserve donc la priorité pour lui-même à son arrivée en gare.

Arrivée/départ en gare

L'arrivée et le départ dans la gare est coordonné grâce à un sémaphore gareSem et un mutexGare. Lorsqu'un le premier arrive en gare, il acquire() le sémaphore gareSem et entrera donc dans une attente passive. Le deuxième train s'occupera de release() ce sémaphore, ce qui permettra aux deux trains de partir en même temps.

Arrêt d'urgence

Après discussion avec le professeur, dans l'implémentation actuelle, il n'y a pas moyen de terminer le thread des trains correctement, exactement au moment de l'arrêt d'urgence. Nous avons donc choisi l'option fonctionnelle de fixer la vitesse des locos à 0 et de les arrêter (méthodes fixerVitesse() et arreter()). Elles n'ont donc aucun moyen de reprendre leur fonctionnement normal.

Pour l'arrêt d'urgence, il suffit d'appeler la méthode arrêter() des deux locomotives.

Tests

Gare et priorité des trains

- ✓ Les trains attendent 5 secondes en gare (à partir de l'arrivée du deuxième train) et repartent en même temps.
- ✓ Le deuxième train a toujours la priorité sur la section partagée qui suit la gare.

Section partagée

- ✓ Il n'y a qu'un seul train à la fois qui peut se trouver dans la section partagée.
- ✓ Les trains s'arrêtent largement avant la section partagée (et prennent en compte leur inertie).

Arrêt d'urgence

- ✓ Peu importe l'état des locomotives, elle s'arrêteront immédiatement et ne peuvent pas redémarrer (même si on touche les boutons).