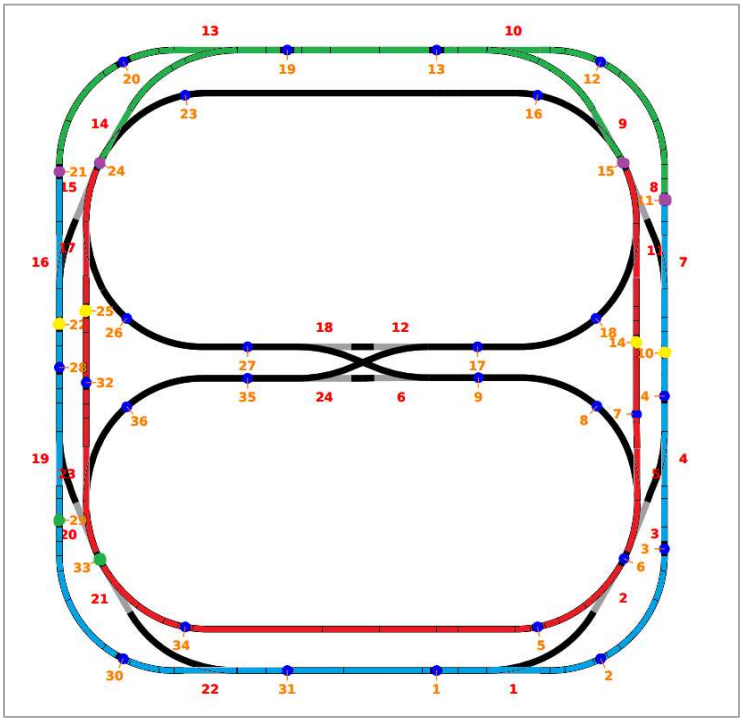
1 INTRODUCTION

Ce rapport est le rendu du laboratoire n°4 qui a eu lieu durant le cours de PCO (Programmation Concurrente) de la HEIG-VD.

Il va décrire la manière dont les fonctionnalités ont été implémentées durant les différentes étapes du laboratoire, ainsi que les tests réalisés pour vérifier que celle-ci étaient correctes. Le laboratoire avait pour but de simuler un système ferroviaire composé de deux locomotives. Il y avait certaines spécificités concernant les trajets :

- Chaque locomotive avait son propre trajet distinct de l'autre locomotive. Après N tours de circuit, la locomotive devait réaliser ce trajet dans le sens inverse.
- Il devait tout de même y avoir un tronçon de rail partagé entre les deux locomotives. Il fallait gérer l'accès à celui-ci.
- Une gare se situait sur le parcours de chaque locomotive, après M tours de trajet les locomotives devaient s'attendre en gare.

Ci-dessous, on retrouve une représentation des chemins parcourus par les deux locomotives, ainsi que les autres particularités de notre trajet.



Légendes :

- Chemin rouge → Locomotive A
- Chemin bleu → Locomotive B
- Chemin vert → Section partagée
- Points violets → Points de contrôle d'acceptation
- Point jaunes → Points de contrôle pour les requêtes
- Points Verts → Gare de chaque locomotive

2 ÉTAPE 1

Cette première étape du laboratoire consistait à réaliser un système avec une section qui était partagée pour deux locomotives. L'accès à cette session se faisait en fonction de la première locomotive qui arrivait sur le point de contact. La gare que l'on a cité précédemment dans ce rapport a aussi été implémentée durant cette étape.

2.1 Comportement de la locomotive

Cette partie implémente la manière dont la locomotive réagit lorsqu'elle arrive aux différents points de contrôle. Le comportement d'une locomotive est décomposé en 5 parties que l'on répète en boucle.

- Vérifie que l'on atteigne le point de contrôle avant notre section partagée, afin de gérer les accès concurrents. Ici, c'est la première locomotive qui arrive sur son point qui accède à la section. Elle change son aiguillage pour y rentrer.
- Réglage de l'aiguillage pour sortie de la section partagée.
- Annonce que la locomotive n'est plus dans la section partagée, c'est le point de contrôle après la section. Cela permet possiblement à l'autre locomotive qui serait bloquée à l'entrée de la section de repartir.
- Arrivée sur le point de contrôle de la gare. On vérifie si la locomotive a réalisé ses tours pour attendre en gare, si ce n'est pas le cas elle peut continuer. Sinon elle s'arrête en attendant l'autre locomotive.
- Sur le même point que précédemment, on incrémente le nombre de tours et on vérifie si celle-ci doit repartir dans l'autre sens.

2.2 Section partagée

La section partagée est divisée en deux parties, l'accès et la sortie de celle-ci.

2.2.1 Accès

L'accès se sépare en deux parties possibles :

- Si le train qui souhaite rentrer est le premier à arriver en section partagée. Alors à ce moment-ci, il va le signaler via un booléen et continuer son chemin.
- Dans le second cas, où une autre locomotive est déjà dans la section. La locomotive voulant rentrer devra attendre. Elle va donc s'arrêter et grâce à une attente en utilisant les sémaphores de la librairie PcoSynchro, elle attendra tout le temps nécessaire jusqu'à ce que l'autre locomotive termine et la relâche.

2.2.2 Sortie

La sortie est un mécanisme assez simple que l'on peut aussi couper en deux parties :

- Si une locomotive attend en entrée de section, on lui signale qu'elle peut repartir en relâchant l'attente.
- S'il n'y a personne qui attend, alors on sort simplement de la section.

2.3 Gare

La gare possède un système assez similaire à un tronçon partagé. Lorsque l'on arrive en gare en ayant réalisé les tours demandés, on peut retrouver deux cas possibles :

- Si la locomotive est la première des deux à avoir réaliser son nombre de tours, alors elle s'arrête et attend la suivante.
- Si on est la deuxième à arriver, on s'arrête aussi pendant 2 secondes. Puis, on repart et on signale à l'autre locomotive que celle-ci peut aussi repartir de son côté. Elles repartent à condition que l'arrêt d'urgence n'aie pas été pressé entre deux.

3 ÉTAPE 2

Pour cette étape, il fallait réaliser un système de priorité entre les locomotives. Celle-ci permet l'accès à la section critique pour la locomotive qui possède la priorité la plus haute ou la plus basse selon le mode de priorité choisi.

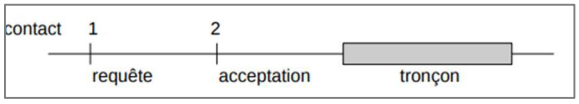
3.1 Comportement de la locomotive

La différence par rapport à l'étape précédente vient de l'ajout d'un nouveau point de contact. Il se situe un point avant le point de contact qui décide qui accède en section partagée et celui-ci permet d'annoncer que la locomotive souhaite accéder à la section. C'est à ce moment que l'on va comparer les priorités des locomotives pour savoir laquelle peut accéder à la section partagée.

On a aussi rajouter à la fin de la fonction qui gère le comportement de la locomotive un petit système pour choisir une nouvelle priorité à la locomotive et inverser le mode de priorité. Plus de détails seront donnés dans les sections correspondante plus bas.

3.2 Section partagée

Comme dit précédemment, on retrouve un point supplémentaire par rapport à la précédente étape (le point requête ci-dessous).



Lorsqu'une locomotive arrive au point de requête, il y a une comparaison entre la priorité nécessaire pour accéder actuellement à la section et la priorité de la locomotive. Selon le mode de priorité, on met à jour la priorité nécessaire avec la priorité de notre locomotive pour que ça soit elle qui puisse accéder à la section.

Pour une brève explication de ce mode de priorité, il fonctionne ainsi :

- Si le mode est en priorité basse, alors c'est la locomotive avec la priorité la plus basse qui peut accéder à la section.
- Si le mode est en priorité haute, alors c'est la locomotive avec la priorité la plus haute qui peut y accéder.

Une fois que notre locomotive arrive au second point (qui était le seul point présent dans l'étape 1), on vérifie que celle-ci possède bien la priorité définie précédemment pour rentrer en section. Sinon, comme pour l'étape 1, elle doit attendre.

3.3 Gare

Les changements de cette partie étaient surtout sur les effets provoqués par une attente en gare. Dorénavant une attente en gare de la part des deux trains provoque les effets suivants :

- Chaque locomotive voit sa priorité choisie aléatoirement entre 0 et 10.
- Le mode de priorité expliqué plus haut est inversé. Si on est en mode bas, on passe en mode haut et inversement.

4 REMARQUES

4.1 Gestion des ressources partagées

Dans ce laboratoire, nous avons dû nous soucier de certaines variables partagées entre les locomotives. Nous avons donc utiliser des verrous pour protéger les divers booléen qui permettent au programme de savoir dans quel état sont les différentes entités.

Par exemple, protéger la variable qui indique si la section partagée est déjà occupée ou si une autre locomotive attend en entré de section. On peut aussi relever le booléen qui nous permet de savoir si le mode de priorité a déjà été inversé par l'autre locomotive suite à une sortie de gare pour éviter de l'inverser deux fois et donc de revenir au point de départ.

5 TESTS

Nous n'avons pas utiliser de librairie de tests unitaires tel que GoogleTest. Cependant, nous avons essayé de faire varier de multiples paramètres lors de l'exécution du programme :

- Nombres de tours de chaque locomotive similaire au nombre de tours de gare.
- Nombre de tours de chaque locomotive différent du nombre de tours de gare.
- Priorité des locomotives définies manuellement pour être sûr que le système de mode de priorité fonctionne.
- Utilisation de l'arrêt d'urgence lors de l'exécution du programme.
- Utilisation de l'arrêt d'urgence en sortie de section critique d'une des locomotives pour s'assurer que l'autre ne reparte pas.
- Utilisation de l'arrêt d'urgence lors d'une attente en gare.
- Changement des aiguillages de la section partagés pour s'assurer que la locomotive les mettent dans le bon sens.
- Exécution du programmes pendant plusieurs minutes(environ 5-10 minutes) en s'assurant qu'aucune collision n'ait lieu.

Tous ces tests ont été réalisés avec rigueur et étaient tous fonctionnels lorsque on les a réalisés. On peut donc déterminer que le programme répond bel et bien au cahier des charges et que celui-ci est complet.