

## **PRESENTATION DU PROJET DE MODELISATION DU VAL**

Afin de pouvoir modéliser ce système de métro, nous avons eu besoin de créer deux classes, représentant respectivement une rame et une station.

Une rame connaît son nombre de passagers, son prochain arrêt, sa vitesse actuelle, sa position relative à son prochain arrêt (allant de 0 à 100 %), s'il est dans une station ou en déplacement, son numéro l'identifiant et s'il traverse les stations dans le sens croissant ou décroissant. On peut modifier sa vitesse, sa position, le nombre de passagers et son sens de parcours.

Une station connaît le nombre de passagers présents sur le quai, son numéro et si un train est présent. On peut modifier le nombre de passagers sur le quai ainsi que la présence d'un train.

D'autres fonctions sont également présentes afin de gérer la partie graphique pour fournir une visualisation plus claire à l'utilisateur. Elles gèrent l'affichage des gares, des voies, et des rames pendant leur déplacement.

Toutes les actions relatives aux rames s'effectuent au sein de threads séparés interagissant l'un sur l'autre dans certains cas. Nous avons décidé de créer une ligne de métro de 9 stations et 3 rames, opérant sans arrêt jusqu'à la fermeture du programme.

Chaque rame démarre d'une station différente, prend un nombre de passagers aléatoires à son bord (entre 1 et tous les passagers sur le quai), puis quitte la station. Elles accélèrent jusqu'à atteindre leur vitesse de croisière avant de ralentir à l'approche de la station suivante. Un nombre aléatoire de passagers descendent sur le quai et montent dans la rame avant de repartir. Le temps d'attente à chaque gare est de 3 secondes. Si une rame peut repartir mais que sa destination est occupée, celle-ci attend que la station redevienne « libre » avant de partir.

Au cours du trajet d'une rame, il y a une faible probabilité (1/1000) à chaque déplacement que celle-ci tombe en panne, entraînant un arrêt complet de la ligne jusqu'à réparation après 30 secondes.

Lorsqu'une rame atteint le terminus, celle-ci emprunte une voie pour effectuer un demi-tour et parcourir la ligne dans le sens inverse.



**Figure 1 : Rames en déplacement dans le sens croissant**

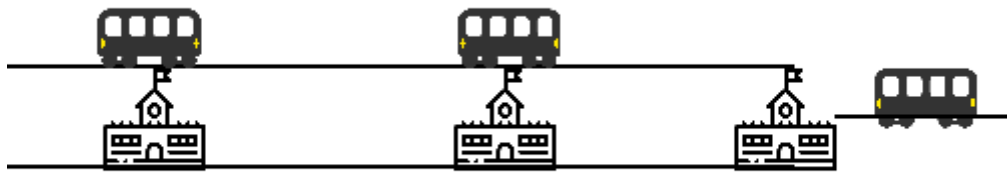


Figure 2 : Une rame effectue un demi-tour

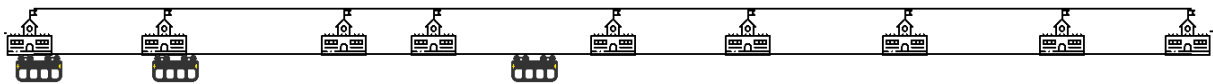


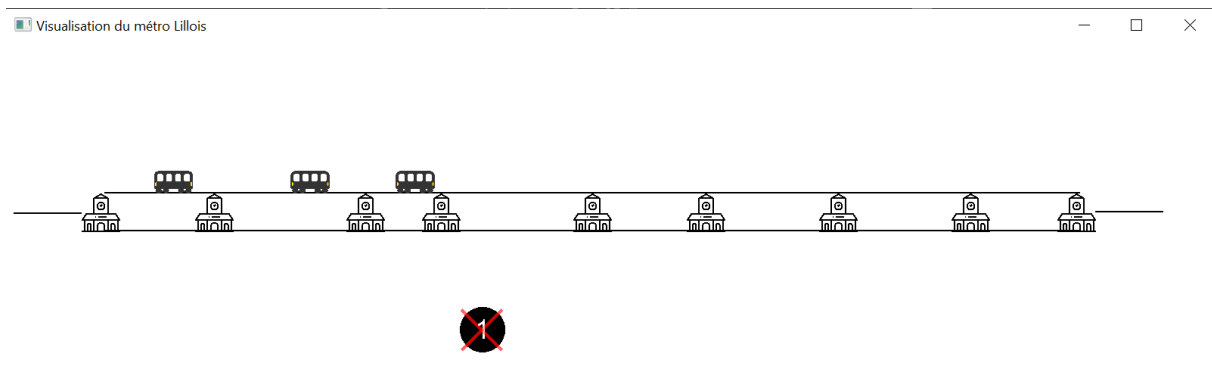
Figure 3 : Rames en déplacement dans le sens décroissant

On peut aussi suivre la progression des rames via la console, chaque rame possède sa colonne, et on y indique la progression d'une rame, son arrivée dans une station, son départ, le nombre de passagers quittant la rame et y entrant, etc.

Progression de la rame 1 : 60 %		Progression de la rame 3 : 74 %
Progression de la rame 1 : 67 %	Progression de la rame 2 : 67 %	Progression de la rame 3 : 81 %
Progression de la rame 1 : 74 %	Progression de la rame 2 : 74 %	Progression de la rame 3 : 88 %
Progression de la rame 1 : 81 %	Progression de la rame 2 : 81 %	Progression de la rame 3 : 92 %
Progression de la rame 1 : 88 %	Progression de la rame 2 : 88 %	Progression de la rame 3 : 96 %
Progression de la rame 1 : 92 %	Progression de la rame 2 : 92 %	Arrivée de la rame 3 à la station numero 6 Descente de 2 passagers de la rame 3.
Progression de la rame 1 : 96 %	Progression de la rame 2 : 96 %	
Arrivée de la rame 1 à la station numero 4 Descente de 0 passagers de la rame 1.	Arrivée de la rame 2 à la station numero 5 Descente de 2 passagers de la rame 2.	Montée de 1 passagers dans la rame 3. Départ de la rame 3 de la station 6 Progression de la rame 3 : 0 %
Montée de 0 passagers dans la rame 1. Départ de la rame 1 de la station 4	Montée de 11 passagers dans la rame 2. Départ de la rame 1 de la station 5	
Progression de la rame 1 : 0 %	Progression de la rame 2 : 0 %	Progression de la rame 3 : 4 %
Progression de la rame 1 : 4 %	Progression de la rame 2 : 4 %	Progression de la rame 3 : 11 %
Progression de la rame 1 : 11 %	Progression de la rame 2 : 11 %	Progression de la rame 3 : 18 %

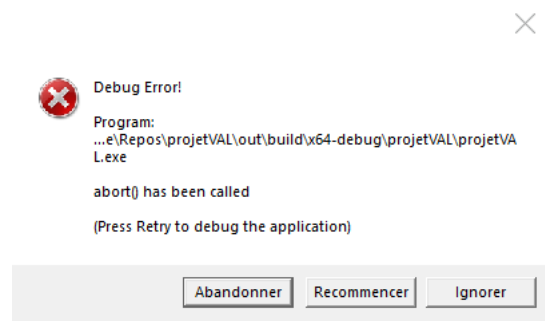
Figure 4 : Console indiquant la progression de chaque rame

Lors de chaque panne, un indicateur de Panne apparaît (différent selon le métro en panne). Cela nous permet de savoir quel métro est en panne sans devoir passer par la console.



**Figure 5 : Indicateur de Panne**

Cependant, une erreur dont nous ne comprenons pas l'origine apparaît, et nous n'avons pas réussi à la corriger. Cette erreur n'est pas présente à chaque lancement du code, et ne pose pas de problème pour son lancement, cependant, il reste préférable de relancer le code à chaque fois qu'elle apparaît.



**Figure 5 : L'erreur apparaissant certaines fois**

Travailler sur ce projet fût très intéressant, et nous a permis d'apprendre à manipuler avec une certaine efficacité le module SFML et les threads. Nous sommes satisfaits de notre projet, et même si des améliorations sont possibles (comme l'ajout d'autres rames, de plusieurs lignes avec des stations interconnectées...), nous estimons qu'il répond aux attentes énoncées dans le cahier des charges.