Colt Express
Poumier Antonin
Bernier Vincent
L2-B STN

Lors de notre travail nous avons traité les parties du sujet suivantes :

- Les déplacements dans les wagons (gauche, droite, haut ,bas).

- Les déplacements interdis, exemple : Le bandit grimpe sur le toit du wagon alors qu'il y est déjà (impossible).

- L'action qui nous permet de braquer les passagers pour récupérer un butin dans un wagon.

- Les 4 butins aléatoires parmi une bourse de 100, une bourse de 200, un bijou et un magot sont ajoutés dans chaque wagon, donc un total de 4 butins par wagon.

- Un Marshall fonctionnel placé initialement a l'intérieur de la locomotive

- Chaque action provoque un déplacement aléatoire du Marshall

-Si le Marshall rencontre un bandit, il tire sur ce dernier le forçant à fuir sur le toit. Durant la fuite du bandit touché, un de ses butins tombe dans le wagon.

- Affichage des différents éléments du train et la position du bandit/Marshall

- Affichage mis à jour après chaque action du bandit

- Affichage des actions du jeu dans une sortie de l'interface

- Affichage des butins de chaque wagon dans l'interface

- Affichage du butin total actuel du bandit sous forme de score.

- Affichage des actions sélectionnées lors de la phase de planification

-Stockage des commandes pour les exécuter les une après les autres, suite à l'utilisation d'un bouton de confirmation

-Limitation du nombre d'actions à 3

Colt Express
Poumier Antonin
Bernier Vincent
L2-B STN

- Le nombre max de tour est défini à 10

-Ajout de musique et bruitage de fin de jeu

Problèmes rencontrés et à régler

Pour le jeu :

- Pour la création des butins, le plus dur était de trouver la bonne utilisation de randint pour que chaque butin de chaque wagon soit aléatoire/différent à chaque lancement du jeu

Au départ, tous les butins se mettaient dans le premier wagon uniquement ou 5 butins dans le premier wagon et 0 dans le troisième par exemple. On a donc créé une liste contenant l'ensemble des différents butins et à chaque fois, on en choisit un au hasard et on le place dans le premier wagon (4 fois avant de changer de wagon)

- Les déplacements du bandit étaient parfaits, mais à un seul détail près. Il pouvait se déplacer à l'infini vers la gauche ou vers la droite. C'est à dire, en étant dans la locomotive, le bandit pouvait se déplacer vers la droite. Il se retrouvait donc dans le dernier wagon tout à gauche car notre train est représenté comme une liste qui contient 5 sous listes (indice 4 est la locomotive donc l'indice 5 de la liste est par défaut le dernier wagon à l'indice 0). De même, vers la gauche dans le dernier wagon.

Nous avons donc créé une limite à ne pas dépasser pour que le bandit soit bloqué quand il est dans le dernier wagon ou dans la locomotive

- Pour le stockage des actions, au début, nous ne savions pas comment trouver un moyen de conserver les commandes entrées par le joueur pour les exécuter par la suite les une après les autres. Nous les avons au final stockées dans une liste. La seconde difficulté était de retranscrire les instructions contenues dans la liste pour les exécuter. Nous avons donc écrit des foncions qui vérifient le contenu de chaque indice de la liste pour exécuter les fonctions qui y correspondent.

Pour l'intelligence artificielle du Marshall :

Colt Express
Poumier Antonin
Bernier Vincent
L2-B STN

-Il ne se déplaçait que de droite à gauche puis de gauche à droite du train à chaque fois, ses mouvements étaient donc très prévisibles. Pour régler cela, le Marshall a un choix de déplacements prédéfinis car au départ, il est dans la locomotive du train donc, il ne peut se déplacer que vers la gauche. De même, pour la fin du train (vers la droite). Pour les autres cas, le Marshall est libre de choisir s'il veut se déplacer à gauche ou à droite. Grâce à la fonction randint, le programme va pouvoir choisir soit 0 soit 1 pour le déplacement du Marshall (0 pour la gauche et 1 pour la droite). Maintenant, ses déplacements sont beaucoup moins prévisibles.

Pour l'interface :

Le plus gros problème était lors de l'affichage des déplacements du bandit et du Marshall.

Il fallait gérer des coordonnées à chaque fois qu'il y avait un déplacement pour afficher correctement l'icône de chaque personnage. Le bandit avait tendance à sortir complètement du train et a se décaler totalement par rapport au train de l'interface. Donc, à chaque fois que la coordonnée en x de l'icône dépasse la coordonnée en x de départ du bandit, on replace l'icône au bon endroit. Et inversement, pour l'autre coté.

Une difficulté plus mineure a était de déterminer la meilleure option pour afficher le compte rendu des actions qui s'exécutent dans l'interface. Par défaut, une zone de texte a été utilisée, mais donnait la possibilité à l'utilisateur de manipuler le texte dedans. Nous avons finalement opté pour une listbox qui nous permet d'afficher les actions une à une de façon à ce que l'utilisateur ne puisse plus écrire dessus.

Problèmes rencontrés non réglés :

De temps en temps, lors du braquage, l'icône du Marshall atteint celle du Bandit (même wagon) mais le jeu ne le détecte pas et le programme continue à fonctionner comme si de rien n'était.

Code Emprunté :

Colt Express
Poumier Antonin
Bernier Vincent
L2-B STN

Le code qui permet le switch entre les frames du projet est celui du programme «frame_switch.py» que vous nous avez fourni dans les fichiers de code PI_Exemples