

Projet d'Algorithmique et de Développement Logiciel
Les Aventuriers du Rail

Incrément 3
Séances 12 et 13

Sommaire

1	Préambule	3
2	Évaluation du travail	3
3	Travail demandé	3
3.1	Préparation de l'environnement	3
3.2	La classe Route	4
3.2.1	Modifier la classe Route	4
3.2.2	Accesseurs	4
3.2.3	Vérifier l'implémentation de la classe Route	4
3.3	La classe Jeu	4
3.3.1	Méthode selectionRoute	5
3.3.2	Vérifier l'implémentation de la méthode <code>selectionRoute()</code>	5
3.3.3	Modifier les actions du joueur	6
3.3.4	Exécution du jeu	6
4	Analyse du code ADR	6
5	Fin de l'incrément 3	7
6	Extensions	7
6.1	Nombre d'actions par tour	7

1 Préambule

L'incrément 3 de PADL s'appuie sur la version produite pendant l'incrément 2 du jeu des Aventuriers du Rail (ADR).

L'objectif principal du troisième incrément est d'implémenter une version simplifiée de la prise de possession d'une route, c'est-à-dire, permettre à un joueur de s'approprier une route du plateau de jeu et ainsi empêcher un autre joueur de prendre cette route.

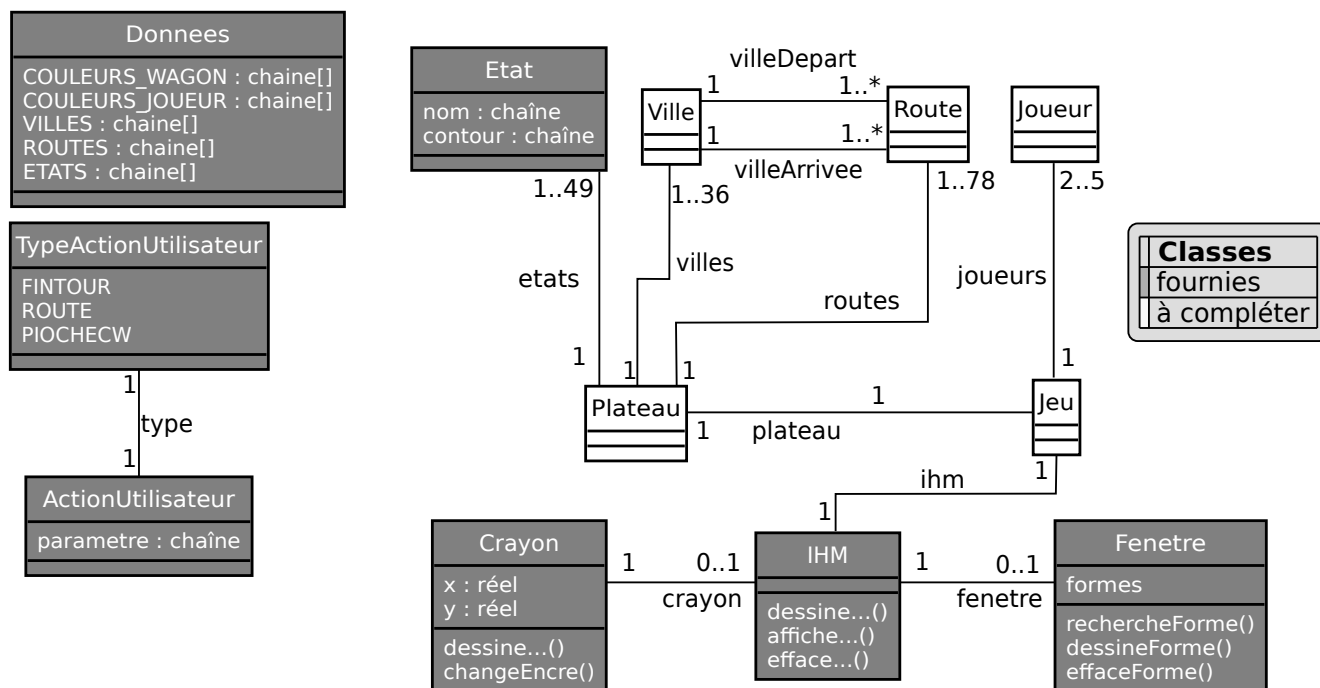


Figure 1: : Classes identifiées dans le jeu des ADR - Incrément 3

Pour implémenter cette fonctionnalité, nous nous appuyons sur les classes **Route**, **Plateau** et **Jeu** déjà implémentées.

2 Évaluation du travail

Le troisième incrément consolide vos compétences en programmation Java en général.

3 Travail demandé

Prendre possession d'une route sur le plateau de jeu implique de modifier le code existant pour (1) associer un joueur à une route dont il devient propriétaire, (2) détecter et réagir à l'évènement de « prise de route » sur l'interface graphique et (3) afficher un résultat.

Pour ce faire, la classe **Route** est modifiée pour stocker le joueur propriétaire d'une route. Il est également nécessaire de modifier la classe **Jeu** pour que le joueur puisse, sur l'interface graphique, sélectionner une route dont il veut devenir propriétaire.

Au cours de la réalisation de ce troisième incrément, le diagramme de classes fourni dans l'incrément précédent est complété avec les informations nécessaires.

On utilise l'objet **Joueur** et des informations supplémentaires pour identifier le joueur qui joue et « passer au joueur suivant ».

3.1 Préparation de l'environnement

Importez l'incrément 3 dans votre projet existant (incréments 1 et 2) en suivant la procédure décrite dans la documentation du logiciel DevCube.

3.2 La classe Route

La prise de possession d'une route permet de lier un joueur à une route dont il est propriétaire. On modifie la classe Route pour stocker cette information.

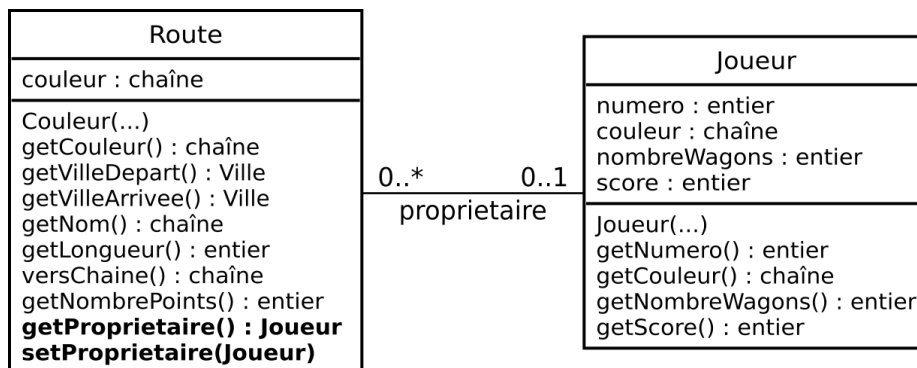


Figure 2: Diagramme de classe de la classe Route

3.2.1 Modifier la classe Route

Travail à faire

Modifier la définition de la classe **Route** telle qu'elle est représentée dans la figure 2).

3.2.2 Accesseurs

Travail à faire

Implémenter l'accesseur et le mutateur (accesseur en écriture) de la classe **Route** (voir figure 2) pour le nouvel attribut **proprietaire**.

3.2.3 Vérifier l'implémentation de la classe Route

Travail à faire

Exécuter la classe **TestRouteProprietaire** pour valider l'implémentation des modifications apportées à la classe **Route**. Si vous rencontrez des erreurs pendant la compilation ou l'exécution, corrigez-les avant de poursuivre.

3.3 La classe Jeu

Dans cet incrément, la classe **Jeu** est modifiée pour réagir à la sélection d'une route par le joueur courant et mettre à jour les informations de la route sélectionnée.

Il est nécessaire de :

1. capturer l'évènement de sélection d'une route du jeu (voir Section 3.3.9 de l'incrément 2).
2. implémenter une nouvelle méthode (voir figure 3) pour réagir à la sélection d'une route.

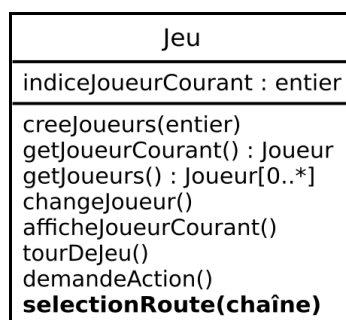


Figure 3: Diagramme de classe de la classe Jeu

3.3.1 Méthode selectionRoute

La méthode `selectionRoute()` a pour objectif de modifier le propriétaire de la route sélectionnée par le joueur courant.

L'algorithme de cette méthode est défini par le diagramme d'activité présenté en figure 4.

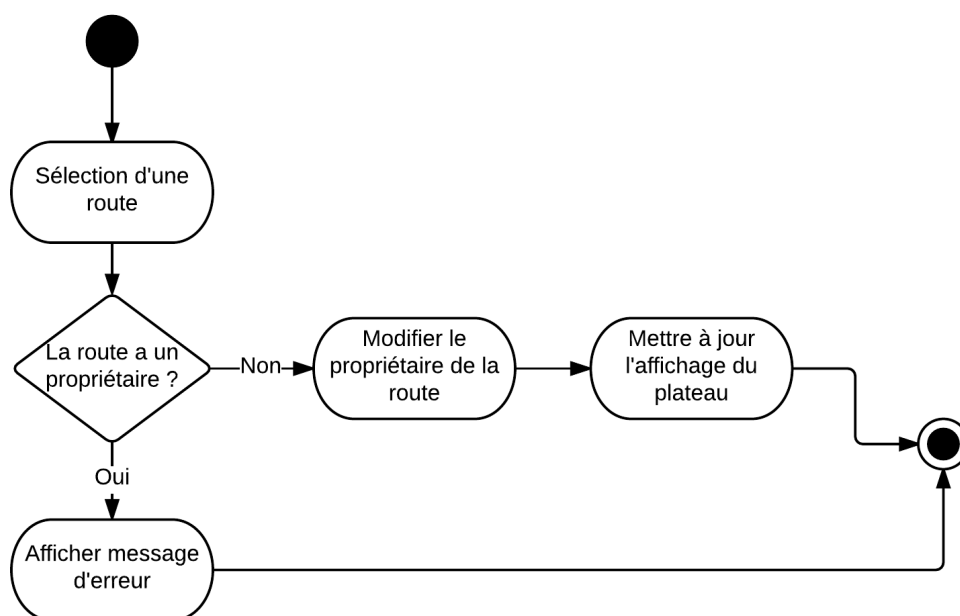


Figure 4: Algorithme de la sélection d'une route par le joueur courant

Travail à faire

Écrire la méthode `selectionRoute()` suivant le diagramme d'activité fourni. La « sélection d'une route » est une demande réalisée au travers de la méthode `demandeAction()`. L'affichage d'un message d'erreur sur l'interface graphique est réalisé par la méthode `afficheMessage()` de la classe **IHM**. La mise à jour du plateau de jeu consiste à changer l'affichage de la route sélectionnée par le joueur (voir figure 5). Pour afficher correctement la route modifiée, penser à redessiner les villes de la route après avoir redessiné la route.

3.3.2 Vérifier l'implémentation de la méthode selectionRoute()

Travail à faire

Compléter la classe **TestJeuSelectionRoute** avec les méthodes nécessaires pour valider l'implémentation de la méthode **selectionRoute()**.
Si vous rencontrez des erreurs pendant la compilation ou l'exécution, corrigez-les avant de poursuivre.

3.3.3 Modifier les actions du joueur

On offre la possibilité au joueur de sélectionner une route sur le plateau de jeu. Pour ce faire, la méthode **demandeAction()** est modifiée pour capturer l'évènement de sélection d'une route sur l'interface graphique.

Travail à faire

Attendre un évènement de la part du joueur.
La méthode **demandeAction()** renvoie l'évènement « ROUTE » lorsque le joueur clique sur une route du plateau et appelle la méthode **selectionRoute()** pour traiter cet évènement.

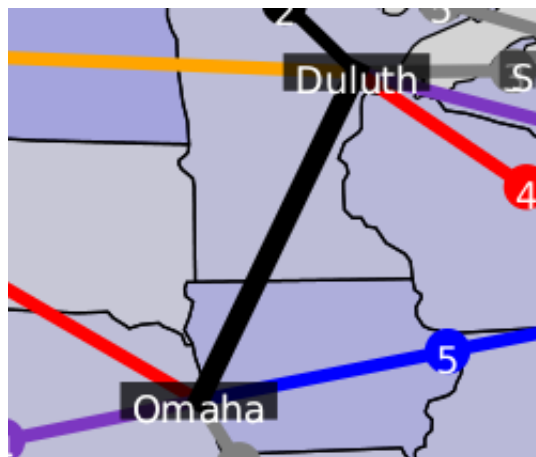


Figure 5: La route sélectionnée par le joueur prend la couleur du joueur courant et sa longueur est automatiquement supprimée sur l'interface graphique.

3.3.4 Exécution du jeu

Travail à faire

Exécuter la classe **Jeu** et observer le résultat de votre implémentation. En cas de problème, vérifier les implémentations du joueur et du jeu pour effectuer des corrections si nécessaire.

4 Analyse du code ADR

Pour concrétiser et mieux comprendre le fonctionnement des actions et de l'interface graphique, il est demandé de fournir une analyse et une conception préliminaire des classes **IHM** et **Jeu**. Un travail a déjà été réalisé sur la structure de ces classes, l'analyse va désormais porter sur le fonctionnement.

Travail à faire

Proposer un diagramme d'activités, similaire à la figure 4 pour expliquer le fonctionnement des méthodes **demandeAction()** de la classe **Jeu** et **dessineRoute()** de la classe **IHM**.

5 Fin de l'incrément 3

Vous avez atteint le principal objectif qui est de pouvoir sélectionner une route du plateau de jeu et d'y affecter un propriétaire.

En fonction du temps qu'il vous reste avant l'incrément suivant, vous pouvez :

- Réaliser les extensions de l'incrément 3 (optionnel - section 6)

6 Extensions

6.1 Nombre d'actions par tour

Dans les règles du jeu des ADR, un joueur ne peut réaliser qu'une action par tour. Dans la version actuelle des incréments 2 et 3, le nombre d'actions du joueur dans un tour n'est pas limité : il peut sélectionner plusieurs routes avant de « passer son tour ».

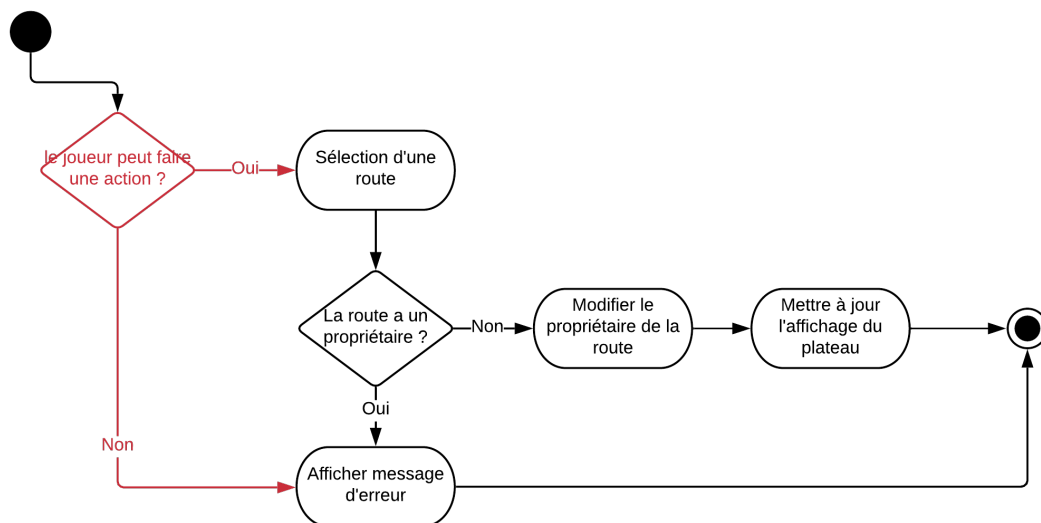


Figure 6: Extension (en rouge) de l'algorithme de sélection d'une route par le joueur courant.

Travail à faire

Proposer un mécanisme pour limiter le nombre d'actions d'un joueur pendant son tour (voir figure 6).