

Encre, Routes & Rails : Objectif Réseaux

Projet transversal
License informatique S6 2024 – 2025

Situation

Le club *Fantastiques Informatiques et Ludiques*¹ organise un tournoi autour d'un jeu de société. Pour planifier les parties, suivre les victoires et défaites des participants, et établir le classement final, le club utilise un logiciel de gestion de tournoi indépendant du jeu.²

Cependant, **le tournoi est informatisé** : les parties sont virtuelles, on y joue par l'intermédiaire d'un service de type *file de messages*. Le club encourage les participants à développer leurs propres programmes de connection au service, et autorise même les algorithmes d'assistance au jeu.

Cette année, le club a sélectionné le jeu **Railroad Ink**, édité par Horrible Guild :

<https://horribleguild.com/eu/product/railroad-ink-deep-blue/>

C'est un jeu de placement de tuiles, jouable en solo ou à plusieurs. À chaque tour, des tuiles portant des sections de route ou de train sont tirées aux dés ; chaque joueur dessine ces tuiles sur les espaces libres de sa carte, en respectant des contraintes de placement. Le score découle des caractéristiques du réseau routier et ferroviaire obtenu après un nombre de tours prédéfini —voir le site de l'éditeur pour les règles complètes.

Le succès du tournoi repose sur l'interopérabilité des programmes de jeu des participants, entre eux et avec la suite d'outils des animateurs du tournoi. Le club prépare donc une spécification technique, qui décrit l'architecture générale du système, le protocole de communication, et un format de persistance.

Votre équipe est chargée de proposer et implémenter une **suite d'outils cohérente** pour les tournois de Railroad Ink. Vous avez accès à une version préliminaire de la spécification qui sera fournie aux participants. Pour mener à bien ce projet, vous devrez appliquer les compétences théoriques et pratiques acquises jusqu'ici dans la licence, mais aussi organiser votre travail en équipe.

¹ Toute ressemblance avec un acronyme existant serait purement fortuite.

² ...grâce aux efforts d'une promotion L3 précédente.

Architecture technique

Réflecteur Chaque partie est représentée par un service appelé **réflecteur** auquel des programmes clients se connectent par une WebSocket. Les communications ont lieu exclusivement par l'intermédiaire du réflecteur, qui n'a aucune connaissance du jeu : il accepte les connections et diffuse tout message *valide sur la forme*. Les clients doivent reconnaître et valider sémantiquement les messages spécifiés pour le jeu (Tab. 1).

Séquencement global Chaque client maintient sa propre vision de la partie, et émet des messages à sa propre initiative. Cependant, un client ne peut pas immédiatement prendre en compte ses propres messages : d'autres messages pourraient avoir été émis mais pas encore reçus.

C'est le rôle fondamental du réflecteur, qui diffuse les messages dans l'ordre où il les reçoit, garantissant ainsi que les clients reçoivent tous *exactement* la même séquence de messages, quelque soit le moment où ils se sont connectés, y compris les messages échangés avant leur connection. C'est le **rang** des messages dans cet **ordre global unique** imposé par le réflecteur qui fait foi ; le premier message a le rang 1.

Format de transmission Le format des messages sur la WebSocket client-réflecteur est assez simple mais strict :

- mode texte (pas de message binaire)
- un, et un seul, message de données WebSocket par message applicatif
- le texte du message alterne des séquences de caractères de deux types :
 - *mots* : lettres minuscules ou majuscules, chiffres, caractère souligné
 - *blancs* : espace, tabulation, retour chariot, ou retour à la ligne
 - tout autre caractère est une faute de protocole³
- le premier mot du message est un identifiant attribué à l'émetteur

Le réflecteur diffuse tout message respectant le format de transmission. La charge utile sémantique du message est la séquence des mots, identifiant compris. Les clients sont responsables de valider *sémantiquement* les messages qu'ils reçoivent, et d'ignorer ceux qui sont incomplets ou incohérents (Tab. 1).

Persistance Le format de persistance est un simple fichier texte avec un message par ligne. Les lignes constituées de blancs ou dont le premier caractère visible est # sont autorisées mais ignorées (commentaires).

Client passif Quand un client se connecte au réflecteur, celui-ci commence par lui envoyer l'historique des messages déjà échangés par les éventuels autres clients, puis continue avec la diffusion des messages en temps réel. Immédiatement après connection, le client est *passif*, et peut le rester indéfiniment ; il reçoit les messages diffusés mais ne peut pas en émettre. Comme il ne contribue pas à l'historique de message, un client passif est invisible aux autres. Des clients passifs peuvent superviser une partie sans l'influencer : sauvegarde, affichage en temps réel...

Client actif Pour devenir participant *actif*, un client doit annoncer son identifiant avec un message ENTERS ; c'est l'unique message qu'un client anonyme a le droit d'émettre. Un client —plus précisément un identifiant— est considéré actif dès la diffusion de son message de déclaration. Ce message peut être envoyé immédiatement après connection, mais ne sera reçu qu'à la suite des éventuels messages de rang inférieur dans l'historique.

³ Le réflecteur fourni est *censé* être robuste, et sera punitif envers les clients ne respectant pas le format de transmission —déconnection immédiate.

contexte	émetteur, mot clé, paramètres du message	signification
identifier	<i>id</i> ENTERS <i>id</i> LEAVES	client actif client passif
jouer	<i>id</i> THROWS <i>tuile</i> + <i>id</i> PLACES <i>tuile orientation cellule</i> <i>id</i> YIELDS	tirage de dés placement d'une tuile inaction, cession d'initiative
juger	<i>id</i> SCORES <i>rang score points</i> <i>id</i> BLAMES <i>rang</i>	attribution d'un score contestation d'un message
organiser	<i>id</i> GRANTS <i>id2 message</i> <i>id</i> AGREES <i>règle</i> + <i>id</i> ELECTS <i>joueur</i> +	droit d'émettre <i>message</i> choix d'une variante de jeu nomination des participants

Tab. 1 Syntaxe des messages autorisés. Le premier mot est toujours l'identifiant de l'émetteur, et le deuxième un mot-clé qui détermine comment interpréter les mots suivants. Les paramètres sont tous obligatoires, certains peuvent être répétés (suffixe +).

Déconnection La déconnection d'un client passif n'a d'impact que sur le réflecteur, et n'est pas perceptible par les autres clients. Un client peut se déconnecter en fermant la WebSocket de lui-même. Si un client commet une **faute de protocole**, c'est le réflecteur qui le déconnecte. Dans les deux cas, si le client déconnecté était actif, le réflecteur diffuse un message LEAVES.⁴

Identifiants Les messages ENTERS et LEAVES sont les seuls que réflecteur traite spécifiquement : pour empêcher les usurpations d'identifiants, il associe chaque identifiant à la connection du client correspondant. Tout message dont l'identifiant n'est pas associé à la connection par laquelle il provient constitue une faute de protocole.

Le message de déconnection d'un client actif pourrait signifier que cet identifiant est de nouveau disponible. Le joueur ou programme pourrait alors reprendre sa participation depuis une nouvelle connection en redéclarant le même identifiant... à discuter.

Messages spécifiques à Railroad Ink

Tirage des tuiles Le message THROWS annonce le tirage d'un ou plusieurs dés. Chaque occurrence du paramètre *tuile* est l'un des codes indiqué en Fig. 1.

Placement Le message PLACES exprime le placement d'une *tuile*, dans une *orientation* particulière, sur une *cellule* de la carte.

Le paramètre *orientation* exprime la rotation dans le sens trigonométrique, horaire, ou d'un demi-tour, à partir de l'orientation de référence ou d'une symétrie d'axe vertical de celle-ci (Fig. 2). Si une tuile a un dessin équivalent dans plusieurs orientations, toutes ces orientations sont autorisées et valides —voir par exemple la tuile HH qui est équivalente dans toutes ses orientations, par opposition à Sc qui est chirale.

Le paramètre *cellule* adopte la notation des échecs : une lettre de A à G pour la colonne, suivie d'un chiffre de 1 à 7 pour la ligne, en partant du coin inférieur gauche (Fig. 3).

⁴ On pourrait implémenter le réflecteur de façon qu'un client redevienne passif si il émettant son propre message de départ... mais quelle en serait l'utilité?

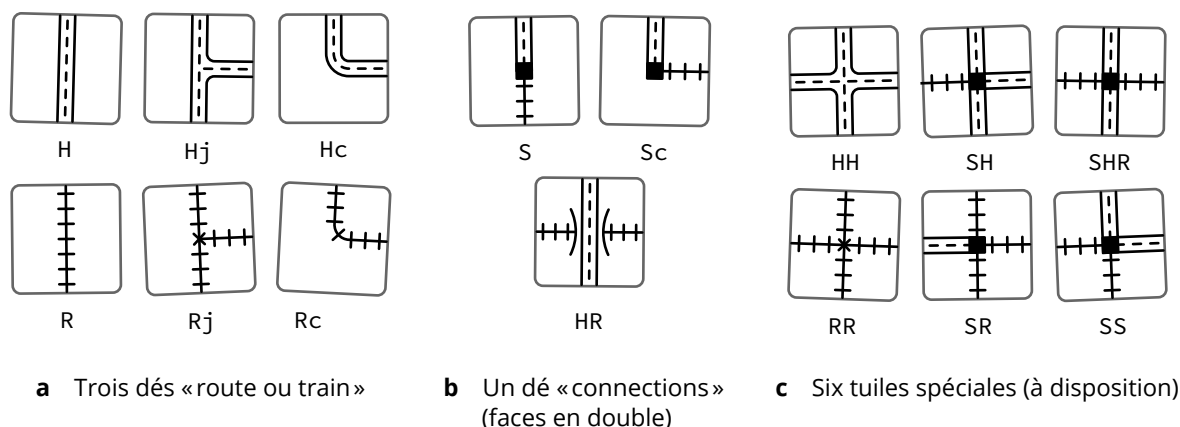


Fig. 1 Nomenclature des tuiles dans leur orientation de référence.
Le nommage est arbitraire, mais basé sur les mots *Highway, Railway, Station, junction, curve*.



Fig. 2 Nomenclature des orientations à partir de l'orientation de référence et de sa symétrie d'axe vertical. Mnémonique : *straight* ou *flipped*, *left*, *right*, *upside-down*.

Jugements Les messages SCORES et BLAMES font référence à une action de jeu désignée par le *rang* du message correspondant dans le séquençement global. Ces messages n'impactent pas le déroulement de la partie proprement dite, mais peuvent servir à l'équipe arbitrale soit pour automatiser l'arbitrage avec un client certifié, soit pour départager les joueurs selon la pertinence des messages qu'ils envoient.

Le message SCORES doit porter le *rang* d'un message PLACES; le code *score* indique à quel titre on attribue les *points* (un entier en notation décimale) : réseaux / sorties connectées (N), plus longue autoroute (H) ou ligne de chemin de fer (R), occupation du carré central (C), pénalités d'adjacence (X), points d'extension (E) ou score TOTAL.

Le message BLAMES conteste l'action de jeu référencée, par exemple un message PLACES, quand la tuile ou son placement seraient contraire aux règles, ou un message YIELDS émis par un client qui a encore une action imposée à ce moment de la partie. Le message THROWS est également contestable car le tirage doit être cohérent avec les dés existants (Fig. 1); par exemple, il est impossible de tirer deux tuiles HR au même tour.

Restrictions Les messages GRANTS, AGREES, et ELECTS servent à organiser la partie. Leurs effets sont définitifs et ne peuvent pas se contredire.⁵

Le message GRANTS donne l'exclusivité d'émettre le type de *message* donné à *id2*. Tout message de ce type provenant d'un autre identifiant doit être ignoré. Lors du tournoi, un client *animateur de partie* pourra être connecté au réflecteur avant que l'adresse de la socket soit rendue publique, et limiter l'émission de certains messages à l'un de ses propres identifiants, par exemple.

Le message AGREES désigne la *règle* ou l'ensemble de règles avec lesquelles le jeu se déroulera. Les codes de règles désignent les dés d'extension (*rivers* ou *lakes* pour

⁵ ...donc AGREES et ELECTS ne peuvent pas être répétés légitimement, et GRANTS ne peut l'être que pour un autre type de *message*.

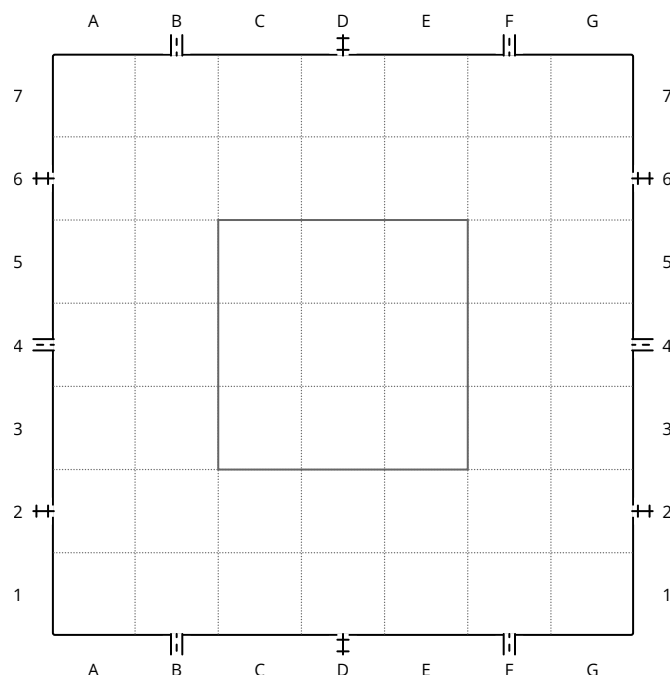


Fig. 3 Carte de jeu vierge avec indication des coordonnées, des sorties, et du carré central.

l'édition bleue), les bâtiments spéciaux et les objectifs supplémentaires dans le cas d'une partie de l'édition *challenge*.

Le message ELECTS désigne les participants de la partie; seuls ces identifiants seront légitimes à émettre THROWS, PLACES, ou YIELDS.

Besoins

Les participants vont eux-mêmes développer des clients de jeu, c'est même l'intérêt du tournoi; il est donc inutile pour le projet d'en développer un, sauf à titre expérimental.

Le club FIL a quelques autres suggestions de fonctionnalités, à creuser :

- des implémentations **en divers langages** du côté client du protocole, à titre d'exemple pour les participants, voire d'implémentation de référence;
- calcul du **score optimal** atteignable pour une série de tirage de dés connue;
- écoute et persistance d'une partie, ou *replay* de messages persistés;
- examen d'une partie et classement des clients actifs selon leurs réactions;
- client arbitral qui s'accapare le message THROWS pour contrôler les tirages de dés;
- visualisation graphique d'une partie, pour le public...

Le club attend également de votre groupe que vous mettiez à l'épreuve le protocole, et considérera toute proposition faite dans l'optique de rendre le tournoi plus intéressant pour les participants. Par exemple il est possible qu'une mauvaise interaction entre clients bloque toute une partie; il faudrait détecter cela, interrompre le réflecteur, et probablement pénaliser le ou les participants.

Évaluation et livrables

Hébergement du code source Pour travailler sur le projet vous créerez un groupe Gitlab qui contiendra le ou les dépôts Git de votre suite logicielle. Ce groupe doit être maintenu comme s'il s'agissait d'une vitrine publique du projet :

- fichiers README de groupe et de dépôt facilitant la découverte du projet,
- gestion de dépendances par un outil standard et validé par l'intégration continue,
- historique et messages de commit cohérents,
- exploitation des fonctionnalités collaboratives (issues, pull-requests).

Une analyse de code permettra de retrouver les bonnes pratiques de développement et de travail collectif : faible complexité cyclomatique, couverture de tests, principes de conception et de modularité.

Dossier rétrospectif Ce rapport synthétique et factuel étaye les arguments présentés dans la soutenance par des éléments concrets.

Pour chaque composant de votre suite logicielle :

- le ou les cas d'utilisation auxquels ce composant répond,
- les packages et principales classes implémentant ce composant,
- une indication de l'effort de développement dédié à ce composant : nombre de classes, de tests, de lignes de code, de commits, de personnes impliquées...

Pour chaque membre du groupe :

- les rôles assumés au sein du groupe (avec une chronologie si c'est pertinent),
- quelques éléments concrets démontrant cette implication,
- un retour d'expérience personnel sur le déroulement du projet, vos points de satisfaction et les difficultés rencontrées.

Soutenance technique Cette présentation collective vous permettra d'expliquer quels besoins que vous avez choisi de couvrir, et de montrer que votre suite logicielle fonctionne, qu'elle résulte d'un effort collectif, et qu'elle est bien conçue. Votre présentation doit donner envie au club FIL, représenté par votre encadrant, de faire confiance à votre collectif et d'investir dans vos outils pour organiser son tournoi.

Vous donnerez également un retour critique sur l'expérience de pilotage du projet et les axes de progrès identifiés. Tous les membres doivent être en mesure de représenter le groupe, par exemple en répondant à des questions sur le travail des autres membres.