

Ce challenge est basé sur un système de ligues.

Pour ce défi, plusieurs ligues pour le même jeu sont disponibles. Une fois vos compétences prouvées contre le premier Boss, vous accéderez à une ligue supérieure et des règles supplémentaires seront disponibles.

Dans les premières ligues, votre soumission affrontera uniquement le boss dans l'arène. Gagnez au moins 3 fois sur 5 pour avancer.

🎯 Objectif

Les deux premières ligues ont un **objectif** spécial à atteindre. La partie complète se jouera, mais vous ne pouvez gagner qu'en remplissant l'objectif. Une fois fait, vous pourrez commencer à travailler sur votre bot complet.

📖 Quête annexe

Un classement secondaire **alternatif** est calculé pour ceux qui choisissent de jouer un peu différemment : pouvez-vous construire des voies pour créer des chemins vers des lieux intéressants au lieu de vous battre pour la première place ?

Plus de détails dans la ligue finale.

🎯 Objectif de ligue 1 :

Reliez deux **villes** pour former une **connexion ferroviaire active** et gagner instantanément la partie.

Votre adversaire **Boss** dans l'arène passera ses tours. Si vous ne parvenez pas à former une connexion en moins de **100** tours, vous perdrez. Gagnez au moins **3** fois sur **5** pour progresser à la ligue suivante.

✓ Règles

Dans ce jeu, les deux joueurs utilisent de la **peinture** pour tracer des voies ferrées sur une carte **magique**. Relier les villes de la carte apportera la prospérité à votre propre monde.

La carte est représentée dans le jeu par une **grille**.

🗺 Carte

La grille est composée de cases qui peuvent être de quatre types :

- Type **0** pour les **plaines**.
- Type **1** pour les **rivières**.
- Type **2** pour les **montagnes**.
- Type **3** pour un **lieu d'intérêt**.

La grille est divisée en **régions**. Chaque région est composée de plusieurs cases voisines. Chaque région possède un **regionId** (identifiant de région) unique. Les régions sont sensibles à la **perturbation** par les joueurs. *Plus d'informations sur la perturbation dans la ligue suivante.*

Certaines régions contiendront une **ville**. Les villes ne peuvent se trouver que sur des cases de **plaine**.

🏠 Villes

Chaque partie commence avec plusieurs **villes** placées aléatoirement sur la carte. Il y en aura au maximum une par **région** et deux régions partageant une frontière ne contiendront jamais toutes deux une ville.

Chaque ville possède un **townId** (identifiant de ville) unique.

Chaque ville aura une liste de **desiredConnections** (connexions souhaitées) : une liste d'identifiants de villes représentant toutes les **autres villes** auxquelles cette ville souhaite être **connectée** via des **rail** placées par les joueurs.

Fournir à une ville des **rails** la reliant à une ville souhaitée est la façon dont les joueurs marquent des **points**.

Les connexions souhaitées sont **unilatérales**. Cela signifie que si la ville **0** apparaît avec une connexion souhaitée vers la ville **1**, la ville **1** **ne voudra pas** se connecter à la ville **0**.

Une ville peut avoir **zero** **desiredConnections** (connexions souhaitées), mais sera toujours l'objet d'au moins une autre ville ayant des **desiredConnections**.

Placement des rails

À chaque tour, les joueurs reçoivent **3 points de peinture** qu'ils peuvent utiliser pour placer des **rails** sur la carte. **!** Ces points ne se reportent pas au tour suivant et seront perdus s'ils ne sont pas utilisés.

Coût :

- **1** point de peinture pour placer un rail sur les **plaines**.
- **2** points de peinture pour placer un rail sur une **rivière**.
- **3** points de peinture pour placer un rail sur les **montagnes**.
- **3** points de peinture pour placer un rail sur un **lieu d'intérêt**.

Le **owner** (propriétaire) d'un rail est le **playerId** (**0 - 1**) du joueur qui l'a placée.

Si les deux joueurs placent un rail au **même tour** et au **même endroit**, le **owner** du rail sera **2**, indiquant un rail neutre.

un rail ferrée **ne peut pas** être placée sur une **ville** ni sur un rail existante.

Une fois placée, un rail ferrée se **connectera automatiquement** aux autres **rails** et **villes** adjacentes orthogonalement.

Connexions

Pour chaque paire de villes dont l'une possède l'autre dans ces **desiredConnections** (connexions souhaitée), s'il existe au moins un **chemin** entre les deux, le plus court de ces **chemins** devient la **connexion active** entre ces villes.

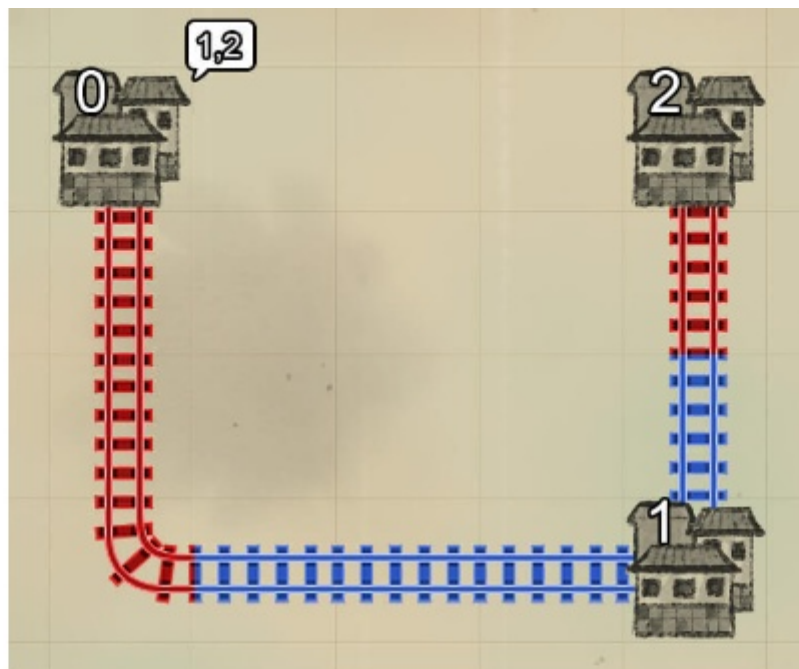
Un **chemin** est une séquence ininterrompue de cases adjacentes orthogonalement contenant un **rail** ou une **ville**.

S'il y a **plusieurs** chemins le plus court possible, le chemin choisi **privilégiera** toujours la direction dans cet ordre lorsqu'on part de la ville demandeuse vers la ville souhaitée :

1. **NORTH**
2. **EAST**
3. **SOUTH**
4. **WEST**

À la fin de chaque tour, chaque **connexion active** rapportera **1** point à chaque joueur pour chaque rail qu'il **possède** dans le **chemin**.

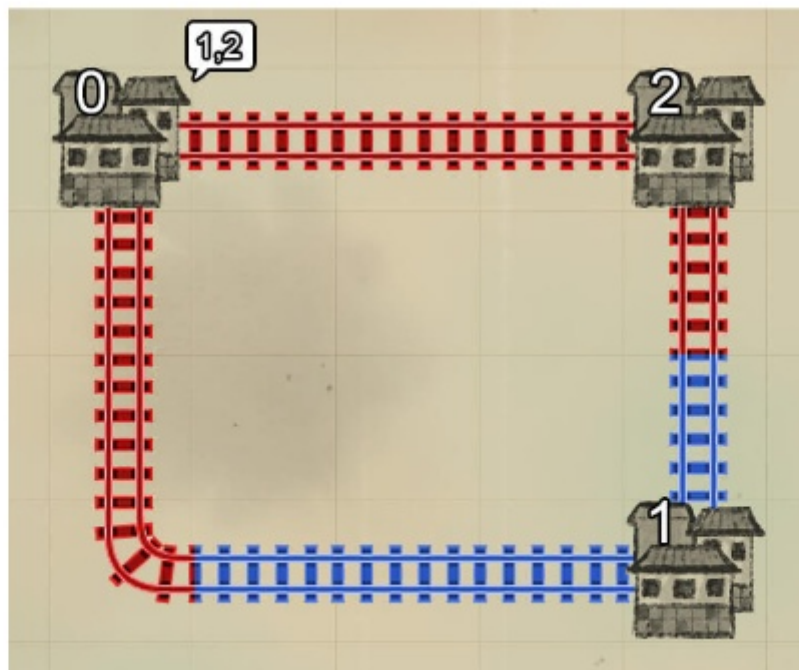
Exemple 1 :



Ici, il y a une **connexion active** de la ville 0 à la ville 1 et une autre de la ville 0 à la ville 2.

Les joueurs rouge et bleu gagneront tous deux 3 points pour la connexion 0-1 et 4 points pour la connexion 0-2 à la fin du tour.

Exemple 2 :



Ici, seul le plus court chemin de 0 à 2 est utilisé pour la connexion 0-2. Cela signifie que le joueur rouge gagnera 3 points pour cette connexion, et le joueur bleu n'en gagnera aucun.

Il existe deux chemins de même longueur entre 0 et 1, mais comme **EAST** a une priorité plus élevée que **SOUTH**, le chemin choisi sera celui qui passe par la ville 2. Le joueur rouge gagnera 4 points pour cette connexion, et le joueur bleu n'en gagnera que 1.

Actions

À chaque tour, les joueurs doivent fournir au moins une action sur la sortie standard.

Les actions doivent être séparées par un point-virgule ; et doivent être l'une des suivantes :

- `PLACE_TRACKS x y` : placer un rail sur une case libre.
- `AUTOPLACE fromX fromY toX toY` : génère automatiquement une liste d'actions pour le chemin **le moins coûteux** des coordonnées `from` au coordonnées `to` en termes de points de peinture. Cela ne fera rien si un chemin existe déjà.
Les actions générées remplacent cette commande.
- `WAIT` : ne rien faire.



Conditions de victoire

Former une **connexion active** et marquer n'importe quel nombre de points.



Conditions de défaite

- Votre programme ne fournit pas de commande dans le temps imparti ou l'une des commandes est invalide.
- Vous ne complétez pas l'objectif en moins de **100** tours.



Conseils de débogage

- Survolez la grille pour voir des informations supplémentaires sur la case sous votre souris.
- Appuyez sur l'icône d'engrenage du visualiseur pour accéder à des options d'affichage supplémentaires.
- Utilisez le clavier pour contrôler l'action : barre d'espace pour lire/mettre en pause, flèches pour avancer d'une image à la fois.

Cliquez pour développer

 Protocole du jeu

Entrée d'initialisation

Pour Démarrer

Pourquoi ne pas se lancer dans la bataille avec l'un de ces **IA Starters**, fournis par l'équipe :

- C++ <https://gist.github.com/CGjupoulton/bbd4b720ae7e0f1e5f2e970bb42ed066>
- JavaScript <https://gist.github.com/CGjupoulton/279a0f6ce4995bf4c55391fd40ae8bff>
- Java <https://gist.github.com/CGjupoulton/17397683833e324a7b8c7c7c642239d3>
- Python <https://gist.github.com/CGjupoulton/5a73bbd1142af98c6ca6887648b07cc2>

Vous pouvez les modifier selon votre style, ou les prendre comme exemple pour tout coder à partir de zero.