

# DOCUMENTATION:

## **Bonnes pratiques en Rust**

- Utiliser la propriété et le système de typage de Rust pour garantir la sécurité de la mémoire et la robustesse du code.
  - Préférer les types immuables whenever possible.
  - Utiliser des références et des pointeurs intelligents de manière appropriée.
  - Éviter les erreurs de nullité.
- Profiter des fonctionnalités de sécurité de la mémoire de Rust, telles que la vérification des limites de tableau et la détection des fuites de mémoire.
- Utiliser des bibliothèques Rust standard et éprouvées pour les tâches courantes, telles que la gestion des entrées/sorties, la sérialisation et la concurrence.
- Écrire des tests unitaires complets pour s'assurer que le code fonctionne comme prévu.
- Documenter le code de manière appropriée à l'aide de commentaires et de documentation générée automatiquement.

## **Outils pour améliorer la qualité du code**

- `rustfmt`: Formate le code Rust selon un style cohérent.
- `cargo test`: Exécute les tests unitaires pour le code Rust.
- `clippy`: Outil de linting pour identifier les problèmes potentiels dans le code Rust.
- `rustdoc`: Génère une documentation à partir des commentaires du code Rust.

# Changelog

## 0.1.0

- Version initiale avec une carte 2D, des robots de base et une station centrale.

## 0.2.0

- Ajout de la spécialisation des robots pour l'analyse chimique, le forage et l'imagerie.
- Implémentation de la reconnaissance du terrain et de la collecte d'échantillons.
- Amélioration de la communication et du partage d'informations entre les robots.

### ADR-1 : Conception modulaire des robots

- **Motivation** : Permettre des configurations flexibles des robots pour différentes missions.
- **Solution** : Définir des modules spécialisés pour des tâches spécifiques (analyse, forage, imagerie, etc.).
- **Avantages** :
  - Flexibilité et adaptabilité des robots.
  - Facilite la maintenance et l'évolution du code.
- **Inconvénients** :
  - Augmentation de la complexité de la conception.
  - Nécessite une gestion attentive des interfaces entre les modules.

**Décision** : Implémenter une conception modulaire pour les robots avec des modules spécialisés pour maximiser la flexibilité et l'adaptabilité.

### ADR-2 : Stratégie de communication entre les robots

**Motivation** : Définir une stratégie efficace pour la communication et le partage d'informations entre les robots.

#### Solutions envisagées :

- **Diffusion** : Les robots diffusent périodiquement leurs découvertes à tous les robots à portée de portée.
- **Communication** : Les robots échangent des informations avec les robots qu'ils rencontrent, propageant les informations à travers le réseau.
- **ROUTAGE** : Les robots envoient des messages ciblés à d'autres robots en fonction de leurs besoins spécifiques.

#### Avantages et inconvénients :

Solution	Avantages	Inconvénients
Diffusion	Simple à implémenter, faible latence pour les mises à jour locales	Inefficace pour les grandes cartes, peut entraîner une surcharge du réseau
Gossip	Robuste aux pannes de robots, échelonnable	Peut entraîner des délais de propagation des informations
Routage	Efficace pour les communications ciblées, réduit la surcharge du réseau	Plus complexe à implémenter, nécessite une connaissance de la topologie du réseau

drive\_spreadsheetExporter vers Sheets

**Décision** : Implémenter une stratégie de communication hybride combinant la diffusion et le routage. La diffusion sera utilisée pour les mises à jour locales et la découverte de robots, tandis que le routage sera utilisé pour les communications ciblées et le transfert de données volumineuses.

### ADR-3 : Modèle de décision pour la création de robots

**Motivation** : Déterminer comment et quand la station centrale doit créer de nouveaux robots.

**Solutions envisagées** :

- **Basé sur les ressources** : Créer de nouveaux robots lorsque l'énergie ou les minerais sont suffisants.
- **Basé sur la couverture** : Créer de nouveaux robots pour explorer les zones non cartographiées de la carte.
- **Basé sur les objectifs** : Créer de nouveaux robots en fonction des objectifs de mission spécifiques, tels que la collecte d'échantillons ou la recherche de points d'intérêt scientifiques.

**Avantages et inconvénients** :

Solution	Avantages	Inconvénients
Basé sur les ressources	Simple à implémenter, garantit que les robots ont les ressources nécessaires pour fonctionner	Peut retarder la création de robots si les ressources sont rares
Basé sur la couverture	Garantit une exploration complète de la carte	Peut créer des robots redondants si la carte est déjà bien explorée
Basé sur les objectifs	Optimise la création de robots pour les objectifs de la mission	Plus complexe à implémenter, nécessite une planification et une coordination précises

**Décision** : Implémenter un modèle de décision hybride basé sur les ressources et les objectifs. La station centrale créera de nouveaux robots en fonction de la disponibilité des ressources et des objectifs de mission prioritaires.