

Sujet commun SAÉ 2.01-SAÉ 2.02

Description du monde à simuler

Comme pour les SAÉ 1.01 & SAÉ 1.02, les deux SAÉ partageront un noyau commun.

Ce noyau consiste en la modélisation et la simulation de robots mineurs exploitant un nouveau monde.

Principes

Il y aura dans ce monde des mines ; chaque mine contient un nombre (entre 50 et 100) de minerais d'une seule nature : soit de nickel (1 ou 2 mines), soit d'or (1 ou 2 mines). Deux entrepôts permettent de stocker les minerais récoltés : l'un permettant de stocker l'or, l'autre le nickel. Des robots (au moins un de chaque type et au plus 5), chacun étant spécialisés dans l'extraction de minerai soit de nickel ou d'or, exploreront le monde afin d'aller dans les mines, extraire des minerais et les amener dans les entrepôts. Chaque robot a une capacité de stockage de minerais (entre 5 et 9) qui lui est propre, et une capacité (entre 1 et 3) d'extraction de minerais (nombre de minerais récoltés en une seule action d'extraction).

Le monde est décomposé en une matrice de 100 secteurs (10 x 10). Chaque secteur du monde sera soit un terrain (au moins 90), soit un plan d'eau.

Les robots peuvent aller sur un terrain du moment qu'il n'est pas occupé par un autre robot.

Les robots ne peuvent pas aller sur les plans d'eau.

Sur les terrains il peut y avoir soit rien, soit un entrepôt, soit une mine. On ne peut pas avoir un entrepôt et une mine sur un même secteur.

Les robots peuvent :

- Avancer dans une direction : Nord, Est, Sud ou Ouest,
- Récolter des minerais s'ils sont dans une mine de leur spécialité et s'il reste au moins un minerai. Le robot récolte, s'il y a suffisamment de minerai, autant que sa capacité d'extraction, sinon il récolte ce qu'il reste.
- Déposer ses minerais s'il est dans l'entrepôt dédié au minerai dont il est spécialisé.

Chaque robot a un numéro unique, chaque mine a un numéro unique, chaque entrepôt a un numéro unique.

À chaque tour, chaque robot peut réaliser au plus une action valide.

Informations à visualiser

Visuellement un secteur sera représenté par une matrice 2x2, chaque case pouvant avoir un caractère.

Secteur vide :

Secteur représentant un plan d'eau :

X	X
X	X

Secteur possédant une mine : la mine est identifiée sur la première ligne :

M	3

ici il s'agit de la mine n°3

Secteur possédant un entrepôt, l'entrepôt est identifié sur la première ligne :

E	1

ici il s'agit de l'entrepôt n°1

Secteur possédant un robot, celui-ci est identifié sur la deuxième ligne :

R	2

ici le robot n°2

Un secteur peut donc être occupé par un entrepôt ou une mine ET SIMULTANÉMENT

par un robot :

M	1
R	1

 ; ici dans la Mine 1 se trouve le robot n°1.

La représentation du monde pourra se faire de la façon suivante (à noter que l'affichage pourra, dans un premier temps, se faire en mode console) :

	0	1	2	3	4	5	6	7	8	9	
0		X	X						E	2	
1		X	X					R	2		
2						X	X			M	3
3				X	X						
4				X	X						
5									M	2	
6							X	X			
7							X	X			
8			E	1	X	X					
9					X	X					
						M	1				
						R	1				
tour	75										
M1	9	5	OR	72	/	90					
M2	5	8	NI	55	/	75					
M3	2	9	OR	0	/	80					
E1	8	2	NI	20							
E2	0	8	OR	96							
R1	9	5	OR	2	/	5					
R2	7	3	NI	0	/	7					

On affichera le numéro du tour actuel et les informations concernant toutes les mines, robots et entrepôts

Pour les mines : M suivi de son n°, puis sa ligne et sa colonne, son type, son nombre de minerais actuel et sa capacité initiale.

Pour les entrepôts : E suivi de son n°, puis sa ligne et sa colonne, son type, son nombre de minerais actuel.

Pour les robots : R suivi de son n°, puis sa ligne et sa colonne, son type, son nombre de minerais actuel et sa capacité de stockage.

Livrables pour la SAÉ2.01 :

Dans le cadre de la SAÉ2.01, c'est l'utilisateur qui pilote chaque robot.

À chaque tour, pour chaque robot, l'utilisateur indique l'action à réaliser.

Livable 1.01 : Diagrammes des classes (deadline 1ère version 08/04, 2ème version mise à jour après le code final – fin mai)

Livable 1.02 : Création du projet sur git-hub. Rapport des premières commits. (deadline 08/04/2022)

Livable 1.03 : Maquettage des interactions pour la saisie des actions à réaliser par l'utilisateur (deadline fin mai)

Livable 1.04 : Rapport de test automatisé JUnit pour au moins deux des principales méthodes (fin mai). Le rapport doit contenir (a) le jeu de test (tableau de test tel que réalisé dans R2.03) avec l'indication des critères de test qui ont été utilisés, (2) un rapport du résultat du test de l'application (par exemple, une capture d'écran du résultat de l'exécution). (deadline fin mai)

Livable 1.05 : Code java final intégrant les interactions utilisateur graphiques. (deadline fin mai)

Livable 1.06 : Rapport final des commits réalisées. (deadline fin mai)

Attention : Organisez-vous pour utiliser GIT tout au long du projet. Nous attendons différents commits de la part de différentes personnes de l'équipe.

Nous vous conseillons d'avancer sur le projet avec cette organisation :

- 1ère séance en autonomie : création du modèle UML et Java (vous utiliserez les compétences acquises en R2.01)
- 2ème séance en autonomie : création du GIT et dépôt du code fait en séance 1 (à compléter au fil des avancées) (R2.03) – création du maquettage (interaction de l'utilisateur avec une interface graphique pour indiquer les actions des robots à chaque tour).
- 3ème séance en autonomie : Codage de la partie commune et des interactions (R2.02 – programmation événementielle) (possible à partir de mi-avril)
- 4ème séance en autonomie : création des tests et mise en œuvre dans JUnit / code à debugger (R2.03) – finalisation du code et dépôt.

Livrables pour la SAÉ2.02 :

Dans le cadre de la SAÉ2.02, c'est l'ordinateur qui pilote chaque robot. À chaque tour, pour chaque robot, l'ordinateur indique l'action à réaliser.

De nombreux algorithmes sont envisageables pour répondre à la question. Nous vous conseillons de vous consacrer à l'un des algorithmes suivants :

- Parcours en profondeur.
- Parcours en largeur.
- Colonies de fourmis.

Le rapport attendu dans les livrables doit expliquer votre approche, en fournissant des exemples illustrés. Il ne s'agit pas d'un bloc-notes ! Des phrases bien construites et sans faute sont attendues. Des pénalités seront appliquées dans le cas contraire.

Livable 2.01 : Rapport explicatif (deadline fin mai).

Livable 2.02 : Code Java (deadline fin mai)