


Projet Robot

Master CCI 2015 – 2016

TELO Reddy

 — □ ×

Nettoyer

Polluer Tout

RAZ

Quitter

Nombre de papier gras : 0

Pollueur Droit											Nettoyeur Droit
Pollueur Sauteur											Nettoyeur Sauteur
Pollueur Libre											Nettoyeur Libre
Pollueur Smart											Nettoyeur Smart

I. Description général du projet :

Ce projet consiste à implémenter une application qui gère le déplacement des robots via une Interface graphique en Java.

Je vais tout d'abord décrire les différentes classes du projet puis les différents éléments qui constituent l'Interface graphique.

Afin de mettre en place notre application, nous avons eu à implémenté les classes et les méthodes suivantes :

- Une classe « Monde » qui est constituée de :
 - trois attributs, le nombre lignes et de colonnes de type « entier » ainsi que d'une matrice (10X10) de type « booléen ».
 - Un constructeur qui initialise le Monde.
 - Une méthode « **ajoutPapier** » qui permet d'ajouter un papier gras dans le Monde
 - Une méthode « **LirePapier** » qui affiche la case du Monde passée en paramètre si elle contient un papier gras,
 - Une méthode « **testPapierGras** » qui nous renvoie vrai ou faux selon que la case du Monde passée en paramètre contient un papier gras ou non
 - Une méthode « **comptePapierGras** » qui renvoie le nombre de papier gras dans le Monde.
- Une classe « ajoutPapierexception » qui affiche un message à l'écran quand le nombre de lignes et de colonnes passés en paramètre sont plus grands que ceux du Monde.
- Une classe « Robot » qui est composée de :
 - De trois attributs, la position x, la position y ainsi qu'un d'un Monde,
 - Un constructeur **Robot (int x, int y)** qui crée un robot à la position (x, y),
 - Un constructeur qui crée un Robot avec une position aléatoire,
 - Une méthode pour se déplacer en (i, j),
 - Une méthode qui permet de parcourir le Monde.
- Une classe « **RobotPollueurs** » qui est constituée de :
 - Un constructeur (**RobotPollueurToutDroit**) qui peut mettre du papier gras là où il se trouve. Il a un numéro de colonne, le numéro de la colonne où il va se rendre pour commencer sa mission.
 - Un constructeur (**RobotPollueurSauteurs**) qui prend en paramètre le nombre de colonnes, de lignes et le pas. Il peut mettre du papier gras selon les valeurs passées en paramètre.
 - Un constructeur (**RobotpollueurLibre**) qui peut mettre du papier gras de manière aléatoire.
 - Un constructeur qui met du papier gras dans toutes les cases du Monde.

- Une classe « **RobotNettoyeurs** » qui est composée de :
 - Un constructeur qui parcourt toutes les cases du Monde pour les nettoyer.
 - Un constructeur (**RobotNettoyeurToutDroit**) qui parcourt la colonne du Monde passée en paramètre pour la nettoyer.
 - Un constructeur (**RobotNettoyeurSauteur**) nettoie le Monde en fonction des valeurs passées en paramètre (nombre de colonnes, de lignes et le pas).
 - Un constructeur (**RobotNettoyeurLibre**) qui nettoie le Monde façon aléatoire.

- Une classe « **RobotPollueurSmart** » qui est constituée de :
 - Un constructeur qui prend un Monde en paramètre et qui fait appel à toutes les méthodes de la classe,
 - Une Méthode « **PollueurSmart** » qui pollue le Monde toutes les secondes (timer) lorsqu'on appuie sur le bouton « **Pollueur Smart** ».

- Une classe « **RobotNettoyeurSmart** » qui est composée de :
 - Un constructeur qui prend un Monde en paramètre et fait appel toutes les méthodes de la classe,
 - Une méthode « **NettoyeurSmart** » qui nettoie les cases polluées par le **RobotPollueurSmart** toutes les $\frac{1}{2}$ secondes lorsqu'on appuie sur le bouton « Nettoyeur Smart ».

- Une classe « **panelNord** » qui est constituée des éléments suivants :
 - Un constructeur qui fait appel tous les méthodes de la classe
 - Une méthode qui « **setPanelNord** » qui permet créer et de configurer (création et configuration des boutons du panel) le panel nord.
 - Une méthode « **PolluerTout** » qui fait appel au constructeur qui met un papier gras dans toutes les cases du Monde lorsqu'on appuie sur le bouton « Polluer Tout »,
 - Une Méthode « Nettoyeur » qui fait au constructeur qui nettoie toutes les cases du Monde quand on actionne le bouton « Nettoyer »,
 - Une méthode « Quitter » qui implémente le code qui permet de sortir de l'application lorsqu'on appuie sur le bouton « Quitter ».

- Une classe « **PanelOuest** » qui est composée comme suit :
 - Un constructeur qui prend en paramètre un Monde et qui fait appel à toutes les méthodes de la classe,
 - Une méthode « **setPanelOuouest** » dans lequel, on gère tous les boutons qui se trouvent le panel ouest, notamment les boutons « Pollueur Droit, Pollueur Libre, Pollueur Sauteur et Pollueur Smart »,
 - Une méthode « **pollueurToutDroit** » qui fait appel au constructeur qui pollue une colonne du Monde à la fois en fonction de la valeur passées en paramètre lorsqu'on appuie sur le bouton d même nom,

- Une méthode « **pollueurLibre** » qui fait appel au constructeur qui remplit aléatoirement une case du Monde à chaque fois que l'on actionne le bouton du même nom.
- Une méthode « **pollueurSauteur** » qui fait appel au constructeur qui remplit le tableau en fonction du numéro de la ligne, de la colonne et du nombre de pas passés en paramètres lorsqu'on appuie sur le bouton « Pollueur Sauteur »,
- Une classe « **PanelEst** » qui est composée comme suit :
 - Un constructeur qui prend en paramètre un Monde et qui fait appel à toutes les méthodes de la classe,
 - Une méthode « **setPanelEst** » dans lequel, on gère tous les boutons qui se trouvent sur le panel ouest, notamment les boutons « Pollueur Droit, Pollueur Libre, Pollueur Sauteur et Pollueur Smart »,
 - Une méthode « **nettoyerToutDroit** » qui fait appel au constructeur qui nettoie une colonne du Monde à la fois en fonction de la valeur passées en paramètre lorsqu'on appuie sur le bouton d même nom,
 - Une méthode « **nettoyerLibre** » qui fait appel au constructeur qui nettoie aléatoirement une case du Monde à chaque fois que l'on actionne le bouton du même nom,
 - Une méthode « **nettoyerSauteur** » qui fait appel au constructeur qui nettoie le tableau en fonction du numéro de la ligne, de la colonne et du nombre de pas passés en paramètres lorsqu'on appuie sur le bouton « Pollueur Sauteur ».

Maintenant que j'ai décrit les différentes classes de mon projet, je vais brièvement des éléments qui constituent mon interface graphique.

Mon interface graphique comprend quatre panels :

- Un « **Panel Nord** » qui est constitué des boutons « *Nettoyer* » (qui permet de nettoyer toutes les cases du Monde), « *Polluer Tout* » (qui permet de polluer toutes les cases du Monde), « *RAZ* » (qui ne fait, faute de temps et manque d'inspiration), « *Quitter* » (qui permet de sortir de l'application) ainsi que d'un champ qui permet d'afficher le nombre de papier gras dans le Monde,
- Un « **Panel centre** » qui est constituée d'une grille (10X10) qui fait office de Monde,
- Un « **Panel Ouest** » qui comprend les boutons « *Pollueur Droit* » (qui permet de polluer une seule colonne du Monde à la fois selon la valeur passée en paramètre), « *Pollueur Sauteur* » (qui pollue le Monde en sautant des lignes et colonnes selon les valeurs passées en paramètre), « *Pollueur Libre* » (qui met à chaque fois un papier gras de manière aléatoire dans une case du Monde), « *Pollueur smart* » (qui pollue quelques cases du Monde toutes secondes).

- Un « **Panel Est** » qui comprend les boutons « *Nettoyeur Droit* » (qui permet de Nettoyer une seule colonne du Monde à la fois selon la valeur passée en paramètre), « *Nettoyeur Sauteur* » (qui nettoie le Monde en sautant des lignes et colonnes selon les valeurs passées en paramètre), « *Nettoyeur Libre* » (qui nettoie à chaque fois de manière aléatoire une case du Monde), « *Nettoyeur smart* » (qui nettoie quelques cases du Monde toutes ½ secondes).