

Rapport de projet Sokoban

Sommaire

Répartition des responsabilités entre les classes	2
Package boards	2
La classe board	2
La classe position	2
La classe Type	2
Package "databaseAndFiles"	2
L'interface BoardBuilder	2
La classe Builder	2
La classe Database	2
Le package mains	3
La classe Administrator	3
La classe Player	3
Le package playermanager	3
La classe Direction	3
La classe KeyboardEntry	3
Le package de Test	3
La classe KeyboardEntryTest	3
Choix effectués pour la programmation	4
Administrator	4
Board	4
BoardBuilder	4
Builder	4
Database	5
Direction	5
KeyboardEntry	5
Player	5
Position	5
Type	5

Répartition des responsabilités entre les classes

Package boards

Le package boards contient trois classes à savoir les classes Board, Position et Type. Ce package gère donc comme son nom l'indique le plateau et ses caractéristiques.

La classe board

Elle possède un nom (boardName) deux entiers qui sont les nombres de colonnes et les nombre de lignes ainsi qu'un tableau à deux entrées de Position pour stocker les position de chaque case et enfin une HashMap de Position et de Type qui sauvegarde les type de chaque positions au début de la partie. Elle possède plusieurs méthodes dont notamment des méthodes d'affichage ainsi que des méthodes "get" et des méthodes qui permettent d'ajouter divers éléments à un plateau (mur, player, box ...).

La classe position

Elle possède deux entiers qui sont la colonne et la ligne, et un type qui par défaut est "empty". Elle possède également des méthodes qui sont la pour créer une position vide et pour remplacer une position par un joueur ou par une boîte (utile pour déplacer le joueur).

La classe Type

Est une énumération des différents type de positions possible et qui a une méthode ToString qui override la méthode originelle afin de les afficher dans le plateau plus joliment

Package "databaseAndFiles"

Le package databaseAndFiles regroupe les classes qui gèrent les bases de données et la gestion des fichiers.

L'interface BoardBuilder

Cette interface donne la structure de la classe builder avec sa méthode build.

La classe Builder

Cette classe qui implémente donc BoardBuilder permet de générer à partir d'un fichier un plateau. Notamment grâce à ses méthode de lecture de fichier et de constructeur de plateau

La classe Database

Elle regroupe toutes les méthodes qui seront utiles pour lire, écrire et modifier les bases de données. Elle a comme attribut deux chaîne de caractères à savoir le driver et le chemin du fichier de base de données (ici à la source du projet). Elle a donc des méthodes pour consulter la base de données, pour supprimer des plateaux et pour get un plateau d'une

base de donnée. Elle peut aussi créer à partir d'un fichier un plateau et l'ajouter à la base de données.

Le package mains

Ce package contient les deux classes main du projet (celle de l'admin et celle du joueur)

La classe Administrator

Cette classe est réservée à l'administrateur de la base de données afin qu'il puisse ajouter, supprimer et lire les différents plateaux de la base. Elle n'est constituée que d'une méthode main.

La classe Player

Cette classe main est réservée aux joueurs qui peuvent donc ainsi profiter du sokoban et faire une partie. Pour ce faire, ils peuvent consulter les plateaux et en choisir un. Ensuite il faudra entrer la combinaison gagnante pour espérer vaincre le sokoban. Elle possède aussi une méthode qui permet de déplacer le joueur

Le package playermanager

Ce package contient les classes qui s'occupent du player et des entrées au clavier.

La classe Direction

Cette classe est une énumération des différentes directions que peut prendre le joueur. Elle possède une méthode qui regroupe toutes cette énumération ainsi que deux méthodes qui gèrent le nombre de "pas" après un déplacement vertical ou horizontal.

La classe KeyboardEntry

Elle gère seulement la gestion des touches entrées au clavier par l'utilisateur.

Le package de Test

La classe KeyboardEntryTest

Elle test uniquement si les entrées sont valides

Choix effectués pour la programmation

Administrator

J'ai choisi un switch pour gérer les différentes options plutôt qu'une succession de if/else if ... afin de rendre le code plus clair

Board

J'ai choisi de faire un tableau à double entrée pour faciliter son parcours grâce à deux boucles for qui parcourt d'abord les colonnes puis les lignes. De plus ce tableau est de type position ce qui fait que l'on peut avoir accès directement au type de la case. J'ai aussi une HashMap qui associe une position à un type qui me permet de savoir quel était l'état initial du plateau.

BoardBuilder

J'ai choisi d'en faire une interface qui permet de donner le squelette de la classe Builder.

Builder

J'ai choisi de prendre une ArrayList de String pour récupérer ce qu'il a dans le fichier car cela me permet de le consulter plus aisément. Je l'utilise dans la méthode build qui renvoie un plateau grâce à la méthode de java ".get"

Database

J'ai décidé de laisser le choix à l'utilisateur quant au chemin de la base de données alors que le driver lui est directement dans le constructeur. J'ai également choisi de faire des requêtes préparées afin de me faciliter la tâche grâce aux setString.

Direction

J'ai choisi une énumération car cela est facile à consulter.

KeyboardEntry

J'ai choisi de créer cette classe pour éviter d'avoir à utiliser un nouveau scanner à chaque fois que je veux lire une entrée au clavier.

Player

J'ai également choisi un switch pour faciliter la lecture et la compréhension de mon code.

Position

Une position associe une colonne une ligne et un type pour permettre d'en faire un double tableau.

Type

J'ai choisi une énumération car c'est plus facile à consulter. La méthode toString permet de dessiner plus joliment sur le plateau.