Rapport « Sokoban »

M2103 2021

Ayman KACHMAR, groupe S2A”

Ce rapport décrit la conception et la réalisation d’un programme de jeu en mode texte. Dans ce jeu, le joueur doit déplacer son personnage et pousser les caisses aux points de destinations pour gagner.

Plan du rapport

1. Analyse

1.1. Spécification des besoins

Le programme doit fonctionner en mode texte. Les mouvements joués modifient l'affichage du plateau.

1.2. Conception

Les plateaux de jeu sont constitués de caractères sont stockés dans des fichiers .txt, pour cela :

- Database : une base de données qui regroupe les caractéristiques d'un plateau (id, nom, nombre de lignes, nombre de colonnes) et effectuer des requêtes SQL pour pouvoir ajouter ou supprimer des données.
- Administrator : un menu de gestion pour modifier la base de données, ajouter ou supprimer des plateaux de jeu.

Choix de conception : pour représenter l'état de l'interaction avec le joueur, la classe Player contient un plateau de type Board qui sera utilisé par le joueur après sélection dans la base de données, voici choix possibles pour choisir le plateau de jeu :

- choix 1 : par la base de données
    - demander de sélectionner un plateau parmi la liste des plateaux de la base
    - l'utilisateur choisit un plateau en saisissant l'id correspondant
    - la partie se lance
- choix 2 : par un fichier

1.3. Données et traitements

Le programme principal consiste après lancement de partie, une boucle dépendante à la victoire ou à l'abandon, puis à :

- demander à l'utilisateur de saisir un ou plusieurs mouvement(s) de direction (L, R, U ou D).
- soumettre la modification
- afficher le plateau (Board)
- demander un mouvement à nouveau et ainsi de suite

2. Réalisation

2.1. Organisation en packages

Le projet est découpé en packages, avec les responsabilités indiquées :

```
src
| - - sokoban                          points d'entrées (main)
|     | - - Administrator.java
|     | - - Player.java
| - - sokoban.board                    constitution des objets du
jeu
|     ` - - Board.java
|     ` - - BuildException.java
|     ` - - Cell.java
|     ` - - Direction.java
|     ` - - Position.java
| - - sokoban.builder                  construction d'un plateau
|     ` - - BoardBuilder.java
|     ` - - FileBoardBuilder.java
|     ` - - TextBoardBuilder.java
| - - sokoban.database                 gestion de base de données
|     ` - - Database.java
|     ` - - DatabaseException java
| - - sokoban.tests                    tests unitaires
|     ` - - BoardBuilder.java
|     ` - - FileBoardBuilder.java
|     ` - - TextBoardBuilder.java
```

2.2. Package sokoban

- La classe Administrator est un gestionnaire de la base de données, pour ajouter ou supprimer les plateaux de jeu.
- La classe Player est une lanceuse de partie et permet au joueur de :
    - choisir son plateau à partir de la méthode menuPlayer()
    - après fin du menu de lancer game() puis jouer ses mouvements à l'aide de la méthode refreshBoard(Board b)

Détails techniques :

Administrator interagit avec l'utilisateur :

- pour la méthode addBoard() : nous faisons appel à la fonction requestBoard() dans laquelle le chemin absolu est récupéré, puis l'utilisateur doit saisir le nom du fichier .txt contenant le plateau en caractères. Après cela, un FileBoardBuilder est créé à l'aide du chemin (chemin absolu + le fichier .txt), ensuite nous faisons appel à build() et demandons un id à ce plateau pour enfin l'ajouter.

```
private static void addBoard() throws BuildException, DatabaseException {
    Board board = requestBoard();
    out.println("Veuillez saisir l'id de ce plateau dans la base : ");
    String id = readLine();
    database.add(id, board);
}
```

```
public static Board requestBoard() throws BuildException {
    String sourcePath = System.getProperty("user.dir").replace('\\', '/') + "/datafiles/";

    out.println("Veuillez saisir le nom du fichier texte situé dans datafiles (exemple : 'nomdufichier.txt')");
    String boardFile = readLine();
    String path = sourcePath + boardFile;

    out.println("Veuillez nommer votre plateau : ");
    String name = readLine();
    FileBoardBuilder fileBoard = new FileBoardBuilder(path, name);
    Board board = fileBoard.build();
    return board;
}
```

- pour la sélection d'un plateau pour son affichage, sa suppression, ou sa sélection en jeu chaque méthode displayABoard(), removeBoard(), getBoard() vérifie si l'id existe, sinon demander une nouvelle saisie récursivement, exemple

```
private static void displayABoard() throws DatabaseException {
    database.displayBoards();
    out.println("Veuillez saisir l'id du plateau à afficher : ");
    String id = readLine();
    if (database.getAllIDs().contains(id)) {
        database.displayRows(id);
    } else if (!id.equals("q")) {
        out.println("id introuvable.");
        displayABoard();
    }
}
```

pour la méthode refreshBoard(Board b), nous procédons ainsi :

1. demander la saisie du mouvement
2. vérification de la validité du coup à l'aide de la méthode possibleEntry(String entry), seuls les caractères L,R,U,R et leurs minuscules sont acceptés. "q" est aussi une possibilité pour quitter.
3. si l'entrée est valide alors utiliser les méthodes refreshPositions(String entry) et setMyPos(int numRow, int numCol)

2.3. Package sokoban.board

Ce package contient les classes Board, Cell, Direction et Position

- Board permet de construire le plateau : il contient la méthode d'affichage display(), les méthodes d'ajouts d'objets dans le plateau (caisses, murs, destinations, personnage) avec les méthodes de type "add…"(), la méthode refreshPositions(String entry) permettant de déplacer notre personnage si l'on ne rencontre pas de murs, sinon de déplacer la caisse et le personnage si le mouvement le permet. victory() vérifie si les caisses sont à destination pour retourner vrai, et getRowsBoard() permet de renvoyer un tableau de String contenant chaque ligne du plateau.
- La classe Position crée une position avec ligne et colonne et contient les méthodes isInBoard(Board b) pour vérifier que l'on reste dans le plateau, et nextPosition(Direction d) qui va à la position suivante en fonction de la direction.
- La classe Cell est un mixte entre Position et le caractère qui sera affiché, pour aussi distinguer les objets entre eux : caisses, murs, personnage, destinations
- Direction est une énumération des 4 directions et possède la méthode dirCorrespond(char c) qui renvoie la direction en fonction du caractère (servant à refreshPositions(String entry))

Détails techniques :

```
Direction dir = Direction.dirCorrespond(entry.charAt(i));
Position myNewPos = myPosition.nextPosition(dir);
Position lastBoxPos = findBox(myNewPos.nextPosition(dir), dir);
```

La position myNewPos est la nouvelle position de notre personnage, lastBoxPos est la dernière position de la caisse, trouvée à l'aide de la méthode suivante : la méthode findBox sert à retourner la position de la caisse finale parmi la "queue" des caisses collées, en procédant par une boucle while qui vérifie à chaque fois que la position suivante est dans le plateau et qu'il y a une caisse sur celle-ci. Le but est de pousser seulement la dernière caisse en la faisant apparaître avec cette position finale, et notre personnage écrasera la caisse qui la suit.

```
private Position findBox(Position nextMyNewPos, Direction dir) {
    Position next = nextMyNewPos;
    while (next.isInBoard(this) && next.nextPosition(dir).isInBoard(this) && getCell(next).isBox()) {
        next = next.nextPosition(dir);
    }
    return next;
}
```

Note : dans cette méthode, nous actualisons la liste des caisses dans le hashset listeBoxes qui sert à détecter la victoire :

```
getCell(newPos).setCar('B', false);
listBoxes.remove(getCell(oldPos));
listBoxes.add(getCell(newPos));
```

2.4. Package sokoban.builder

Ce package contient les classes FileBoardBuilder et TextBoardBuilder, implémentant l'interface BoardBuilder qui contient la méthode build()

- TextBoardBuilder contient deux méthodes indispensables : addRow(String row) et build() de l'interface :
    - addRow(String row) ajoute la ligne entrée à textBoard étant un attribut de type String représentant le plateau en chaine de caractère
    - build() renvoie un plateau de type Board, elle initialise d'abord un tableau de String pour distinguer chaque ligne de TextBoard, puis parcourt chaque caractère, pour chaque ligne, et utilise les fonctions de Board pour modifier le plateau en attribuant à chaque position le caractère correspondant.
- FileBoardBuilder est créé à l'aide un chemin (correspondant à l'accès au fichier .txt), et le nom du plateau, le chemin permettra à la méthode build() de l'interface de :
    - créer un objet TextBoardBuilder
    - utiliser un scanner sur le chemin du fichier
    - pour chaque ligne lue nommé row, utiliser la méthode addRow(row) pour l'objet TextBoardBuilder créé
    - enfin créer le board à partir de cet object et retourner le board

Détails techniques : le Board créé dans la méthode build est initialisé avec un width et un height : le width correspond à la longueur d'une ligne du plateau et height est incrémenté à chaque utilisation de la fonction addRow(String row).
En ce qui concerne l'attribution des caractères à chaque position correspondante, l'indice i est la colonne et permet de lire chaque caractère d'une ligne, puis numRow est incrémenté pour passer à la ligne suivante.

```
for (String s : rows) {
    for (int i = 0; i < width; i++) {
        char object = s.charAt(i);
        switch (object) {
            case '#' :
                board.addHorizontalWall(numRow, i, 1); break;
            case 'x' :
                board.addTarget(numRow, i);
                nbTargets++; break;
            case 'B' :
                board.addBox(numRow, i);
                nbBoxes++; break;
            case 'P' :
                board.setMyPos(numRow, i);
                nbMyPos++; break;
        }
    }
    numRow++;
}
```

Les attributs nbTargets, nbBoxes, nbMyPos servent dans la méthode booléenne isAValidBoard() qui retourne vrai si seulement si nbTargets = nbBoxes tout deux strictement positifs et nbMyPos = 1 puis un width et un height supérieur ou égal à 3.

2.4. Package sokoban.database

Ce package contient les classes Database et DatabaseException.

Database permet la création d'une base de données et contient :

- la fonction createDatas() permet de créer les tables pour l'affichage des plateaux (boards) et les lignes d'un plateau (rows) en donnant à chaque attribut son type (clé primaire, integer…)
- add(String id, Board b) permet l'ajout de plateau de type Board dans la base de données
- remove(String id, boolean isDeletingOne) permet la suppression d'un ou de tous les plateaux en fonction de ce booléen (ceci est optionnel mais peut s'avérer utile dans certains cas pour tout supprimer)
- get(String id) permet de récupérer et de retourner le Board correspondant à l'id d'entrée, cela sert pour la classe Player, lorsque le joueur choisit son plateau
- les méthodes d'affichages displayBoards() et displayRows(String boardId)
- idExists(String id) utilise la fonction getAllIds(), pour vérifier si l'id entré n'est pas inconnu et existe dans le HashSet et donc dans la base.

Détails techniques :
Chaque fonction utilise précisément des requêtes SQL faites pour :

- création de tables : CREATE IF NOT EXISTS
- insertion de valeurs : INSERT INTO … VALUES(...)
- sélection de données : SELECT … FROM … WHERE ...

En ce qui concerne l'exécution des opérations, nous utilisons des objets Statement ou PreparedStatement puis les méthodes associées : executeQuery(), executeUpdate(), setString(1, id) par exemple pour retrouver l'id voulu dans la clause WHERE = ?.

```
public Board get(String id) throws DatabaseException, BuildException {
    String getNameSQL = "SELECT name FROM boards WHERE board_id = ?";
    String getBoardSQL = "SELECT description FROM rows WHERE board_id = ?";
    TextBoardBuilder builder;

    try {
        PreparedStatement statementName = connection.prepareStatement(getNameSQL);
        statementName.setString(1, id);
        ResultSet selectName = statementName.executeQuery();

        PreparedStatement statementRows = connection.prepareStatement(getBoardSQL);
        statementRows.setString(1, id);
        ResultSet selectRows = statementRows.executeQuery();
        builder = new TextBoardBuilder(selectName.getString(1));

        while (selectRows.next()) {
            builder.addRow(selectRows.getString(1));
        }
    } catch (SQLException ex) {
        throw new DatabaseException(ex.getMessage());
    }
    return builder.build();
}
```

Ici, le but est de récupérer un plateau voulu dans la base, puis de retourner l'objet de type Board. Pour cela il faut récupérer chaque ligne composant ce plateau, ces lignes se trouvent

dans la colonne ‘description’. Après exécution de la requête un TextBoardBuilder est créé avec le nom du Board correspondant à l’id entré, ensuite nous utilisons la méthode addRow pour chaque ligne lue de la colonne ‘description’, enfin toutes les lignes ont été ajoutées et le Board construit est retourné.

2.5. Tests unitaires

Trois tests unitaires sont écrits : DatabaseTest.java, FileTextBoardBuilder.java, GameTest.java :

- Le test de Database génère un Board nommé b1 part le fichier simple.txt et ayant pour id “1”, la base de données ajoute b1, puis nous utilisons la méthode get(String id) pour récupérer le Board d’id “1”. Par un assertEquals nous vérifions que les boards sont les mêmes. Nous supprimons tout et vérifions aussi que la méthode getAllIds fonctionne en ajoutant dans la base et dans le HashSet des boards, puis en faisant un assertEquals.
- FileTextBoardBuilder utilise pour le TextBoard la méthode addRow(String row) pour créer un plateau et FileBoard récupère quant à lui un plateau par le fichier simple.txt, puis un assertEquals est fait.
- GameTest simule des mouvements et détecte la victoire.

2.6. Exemple extrait d’exécution

a) Un ajout de plateau dans la base avec Administrator

```
* Menu
1. Créer une nouvelle base de données
2. Liste des plateaux
3. Montrer un plateau
4. Ajouter un plateau depuis un fichier
5. Supprimer un plateau de la base [DANGEREUX]
q. Quitter.
? 2
BOARDS
|------------|--------------------------|---------|---------|
| board_id   | name                     | nb_rows | nb_cols |
|------------|--------------------------|---------|---------+
| Moyen      | A Simple Board           | 5       | 10      |
| Facile     | A Simple Board           | 5       | 10      |
| Difficile  | A Simple Board           | 5       | 10      |
|------------|--------------------------|---------|---------|
* Menu
1. Créer une nouvelle base de données
2. Liste des plateaux
3. Montrer un plateau
4. Ajouter un plateau depuis un fichier
5. Supprimer un plateau de la base [DANGEREUX]
q. Quitter.
? 4
Veuillez saisir le nom du fichier texte situé dans datafiles (exemple : 'nomdufichier.txt')
difficult.txt
Veuillez nommer votre plateau :
nouveau
Veuillez saisir l'id de ce plateau dans la base :
33
```

```
? 3
BOARDS
|-------------|---------------------------|---------|---------|
| board_id    | name                      | nb_rows | nb_cols |
|-------------|---------------------------|---------|---------+
| Moyen       | A Simple Board            | 5       | 10      |
| Facile      | A Simple Board            | 5       | 10      |
| Difficile   | A Simple Board            | 5       | 10      |
| 33          | nouveau                   | 7       | 7       |
|-------------|---------------------------|---------|---------|
Veuillez saisir l'id du plateau à afficher :
33
ROWS
|-------------|-------------------------------------|
| board_id    | row_num | description               |
|-------------|-------------------------------------|
| 33          | 0       | # # # # # # # #           |
| 33          | 1       | # . x . . # #             |
| 33          | 2       | # . . # . x #             |
| 33          | 3       | # . . P . . #             |
| 33          | 4       | # . . . B . #             |
| 33          | 5       | # . . . B . #             |
| 33          | 6       | # # # # # # # #           |
|-------------|-------------------------------------|
```

b) Lancement d'une partie avec Player

```
* Menu
1. Sélectionner un plateau de la base pour jouer.
2. Sélectionner un plateau depuis un fichier.
q. Quitter.
? 1
BOARDS
|-------------|---------------------------|---------|---------|
| board_id    | name                      | nb_rows | nb_cols |
|-------------|---------------------------|---------|---------+
| difficile   | plateaul                  | 7       | 7       |
| 2           | tres simple               | 5       | 10      |
|-------------|---------------------------|---------|---------|
Veuillez saisir l'id du plateau de jeu :
2
# # # # # # # # # #
# x x x # . . . . #
# B B B . . . P . #
# . . . . . . . . #
# # # # # # # # # #

Saisissez un mouvement (L,R,U,D) :
LLDDLLLU
# # # # # # # # # #
# x B x # . . . . #
# B P B . . . . . #
# . . . . . . . . #
# # # # # # # # # #

Saisissez un mouvement (L,R,U,D) :
q
Bye.
```