Rapport
The Big Adventure

TOUNSI Mourad

&

TOUSSAINT Lesly Jumelle

Sommaire

- ✓ **Introduction**
- ✓ **Organisation**
- ✓ **Conclusion**

- Introduction

Le but de ce projet est de jouer à un jeu d'aventure et/ou de fabriquer les mondes du jeu d'aventure. Il convient de créer des éléments de carte, des éléments décoratifs, des éléments obstacles, des éléments intermédiaires, des éléments amis ou ennemis, des éléments inventaires, des éléments nutritifs, des personnages, des éléments de transports, des éléments d'armes, des éléments spéciaux.

- Organisation

Notre projet est reparti dans plusieurs packages tels que : lexer.parser, objets, objets.primitive, vue et un package par défaut. Il contient également autres dossiers : maps, docs, lib, src...

- Les packages

lexer.parser:

DataCarte.java, qui est la carte qui contient les informations du jeu, c'est à dire la grille et tous les éléments y compris le joueur. Les champs de ce fichier sont des String, et contient également un champ elements qui est une liste qui stocke les informations de tous les éléments du fichier map ([element], [element], …)

Lexeme.java (celui du professeur). Un fichier Lexemes.java qui contient une liste de lexemes(de Lexeme.java) et un pointeur qui est un entier indiquqant la position ou l'on se trouve durant la construction de la carte (la DataCarte).

Lexer.java, celui du professeur, il sert à retourner un seul lexeme (du Lexeme .java).

Parser.java, il va prendre un objet de type Lexemes.java et construire un objet de type DataCarte (c'est l'objet qui contient les éléments nécessaires pour construire la vrais carte Carte.java).

NB : pour plus de précision, on vous fournit l'image de notre parseur.

```
MAP  --->   [element] MAP1 [grid] MAP2 [element] MAP1

MAP2 --->   size : V1 encodingd : V2 data : V3
        |   size : V1 data : V3 encodingd : V2

        |   data : V3 size : V1 encodingd : V2
        |   data : V3 encodingd : V2 size : V1

        |   encodingd : V2 size : V1 data : V3
        |   encodingd : data : V3 V2 size : V1

V1   --->   ( NUMBER , NUMBER )

V2   --->   IDENTIFIER ( IDENTIFIER ) V2
        |   epsilon

V3   --->   QUOTE

MAP1 --->   position : COUPLE
        |   zone : ZONE
        |   locked : LOCKED
        |   steal : STEAL
        |   trade : TRADE
        |   IDENTIFIER : ID

ID      --->    IDENTIFIER

COUPLE  --->    ( NUMBER , NUMBER )

ZONE    --->    ( NUMBER , NUMBER ) ( NUMBER x NUMBER )

LOCKED  --->    IDENTIFIER  IDENTIFIER

STEAL   --->    IDENTIFIER , STEAL
           |    IDENTIFIER

TRADE   --->    IDENTIFIER -> IDENTIFIER  IDENTIFIER , TRADE
           |    IDENTIFIER  -> IDENTIFIER , TRADE
           |    IDENTIFIER -> IDENTIFIER  IDENTIFIER
           |    IDENTIFIER  -> IDENTIFIER
```

Token.java c'est celui du professeur, c'est un enum qui contient les expressions regulieres.

GestionsErreurs.java : on capte toutes les erreurs du fichier (.map).

Le package objets contient les classes des vrais éléments du jeu

Biome.java, ce fichier contient quelques éléments de la carte tels que : eau, feu, glace.

Carte.java : ce fichier contient les informations de grid(size, encodings, grille).

size,c'est un couple (x, y) qui indique les dimensions de la grille ;

encodings, c'est une map qui fait correspondre une lettre a un skin ;

grille qui un tableau de deux dimensions qui contient des lettres représentant la forme de la carte dont chaque lettre représente un élément de la carte. Ce fichier contient également un ensemble de liste. Par ailleurs, la construction des éléments de la carte se fait en deux manieres:

1. La premiere méthode : en parcourant la grille, on transforme chaque lettre lue à un skin à l'aide de l'encodings;

2. La deuxième méthode : c’est en utilisant la liste des éléments de CarteData. Cette liste contient les attributs d’un élément ([element]).

Decoratif.java c’est les éléments décoratifs, il contient les champs name, skin, position et des méthodes pour créer un type Decoratif.

Enemy.java, contient les champs et les méthodes pour créer un type Enemy.

Friend.java contient les champs et les méthodes pour créer un type Friend

Intermitent.java contient les champs et les méthodes pour créer un type Intermittent.

Item.java contient les champs et les méthodes pour créer un type Item.

Obstacle.java contient les champs et les méthodes pour créer un type Obstacle.

Player.java contient les champs et les méthodes pour créer un type Player.

objets.primitive:

Comme son nom l’indique contient les objets de types primitifs qui sont utilisés pour créer des champs dans les autres packages, pour la sémantique et la lisibilité du code.

Couple.java ce fichier contient deux champs x et y permettant de créer un couple.

Direction.java c’est un enum qui représente les directions. (Pour la direction du joueur)

Flow.java c’est un enum qui représente les directions, pour le sens de l’écoulement de l’eau.

Skins.java il contient les listes des noms des skins.

Zone.java contient deux couples pour représenter une zone rectangulaire.

vue:

Ce package contient une classe Draw.java, dans laquelle on dessine tout, c’est a dire il contient toutes les méthodes statiques qui dessinent les objets du jeu (Player, Enemy, Decoratif, Item...)

menu.game:

Ce package contient les classes qui exécutent le jeu.

LoadImage.java, ce fichier contient une fonction qui créer les images

Game.java, c’est vraiment la boucle du jeu.

Main.java, charge la liste des cartes.

- **Les dossiers**

src : ce dossier contient les fichiers sources du projet
data: ce dossier contient deux sous-dossiers :

1. cartes, qui contient fichiers cartes(.map)
2. images, qui contient les images du jeu.

doc : ce dossier contient le rapport du projet.

lib : ce dossier contient zen5.jar.

- **Conclusion**

Ce projet nous a permis d'utiliser les concepts que nous avons étudié en classe et d'approfondir un peu plus dans le langage de programmation objet Java. En revanche on n'a pas réussi à implémenter toutes les spécifications du jeu. Par exemple, on affiche l'inventaire de l'ami, cependant on a pas fait les méthodes pour effectuer l'échange entre un joueur et un ami. Par contre, quand le joueur rencontre un ami automatiquement ils peuvent dialoguer, et l'inventaire de l'ami s'affiche avec un ensemble d'éléments aléatoire entre 3 et 10.

Annexe

Image d’une map avec des éléments :

Image d’un joueur tenant un objet comme une clé par exemple.