Il faut retrouver Ponette !

http://www.esiee.fr/~esencet/

(A3P(AL) 2012/2013 8H)

I.A) Auteurs

Eric TRAN
Thomas ESENCE

I.B) Thème

Lieu : A travers de nombre paysage (forêt, plaine, cascade, grotte, …)
Objectif : Retrouver Ponette

I.C) Scénario

Ce jeu retrace l'histoire de deux amis, Poney et Ponette.

Par un bel après-midi d'été, nos deux amis Poney et Ponette décidaient d'aller se rafraîchir dans une forêt. Plus ils s'avançaient, plus l'atmosphère était froid et obscure. C'est alors qu'au détour d'un arbre, Poney perdit Ponette de vue et commença à paniquer : elle avait disparu sans laisser de traces…
Ne connaissant pas les lieux, Poney fit demi-tour et rentra chercher l'aide de son ami Biactol. Une fois Biactol prévenu, nos deux courageux compagnons s'enfoncèrent dans la forêt, suivis de près... Mais une fois retournés sur les lieux de la disparition, non loin d'un gros chêne, un message avait été laissé à leur attention :

«Votre amie sera sauvée quand toutes les réponses auront étés trouvées.
Une fois ceci accomplit, l'indice ainsi déterminé, votre amie pourra être libérée.
».

Allez-Vous réussir à sauver Ponette… ?

I.D) Plan (complet, avec indication de la partie "réduit")

I.E) Scenario détaillé (complet, avec indication de la partie "réduit")

I.F) Détail des lieux, items, personnages

I.G) Situations gagnantes et perdantes

I.H) Eventuellement énigmes, mini-jeux, combats, etc.

TP1

Exercice 7.1

What does this application do? :

This application creates an object ***Game*** thanks to the Game class' constructor. Then we launch the game by using the method ***play().***

What commands does the game accept? :

The game accepts the following commands:

- go north/east/south/west
- quit
- help

What does each command do?

- **go north/east/south/west**: allows moving yourself in the associated direction

- **quit**: allows quitting the game

- **Help**: displays the following message:

"You are lost. You are alone. You wander around at the university.

Your command words are:
go quit help"

How many rooms are in the scenario?

There are 5 rooms:

- Outside the main entrance of the university
- Campus pub
- Lecture theatre
- Computer lab
- Computing admin office

Exercice 7.2

After you know what the whole application does, try to find out what each individual class does.
Write down for each class the purpose of the class.

Game : This class creates and initializes all other classes. It allows executing them.

Room : This class create the game's rooms and define commands of exit

Command : This class hold the informations the user send.

Parser : This class read user's inputs and try to interpret them as a game's command.

CommandWords : This class lists all the commands that the game accepts

Exercice 7.4

Open the zuul-bad project, and save it under a different name (e.g. zuul-v4). This is the project you will use to make improvements and modifications throughout this chapter.
You can leave off the -bad suffix, since it will soon (hopefully) not be that bad anymore. As a first step, change the createRooms method in the Game class to create the rooms and exits you invented for your game. Test!

```
private void createRooms()
  {
    Room foret, plaine, cascade, clairiere, prairie;

    // create the rooms
    foret = new Room("à la lisière de la forêt");
    clairiere = new Room("à l'entrée d'une petite clairière");
    prairie = new Room( "dans une vaste prairie");
    plaine = new Room("au milieu des plaines");
    cascade = new Room( "en face d'une grande cascade");

    // initialise room exits

    foret.setExits(null, clairiere, null, prairie);
    clairiere.setExits(cascade, null, null, prairie);
    prairie.setExits(plaine, cascade, null, null);
    plaine.setExits(null, null, prairie, null);
    cascade.setExits(null, null, clairiere, null);

    currentRoom = foret;  // start game in the forest

  }
```

Exercice 7.5

Implement and use a separate printLocationInfo method in your project, as discussed in this section. Test your changes

```
private void printLocationInfo()
   {
      System.out.println( "You are " + currentRoom.getDescription());

      System.out.print( "Exits: ");

      if(currentRoom.northExit != null)
         System.out.print( "north ");
      if(currentRoom.eastExit != null)
         System.out.print( "east ");
      if(currentRoom.southExit != null)
         System.out.print( "south ");
      if(currentRoom.westExit != null)
         System.out.print( "west ");
   }
```