Object-Oriented Programming

Project Statement

Academic Year 2020-2021

This project can be completed by groups of up to two students and must be submitted to `bvergain@uliege.be` *for April 25th at the latest. Projects returned after the deadline will not be corrected. Plagiarism is not tolerated, in line with the policy of the university (*`https://matheo.uliege.be/page/plagiat`*).*

The goal of this project is to program in Java a simple text-based adventure game.

The player controls a character that moves in a 2D world represented by a grid of square cells. Each cell is connected to one or several of its four neighbors by doors, that can possibly be locked. The character can change its orientation by increments of 90 degrees, move forward to the cell in front of him/her, and perform actions. Actions consist in picking up objects that are present in the cells, dropping them, or performing a specific operation on a given object. Every object in the game defines some operations that can be carried out on this object, possibly with the help of one or several other objects. For instance, a door can be locked or unlocked, provided that the character owns the correct key for this operation. An object such as a box can opened or closed, possibly with the help of a key, and may contain other objects that could then be picked up and used in the game.

The player interacts with the game by entering commands expressed in a simple syntax. In particular, the following commands are understood:

- `Turn` (`right` | `left`): Turn the player around by 90 degrees in the specified direction.
- `Move`: Move forward to the cell located in front of the character.
- `Take` *object*: Pick up the object *object*.
- *action object1* [`with` *object2*]: Perform action *action* on object *object1*, possibly with the help of object *object2*.
- `Help` [*object*] : Display a list of all commands understood by the game, with a brief documentation. With the argument *object*, display a list of all actions that can be performed on this object.

After processing a command, the game engine displays messages indicating the objects that are visible to the character in his/her current cell, including doors to neighboring cells, as well as the objects carried by the character. The results of an action are also clearly indicated. In particular, whenever an action fails, some information explaining how it could be carried out successfully is provided to the player.

An example of interaction between a player and the game is given below. (The commands entered by the player are prefixed by "`%`".)

```
Welcome to the game!

There is one door to the right, and one door behind you.
There are two keys and one box on the floor.
```

```
You do not carry anything.

% Help

Turn ( right | left ) : Turn in the specified direction
Move                  : Move forward
Take <object>         : Pick up object <object>
Drop <object>         : Drop object <object>
Help [ <object> ]     : This command

There is one door to the right, and one door behind you.
There are two keys and one box on the floor.
You do not carry anything.

% Turn right

There is one door in front of you, and one door to your right.
There are two keys and one box on the floor.
You do not carry anything.

% Turn right

There is one door in front of you, and one door to your left.
There are two keys and one box on the floor.
You do not carry anything.

% Move

The door is locked, you need a key to unlock it.

There is one door in front of you, and one door to your left.
There are two keys and one box on the floor.
You do not carry anything.

% Take key

There is one door in front of you, and one door to your left.
There are one key and one box on the floor.
You carry a key.

% Help door

Unlock <door> with <key> : Unlock door
Lock <door> with <key>   : Lock door

There is one door in front of you, and one door to your left.
There are one key and one box on the floor.
You carry a key.

% Unlock door with key
```

```
Wrong key.

There is one door in front of you, and one door to your left.
There are one key and one box on the floor.
You carry a key.

% Take key

There is one door in front of you, and one door to your left.
There is one box on the floor.
You carry two keys.

% Unlock door with key 2

Door unlocked.

There is one door in front of you, and one door to your left.
There is one box on the floor.
You carry two keys.

% Move

There is one door in front of you, and one door behind you.
There are two gold coins on the floor.
You carry two keys.
```

In addition to the elements defined in this statement, you are free to add you own objects and your own actions to the game. You are allowed to deviate slightly from the proposed syntax in order to accommodate your own game mechanisms. Be creative and have fun! But do not get carried away too much, the goal is to keep the project simple.

Notes:

- Your submission must take the form of a .zip or .tar.gz archive containing the Java source code of your project. After extracting the contents of this archive, it should be possible to compile your project with the command `javac AdventureGame.java`, and to run it with `java AdventureGame`.
- Your archive should also contain a file “`readme.txt`” containing the list of all commands and objects implemented in your game, together with an example of interaction with the player (similar to the one given in this problem statement).
- Your project will be evaluated with Java 8. It is thus important to make sure that your submission does not rely on features introduced in later versions of the Java language.
- The evaluation will take into account not only the correct implementation of the requirements of this problem statement, but also the compliance of your source code to the principes of object-oriented programming, as well as its modularity, readability, simplicity, and portability.