Laboratory 10: The World of You

Pre-lab

Louis Ohlow

This week's lab work will begin a larger project that will occupy us until the end of the semester. You will be working alone for the next three exercises.

What to Bring to Lab

Please bring these exercises printed out or written out with you to lab. Please have your name on your page. You may do this together with 1 or 2 others, but each of you should implement the game yourselves.

P1. Design your own game scenario away from the computer. Don't worry about implementation or clases or programming. Just try and come up with something interesting.

In this game the main Character is a student who is travelling through the world and different countries to find the motivation for learning for the exams.

P2. What is the goal of your game, that is, when does the player win?

When he finally finds the motivation in a country.

P3. What could you add to the game to make it interesting? Trap doors, treasure, monsters, ...

Wise People who already passed the university and are helping me findiing the motivation by giving me hints which are either helpful or not. You also have money you can spend either on the advise of those wise guys or on the next country you can travel to. The more hints you get the more exactly you know where to travel next but you won’t have the money to travel them all.

P4. Draw a map of your game layout.

Because the map represents the continents i just copied a map from the net

Assignment

World of You

The next tasks are inspired by the tutorium from Alexander Becker and i also changed the game to my version from the prelab plus the fact that you have an amount of money and you can win if you're entering a specific continent. (Room=continent)

1. **Start with the bad Zuul game and refactor it as discussed in the lectures. If you are doing mole burrows instead of rooms, you can change the variable names as needed. The bored can prepare a multi-lingual version and use enums.**

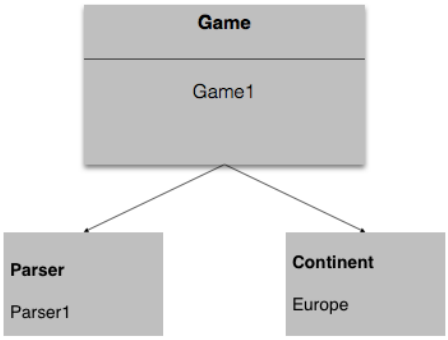
   Right here used a HashMap for every continent to use the exit continent as a Value and the cardinal direction as the Value. For this, i also had to create a method in the Continent (=Room) class to make sure i don't have to create a new HashMap eveytime.

```
public Continent(String description)
{
    this.description = description;
    this.exits = new HashMap<String, Continent>();
    this.detailedDescription = description;
}

public void setDetailedDescription(String text){
   detailedDescription += "\n" + text;
}

public HashMap<String, Continent> getExits(){
  return exits;
}

/**
 * Define the exits of this Continent.  Every direction ei
 * to another room or is null (no exit there).
 * @param north The north exit.
 * @param east The east east.
 * @param south The south exit.
 * @param west The west exit.
 */
public void setExit(String direction, Continent whereToGo)
{
    exits.put(direction, whereToGo);
}

public String getExitsAvailable(){
    String result = "";

    for(String key : exits.keySet()){
        result += key + " ";
    }
    return result;
}
```

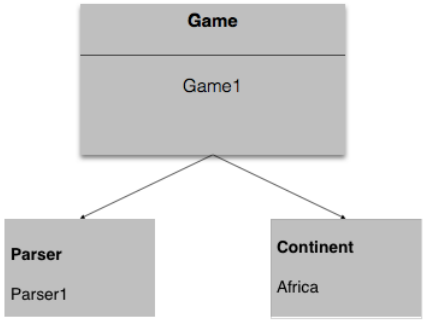
```
Australia.setExit("north", Asia);
Australia.setExit("south", Antarctica);
Australia.setExit("west", Africa);

Antarctica.setExit("north", Africa);
Antarctica.setExit("east", Australia);
Antarctica.setExit("west", SouthAmerica);

SouthAmerica.setExit("north", NorthAmerica);
SouthAmerica.setExit("east", Africa);
SouthAmerica.setExit("south", Antarctica);

NorthAmerica.setExit("east", Europe);
NorthAmerica.setExit("south", SouthAmerica);

Asia.setExit("west", Europe);

Europe.setExit("north", Antarctica);
Europe.setExit("east", Asia);
Europe.setExit("south", Africa);
Europe.setExit("west", NorthAmerica);
Africa.setExit("north", Europe);
Africa.setExit("east", Asia);
Africa.setExit("south", Antarctica);
Africa.setExit("west", SouthAmerica);

currentContinent = Europe;  // start game at h
```

2. **Draw an object diagram showing the state of your system just after it has been started. Does it change if you issue a "go" command**?

After starting:



after typing „go south“:



3. **Add a "look" command to your game.**

Now, i needed to add the „look“ command first in the command class so it will be recognized as a Commandword. Afterwards, i just had to add another else if statement which can still be filled with some field to declare what should happen when „look“ is typed in.

```
else if (commandWord.equals("quit")){
    wantToQuit = quit(command);
}
else if(commandWord.equals("look")){
}

return wantToQuit;
```

```
// a constant array that holds all valid command words
private static final String[] validCommands = {
    "go", "quit", "help", "look"
```

4. **Add an additional command (such as "eat", which for now just prints out "You have eaten now and are not hungry any more". In the next exercise, when we have added items, you can make it so that you can only eat if you have found food.**

   Now, i used the command dig instead of eat because it fit's

```
}
else if(commandWord.equals("dig")){
    System.out.println("You digged a hole but did'nt find anything useful..");
}

return wantToQuit;
```

   better to my game. This was the same process as in Nr. 3

5. **Implement an improved version of printing out the command words.**

   To create an improved version, i had to create another Commandword object in the game class to access the a new method which prints out every command from the array.

```
public void commandSum(){
 for(int i=0; i< validCommands.length;i++){
  System.out.println(validCommands[i]);
    }
}
```

```
    private CommandWords commandwords;
    /**
     * Create the game and initialise its internal map.
     */
    public Game()
    {
        createContinents();
        parser = new Parser();
        rich = true;
        money = 1200;
        isfound = false;
        commandwords = new CommandWords();
    }
```

```
private void printHelp()
{
    System.out.println("not motivated yet.. will you ever be?");
    System.out.println("travel somewhere you haven't been before");
    System.out.println("just by typing go and the direction");
    System.out.println("Your command words are:");
    commandwords.commandSum();

}
```

6. **Add another command - did you have to change the Game class? Why or Why not?**

   No i don't have to except if i want it to return something like a message. In this case i have to add another if clause in the processCommand class.

```
else if(commandWord.equals("look")){
    System.out.println("you've looked around");
}
else if(commandWord.equals("dig")){
    System.out.println("You digged a hole but did'nt find anything useful..");
}
else if(commandWord.equals("ask")){
    System.out.println("first think about what you'd like to ask");
}
return wantToQuit;
```