Mars Rover Task Documentation

Description of the methods in file: **mars\_rover\_task\_class.py**

The program is based on the scenario, where several rovers are deployed on Mars and with the accurate instructions, they have to send information, about the planet (pictures).

More detailed information about the task, you can find in the pdf document: **Mars Rovers Practical Test (Internship exercise).pdf** -> attached in the email.

I have explained every move and step, into the **mars\_rover\_task\_class.py**.

**My comments are before each step or in front of the method I am describing.**

Below you can find the methods and their actions explained.

1. **ROTATE\_RIGHT\_DICT** 🡪 Dictionary with the direction, which will take the rover after right rotation command appears.
2. **ROTATE\_LEFT\_DICT** 🡪 Dictionary with the direction, which will take the rover after left rotation command appears.
3. Property **move\_rover 🡪** method, which is used for moving the rover after command 'M' appears according to the current direction**.** Returns the rovers coordinates after the move is made.
4. Property **return\_rover\_to\_previous\_position 🡪** method, which returns the rover to its previous position, if the current coordinates after command 'M' are not valid (out of the plateau dimensions). The property returns the rover’s coordinates after it have been detected that after the move, its coordinates are not valid.
5. Static method: **\_check\_rover\_valid\_coordinates 🡪** method that checks if the current rover coordinates are valid (Not out of the plateau dimensions).
6. Property **rotate\_right 🡪** Property for left rotation. Returns the rover’s direction after some command (‘L’ or ‘R’) appears.
7. Property **rotate\_left**  Property for left rotation. Returns the rover’s direction after some command (‘L’ or ‘R’) appears.
8. Method **print\_rovers\_final\_coordinates ->** method for printing the final output after rovers are finished their moves.
9. I used 2 For Cycles in order to go through all the commands and details, which comes from the hard coded inputs.
10. There is only one class created: **MarsRover.** I don’t think more are necessary.

I created several instances of the same class in order to have couple of examples with different inputs.

Every hard coded details are tested several times.

Miroslav Vatov 03.06.2021