Assignment 3 – State Machines
UML State Machine

In this assignment, you will practice developing UML State Machines to specify system behavior, and will practice writing code that conforms to the State Machine model. You can draw UML State Machine model by hand, or by using any software tool of your choosing.

Lunar Rover

Back in July 30th 1971, the crew of Apollo 15 reached the moon surface. The first American astronaut reached the moon along with his team. But they needed a mean of transportation. Their solution: A lunar buggy called "Lunar Rover" that could travel speeds up to 12 km/h. This was an awesome opportunity for astronauts to travel long distances outside Earth, but there were some technological constraints 45 years ago. The Lunar Roving Vehicle could had only two pedals and two buttons to control the vehicle's systems.

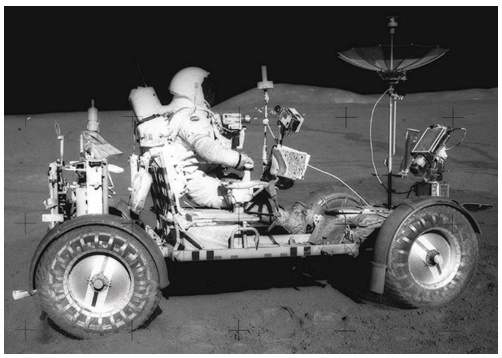

Figure 1: Lunar Rover on the surface of the moon

Movement Control:

Engineers had to be creative in order to add all the functionality need for a lunar cruise, so they came up with the following ideas:

- When the left pedal was pressed once it accelerates the buggy forward.
- If accelerating forward and you press right pedal twice it deaccelerates.
- To achieve constant forward speed press the right pedal for more than 5 seconds.
- If the buggy is at rest and the left pedal is pressed for more than 5 seconds, it will accelerate backwards.

1. Design a state machine to model the Movement Control as described above (5 points).
2. Provide an implementation for the state machine in Java (5 points)
   Every state should implement an entry action that prints on the console the current state.

Note: The description above is not complete. Your task includes filling in these missing details. Make sure to write down any assumptions you make as you develop the state machine model.

Camera & Drill (Bonus – 4 points):

The coolest feature of the buggy was not it's speed but rather the color television camera, the 16-mm camera, and the drill. The control was packed inside a television control unit. The only problem was that the cameras, and the lunar rover drill were also controlled by the same control unit. So again engineers used the two button controller to manipulate all devices:

- Press the button 1 for five seconds to interact with the color camera.
- Press the button 1 for ten seconds to interact with the 16-mm camera.
- Press button 1 twice to interact with the drill.
- Pressing button 2 at any of these modes return to idle state.
- In any of the two camera modes, pressing button 1 takes a picture, while if pressed for 5 seconds activates the temporizer (moon selfies).
- Last inside the drill mode button 1 only acted as an on off switch.

1. Design a state machine to model the Camera and Drill. Extend the model you developed for the movement control (2 points).
2. Provide an implementation for this extended state machine in Java (2 points).
   Every state should implement an entry action that prints on the console the current state.

   Note: Submit the bonus part as a separate state machine and separate code.

   What to submit

   Submit your state machine models, code, and sample runs. Submit by the deadline, late assignments are not accepted.