Century College

CSCI 2082.90

Fall 2016

## PA1

## Due on 09/13 @ 11:30PM

(25 Points)

***(Project 10: Page 98)*** Specify, design and implement a class that can be used to simulate a lunar lander, which is a small spaceship that transport astronauts from lunar orbit to the surface of the moon. When a lunar lander is constructed, the following items should be initialized as follows:

1. Current fuel flow rate as a fraction of the maximum fuel flow (initially zero)
2. Vertical speed of the lander (initially zero meters/sec)
3. Altitude of the lander (specified as a parameter of the construction)
4. Amount of fuel (specified as a parameter of the construction)
5. Mass of the lander when it has no fuel (specified as a parameter of the constructor)
6. Maximum fuel consumption rate (specified as a parameter of the constructor)
7. Maximum thrust of the lander’s engine (specified as a parameter of the constructor)

Don’t worry about other properties (such as horizontal speed).

The lander has accessor methods that allow a program to retrieve the current values of any of the preceding seven items.

The first modification method changes the current fuel flow rate to a new value ranging from 0.0 to 1.0. This value is expressed as a fraction of the maximum fuel flow.

The second modification method simulates the passage of a small amount of time. This time, called ***t***, is expressed in seconds and will typically be a small value such as 0.1 seconds. The method will update the first four values in the preceding list to reflect the passage of t seconds. To implement this method, you will require a few physics formulas, listed below. These formulas are only approximate because some of the lander’s values are changing during the simulated time period. But if the time span is kept short, these formulas will suffice.

**Fuel flow rate:** Normally, the fuel flow rate does not change during the passage of a small amount of time. But there is one exception: if the fuel flow rate is greater than zero and the amount of fuel left is zero, then you should reset the fuel flow rate to zero (because there is no fuel to flow).

**Velocity change:** During t seconds, the velocity of the lander changes by approximately this amount (measured in meters/sec):

***t \* ( (f/m) - 1.62)***

The value ***m*** is the total mass of the lander, measured in kilograms (i.e., the mass of a lander with no fuel plus the mass of any remaining fuel). The value ***f*** is the thrust of lander’s engine, measured in newtons. You can calculate ***f*** as the current fuel flow rate times the maximum thrust of the lander. The number -1.62 is the downward acceleration from the gravity on the moon.

**Altitude change**: During ***t*** seconds, the altitude of the lander changes by ***t \* v***meters, where v is the vertical velocity of the lander (measured in meters/sec with negative values downward).

**Change in remaining fuel:** During ***t*** seconds the amount of remaining fuel is reduced by ***t \* r \* c*** kilograms. The value of r is the current fuel flow rate, and ***c*** is the maximum fuel consumption (measured in kilograms per second)

We suggest that you calculate the changes to the four items in the order just listed. After all the changes have been made, there are two further adjustments. First, if the altitude has dropped below zero, then reset both altitude and velocity to zero (indicating that the ship has landed). Second, if the total amount of remaining fuel drops below zero, then reset this amount to zero (indicating that we have run out of fuel).

Graphical User Interface (GUI) basic requirements:

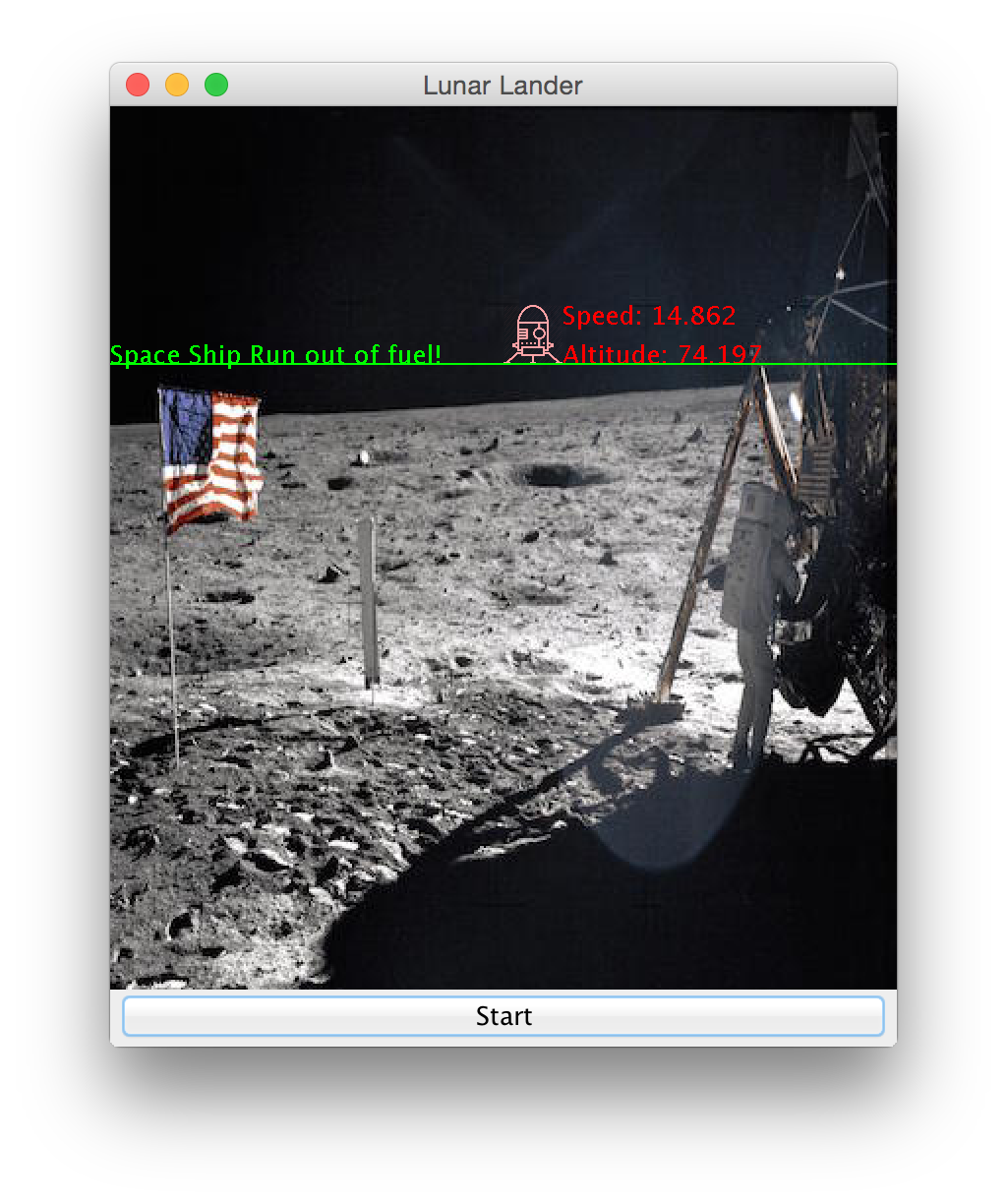
* Button to start the simulation.
* Display the updated values of the speed, altitude and remaining fuel of the lunar lander.
* Indicate the status of the lunarlander:
  + Landed
  + Run out of fuel

With or without the GUI you still need to display the speed, altitude, and remaining fuel to the console.

Your program will be graded based of the followings:

|  |  |
| --- | --- |
| **Compiles and runs correctly** | 16 Points |
| **Method’s specifications** | 4.5 Points |
| **Package and project structure** | 2 Points |
| **Graphical User Interface** | 2.5 Points |

Here is a simple screen shot (Yours doesn’t have to look like this one and you don’t have to use background image):



## What to turn in:

1. Create a project called LunarLander
2. Add a package called century.edu.pa1 to your project
3. Add all your classes to the century.edu.pa1
4. Generate a Zip file of your LunarLander project and that’s what you need to upload to pa1 dropbox.