GUIDE TO THE PROJECT

NEW IN MECHANICS

1. Training the PID CONTROLLER:
   An offline simulation of the lander was written in python. Nelder mead simplex search algorithm was used to find the optimum values for three cases – minimum descent velocity, minimum descent time, minimum fuel usage.
   The detailed python implementation and discussion on tuning the PID Controller is in the "pid.ipynb" python notebook.
2. Added controls for any-angle altitude control. These controls will only work if the lander is in attitude stabilization mode (scenarios 1 to 5) The user can rotate the lander clockwise or anticlockwise using keyboard inputs (further documented in the graphics section of this document). In scenarios 7 and 8 where the autopilot uses any – angle attitude control.
3. Scenario 6 – Added a scenario for the "Aerostationary" orbit of mars.
4. Scenario 7 – Added a case for orbital injection into any orbit with arbitrary specified apogee and perigee. This scenario also incorporates the option of de-orbiting and landing safely back which should be used after the lander is correctly injected into the orbit. A new key-board control and indicator lamp in the graphics has been added for this purpose. On pressing "d", the lander would enter deorbiting and landing mode. It would then land smoothly onto the earth's surface. To specify arbitrary values of apogee and perigee, please change the corresponding values in the initialize_simulation function in lander.cpp file.
5.

```
case 7:

    position = vector3d(0.0, -( MARS_RADIUS + LANDER_SIZE/2 + 1), 0.0);
  // position = vector3d(0.0, -( 1.2 * MARS_RADIUS ), 0.0);
    velocity = vector3d(0.0, 0.0, 0.0);
    orientation = vector3d(0.0, 0.0,0.0);
    delta_t = 0.1;
    parachute_status = NOT_DEPLOYED;
    stabilized_attitude = true;
    autopilot_enabled = true;
    perigee = MARS_RADIUS * 9;
    apogee = MARS_RADIUS * 1.2;
    infinite_fuel = true;
    autopilot_no = 3;

    break;
```

   NOTE : Please specify apogee and perigee values greater than (MARS_RADIUS + EXOSPHERE) – so that the lander does not suffer from drag once injected in orbit. The perigee and apogee values are the absolute distances from the centre of the coordinate system and are not altitudes measured from the surface of Mars. For running this scenario, you might want to run on the maximum simulation speed.

6. Scenario 8 – Added a case for orbital transfer from a circular orbit to an orbit with apogee equal to the radius of the circular orbit and perigee greater than the radius. To initiate this hoffman-like transfer, again press the keyboard control "d". To tweak around with the scenario, change the radius in initialize_simulation function and the global variable new_perigee at the top of lander.cpp file.

```
case 8:

    radius = 1.25 * MARS_RADIUS;
    req_velocity = sqrt((GRAVITY * MARS_MASS  / radius ));
    position = vector3d(0.0, - radius, 0.0);
    velocity = vector3d(req_velocity, 0.0, 0.0);
```

```
 //#include "lander.h"
#include <cmath>
#include "lander.h"

// variable for case 8 to specify the
// is the radius of the circular orbi
double new_perigee = 2 * MARS_RADIUS;
```

NOTE : Again, please specify the values of new_perigee and radius greater than (MARS_RADIUS + EXOSPHERE)

Autopilots for both Scenarios 8 and Scenarios 9 use any angle attitude control to manoeuvre the lander appropriately.

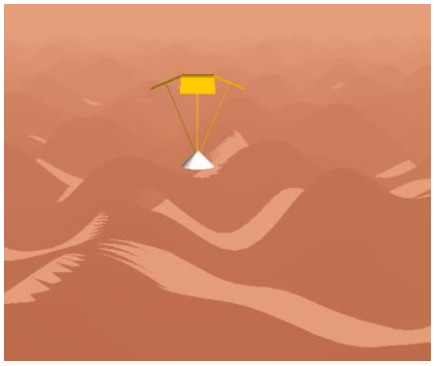
7. – Scenario 9 - Added a case for descent from 3386000km. To tweak around, simple change the position vector in initialise_simulation function in lander.cpp.

```
case 9:


    position = vector3d(0.0, -(2*MARS_RADIUS), 0.0);
    velocity = vector3d(0.0, 0.0, 0.0);
```

NEW IN GRAPHICS:

1. To generate realistic terrain for mars in the closeup window, Perlin 2D noise is used. A mesh with with triangular strips is created in one plane and the heights are determined by the Perlin 2D function noise. This is best visualized when the lander has not yet landed.



2. 4 new indicator lamps – autopilots for minimum descent velocity, minimum descent time, minimum fuel usage and deorbiting/transferring - have been added to the instrument window
   The keyboard controls are –
   "a" – autopilot for minimum descent velocity
   "b" -- autopilot for minimum descent time
   "c" – autopilot for minimum fuel usage
   "d" – autopilot for deorbiting and transferring once in "scenarios 7" and " scenarios 8 ".
   NOTE : its best not to change the autopilot from one mode to other in scenarios 1 to 5. In case you might want to change, you might have to press the respective control key twice.
   (i.e press twice the keyboard control for the new autopilot mode you want to go into until you see its indicator lamp switched on). The last autopilot "d" only is enabled to work for scenarios 7 and 8.
3. 2 new keyboard controls have been added for any – angle – attitude control
   "r" = rotate the lander clockwise
   "u" – rotate the control anticlockwise
   NOTE: These keyboard controls only work if the lander is in attitude stabilization mode and is in scenarios 1 to 5.

The corresponding graphics for the help text has also been updated.