Mars Lander - Soham Karmarkar, sk2111, King's College

GRAPHICS

One of my main interests in pursuing this project beyond the required tasks was learning how to improve the graphics. I started off small: I added arrowheads onto the dial ticks, used anti-aliasing to make the dial ticks look smoother, added colour gradients to the fuel, simulation speed, throttle bars, and even to the lander as it gains speed in the atmosphere.

As my understanding of legacy OpenGL improved, I began improving the graphics in more substantial ways: I added legs and solar panels to the lander that extend out of the lander as it lands. I also made the parachute more realistic, so that 3 hemispherical parachutes are deployed rather than a singular blocky one. Upon successful landing the parachutes are animated to fall to the ground. I added a label on the lander which indicates the layer of the atmosphere that it is in. Similarly, there's an arrow which shows the wind load on the lander. I made the exhaust flare look a bit more 3D by simply adding more triangles around the cone. I really wanted to add realistic textures to the planet, but despite many attempts, I could never get it to work. I did, however, manage to change the caps of the sphere to resemble the polar caps of Mars. Finally, I manually generated some terrain, which can be seen upon successful landing in scenario 7, because all Fracplanet installs were either outdated or corrupted.

I used the ImGui library to add sliders and buttons to change the value of constants without having to recompile every single time. As the instrument subwindow became a bit crowded, ImGui allowed for significantly more opportunities to test and customise the program, as well as provide prompts and help to the user. For instance, I used the slider in scenario 8 to tune the autopilot, by seeing how far the lander could drop before actually needing autopilot to land successfully. After several attempts, I managed to add images, which I decided to incorporate as a "fun facts" tab about Mars. To avoid the ImGui display crowding the main window, I created a separate window for it. **If you experience lag at high simulation speeds, simply minimise the ImGui window or just comment out the ImGui window code i.e. lines 3271 to 3292.**

AUDIO

Finally, I decided to add even more depth to the program by adding music with the IrrKlang library! I used the Interstellar theme song as background music and have added a fanfare sound and a crash sound for successful and unsuccessful landings, respectively. There are also the options to change volume and mute as well in the ImGui window! To not oversaturate the program with audio, I decided not to add more sounds,

.

OTHER EXTENSION TASKS

- I added an areostationary scenario by calculating the required velocity and radius.
- Using axis-angle rotation, I added the 'g' and 'h' keys to rotate the craft in the plane of the orbit.
- The autopilot now uses the parachute, allowing landing from any orbital scenario.
- The program models wind varying with altitude, as recorded by NASA's data of Mars, and also has the option of adding random gusts of wind, which the autopilot can successfully land with.
- The orbital injection function (key: 'i') uses Hoffman transfers to enter orbit. The eccentricity can be varied in the ImGui display.
  [Start in scenario 1 and press 'i' before the lander crashes. You can change the simulation speed to max until it enters a stable orbit]
- The orbital reentry function (key: 'r') allows the lander to land from any of the orbits.
- The autopilot, or anything really, can be tuned by changing certain constants in the ImGui menu.