

Lunar Lander System – Documentation

1. How does your physics system work?

Instead of relying on Godot's rigid body physics, forces are applied manually and integrated into velocity and position. Also collisions are interesting. When the Lunar Lander collides with something it should not, we pause the `process()` function of its script, which stops it and then we play audio and visuals that indicate a crash. The collision checks are done with `Area3D` nodes. We check if the lunar lander entered and then pause the game. This workaround is due to me being unsure whether we could have used the built in `slide_and_move()` godot function, which is part of the physics engine and applies a built in velocity vector to the character.

- **Basic Physics**

When the player applies input (arrows or WASD), a thrust force is added to the force accumulator.

Vertical thrust is applied with `ui_up` (ascend) and `ui_down` (descend).

Horizontal thrust is applied in the world's local x/z directions for WASD keys.

Originally, I had different strengths for horizontal and vertical thrust. Inside a 3D Vector called force we save the upwards and downwards thrust. inside another we save the direction of WASD input. IF the WASD input is not zero, we add it to the force vector and then we do this every tick:

```
acceleration = force / mass
```

```
velocity += acceleration * delta
```

```
global_position += velocity * delta
```

- **Tilting**

To simulate helicopter/lander tilt, an interpolation (`lerp`) is performed between the target tilt (based on input direction) and the current tilt. This changes the node's rotation degrees for pitch (X-axis) and roll (Z-axis), without directly affecting physics.

- **Landing Detection**

Two `Area3D` nodes (front wheel and back wheels) detect contact with a landing pad group.

- If both are on the pad and vertical speed (`velocity.y`) is below the safe landing speed, the landing is successful.
- Otherwise, the lander crashes, triggering particle explosions and restarting the scene.

- **Fuel System**

Thrust consumes fuel at a burn rate of 10 units/second.

If fuel is depleted, thrust inputs no longer produce force, and the engine sound stops.

2. What values did you set your parameters to?

"Level1":

OptionsMenuVars.gravity = -1.625

OptionsMenuVars.landing_speed = 20.0

OptionsMenuVars.fuel = 100.0

OptionsMenuVars.thrust = 15000.0

"Level2":

OptionsMenuVars.gravity = -3.711

OptionsMenuVars.landing_speed = 25.0

OptionsMenuVars.fuel = 250.0

OptionsMenuVars.thrust = 12000.0

"Level3":

OptionsMenuVars.gravity = -3.711


OptionsMenuVars.landing_speed = 250.0

OptionsMenuVars.fuel = 80.0

OptionsMenuVars.thrust = 37000.0

3. How is the lunar lander being controlled (input keys)?

- **Vertical movement:**

-  (ui_up): Ascend (apply upward thrust)

- ↓ (ui_down): Descend (apply downward thrust)
- **Horizontal movement (thrust in x/z directions):**
 - W: Move forward, pitch nose down
 - S: Move backward, pitch nose up
 - A: Move left, roll left
 - D: Move right, roll right
- **Rotation (yaw):**
 - ← (ui_left): Rotate left around Y-axis
 - → (ui_right): Rotate right around Y-axis
- **Other controls:**
 - When pressing the space bar the camera gets reset. I planned on doing zooming, but ran out of time.
 - Looking around with mouse.