Rocket Simulation Program Documentation

This C++ program simulates the launch and flight of a rocket using various parameters and models to approximate altitude, speed, and orientation. The program incorporates Euler angles for representing orientation and models the impact of these angles on the rockets fin angles, affecting its flight stability. Key elements of the rockets dynamics, such as fuel consumption, thrust, altitude, and vertical speed, are computed over simulated time steps until the fuel runs out and the rocket lands.

Components and Classes

The program is organized into three main classes:
1. EulerAngles: Simulates changes in roll, pitch, and yaw angles, representing the rockets orientation in 3D space.
2. FinAngle: Represents the angle of each fin and is adjusted based on the rockets orientation to maintain stability.
3. Rocket: Manages the rockets fuel, thrust, altitude, and speed, and controls the fins using EulerAngles for stability.

Class Details

EulerAngles

- Attributes:
  - roll, pitch, yaw (type: double): Represent the rockets orientation in degrees along three axes.
- Methods:
  - EulerAngles(): Initializes roll, pitch, and yaw to 0.
  - operator++: Overloads the ++ operator to simulate random changes in orientation. Roll, pitch, and yaw change by small random values to simulate atmospheric fluctuations.
  - display(): Displays the current roll, pitch, and yaw values to the console.

FinAngle

- Attributes:
  - angle (type: double): Represents the angle of an individual fin.
- Methods:
  - FinAngle(): Initializes angle to 0.
  - setAngle(double pitchInfluence, double yawInfluence, double rollInfluence): Sets the fin angle based on the rockets current orientation (pitch, yaw, and roll).
  - display(int finNumber): Displays the fin number and its current angle.

Rocket

- Attributes:
  - fuel (type: double): The rockets fuel in kilograms.
  - mass (type: double): The total mass of the rocket in kilograms.
  - thrust (type: double): The rockets thrust force in Newtons.

  - burnRate (type: double): The rate at which fuel is consumed (kg/s).
  - altitude (type: double): The rockets altitude above the ground.
  - verticalSpeed (type: double): The vertical component of the rockets speed.
  - acceleration (type: double): The rockets current vertical acceleration.
  - fins[4] (array of FinAngle): Array to represent the four fins of the rocket.
- Methods:
  - Rocket(double initialFuel, double rocketMass, double thrustForce, double burnRate): Initializes the rockets parameters with provided values.
  - operator+=(double timeStep): Overloads the += operator to simulate fuel consumption and updates altitude and speed at each time step.
  - getAltitude(): Returns the rockets current altitude.
  - operator[](int index): Overloads [] to access a specific fin.
  - isFuelAvailable(): Checks if fuel is still available.
  - displayStatus(): Displays the rockets fuel level, vertical speed, and acceleration.
  - preLaunchCheck(): Simulates a pre-launch checklist to confirm that systems are ready.

Simulation Workflow

1. Initialization:
   - The user inputs the initial fuel, thrust, and burn rate of the rocket.
   - Rocket, EulerAngles, and fin objects are initialized based on these parameters.

2. Pre-Launch Check:
   - The preLaunchCheck() function simulates a systems check with brief delays.

3. Launch Loop:
   - The loop runs as long as fuel is available or the rockets altitude is above zero.
   - The program simulates incremental changes in Euler angles (roll, pitch, and yaw).
   - Fin angles are set based on the current Euler angles, adjusting their orientation for flight stability.
   - The Rocket object updates altitude, speed, and acceleration based on fuel, burn rate, and thrust.
   - After fuel is exhausted, only gravity affects the rocket, decreasing altitude and speed until it lands.

Key Features and Edge Cases

- Fuel Depletion: The rockets altitude stops increasing once fuel is exhausted. Gravity acts alone afterward, decreasing altitude until the rocket lands.
- Randomized Euler Angles: The operator++ for EulerAngles introduces small random changes to simulate unpredictable environmental factors.
- Fin Control: Fin angles are dynamically set based on Euler angles to maintain stability, crucial for guidance in real-world rockets.
- Time Delay for Visualization: The program uses a delay to simulate real-time progression, making the console output readable.