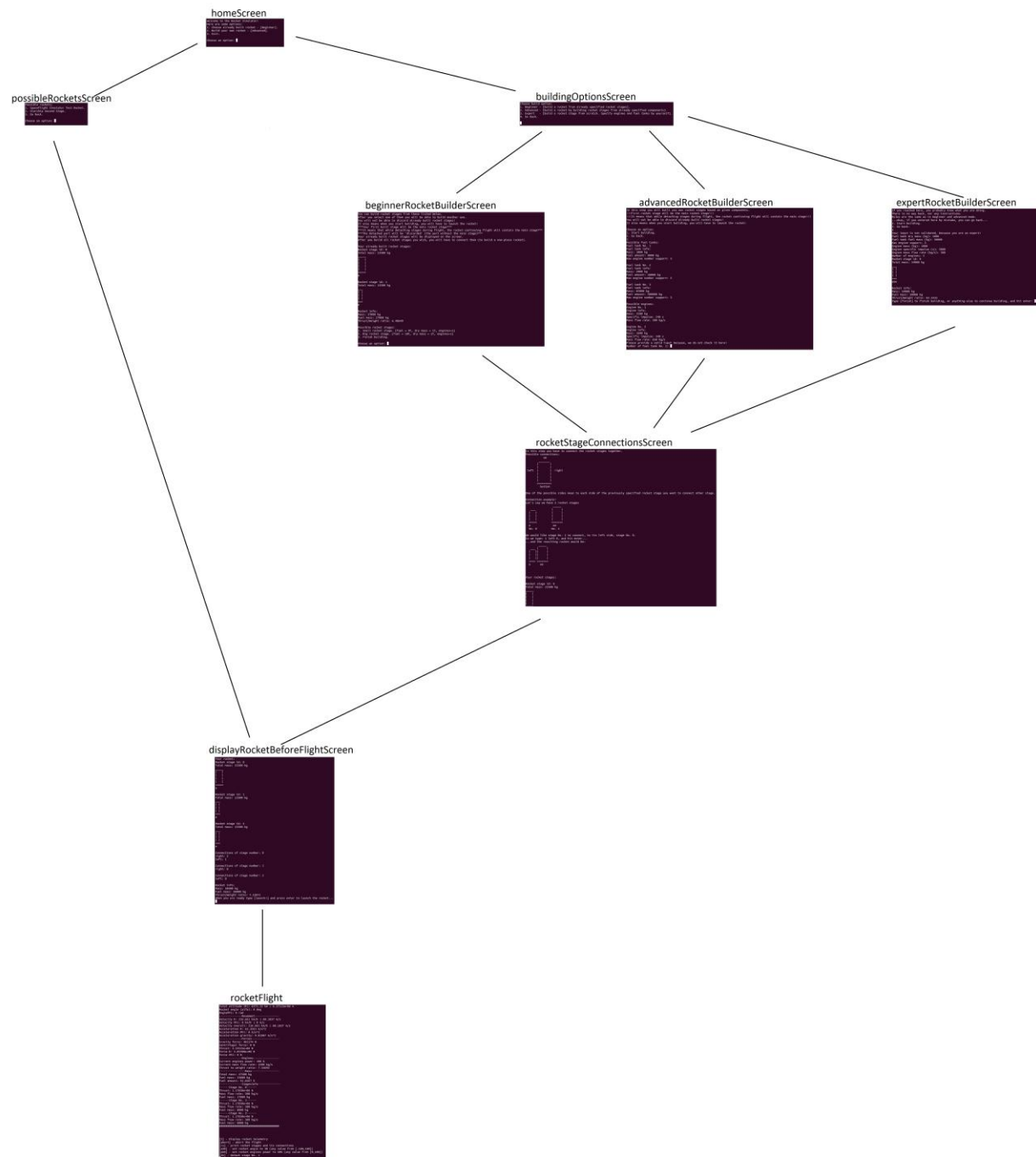# Spaceflight simulator possible user paths

src: /documentation/pages/allScreensPathNames.jpg

Flight physics

During the flight we can control the rocket by:

- Changing rocket angle (α)
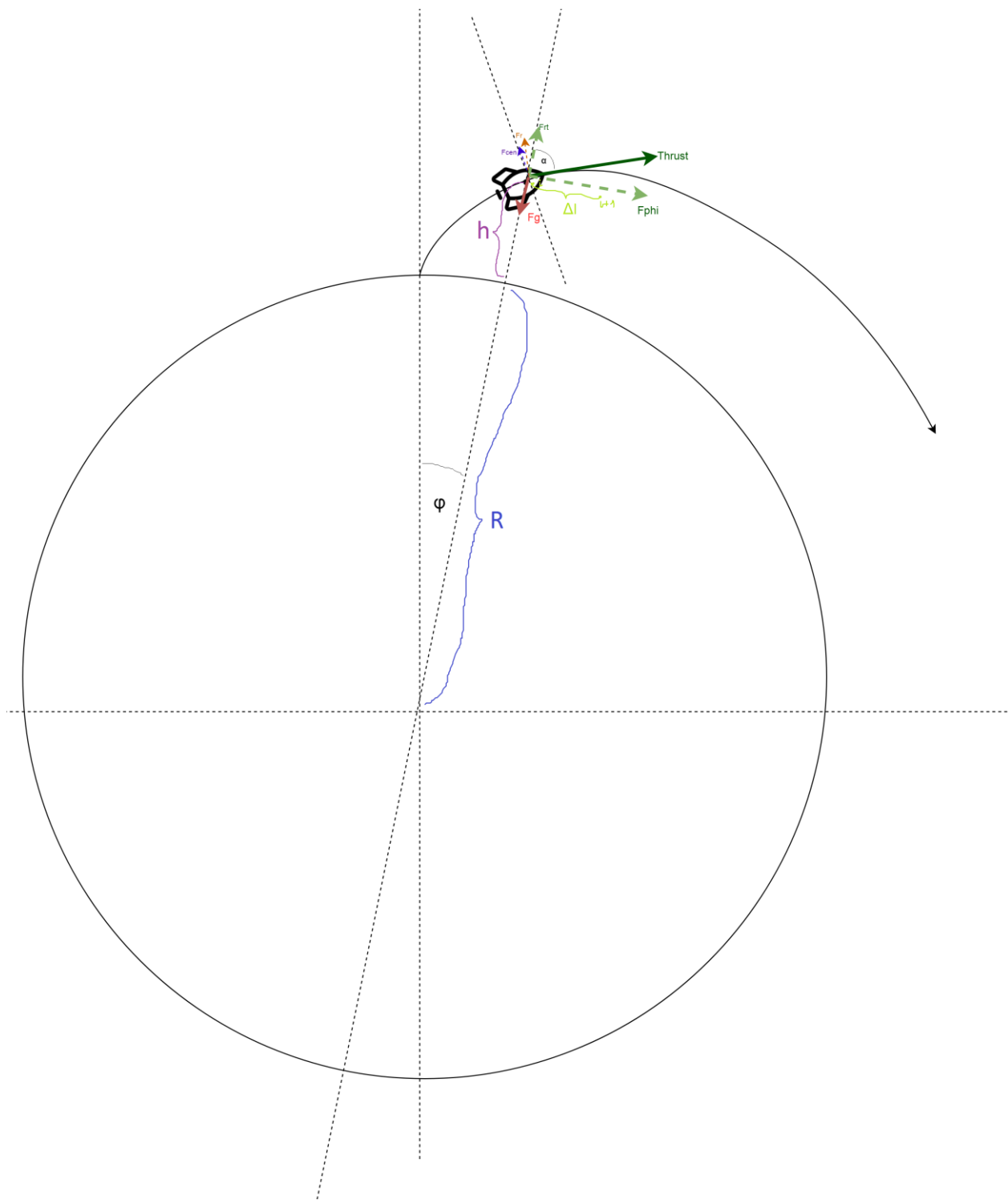- Changing Power for all engines
- Detaching rocket stages.

All flight data is computed by interval time $\varDelta t = 1\ s$.

Gravity varies with altitude.

Rocket mass is decreasing while engines are on, because engines burn some fuel during $\varDelta t$. It is crucial in the simulation.
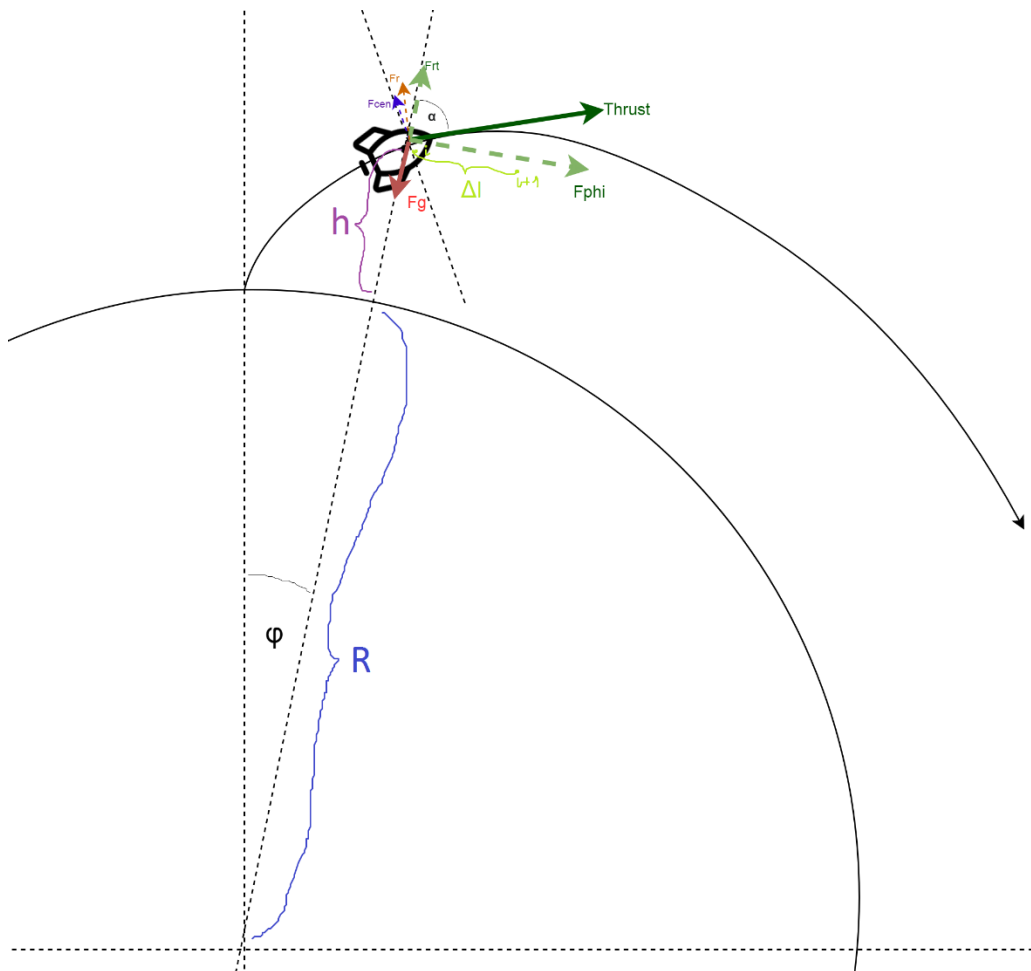
src: /documentation/physics/physics.jpg

Frt

Fr

Fcen

α

Thrust

Δl

Fphi

Fg

h

φ

R

Forces

$$F_{r_i} = Thrust_i \cdot \cos\alpha - F_{g_i} + F_{cen_i} \cdot \sin\alpha$$

$$F_{g_i} = m_i \cdot g_i$$

$$Thrust_i = g_i \cdot specificImpulse \cdot massFlowRate \cdot \left(\frac{currentPower}{100}\right)$$

$$F_{cen_i} = V_{\varphi_{i-1}} \cdot \frac{\sin(\Delta\varphi_{i-1})}{\Delta t} \cdot m_i$$

$$F_{\varphi_i} = F_i \cdot \sin\alpha$$

Acceleration and velocity

$$a_{r_i} = \frac{F_{r_i}}{m_i}$$

$$V_{r_i} = a_{r_i} \cdot \Delta t + V_{r_{i-1}}$$

$$a_{\varphi_i} = \frac{F_{\varphi_i}}{m_i}$$

$$V_{\varphi_i} = a_{\varphi_i} \cdot \Delta t + V_{\varphi_{i-1}}$$

Distance and angle

$$\Delta h_i = V_{r_i} \cdot \Delta t + \frac{a_{r_i} \cdot \Delta t^2}{2}$$

$$\Delta l_i = V_{\varphi_i} \cdot \Delta t + \frac{a_{\varphi_i} \cdot \Delta t^2}{2}$$

$$\frac{\Delta l_i}{R + \Delta h_i} = \sin(\Delta\varphi_i) \approx \Delta\varphi_i \text{ for small angles}$$

$$\Delta\varphi_i = \sin^{-1}\left(\frac{\Delta l_i}{R + \Delta h_i}\right)$$

$$\varphi_{i+1} = \varphi_i + \Delta\varphi_i$$

$$h_{i+1} = h_i + \Delta h_i$$

Determining the position of the rocket in space

All calculations are performed in the polar coordinate system (because it is much easier). So the rocket position is determined by some point O and rocket's distance from point O (distance = R + altitude in our case), and the angle φ.
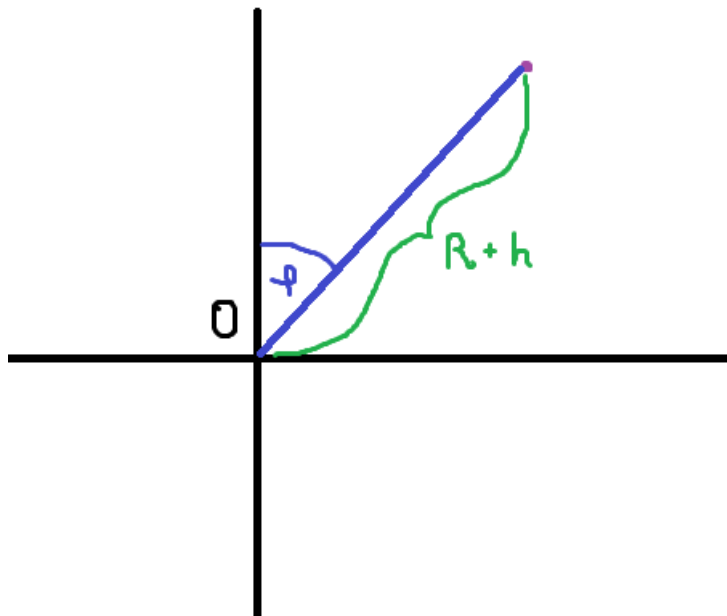
We can easily convert these values into Cartesian coordinates

$$x = (R + h) \cdot \sin \varphi$$

$$y = (R + h) \cdot \cos \varphi$$

Flight simulator telemetry data

After the flight, three files are generated:

1. rocketTelemetry_live.txt
   a. this file contains values for all variables during the flight in each time interval.
   b.
   ```
   4|0|6371.03|6371|6.37103e+06|1.17839e+06|0|10.9893|3.81885|9.81991|848440|0|329949|0|0|0
   5|0.000358975|6371.05|6371|6.37105e+06|1.17838e+06|0|14.8082|3.87943|9.81987|843527|0|333243|61671.9|3|0.0523599
   6|0.001438|6371.07|6371|6.37107e+06|1.17838e+06|0.71795|18.6876|3.95961|9.81982|838613|0|338151|61671.6|3|0.0523599
   7|0.0032413|6371.09|6371|6.37109e+06|1.17837e+06|1.4401|22.6472|4.04074|9.81976|833697|0.0103234|343059|61671.2|3|0.0523599
   ```
2. rocketTrajectory.png
   a. this file show rocket trajectory zoomed into rocket it means that only a part of earth is visible.
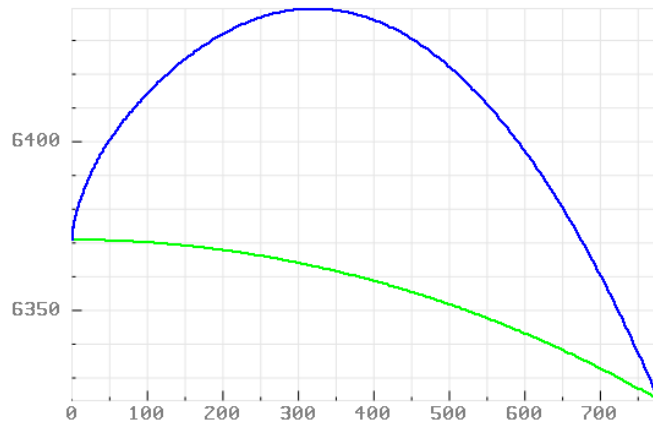
   b.
3. rocketTrajectoryFull.png
   a. this file show rocket trajectory in the view of full earth size.

   b.

Implementation

EngineDirector
+ constructRaptorEngine(IEngineBuilder)

FuelTankDirector
+ constructSmallFuelTank(FuelTankBuilder)

«use»

«use»

RocketComponent
- mass : double (kg)

IEngineBuilder
+ reset() : void
+ setThrust() : void
+ setMass() : void

IFuelTankBuilder
+ setFuelAmount() : void
+ setMaxEngineNumberSupport() : void
+ reset() : void
+ setMass() : void

EngineInfoProvider

EngineBuilder

Engine
- thrust : double
- currentPower : int

FuelTank
- fuelAmount : double (kg)
- maxEngineNumberSupport

FuelTankBuilder

FuelTankInfoProvider

RocketStage
- engines : Vector<Engine>
- fuelTank : FuelTank
- connections : Map<MountSide, *RocketStage>
- id : unsigned int

«use»

Enum MountSide
+ up = 0
+ right = 1
+ bottom = 2
+ left = 3

Rocket
- stages : Vector<RocketStage>
- angle : double
- velocityR : double
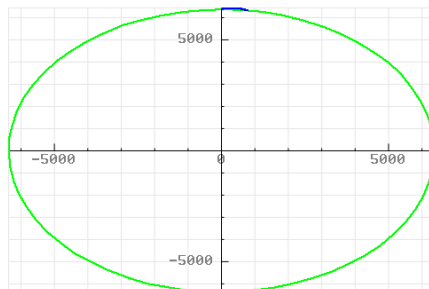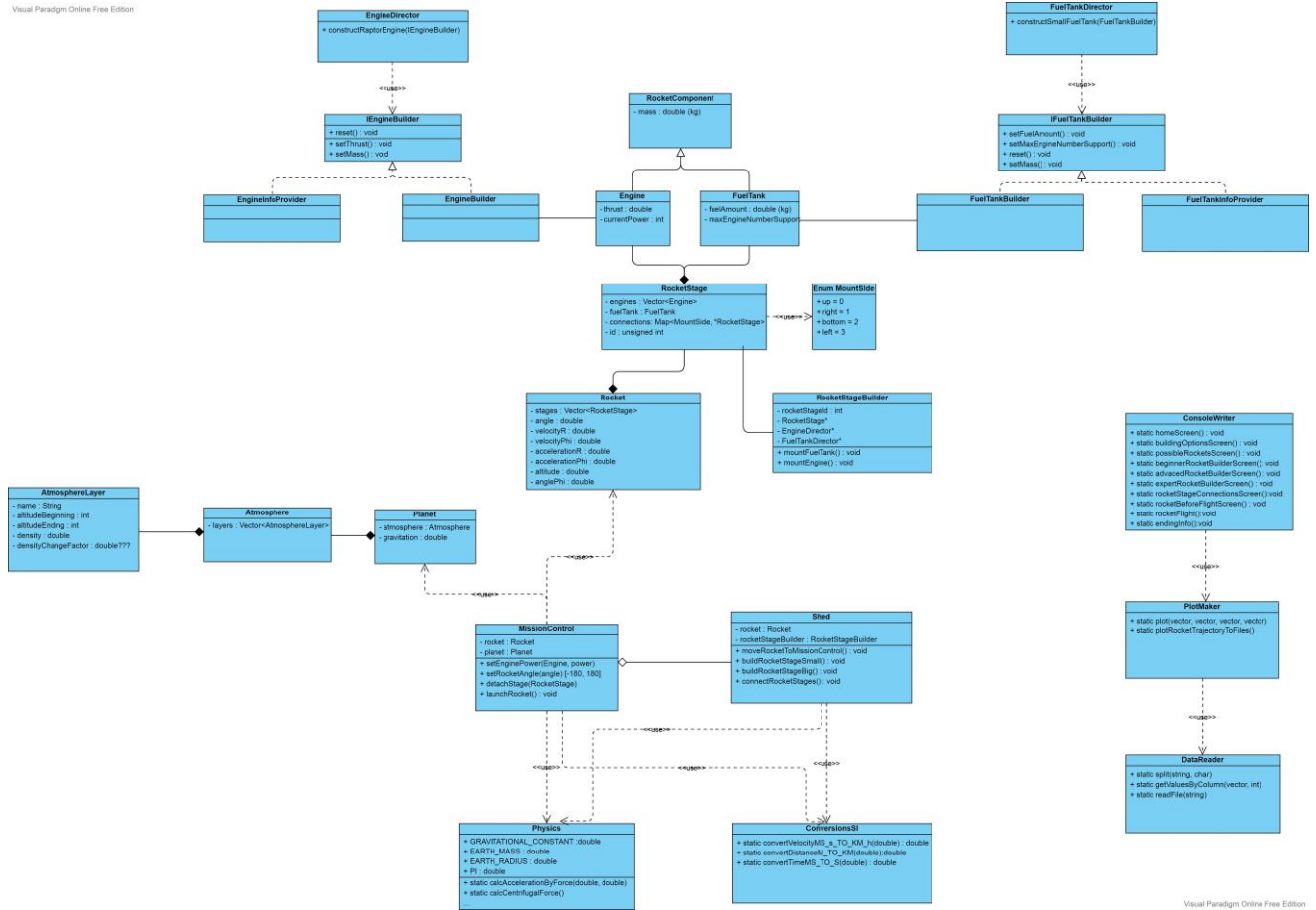- velocityPhi : double
- accelerationR : double
- accelerationPhi : double
- altitude : double
- anglePhi : double

RocketStageBuilder
- rocketStageId : int
- : RocketStage*
- EngineDirector*
- FuelTankDirector*
+ mountFuelTank() : void
+ mountEngine() : void

ConsoleWriter
+ static homeScreen() : void
+ static buildingOptionsScreen() : void
+ static possibleRocketsScreen() : void
+ static beginnerRocketBuilderScreen() : void
+ static advacedRocketBuilderScreen() : void
+ static expertRocketBuilderScreen() : void
+ static rocketStageConnectionsScreen():void
+ static rocketBeforeFlightScreen() : void
+ static rocketFlight():void
+ static endingInfo():void

AtmosphereLayer
- name : String
- altitudeBeginning : int
- altitudeEnding : int
- density : double
- densityChangeFactor : double???

Atmosphere
- layers : Vector<AtmosphereLayer>

Planet
- atmosphere : Atmosphere
- gravitation : double

«use»

«use»

MissionControl
- rocket : Rocket
- planet : Planet
+ setEnginePower(Engine, power)
+ setRocketAngle(angle) [-180, 180]
+ detachStage(RocketStage)
+ launchRocket() : void

Shed
- rocket : Rocket
- rocketStageBuilder : RocketStageBuilder
+ moveRocketToMissionControl() : void
+ buildRocketStageSmall() : void
+ buildRocketStageBig() : void
+ connectRocketStages() : void

PlotMaker
+ static plot(vector, vector, vector)
+ static plotRocketTrajectoryToFiles()

«use»

«use»

«use»

«use»

«use»

Physics
+ GRAVITATIONAL_CONSTANT :double
+ EARTH_MASS : double
+ EARTH_RADIUS : double
+ PI : double
+ static calcAccelerationByForce(double, double)
+ static calcCentrifugalForce()

ConversionsSI
+ static convertVelocityMS_s_TO_KM_h(double) : double
+ static convertDistanceM_TO_KM(double):double
+ static convertTimeMS_TO_S(double) : double

DataReader
+ static split(string, char)
+ static getValuesByColumn(vector, int)
+ static readFile(string)

src: /documentation/impl/UML.jpg