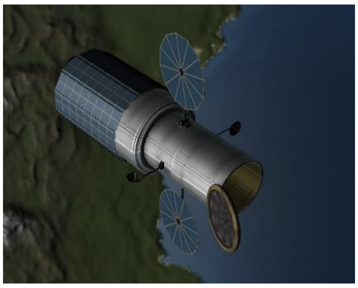
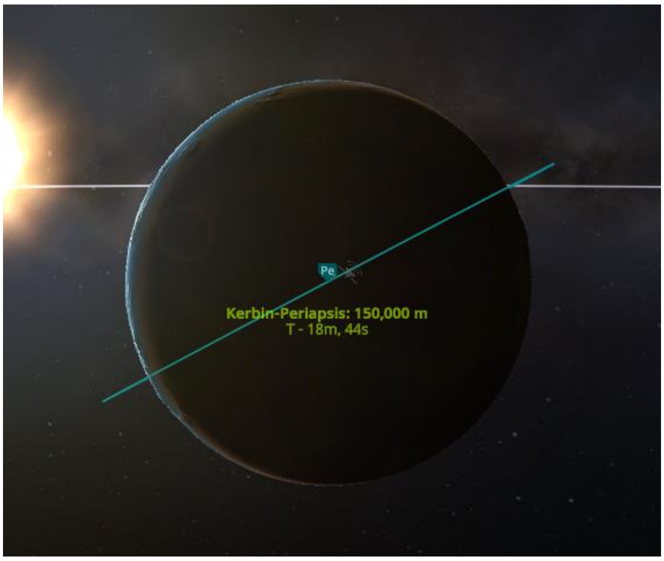
# Joolyter Demo Mission Briefings

## Mission One

Kerbal's new administration wants to become a respected space travelling civilization. To achieve that they ordered a refurbished space telescope on Omozan. Shipping delays caused it to arrive only one week before the launch so no time left to test the launch vehicle. To be safe though the engineers at Kerbal Space Center did manage to weld a tank and two powerful rocket engines to the telescope to be prepared if something goes wrong. And just like Jeb's great grandfather Edward A. Murphy-Kerman used to say "Everything that can go wrong, will go wrong", it does. Luckily KSC's engineers thought ahead but they still need your help to fulfill the mission objectives.






The launch vehicle was able to reach a circular orbit with a semi-major axis of 750 km above mean sea level with an inclination of 28.45°. At the start to perform a Hohmann-Transfer to reach the target orbit, one engine of the orbital maneuvering system failed. The telescope should be able to reach the target orbit on its own, if the transfer is performed as efficiently as possible. Unfortunately, the engineers just forgot to program the autopilot since they were busy calculating welds. Therefore, please calculate the change of velocity that is needed to perform the maneuvers. The target orbit has an altitude of 650 km and shares its orbital plane with the current one. Please choose the appropriate type of transfer and provide the change of velocity for each maneuver and in total.

### Constants and given values

$$
\begin{array}{lrl}
Radius\ of\ Kerbin & r_{Kerbin} = & 600\ km \\
Gravitational\ parameter\ (Kerbin) & \mu_{Kerbin} = & 3.5316 * 10^{12}\,\frac{km^3}{s^2} \\
Semi-major\ axis\ (parking\ orbit) & H_{park} = & 750\ km \\
Inclination & i = & 28.45° \\
Altitude\ (target\ orbit) & a_{target} = & 650\ km
\end{array}
$$

## Tips and hints

*Disclaimer: all centered green text blocks are not supposed to be used in your code. Those are mainly headers of functions or classes to make you familiar with the parameters and types. The green code blocks (with line numbering) could be executed and may mark a good starting point for you to add formulas and logic.*

To be able to accomplish the mission you are advised to read Joolyter's manual and complete all steps as the installation must be executed in the right order to work. As soon as that is done, the Joolyter Demo package may be imported by the following command:

```
import joolyterdemo
```

At the end of your script the method

```
joolyterdemo.Mission.one(
    dv12: float,
    dv23: float,
    unit: str = ‘m/s’
) -> bool
```

must be called.

`dv12` is the variable describing the change of velocity of the first impulse maneuver. `dv23` is the change of velocity of the second maneuver. All three variables are of the type float although Python does auto assignment and thus, also integers will work. `unit` is the unit of the provided values. It is an optional parameter and defaults to $\frac{m}{s}$ if you provide values in $\frac{km}{s}$ pass `‘km/s’`. Your naming of the variables can differ but to be able to call the method without further assignment you must keep the order.

In the following an example is shown with fictional values. The use of this code will cause a catastrophic failure of the mission, so handle with care.

```
import joolyterdemo

v12 = .5   # km/s
v23 = 2.   # km/s
v_tot = dv12 + dv23   # km/s
unit = ‘km/s’

if joolyterdemo.Mission.one(dv12, dv23, unit) == True:
   print(“dv12 =”, round(dv12,4), “km/s”)
   print(“dv23 =”, round(dv23,4), “km/s”)
   print(“dv_tot =”, round(dv_tot,4), “km/s”)
else:
   print(“Values used to call /‘mission_one/’ are incorrect!”)
```

For further assistance please see the joolyterdemo package's documentation first. When asking for support please always provide the file InstallationContainer/Kerbal_Space_Progam/KSP.log as it contains important information on what may cause the issue.

## Mission Two

After your tremendous support on the last mission the engineers have asked for your assistance again. A generic communication satellite is supposed to be placed in a synchronous orbit around Kerbin. Since we don't have a launch vehicle that can reach such an orbit yet, the satellite has been released at an inclined low Kerbin orbit. Fuel is very limited, so the transfer must be the most efficient. We trust in your abilities!

The satellite has been launched already into a low-Kerbin-orbit with a semi major axis of 750 km and inclination of 7°. The target orbit is a "Kerbisynchronous" orbit. Kerbin's sidereal rotation period is 5 h 59 min 7 s. Please calculate the semi-major axis of the target orbit first. Afterwards calculate the change of velocity needed for each of the following options, choose the most efficient and pass those values to the spacecraft:

Option 1:

Change of inclination in dedicated maneuver followed by a traditional Hohmann-transfer

Option 2:

Combined change of velocity and inclination in first maneuver and only change of velocity in second maneuver to circularize the orbit.

Option 3:

First maneuver lifts apogee to target altitude and in second maneuver a combined change of velocity and inclination leaves the vessel in target orbit.

### Constants and given values

$$\begin{aligned}
&Radius\ of\ Kerbin & r_{Kerbin} =\ & 600\ km \\
&Gravitational\ parameter\ (Kerbin) & \mu_{Kerbin} =\ & 3.5316 * 10^{12}\,\frac{km^3}{s^2} \\
&Siderial\ day (Kerbin) & d_{Kerbin} =\ & 5\ h\ 59\ \mathrm{min}\ 7s \\
&Semi - major\ axis\ (parking\ orbit) & H_{park} =\ & 750\ km \\
&Inclination\ (parking\ orbit) & i_{park} =\ & 7°
\end{aligned}$$

## Tips and hints

*Disclaimer: all centered green text blocks are not supposed to be used in your code. Those are mainly headers of functions or classes to make you familiar with the parameters and types. The green code blocks (with line numbering) could be executed and may mark a good starting point for you to add formulas and logic.*

As in the previous mission the Joolyter Demo package is imported by the following command:

```
import joolyterdemo
```

For this mission three new types are introduced. Each represents another variation of an impulse maneuver.

### *In-plane maneuver:*

This maneuver represents a change only in velocity and therefore, no change of the orbital plane.

```
joolyterdemo.VelocityChangeManeuver(
    dv: float,
    unit: str = ‘m/s’
)
```

### *Spatial maneuvers*

The next maneuver represents a change of the inclination of the orbital plane only.

```
joolyterdemo.InclinationChangeManeuver(
    dv: float,
    v: float,
    unit: str = ‘m/s’
)
```

The last maneuver represents a combined change of velocity and inclination.

```
joolyterdemo.CombinedManeuver(
    dv: float,
    v1: float,
    v2: float,
    unit: str = ‘m/s’
)
```

For information on the parameters, please consult the documentation. When asking for support please always provide the file InstallationContainer/Kerbal_Space_Progam/KSP.log as it contains important information on what may cause the issue.

An example of assigning a maneuver with fictional values follows:

```
import joolyterdemo

v1_1 = 1500  # m/s
v2_1 = 2000  # m/s
dv_1 = 800  # m/s

maneuver1 = joolyterdemo.CombinedManeuver(dv_1, v1_1, v2_1)

v_2 = 2000  # m/s
dv_2 = 500  # m/s
maneuver2 = joolyterdemo.InclinationChangeManeuver(dv_2, v_2)

dv_3 = 100  # m/s
maneuver3 = joolyterdemo.VelocityChangeManeuver(dv_3)
```

Afterwards the method that sends the maneuvers to the vessel can be called.

```
joolyterdemo.Mission.two(
    maneuver1: ImpulseManeuver,
    maneuver2: ImpulseManeuver,
    maneuver3: ImpulseManeuver = None,
) -> bool
```

Providing a third instance of a maneuver is optional since the most efficient combination of may require only two impulse maneuvers.

Continuing the example from above a call would look like this:

```
joolyterdemo.Mission.two(maneuver1, maneuver2, maneuver3)
```