

# ORBITER Scenario Editor

## Редактор сценариев для ORBITER

Copyright © 2006 Martin Schweiger

20 октября 2006

Orbiter web-сайт: [orbit.medphys.ucl.ac.uk/](http://orbit.medphys.ucl.ac.uk/) или [www.orbitersim.com](http://www.orbitersim.com)

Перевод - Юрий Кульчицкий ([www.kulch.spb.ru](http://www.kulch.spb.ru))



### Содержание

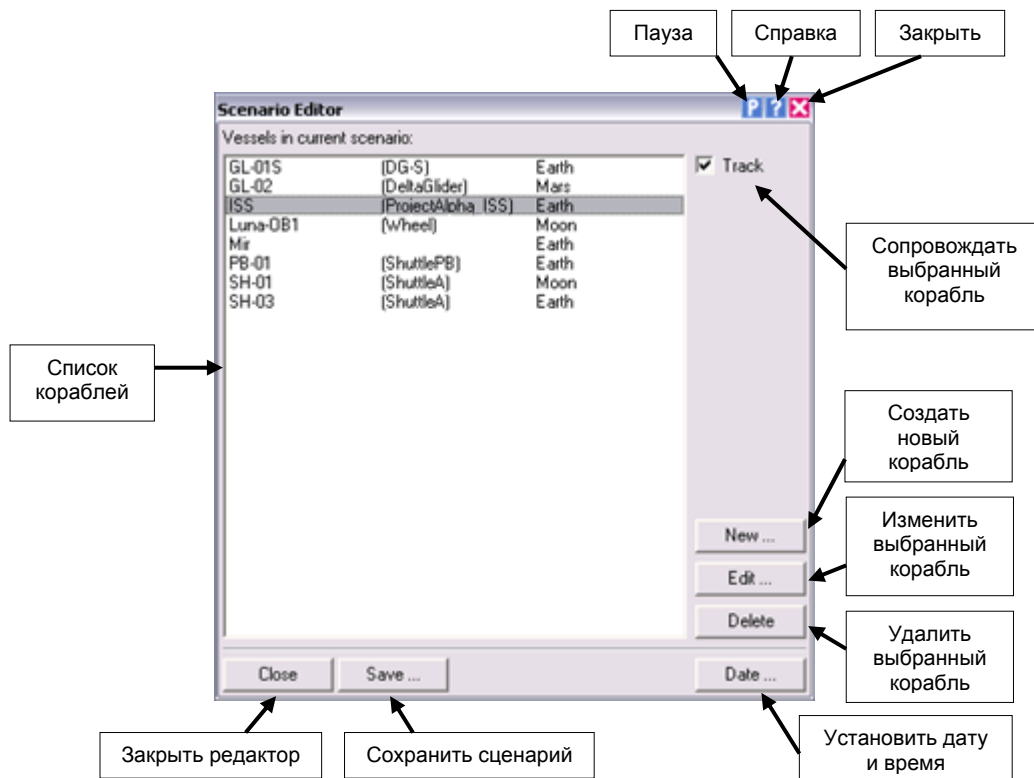
|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>ВСТУПЛЕНИЕ</b> .....                                     | <b>2</b>  |
| <b>2</b> | <b>ИСПОЛЬЗОВАНИЕ РЕДАКТОРА</b> .....                        | <b>3</b>  |
| 2.1      | Добавление нового корабля.....                              | 3         |
| 2.2      | Конфигурирование корабля.....                               | 3         |
| 2.3      | Элементы орбиты (Orbital elements).....                     | 4         |
| 2.4      | Векторы состояния корабля (State vectors).....              | 6         |
| 2.5      | Ориентация корабля (Orientation).....                       | 8         |
| 2.6      | Угловая скорость вращения корабля (Angular velocity).....   | 9         |
| 2.7      | Расположение корабля на поверхности планеты (Location)..... | 9         |
| 2.8      | Уровни топлива в баках корабля (Propellant).....            | 11        |
| 2.9      | Стыковка кораблей (Docking).....                            | 11        |
| 2.10     | Удаление корабля из симуляции.....                          | 12        |
| 2.11     | Изменение времени в симуляции.....                          | 13        |
| 2.12     | Сохранение сценария.....                                    | 14        |
| <b>3</b> | <b>РАЗДЕЛ ДЛЯ РАЗРАБОТЧИКОВ</b> .....                       | <b>16</b> |
| 3.1      | Конфигурационные файлы Vessel.....                          | 16        |
| 3.2      | Добавление в редактор vessel-специфических страниц.....     | 16        |

# 1 Вступление

Орбiter имеет специальный редактор, позволяющий создавать, конфигурировать и удалять корабли прямо во время симуляционной сессии. Для каждого корабля вы можете задать элементы орбиты (если он летит) или расположение на планете, параметры ориентации, указать количество топлива в баках, стыковочные соединения с другими кораблями, а также настроить различные параметры, специфические для данного типа корабля. Созданный сценарий можно сохранить в файл и использовать позднее или передать другим пользователям Орбitera.

Редактор представляет собой модуль-расширение (plugin module). Чтобы иметь возможность пользоваться редактором, убедитесь в том, что модуль *ScnEditor* активирован во вкладке *Modules* основного диалога Орбitera «Стартовая площадка» (*Launchpad dialog*) (подробнее см. раздел «Панель Modules» («Modules Tab») в Руководстве пользователя Орбitera).

Во время симуляции доступ к редактору сценариев осуществляется через диалог *Custom Functions*, который открывается нажатием клавиш **Ctrl** **F4**. В списке доступных расширений выберите пункт *Scenario Editor*. На экране откроется главная страница редактора. Страница содержит список кораблей, которые присутствуют в симуляции на данный момент. Каждый элемент списка содержит имя корабля, класс, к которому он относится, а также имя небесного тела, вокруг которого обращается или на поверхности которого находится корабль. Если включен флажок *Track*, камера Орбitera автоматически переключается на выбранный в списке корабль.



Теперь у вас появились следующие возможности:

- добавить в симуляцию новый корабль,
- конфигурировать выбранный корабль,
- удалить из симуляции выбранный корабль,
- задавать время в симуляции или
- сохранить полученное состояние в файл сценария.

## 2 Использование редактора

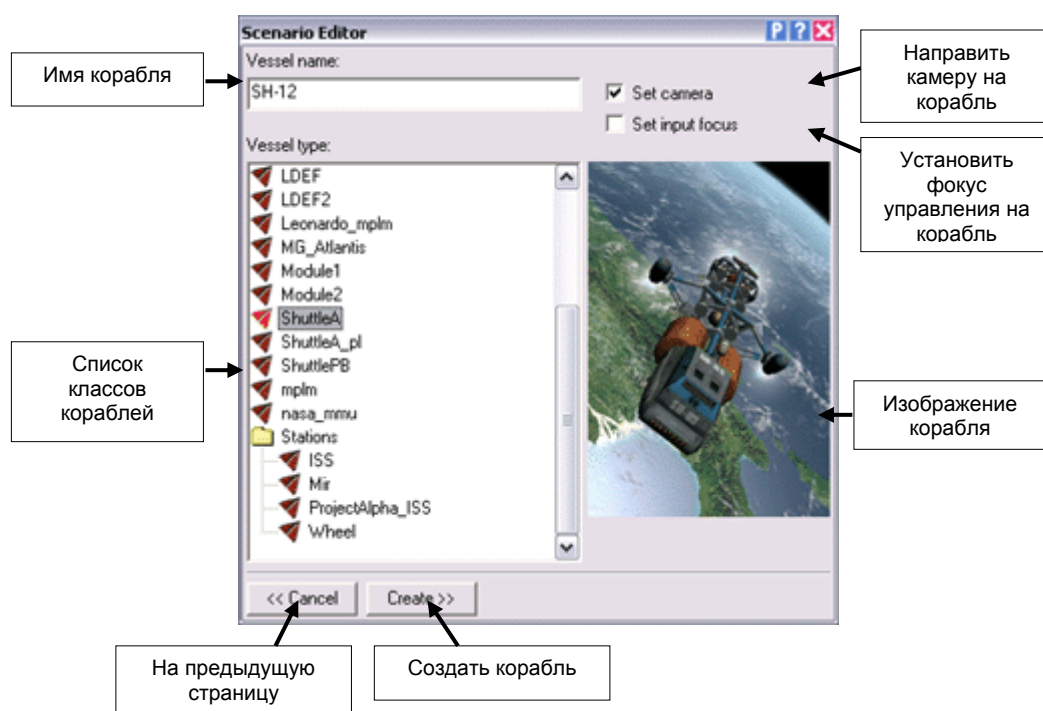
### 2.1 Добавление нового корабля

С нажатием на кнопку *New* открывается страница создания нового корабля. Здесь перечислены все доступные в Орбитере классы кораблей. Обратите внимание на то, что некоторые классы кораблей могут находиться в подпапках. Если вы хотите узнать, как выглядит корабль того или иного класса, щелкните мышью на соответствующем элементе списка – справа появится изображение корабля. (Возможно, не все классы судов имеют пример изображения). Выберите класс корабля и введите уникальное имя корабля в строке *Vessel name*.

Если вы хотите, чтобы камера симулятора была направлена на корабль, как только он будет создан, включите флажок *Set camera*.

Можно также включить флажок *Set input focus*, тогда фокус управления будет передан созданному кораблю, т.е. именно этим кораблем вы будете управлять с клавиатуры, мышью или джойстиком. Имейте в виду, что некоторые корабли могут не допускать передачу на них фокуса управления.

Чтобы создать корабль, нажмите кнопку *Create*. Все вновь созданные корабли по умолчанию помещаются на низкую околоземную орбиту.

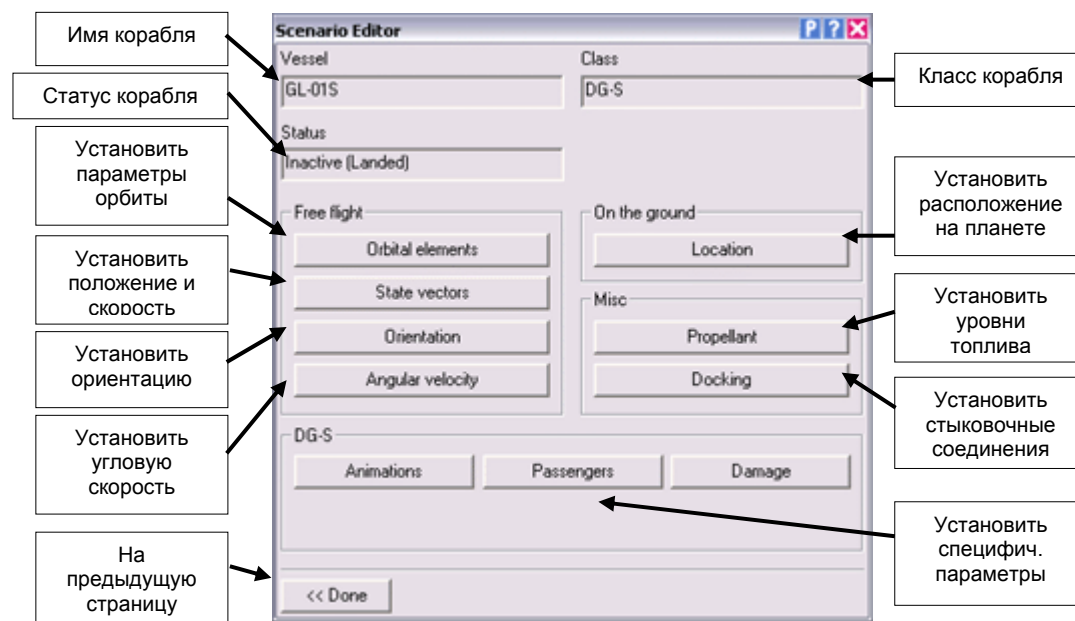


После создания корабля редактор переключается в режим *конфигурирования* корабля. Здесь можно задать позицию, ориентацию и другие параметры нового корабля.

### 2.2 Конфигурирование корабля

С нажатием на кнопку *Edit* в главном окне или после создания нового корабля, редактор переключается на страницу конфигурирования выбранного корабля. Отсюда вы получаете доступ к различным параметрам корабля, таким, как позиция в пространстве, скорость, ориентация. Также можно задать параметры стыковки корабля с другими кораблями и настроить параметры, специфические для кораблей данного класса.

**Примечание:** Хотя это требование не является строгим, все же рекомендуется включить паузу в Орбитере (нажмите **Ctrl P** в окне симулятора) перед тем, как настраивать параметры корабля. В противном случае, постоянное обновление симулятором параметров корабля (скорость, положение в пространстве) сильно усложнит точную настройку.



## 2.3 Элементы орбиты (Orbital elements)

Здесь настраиваются параметры, определяющие положение корабля на орбите вокруг небесного тела.

Сначала в списке *Orbit reference* выберите небесное тело (планету, луну или солнце), вокруг которого должен обращаться космический корабль.

Теперь в строке *Frame* выберите систему отсчета, относительно которой будут заданы параметры орбиты. Вы можете выбрать эклиптическую (*ecliptic*) или экваториальную (*ref. equator*) систему. Эклиптическая система определяется плоскостью земной орбиты и направлением на точку весеннего равноденствия (в эпоху J2000.0). Эта система отсчета удобна, если вам нужно расположить корабль на траектории межпланетного перелета. Экваториальная система отсчета определяется плоскостью экватора небесного тела, вокруг которого происходит обращение (т.е. выбранного в списке *Orbit reference*). Экваториальная система отсчета удобна, если нужно определить орбиту корабля относительно поверхности планеты, например, построить геостационарную или полярную орбиту.

Наконец, нужно установить эпоху, т.е. точку отсчета времени (*reference epoch*) для задаваемых элементов орбиты. На эту дату опирается средняя долгота (*mean longitude*). Если вы укажете значение «настоящий момент» (*current*), тогда значение средней долготы будет пониматься, как средняя долгота на текущий момент времени симулятора. В противном случае, значение средней долготы будет означать среднюю долготу на указанный момент Модифицированной Юлианской Даты (*MJD, Modified Julian Date*). Последний выбор более полезен при редактировании сценария без приостановки симуляции, потому что средняя долгота корабля непрерывно изменяется при движении по орбите.

Теперь все готово к определению элементов орбиты, сгруппированных в столбец в левой части страницы. Перечислим эти параметры.

### **Semi-major axis [SMa], большая полуось**

Для эллиптических орбит этот параметр является наибольшим полуэксцентриситетом эллипса орбиты. Для гиперболических орбит большая полуось является отрицательной величиной и означает расстояние от пересечения асимптот гиперболы до перицентра.

### **Eccentricity [Есс], эксцентриситет**

Определяет форму орбиты:

|             |                        |
|-------------|------------------------|
| $e = 0$     | круговая орбита        |
| $0 < e < 1$ | эллиптическая орбита   |
| $e = 1$     | параболическая орбита  |
| $e > 1$     | гиперболическая орбита |

### **Inclination [Inc], наклонение**

Определяется углом между плоскостью орбиты и плоскостью выбранной системы отсчета. Например, если вы выбрали экваториальную систему отсчета (*ref. equator*), а затем установили  $Inc = 0$ , это значит, что корабль будет двигаться по экваториальной орбите, т.е. по орбите, проходящей над экватором планеты. Наклонение  $Inc = 90$  означает движение по полярной орбите.

### **Longitude of ascending node [LAN], долгота восходящего узла**

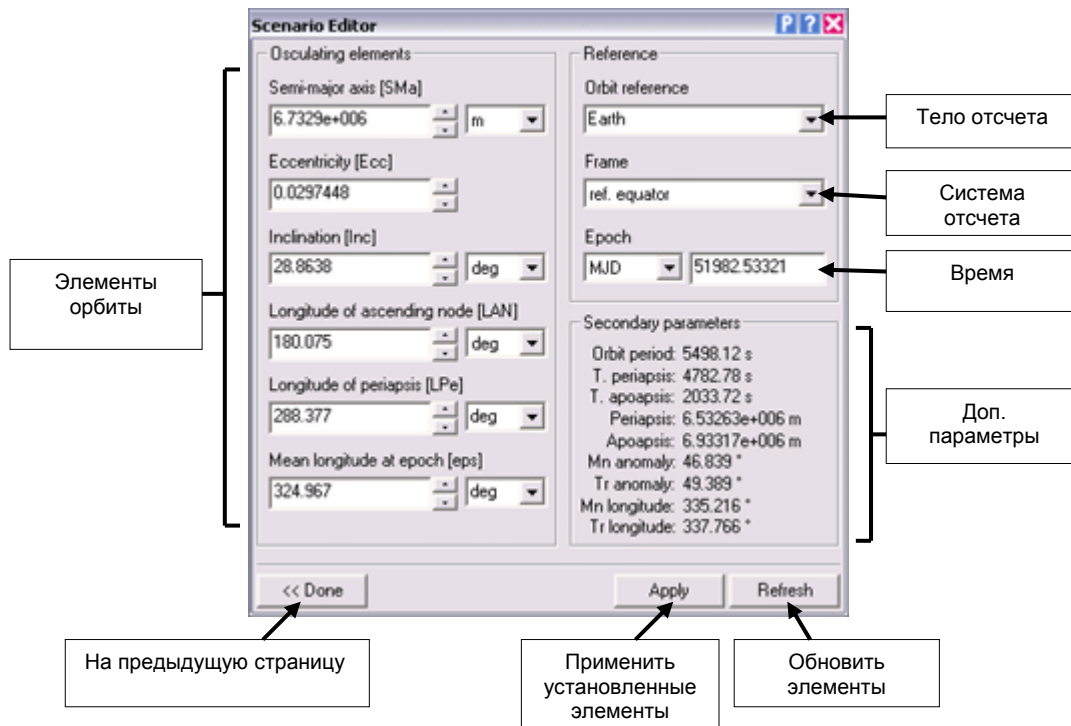
Угловое расстояние между направлением из фокуса орбиты на точку весеннего равноденствия и линией узлов орбиты (линия узлов – это линия образованная пересечением плоскости орбиты и плоскости системы отсчета).

### **Longitude of periapsis [LPe], долгота перицентра**

Это – сумма долготы восходящего узла LAN и аргумента перицентра (аргумент перицентра – это угол между направлением из фокуса орбиты на перицентр и линией узлов).

### **Mean longitude at epoch [eps], средняя долгота на момент времени**

Этот параметр определяет положение корабля на орбитальной траектории. Он определяется как сумма долготы восходящего узла LAN и средней аномалии (средняя аномалия – угол между перицентром и позицией корабля на воображаемой круговой орбите, имеющей тот же период обращения, что и настоящая орбита). Значение средней долготы привязано к конкретному моменту времени. Если момент времени (*Epoch*) – текущий (*current*), значение описывает текущую (непрерывно меняющуюся) среднюю долготу. В противном случае речь идет о средней долготе для указанной даты (и для данной орбиты).



## 2.4 Векторы состояния корабля (State vectors)

На этой странице настраиваются *позиция и скорость* корабля относительно небесного тела.

Прежде всего, в списке *Orbit reference* выберите небесное тело (солнце, планету или луну). Все векторы положения и скорости будут задаваться относительно центра выбранного небесного тела.

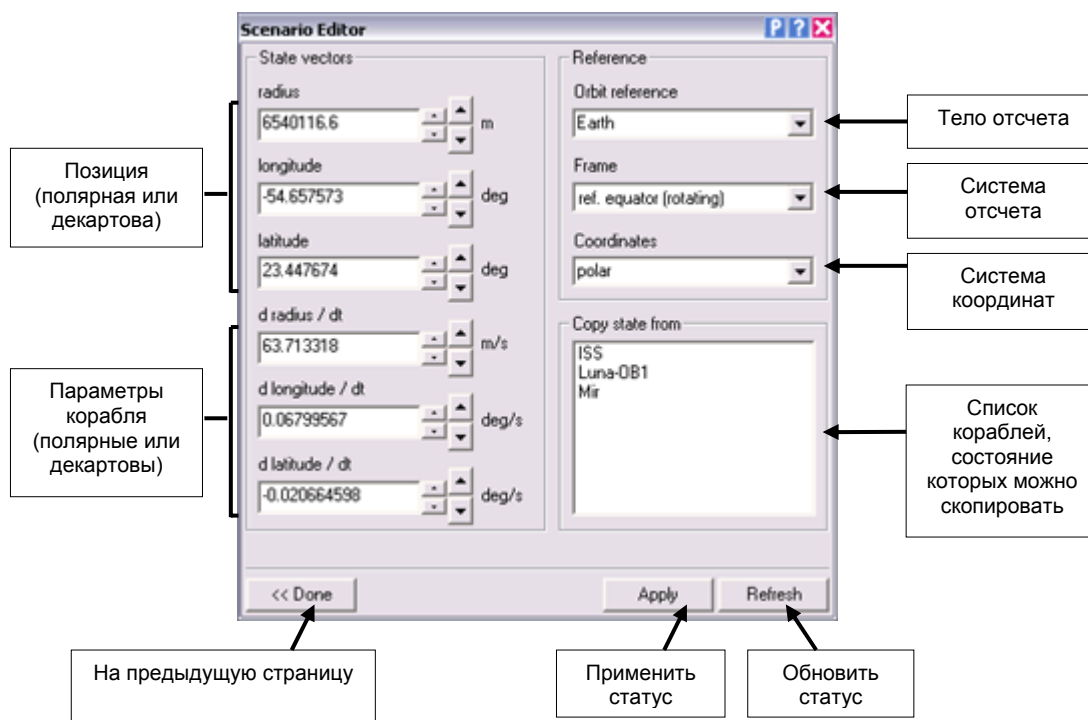
Далее, в списке *Frame* следует выбрать систему отсчета, в которой будут измеряться векторы положения и скорости. Доступные варианты:

**ecliptic, эклиптика:** Эклиптическая система отсчета определяется плоскостью орбиты Земли при обращении вокруг Солнца. Ось *x* определяется направлением на точку весеннего равноденствия, ось *y* – это направление на эклиптический северный полюс, ось *z* лежит в плоскости эклиптики, ортогональна осям *x* и *y*, и ориентирована таким образом, чтобы образовать левостороннюю систему координат.

**ref. equator (fixed), неподвижный экватор:** Система отсчета, определяемая экватором небесного тела. Оси *x* и *z* лежат в экваториальной плоскости, а ось *y* направлена на северный полюс. Система отсчета не вращается вместе с небесным телом.

**ref. equator (rotating), вращающийся экватор:** Тот же случай, только система отсчета вращается вместе с небесным телом, а ось *x* указывает на точку экватора с географической долготой 0°.

Наконец, нужно выбрать систему координат. Можно выбрать картезианскую (декартову) систему координат (*x*, *y*, *z* и  $dx/dt$ ,  $dy/dt$ ,  $dz/dt$ ) или сферическую полярную систему координат (*r*,  $\varphi$ ,  $\theta$  и  $dr/dt$ ,  $d\varphi/dt$ ,  $d\theta/dt$ ). Обычно, полярная система координат является интуитивно более понятной.



Теперь все готово для настройки векторов положения и скорости. Поскольку положение и скорость корабля непрерывно меняются, рекомендуется поставить симулятор на паузу, это можно сделать, нажав на кнопку *P* в заголовке диалога. После этого нажмите кнопку *Refresh*, для того, чтобы обновить на странице все параметры. Теперь вы можете ввести новые параметры и применить их к симуляции, нажав на кнопку *Apply* (или пользуясь кнопками прокрутки значений).

### Копирование состояния

Если вам нужно поместить корабль рядом с другим кораблем, вы можете просто скопировать его состояние. Для этого щелкните мышью по имени нужного корабля в списке *Copy state from*. (В списке перечислены только те корабли, которые находятся в полете).

Если после этого нажать на кнопку *Apply*, ваш корабль получит скопированное состояние. Имейте в виду, что если пауза выключена, нужный вам корабль за это время успеет улететь. Для того чтобы скопировать состояние другого корабля и сразу же назначить его вашему кораблю, сделайте двойной щелчок в списке.

После того, как ваш корабль окажется на месте другого, вы можете дополнительно отредактировать его вектора состояния для того, чтобы получить более аккуратное относительное положение.

### Приемы использования:

Редактирование векторов положения и скорости наиболее применимо, если корабль находится близко к поверхности планеты, например, летит в атмосфере. На орбитальных высотах и скоростях удобнее задавать положение корабля, редактируя *элементы его орбиты*. Установить стабильную орбиту, пользуясь редактированием векторов состояния намного труднее.

Вблизи поверхности планеты удобнее пользоваться системой отсчета, связанной с вращающимся экватором. В сочетании с полярной системой координат вектор положения напрямую даст широту и долготу местонахождения корабля и высоту его полета, а составляющие вектора скорости будут означать отсчет скорости относительно поверхности планеты в широтном, долготном и вертикальном направлениях.

Редактируя векторы, вы автоматически устанавливаете кораблю состояние «в полете». Убедитесь в том, что параметры, которые вы задаете, располагают корабль *над* поверхностью планеты. Если вам нужно расположить корабль на поверхности, пользуйтесь страницей настроек *Surface location*.

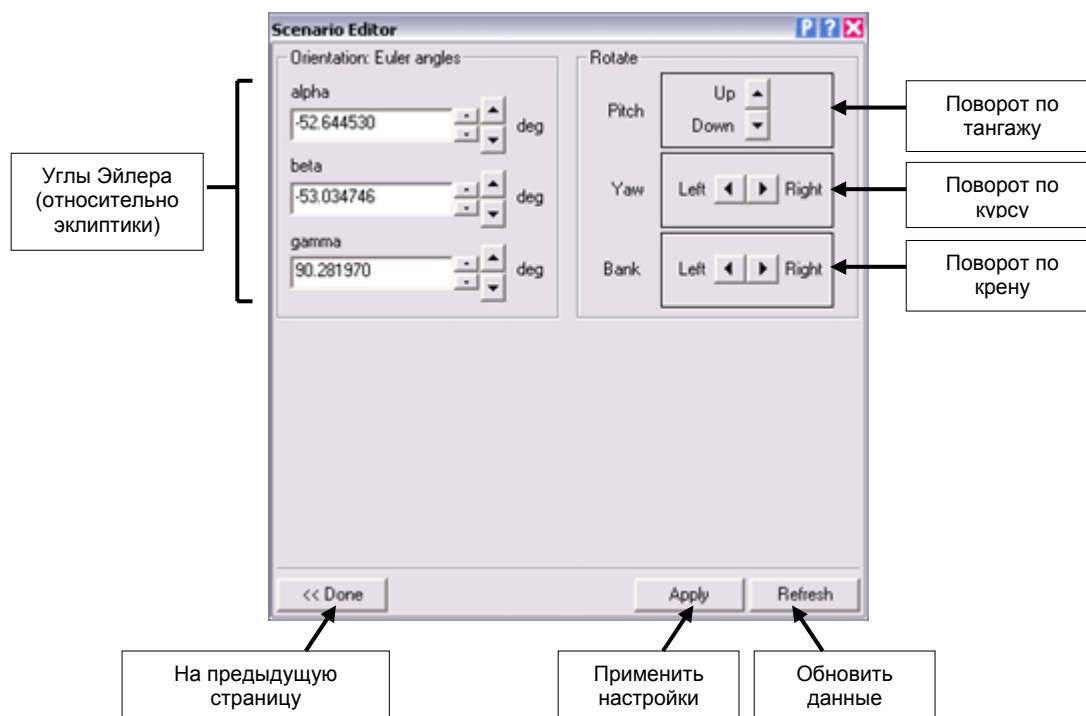
## 2.5 Ориентация корабля (Orientation)

На этой странице можно определить ориентацию корабля в свободном полете. Вы можете напрямую указать *Эйлеровы углы*, задающие ориентацию корабля относительно глобальной эклиптической системы координат, или можете *вращать* корабль вокруг его осей тангажа, курса и крена.

Эйлеровы углы  $\alpha$ ,  $\beta$  и  $\gamma$  определяют матрицу вращения  $\mathbf{R}$ , которая используется для трансформаций векторов из локальной системы координат корабля в глобальную эклиптическую систему:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Интуитивно более понятно вращение самого корабля вокруг осей его локальной системы координат. Это делается при помощи кнопок *Pitch*, *Yaw* и *Bank* соответственно для углов тангажа, курса и крена. Угол тангажа определяется вращением корабля вокруг оси X, угол курса определяется вращением корабля вокруг оси Y, угол крена – вращением вокруг оси Z (продольная ось корабля).



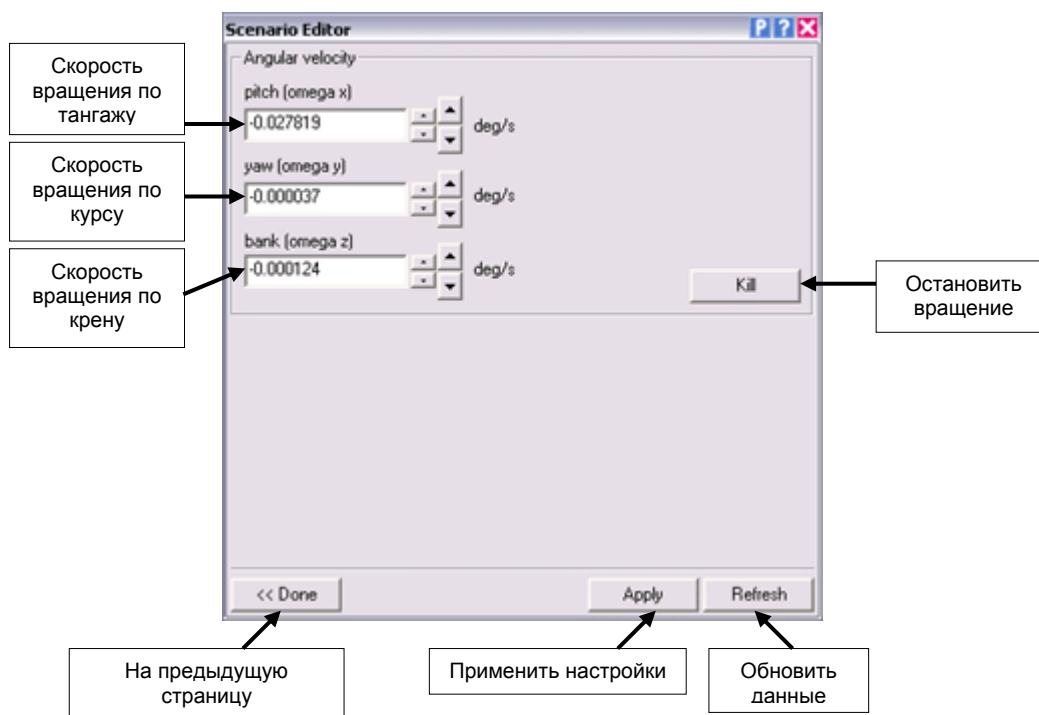
Если корабль является частью составной конструкции (например, пристыкован к орбитальной станции), вращаться будет вся конструкция.

Настройка углов ориентации не дает эффекта для кораблей, находящихся на поверхности планеты.



## 2.6 Угловая скорость вращения корабля (Angular velocity)

Здесь можно определить угловую скорость вращения корабля вокруг его координатных осей – оси x (тангаж), оси y (курс) и оси z (крен).



Чтобы полностью остановить вращение, нажмите кнопку *Kill*.

Обратите внимание на то, что значения компонентов угловой скорости могут колебаться между этими тремя осями в зависимости от распределения массы корабля, т.е. от его тензора моментов инерции (principal moments of inertia, PMI). Это означает, что, если последовательно нажимать кнопку *Refresh*, значения компонент угловой скорости могут меняться даже при отсутствии посторонних сил, могущих создать момент вращения.

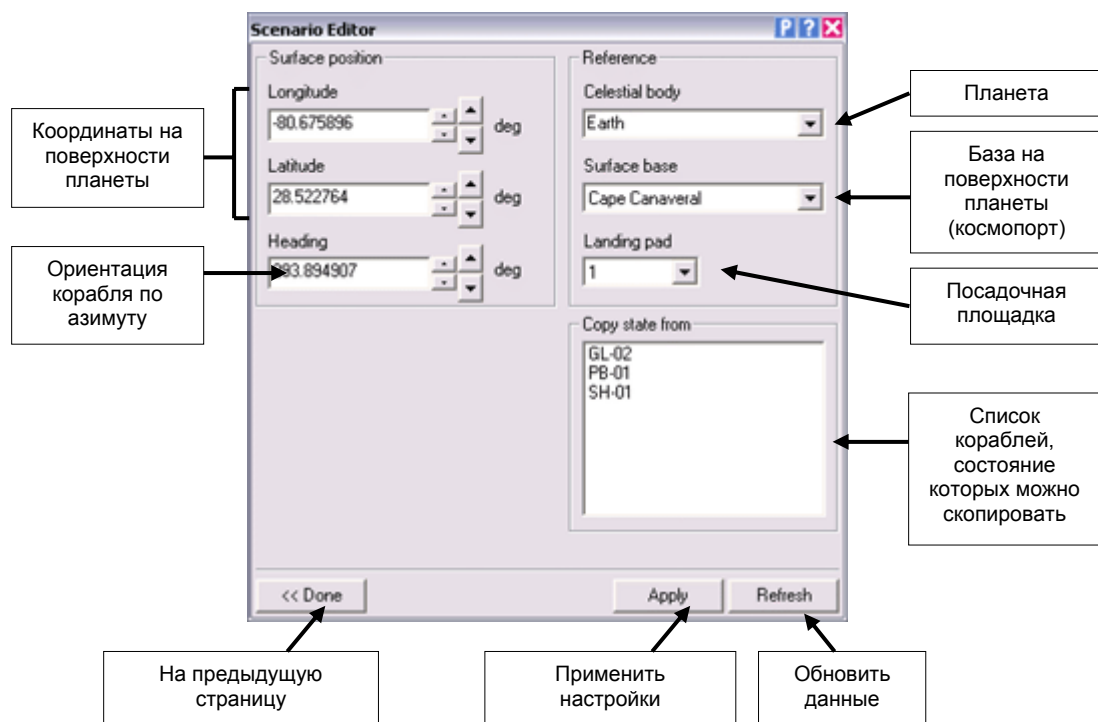
Аналогично, последовательное нажатие кнопки применения настроек (*Apply*) с одними и теми же значениями угловой скорости может вызвать резкое изменение вращения корабля.

Настройка скоростей вращения не дает эффекта для кораблей, находящихся на поверхности планеты.

## 2.7 Расположение корабля на поверхности планеты (Location)

На этой странице редактора вы можете настроить местонахождение и положение корабля на поверхности небесного тела.

Во-первых, в списке *Celestial body* нужно выбрать планету, на которой должен находиться корабль.



Если на планете есть космопорты, вы можете указать, в каком из них и на какой посадочной площадке следует расположить корабль. Космопорт выбирается в списке *Surface base*, а площадка – в списке *Landing pad*.

Кроме того, можно напрямую указать экваториальные координаты места, куда следует поместить корабль, в полях *Longitude* (долгота) и *Latitude* (широта).

В поле *Heading* можно указать ориентацию корабля по азимуту (его курс).

Менять значения широты и долготы можно при помощи кнопок прокручивания, расположенных справа от соответствующих полей. Маленькие кнопки меняют значения с маленьким шагом, большие – с большим шагом.

Если вы ввели цифровые значения напрямую, для обновления положения корабля нажмите кнопку *Apply*.

В настоящей версии Орбитера нельзя разместить на поверхности планеты конструкции, состоящие из состыкованных друг с другом кораблей. Поэтому не следует задавать наземное расположение для кораблей, состыкованных с другими кораблями.

### Копирование состояния

По аналогии со страницей настройки векторов состояния (*Status vector*), здесь также можно задать наземное положение своего корабля, скопировав его с другого корабля. В списке *Copy state from* перечислены все корабли, находящиеся на поверхности данной планеты. Щелчок мышью на выбранном в списке корабле сразу же устанавливает вашему кораблю такое же положение (независимо от того, включена ли пауза в симуляторе).

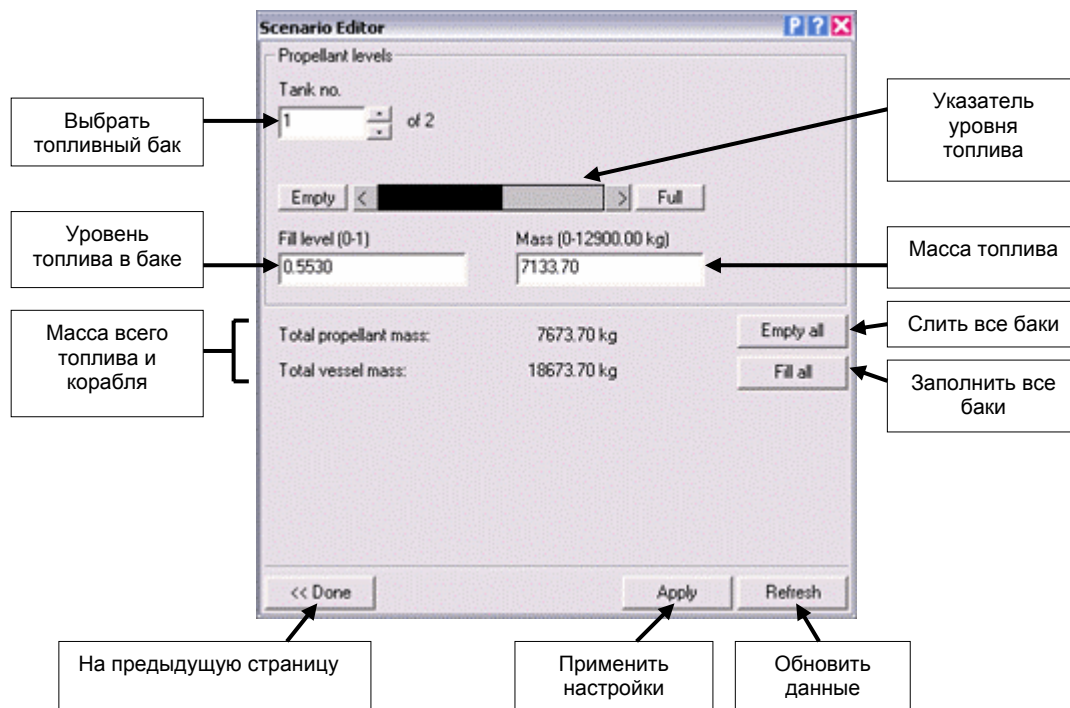
После того, как ваш корабль будет помещен в то место, которое уже занимает другой корабль, вы можете дополнительно отредактировать его координаты для того, чтобы получить более точное взаимное расположение кораблей.

## 2.8 Уровни топлива в баках корабля (Propellant)

На этой странице вы можете установить уровни топлива во всех баках корабля.

Выберите бак в поле *Tank no.* После этого вы сможете установить для этого бака уровень топлива. Можно воспользоваться линейным указателем (слайдером) или напрямую ввести уровень топлива (число от 0 до 1) или указать массу топлива в килограммах. Нажатием на кнопку *Apply* введенные данные применяются к кораблю.

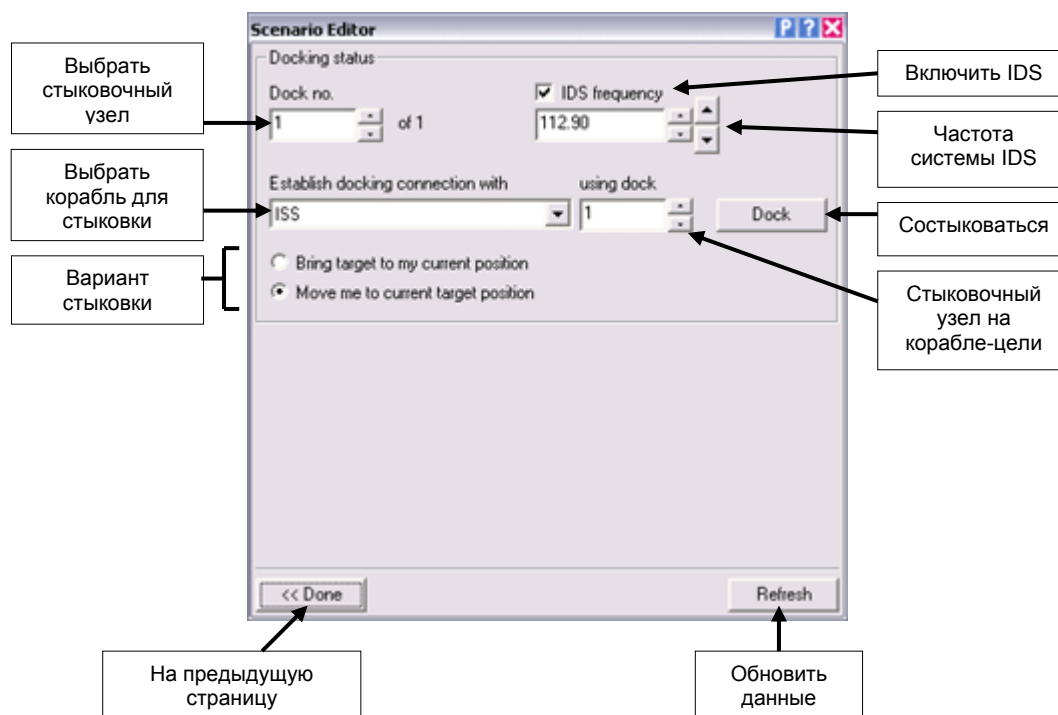
Полностью заправить или полностью опорожнить все баки можно, нажав на кнопки *Fill all* и *Empty all* соответственно.



## 2.9 Стыковка кораблей (Docking)

Если корабль оборудован стыковочным узлом, на этой странице редактора можно настроить все стыковочные соединения корабля с другими кораблями.

Во-первых, выберите стыковочный узел корабля. Для этого укажите номер стыковочного узла в поле *Dock no.* (слева вверху). Большинство кораблей имеют только один стыковочный узел, но некоторые – такие, как космические орбитальные станции – по несколько.



Далее, вы можете включить или не включить передатчик системы IDS (Instrument Docking System, система инструментальной стыковки) для данного стыковочного узла. Для этого используется флажок в правом верхнем углу страницы. Система IDS позволяет другим кораблям использовать их МФД Стыковки (Docking MFD) и ИЛС (HUD) для сближения с вашим кораблем в режиме радиосопровождения. Включив передатчик IDS, вы можете установить для него конкретную частоту в диапазоне от 108.00 до 139.95 МГц с шагом в 0.05 МГц. Убедитесь в том, что для разных стыковочных узлов вы установили различные частоты системы IDS!

Наконец, вы можете управлять стыковочными соединениями. Если с выбранным стыковочным узлом уже состыкован другой корабль, вы увидите его имя. Расстыковаться с кораблем можно, нажав кнопку *Undock*.

Если стыковочный узел не занят, вы можете выбрать корабль в предлагаемом списке и состыковаться с ним, нажав кнопку *Dock*. Если этот корабль имеет более одного незанятого стыковочного узла, вам потребуется указать его.

Вы можете также указать, как нужно переместить корабли при установлении стыковочного соединения. В варианте *Bring target to my current position* (переместить корабль-цель ко мне), корабль, с которым вы стыкуетесь, будет перенесен к вашему кораблю. Если вы выберете вариант *Move me to current target position* (переместить меня к кораблю-цели), тогда ваш корабль будет перенесен к кораблю, с которым вы стыкуетесь.

Имейте в виду, что состыковать корабли, находящиеся не в полете, а на поверхности планеты нельзя. По крайней мере, один из стыкуемых кораблей должен быть в полете. Стыковка двух приземленных кораблей или стыковка летящего корабля с приземленным с перемещением летящего корабля к приземленному приведет к непредсказуемому поведению этих кораблей в симуляции.

## 2.10 Удаление корабля из симуляции

Нажатие на кнопку *Delete* в главном диалоге редактора приведет к удалению выбранного корабля из списка. Фокус управления и камера автоматически переключатся на другой корабль, если это необходимо.

Нельзя удалить все корабли из симуляции: для правильной работы Орбитеру требуется хотя бы один корабль, которым можно управлять (имейте в виду, что некоторые корабли разработаны как неуправляемые, т.е. такие корабли не могут получить фокус управления).

## 2.11 Изменение времени в симуляции

Кнопка *Date* в главном диалоге открывает страницу редактирования даты и времени. Здесь вы можете изменить дату и время запущенной симуляции. При этом вы можете указать, как именно будут перенесены корабли из ранее заданного времени в новое время.

Дату и время можно задавать в нескольких форматах:

- **Universal Time (UT), Универсальное время:** Глобальное время, основанное на сидерическом времени. Соответствует местному времени на долготе 0°.
- **Julian Date (JD), Юлианская дата:** Интервал времени, выраженный в днях и долях дня, прошедший с момента полудня по Гринвичу 1 января 4713 года до Рождества Христова.
- **Modified Julian Date (MJD), Модифицированная Юлианская дата:** Юлианская дата минус 240 0000.5.
- **Julian Century (JC2000), Юлианское столетие:** Интервал времени, выраженный в столетиях и долях столетия, прошедший с момента полуночи по Гринвичу 1 января 2000 года.
- **Epoch, Эпоха:** Год и доля года.

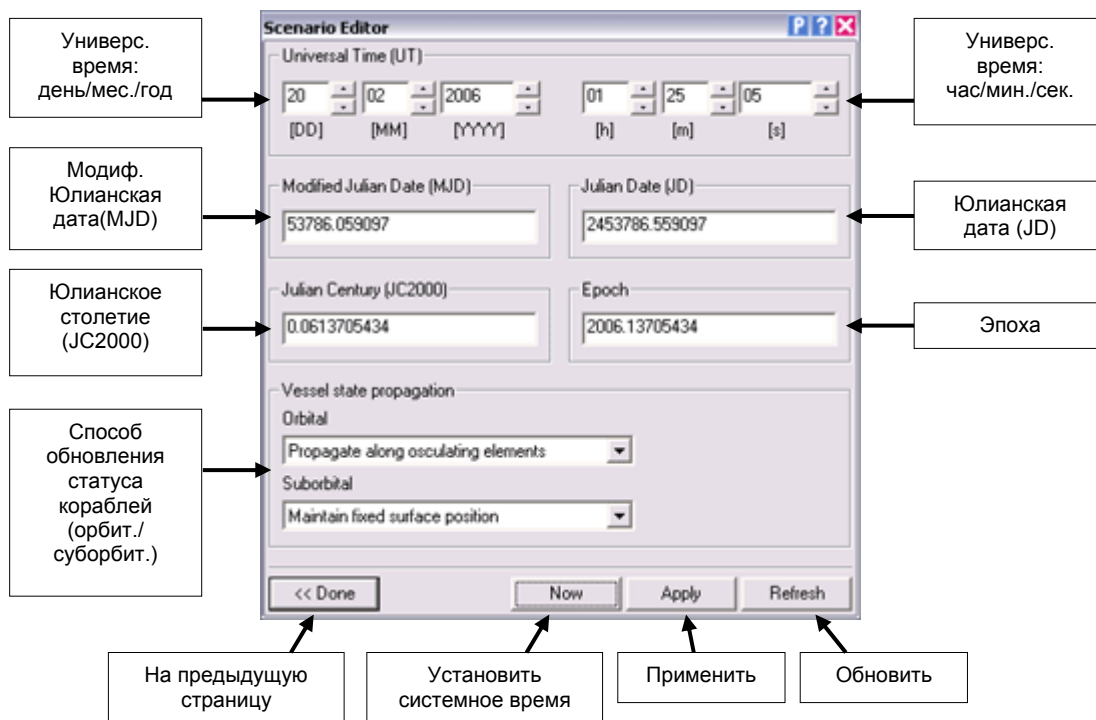
Если вы измените дату в любом из форматов, представление даты будет автоматически вычислено и во всех остальных форматах. Чтобы установить в диалоге текущее время симуляции, нажмите кнопку *Refresh*. Чтобы назначить введенное новое время симуляции, нажмите кнопку *Apply*. Нажатие на кнопку *Now* устанавливает в симуляции текущее системное время компьютера. Если вы меняете время кнопками прокрутки в секции *UT*, время в симуляции обновляется автоматически.

Перед тем, как установить новое время, вам следует определить, как это должно сказаться на существующих в симуляции кораблях. Предлагается несколько вариантов для орбитальных кораблей (т.е. кораблей, находящихся в орбитальном полете, чья траектория не пересекает поверхность центрального небесного тела) и для суборбитальных кораблей (т.е. кораблей, находящихся в суборбитальном полете и чья траектория упирается в поверхность планеты). Для орбитальных кораблей доступны следующие возможности:

- **Maintain fixed state vectors (Сохранить векторы состояния):** сохранить позицию, скорость и ориентацию корабля относительно центрального тела в невращающейся системе координат. Это означает, что корабль будет неподвижно зафиксирован относительно планеты, а сама планета провернется под ним согласно своему суточному вращению.
- **Maintain fixed surface position (Сохранить позицию над поверхностью планеты):** сохранить позицию, ориентацию и скорость корабля относительно поверхности планеты. Это означает, что корабль будет «провернут» вместе с планетой и его положение над ней не изменится.
- **Propagate along osculating elements (Пересчитать согласно оскулирующим элементам орбиты):** переместить корабль вдоль его орбитальной траектории, учитывая только гравитацию центрального небесного тела.

Для кораблей, находящихся в суборбитальном полете, к перечисленным возможностям добавляется еще одна:

- **Удалить корабли (Destroy vessels):** все суборбитальные корабли удаляются из симуляции (как если бы эти корабли разбились о планету в течение прошедшего времени).



Вы можете менять время, как в сторону будущего, так и в сторону прошлого. Но помните о том, что движение в прошлое не обязательно восстановит прошлое состояние симуляции, так как различные события, связанные с кораблями (например, работа двигателей), не учитываются.

## 2.12 Сохранение сценария

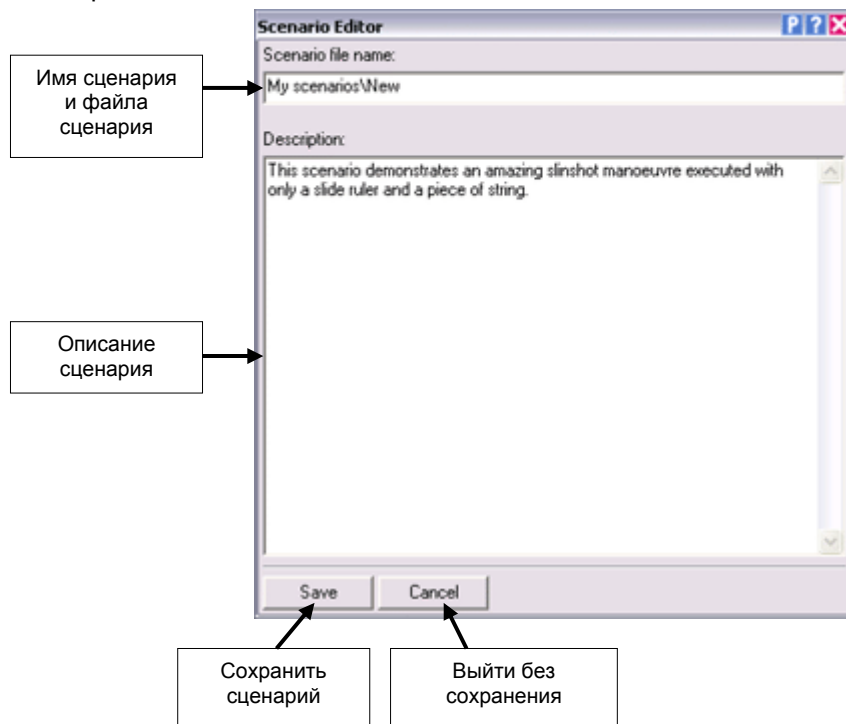
Если вы довольны созданным сценарием, вы можете сохранить его в файл сценария (.scn). Впоследствии вы сможете запустить созданный сценарий или передать его другим пользователям Орбитера.

Чтобы сохранить текущее состояние симуляции, нажмите кнопку *Save* в главном диалоге редактора. Откроется диалог *Save scenario*.

Введите имя файла и краткое описание сценария. Имя файла может содержать путь к папке. Путь к папке понимается как путь относительно корневой папки сценариев Орбитера (*Orbiter\Scenarios*). Указанная папка должна быть создана заранее.

Текст, введенный в поле *Description*, будет показан в стартовом диалоге Орбитера (*Launchpad dialog*) в тот момент, когда пользователь выберет ваш сценарий в списке доступных сценариев Орбитера. Предполагается, что в этом тексте содержится краткое пояснение к предлагаемому вами сценарию.

Нажмите кнопку *Save*, чтобы сохранить сценарий, или кнопку *Cancel*, чтобы выйти из редактора без сохранения.



## 3 Раздел для разработчиков

Настоящий раздел содержит информацию, полезную для тех разработчиков добавлений (addon'ов), которые хотят подготовить свои корабли для совместной работы с редактором сценариев.

### 3.1 Конфигурационные файлы Vessel

Редактор сценариев может работать с любыми кораблями, установленными пользователем в симуляторе. Единственное требование – конфигурационный файл корабля (.cfg) должен располагаться в каталоге *Config\Vessels*. Когда редактору требуется показать список доступных классов кораблей в диалоге *Vessel creation*, он сканирует этот каталог и добавляет в список корабли, для которых находит их конфигурационный файл.

Конфигурационный файл корабля может располагаться также и в подкаталоге каталога *Config\Vessels*, например *Config\Vessels\MyAddon\MyVessel.cfg*. Такие подкаталоги будут показаны в редакторе на странице создания корабля, и пользователь сможет зайти в них. Путь к подкаталогу относительно каталога *Config\Vessels* является частью имени класса корабля и соответствующим образом включается в описание класса корабля в сценарии.

Пример описания корабля в сценарии:

```
[...]  
BEGIN_SHIPS  
MyShip-01:MyAddon\MyVessel  
  [...]  
END  
[...]  
END_SHIPS
```

Редактируя сценарий вручную, не забывайте включать путь к конфигурационному файлу относительно каталога *Config\Vessels* (или, по старому, относительно каталога *Config*). В противном случае симулятор прервет загрузку сценария сообщением об ошибке.

Если вы не хотите, чтобы корабль определенного класса мог быть создан при помощи редактора сценариев, добавьте параметр

```
EditorCreate = FALSE
```

в конфигурационный файл этого класса. Такие конфигурационные файлы пропускаются Редактором при наполнении списка создаваемых кораблей.

**Примечание:** В старых версиях Орбитера конфигурационные файлы кораблей располагались в каталоге *Config*. Подобное расположение файлов по-прежнему поддерживается симулятором, но в редакторе сценариев соответствующие корабли не будут доступны. Поэтому расположение конфигурационного файла корабля в каталоге *Config* дает тот же эффект, что и значение параметра “EditorCreate = FALSE” в конфигурационном файле.

### 3.2 Добавление в редактор vessel-специфических страниц

Разработчики кораблей могут включать в свои проекты диалоги редактора, в которых пользователю дается возможность настроить параметры, специфические для этих кораблей. Это могут быть настройки анимации, управление полезным грузом, взлетной конфигурацией и т.п.



Чтобы создать в редакторе диалог для вашего корабля, вам понадобится скомпилировать DLL, в которой содержится шаблон диалога и код, обслуживающий диалоговые компоненты. Если ваш корабль уже имеет свою DLL, вы можете добавить необходимые элементы в нее. Альтернативный способ – создать отдельную DLL. Этот способ полезен, если ваш корабль управляется разборщиком скриптов, таким как, например, `spacescraft.dll`.

Чтобы сообщить Орбитеру, где он может найти расширения редактора, конфигурационный файл корабля (`.cfg`) должен содержать строку

```
EditorModule = <editor-module>
```

где `<editor-module>` – имя DLL, содержащей элементы редактора. Если вы используете для этого ту же DLL, что и для управления кораблем, значение `EditorModule` будет совпадать со значением `Module`. Если конфигурационный файл не содержит параметра `EditorModule`, редактор позволит задавать кораблю только общие для всех кораблей параметры.

DLL с расширениями редактора должна содержать callback-функцию `secInit` (`sec` расшифровывается как *scenario editor callback*) в следующем формате:

```
DLLCLBK void secInit (HWND hEditor, OBJHANDLE hVessel)
```

где `hEditor` – window handle редактора, а `hVessel` – handle созданного экземпляра вашего vessel.

Доступ к Vessel-специфическим опциям из редактора осуществляется через создаваемые дополнительные кнопки, которые помещаются в диалог *Vessel configuration* (см. раздел 2.2). Поддерживается до 6 добавочных кнопок. Создать дополнительную кнопку можно в теле функции `secInit`.

Есть два подхода:

#### Создание в редакторе дополнительного диалога

Чтобы создать кнопку, открывающую дополнительный диалог, добавьте в тело функции `secInit` следующий код:

```
EditorPageSpec eps = {  
    "Animations", hDLL, IDD_EDITOR_PG, EdPgProc};  
SendMessage (hEditor, WM_SCNEDITOR, SE_ADDPAGEBUTTON,  
    (LPARAM)&eps);
```

*EditorPageSpec* – это структура, содержащая параметры для новой функции редактора. Формат структуры таков:

```
typedef struct {  
    char btnlabel[32];  
    HINSTANCE hDLL;  
    WORD ResId;  
    DLGPROC TabProc;  
} EditorPageSpec;
```

где

|                       |  |
|-----------------------|--|
| <code>btnlabel</code> | надпись на создаваемой кнопке в конфигурационном диалоге редактора |
| <code>hDLL</code>     | handle на ваш модуль DLL   |
| <code>ResId</code>    | идентификатор ресурса, содержащего открываемый по кнопке диалог    |
| <code>TabProc</code>  | процедура dialog page window                                       |

*ResId* – идентификатор ресурса диалога, ресурс должен быть добавлен в DLL. Размер и компоновка диалога должны соответствовать стандартным диалогам редактора. Простейший путь – скопировать один из стандартных диалогов из файла ресурсов редактора (см. в исходники редактора – *Orbitersdk\samples\ScnEditor\ScnEditor.rc*), и затем переделать его.

*TabProc* должна быть реализована в модуле DLL как стандартная процедура Windows dialog message procedure, т.е. с интерфейсом

```
BOOL CALLBACK TabProc (HWND hWnd, UINT uMsg, WPARAM wParam,
    LPARAM lParam)
```

Эта функция получает все вводы пользователя как обычная процедура dialog message procedure. Модуль может использовать сообщения, посылаемые процедуре, чтобы установить или отобразить параметры корабля. Чтобы получить handle корабля внутри процедуры, используйте код

```
OBJHANDLE hVessel = (OBJHANDLE)lParam;
```

тогда, когда обрабатывается сообщение *WM\_INITDIALOG*, и код

```
OBJHANDLE hVessel;
SendMessage (hDlg, WM_SCNEDITOR, SE_GETVESSEL, (LPARAM)&hVessel);
```

тогда, когда обрабатываются любые другие сообщения.

### Создание callback-функции

В противоположность дополнительному встроенному диалогу, можно сделать так, чтобы редактор просто сообщал вашему модулю о том, что была нажата дополнительная кнопка. Это дает больше свободы для определения функциональности, которую вы хотите добавить. Например, вы можете открыть отдельный диалог или выполнить в своем модуле некоторые другие действия.

Чтобы создать дополнительную кнопку, вызывающую некоторую функцию в вашем модуле, добавьте в тело функции *seclnit* следующий код:

```
EditorFuncSpec efs = {
    "Payload", EditorFunc};
SendMessage (hEditor, WM_SCNEDITOR, SE_ADDFUNCBUTTON,
    (LPARAM)&efs);
```

*EditorFuncSpec* – структура, содержащая параметры для дополнительной кнопки. Формат структуры таков:

```
typedef struct {
    char btnlabel[32];
    CustomButtonFunc func;
} EditorFuncSpec;
```

где

*btnlabel*            надпись на создаваемой кнопке в конфигурационном диалоге редактора  
*func*                ссылка на callback-функцию в вашем модуле, которая должна вызваться при нажатии на кнопку.

Функция *CustomButtonFunc* должна быть определена в вашем модуле в следующем формате:

```
void EditorFunc (OBJHANDLE hVessel)
```

Функция получает в виде параметра handle на ваш vessel.

Примеры реализации vessel-специфических диалогов в редакторе сценариев можно найти в исходниках корабля *DeltaGlider*, которые поставляются в составе пакета Orbiter SDK (*Orbitersdk\samples\DeltaGlider*).