# **Editor Documentation**

|  | *Как?! Ты ещё не видел наш редактор?!*  (*Главный редактор*) |
| --- | --- |

# Введение

Что такое редактор? Старики утверждают, что это мощный инструмент по настройке данных, использующихся в игре. По сути, всё, что не программируется непосредственно в игровом коде, и не рисуется в Maya, настраивается из редактора. А это немало: компоновка карты, поведение героев, привязка эффектов, настройки игровой логики, рисование UI-форм и т.д.

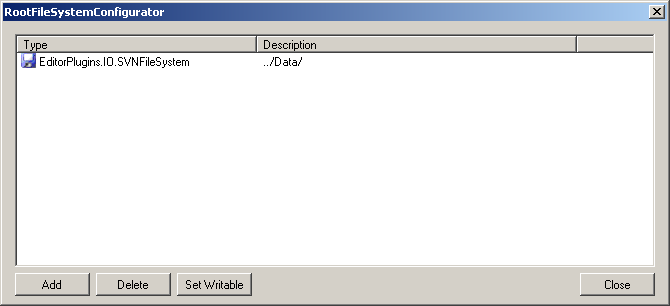
Редактор знает форматы исходных данных (в основном Maya), и умеет импортировать их в игровой формат, который затем грузится в игру. Редактор позволяет отображать игровые визуальные штуки очень близко к тому, как они будут видны в игре. При редактировании местами применяется WYSIWYG (например, редактирование карты).

Редактируемые данные принято называть ресурсами, они будут затем загружены в игру. Ресурсы - это граф связанных типизированных объектов. Ресурсы хранятся в XML-формате, со ссылками на бинарные файлы. Ресурсы лежат в папке **/Data**, образуя иерархию, которая доступна из редактора.

# **Getting Started**

## **Настройка файловой системы**

Предположим, мы запускаем редактор первый раз. Найдём **/Bin/PF\_Editor.exe** и запустим. Лучше сразу сделать необходимые настройки, и забыть про них по возможности надолго. Сначала надо указать редактору, где лежат данные, которые мы будем редактировать. В меню **Tools** нажимаем **File System Configuration** и видим такое окно:

  
**Рис. 1. Настройка файловых систем**

Одинокая строчка с иконкой дискетки в этом окне означает, что всё уже настроено - у нас уже есть файловая система, доступная для записи (Writable), и данные будут браться из соседнего каталога (**../Data/**). Очень рекомендуется пока на этом остановиться и ничего руками не трогать. Но для любопытных скажем следующее. EditorPlugins.IO.SVNFileSystem - это указание на то, что файловая система интегрирована с SVN и можно при сохранении данных тут же их и коммитить. Помимо SVN-файловых систем, можно добавлять обычные файловые системы (EditorPlugins.IO.WinFileSystem).

Зачем нужны несколько "файловых систем" в одном редакторе? Старики утверждают, что при этом они образуют составную файловую систему, когда ресурсные файлы из папок, указанных для каждой, используются совместно. Это означает, что несколько иерархий папок сливаются в одну иерархию по следующим правилам:

* Объекты из одинаковых папок будут соседствовать в редакторе, хотя на самом деле могут лежать далеко друг от друга;
* При совпадении имён файлов в одинаковых папках загружается последний по времени редактирования файл;
* Изменения в файлах сохраняются только в папки writable-файловой системы (той, у которой иконка дискетки).

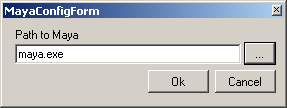
Не рекомендуется менять конфигурацию файловой системы посреди редактирования ресурсов. Получится непредсказуемый результат, который нам не нужен.

## **Указание пути до исходных данных**

Если вы дружите с Maya, то, скорее всего, вы захотите импортировать данные из майского формата в игровой формат. Чтобы это проходило как по нотам, надо указать редактору, где майские данные лежат. Делаем так: в меню **Tools** выбираем **Select Source Folder...** и указываем папку, где хранятся исходные данные. Иерархия папок там должна быть такой же, что и в каталоге Data, чтобы редактору (и всем остальным) было проще жить.

## **Настройка Майки**

В меню **Tools** выбираем **Maya.exe usage configuration** - и попадаем в окошко, где можно настроить использование Майки:

  
**Рис. 2. Настройка использования Майи**

Как нетрудно видеть, exe-файл Майки теперь может быть запущен только локально.

|  | **Maya**  Не знаете что такое Maya вообще? Ну и ладно. А вообще - вот: <http://ru.wikipedia.org/wiki/Maya>. |
| --- | --- |

На этом настройку редактора пока закончим.

# **Навигация, просмотр и редактирование объектов**

Для плодотворной работы в редакторе надо, чтобы на экране были открыты как минимум две основные панели. Ищем взором в открытом редакторе, и, если не находим, в меню **View** выбираем:

* **Object Browser**
* **Properties Editor**

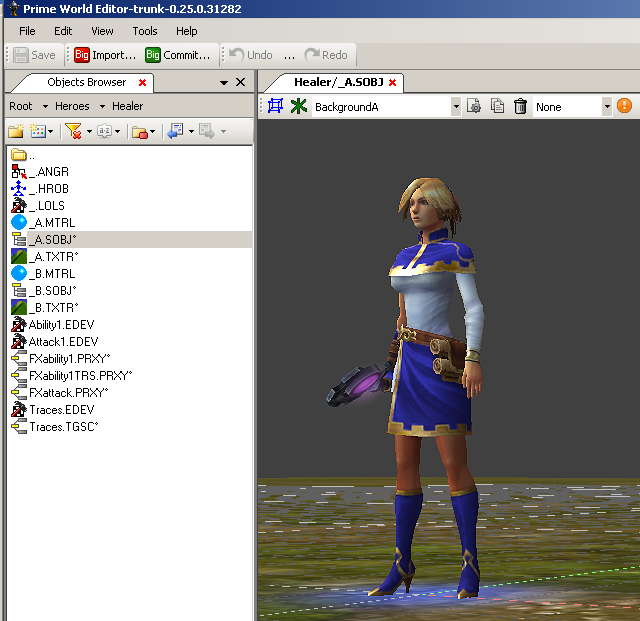
Работа с объектами в основном идёт в этих панелях.

|  | **Undo/Redo**  В редакторе большинство произведённых операций можно откатить назад (**Ctrl+Z**), а также повторить выполнение отменённой операции (**Ctrl+Y**). Старики говорят, что иногда работает. |
| --- | --- |

## **Object Browser. Иерархия папок**

В **Object Browser** есть всё необходимое для навигации по папкам и объектам в них, чем мы сейчас и займёмся.

Начнём, пожалуй, с красивых девушек. На рисунке ниже видно, что путь до девушки такой: **/Heroes/Healer/\_A.SOBJ**. Добравшись по иерархии до объекта, надо дабл-кликнуть по нему, или в popup-меню выбрать **Open**.

  
**Рис. 3. Object Browser и SceneObject**

Девушка начнёт жить своей непростой жизнью в панели **SceneObject**. Это значит, что мы открыли не просто героя, а целую сцену (кстати, ей соответствует файл \_A.SOBJ.xdb). Как профессионально закатывать сцены - см. [Просмотр сцены](https://confluence.nivalnetwork.com/display/prog/Editor+Documentation#EditorDocumentation-%D0%9F%D1%80%D0%BE%D1%81%D0%BC%D0%BE%D1%82%D1%80%D1%81%D1%86%D0%B5%D0%BD%D1%8B).

В панели **Properties Editor** можно будет смотреть, как устроена сцена, и даже её редактировать. Но сначала давайте разберёмся с навигацией.

Наверху панели **Object Browser** с помощью хитрого контрола показаны уровни иерархии. Они соответствуют папкам, и, нажав на их названия (в данном случае - Root, Heroes, Female), мы в них и попадаем. Этот же контрол позволяет увидеть вложенные папки на любом уровне иерархии - список выпадает в виде меню. А если кликнуть в этом контроле на свободное место от названий папок, он превратится в адресную строку, что очень удобно, когда Вы хотите кому-то передать путь до нужного объекта, или перейти по заданному в виде текста пути.

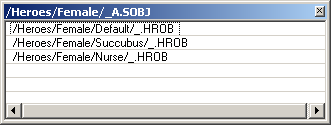
Ещё один способ ходить по папкам (самый привычный) - дабл-кликать непосредственно в панели по папкам или на уровень вверх.

Кнопками на тулбаре можно создавать новые папки, а также новые объекты, выбрав предварительно из списка тип. И ещё две кнопки в этой панели позволяют переименовать объект, а в крайнем случае и удалить вовсе.

Но и это ещё не всё. Если на объекте нажать на правую кнопку мыши, то в popup-меню мы увидим ещё кучу полезных действий, которые можно совершить над объектом, например:

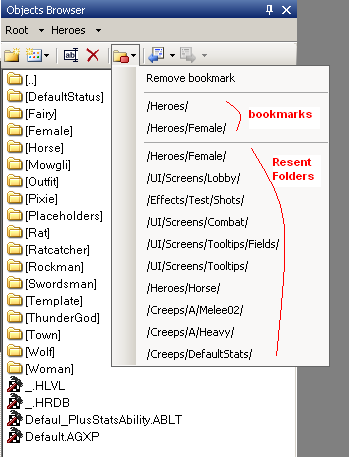
* **Import** (**Recursive Import**) - импорт данных из исходного формата (в т.ч. рекурсивно, т.е. по вложенным объектам)
* **Show backlinks** - показать список объектов, которые ссылаются на данный, т.е. обратные ссылки
* **Change Type** - изменить тип объекта, причём можно выбрать только подходящие типы, а не абы что
* **Copy** - копировать (для последующей вставки)
* **Cut** - вырезать (для последующего перемещения)

Пока более-менее интересным для навигации здесь является показ бэк-линков:

  
**Рис. 4. Backlinks**

Дабл-кликаем на строчке списка - и попадаем в соответствующий объект.

И, наконец, удобным средством быстрой навигации являются закладки (**Bookmarks**) и список последних открытых папок (**Recent Folders**) - они объединены в один выпадающий список:

​​  
**Рис. 5. Закладки и последние открытые папки**

Когда на текущую открытую папку уже есть закладка, верхний пункт (посмотрите на рисунок) предлагает его удалить. Если же закладки на текущую папку нет, верхний пункт предлагает его создать.

Две кнопки со стрелочками позволяют перемещаться по истории папок на манер броузера (**Back/Forward**).

Слева от кнопки закладок есть кнопка  "Reload folder content from disk", делающая именно это: перезагружающая с диска содержимое папки.

## **Properties Editor**

Хорош Object Browser, но основная работа в редакторе происходит в **Properties Editor**. Здесь можно редактировать любые поля объектов, спускаясь на какой угодно уровень вложенности объектов. Окно редактора свойств разделено на две части: **Outliner** и собственно панель редактирования.

### **Outliner**

**Outliner** - это древовидное представление текущего объекта. Удобно оно тем, что древесина свойств отображается с иконками, и можно, не вчитываясь особенно в названия свойств, быстро осуществлять навигацию по знакомым иконкам:

  
**Рис. 6. Properties Editor**

Также это представление позволяет не утонуть в простых полях - отображаются только узлы графа объектов. При этом панель редактирования содержит свойста объекта, выделенного в иерархии Outline.

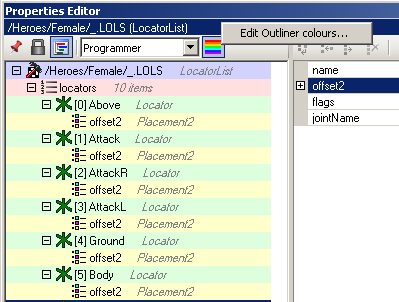
Теперь посмотрим внимательно на кнопки слева:

* **Pin Property Editor** - отпочковывает часть дерева объектов во главе с выделенным в Outline объектом в новое окно;
* **Lock current selection in Outliner** - фиксирует текущий вид Outline и не даёт ему меняться при выделении объектов в Object Browser;
* **Show/hide Outliner** - ну, понятно.

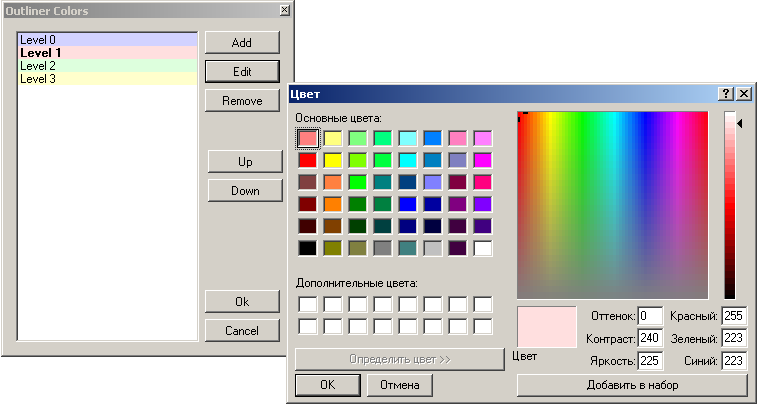
До кнопок справа мы обязательно дойдём, когда коснёмся редактирования списков.

#### **В цвете**

Outliner можно раскрасить. Для этого есть специально обученная цветная кнопка:

  
**Рис. . Outliner Colors**

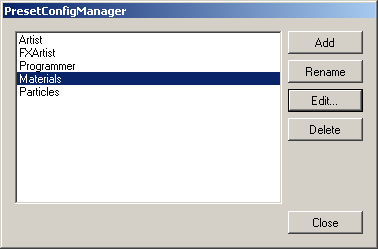
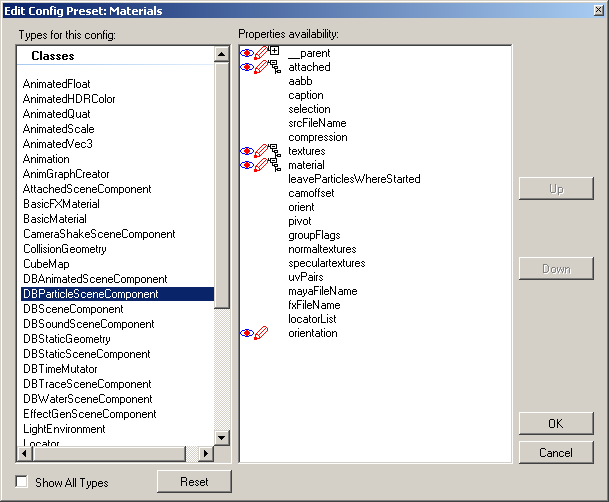
По этой кнопке каждому иерархическому уровню в Outliner показывается свой цвет. Цвета уровней настраиваются в таком окошке:

  
**Рис. . Настройка Outliner Colors**

Чтобы вызвать это окошко, надо правой кнопкой мыши нажать на цветной кнопке и выбрать единственный пункт popup-меню "Edit Outline Colors". Нетривиально, но экономно.

### **Редактирование пресетов свойств**

Выпадающий список в тулбарчике редактора свойств - установка текущего пресета свойств. В пресете указывается, в каких классах какие свойства показывать и в каком порядке. Это можно настроить в менеджере пресетов и редакторе пресета:

   
**Рис. 7. Менеджер пресетов и Редактор пресета**

Из этих картинок ясно почти всё, что нужно, а что не ясно - сейчас поясним. Для каждого поля любого класса можно настраивать его видимость (**красный глазик**), редактируемость (**карандашик**), и способ отображения в дереве (для сложных типов). Кнопками **Up** и **Down** можно упорядочивать поля как душе угодно. Кнопка **Reset** делает страшную вещь - сбрасывает установленные пользователем настройки для выделенного класса.

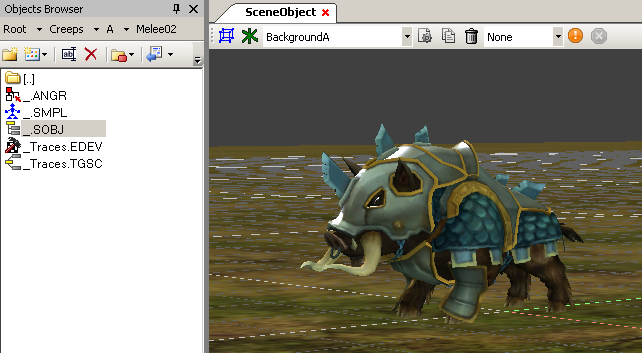
Это же окно можно вызвать из popup-меню в панели редактирования свойств (там появляются пункты наподобие **Configure DBSceneComponent**, т.е. предлагается законфигурить текущий класс). При этом редактируется всё в рамках текущего пресета.

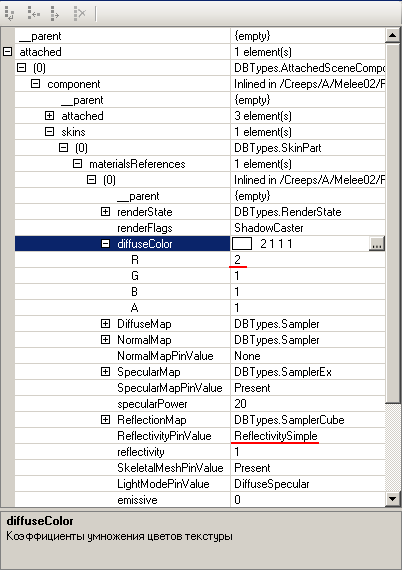
Пресет **Programmers** - это полный набор полей, здесь видно всё, в том числе и те поля, которые менять из редактора бесполезно или даже вредно. А, например, пресет **Materials** позволяет сделать упор на классах и полях, связанных с "материалами", скрывая всё прочее.   
Да, ещё. Нерусское слово **пресет** означает именованный набор настроек. Как видите, ничего страшного.

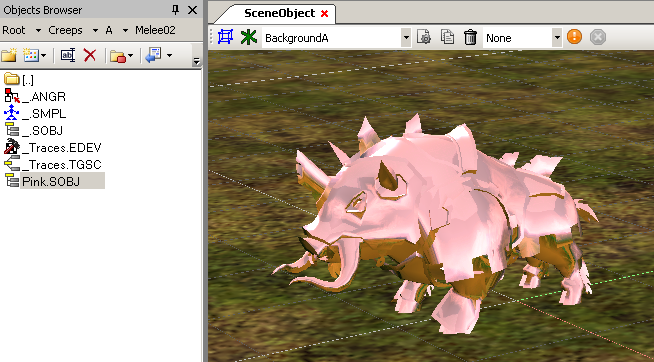
### **Пример редактирования простых свойств**

Давайте попробуем сделать некоторое законченное действие по редактированию объектов. Для этого нам понадобится новый объект, которого не жалко будет редактировать.

**1.** Найдём в Object Browser объект **/Creeps/A/Melee02/\_.SOBJ**, дабл-кликнем по нему (это объект сцены - боевая хрюшка пехоты, как видно из просмотра объекта сцены)  
**2.** Сделаем ему копию (в popup-меню - **Copy** и **Paste**), покажем созданный объект (дабл-клик в Object Browser).  
**3.** Отредактируем ему пару свойств во вложенном объекте материала (см. рисунки ниже):   
 **а)** сделаем множитель красного цвета **diffuseColor.R = 2**, так что хрюшка приобретёт здоровый розовый окрас,   
 **б)** установим переключатель отражения **ReflectivityPinValue** в значение **ReflectivitySimple**, выбрав его из списка, так что наша хрюшка засияет как начищенный пятак.  
**4.** Назовём нашу хрюшку **Pink.SOBJ**

**  
Рис. ... Исходный объект**

**  
Рис. ... Меняем значения двух свойств материала...**

**  
Рис. ... В результате хрюшка приобрела здоровый розовый окрас и засияла начищенной медью**

Эта хрюшка дальше нам ещё пригодится для многих славных дел, но пока посмотрим более внимательно на поля, которые мы отредактировали.

Поле множителя красного цвета - это числовое поле в составе структуры **diffuseColor**. Поле R (а также G, B, A) может принимать и дробные значения - редактор одинаково успешно примет точку и запятую в качестве разделителя.

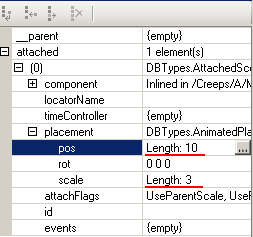
Поле **ReflectivityPinValue** - это перечисление. Т.е. его значение выбирается из списка.

Кстати, для редактирования полей материалов удобно использовать пресет **Materials**.

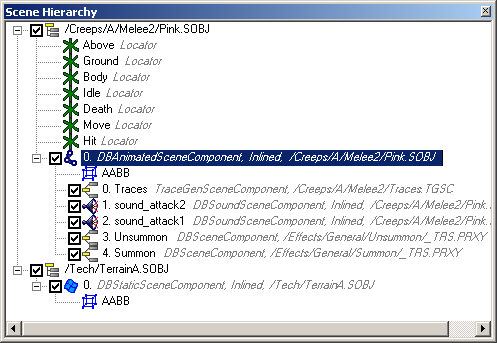
### **Редактирование анимированных типов**

**Редактирование анимированных свойств**

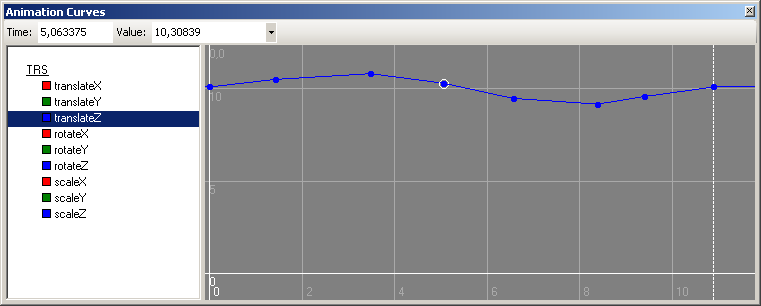
Теперь попробуем поиграться с анимированными свойствами - сейчас с их помощью мы научим нашу розовую хрюшку летать.

  
**Рис. ... Анимированные поля.**

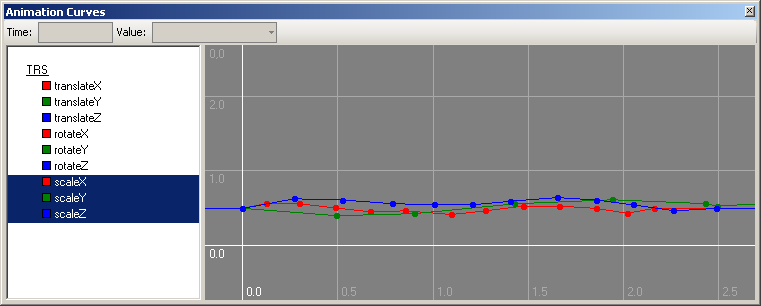
Поработаем с полем **placement.pos**. Это - анимированный трёхмерный вектор (x, y, z). Для каждой координаты можно задать график его изменения от времени. Мы ограничимся высотой (z): повесим хрюшку на 10 метров, и пусть её немного поболтает на ветру.



Для этого откроем окно **Scene Hierarchy** (на рис. выше), выделим в нём компонент **DBAnimatedSceneComponent**, и откроем окно **Animation Curves** (меню **View**). Для редактирования анимированной структуры надо оставить выделенной только одну координату **translateZ**, и поставить мышкой несколько точек на нужной нам высоте:

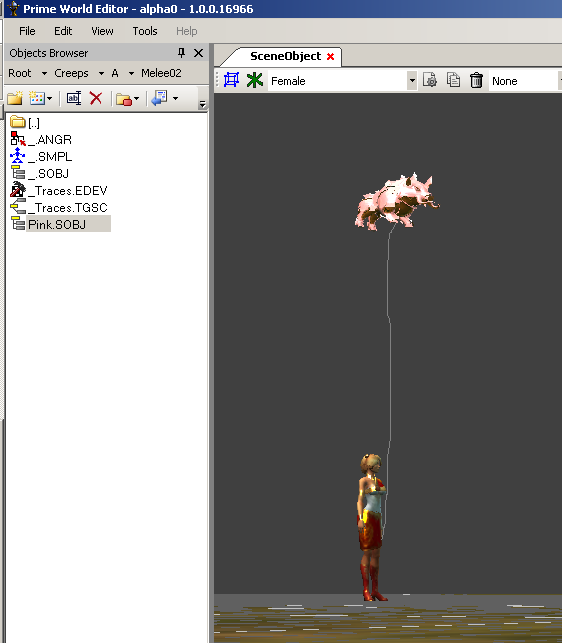
  
**Рис. ... Редактирование анимированного поля pos.**

Теперь пусть хрюшка будет деформироваться по всем трём координатам (а для этого нам надо поредактировать анимированное свойство масштаба **placement.scale**). Установим здесь тип анимации тоже в Cycle, длительность анимации в 3 секунды, и, выделяя поочерёдно поля структуры, нарисуем точками графики их изменения:

  
**Рис. ... Редактирование анимированного поля scale.**

|  | **Как зациклить анимированные свойства**  Наша хрюшка оживает только на один цикл, и потом опять ведёт себя очень холодно. Чтобы анимированные свойства зацикливались, надо создать **TimeController** у компонента **DBAnimatedSceneComponent**.  Это делается так: в **Properties Editor** правой кнопкой мыши кликаем на свойстве **timeController** и выбираем **Create Inlined -> TimeController**. В созданном тайм-контроллере обнуляем свойство **cycleCount**, и теперь наши анимированные значения будут изменяться в цикле, пока не кончится электричество. |
| --- | --- |

Ну, вот теперь, помимо того, что наше замечательное летающее животное в соответствии со своей анимированной сущностью дышит, её ещё и периодически плющит. Теперь всё готово для мирной демонстрации с "воздушным шариком":

  
**Рис. ... Мир, труд, май.**

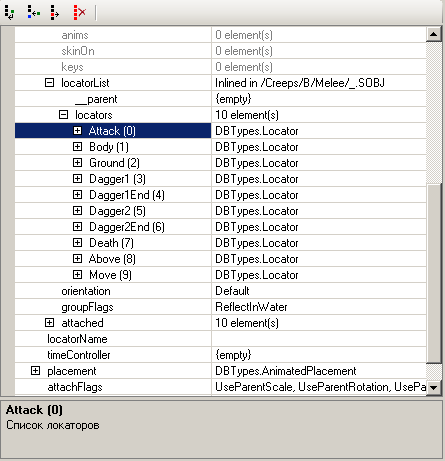
(Здесь при просмотре сцены использован пресет **Female**. Верёвочка предоставлена **MS Paint**, а идея сцены - группой **Pink Floyd**).

|  | **Ещё несколько советов при редактировании анимированных параметров:**   * Удалять точки значений можно нажатием правой кнопкой мыши. * Можно выделить несколько точек и редактировать их всем скопом (таскать по экрану, * **Ctrl+A** выделяет все точки, и масштабирует так, чтобы они все были видны (при нажатии на **A** делается то же самое, но без выделения точек) * При нажатии на **F** выделенная точка отображается "в фокусе" - в центре экрана. Если выделить несколько точек и нажать **F**, график будет смасштабирован для того, чтобы выделенные точки были видны и занимали максимальное пространство окна. * Работает колесо мыши и масштабирует окно * В левой панели окна можно выделять сколько угодно переменных, они все будут показываться в области графиков. Применяем **Ctrl** и **Shift** при выделении переменных. |
| --- | --- |

### **Редактирование списков**

**Работа со списками объектов**

Среди типов полей объектов есть тип "список". Такие поля знают, списки каких объектов они могут хранить. Работа со списками объектов идёт в Properties Editor с помощью кнопок в правой части тулбара:

  
**Рис. ... Пример списка (список локаторов).**

Кнопки (слева направо) делают следующее:

* Добавление нового элемента в конец списка
* Вставка нового элемента перед текущим
* Удаление текущего элемента
* Очистка списка

Когда в список вставляется новый элемент, он представляет из себя объект с полями, заполненными значениями по-умолчанию.

На рисунке показан список локаторов (у **.SOBJ** он находится, например, здесь: **attached(0).component.locatorList.locators**). Давайте добавим новый локатор:

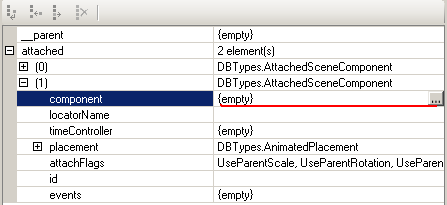
1) Встаём на корень списка (поле **locators**);  
2) Жмём на первую кнопку - в конце списка появится новый элемент (в скобочках будет его порядковый номер);  
3) Переходим на этот объект и редактируем его поля, например: **name = Ground2**, **offset2.translateZ = 2** и **offset2.translateY = 2**.

Теперь, открыв наш объект в просмотре сцены и показав его локаторы, можно увидеть, что появился новый локатор, расположенный на высоте 2 м и в 2 м сзади от объекта.

### **Ссылки на объекты. Inlined**

Ссылочные поля содержат ссылки на другие объекты. Существует два типа ссылок: inlined и внешняя. **Inlined** означает вложение одного объекта в другой. Вложенный объект является неотъемлемой частью родителького объекта и не может быть объектом других ссылок. Внешняя же ссылка позволяет ссылаться на один объект из нескольких мест.

Попробуем создать inlined-объект. Перейдём в любой **.SOBJ** и добавим новый элемент в поле attached. При этом создастся новый объект типа **AttachedSceneComponent**. В нём нас интересует поле **component**, которое как раз является ссылкой:

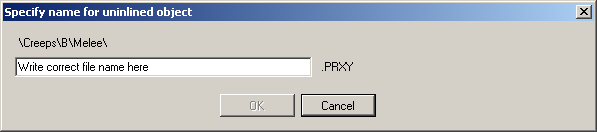
  
**Рис. ... Сomponent - поле-ссылка**

Сейчас эта ссылка пустая. Сделаем inlined-объект. Для этого надо дабл-кликнуть по названию поля component, и появится выпадающий список классов (также в контекстном меню можно выбрать пункт **Create inlined**). Нам надо выбрать, какого класса будет создаваемый объект. Выбираем **DBSceneComponent**. Теперь в значении поля появилось **"Inlined in /Creeps/B/Melee/\_.SOBJ"**, и мы теперь можем редактировать этот вложенный объект.

Второй способ заполнить пустую ссылку - указать уже существующий объект. Для этого надо кликнуть на кнопку  и появится окно выбора объекта:

  
**Рис. ... Окно выбора объекта.**

Объект, который подошёл бы под эту ссылку, должен иметь расширение .PRXY (он должен быть наследником DBSceneComponent). Но у нас в показанной папке нет таких объектов. Его, конечно, можно создать прямо из окна выбора объекта (а это окно - клон Object Browser). Но мы сделаем иначе. Вместо того, чтобы рыскать в поисках подходящего объекта, или создавать его, давайте вынесем свежесозданный inlined-объект из нашего поля. В контекстном меню (по правой кнопке мыши) выбираем пункт **Uninline**. Появится формочка задания названия нового объекта:

  
**Рис. ... Окно задания имени объекта в операции Uninline.**

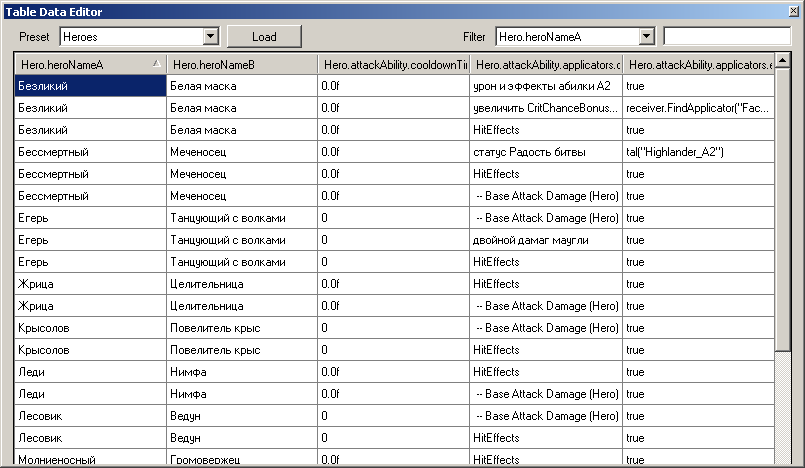
После задания имени объекта и нажатия OK мы увидим через Object Browser, что новенький объект появился под боком у родительского объекта, и на него теперь можно ставить ссылки отовсюду и столько раз, сколько влезет.

Можно сделать и обратную операцию - внешний объект, на который есть ссылка, засунуть в поле как inlined: в контекстном меню для этого есть пункт **Inline**. Инлайнить объект можно только в том случае, если данная ссылка на него - единственная. При инлайне внешний самостоятельный объект удаляется.

Чтобы удалить ссылку на объект, либо удалить inlined-объект, надо просто очистить значение поля ссылки. Клавиша **Delete** forever!

## **Табличное редактирование объектов**

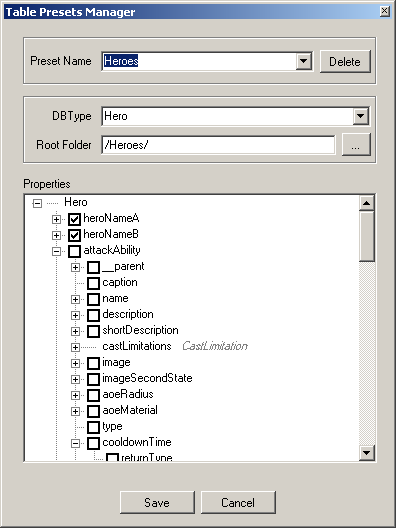
Режим табличного редактирования удобен для массовых изменений однотипных объектов. В меню **View** выбираем **Table Data Editor**, загружаем пресет, и в таблице выводятся значения выбранных полей, которые тут же можно редактировать:



Но сначала посмотрим, как настраиваются пресеты.

### **Настройка пресетов**

В выпадающем списке **Preset** выбираем пункт **Manage...**. Откроется такое вот окно:



Разберём, что там есть и как это всё настраивается. Пресеты именуются в верхнем комбобоксе, из него же можно выбирать настраиваемый пресет. Такая комбинация функций приводит к тому, что переименовать созданный пресет уже нельзя (ввод имени, которого нет среди пресетов, приводит к заведению нового пресета).

Поле **DBType** позволяет выбирать класс объектов, которые мы хотим видеть в таблице. При этом покажутся не только объекты выбранного класса, но и классов-наследников.

Поле **Root Folder** задаёт, в какой папке искать объекты (включая вложенные, конечно).

|  | Очень рекомендуется не оставлять в этом поле одинокий слэш, т.к. это очень сильно замедляет выборку объектов. Старайтесь локализовать (в хорошем смысле) объекты одинаковых типов в определённых папках, если это возможно. |
| --- | --- |

В дерево **Properties** мы настраиваем, какие поля будут отображаться в виде столбцов в таблице. При этом показываются не только поля объектов выбранного класса (верхний уровень), но и поля связанных объектов.

Пресеты сохраняются локально в файле **TableDataPresets.config**.

Итак, мы настроили пресет, сохранили его, выбираем его из списка, и жмём кнопку **Load**. Что можно делать в загруженной табличке?

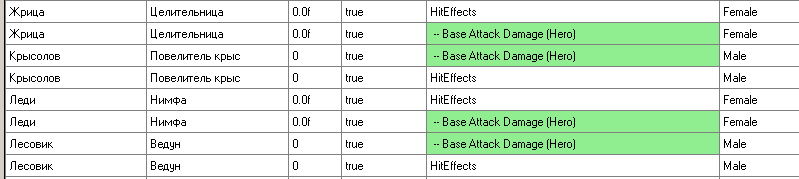
### **Редактирование**

Поля, которые можно редактировать, выводятся обычным образом - как ячейки. Редактированные ячейки помечаются зелёным бэкграундом:

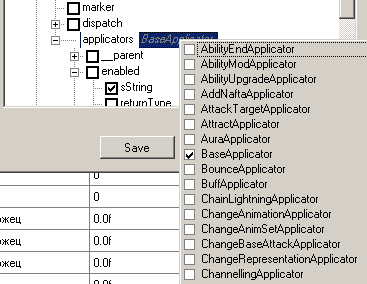


Почему при редактировании одной ячейки пометилось зелёным сразу три? Дело в том, что в случае нескольких вложенных объектов у одного поля-коллекции (на любом уровне иерархии) кол-во строк будет соответствовать количеству вложенных объектов. Так дерево объектов показывается в таблице: поля родительского объекта неизбежно будут дублироваться в своих значениях. В нашем примере мы показываем названия аппликаторов атакующей абилки героев, а в общем случае аппликаторов в абилке несколько.

Второй аналогичный случай редактирования сразу нескольких ячеек таблицы одним движением - это когда несколько объектов ссылаются на один и тот же объект. Например, некоторые герои используют один и тот же аппликатор **-- Base Attack Damage (Hero)**, а мы решили его (аппликатор) зачем-то переименовать, прямо не сходя с этого места:



На примере тех же аппликаторов рассмотрим одну фичу выбора полей в менеджере пресетов. Классов аппликаторов великое множество, и в каждом - свои наборы полей. Чтобы выводить поля нужных нам аппликаторов, нужно отметить явно дочерние классы:



Это делается правой кнопкой мыши на нашем поле *applicators* - в выпадающем списке видны все наследники класса **BaseApplicator**. При отмечании этих классов в дереве появятся их поля, и тогда мы сможем их редактировать из таблицы.

Как и положено в приличном обществе, работают команды **Undo/Redo**.

### **Фильтрация и сортировка**

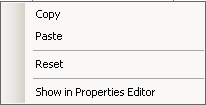
Два поля справа вверху позволяют фильтровать содержимое таблицы: выбираем столбец в выпадающем списке **Filter**, и вводим то, что ищем, в соседнее текстовое поле. Поиск идёт по вхождению подстроки.

Сортировка идёт по клику на столбец.

Ещё можно таскать столбцы, меняя их местами.

### **Дополнительные ништяки**

Ништяки живут в popup-меню:



Копирование и вставка - это стандартные операции с буфером обмена. При этом можно копировать и вставлять сразу фрагмент таблицы, и при этом это всё работает вместе с **Excel** - можно вставить скопированное в **Excel**, там поредактировать, и вставить обратно. **Ctrl+A** выделяет всё, и если в таком виде скопировать, то в Excel вставится таблица с названиями столбцов.

**Reset** - устанавливает значение выбранных ячеек в первоначально значение, которое было при загрузке таблицы.

**Show in Properties Editor** - открывает **Properties Editor** с соответствующим объектом. Редактирование поля в таблице отражается в **Properties Editor**. И верно обратное: отредактировав поля в **Properties Editor**, мы видим изменения в таблице - зелёным цветом.

# **Просмотр сцены**

Теперь пора разобраться с тонкостями просмотра сцены. Объектами сцены называются объекты с расширением **.SOBJ** (SceneObject). Для просмотра объекта надо дабл-кликнуть по нему в Object Browser или из popup-меню нажать **Open** - появится панель просмотра **SceneObject**.

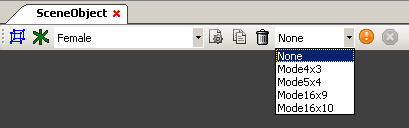
Пользователь смотрит на сцену с некоторой точки. Настройка точки просмотра называется **камерой**. С неё и начнём.

## **Управление камерой**

Всё управление камерой в редакторе происходит с нажатой кнопкой **Alt**:

* **Поворот камеры**: нажать Alt и левую кнопку мыши и двигать курсор.
* **Перемещение камеры в фокальной плоскости**: нажать Alt и среднюю кнопку мыши и двигать курсор.
* **Приближение/отдаление (Zoom in/Zoom out)**: нажать Alt и вращать колёсико мыши, либо нажать Alt и правую кнопку мыши и двигать курсор.

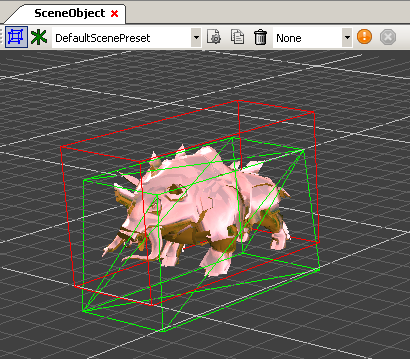
Настройки камеры выбираются выпадающим списком, как показано на рисунке:

  
**Рис. ... Тулбар просмотра объектов и выбор формата камеры.**

Эти настройки определяют формат игровой камеры, через которую идёт просмотр сцены: соотношение сторон экрана отрисовки, ограничения на поворот и приближение камеры.

## **Коллизии**

**Коллизии** - не совсем точное название, под которым понимаются "габариты" объекта. Коллизии влияют на "личное пространство", занимаемое объектом - пространство, в которое другие объекты не могут попасть. Включить показ коллизий можно кнопкой **Collision** (левая кнопка на тулбаре):

  
**Рис. ... Отображение коллизий.**

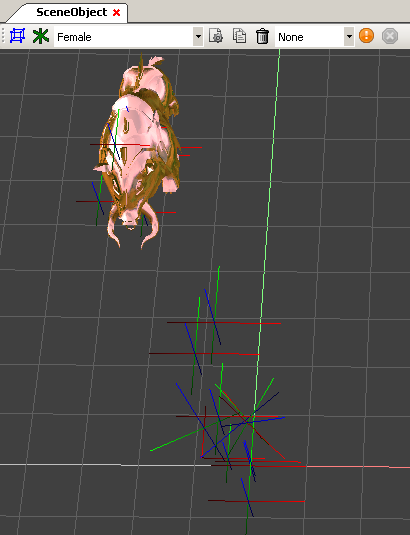
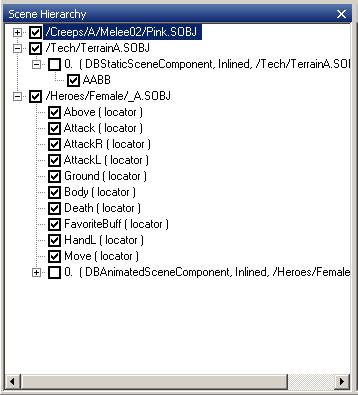
Коллизии отображаются зелёным цветом. Красным отображается граница, которая влияет на отрисовку объекта. Если в камеру не попадает ни один край пространства, ограниченного красной границей, объект рендериться не будет - так правильно с точки зрения оптимизации вычислений.

Если включить коллизии для нашей воздушной свинки с анимированным **placement**, то можно заметить, что зелёные коллизии не меняют своих параметров, и остаются жёсткими, в то время как красные границы "дышат" вместе с хрюшкой, отражая все эволюции масштаба, положения, и прочие душевные порывы анимированного персонажа.

## **Локаторы и панель иерархии сцены**

**Локатор** - это именованная ориентируемая точка, принадлежащая объекту. Локаторы нужны для привязки к определённым точкам всякой полезной шняги - мечей, эффектов и т.д. Локаторы редактируются в Properties Editor (см. пример задания локатора в [Редактирование списков](https://confluence.nivalnetwork.com/display/prog/Editor+Documentation#EditorDocumentation-%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D1%81%D0%BF%D0%B8%D1%81%D0%BA%D0%BE%D0%B2)).

Список локаторов, а также список объектов сцены виден в специальной панели **Scene Hierarhy**, которую можно показать через меню **View**. В этой панели чек-боксами можно скрыть/показать объекты сцены. Например, в нашей сцене мирной демонстрации можно скрыть террейн и девушку:

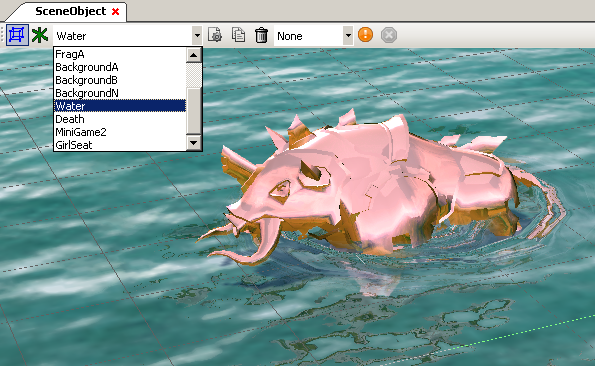
   
**Рис. ... Показ локаторов, панель иерархии сцены.**

Локаторы девушки при этом продолжаются отображаться и забавно шевелятся.

При выделении в Scene Hierarhy объектов сцены их свойства показываются в Properties Editor.

## **Пресеты просмотра сцены**

Для того, чтобы посмотреть, как выглядят объекты в разном окружении, применяются различные пресеты - они выбираются из левого выпадающего списка на тулбаре. Например, с помощью пресета **Water** наша свинка становится морской (надо только не забыть спустить её с небес на землю, точнее, на воду):

  
**Рис. ... Купание красного коня.**

Набор пресетов можно настраивать. Для этого нажимаем на кнопку **Options** и... попадаем в **Properties Editor**. Здесь можно редактировать параметры текущего пресета и настройки камеры, которые привязываются к пресету.

**TODO:** описать поля пресета.

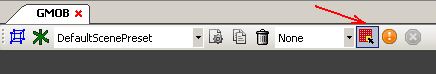
Также можно создать новый пресет копированием уже существующего (при этом спрашивается имя нового пресета), а также удалить текущий пресет (при этом спрашивается, хотим ли мы его удалить). Для всего этого есть специально обученные кнопки на тулбаре.

## **Запись видео**

Это очень грустный и короткий параграф нашего учебника. Потому что кнопка есть, а записи видео нет (**28.01.2010**).

## **Редактирование проходимости объекта**

До сих пор речь шла про просмотр сцены (объекты, название которых оканчивается на **.SOBJ**). Теперь откроем в Object Editor объект с расширением **.QRTR**, например **/Buildings/A/Barrack/\_Bottom.QRTR** (это такое красивое здание, которое вот-вот сгорит). Этот объект открывается для просмотра также по дабл-клику, но в тулбаре при этом появляется новая кнопка: **TileSet**:



Эта кнопка позволяет редактировать тайлы (элементы поверхности), которые будут проходимы для других объектов. Это нужно для таких сложно устроенных объектов, как здания.

По нажатию на эту кнопку здание становится полупрозрачным, и под ним обнаруживаются редактируемые мышкой и видимые невооружённым взглядом тайлы (непроходимые показаны красным цетом).

Есть ещё одна особенность редактирования тайлов проходимости: в **Properties Editor** во поле **collisions** задаются границы карты проходимости для объекта. Вот их можно менять. Руками.

## **Режим раскраски mip'ов**

**Mips** - это менее детальная прорисовка текстур (при удалении камеры). Отображать, где и какие используются mip'ы, позволяет специальный режим **Show MipMaps**, который включается из меню Tools.

  
**Рис. ... Режим отображения mip'ов.**

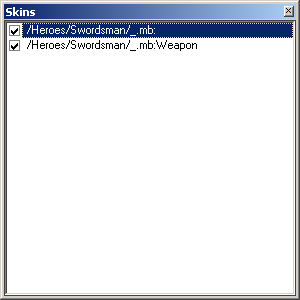
В этом психоделичном представлении разные уровни mip'ов раскрашиваются в разные цвета. Это делается следующим образом. Для каждого уровня mip'а (от 0 до 7) имеется свой цвет, на который умножается цвет пикселов:

| **Уровень mip'а** | **Цвет** |
| --- | --- |
| **0** | белый (255, 255, 255) |
| **1** | красный (255, 0, 0) |
| **2** | зелёный (0, 255, 0) |
| **3** | синий (0, 0, 255) |
| **4** | жёлтый (255, 255, 0) |
| **5** | малиновый (255, 0, 255) |
| **6** | голубой (0, 255, 255) |
| **7** | серый (128, 128, 128) |

0-й уровень mip'а - самая детальная прорисовка.

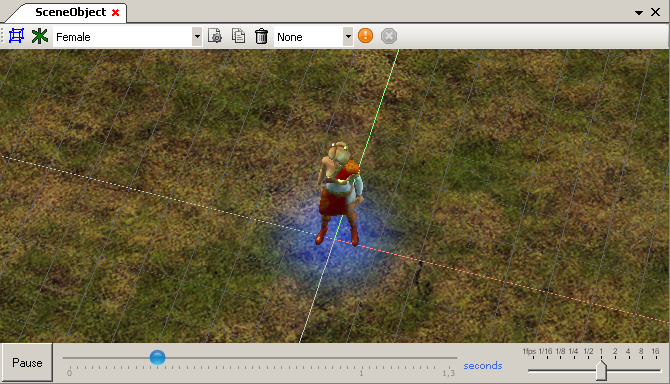
## **Скины**

Скины - это часть экипировки или одежды объектов. Их можно показывать/скрывать в специальной панели **Skins** (доступна через меню **View**). Причём скрыть можно как скины, так и сам объект:

  
**Рис. ... Выбор скинов.**

## **Управление анимацией (Time Slider)**

Из просмотра сцены и объектов можно управлять скоростью анимации, остановить/возобновить проигрывание анимации, а также перейти на нужный кадр:

  
**Рис. ... Панель Time Slider.**

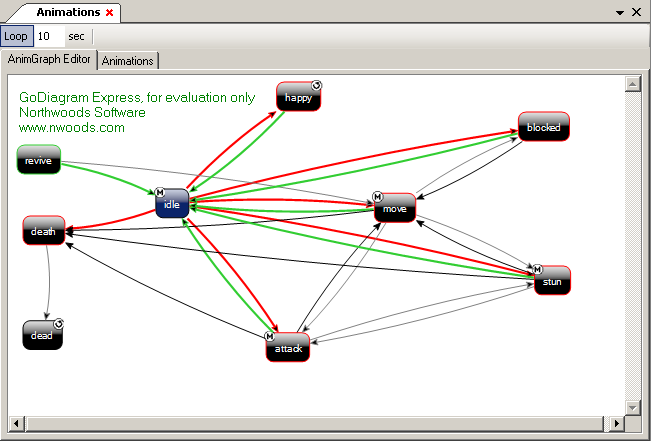
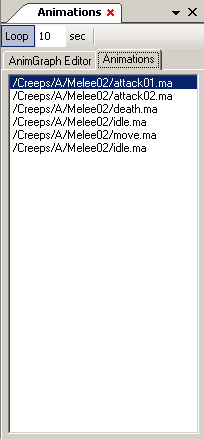
Из неочевидных возможностей здесь можно отметить переключение между единицами измерений - клик на **seconds** приводит к единицам измерений **frames**, и обратно.

# **Редактирование графа анимаций**

**Граф анимаций**

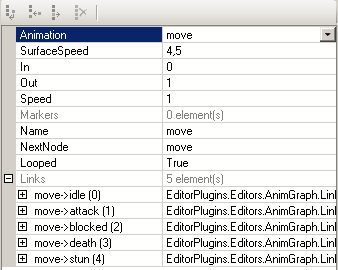
|  | **Граф анимаций**  Граф Анимаций - это не "кто", а "что". Так называют конструкцию, состоящую из узлов-анимаций и рёбер-переходов между ними. Редактор позволяет редактировать этот граф. |
| --- | --- |

Откроем панель анимаций в меню **View -> Animations**. Эта панель состоит из двух вкладок: собственно графа и списка анимаций:

​​   
**Рис. ... Панель графа анимации.**

В графе можно создавать и удалять узлы (nodes), и связывать их переходами. Узлы бывают двух видов: нормальные и рандомные. Нормальным узлам соответствует одна анимация, а рандомным можно привязывать несколько анимаций, и тогда какие из них проигрывать, будет выбираться в случайном порядке. Рандомные узлы помечаются на графе буквой **M** (Macro).

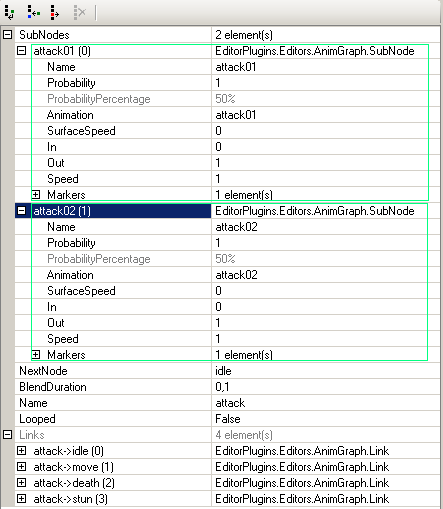
Свойства узлов графа редактируются в Properties Editor:

  
**Рис. ... Редактирование свойств нормального узла графа (с одной анимацией).**

Что означают эти поля:

| **Поле** | **Что означает** |
| --- | --- |
| **Animation** | Анимация - выбирается из списка |
| **SurfaceSpeed** | Скорость движения террейна при показе (используется в узлах **Move**) |
| **In** | Начальный кадр анимации |
| **Out** | Конечный кадр анимации |
| **Speed** | Скорость проигрывания анимации |
| **Markers** | Список маркеров |
| **Name** | Название |
| **NextNode** | Узел, на который идёт переход по-умолчанию |
| **Looped** | Признак того, что узел зациклен сам на себя (тогда NextNode - сам же этот узел) |
| **Links** | Список переходов из данного узла |

|  | **Узлы Looped на графе**  Узлы, у которых **Looped=true**, отмечены на графе специальным значком: |
| --- | --- |

  
**Рис. ... Редактирование свойств рандомного узла графа (с двумя анимациями).**

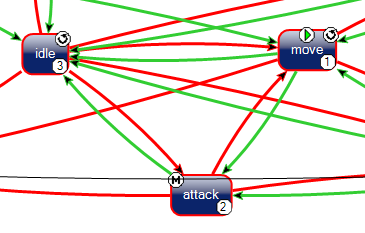
Рандомный узел имеет дополнительно следующие поля: **SubNodes** - вложенные узлы, между которыми идёт случайный выбор, **BlendDuration** - время блендинга между вложенными узлами. Вложенные узлы аналогичны обычным, но они содержат дополнительное поле **Probability**, которое означает вес при расчёте вероятности.

Переходы тоже имеют свои свойства. **Blending** - тип блендинга при переходе (сейчас применяются PoseToPose и PoseToAnimation). **MinFrom** и **MaxFrom** - это окно в анимации, в течение которого может осуществляться переход в другую анимацию, может быть в интервале от 0 до 1, то есть в долях от длительности анимации-From. **Duration** - время блендинга при переходе.

Исходящие переходы раскрашиваются в зависимости от значения **MinFrom**: если оно равно 0 (быстрый переход), то он красится в красный цвет, иначе (не быстрый) - в оранжевый. Входящие переходы раскрашены в зелёный. Раскраска, разумеется, меняется при смене текущего узла графа.

Список анимаций (вторая вкладка панели Animations) позволяет только проигрывать выделенную анимацию. Сам набор анимаций настраивается в Майке, из редактора его не изменить.

Граф анимаций позволяет также проигрывать анимации в угодном нам порядке. Чтобы установить порядок проигрывания узлов графа, выделим мышью с нажатой **Ctrl** несколько узлов - они при этом будут помечены номером (1, 2, 3, ...):

  
**Рис. ... Проигрывание узлов анимации в заданном порядке.**

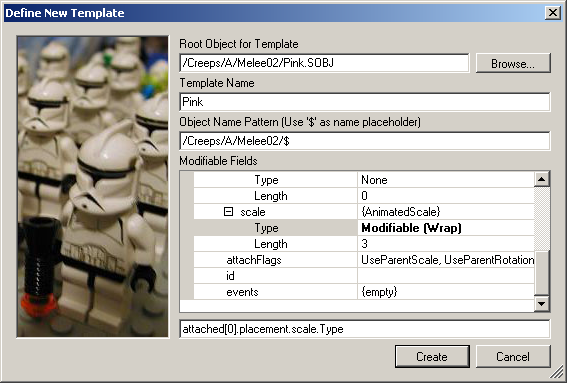
Нажмём на кнопу **Loop** сверху графа, если она ещё не нажата. По мере того как в панели просмотра объекта анимации будут сменять друг друга, проигрываемые узлы будут помечаться зелёным треугольничком - значит, сейчас "играет" этот узел. Сколько времени идёт на проигрывание одного узла - настраивается в поле рядом с кнопкой **Loop**.

# **Работа с шаблонами**

Шаблонами (template) называются объекты, из которых можно затем быстро создавать другие объекты, настраивая необходимый минимум их свойств при создании.

## **Создание шаблона**

Попробуем создать шаблон. В меню **File** выбираем **New Template...** - появляется диалог определения шаблона:

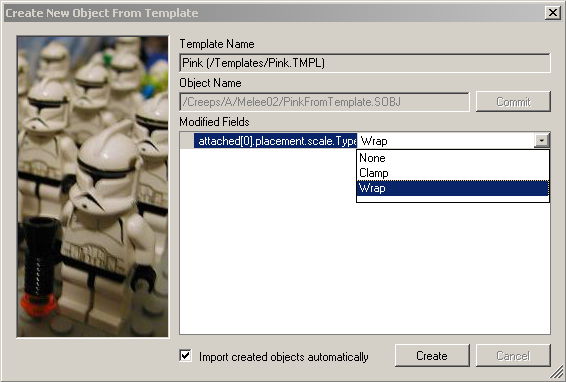
  
**Рис. ... Создание шаблона.**

В этом диалоге надо задать объект, по образу и подобию которого будет создан шаблон (это делается кнопкой **Browse...**), имя шаблона, а также место, где будут создаваться объекты из этого шаблона.

В разделе **Modifiable Fields** настраивается, какие поля можно пометить как настраиваемые позже, при создании объектов из шаблона. Ключевое слово здесь **Modifiable**: в выпадающих списках для таких полей нада выбрать пункт **Modifiable(<имя типа>)**.

## **Создание объекта из шаблона**

Теперь сделаем на основе этого шаблона новенький объект. В меню **File -> Create Object From Template** открывается подменю со списком всех заведённых шаблонов, в том числе и наш **"Pink"**. В диалоге сначала набираем имя объекта, затем жмём кнопку **Commit**, после этого объект будет создан, и можно будет настроить Modifiable-поля:

  
**Рис. ... Создание объекта из шаблона.**

Теперь осталось нажать на кнопку **Create**, и изменения в полях дойдут до объекта (он всё равно уже будет создан после нажатия Commit).

# **Исходные и игровые данные, импорт**

**Процесс конвертации данных из исходного формата**

**Импорт** - это процесс конвертации данных из исходного формата (например, анимаций и геометрии, которые редактируются в Майке) во внутренние форматы, которые загружаются игровым кодом. Для импорта надо настроить каталог исходных данных (и, конечно, чтобы сами исходные данные тами были), а также настроить использование Майки (см. раздел [Getting Started](https://confluence.nivalnetwork.com/display/prog/Editor+Documentation#EditorDocumentation-EditorDocumentationGettingStarted)).

Импорт ресурсных данных доступен в редакторе из разных мест:

Из меню **File**:

* **Import...** - импортирует ресурсы из указанного файла
* **Import Recent** - предлагает список последних импортированных файлов

Из контекстного меню в **Object Browser** и **Properties Editor**:

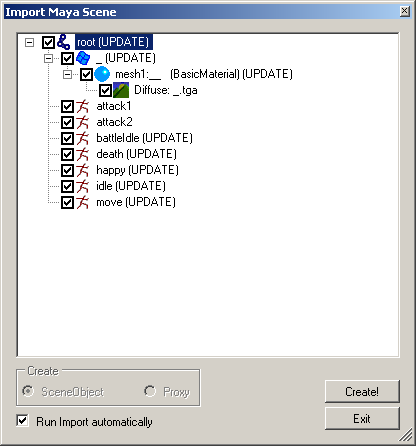
* **Import** - импортирует ресурсы выделенного объекта
* **Import Fast** - импорт объекта из имеющихся в **/Data** файлов, без запуска Майки, и вообще без использования каталога исходных данных
* **Recursive Import...** - рекурсивный импорт вглубь структуры объектов

## **Import Resource, или Большая Красная Кнопка**

Большой Красной Кнопкой называется у художников загадочный, но очень полезный функционал, который выполняется при импорте файла (вызовом **Import...**). Этот функционал отличается от того, что вызывается при контекстном меню из объектов. Отличия заключаются в следующем:

1) По Большой Красной Кнопке создаётся структура объекта, а из "контекстного" импорта структура не трогается.  
2) По Большой Красной Кнопке импортятся некоторые анимированные поля из майских данных, которые из "контекстного" импорта тоже не трогаются (см. ниже).

Хотя это не кнопка, не большая, и не красная, но при нажатии на неё вызывается диалог **Import Maya Scene**:

  
**Рис. ... Диалог импорта сцены из файла.**

Если выбранному исходнику ещё не соответствует объекта с таким же именем, то он будет создан (учитывается наличие объекта именно в том же месте в папках, что и импортируемый файл, но в каталоге Data). Радиобатонами **SceneObject/Proxy** выбирается в этом случае, какой объект будет создаваться (**Proxy** - это объект **DBSceneComponent**, он нужен для эффектов, в остальных случаях нужно создавать SceneObject). Если объект, соответствующий названию файла исходника, уже существует, то этот диалог настраивает изменения существующего объекта, и выбор SceneObject/Proxy в этом случае недоступен.

Если галка **Run Import automatically** установлена, значит, по полям объектов будет делаться импорт. В противном случае будет только создаваться/меняться структура объекта. Сбрасыванием галок в показываемой структуре исходника можно удалять соответствующие элементы в структуре объекта.

Вот перечень полей, по отношению к которым импорт неравнодушен, и будет переписывать их из майских структур по нажатию **Import Resource...**:

Класс **BasicFXMaterial**:

* **DiffuseMul1**
* **opacity**
* **uOffset1**
* **vOffset1**

Класс **AttachedSceneComponent**:

* **placement**

В Майке атрибуты, данные которых идут в эти поля, специальнам образом помечаются. Но это уже совсем другая история.

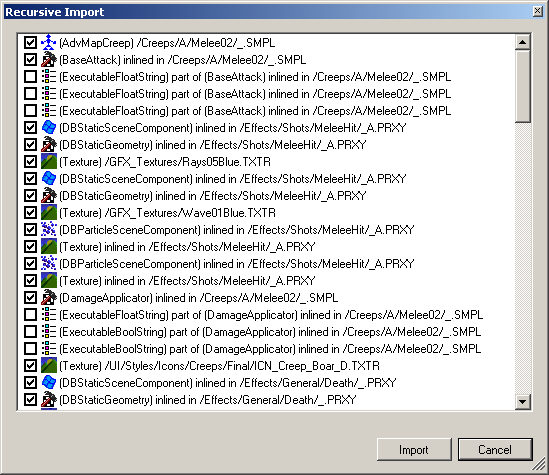
## **Обычный импорт**

Теперь рассмотрим, как делается импорт со стороны конкретных объектов (из контекстного меню).

|  | **Import Fast**  Быстрый импорт - для майских данных это импорт из промежуточного формата файлов **.maya**, которые находятся в каталоге Data. Эти файлы появляются там во время обычного импорта. В других форматах быстрый импорт может означать что-то другое. |
| --- | --- |

|  | **Рекурсивный импорт**  Рекурсивный импорт - это операция, про которой сначала составляется инвентарный список объектов, на которые идут ссылки от текущего, и по всем этим объектам идёт импорт. А обычный импорт - это импорт только inlined-объектов.  В случае если нажать **Recursive Import** на каталоге, то покажется список объектов внутри каталога. И внизу будет галочка: учитывать ли ещё и связанные объекты (т.е. делать рекурсивный список всех объектов в каталоге). |
| --- | --- |

Вот так выглядит диаложек выбора объектов, которые будут импортиться, при рекурсивном импорте некоторого объекта:

  
**Рис. ... Диалог рекурсивного импорта.**

После импорта можно посмотреть в панель Output. Даже нужно. Мало ли что.

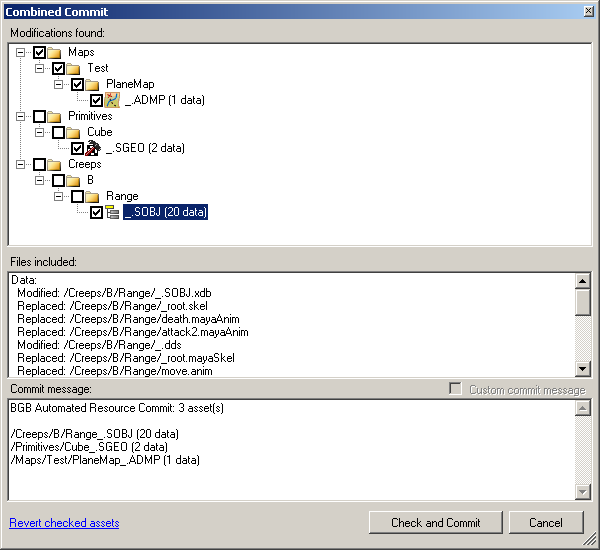
|  | **Нормально ли прошёл импорт?**  Критериями корректного завершения импорта являются:  1) В Output нет ошибок (ворнинги и сообщения не в счёт) 2) Сцена открывается |
| --- | --- |

Бинарные данные в ходе импорта автоматически изменяются, поэтому, даже если при этом не была изменена структура объектов и значения полей, данные ресурсов реально меняются даже без нажатия **Save**.

# **Save And Commit, или Большая Зелёная Кнопка**

**Commit** - функция коммита изменённых ресурсов (в папке Data), а также исходных данных. Основная идея - быстро и удобно делать коммиты, не вылезая из редактора. В народе этот функционал ещё известен как Большая Зелёная Кнопка.

Из меню **File -> Commit...** вызывается окно:

  
**Рис. ... Окно Combined Commit**

После поиска изменённых объектов в окне показываются три панели. В верхней панели показываются изменённые объекты, с указанием их места в дереве, и кол-ва изменённых файлов по каждому объекту (текст в скобках). Эти ресурсы можно выбирать, и только отмеченные пойдут в коммит. Выбирать можно и папки целиком.

Если встать на какой-то объект, то средняя панель показывает список файлов, который пойдёт в коммит, если объект будет выбран - отдельно список файлов из папки Data, отдельно - список исходных файлов.

Нижняя панель показывает текст сообщения коммита, которое генерится окошком. Текст может быть изменён руками (тогда установится галочка **Custom commit message**). При сбросе галочки текст меняется на автосгенерённый.

Жмём на кнопку **Check and Commit** - и коммит идёт.

|  | **Коммит исходных файлов**  В коммит пойдут не только изменённые файлы из папки **Data**, но и исходные файлы (папка **pf\_art**), если они были изменены. В случае, если изменён только некоторый исходный ресурс из pf\_art, но файл в Data не изменён (т.е. не вызывался импорт), то такой ресурс в коммит не идёт и в окне Combined Commit даже не отображается. |
| --- | --- |

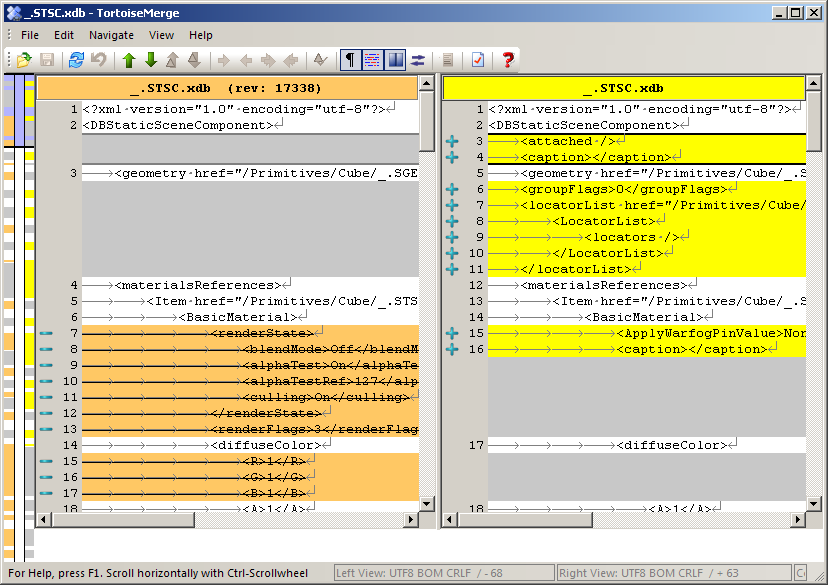
|  | **Коммит удаления**  Если удален xdb-файл, то он возникнет в списке Modifications found с красным крестиком, и его удаление можно закоммитить. Связанные с ним файлы при этом не трогаются. |
| --- | --- |

**Revert и Diff**

По нажатию правой кнопки мыши на элементе в дереве появляются два пункта: **Revert** и **Diff**. Revert позволяет откатить изменения по данному ассету (изменения откатятся на диске - и в /Data и в /pf\_art, в базу SVN отреверченные изменения не пойдут).

При выделении галкой элементов внизу окна Combined Commit появляется ссылочко-кнопка **Revert checked assets**: при её нажатии все выделенные ассеты будут отреверчены.

Diff позволяет сравнить в текстовом виде то, что есть в базе SVN, и изменённую версию данного xdb-файла:



При этом вызывается diff-утилита TortoiseSVN'а.

# **Редактирование карты**

Теперь мы займёмся захватывающей деятельностью - станем редактировать карту. Карты представлены в редакторе объектами с расширением **.ADMP**.

Пройдёмте: **Root/Maps/Multiplayer/MOBA/\_.ADMP**. Откроем дабл-кликом эту карту. На ней видны не только складки местности, но и разные объекты, населяющую эту территорию: от камней, мелких кустиков и цветочков до башен и крипов.

Управление камерой здесь такое же, что и при просмотре сцен. Редактор карт позволяет ставить, настраивать и удалять объекты на карте, модифицировать рельеф и ландшафтное покрытие террейна (местности), настраивать освещение.

|  | **Минимап**  Полезная фишка - минимап, который показывается в отдельной панельке (показывается из меню View):  **Рис. ... Минимап**  Кнопки + и - масштабируют карту в основном окне. Красный кружочек показывает место на карте, где мы находимся. |
| --- | --- |

## **Настройка карты**

В поле mapType у AdvMapDescription есть специальный тип – Series. На карте такого типа не показываются announcement'ы, а также по умолчанию не видны портреты героев (кроме своего; других игроков можно показать из скрипта).

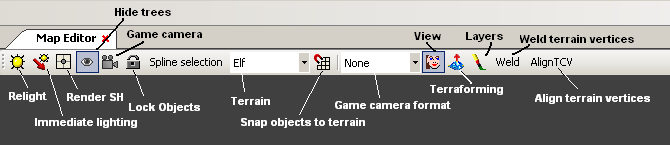
В AdvMapSettings с некоторых пор есть некоторые параметры, которые задавались глобально в SessionRoot, AILogic и т.д.:

* **customBattleStartAnnouncement** – оверрайдит battleStartAnnouncementParams в UI/Content/\_.UIDT.xdb, если **UseСustomBattleStartAnnouncement** установлен в **true**;
* **overridePortalTalent** – оверрайдит portal в /GameLogic/AILogic.AILP;
* **overrideGlyphSettings** – оверрайдит glyphsDB в /GameLogic/SessionLogic.ROOT и glyphSettings в /GameLogic/AILogic.AILP;
* **creepLevelCap** – оверрайдит creepLevelCap в /GameLogic/AILogic.AILP, если >= 0.

Также есть параметр **natureMap.visual.overburnRethreat** в Terrain (обычно \_.TERN). Если **true**, то при смене террейна на нейтральный (отступлении) на него накладывается эффект "засветки", как при наступлении фракционного террейна. Да, retHreat. Пока, увы, с опечаткой.

## **Тулбар**

На рисунке ниже показан тулбар редактора карт.

  
**Рис. ... Тулбар редактора карт.**

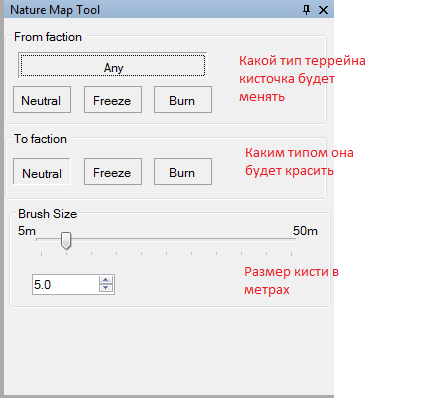
Вот что означают эти кнопки:

| **Relight** | Пересчитать освещение полностью |
| --- | --- |
| **Immediate lighting** | Включить/выключить режим постоянного пересчёта освещения |
| **Render SH** | Показать точки сферических гармоник |
| **Hide trees** | Скрыть/показать деревья |
| **Game camera** | Показывать картинку игровой камеры |
| **Lock Objects** | Включить/выключить режим блокировки объектов |
| **Spline selection** | Включить/выключить режим редактирования только сплайнов |
| **Terrain** | Выбор типа террейна |
| **Snap objects to terrain** | Включить/выключить режим учёта высоты террейна при перемещении объектов |
| **Game camera format** | Выбор формата игровой камеры |
| **View** | Режим просмотра/редактирования террейна/правки слоёв |
| **Terraforming** | Режим просмотра |
| **Layers** | Режим правки слоёв |
| **Weld terrain vertices** | Чинить стыки террейна |
| **Align terrain vertices** | Выровнять вершины террейна |

## **Настройка природной карты**

По умолчанию атакующий террейн распространяется по сегментам, по мере поднятия флагов. Подробности сегментного разбиения карты расписаны в статье [Новый атакующий террейн](http://confluence.nivalnetwork.com/pages/viewpage.action?pageId=75598152). При этом имеется возможность вместо сегментов задавать природную карту через раскраску террейна кисточкой в редакторе. Тогда управлять атакующим террейном можно через [скрипт](https://confluence.nivalnetwork.com/display/dot/Lua+scripts) (usecase: всякие PvE'шные карты). Для использования раскраски природной карты в настройках террейна надо включить галочку editableNatureMap (.TERN/NatureMap). Так же ее можно указать в диалоге создания карты.

Далее, в панели инструментов редактора карт есть кнопка режима редактирования типа террейна (Nature map). При её нажатии становится доступна панель Nature Map Tool (сделана по аналогии с Terrain Height Tool):

  
**Рис. ... Редактирование природной карты через кисточку**

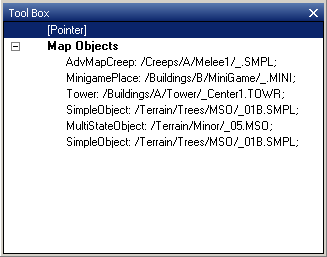
**Замечание:** стоит ещё раз подчеркнуть, что опции взаимоисключающие: природную карту можно задавать либо через сегменты, либо кисточкой. Кроме того, если природная карта задана через сегменты, на продвижение террейна нельзя влиять скриптовыми функциями.

## **Выделение, перемещение и вставка объектов**

Любой объект на карте можно выделить мышкой, и будут показаны его коллизии/габариты. Объекты можно перетаскивать по поверхности террейна.

Чтобы вставить новый объект, надо выделить его в Object Browser, и мышкой в карте кликнуть на место, где объект поставить. Ставятся на карту объекты с расширением **.QRTR**, **.MSO** и **.GMOB**. Можно один раз выбрать объект и поставить его на карту множество раз. Чтобы выключить режим вставки объекта, нажимаем на **Esc**.

Для постановки объектов также полезно пользоваться тулбоксом. Тулбокс здесь работает в полном соответствии со своим названием как ящик, в который можно положить нужную вещь, чтобы затем её оттуда быстро достать и водрузить на карту.

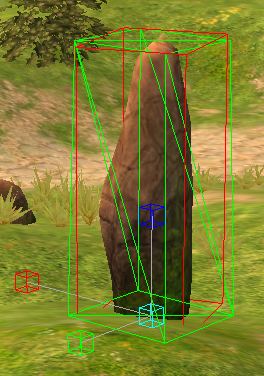
  
**Рис. ... Тулбокс, использование для карты**

Чтобы добавить в этот ящик какую-то вещь, надо выделить эту вещь на карте, а затем в тулбоксе в контекстном меню выбрать пункт **Add Item From Editor**. В списке появится объект. Чтобы его поставить на карту, надо щёлкнуть на объект, и поставить на нужное место.

|  | Сейчас этот функционал глючит - Toolbox будет переписываться (15 марта 2010). |
| --- | --- |

## **WER-операции**

Объекты можно выделять, в Properties Editor при этом будут показываться их свойства. Ряд операций редактирования можно делать в удобном визуальном режиме: **перемещение (W)**, **поворот (E)**, **масштабирование (R)** - в скобках показаны клавиши, которые включают режим редактирования положения объекта.

​​   
**Рис. ... Режимы перемещения, поворота и масштабирования объекта на карте.**

В этих режимах можно мышкой делать с объектами всякое. Попробуйте.

## **Изменение террейна**

При нажатии на кнопку **Terraforming** включается режим редактирования террейна - в нём удобно делать холмы, овраги и прочие складки местности. В этом режиме белыми квадратами показываются ориентируемые точки по краям элементов террейна, которые можно двигать и поворачивать. Для того, чтобы начать редактирование точки, надо выделить её мышью (квадратик станет красным), а затем нажать **W** (перемещение) или **E**(вращение).

  
**Рис. ... Режим редактирования террейна (перемещение точки).**

Масштабирование (**R**) включить тоже можно, но изменение масштаба на террейне работать не будет.

|  | **Множественное выделение**  Полезная фича - выделение сразу нескольких точек террейна - можно редактировать одним движением сразу несколько точек. |
| --- | --- |

## **Раскраска слоёв**

Слои - это покрытие террейна (например, трава, дорога), которое рисуется на карте наподобие рисования кистью. При рисовании слоёв можно выбрать размер кисти, и каким слоем будем рисовать. Для этого надо открыть панель **Terrain layers** (из меню **View**):

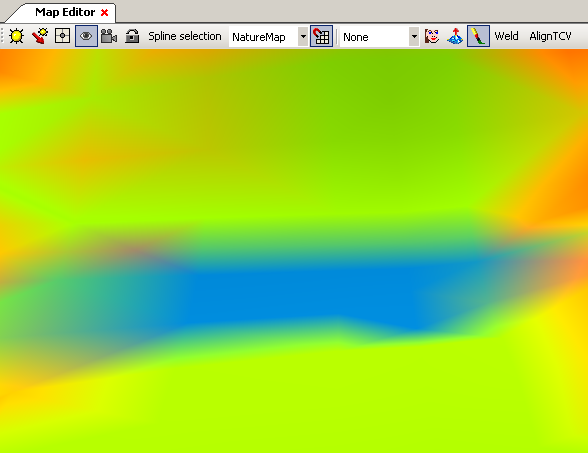
  
**Рис. ... Панель выбора кисточки закрашивания.**

Режим редактирования слоёв включается кнопкой **Layers** в тулбаре редактора карт. При этом кисть показывается на местности траурным квадратом:

  
**Рис. ... Режим редактирования слоёв.**

**Holes layer** сейчас не используется (придуман для дырок в террейне). **Local layer** - слой, который будет менять свой окрас в зависимости от того, кому принадлежит местность. Глобальные слои не меняются от смены "полярности" территории.

Слои на карте соединяются плавно, их граница получается размытой. Чтобы видеть более наглядно границы между слоями, применяется режим отладки слоёв (**Debug Mode**), который включается кнопкой на панели Terrain layers.

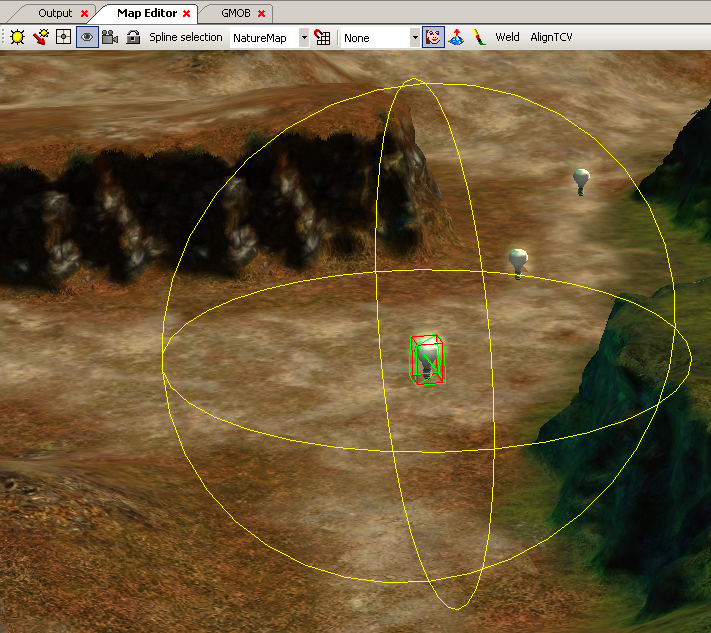
  
**Рис. ... Debug Mode редактирования слоёв (это тот же участок террейна).**

Есть ещё один режим: **Debug Mode Issues Only**. В нём отображаются в режиме отладки только те части террейна, где количество слоёв больше 5.

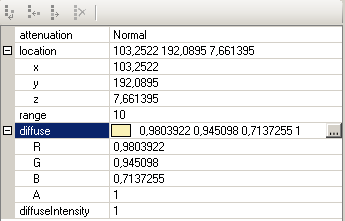
## **Освещение**

Освещение настраивается в редакторе карт двумя основными способами. Первый - настройка точечных источников света **PointLightInfo** (отображаются в виде лампочек). Можно перемещать источники света и настраивать их параметры. Второй - общий свет через объект **LightEnvironment**.

### **Точечные источники света: PointLightInfo**

**  
Рис. ... Источники света (PointLightInfo).**

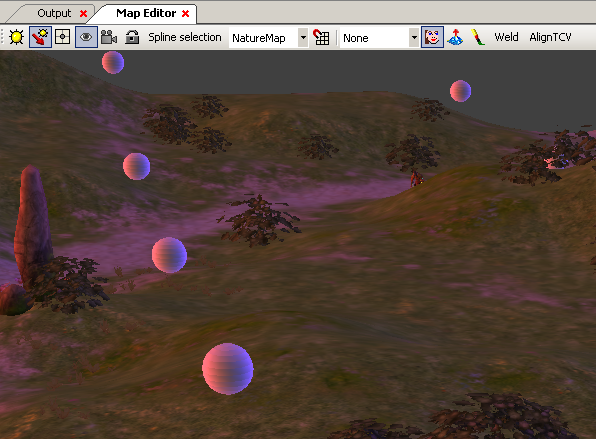
Параметрами являются: затухание, координаты, размер, цвет, яркость:

  
**Рис. ... Свойства PointLightInfo**

Чтобы изменения этих свойств отразились на освещении объектов на карте, надо нажать кнопку **Relight** на тулбаре. Либо включить режим **Immediate lighting**.

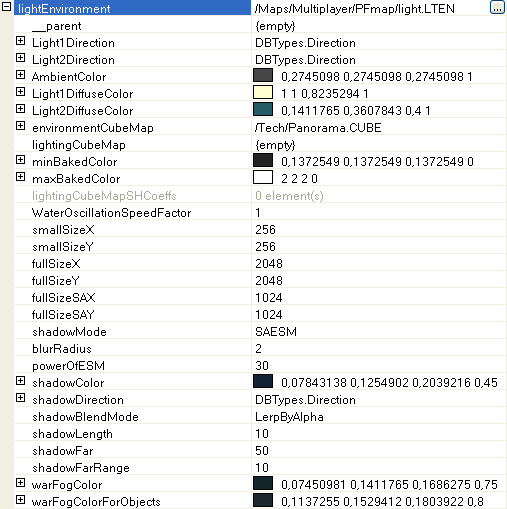
### **Объекты LightEnvironment и BakedLigting**

На рисунке ниже показано, до чего можно доредатироваться при настройке освещения:

  
**Рис. ... Отображение сферических гармоник.**

Показаны точки сетки сферических гармоник (вызывается кнопкой **Render SH**). В этих точках рассчитывается освещение при нажатии на **Relight** (и в режиме **Immediate lighting**).

Этот тип света настраивается через объект **LightEnvironment** (поле **lightEnvironment** у объекта карты):

  
**Рис. ... Свойства объекта LightEnvironment.**

Этот объект задаёт два направленных источника света и один рассеянный. На показанном примере настройки направленных источников такие: один источник - красный ("Солнце на закат"), второй - синий ("Луна на восход"). Описание полей LightEnvironment, которые можно трогать:

| **Поле** | **Что означает** |
| --- | --- |
| **Light1Direction** | Направление источника света 1 |
| **Light2Direction** | Направление источника света 2 |
| **AmbientColor** | Цвет рассеянного света |
| **Light1DiffuseColor** | Цвет источника света 1 |
| **Light2DiffuseColor** | Цвет источника света 2 |
| **minBakedColor** | Ограничение на цвет снизу |
| **maxBakedColor** | Ограничение на цвет сверху |
| **WaterOscillationSpeedFactor** | Скорость колебания воды (задаётся для всей воды на кате) |
| **fullSizeX** | Размер по X карты теней для режима PCF ( Оптимально 2048 ) |
| **fullSizeY** | Размер по Y карты теней для режима PCF ( Оптимально 2048 ) |
| **fullSizeSAX** | Размер по X карты теней для режима SAESM ( Оптимально 1024 ) |
| **fullSizeSAY** | Размер по Y карты теней для режима SAESM ( Оптимально 1024 ) |
| **shadowMode** | PCF - старый; SAESM - новый с сильным блюром; AnisoESM - экспериментальный ( do not use ) |
| **blurRadius** | Размер фильтра SAESM. Может принимать четные значения >2. 2,4,6,8 ... |
| **powerOfESM** | Настройка резкости перепада на границе тени. Оптимальное значение ~80. Чем больше, тем граница плотней. При значениях > 150 ( в зависимости от blurRadius ) начинают провялятся артефакты, связанные с точностью floating point. |
| **shadowColor** | Цвет тени |
| **shadowDirection** | Направление тени |
| **shadowBlendMode** | Способ отрисовки тени |
| **shadowLength** | Длина тени в направлении источника света. Определяет какой объем сцены попадает в тень. |
| **shadowFar** | Длина тени в направлении камеры. За этим расстоянием тени плавно исчезают |
| **shadowFarRange** | Параметр позволяющий регулировать плавность исчезания тени |
| **warFogColor** | Цвет тумана войны |
| **warFogColorForObjects** | Цвет тумана войны над объектами |
| **depthBias** | Смещение shadows-текстуры, которое предотвращает артефакты самозатенения |
| **biasSlope** | Коэффициент в выражении для вычисления смещения shadows-текстуры:  Offset = m \* biasSlope + depthBias  где m = max(abs(delta z / delta x), abs(delta z / delta y)) |

Ещё один объект привязан к карте, которые связан со светом: он настраивает сетку расчёта сферических гармоник. Зовут его **BakedLighting**. Вот его поля:

| **Поле** | **Что означает** |
| --- | --- |
| **dynObjLightingGridResX** | Кол-во точек по X |
| **dynObjLightingGridResY** | Кол-во точек по Y |
| **rendererEnvironmentEffect** | Влияние освещения от окружения (земли, деревьев, зданий) |
| **rendererEnvironmentRange** | Расстояние, на которое распространяется освещение от окружения |

## **Редактирование сплайнов**

Сплайн - гладкая кривая линия, которая определяется точками, которые ставит пользователь. Сплайнами в редакторе карт, например, настраивается граница воды:

  
**Рис. ... Настройка сплайнов.**

При выделении центральной метки сплайна (сейчас это флажок) становятся видны как точки сплайна, так и сама кривая, а на тулбаре появляются ещё две кнопки (на рисунке они показаны стрелочкой). Одна кнопка позволяет добавить точку, вторая - удаляет текущую точку.

Сплайн можно двигать целиком, также можно перемещать отдельные точки, предварительно выделив точку мышкой.

|  | **Кнопка Spline selection**  В тулбаре есть кнопка с простым русским названием **Spline Selection**. Она включает режим редактирования только текущего сплайна, т.е. другие объекты на карте в этом режиме не редактируются. Сделано, чтобы при двигании точек сплайнов не порушить всё вокруг. |
| --- | --- |

## **Эх, дороги**

Ещё одной фичей, которая использует сплайны, являются дороги. Дорога ставится на карту и является полоской переменной ширины с бордюрами, которая располагается там и так, как ей скажут.

  
**Рис. ... Редактирование дороги.**

### **Ставим дорогу**

Открываем карту, куда будем ставить дорогу. Дорога ставится, как любой объект, выделением в Object Browser (например, **/Tech/Road.ROAD**), и последующим кликом на нужном месте в карте. При этом на карту ставится головной объект дороги (красный флажок), показывается вся дорога. При выделении этого объекта показываются элементы дороги, которыми её можно редактировать.

### **Редактирование сплайна дороги**

Дорогу можно перетаскивать с место на место (сейчас эта фича временно не работает), а можно редактировать. Дорога устроена следующим образом. Центральный хребет (жёлтый) и границы (зелёные) - задаются сплайнами. Точки этих сплайнов можно двигать, и можно добавлять/удалять, всё как обычно. Двигать точку сплайна можно мышкой, ставить новую - выделением соседней и нажатием на кнопку на тулбаре (см. выше про редактирование сплайнов).

Точки сплайна центрального хребта - красные, обочин - зелёные. Зелёные точки задают ширину дороги в соответствующем профиле. Жёлтые "шпалы" отрисовываются автоматически.

Удалять дорогу совсем просто. Выделяем главный объект (флажок), нажимаем Delete. Была дорога - и нет. (Не забываем про Ctrl+Z).

В редактировании дорог также работает кнопка **Spline selection** (см. выше). При этом можно двигать и красные точки и зелёные.

### **Параметры дороги**

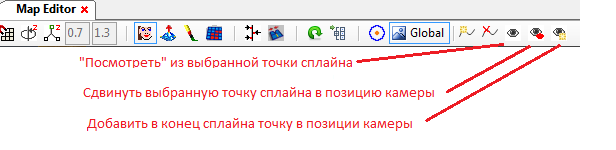
В **Properties Editor** показываются свойства дороги, которые можно (и нужно) редактировать. Объект **gameObject** - это и есть дорога. В нём есть список точек **points** - это точки сплайна дороги. Каждой точке можно задавать прозрачность участка дороги в районе этой точки (**opacity**). Свойство **width** задаёт ширину участка дороги в этой точке, а **location** - координаты точки. Ширину и координаты удобнее двигать мышкой.

Объект **properties** - это свойства всей дороги. Вот что они означают:

* **width** - базовая ширина дороги, относительно которой действуют настройки ширины в контрольных точках, и указываются некоторые другие параметры.
* **border** - значение от 0 до 1, задающая ширину бордюра в процентах от половины базовой ширины дороги. При 1 дорога базовой ширины состоит только из бордюров (поведение меняется при изменении ширины в контрольных точках). При 0 бордюры отсутствуют.
* **falloff** - значение от 0 до 1, задающая ширину затухания по краям полотна дороги в процентах от половины базовой ширины дороги. При 0 затухание отсутствует, дорога непрозрачная (но прозрачность в текстуре учитывается всегда);
* **tileU** - значение тайлинга текстуры полотна поперек направления дороги. Значение N означает, что текстура по ширине укладывается N раз (если fit = false, то укладывается N раз в базовую ширину);
* **tileV** - значение тайлинга текстуры полотна вдоль направления дороги. Имеет тот же смысл, что и tileU, совпадающие значения хорошо использовать для квадратных текстур.
* **tileVBorder** - значение тайлинга текстуры бордюра вдоль направления дороги. Значение 1 означает такую же плотность маппинга, что и по ширине бордюра, его хорошо использовать для квадратных текстур.
* **fit** - флаг, определяющий, должен ли маппинг текстуры растягиваться с изменением ширины дороги (true) или должен быть равномерным (false). При равномерном маппинге для расчета его плотности используется базовая ширина дороги, при растягивающемся (true) — ее текущее значение.
* **materialN**, **materialA**, **materialB** - материалы дороги для нейтральной земли и земель двух фракций. Части дороги, проходящие по разному террейну, текстурируются соответствующими материалами. В материале задаются две текстуры, для полотна дороги и для бордюра. Переходы между материалами плавные как и на террейне. Если materialA или materialB не заданы, вместо них будет использоваться materialN.

## **Камера и сплайны**

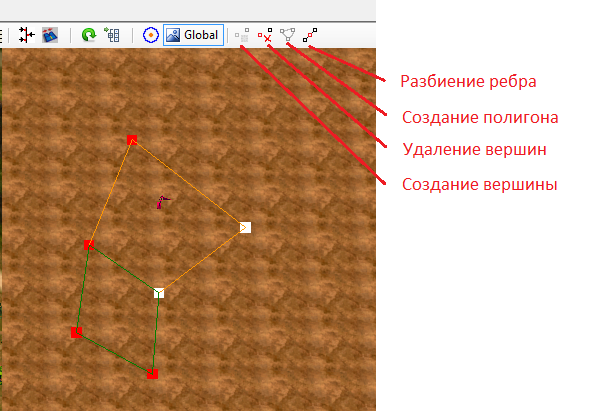
При редактировании AdvMapCameraSpline (камеры, летящей по траектории) теперь есть функции редактирования для того, чтобы можно было создавать траекторию, добавляя в сплайн текущую позицию камеры в редакторе.

  
**Рис. ... Кнопки для редактирования траектории полёта камеры.**

Что позволяет, хоть и неудобно, но редактировать камеру.

## **Скиптовые зоны**

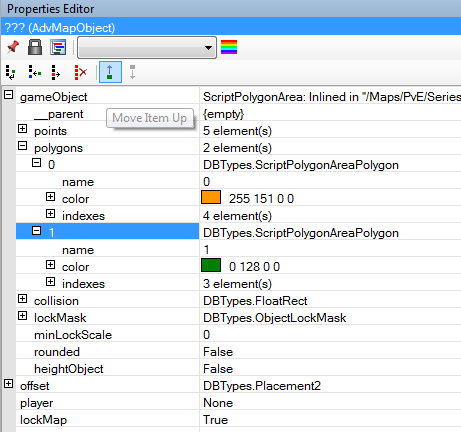
Кроме круглых ScriptedArea (как-то описанных [тут](https://confluence.nivalnetwork.com/display/dot/TriggersToSpecialPoints)) имеется возможность задавать на карте зоны многоугольниками. Для этого предназначены объекты типа ScriptPolygonArea. Несмотря на свое название, каждая такая зона содержит несколько многоугольников, построенных на одном наборе точек. Это сделано, чтобы удобно было использовать многоугольные зоны для имитации продвижения родной земли вдоль дорог силами скрипта.  
Шаблон полигональной зоны находится в /Tech/Default/ScriptPolygonArea.SPOL. При его установке на карту создается новая зона (также, как при добавлении дороги на карту).  
На карте выглядит как малиновый (magenta) флаг. При выборе этого флага на панели инструментов карты появляются кнопки для редактирования.

  
**Рис. ... Полигональная зона.**

На картинке видны 6 вершин – красные и белые (выбранные) квадраты и 2 многоугольника. Выбирать мышкой можно только вершины. несколько вершин выбираются с шифтом.  
Команды редактирования:

* Создание вершины - добавляет новую точку вблизи выбранной. Доступна, если выбрано не более одной вершины
* Удаление вершин - удаляет выбранные точки. Многоугольныки, построенные на этих точках, также удаляются, если в них остается меньше трех вершин. Доступна, если выбрана хотя бы одна точка.
* Создание полигона - создает новый многоугольник на выбранных точках. Точки будут соединены в том порядке, в котором они были выбраны. Доступна, если выбрано по крайней мере три точки и они не образуют уже существующий многоугольник.
* Разбиение ребра - разбивает ребро на два, добавляя точку в середине выбранного ребра. Доступна, если выбрано две точки, соединенные ребром.

Каждый многоугольник имеет имя и цвет (в редакторе). При создании многоугольники автоматически нумеруются, начиная с нуля, и получают дефолтовые цвета:

  
**Рис. ... Color-code в Properties Editor'е.**

К каждому многоугольнику из полигональной зоны есть доступ из скрипта по скриптовому имени зоны и имени многоугольника. Можно получить список объектов, попадающих в многоугольник, и менять родную землю в пределах многоугольника.

## **Дополнительные операции с террейном**

**TODO**

Чинить стыки террейна (Weld terrain vertices)  
Выровнять вершины террейна (AlignTCV)  
Дальше не придумали, импровизируй

# **Редактирование форм**

В редакторе можно создавать и редактировать формы пользовательского интерфейса в режиме WYSIWYG.

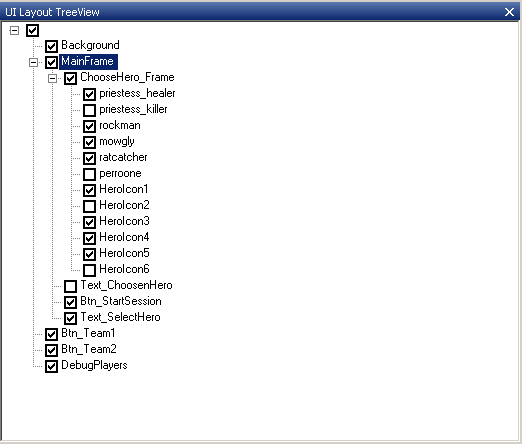
Формы и прочие элементы UI (User Interface) располагаются по пути **Root/UI**. Здесь имеются текстуры, курсоры, материалы, стили шрифтов и экраны.

## **Редактор форм**

Сейчас нас интересуют именно экраны. Открываем, например, **Root/UI/Screens/Lobby\_Main**:

  
**Рис. ... Редактирование экрана.**

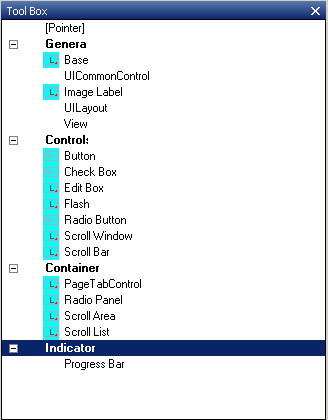
Здесь можно выделять мышкой контролы, таскать их, менять размеры, настраивать их свойства в Properties Editor, и, проделав всё это, удалять. Список контролов, расположенных на текущем экране, показывается в **UI Layout TreeView** - эта панель открывается из меню **View**. В ней контролы показаны в виде дерева, причём можно контролы показывать/скрывать:

  
**Рис. ... UI Layout TreeView.**

Выделение контролов синхронно показывается и в редакторе форм и в дереве. Через popup-меню можно указать, какой именно контрол вы имеете в виду (возможны варианты из-за вложенных панелей-контейнеров).

|  | **Множественное выделение**  Не забываем про такую полезную вещь, как выделение мышой сразу кучи объектов. |
| --- | --- |

Чтобы поставить контрол на экран, нужна ещё одна панель - **Tool Box**:

  
**Рис. ... Тулбокс.**

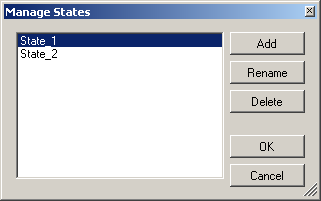
Выделяем в тулбоксе контрол (например, кнопку - **Button**) и рисуем на экране область. Контрол будет располагаться именно в этой области. Правда, отрисовывается при этом всего лишь граница контрола, чтобы кнопка показывалась как кнопка, надо руками настроить ей текстуры и прочие параметры в Properties Editor.

Можно добавлять в тулбокс экраны (объекты UILayout) - открыть экран, и по контекстному меню в тулбоксе выбрать **Add Item From Editor**. Экран появится в разделе **User controls**. Затем эти экраны можно размещать на других экранах, так же, как и прочие кнопки и чекбоксы - кликом в тулбоксе и выделением области на UI-редакторе, куда поставить.

|  | **Копирование**  Ещё очень полезная штука - копировать объекты (**Ctrl+C** и **Ctrl+V**). Удалять объекты можно через **Ctrl+X**. |
| --- | --- |

## **Состояния**

Состояния (States) - механизм, интегрированный в ресурсную систему. Он позволяет держать наборы данных с наследованием объектов по данным. Применяется это в как раз в UI. Можно настраивать состояния в менеджере состояний: создавать, переименовывать, удалять:

  
**Рис. ... Редактирование состояний.**

Текущее состояние выбирается в редакторе форм выпадающим списком. Из него же вызывается менеджер состояний.

Механизм состояний устроен следующим образом. Есть предок состояний - **Default State**. Остальные - конечные - состояния наследуются от него. Например, мы в менеджере создадим два новых состояния **State\_1** и **State\_2**. Все эти три состояния (включая предка) редактируются отдельно. Но редактирование одного конечного состояния не отражается на другом: **State\_1** и **State\_2** никак не могут друг на друга влиять. А редактирование **Default State** отражается на всех предках.

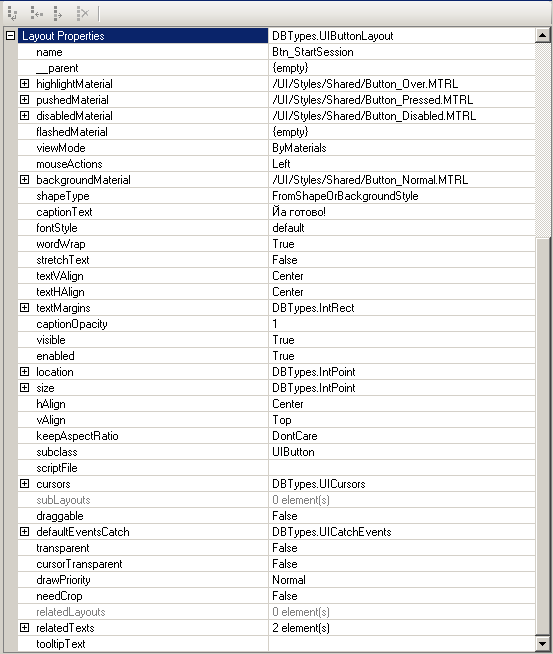
Очень важно понять следующее. Если какой-то параметр (например, положение некоторой кнопки) в конечном состоянии **State\_1** изменён, то этот параметр уже не будет наследоваться от **Default State**, он заживёт своей жизнью. Остальные параметры, которых в **State\_1** мы не меняли, будут автоматом следовать за изменениями в **Default State**.

## **Редактирование свойств контролов**

Хотя можно в редакторе форм мышкой визуально ставить и перетаскивать объекты, но основная работа по настройке ведётся по традиции в редакторе свойств:

  
**Рис. ... Редактирование свойств контрола.**

Все объекты, которые можно редактровать на формах, имеют один и тот же тип **LayoutPropertiesEditorObject**. Различие у них в полях **Type** и **Layout Properties**. У каждого типа контролов поля Layout Properties имеют свою структуру. Вот пример стурктуры **UIButtonLayout** (для кнопок):

  
**Рис. ... Редактирование свойства Layout Properties.**

Их бы надо, по идее, описать. Хотя, с другой стороны...

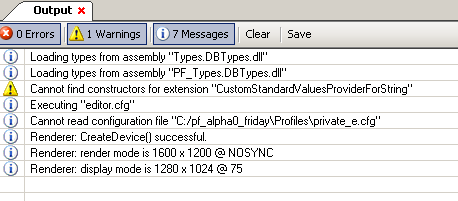
*Много задач в Джире  
Ещё больше звёзд на небе  
Время идти домой*

# **Настройка и прочее**

**TODO:** настройки видео, настройки клавиатуры,

## **Output**

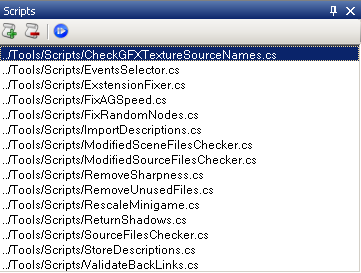
Редактор выводит в окно **Output** всё, что считает нужным сказать. Эти сообщения бывают трёх типов: **Message**, **Warning**, **Error** - в зависимости от серьёзности произошедшего.



Окно Output позволяет фильтровать сообщения по их типу, очищать список сообщений, сохранять его в текстовом файле, а также копировать в буфер обмена выделенные (**Ctrl+C**) или все (**Ctrl+A**) сообщения.

## **Скрипты**

Скрипты - это программы на C#, которые можно вызывать из редактора. Список скриптов, о которых знает редактор, виден в панели **Scripts**.



В этот список можно добавлять новые скрипты (для этого надо уметь их писать), можно удалять скрипты из списка, и можно их запускать. Это предполагает, что вы знаете, что делает тот или иной скрипт.

Примеры скриптов:

**ImportDescriptions.cs** - загружает из Excel-файла описания полей ресурсов  
**StoreDescriptions.cs** - сохраняет описания полей ресурсов в Excel-файл

Скрипты лежат в папке **/Tools/**.

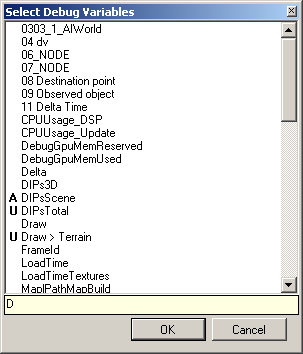
См. также: [Скрипты для редактора](https://confluence.nivalnetwork.com/pages/viewpage.action?pageId=83657551).

## **Debug Variables**

Что такое панель **Debug Variables** и для чего она нужна? Это возможность подглядывать за внутренностями редактора и параметрами его работы:



Как их создать, и что они означают, знают только программисты. А вот выбор дебаг-переменных для показа в редакторе настраивается из такого диалога (вызывается кликом на ссылке **Configure...**):



Для того, чтобы отметить переменную для показа, надо дабл-кликнуть по ней. При этом первый дабл-клик приводит к отметке **U** перед переменной - это значит, что мы изменили пользовательскую настройку только для нас (для текущего пользователя редактора), и она не отразится на отображении дебаг-переменных у других пользователей при коммите и апдейте через SVN.

Второй дабл-клик приводит к установке значка **A** - эта настройка будет отражена в файле **Profiles\PFEditor.config**, и пойдёт в SVN при коммите, т.е. эта переменная будет видна всем пользователям редактора. Третий дабл-клик приводит к снятию значка - поле не будет отображаться.

Нижнее поле ввода в панели выбора переменных - фильтр. Фильтрует по вхождению подстроки в любом месте названия.