

## **Знакомство с blender и пример построения модели.**

Анимация превратилась из сложного дела с целлулоидом в сложное дело с компьютером. Анимация – это оживление, в большой степени создание движений. Компьютерные программы позволяют создавать мультфильмы и видео которые являются результатом анимации. Эти же программы являются рабочей средой для построения объектов которые анимируются.

Если вы этим заинтересовались и считаете что вам это нужно то этот текст для вас. Этот текст рассчитан на читателей которым не надо объяснять что такое компьютер и не надо учить как им пользоваться. Раз это так то все вопросы стандартной эксплуатации операционной системы и подобные будут опущены. Без овладения этими приемами доведённого до лёгкости и автоматизма нельзя и говорить об освоении какой-либо – а особенно сложной специализации, какой является анимация. Те кто вроде бы довольно хорошо владеют основами но хотят в это убедиться могут посмотреть сайт [rsmmanual.narod.ru](http://rsmmanual.narod.ru). Наряду с программированием, моделированием анимация относится к наиболее сложным видам работ – компьютерных или нет. Если вы решили сделать интересную анимацию за ближайшее время или стать великим аниматором за полгода выбросите этот текст и забудьте о нём. Правильное отношение к любому профессиональному, да и вообще сложному знанию в том чтобы уйти от азарта.

Несколько слов о требованиях к аниматору. Пространственное мышление важно, но по мнению автора ещё важнее. наблюдательность и элементарное умение рисовать. умение рисовать – почти необходимое умение потому что без него всё что создано будет казаться замечательным. придирчивость которая вырабатывается у мастера рисунка нужна для того чтобы заметить ошибки. поэтому каждый кто собирается более-менее масштабно править импортированную модель должен уметь рисовать как можно лучше. наблюдательность нужна для того чтобы видеть детали и процесс движения. если вы попросите

человека с улицы нарисовать человеческую фигуру он нарисует её стоящей по стойке «смирно». но это не годится – особенно в анимации. если в рисовании вы можете пойти на идеализацию то в анимации – нет. совершенно неочевидно но факт что человек никогда не бывает в таком положении. как бы не были примитивны ваши персонажи наблюдение за людьми и предметами должно стать вашей привычкой. это вопрос о наблюдении вообще, о сосредоточении фактически.

С другой стороны этот текст рассчитан на тех кто не знает что такое анимация и построения. Поэтому те кто уже ориентируются в этом найдут здесь то что им уже известно. Может быть они не найдут того что им нужно.

Автор считает что лучший способ излагать – это делать это 1)когда автор уже мастер, 2)когда автор учится ненамного обгоняя читателя, поэтому я могу выбрать не самый эффективный путь в примерах. Я могу блуждать но собираюсь довести всё до конца. Сейчас я объясню это мнение. Анимация и масса других работ требуют профессионального знания. Она формирует человека с подготовкой которая далеко превосходит по сложности обычную подготовку пользователя или ту которая нужна для простых дел. Но при этом специалист теряет взгляд ученика полностью уходя в специальную терминологию и концепцию. поэтому ему трудно объяснить как надо действовать. ему также трудно объяснить почему надо делать так или иначе. его мышление вертится вокруг специальных понятий которые могут не быть простыми синонимами обиходных понятий или простыми в определении. в отдельных случаях специальные понятия – я не говорю об анимации – вообще постигаются интуитивно. формальный подход при определении не даёт приемлемого результата. я хочу обратить внимание читателя на то что каждый из этих подходов к обучению не идеален. автор может не просто ошибаться но и не оптимально организовать текст. впрочем это проблема изложения в любом случае.

хороший текст написан так чтобы дать человеку легко усвояемый материал. практическая польза от чтения требует того что расходится с логикой и теорией. поэтому есть разные

подходы и нет идеального подхода.

моя практика показала что моделирование это работа которая требует в большей степени чем другие отказа от формальной методологии. особенность её в том что надо иметь в виду все аспекты практически в равной степени. это не означает отказа от порядка и чёткого мышления. подмена этого требования снижением уровня «планки» в отношении порядка ведёт к хаосу и практически бесполезной работе. разнообразие способов получения схожего результата, сложность, расхождение с обычным темпераментом являются спецификой этой работы. когда возникает анимационная или инженерная задача аниматор должен прежде всего думать о средствах и целях а не о действиях в конкретно-методическом смысле. свобода в выборе средств не только является признаком профессионала но и крайне нужным свойством любого ученика. структурирование и планирование работы существует при том что её цель может меняться по мере работы. только при наличии плана можно работать.

в условиях коммерческого производства анимирование должно выполняться с учётом того что анимационные эффекты и кадры могут быть выполнены не только разными путями но и с помощью некомпьютерных методов. пока кино и мультипликация выполнялись традиционно многие эффекты были невозможны, но многие и достигались гораздо легче и быстрее. монтаж, видеообработка, коллаж конкурирует с компьютерной анимацией и может с ней сочетаться. результат анимации может потребовать послеобработки как видеоматериал. в том числе она может содержать искусственное приближение к обычному видео и кино. если этого не сделать то анимация будет слишком непохожа на действительность.

анимация и построение – это два занятия, две темы в которые будет аккумулироваться любое знание по этой теме. построение – это процесс при котором создаётся анимационная сцена. сцена – это пространство с объектами которое будет использовано для анимации. понятие сцены прямо эксплуатируется в анимационных программах но в этом тексте оно может иметь более

теоретический смысл. результат этой работы будет анимирован. это значит что в зависимости от заданных параметров которые сами не являются объектами сцены будут происходить изменения. анимация – это процесс изменения самых разных свойств сцены и элементов сцены. даже перемещение камеры является анимацией. неизменная, «немая» сцена тоже делается как часть анимации. относительно отдельного состояния и серии кадров выполняется render, то есть прорисовка. в результате прорисовки возникает отдельное изображение или совокупность изображений, или видеоролик. таким образом работа в общем состоит в том чтобы создать или импортировать сцену и определить что и как будет меняться со временем, и что и как будет прорисовано. в анимационных программах можно ограничиться созданием сцен или отдельных объектов. некоторые программы вообще не предусматривают анимации но являются хорошим полигоном для построений. с другой стороны есть отдельные программы для render'a, в которых нет возможности делать что-либо кроме прорисовки готовых сцен. есть программы для вывода отдельных статичных сцен. с теорией анимации познакомиться полезно но её здесь не будет. в руководствах к анимационным программам она более-менее изложена.

## **Блендер.**

в этом тексте будет рассмотрено использование блендер. блендер – это приложение Windows и других операционных систем. это комплексное приложение для выполнения всех этапов работы над анимацией, включая видеоредакцию. блендер – бесплатная программа. блендер снабжается несколькими скриптами которые не требуют оплаты. с другой стороны есть платные плагины и скрипты которые не могут быть полностью учтены и рассмотрены в этом тексте. на фоне 20–30 анимационных приложений присутствующих на рынке блендер выглядит довольно оригинально. сочетание бесплатности и хорошим набором возможностей делает его исключением среди приложений Windows. под управлением других систем есть

подобные программы но обычному пользователю может быть трудно адаптироваться к непривычной среде.

блендер является продуктом с оригинальной историей которая важна практически потому что показывает почему он так специфичен и требует усидчивости. он вырос из внутреннего ПО анимационной фирмы NaN, и естественно что несёт в себе элементы неожиданные и трудные в понимании для ученика, обратная сторона этого факта в том что вы получаете профессиональный инструмент позволяющий регулировать то что нельзя отрегулировать в других программах. тем не менее у автора осталось впечатление что программа создана с хорошим взглядом, упрощением сложных вопросов. дело в том что подход к созданию программы капитально влияет на неё, а профессиональные свойства не означают её недостатков а просто являются следствием большего знания. ни одна подобная программа не легка для освоения. все они требуют «высисживания» даже простого результата. и так и будет пока техника манекенов не станет занимать львиную долю анимации.

блендер – быстро развивающийся проект. поэтому версия имеет значение. автор пользуется версией 2.41. по мере написания этого текста могут быть выпущены новые версии. Сейчас уже есть версия 2.43 и она содержит несколько прогрессивных изменений по сравнению с 2.41. автор видимо постарается отделить в тексте описания разных версий. Хотя они похожи и обе имеют небольшой объём, 2.41 – менее 8 мб в дистрибутиве.

блендер имеет англоязычное руководство которое на момент написания этих строк не переведено. оно стоит 36 евро и распространяется на CD. кроме того есть статьи и самовольно написанное пользователями руководство но его недостаток в том что оно не показывает на этот момент работы в системном и глубоком виде. есть русский перевод этого текста и оглавление. я не советую использовать примеры этого самостийного пособия так как они написаны явно в перегретом состоянии и требуют большой подготовки. к тому же они создают неправильное отношение к построению как к лёгкому, развлекательному делу. Я вижу элементы дяляческого отношения, неподготовленности в

этом руководстве.

блендер отчасти русифицирован. но как всегда русификация происходит на пользу пониманию слов которые не точно отображают смысл являясь переводом, и во вред использованию непереверённого руководства. небольшой файл формата pdf который поставляется с программой не является пособием, а отпиской где изложены основные приёмы. блендер гораздо более мощное приложение чем может показаться судя по этому документу. Проблемы с адаптацией к русской среде проявляются в том числе в том что при попытке открыть каталог с кириллическим названием программа может быть выключена операционной системой. Выдаётся сообщение о недопустимой операции. *Поэтому осматривая графические файлы через блендер надо избегать таких каталогов. Возможно это относится и к другим ситуациям.*

в этом тексте я не стану излагать так чтобы руководство было не нужно. с другой стороны я постараюсь писать так чтобы ключевые вопросы имели хотя бы упоминание. я обращаю ещё большее внимание на трудности и «белые пятна» но лишь те которые видны мне. есть люди знающие анимацию и блендер лучше автора. я рассчитываю на то что читатель сам прочтёт все надписи в окне программы, я 1)даю указание на то что есть возможность 2)показываю важные моменты любого рода которые мне покажутся заслуживающими этого 3)обозначаю подводные камни.

прежде чем писать о том что делать – что, похоже, является дурной склонностью отечественных авторов, я в интересах читателя должен сделать объяснения которые имеют касательство и к блендер, и к другим приложениям. я постараюсь вначале опираться на представление новичка которое попробую в себе создать и вспомнить. анимация и построение имеют конкретные формы которые будут меняться. на сегодня они таковы что есть большое отличие между обывательским представлением и реальностью работы аниматора. такое же отличие есть даже при создании статичных сцен. анимация не есть работа над объектами с интеллектуальной программой. поэтому отдельные простые операции а не эффективное и быстрое создание форм типично.

анимация обычно не есть работа с точками как её представляют невежды. построение тоже обычно не занимается точками. построение – это сложный процесс в котором форма может быть создана по мере анимации. форма может иметь технический, иметь служебный смысл в работе, а не быть объектом. используются объекты разного типа, и тип ограничивает возможности редакции объекта и связанные с ним действия. поэтому важно продумать работу. технология, объекты и результаты находятся в сложных отношениях. есть довольно стандартные приёмы которые надо применять к своему представлению о цели, иначе нельзя работать вообще или работать эффективно. за типами объектов стоит теория которую знать не надо, но она ставит ограничения и задаёт нагрузку на технику при использовании объекта. в каждой программе есть свои особенности которые делают выбор программы важным даже в плане общего подхода но в целом это верно. есть приложение Atari которое позволяет лепить объект как кусок глины но автор обнаружил что правильно применённые стандартные приемы более эффективны в большинстве случаев. не надо идти на поводу у дилетантизма. правильное отношение в том чтобы учиться выбрав хорошее приложение.

различают твёрдотельное моделирование и другие. оно есть построение без учёта того что объект имеет способность деформироваться. анимация есть процесс задания ситуаций и параметров исходя из которых происходит расчёт положений и свойств. была проделана большая работа для того чтобы модель вела себя подобно реальности но обычная теоретическая модель оказывается непригодной в некоторых случаях. для современной технологии «передним краем» является возможность моделирования жидкостей, тканей и волос, пламени и снега (систем частиц). все эти объекты отстоят от простой модели потому что есть сложное движение и\или сложный объект который нельзя рассчитывать обычным путём. поэтому результаты моделирования этих объектов могут быть хуже чем в других случаях. сама возможность сделать это не гарантирована для конкретной программы. возможно, отдельные модули к

приложению могут улучшить ситуацию. они могут быть не бесплатными. блендер рассматриваемой версии непосредственно может создавать анимацию моделей упругих тел (soft bodies) и иллюзию жидкостного поведения тела. возможно что есть или будут некоторые способы с помощью плагинов или иначе улучшить его работу в этом и позволить анимацию тканей. ткани имеют значение для реалистического вида моделей претендующих на сходство с действительностью. блендер также имеет встроенную систему анимации частиц (participles).

результат render'a будет зависеть от качества приложения, его параметров, и использования таких свойств как размытие движением (motion blur), radiosity, и других. radiosity – это технически (программно, не аппаратно) обеспеченный метод отражения лучей света аналогичный естественному. в реальности лучи отражаются создавая сложную картину световых, в том числе цветowych рефлексов которую человек может не замечать саму по себе, но он скорее всего заметит разницу между результатами прорисовки таким и обычным способом. блендер имеет отключаемый параметр для сложного просчёта рефлексов о котором идёт речь. этот метод требует дольшего времени просчёта кадра.

блендер включает механизм создания игр но он видимо не будет здесь разобран.

блендер – приложение одного файла. вы не можете открыть в нём более одного файла в один момент. блендер работает с файлами собственного формата blend и позволяет импорт из трёхмерных форматов dxf, vrmf и других. это происходит с большими или меньшими потерями, или без потерь. поэтому надо посмотреть нет ли стороннего приложения делающего это лучше. Если вы установили версию 2.41 то файлы формата blend не ассоциированы с программой поэтому они не будут иметь значков в Проводнике. В списке недавно открытых документов в меню кнопки Пуск вы тоже их не найдёте.

блендер может импортировать так называемые пути (paths) из файлов форматов ai, eps, ps. path – это замкнутая линия одни участки которой могут быть прямыми, а другие кривыми и которая



создаётся вручную во многих графических программах для ограничения действия инструментов графических программ. пути в блендер при импорте превращаются в объект типа mesh (меш).

блендер может брать части содержимого файлов своего формата чтобы прибавить их к открытой сцене (меню file-append). вы можете импортировать объекты и их свойства вместе с ними.

с помощью объекта text вы можете импортировать как символы произвольные плоские фигуры если они существуют как символы шрифта. то есть созданный оригинальный шрифт позволяет быстро вставлять в сцену тела которые в шрифте считаются символами. в блендер они будут преобразованы в тела.

настройки блендер хранятся в большой степени в файлах своего формата. параметры вида окна приложения могут быть загружены или нет из такого файла смотря по тому нажата ли кнопка LOAD UI в момент открывания файла. с другой стороны есть параметры регулируемые в интерфейсе как настройки в обычных программах. они включают русификацию кнопок, всплывающих подсказок и меню которые регулируются отдельно. это позволяет получать русскоязычные подсказки при правильных надписях на меню и кнопках. там же регулируется масса важных свойств, в том числе свойства копирования объектов. это значит что набор свойств копируемых вместе с объектом может быть настроен. это практически важно. ниже вы узнаете где именно настраивать приложение. чтобы изменить настройки работы вообще надо сохранить их в файле открываемом по умолчанию. для этого надо создать нужную сцену, настроить окна и сохранить этот файл: file-save default settings. настройки определяются открытым файлом.

блендер поддерживает компрессию файлов. использование пункта меню file-compres file вызвало сжатие от 470 до 270 кб. это может быть не важно в плане объёма тк можно вместо этого архивировать файл.

файлы в блендер автосохраняются через каждые 5 минут но при выключении нет запроса на сохранение. есть восстановление открытого прежде файла. он может быть стёрт при переустановке

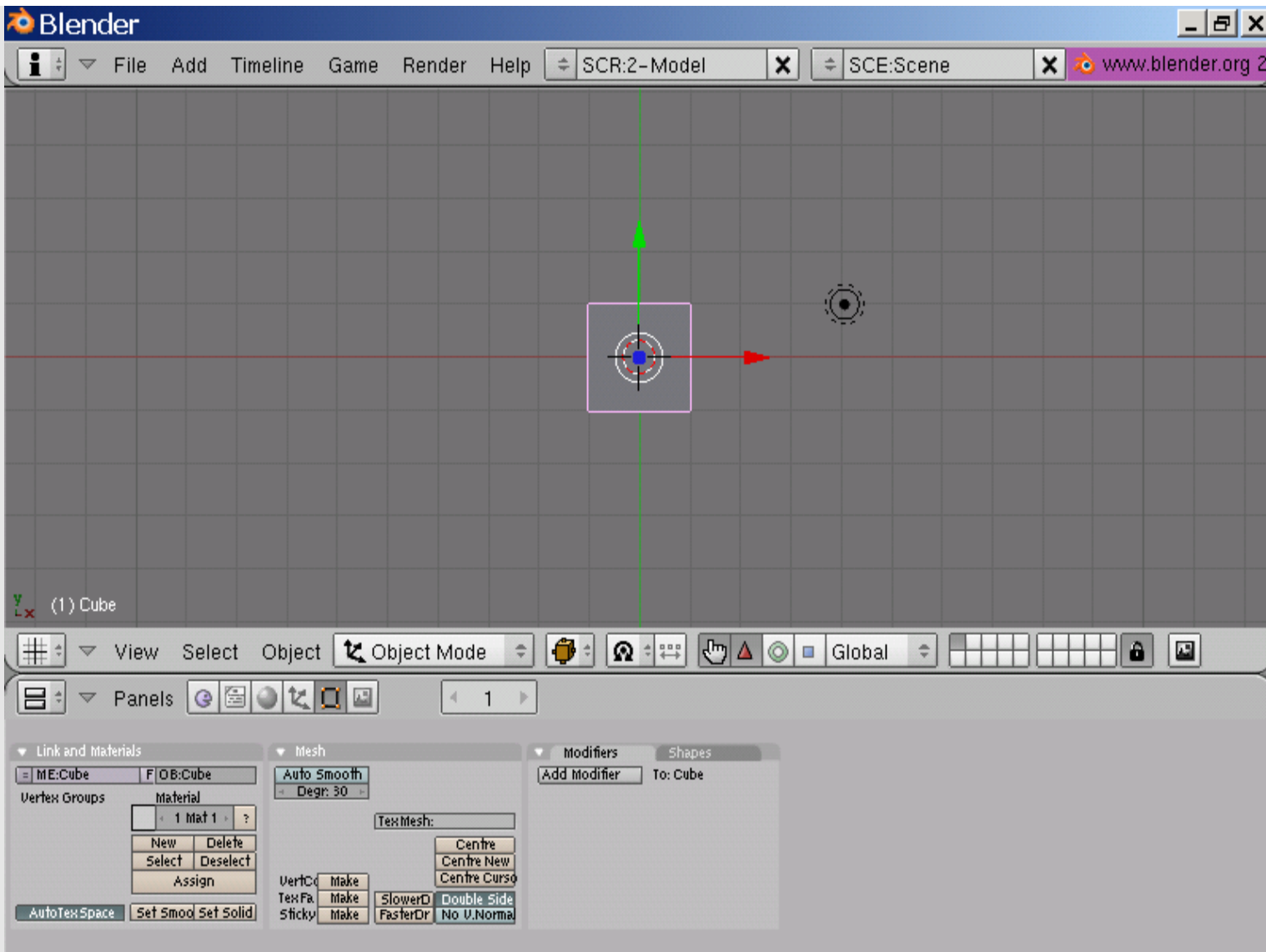
приложения или программ в целом. Всегда сохраняйте файл прежде чем выключить приложение.

в блендер есть понятия сцены и слоя. это разные вещи. сцена может иметь несколько слоёв. слой не есть нечто расположенное глубже или выше. это часть сцены. в одном файле может быть более одной сцены. связь между окнами, свойствами, объектами, слоями может быть разной. в первый момент важно не пугаться исчезновения объектов которые в невидимом слое. в блендер может быть несколько сцен и слоев (layer).

## **Первый запуск и обзор.**

блендер обладает нестандартным интерфейсом. поэтому на него нельзя переносить приемы и ожидания ставшие привычными для владельца Windows. а именно: меню вызывается нажатием на каждый пункт, оно не вызывается нажатием на alt;диалоговые окна исчезают сами если ими не пользоваться;внутри окна блендер есть несколько окон которые не накладываются между собой и не имеют стандартной рамки;эти окна имеют отключаемые заголовки которые могут быть расположены внизу;не всегда возможно убрать окно не убирая другое окно;для убирания и деления окна, а также регуляции заголовка надо нажать правую кнопку мыши на границе окна;одно окно может иметь разное содержимое в разное время, оно регулируется кнопкой в заголовке;содержимое заголовка может быть сокращено нажатием стрелки на нём, при этом исчезает меню;заголовки содержат кнопки и другие знакомые инструменты; нормальный режим работы – сокращённое окно, в развёрнутом виде оно равно имеет размер дисплея. когда я буду писать о меню то имея в виду меню конкретного заголовка окна в котром вы работаете. это окно в котором курсор. если встроенное окно открыто на всё внешнее окно то другие окна не имеют некоторых частей заголовка и их граница не может быть перемещена. *надо знать нестандартный интерфейс.*

после запуска окно блендер с настройкой по умолчанию выглядит так.

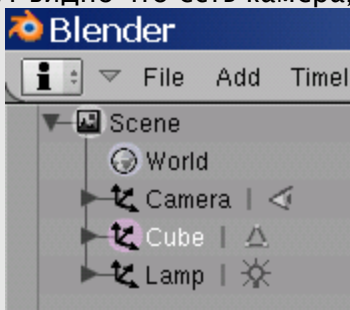


на этой иллюстрации окно раскрыто на весь экран при разрешении VGA. если у вас установлено другое разрешение то

кнопки и меню будут мельче. здесь видно 3 окна. говорить об их функциях можно только имея виду состояние регулируемое кнопкой на заголовке. с этой оговоркой можно считать что вверху виден заголовок окна настроек которое сведено к минимуму, заголовок нельзя уменьшить уменьшая окно. ниже самое большое окно построения. ниже окно кнопок.

для анимационных приложений типично что даже пустая сцена содержит площадку, вы можете включить её вертикальную ось и выключить её для большинства видов. она позволяет ориентироваться. в данном случае она показана лицом к вам поэтому выглядит как школьная тетрадка. каждое приложение может иметь границу построений. за ней ничего не будет показано. но граница выходит далеко за пределы площадки. объект может присутствовать в сцене но быть за пределами зоны отображения. площадка – это центр построения, её клетки имеют размер равный единице (по умолчанию), все тела имеют численные координаты.

с помощью кнопки с изображением решетки переключитесь на outliner. это пассивный обозреватель содержимого сцены. в нем показаны её объекты и подобъекты. теперь переключитесь в режим дерева: view-show outliner. теперь вы видите неразвернутое дерево сцены. щёлкая стрелки вы можете развернуть её объекты. тут видно что есть камера, куб и лампа. куб выделен.



вернитесь обратно в режим построения. куб выделен. теперь для большего драматизма мы посмотрим на сцену с другой точки зрения. выберите view-camera. теперь вы видите 1)содержимое сцены 2)границы кадра 3)площадку искажённую перспективой.

*outliner* позволяет найти и выделить объект.

все анимационные программы показывают сцену либо в перспективном либо в ортогографическом режиме. я сомневаюсь что можно показать объект одним способом а всё остальное иначе. ортогографический режим – это режим без перспективного искажения. он хорош тем что позволяет лучше оценить геометрическую сторону построения и в некоторых случаях быстрее отображается. в блендер он выглядит как режим с обратной перспективой. поэтому надо корректировать своё впечатление. перспективное искажение может быть чрезмерным, оно регулируется свойствами объектива. режим регулируется через меню *view*. в этом меню есть строка *user* которая может быть использована для возврата к предыдущему виду отличному от «строго сбоку» (*side*) и других строгих видов. *вы можете установить точный вид через меню.*

теперь левой кнопкой мыши оттащите вниз границу окон чтобы скрыть окно кнопок. останется только его заголовок. откройте меню *view*. теперь вы можете видеть его целиком. чтобы видеть меню целиком может потребоваться расширение окна.

в этом меню вы можете отрегулировать

- угол зрения,
- режим показа,
- точку зрения;
- максимизировать окно;
- вывести свойства показа,
- изображения подкладки;
- отключить другие объекты кроме выделенного.

подкладка – это графический файл который виден только в ортогографическом режиме и в «строгих» видах: *front*, *top*, *side*, *camera* .

режим *local view* отключает показ всего лишнего что полезно при сложных сценах. используйте *локальный показ если вам мешают другие объекты.*

теперь вы наконец получите возможность поворачивать что-то. введите курсор на окно построения и нажимая среднюю кнопку мыши или колесо мыши проташите курсор. если вы не отпустили

палец то объекты и площадка вращались. теперь кликните левую кнопку мыши и двигайте мышь, кликните ещё раз. выделенный кубик или что там у вас движется. теперь зажмите shift и кликайте по чему-нибудь отличному от площадки. Последний выделенный таким образом объект имеет более яркий цвет. Блендер позволяет без переключения режимов вращать, менять точку зрения и выделять; действия для этого нестандартны но просты.

В середине окна построения есть кружок напоминающий спасательный круг. Это курсор построения. На его место будет установлен новый объект. Вверху окна есть меню. Откройте пункт Add. Тут вы видите список типов объектов и примитивы которые можно вставить в сцену. Вставьте Monkey. Это готовая модель головы обезьяны. теперь вытащите её чтобы увидеть лучше, и приблизьте объект. вы видите первый сложный объект в блендер. через меню вы можете добавлять простые фигуры, они попадают на место курсора построения.

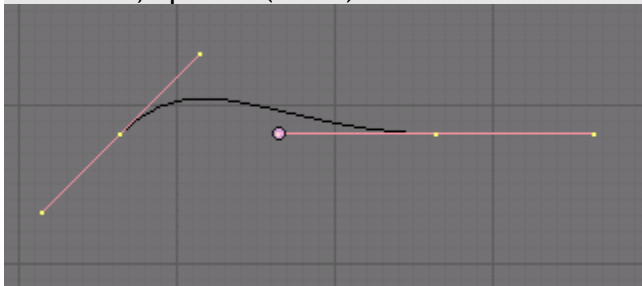
Построение – это процесс создания сцены, он складывается из действий с объектами и редакции объектов. Блендер имеет простое разделение всех работ на действия с объектами без их изменения и действия с редакцией объектов. Переход в другой из этих режимов выполняется нажатием Tab. Когда вы вставляете объект то он переходит в режим редакции. если вы хотите вывести его в режим объектов надо нажать Tab. Вид объекта при этом меняется. Если вы хотите действовать с объектом в целом то надо либо действовать с ним в режиме объектов либо работать со всеми его частями в режиме редакции. *Нажатием Tab вы переносите Блендер в другой режим, в котором у вас появляются новые возможности.*

## **Объекты блендер.**

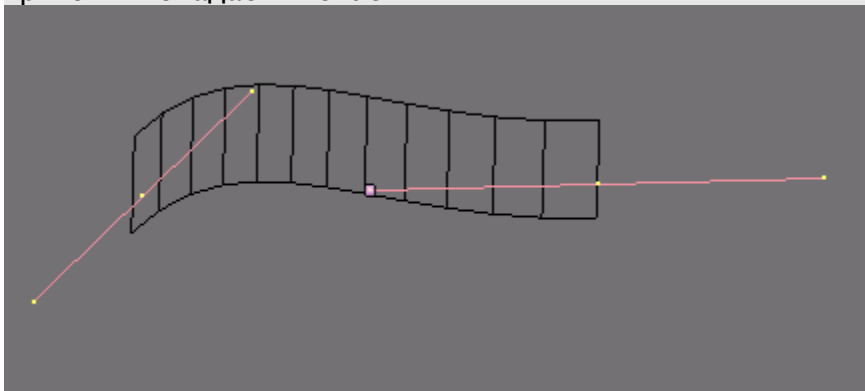
Теперь когда вы убедились что блендер не мёртвая материя и может что-то делать надо обратить внимание на типы объектов. Как вы знаете тип объекта важен потому что он имеет ограничения в плане операций. Блендер работает с разными типами объектов, они различаются своими геометрическими и другими свойствами.

В других программах те же или очень похожие в плане приёмов эксплуатации объекты могут называться иначе но в целом типы почти одинаковы. Мы рассматриваем блендер. Вот его типы объектов. Они выделены не только по теоретическим отличиям и показаны в режиме редакции, как выглядят сразу после вставления.

–линии, кривые (curve).



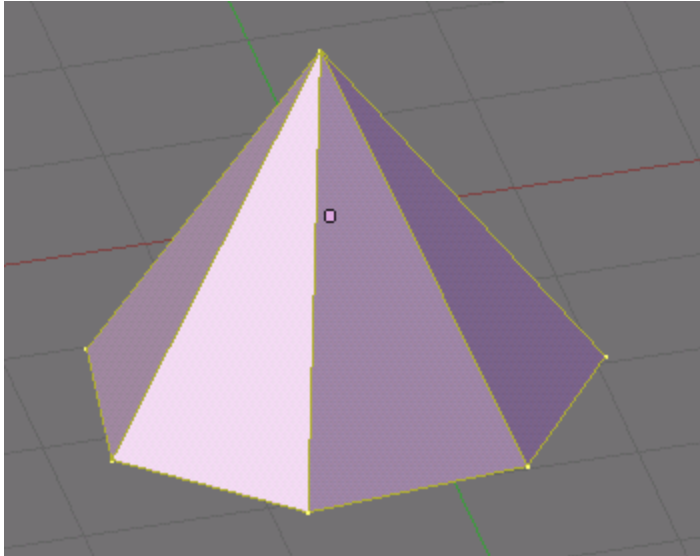
Линии не являются объектами попадающими в render. Это бестелесные «волосы» но они могут иметь привязанную поверхность которая на самом деле является производной от кривой и попадает в render.



Линии имеют значение как основа для дальнейшего построения. я могу называть их кривыми. Слово curve переводится как кривая. Кривая состоит как минимум из одной точки.

–меш (mesh), или иначе полигоны.

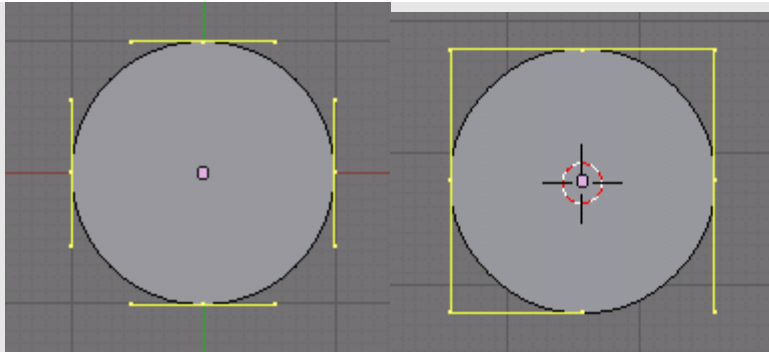
меш – это объект который по определению состоит из точек и



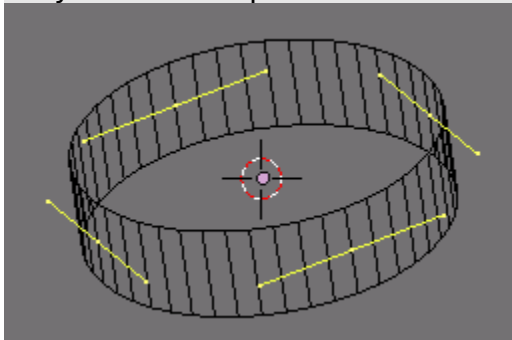
рёбер. Я буду называть их иногда линиями. По контексту будет ясно что это не объекты-кривые. меш отличен от кривой тем что он угловат хотя может применяться для создания гладких моделей. В частном случае меш состоит из одной точки. Меш - самый распространённый тип объекта. Показан меш у которого есть плоскости. Когда речь шла о кривых то там плоскостей не было. Меш может не иметь плоскостей (face, грани). В таком случае он и выглядит иначе если режим показа вообще позволяет вывод граней. Он может не иметь ребер.

- окружности кривых Безье и NURBS.

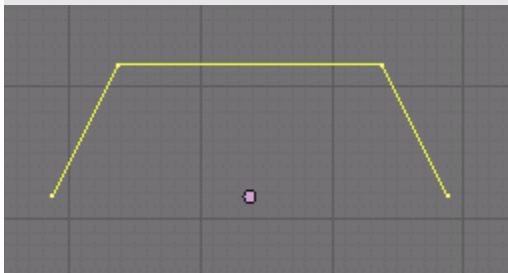




это безобъемные фигуры которые имеют заливку но гранями не обладают. Они также как кривые часто используются как основа. Они являются производными от кривых. Заливка определяется тем включён ли режим 3d для фигуры. Эти окружности тоже могут иметь поверхность.



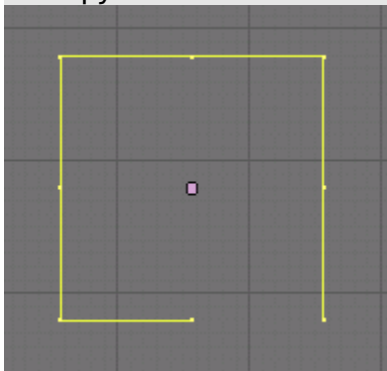
-кривые NURBS.



Это кривые с другой математической основой и другими

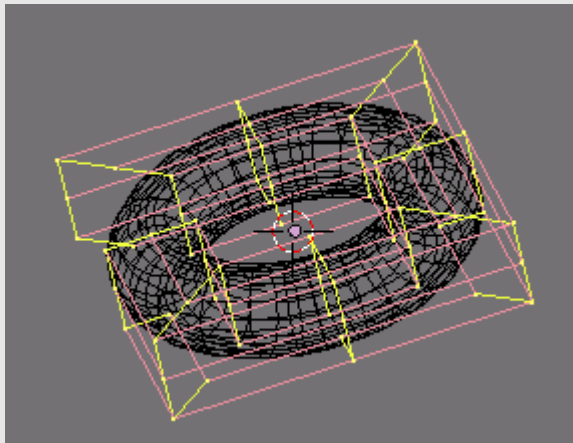
возможностями чем «обычные». Минимальное количество точек по которым строится кривая – три. Вывод кривой будет зависеть от её параметров, в отдельных случаях она не видна хотя есть её точки построения. Они также как кривые часто используются как основа.

–окружности NURBS.



Безобъёмные фигуры без заливки. Они также как кривые часто используются как основа. Они являются производными от кривых NURBS.

–объёмные тела NURBS. Производные от кривых этого типа. Несут на себе наследственные свойства кривых этого типа. Польза этих тел в том что они в отличие от мешей всегда гладкие.



-пути (path). Кривая NURBS подготовленная для создания путей перемещения при анимации объектов.

-метаболы (metabols).



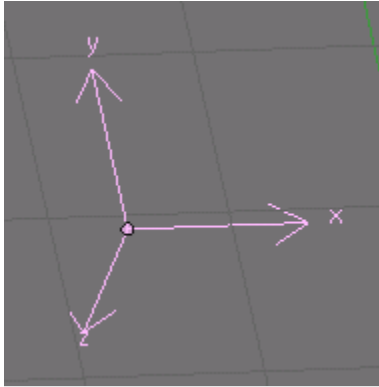
Округлые объемные фигуры с другой математической основой чем у мешей и NURBS. Возможности их редакции очень невелики но метаболы хороши для создания «переливающихся» форм, напоминающих оплавленные. Метаболы автоматически отталкивают или притягивают свои края к другим метаболом. Приближение одной метаболы к другой сразу вызывает этот эффект.

-text.

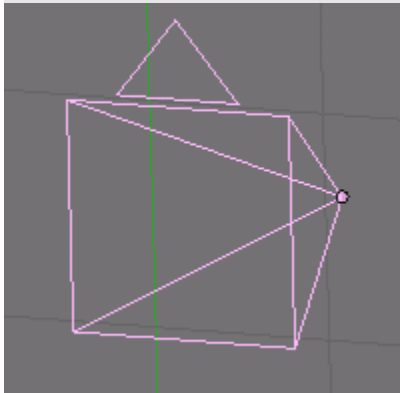


Объект для создания текста. Его смысл в том что не надо вручную делать сложные фигуры объемных букв. Есть взаимодействие со шрифтами доступными на вашем компьютере и поддержка кириллицы.

-empty.



Пустой объект не выводимый в render. Но полезный как технический. Не имеет режима редактирования.  
-camera.



Невидимый в прорисовке объект обладающий свойствами камеры. Камера является средством наблюдения наряду с простым обзором сцены при анимации или без анимации. Необходимый объект для получения результата но не обязательный пока сцена редактируется. Не имеет режима редактирования.

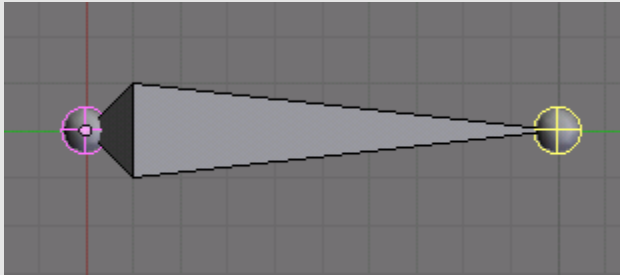
-lamp.

Невидимый в прорисовке объект обладающий свойствами источника света. Необходимый объект для получения результата но не обязательный пока сцена редактируется. Не имеет режима



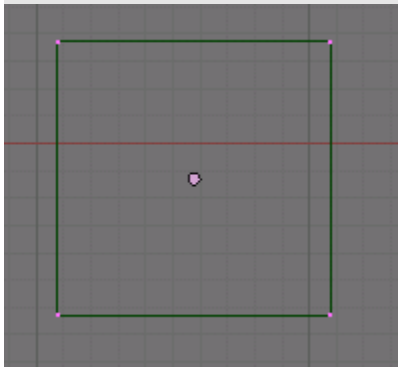
редакции.

-armature.



Технический объект для анимации объектов. В других приложениях он называется костью (bone).

-lattice.



Невидимый в render объект для деформации других объектов. Показан в режиме редакции. В простейшем виде имеет одну точку но тогда он бесполезен.

Блендер ещё имеет захваты, крючки (hook) которые не являются объектами но могут их напомнить внешне. Они очень похожи на пустой объект.

Кроме того блендер позволяет создавать привязанные поверхности кривых и мешей. Для мешей это поверхности

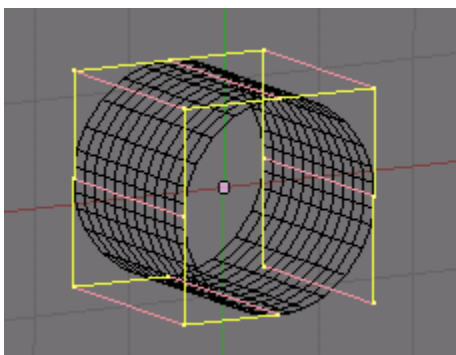
SubSurf. Привязанные поверхности похожи на самостоятельные тела и важно не перепутать их с телами в окне блендер. Они будут удалены вместе с их телами при удалении тел.

Вы столкнулись с несколькими математическими типами объектов. Это прежде всего кривые, меши, NURBS и метаболы. В этом перечислении показаны готовые конкретные тела потому что это ваш инструментарий. Пока это очень поверхностное определение объектов. Оно намеренно сделано таким. Зато сейчас вы знаете главное. Вы понимаете главную роль объектов. Вы знаете как они выглядят. *Блендер имеет 4 основных типа редактируемых тел с разными возможностями.*

## **Вид тел.**

Важно не запутаться потому что вид построения обманчив. Надо различать линии которые 1)блендер считает объектами типа curve 2)линии которые сами не являются объектами но показывают построение 3)окружности 4)части объемных тел которые похожи на линии. То что нечто выглядит линией не значит что является объектом типа curve. 5)пути. Различение объектов типа линии и других частей сцены не является сколько-нибудь проблематичным но ученику надо понимать эту разницу потому что иначе он может пытаться применить к линиям другого типа и нелиниям приёмы работы с линиями. То же верно с точками и объемными частями сцены. Вид и принадлежность типу не связаны как кажется на первый взгляд. Например, тело NURBS будет видно в окружении линий интерполяции. Эти линии выглядят как отрезки и содержат точки но фактически никаких отрезков и точек как частей построения нет, а есть только фантомные линии и точки для редакции тела.

Вид зависит от режима работы. Вы знаете что есть режимы редакции и объектов(тел). Кроме того блендер имеет несколько режимов показа. Они регулируются кнопкой на заголовке окна построения.



Режимы важны в первую очередь потому что позволяют вам  
1)видеть объект как реальное тело и оценивать результаты  
2)видеть объект как инженерную конструкцию и добираться до любой его части 3)видеть объект как пустой параллелепипед который практически не берёт вычислительную мощность 4)скрывать лишние, мешающие части построения. Для режима solid на заголовке окна построения есть кнопка выключающая вывод и выделение невидимых частей. Её нажатие не снимает выделение. Режим wireframe иногда называют проволочным. Он обычно используется для редакции и позволяет относительно быстро вращать сцену. Вид регулируется настройками.

Отдельное тело может быть скрыто, и часть точек тела может быть скрыта. Но это зависит от типа тела. Для отдельного тела можно установить особый режим показа. Так для него можно установить проволочный режим. *Режим и скрытие точек фатально влияет на вид тел.*

## **Трансформация тел. Отмена действий.**

Блендер позволяет работать с телами разного типа которые имеют разные возможности. Поэтому вы заинтересованы понимать какие трансформации типов могут быть осуществлены. Когда речь идёт о трансформации типов имеется в виду что отдельный объект может быть перестроен в подобный объект другого типа. При этом блендер старается сохранить геометрию трансформируемого тела. Но это не всегда возможно сделать

идеально. Это не значит что происходит настолько большое изменение что надо избегать трансформации из-за этого.

Есть важный аспект этого процесса: поскольку вы меняете тип тела его свойства могут быть автоматически отредактированы или утеряны.

Пользователь заинтересован мочь трансформировать все типы тел в любой тип. Но на сегодня блендер как и другие приложения не в состоянии делать преобразование всего во всё. Пустые объекты, камеры, лампы, арматуры не трансформируются. Остальные типы трансформируются в меши.

Для трансформации нужно чтобы объект не был в редакции.

История изменений для трансформированного объекта не обнуляется. Поэтому вы можете перейти к состояниям старого объекта.

Вы можете использовать трансформацию чтобы получить новые возможности, но потеряете старые свойства.

Для отмены есть меню истории действий вызываемое alt-u. Отдельное действие отменяется стандартным нажатием ctrl-z. Количество актов которые можно отменить установлено в настройках и равно по умолчанию 32, для версии 2.41 максимум равен 64. Можно ожидать что в одной из следующих версий ситуация улучшится. Есть приложения с бесконечной историей отмены. Отмена требует ощутимых объемов оперативной памяти поэтому в качестве альтернативы можно предложить создание объекта-копии или объекта-связанной-копии в момент к которому может потребоваться возврат. Хотя связывать это прямо с объемом ОЗУ видимо не надо тк Windows «сливает» её данные во временный файл. *Не рассчитывайте на бесконечную отмену.*

## **Обзор вопросов редакции тел.**

В общем все тела из которых вы реально можете что-то делать, которые имеют структуру являются телами трёх типов построения: 1)отрезочные 2)аппроксимированные или интерполированные 3)метаболы. Первый тип - это тела созданные из отрезков соединяющих точки. Это меши. Второй тип - это тела кривые



которых в общем случае не имеют узлов в точках а строятся по координатам двух и более точек. Кривые этих тел могут в отдельных случаях проходить через точки по которым строится тело. Но проходить не потому что точка сама принадлежит кривой. Метаболы – это тела с параметрическим определением диаметров поэтому они выделены в отдельный вид, хотя кривые тоже имеют параметры которые вы можете настраивать. Но у кривых это не настолько важно как их аппроксимация или интерполяция. Это классификация автора хотя она может совпадать с другой классификацией. Она имеет практическое значение потому что показывает чего следует ожидать от тел определённого типа и какую роль имеют в них точки. В случае мешей вы работаете с точками меша, в случае кривых и NURBS вы работаете с кривыми через точки построения которые не являются точками тела а лишь показаны как инструменты в режиме редакции. Эти тела сами не состоят из точек. Они редактируются не таким очевидным путём как меши.

Поскольку тела 2 типа строятся более чем по одной точке удаление и смещение соседней точки закономерно влияют на положение кривых несмотря на то что соседняя точка сама не была перемещена. Это нормально.

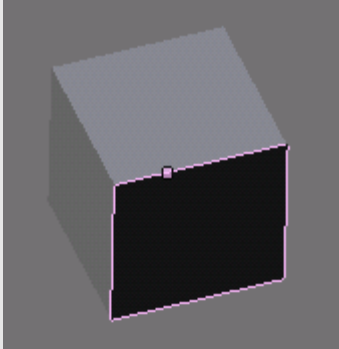
В блендер объект имеет массив данных ObData но это отдельные вещи. Они могут по-разному сочетаться. На вкладке Link and Materials есть список с помощью которого можно регулировать принадлежность массива данных (Datablock) объекту. Если вы имеете один блок у разных тел то они будут редактироваться одинаково. Так вы получаете фактически тело-копию. Хотя его редакция как тела то есть в режиме объектов не скажется на копии. Это даёт дополнительные возможности. При желании можно уравнивать положение, вращение и масштабирование тел стандартными средствами работающими и в других случаях. Правда, привязанные поверхности SubSurf не копируются. Ограничения (constraints) и модификаторы тоже не влияют на копии, но их можно скопировать отдельно.

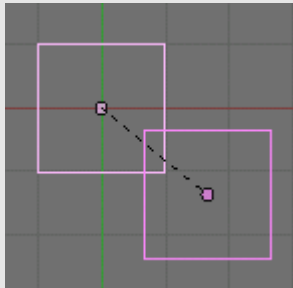
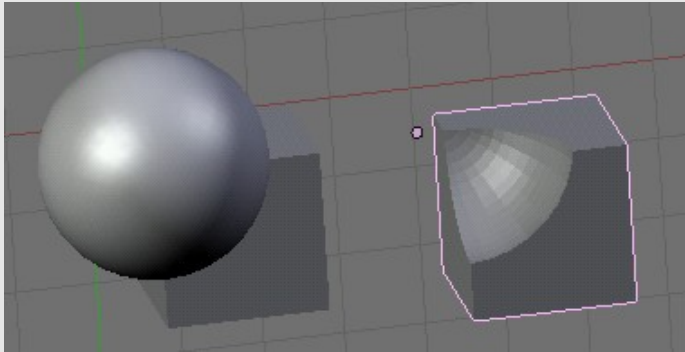
## Обзор методов построения блендер.

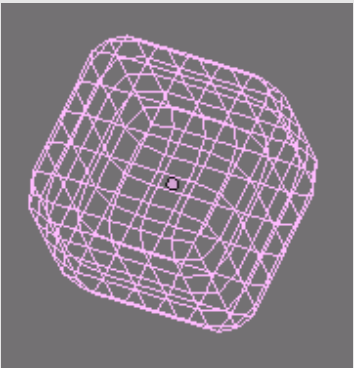
Это не традиционный обзор. Для этого есть руководство на сайте программы и других сайтах, например на <http://mediawiki.blender.org/>. Здесь я хочу сделать общий обзор не чтобы вы знали где что нажать а чтобы было ясно что к чему можно применять. Кроме того у каждого метода есть своё *применение*, область где он работает. Эту область можно очертить по-разному имея в виду разные разрезы работы, разные модели, задачи и тд. Обычно никто не пишет о ней в справке. С одной стороны это хорошо тк нельзя всё предсказать. С другой плохо тк это важно.

Каждый пользователь блендер должен понимать что именно редакция тел возможна только в режиме редакции, но действовать в блендер можно и в режиме объектов. В этом режиме вы работаете с телом в целом, и результаты этого будут сказываться на точках тела. Они будут видны в режиме редакции. Я буду по возможности использовать простой куб или поделённый куб для примера. Тут обозначены не все возможности программы.

### Действия с объектами в целом, вне редакции.

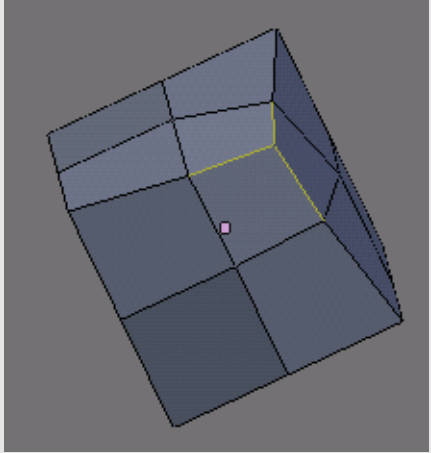
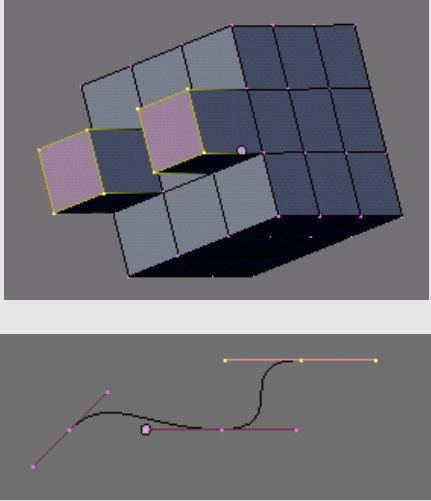
<i>Действие или инструмент</i>	<i>Назначение и комментарии</i>	<i>Вид тела после применения - по возможности на примере меша</i>
separate	Способ применить операцию к части тела или создать новый объект из части старого. Показан куб у которого отняли грань. Режим объектов. Отмена отделения либо отменой действий в режиме редакции либо в копии объекта. Альтернативой такой отмене являются булевы операции.	
вращение		

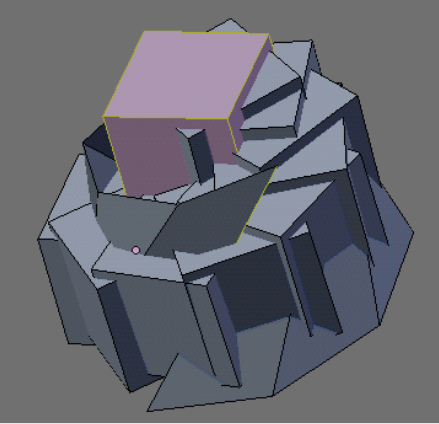
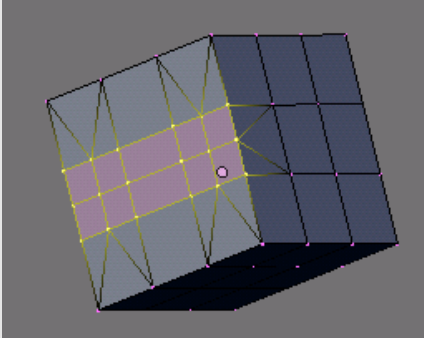
<i>Действие или инструмент</i>	<i>Назначение и комментарии</i>	<i>Вид тела после применения - по возможности на примере меша</i>
масштабирование		
отражение	Обращение по одной из осей в текущей системе координат.	
Копирование		
parenting	Установка отношений «родитель-потомок» для управления другим телом.	
tracking	Установка отношений при которых одно тело движется вдоль другого.	
Булевы операции	Операции сложения и вычитания объёмов. Работают только для закрытых мешей. Если в меше есть разрыв то ничего не будет сделано.	
dupliframes	Расположение копий тела по кривой.	

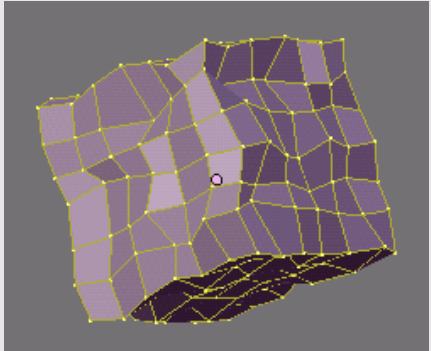
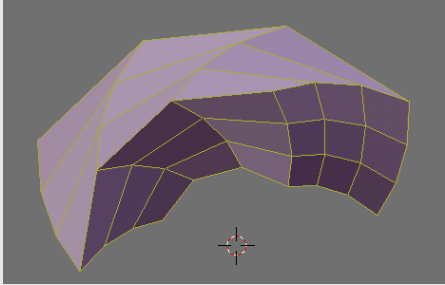
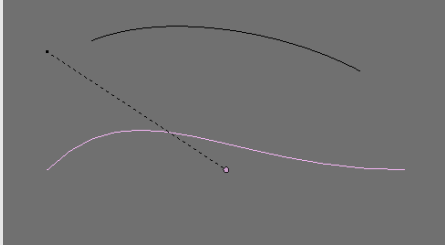
<i>Действие или инструмент</i>	<i>Назначение и комментарии</i>	<i>Вид тела после применения - по возможности на примере меша</i>
duplverts	Расположение копий тела в вершинах другого тела.	
Разброс как частиц	Совместно с parenting позволяет создать копии объектов которые ведут себя нелинейно при анимации.	
subsurf	<p>Создание привязанной поверхности которая может иметь разрывы и является альтернативой процессу сглаживания.</p> <p>В режиме объектов будет видна эта поверхность а не меш управляющий ей.</p>	

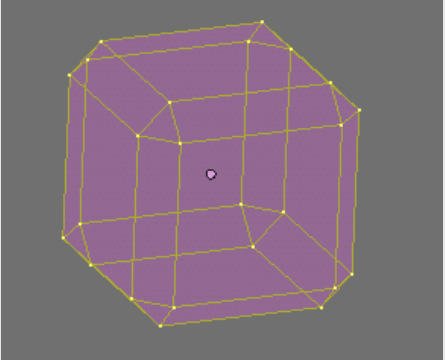
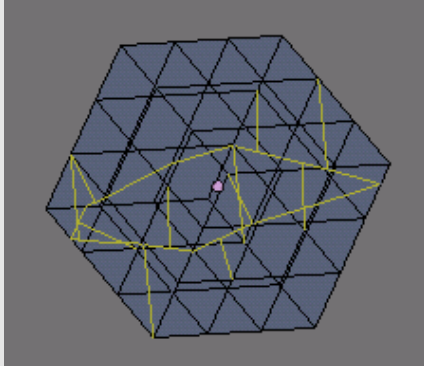
**Действия с частями объектов.**

Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
			меш	кривые	NU RBS	метаболы

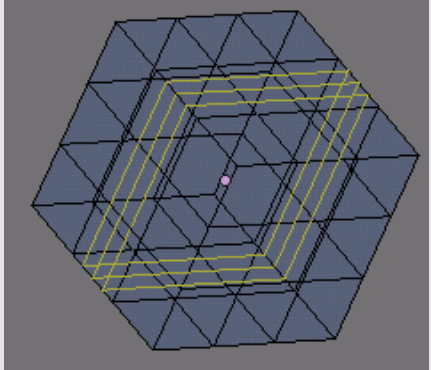
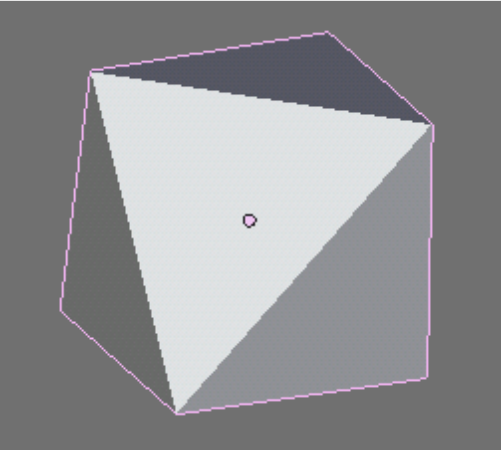
Действие или инструмент	Назначение и комментарии	Вид тела после применения – по возможности на примере меша	Тип тела			
smooth	<p>Сглаживание углов. С параллельным уменьшением поэтому фактически разделённые части станут разделены зрительно. Показан куб после сглаживания угла.</p>		+			
extrude	<p>Вытягивание копии выделенной части. Если возможны варианты появляется меню для выбора. Чтобы увести часть в сторону используя гизмо достаточно нажать enter.</p> <p>Если extrude выполняется для кривой то происходит наращивание кривой.</p>		+	+		

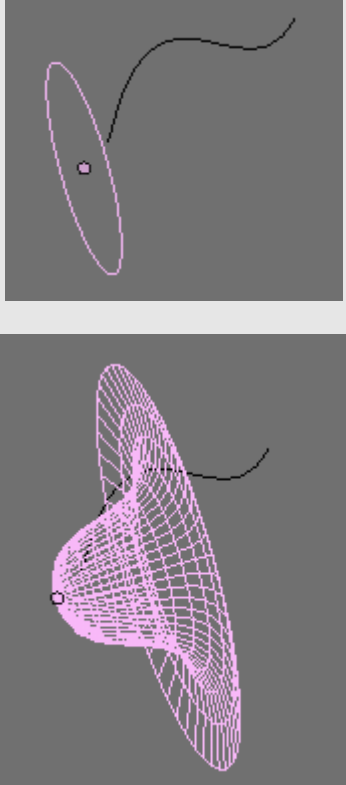
Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
Spin Screw	кручение		+			
subdivide	<p>Деление выделенной части на 2 или более равных частей. Будут поделены все рёбра. При этом будут поделены смежные грани если это меш.</p> <p>Для кривых это создание узла. Для NURBS тел - совокупности узлов.</p>		+	+	+	

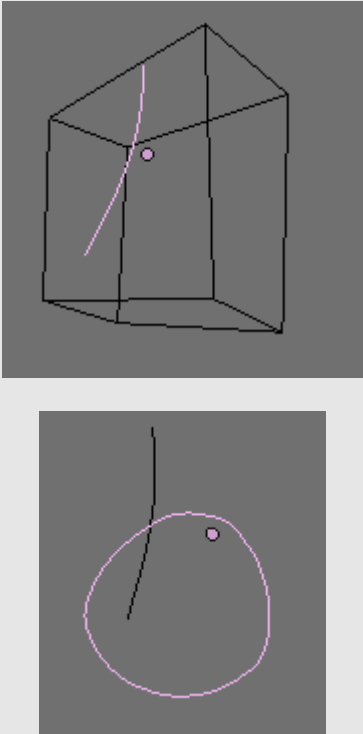
Действие или инструмент	Назначение и комментарии	Вид тела после применения – по возможности на примере меша	Тип тела			
Fractal subdivide	<p>Деление с присвоением хаотического положения точкам. Они смещаются во всех направлениях-- X, Y, Z. Может возникнуть потребность в смещении по одной-двум координатам и в разной степени. Инструмент хорош для моделей реально хаотических процессов и тел, например смятых.</p>		+			
warp	<p>Вытягивание по окружности.</p>		+	+	+	
hook	<p>Управление отдельными вершинами или узлами.</p>		+	+	+	

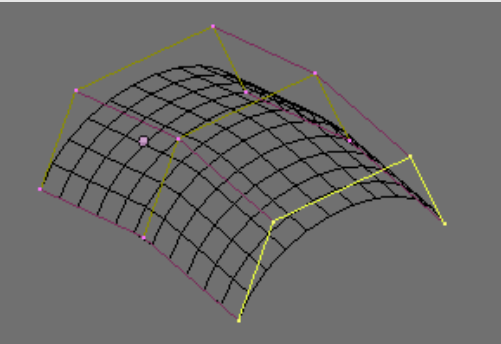
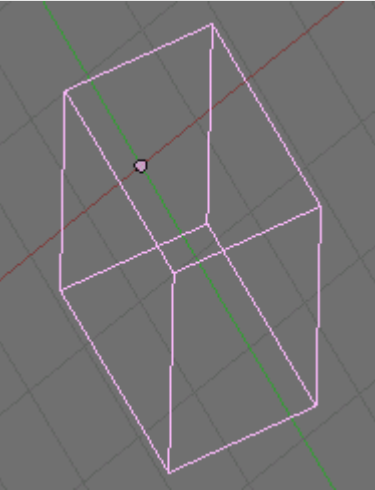
Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
bevel	Создание скоса то есть сглаживание углов.		+			
Knife	Многофункциональный инструмент для разрезания в том числе задних граней.		+			

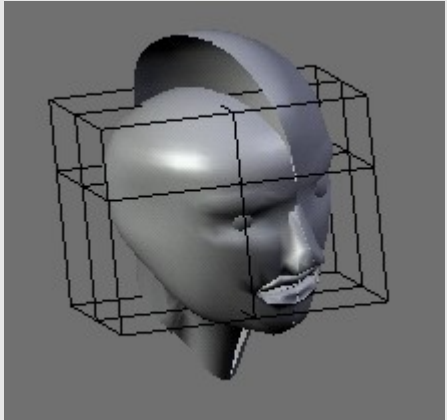
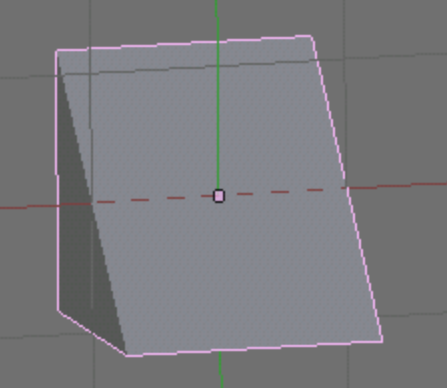


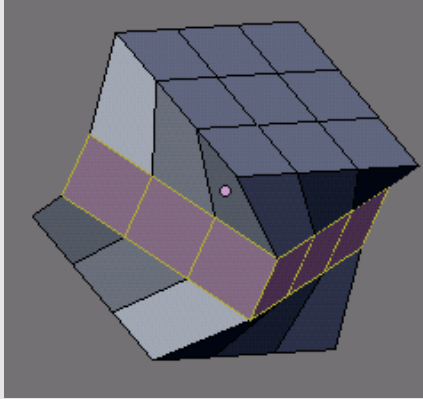
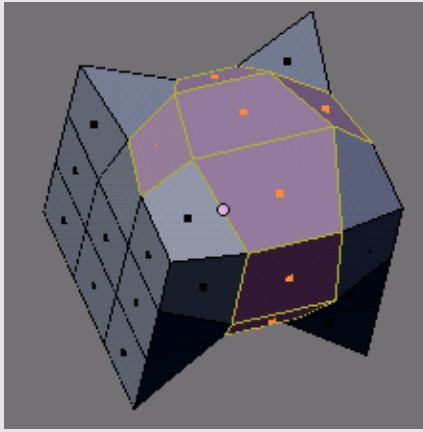
Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
Loop subdivide	Инструмент деления граней действующий на всё тело по периметру.		+			
decimator	Уменьшитель деления, упрощитель.		+			

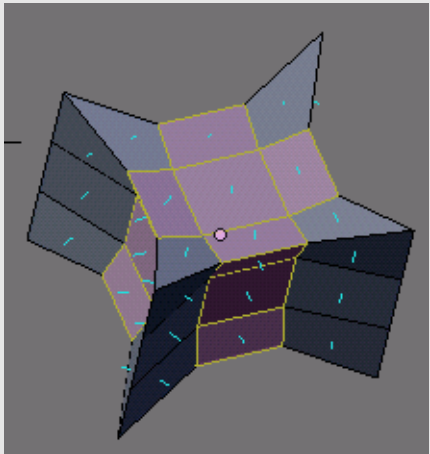
Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
Bev ob	Выдавливание вдоль кривой другой кривой или окружности.				+	+
curvetaper	Заостритель грани.		+			

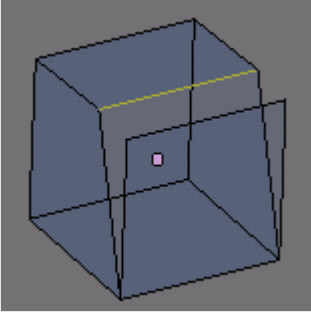
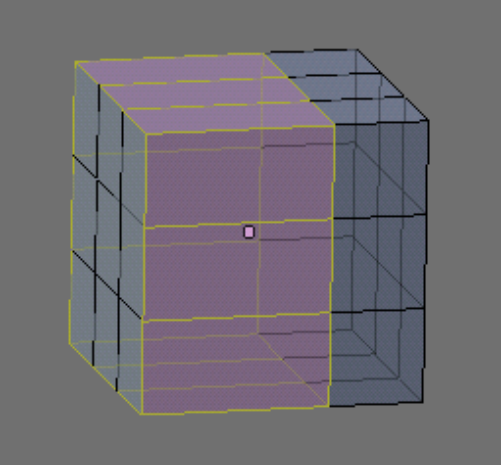
Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
Curve modifier	Деформация по кривой.		+	+	+	

Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
skinning	Заливка поверхностью нескольких кривых.				+	
Soft bodies	Симуляция упругости.		+			

Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
Lattice deformation modifier	Деформация с помощью решетки.		+	+	+	+
(поточечная редакция)	Редакция формы путём перемещения вручную отдельных точек построения.		+	+	+	
Slide			+			

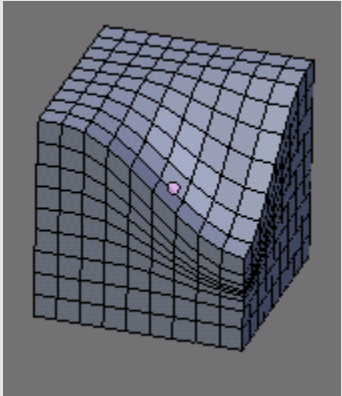
Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
вращение			+	+	+	+
tosphere			+	+	+	
merge	Соединение точек.		+			
Remove doubles	Удаление близко расположенных точек.		+			
Select non-manifold	Выделение края.		+			

Действие или инструмент	Назначение и комментарии	Вид тела после применения – по возможности на примере меша	Тип тела			
inverse	Обращение выделения.		+	+	+	
Random select	Выделение части вершин.		+			
Triangles to quads	Треугольные грани превратить в прямоугольные.		+			
Quads to triangles	Наоборот.		+			
Shrink \fatten along normals	Раздувание \сжатие вдоль нормалей.		+			
Rotate on axis	Вращение вдоль оси координат.		+	+	+	+
Scale on axis	Увеличение \уменьшение вдоль оси координат.		+	+	+	+
mirror	Отражение.		+	+	+	+

Действие или инструмент	Назначение и комментарии	Вид тела после применения - по возможности на примере меша	Тип тела			
Rip	Отделение части тела без разрушения соседней части. Чтобы увести часть в сторону используя гизмо достаточно нажать enter.		+			
Delete edge loop			+			



Действие или инструмент	Назначение и комментарии	Вид тела после применения – по возможности на примере меша	Тип тела			
Fill, beautiful fill	Заливка поверхностью.		+			
hide	Скрытие вершин.		+			
tile	Вращение нормалей.			+	+	

Действие или инструмент	Назначение и комментарии	Вид тела после применения – по возможности на примере меша	Тип тела			
Proportional edition	«резиновая» редакция вершин		+	+	+	

## Построение ножниц.

Для достижения одного результата может быть много способов. Этот пример может не быть лучшим способом.

Вначале надо конкретно представить результат. Ножницы бывают разные. Возьмём за основу классические ножницы. Они имеют

- винт для соединения частей
- 2 почти одинаковые части из
  - кольца,
  - полотна и
  - ножки между ними,
  - между ножкой и полотном плоскость для скрепления

Кроме того, интересно сделать у ножниц современную пластиковую обливку которую делают чтобы было удобнее двигать кольцами.

Теперь надо разобраться с приёмами, технологией и объектами потому что они будут определять действия. Надо посмотреть на детали формы потому что они могут потребовать специального

обращения. Мы делаем хорошую, правдоподобную модель поэтому детали должны 1)быть 2)быть как у реальных ножниц. Детали могут совершенно изменить план работы в случае изменения задач. *Для более грубой модели план может быть хорош, но не годится для лучшей модели.* Ножницы не будут анимированы поэтому в этом плане не важно что будет объектом, а что частью объекта. Вот соображения по этому поводу.

1. есть винт. винт не виден целиком, поэтому делать его весь не надо. Видна лишь его шляпка.

2. половинки ножниц не похожи а одинаковы. Они просто расположены иначе. Поэтому можно использовать тело-копию.

3. у современных ножниц обливка и кольцо разные поэтому вполне упростить работу копией не удастся. Можно сделать копию и отредактировать.

4. обливка должна иметь гладкие края в той части где она кончается. Это можно сделать сглаживанием. Если не сгладить то получится резко.

5. обливку можно сделать с помощью раздувания кольца: shrink-fatten along normals.

6. острый край не есть просто соединение двух основных плоскостей, он имеет свою узкую плоскость.

7. основная плоскость смотрящая наружу полотна имеет кривой край. Конец ножниц тоже.

8. край этой плоскости имеет сложную форму. Он имеет в разрезе поперек себя не угол а кривую. То же верно про скос между круглой частью ножки и её основой которая ближе к винту.

9. реальные ножницы имеют изогнутые полотна.

Так мы в основном обрисовали форму. Ещё надо найти технологию. Тут соображения такие.

1. в целом это слишком трудно сделать с помощью NURBS.

2. с помощью сглаживания можно добиться цели при меше хотя есть округлые части.

3. для колец удобно использовать кривые. Им можно придать любую форму. Поэтому можно сделать кольца самой реальной формы. Кривые можно замкнуть или вместо них применить окружности.

4. кривые колец будут иметь свои поверхности которые потом можно превратить в меш.

5. неясно как соединить рубленные части формы с округлыми. Можно сглаживать края частей пока они не станут округлыми.

6. неясно как делать соединение между кольцами и ножкой. Оно треугольное. Кольцо не может быть метаболой и его неудобно делать из метабол. Можно использовать рядом расположенные кривые с поверхностями.

7. простое опускание части полотна чтобы создать кромку не даст хорошего искривления на краю кромки. Там, как уже написано, сложная форма.

8. полотна лучше делать вручную тк сложная форма для которой не видно основы а работы не много.

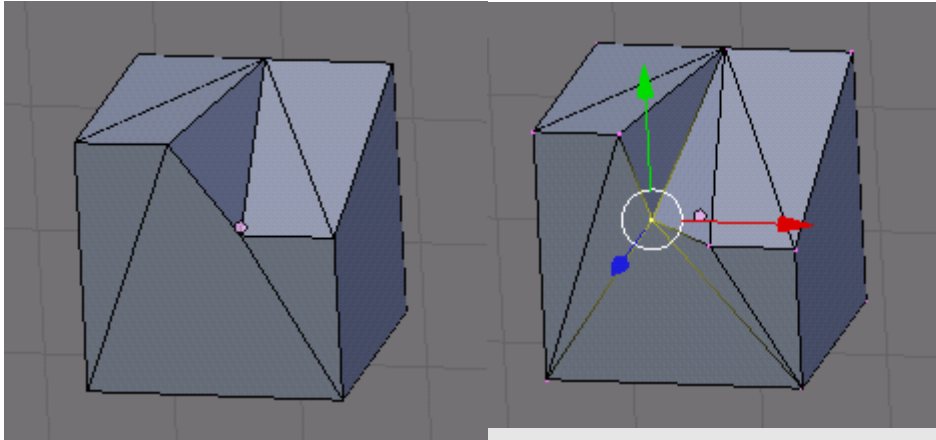
9. Реальные ножницы имеют скос нка углах кроме режущей кромки и на соприкасающихся поверхностях половинок ножниц.

Это ещё не все аспекты но многое прояснилось.

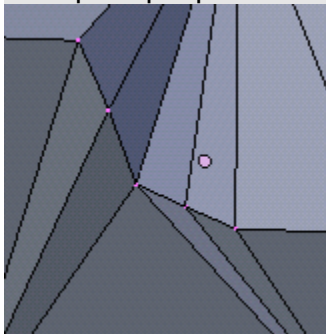
### **Скос у режущей кромки – поиск метода.**

Надо сделать скос у края режущей части. Проверим результат и попробуем использовать сглаживание. Задача: выяснить будет ли получаться нужная кривая, плавная форма если выполнять сглаживание для точек на широкой части этой формы, скоса. Мы решили что объект будет мешевым. Для пробы возьмем куб. Его придётся подготовить. На нём одно ребро будет поделено и отредактировано движением одной из его точек.

Поскольку мы хотим получить треугольный скос и ребра должны идти не к углам куба то надо было поделить другую грань пополам. Если не делить другую грань то ребра пойдут к углам. Приходится соглашаться на лишние ребра которые не портят модель. Теперь есть хорошая модель. Надо поделить грань скоса на несколько частей чтобы полученные точки могли быть сглажены.

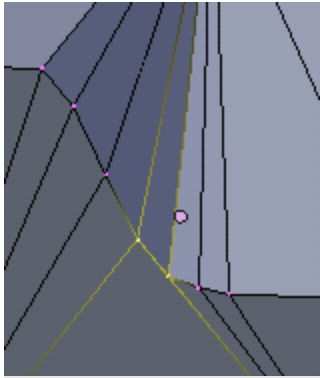


Скос имеет форму несводимую к окружности. Поэтому лучше именно вручную установить точку. Сложные кривые могут потребовать ручной работы. Скос поделен на 2 части. После выполнения subdivide smooth результат неприемлем тк стали выпирать ребра.



Грани сбоку стали другого цвета по этой причине. Сглаживание может сгладить не только то что надо но и что не надо сглаживать. Вместо этого поделим грань на 5 частей и подвинем одну из граней используя пропорциональную редакцию, её режим по умолчанию. Форма скоса теперь напоминает значок на кнопке режима пропорционального редактирования.

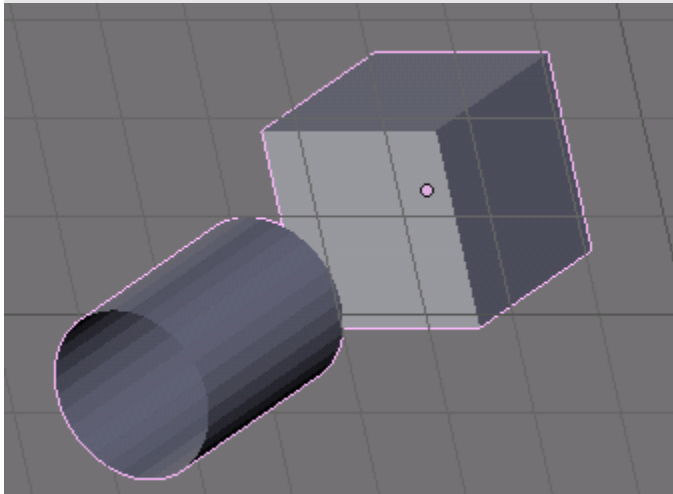
Итак чтобы сделать скос надо двигать его часть при стандартной «резиновой» редакции.



### **Переход от округлой формы к рубленой – поиск метода.**

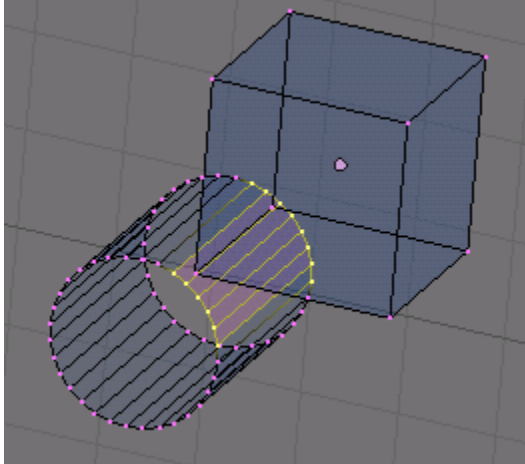
Теперь определим как сделать переход от ножки к середине где винт. После того как ножки будут сформированы с помощью кривых они могут быть преобразованы в меши. Поэтому можно считать это вопросом о соединении мешей. Для этого они должны быть объединены в одно тело. Это не проблема: object-join objects.

Можно взять tube (трубу) и куб. Это как бы части ножниц в этом месте.

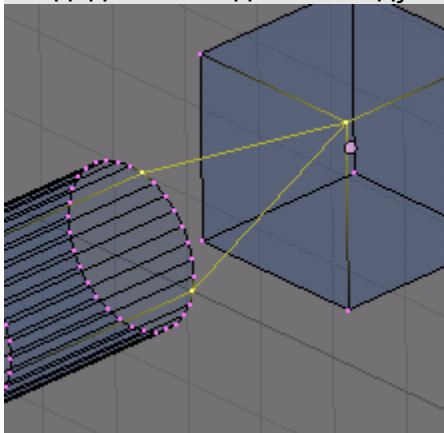


Теперь надо проверить план: соединить часть точек трубы и

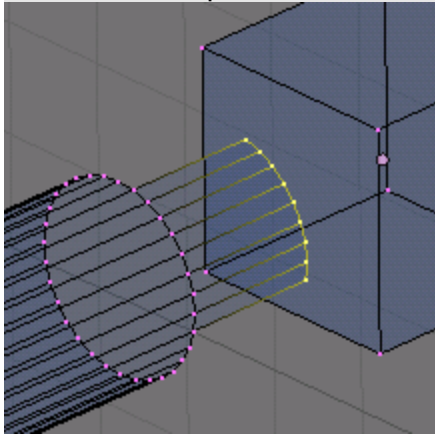
вершину куба. Перейдя в режим редакции выделим вначале середину той части которая должна быть выделена, затем нарастим выделение с помощью + на калькуляторе.



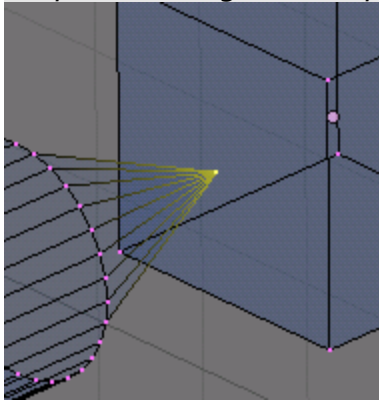
Теперь с помощью кругового выделения (B, B) и зажатого alt снимем лишнее выделение. Иногда гораздо легче сделать несколько операций чем выделять следуя прямолинейному подходу. Выделите ближайшую вершину куба. Нажмите esc чтобы выйти из режима выделения. Заливки fill, beautiful fill не работают. Потому что нет общих точек среди выделенных. Создадим их выделив каждую пару и нажав F.



Теперь блендер предлагает сделать Fgon, но не может. Сделаем иначе. Удалим ребра (edges) нажатием на del. План в том чтобы сделать продолжение выделенной части с новыми точками и дальнюю часть этой части сократить до 1 точки. Для этого выделим их и экструдим: E, only edges. Получается уже заполненная гранями, поделенная плоскость.



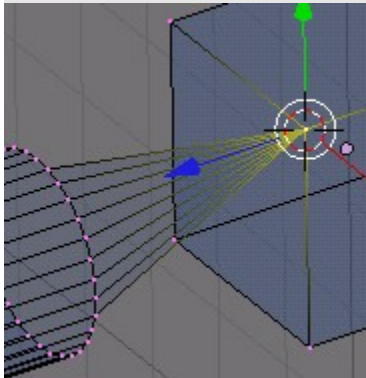
Теперь object-vertices-merge, at center. *Соединением, операцией merge можно редактировать тела.*



Как соединить вершину куба и выделенную точку? Поставить её вместо вершины нежелательно. Можно поставить её на одни координаты с вершиной и соединить их с помощью remove doubles. *Две точки могут быть на одной координате.* Включите

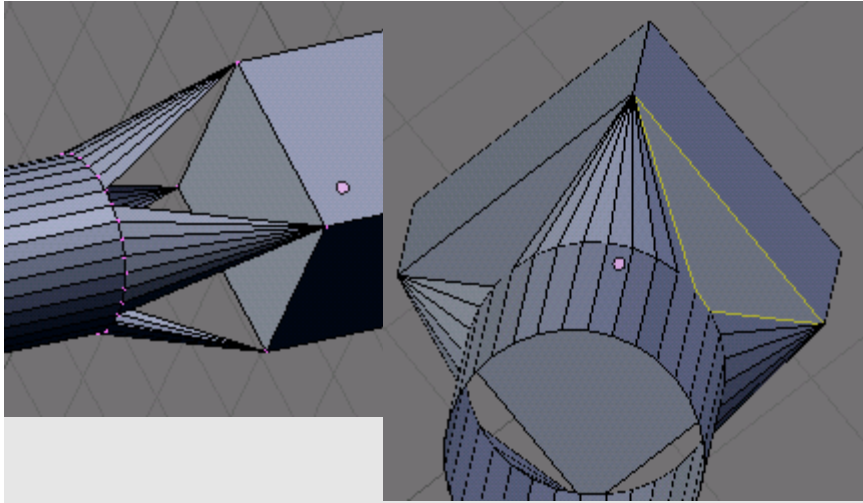


гизмо нажав кнопку с ручкой и со стрелкой (рядом). иначе этот способ не работает. Для выполнения операции может быть нужно включить гизмо. Подгоним курсор к вершине. Выделите вершину, object-snap-cursor-selection. Курсор на вершине. Подгоним точку схода к курсору. Теперь выделите точку где сходятся ребра трубы. Object-snap-cursor-selection. Объекты и их части можно подгонять к курсору, курсор можно использовать для их точной установки. Выделите вершину круговым выделением. Object-vertices-remove doubles. Удалением двойных точек можно выполнять присоединение к части того же тела. Мы выполнили замену точек без смены координат.



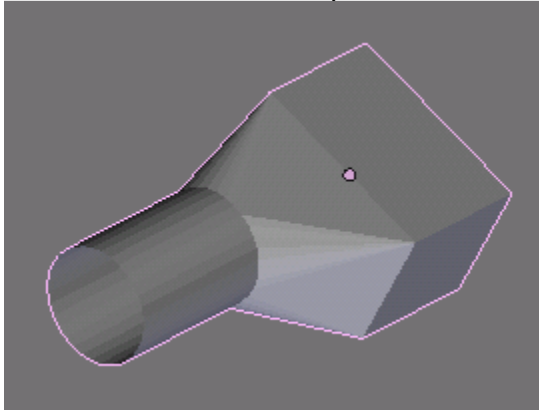
Можно просто экструдировать и соединить на месте курсора, а его ставить на вершину. Если при соединении вершина выделена то результат будет такой же. Вот результат для всех вершин куба.

Между лепестками идущими от трубы осталось по одному ребру. Это позволяет делать грань между трубой и кубом тк общее число её ребер равно 4. Зажав alt и щелкая на любой точке этих ребер выделим ребра.



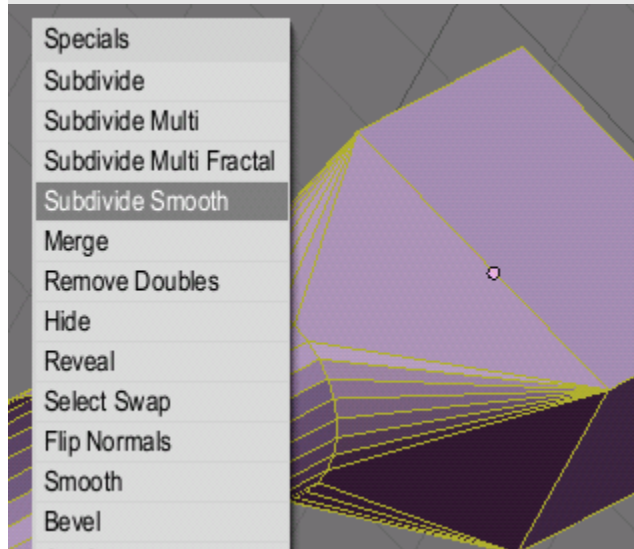
Я нарочно поставил точку зрения чтобы лучше видеть ребра и перешёл в режим ребер. *Точка зрения и режим делают видимыми прежде забытые светом части построения в которых могут быть невидимые ошибки, и которые могут из-за лишнего освещения только казаться имеющими ошибки.*

Выделяя с зажатым alt пустые места и нажимая на F я создал грани, каждую отдельно. Слегка расплющим и переместив куб с помощью гизмо я получил нечто похожее на часть ножниц. Площадка выключена, режим объектов и solid.

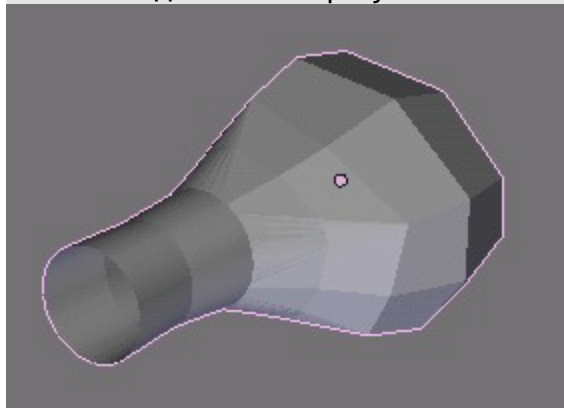


## Закругление краёв – поиск метода.

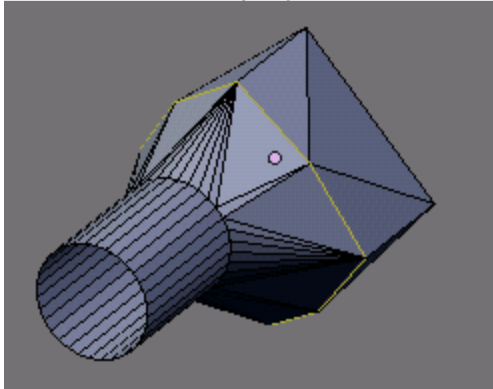
Теперь на той же модели проверим способ сглаживания. Нам не годится просто сглаживать тк всё равно останутся острые углы. Они острые даже если тупые. То есть есть одна вершина. Поэтому нужно делить вершину. Или всё тело – в режиме редакции, разумеется. Для этого можно вызвать меню нажав W. После этого ответим нажав ОК в окошке со значением 1. 000.



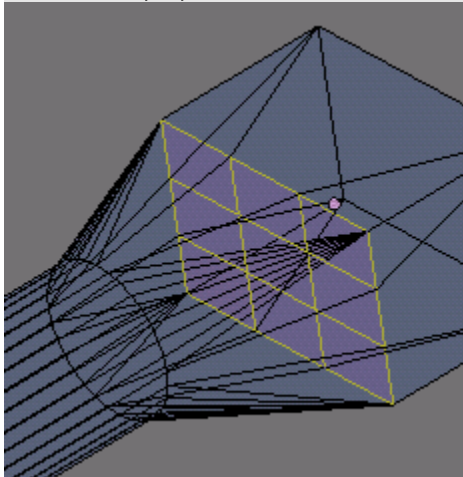
Вот анекдотический результат.



Юмор – это хорошо но не сейчас. Кубическая часть была поделена. Поэтому она искажена. Значит надо сглаживать не всё. А выделенное. Это точки. Перейти в режим точек. Вот анекдотический результат.

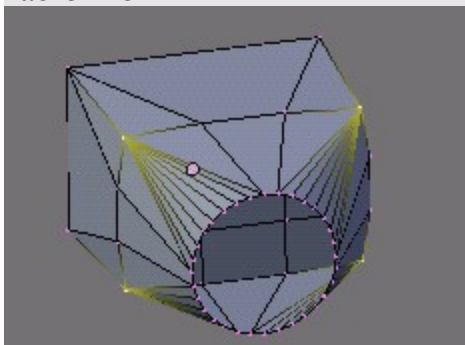


Мораль: это не годится. Нам надо не одну точку а 2 или больше между вершинами. Инструмент *bevel* даёт в этом случае неприемлемый результат уродуя форму. Он работает даже для тех частей которые не выделены. Можно делить ребра куба: W, subdivide, 2, enter.

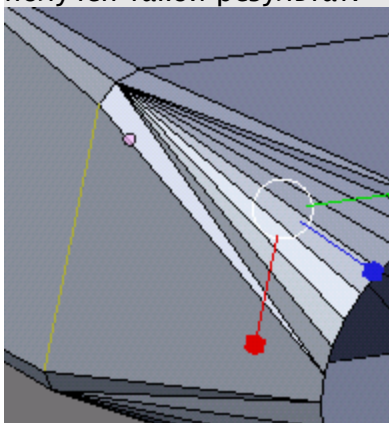


Вопрос: как одновременно для уменьшения работы и одинаково подтащить новые ребра к углам куба? Проще двигать Магомета к

горе. Двигать вершины с помощью масштабирования: S, гонять мышью, enter. Масштабирование (*scale*) может использоваться для одинаковой и быстрой установки положения нескольких тел или частей тел.

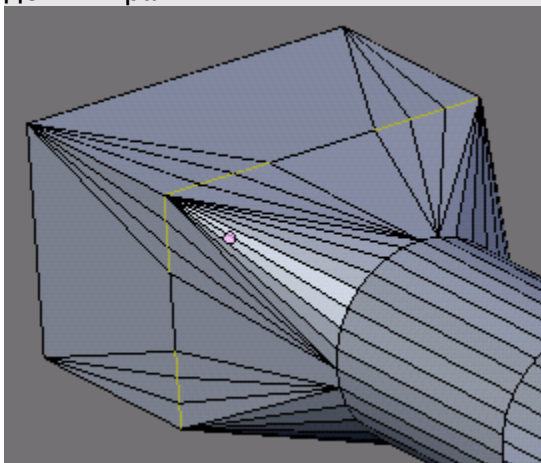


Труба скрыта. И всё же надо ещё сделать края поострее. Они слишком плавные, и именно вершины остались почти теми же. С помощью масштабирования ребер которые между вершинами был получен такой результат.



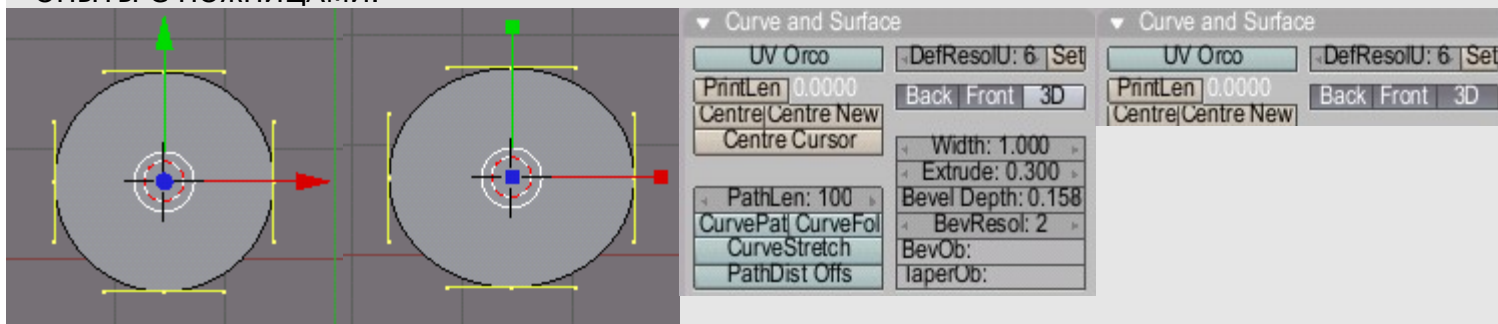
Теперь виден недостаток в том что есть острый конус. В анимации игнорируются ещё не такие дефекты если можно. Но вопрос остаётся на теоретическом уровне. А вот и выход: надо было просто 1)уменьшить 2) переместить часть окружности которая соответствует вершине. Нет, тогда вопрос о конусной части просто уйдёт к другому углу. *Посмотрите не перемещаете*

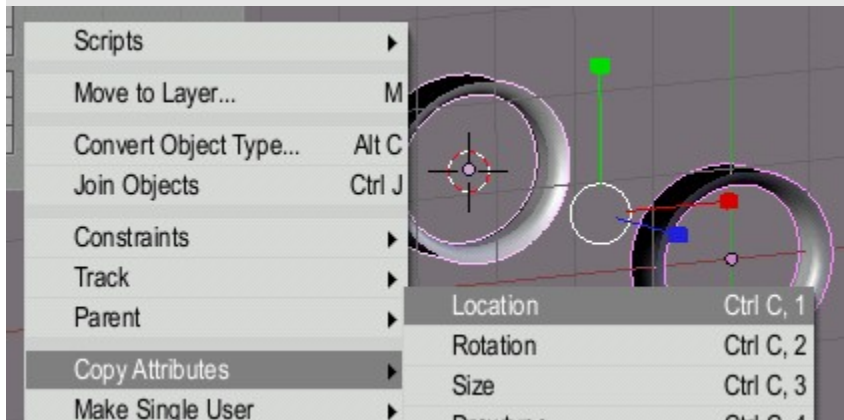
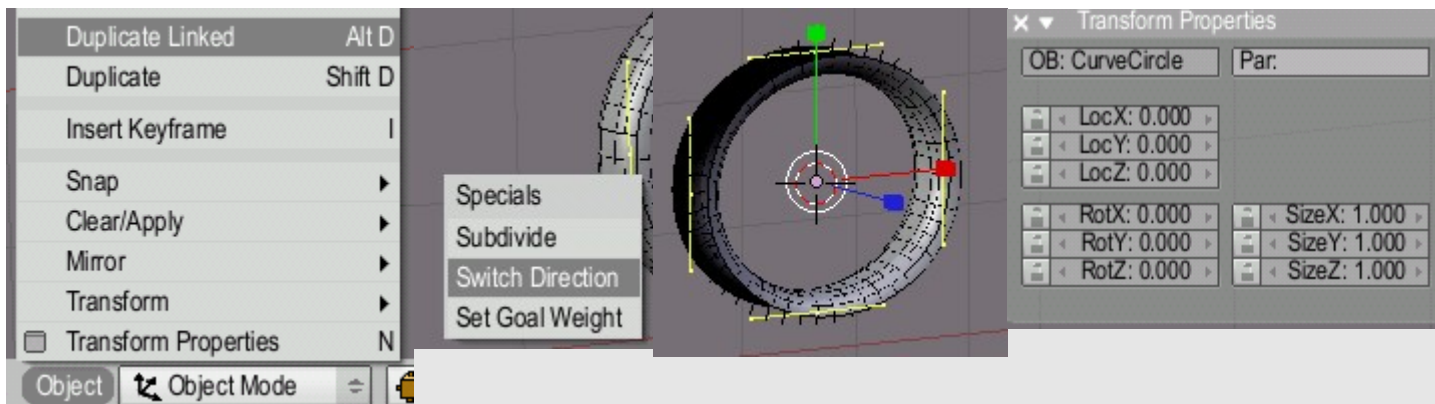
ли вы проблему на другие части тел или тела. Можно просто делить грани.



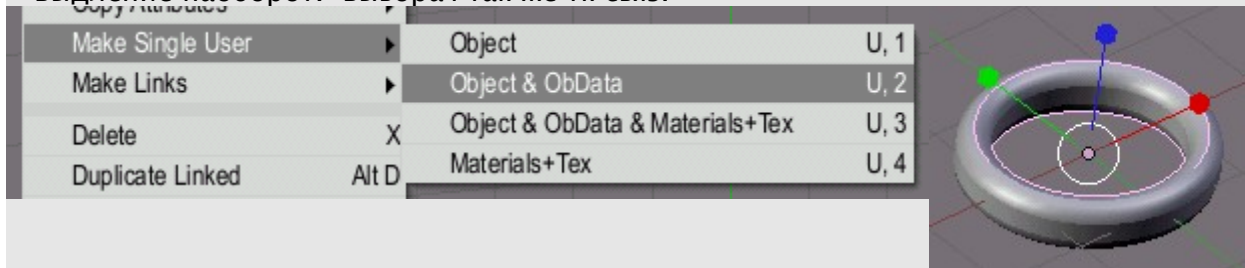
А затем перемещать вершины. Но вопрос о конусной части остался.

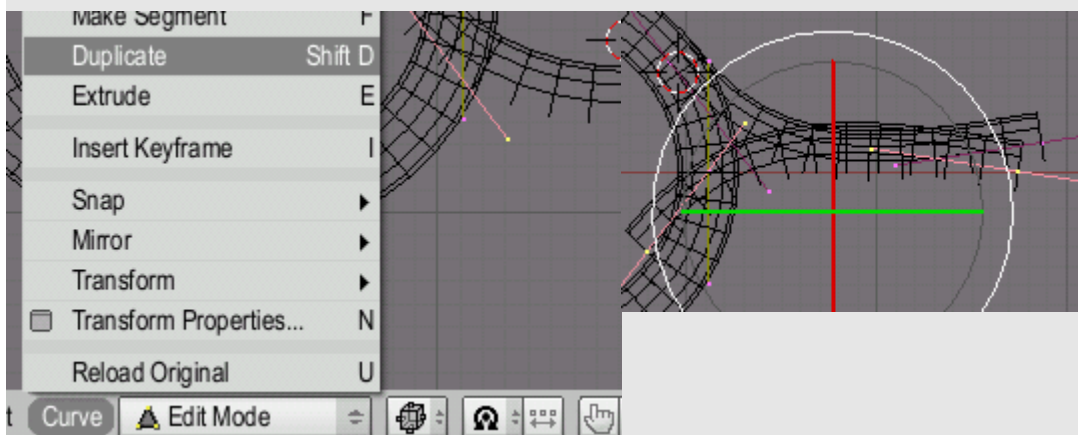
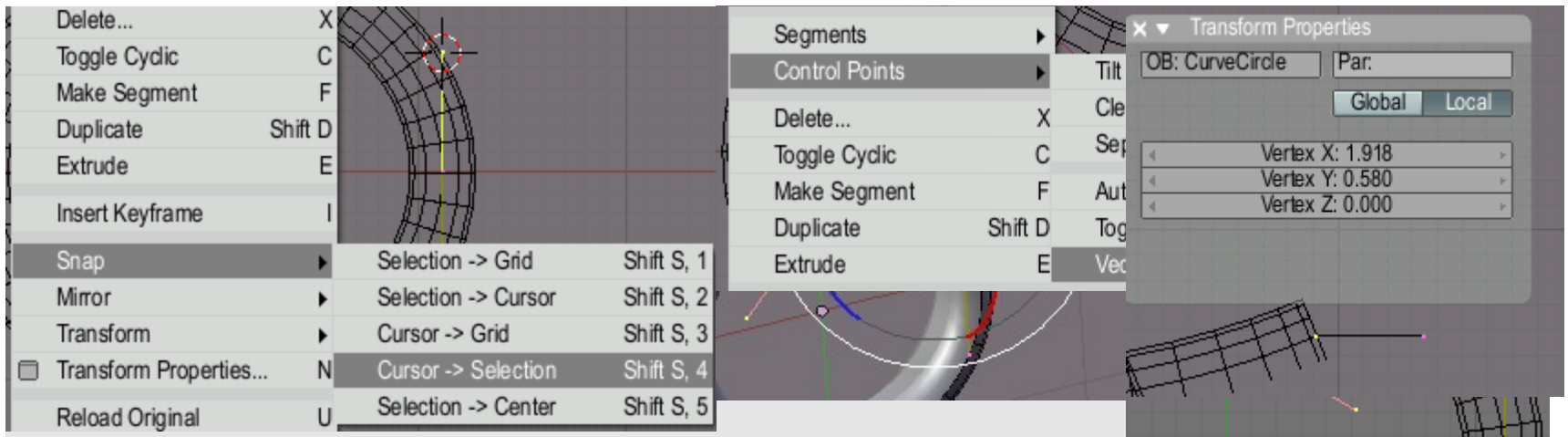
ОПЫТЫ С НОЖНИЦАМИ.





Выделение наоборот. Выворач так же тк связ.







Открытие, сохранение, закрытие файлов.

Имена объектов.

Факторы выбора типа объекта.

Выделение. его необратимость. инверсия. связь с режимом.

сохранение при возврате в редакцию. сложная стратегия.

выделение ряда в НУРБС. выделение после операций.

Минимум для рендера.

Флексагон.

Наличие объекта для редакции.

Системы координат. Гизмо. режим центра гизмо.

Сглаживание разными способами и в рендере.

Режимы точек, линий, граней. выделение в них.

Центр тела и редакция. установка центра. роль центра.

Деление. нож.

Перемещение только центров тел.

Факторы выделения объектов. разделение и слияние объектов.

объединение – только для однотипных объектов.

Сложение и вычитание тел.

Extrude.

Редакция и определение численных значений.

Заливка гранями.

Нормали и вид. Прорисовка видеокарты и вид. Нормали и сглаживание. Нормали и режим. Нормали соседних граней.

Наращивание сегментов кривых и мешей.

Замыкание кривых.

Резиновая редакция для соединенных и всех точек.

Роль количества точек в операциях. математическая основа приводит к неожиданным рез-там.

НУРБС как решётки. отверстия в них.

Поточечное строительство формы.

Автоматическая заливка замкнутых кривых как плоских тел.

Деление на равные части.

Копланарные грани.

Деление грани и новые грани.  
Синусоида.  
Звёзды.  
Спирали.  
Симметричное построение. связанные копии.  
Слои и скорость показа.  
Строительство формы при анимации.  
Построение ножниц.

## **Программирование, плагины и скрипты в блендер.**

Плагины и скрипты в блендере являются мощным средством добавления новых возможностей. Блендер уже содержит несколько скриптов.

Этот текст написан для опытных пользователей но не программистов. Поэтому нужны объяснения. Блендер имеет встроенную систему поддержки плагинов и скриптов. И то и другое являются результатом программирования и распространяется согласно тем условиям которые определены авторами и законом. Все это программные продукты.

Когда идёт речь о программировании то профессионалы часто не понимают что их объяснения обрывочны. Это проявление их большой подготовки. Такие понятия как типизация, объектность для них являются фундаментом. Поэтому они не объяснены или не понятно как они связаны. Не пытаюсь заменить толстые книги я хочу дать некое представление. Блендер создаёт вам мотив программировать. Но для этого надо дать определение и закрыть белые пятна. Иначе руководства и справки могут выглядеть совершенно непонятно или даже понятно, но будут пониматься неправильно. Программирование вертится вокруг нескольких понятий. Разберем некоторые из главных понятий.

Что такое программа на интуитивно-наглядном уровне знает каждый пользователь. Это окна которые позволяют вам что-то делать. Это верно. Но тут нужно дать более глубокое определение. Плагины и скрипты являясь программными

модулями которые сами ничего не делают и могут даже не иметь внешнего выражения. Они используются в блендер и скорее всего нигде больше. Вы можете скачать из интернета разные плагины и скрипты, там же скорее всего написано как с ними обращаться. Вы можете использовать их или модифицировать если это не входит в противоречие с лицензией. Вы можете сами написать скрипт или плагин.

Именно в случае редакции или авторства нужны пояснения. Пользователь в 99 случаях из 100 имеет дело с окнами и не знает что такое программирование. Книги по этому вопросу могут быть необходимы в рассматриваемом случае. Программирование – это создание 1)скриптов (сценариев) или 2)компилированных модулей и программ. В первом случае создается текст который должен быть прочтён некой программой которая будет что-то делать в соответствии с этим текстом. Во втором случае есть файл который не является текстом но выполняется то есть может быть непосредственно использован микропроцессором для технических процессов. Программы сами создаются из скриптов которыми являются тексты на языках программирования (ЯП), они называются кодами (не двоичными кодами!) или source. Это первый случай когда результатом выполнения скрипта является компилированный модуль. Также результатом может быть некое действие в программе читающей скрипт. Так работают скрипты блендера и других программ.

Когда вы создали или отредактировали код плагина блендера то для работы плагина вам потребуется компилировать его. То есть превратить в программу.

Для работы скрипта в блендер встроен усечённый интерпретатор языка программирования python. Это программа которая выполняет скрипт, а результаты этой работы получает блендер.

Язык программирования – это совокупность понятий и правил для описания компьютерных процессов. Сам по себе ЯП ничего не даёт. ЯП существует для того чтобы написанный текст (скрипт) был использован. Не любое, а именно верное, на ЯП заявление может быть использовано. Это значит что некая программная

среда будет обрабатывать заявления на ЯП. Поэтому ЯП обладает строгим набором правил – синтаксическими правилами. Нарушение этих правил приводит к тому что скрипт не выполняется и выводится сообщение об ошибке. Язык считается высокоуровневым если он позволяет высокую степень обобщения. Так, например, Ассемблер не является таковым. В этом языке используются простые команды и для выполнения одного указания на высокоуровневом языке потребуется написать страницу текста на Ассемблере.

Вы обязательно столкнётесь с понятием типов или типов данных. Тип данных – это структура данных которая предусмотрена в языке программирования. Так, например, в обиходе существуют такие типы литературных произведений как проза и стихи, повести, романы и пр. они различаются по содержанию и по формальным признакам. Так же и в ЯП есть несколько предусмотренных типов которые будут 1)понятны при выполнении 2)обслуживаться предусмотренным образом, то есть для них уже есть готовые методы которые не надо создавать самому.

Одним из типов данных является объект. Объект вам важен еще в том отношении что он может быть рассмотрен как подпрограмма. Это не значит что он должен быть описан в отдельном тексте. Но объект – это удобная вещь которая несёт в себе скрытые (инкапсулированные) свойства (property) и выдаёт вам известные результаты при происхождении известных событий (events). Это чёрный ящик созданный для использования в разных программах. Объект – то средство упрощения работы потому что он при правильном использовании снимает вопрос о своих внутренних процессах, и, во-вторых, он понятийно удобен тк соответствует объектному мышлению людей. Антитеза объектного подхода в процедурном подходе. Первый сосредоточен вокруг вопроса «с чем делать», а второй «что делать».

Понятие класса существует для того чтобы отразить академическое описание объекта. Для класса существуют методы которые работают с любым экземпляром класса одинаково.

Функцией ЯП является выражение которое возвращает значение, в противном случае это процедура. Но фактически в отдельном языке функциями могут называться и выражения которые не возвращают никаких значений. В математике функция  $y=3x$  возвращает 6 если  $x=2$ . Так же функция несет значение. Оно может быть не только числом.

Переменными в разных языках могут считаться разные типы. Кроме строк и чисел это могут быть объекты и пр.

Строки и числа являются разными типами потому что с помощью числа можно считать. Язык со строгой типизацией никогда не путает разны типы и программист должен их обозначать в программе. Относительно предусмотренных типов будут работать стандартные функции поэтому их не надо писать самому.

Когда вы работаете то можете столкнуться с понятием модуля. Модуль - это блок кода который обычно подключается к программе выполняющей скрипт. Модули различаются назначением и созданы для использования в разных программах.

Выполняющая программа (среда выполнения) может создавать и выполнять программу двумя способами. Первый - интерпретация, второй - компиляция. Первый в том чтобы последовательно читать код не забегая вперед. Этот подход может привести к остановке выполнения тк найдена ошибка. Второй подход в том чтобы сделать реферат кода а потом выполнять. Компилятор дольше запрягает но быстрее едет. Обычно среда выполнения существует как компилятор.

Используемый в блендер язык python является объектным, он рассчитан на интерпретацию с помощью встроенного интерпретатора.

