

*МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА (РАБОТЫ)*

Целью данной курсового проекта (работы) является разработка анимированной 3D модели в среде Blender с помощью ключевых кадров, драйверов и модификаторов.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Blender — пакет для создания трёхмерной компьютерной графики, включающий в себя средства моделирования, анимации, рендеринга, постобработки видео, а также создания интерактивных игр.

Особенностями пакета являются малый размер, высокая скорость рендеринга, наличие версий для множества операционных систем — FreeBSD, GNU/Linux, Mac OS X, SGI Irix 6.5, Sun Solaris 2.8 (SPARC), Microsoft Windows, SkyOS, MorphOS и Pocket PC. Пакет имеет такие функции, как динамика твёрдых тел, жидкостей и мягких тел, систему горячих клавиш, большое количество легко доступных расширений, написанных на языке Python. Программа является свободным программным обеспечением и распространяется под лицензией GNU GPL.

В зависимости от версии Blender его функционал и интерфейс могут отличаться.

Возможности

Характерной особенностью пакета Blender является его небольшой размер. Установленный пакет занимает около 50 МБ. В базовую поставку не входят развёрнутая документация и большое количество демонстрационных сцен.

Функции пакета:

- Поддержка разнообразных геометрических примитивов, включая полигональные модели, систему быстрого моделирования в режиме subdivision surface (SubSurf), кривые Безье, поверхности NURBS, metaballs (метасферы), скульптурное моделирование и векторные шрифты.

- Универсальные встроенные механизмы рендеринга и интеграция с внешним рендерером YafRay.

- Инструменты анимации, среди которых инверсная кинематика, скелетная анимация и сеточная деформация, анимация по ключевым кадрам, нелинейная анимация, редактирование весовых коэффициентов вершин, ограничители, динамика мягких тел (включая определение коллизий объектов при взаимодействии), динамика твёрдых тел на основе физического движка Bullet, система волос на основе частиц и система частиц с поддержкой коллизий.

- Python используется как средство создания инструментов и прототипов, системы логики в играх, как средство импорта/экспорта файлов (например COLLADA), автоматизации задач.

- Базовые функции нелинейного редактирования и комбинирования видео.

- Game Blender — подпроект Blender, предоставляющий интерактивные функции, такие как определение коллизий, движок динамики и программируемая логика. Также он позволяет создавать отдельные real-time приложения начиная от архитектурной визуализации до видео игр.

Дополнительные особенности:

- В Blender Объект (который представляет собой сущность, взаимодействующую с окружающим миром) и его Данные (определённая форма/функции объекта) разделяемы. Отношение Объект-Данные представляется отношением m:n (термин, относящийся к теории баз данных, обозначает возможность нескольких объектов использовать одни и те же данные — один ко многим или сюръекция) и динамически связаны между собой, позволяя использовать некоторые процессы быстрого моделирования, уникальные для Blender.

- Внутренняя файловая система, позволяющая хранить несколько сцен в едином файле (называемом .blend файл).

- Все «.blend» файлы совместимы как с более старыми, так и с более новыми версиями Blender. Так же все они переносимы с одной платформы на другую и могут использоваться как средство переноса созданных ранее работ.

- Blender делает резервные копии проектов во время всей работы программы, что позволяет сохранить данные при непредвиденных обстоятельствах.

- Все сцены, объекты, материалы, текстуры (только собственные, не импортированные), звуки, изображения, post-production эффекты могут быть сохранены в единый «.blend» файл.

- Настройки рабочей среды могут быть сохранены в «.blend» файл, благодаря чему при загрузке файла вы получите именно то, что сохранили в него. Файл можно сохранить как «пользовательский по умолчанию», и каждый раз при запуске Blender вы будете получать необходимый набор объектов и подготовленный к работе интерфейс.

Тем не менее, внутреннее содержание «.blend» файла менее похоже на структурированное описание объектов и их взаимоотношений, и более близко к прямому дампу области памяти программы. Это делает практически невозможным преобразование «.blend» файлов в другие форматы. При этом

следует заметить весьма продвинутый механизм экспорта в разнообразные форматы, такие как obj, dxf, stl, 3ds и прочие (список постепенно растёт).

После запуска Blender пользователь увидит окно, в котором будет меню с основными пунктами, как то загрузка/сохранение файлов, в центре 3D вид изображения и внизу окно кнопок управления.(рис. 1):

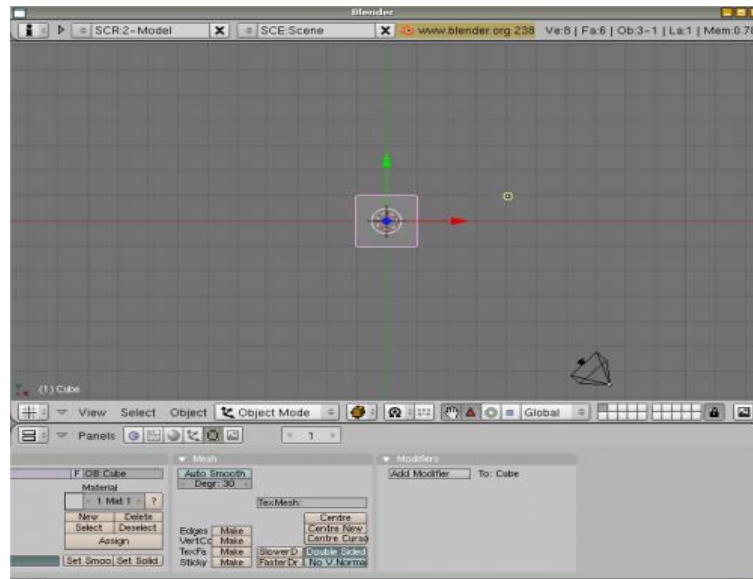


Рис. 1 - Начальное окно Blender

Средняя кнопка мышки (ММВ) используется для вращения сцены и достижения лучшего обзора: стандартная сцена содержит источник света (желтая точка), камеру (объект справа) и куб (он выбран, у него розовые края и манипулятор отображен 3 стрелками, рис. 2). Не надо кликать в 3D окно левой кнопкой мышки (LMB) сейчас, так как это не выберется, но обычно начнется выполнение некоторых функций обработки. Если происходит что-то странное, надо нажать Esc, также в большинстве случаев работает, Ctrl Z для отмены нежелательных изменений.

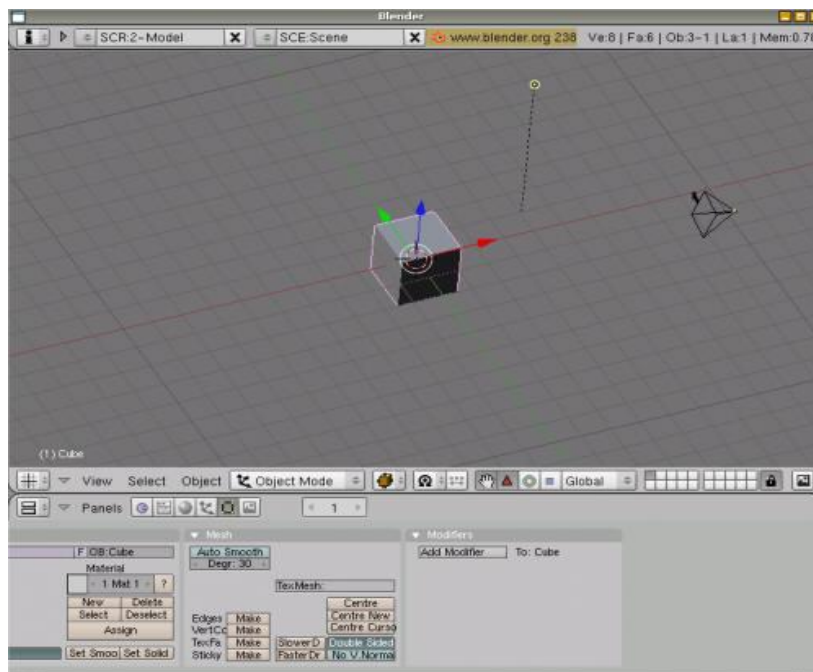


Рис. 2 - Стандартная сцена

Далее необходимо переключить вид. Для этого надо нажать на выделенную кнопку и переключить вид на 'wireframe' (рис. 3). Используйте Shift + MMB для перемещения вида и центруйте где-нибудь между 3 объектами. Используйте колесо прокрутки мышки или Ctrl + MMB для изменения размеров сцены, пока вы не получите вид подобный приведенному на рисунке. Убедитесь что курсор с красными-белыми полосами все еще отцентрирован на кубе (если Вы не кликнули где-нибудь в 3D-окне левой кнопкой мышки, это должно быть так). Это место, где будет создан новый объект.

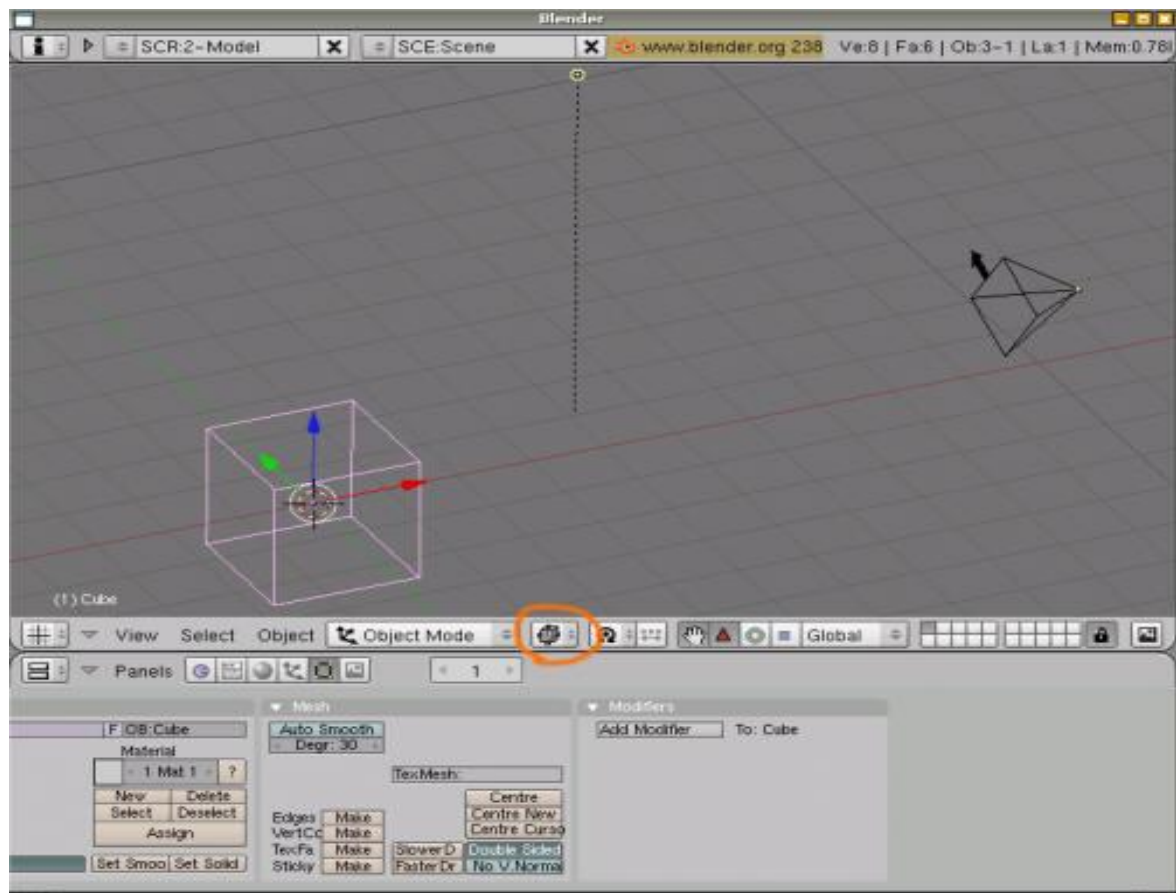


Рис. 3 - Переключение видов

СОЗДАНИЕ СФЕРИЧЕСКОЙ КАПЛИ

Нажмите Space и из меню выберите Add→Mesh→Icosphere как показано на рисунке 4.

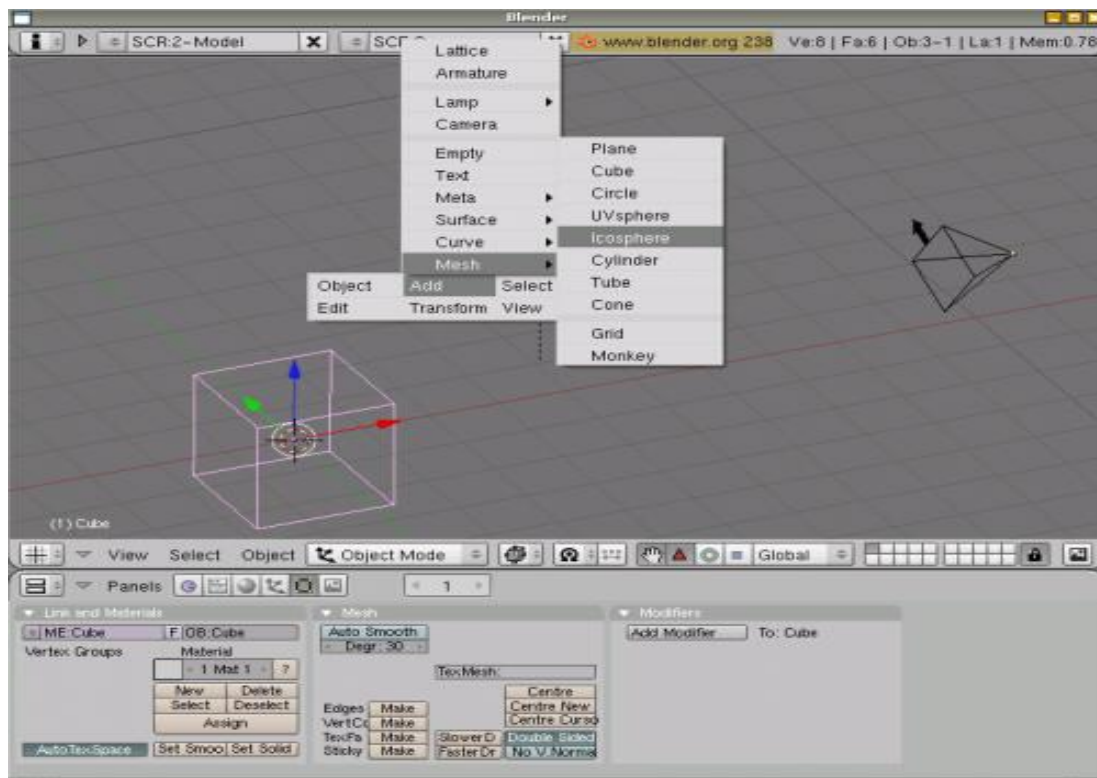
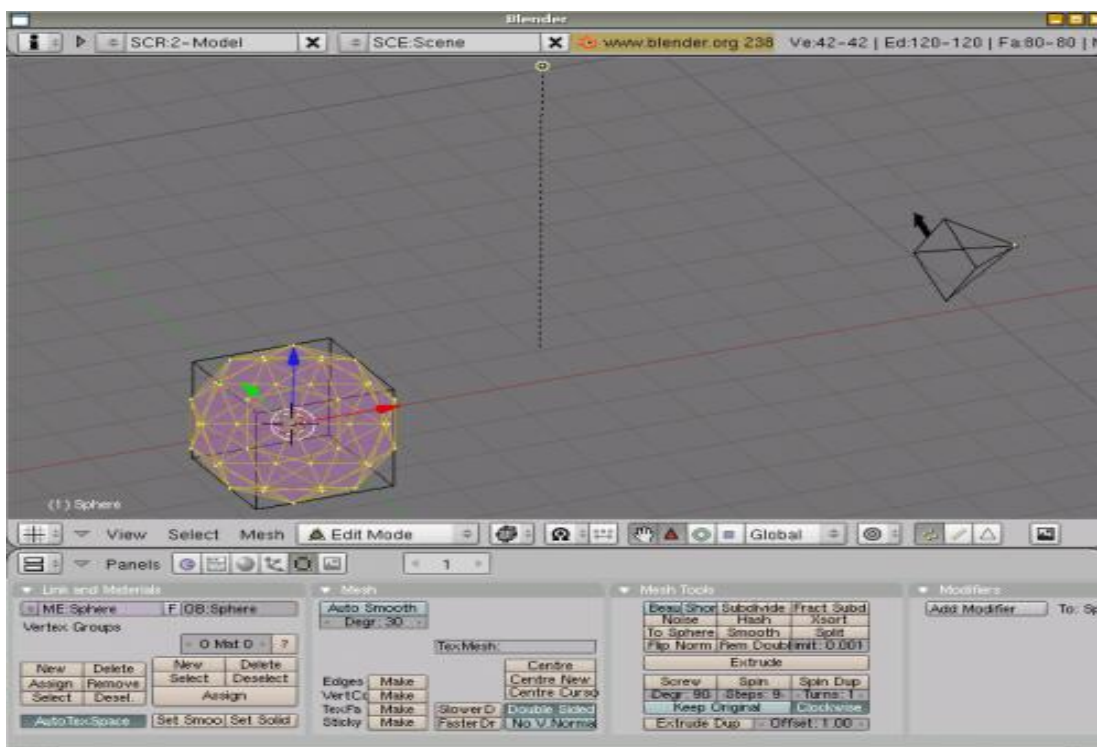


Рис. 4 - Начало создания объекта

Вы получите небольшое окно Subdivision под курсором - не двигайте мышку слишком далеко, иначе окно исчезнет без создания чего либо.



Кликните на кнопке ОК, должно получиться как на рисунке 5:

Рис. 5 - После выбора Subdivision

Здесь новая сетка сферы. Blender автоматически переходит в режим редактирования (это также видно ниже окна 3D), если нажать Tab то перейдете в "нормальный" объектный режим (рис. 5).

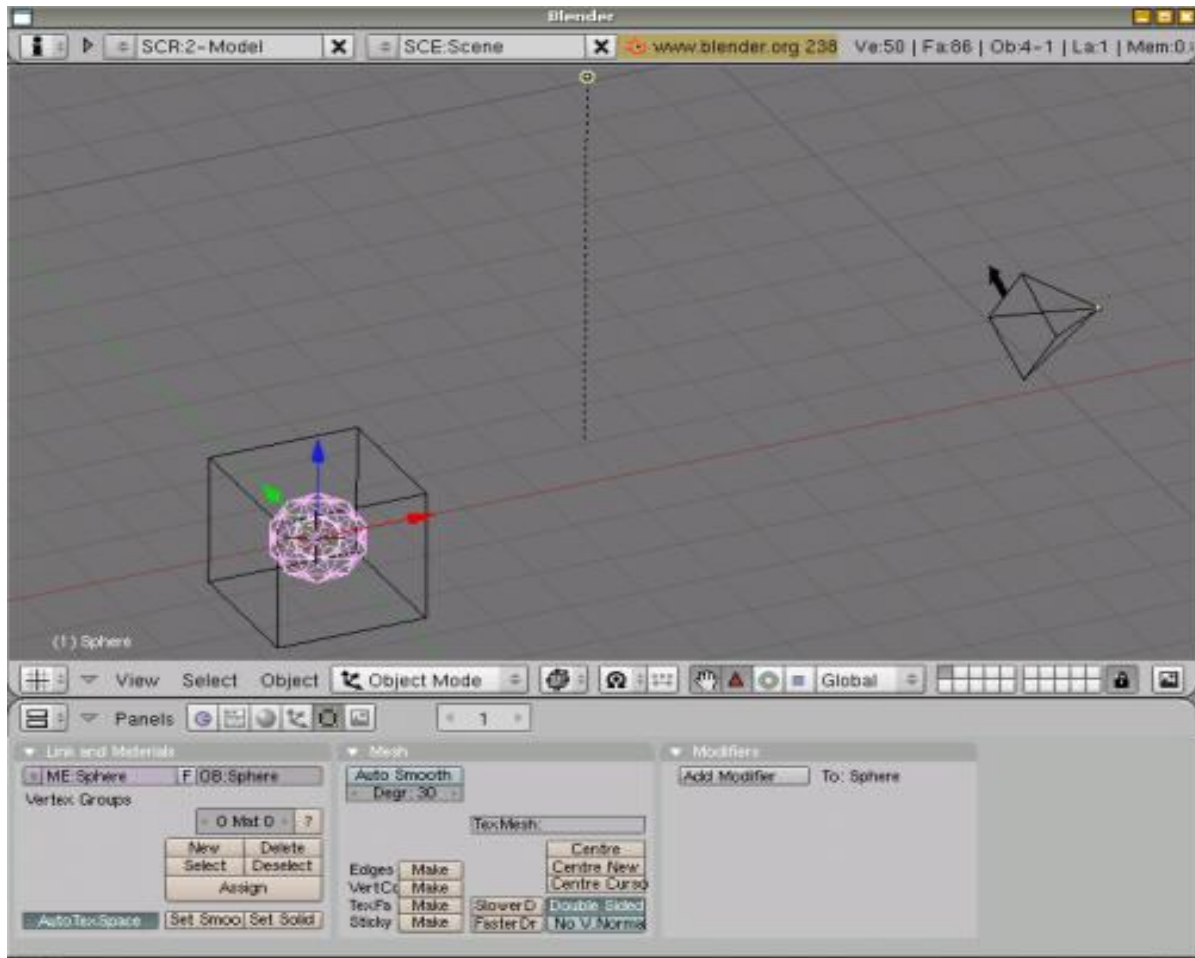


Рис. 6 - Переключение режимов

Нажмите клавишу S для перехода в режим скалирования (изменения размера). Переместите мышку немного внутрь, пока сфера не окажется полностью в ящике (рис. 6). Вы можете нажать Esc или правую кнопку мышки (RMB) в любое время для отказа от операции.

Теперь переключитесь в объектный режим (выделенная кнопка слева), и выберите закладку "fluid simulation" (она крайняя справа). Затем нажмите Enable и Fluid (рис. 7). Это означает, что при моделировании сфера будет полностью заполнена жидкостью.

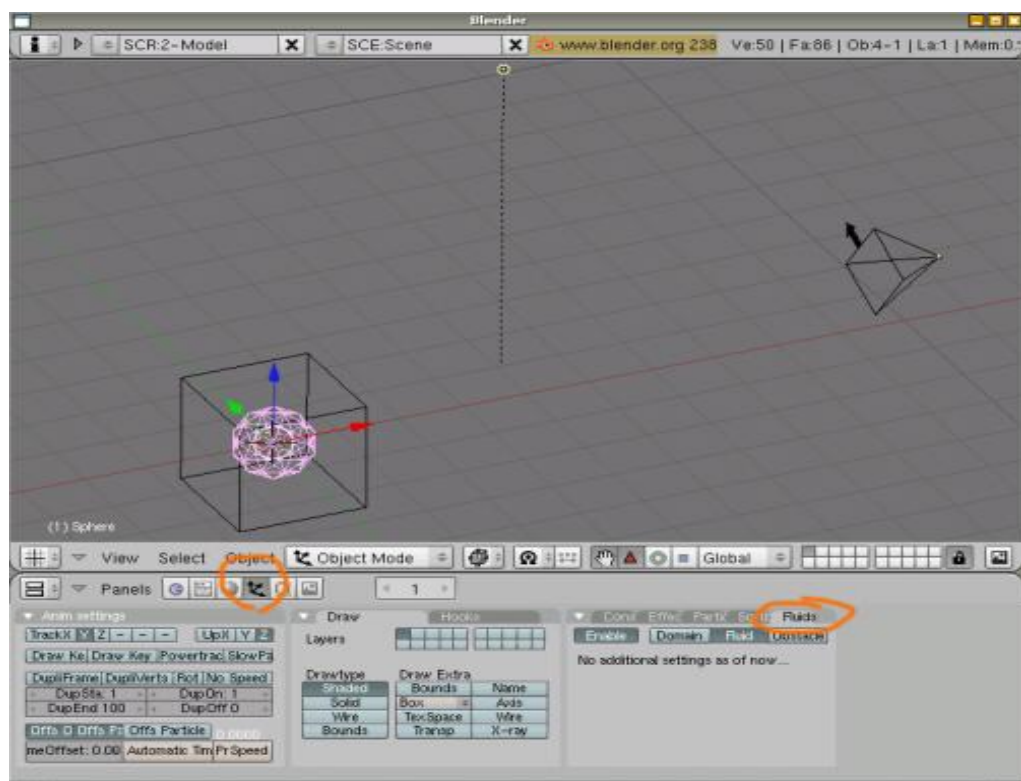
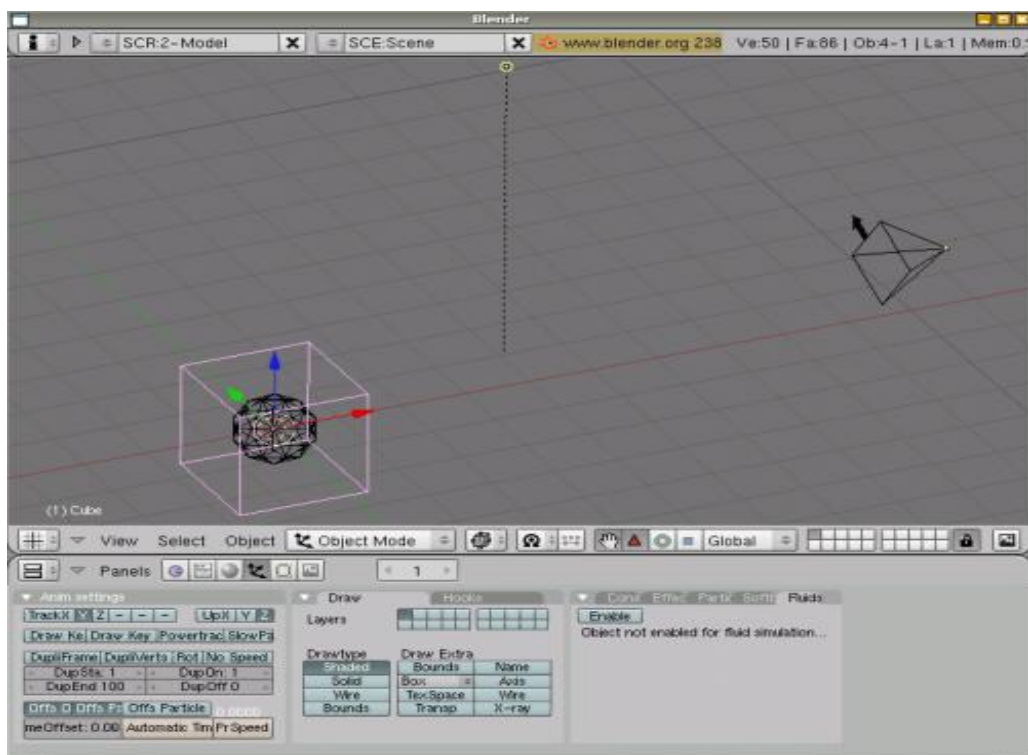


Рис. 7 - Параметры сферы

Выберите куб снова, кликая RMB около одного из его краев. Грани куба станут розовыми как только он будет выбран (рис. 8), и кнопки ниже



изменяться соответственно.

Рис. 8 - Выбранный куб

Закладка "fluid simulation" всё ещё будет отображена (рис. 9). Также разрешите "fluid simulation" для куба, и выберите Domain. Появятся дополнительные настройки, но Вы можете оставить их как они установлены

сейчас. "Domain" является регионом, где будет выполнено моделирование жидкости - всё за его пределами будет проигнорировано, и жидкость не может вытечь из региона. К настоящему времени границы ящика использованы, несмотря на то, что реально они не сформированы.

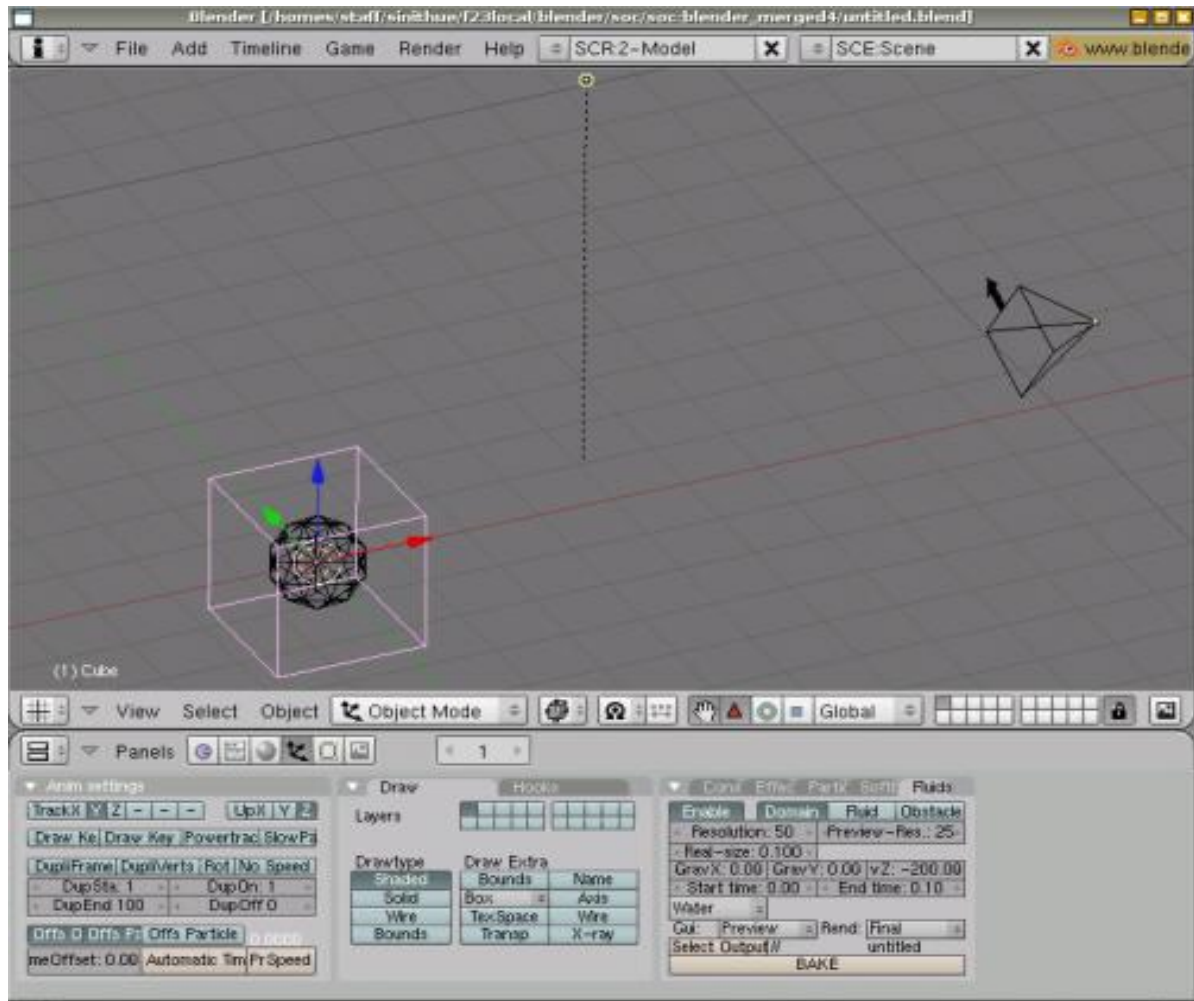


Рис. 9 - Изменение параметров

Прежде, чем выполнить моделирование, Вы вероятно захотите уменьшить количество кадров анимации. Нажмите слева выделенную кнопку (рис. 10) для переключения в Scene настройки сцены. Нажмите клавишу Shift и кликайте мышкой в поле End на закладке Anim. Поле выделится красным. Удалите число 250 и введите приблизительно 50.

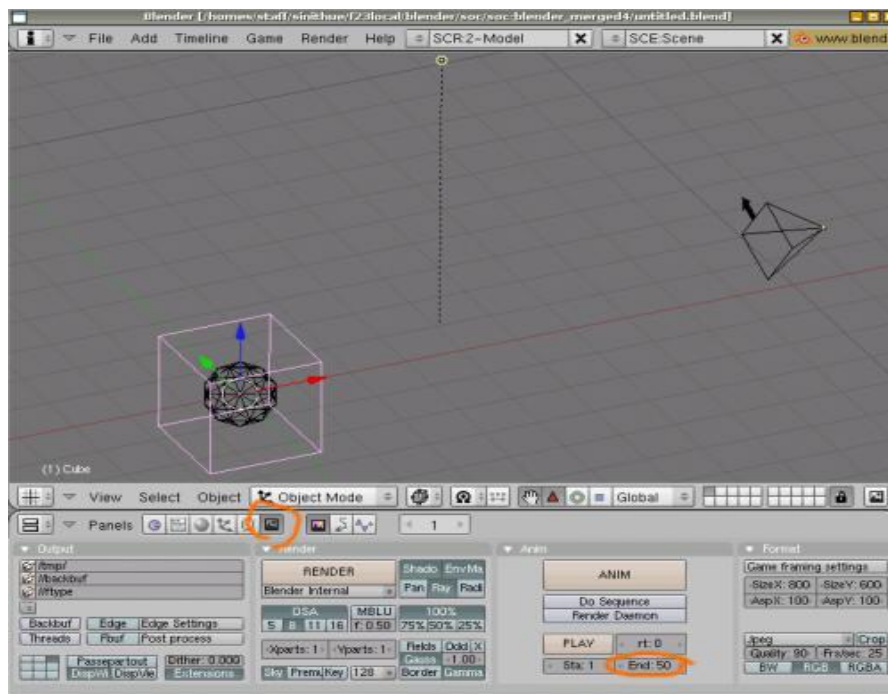
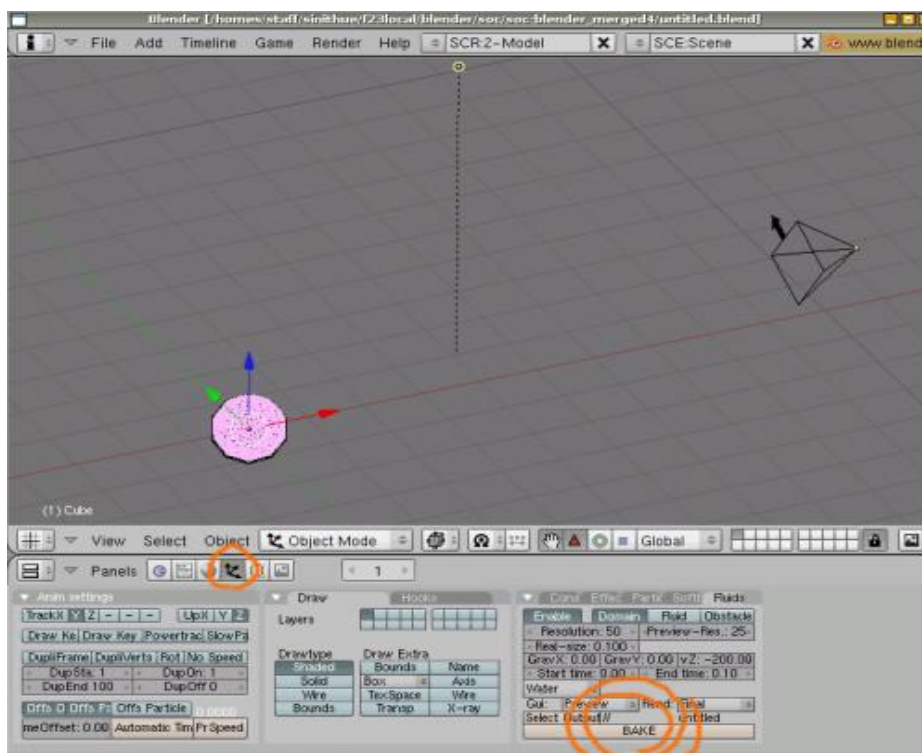


Рис. 10 - Задание количества кадров анимации

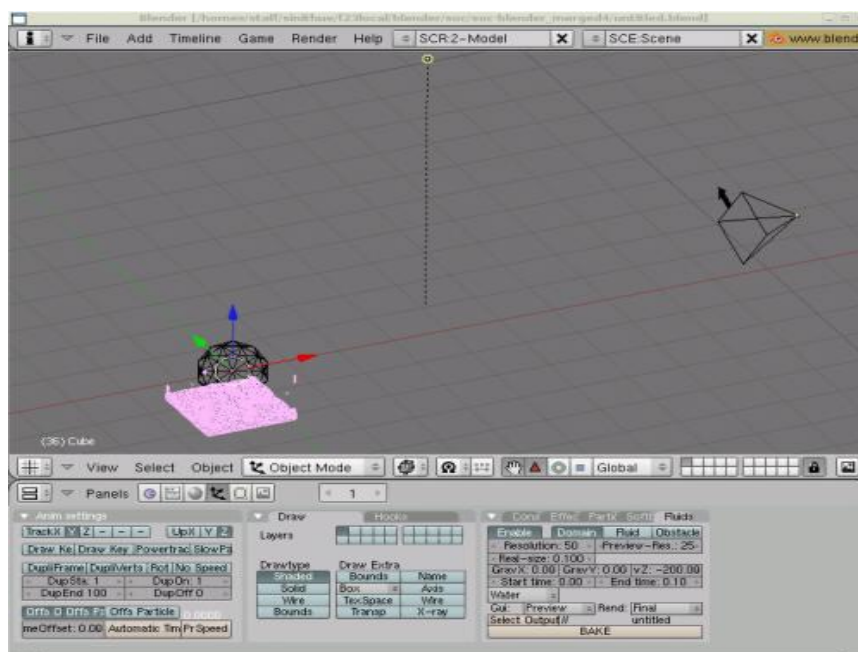
Куб должен все ещё выбран, вернитесь к объектному виду, и нажмите кнопку **BAKE** для старта моделирования. Blender "замерзнет", но курсор мышки начнет считать до 50. Это может занять несколько минут, в зависимости от мощности вашего компьютера. Когда моделирование будет завершено, Blender должен немедленно спрятать доменный объект, и показать первый кадр моделирования. Так как сфера была использована для моделирования жидкости, по завершению она выделится розовым цветом,



как показано (рис. 11).

Рис. 11 - Старт моделирования

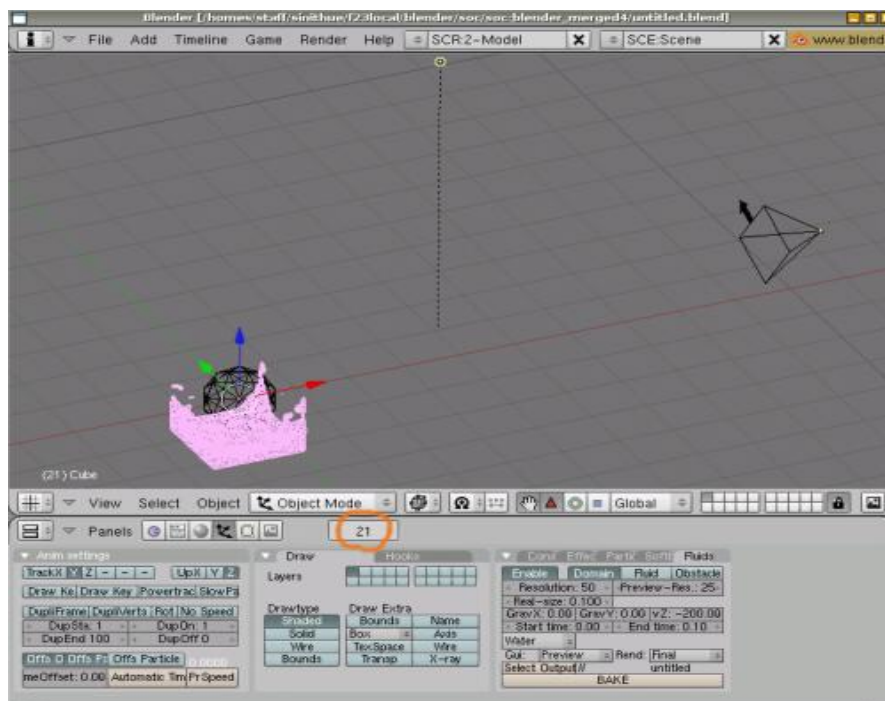
Теперь Вы можете просмотреть анимацию, нажимая Alt + A, когда мышка находится в 3D окне. Курсор мышки начнёт считать кадры снова и Вы должны увидеть как капля падает и плещется на полу (рис. 12). Нажмите



Escape для остановки анимации.

Рис. 12 - Визуальный вид анимации

Это вернет вас обратно в 1 кадр. Нажмите ↑ два раза, для перехода в 21



кадр (это видно на выделенной кнопке, рис. 13).

Рис. 13 - Раскадровка

Прежде чем рендерить, мы должны отделаться от сферы, но вместо удаления, мы спрячем ее в другом слое. Выберите сферу RMB, и нажмите клавишу M. Откроется маленькое окно (рис. 14). Выберите вторую кнопку с левой стороны вверху, и нажмите ОК. В 3D окне видны только те объекты, слои которых разрешены - вы можете включать и выключать их удерживая клавишу Shift, а также нажимая кнопки внизу в закладке Draw.

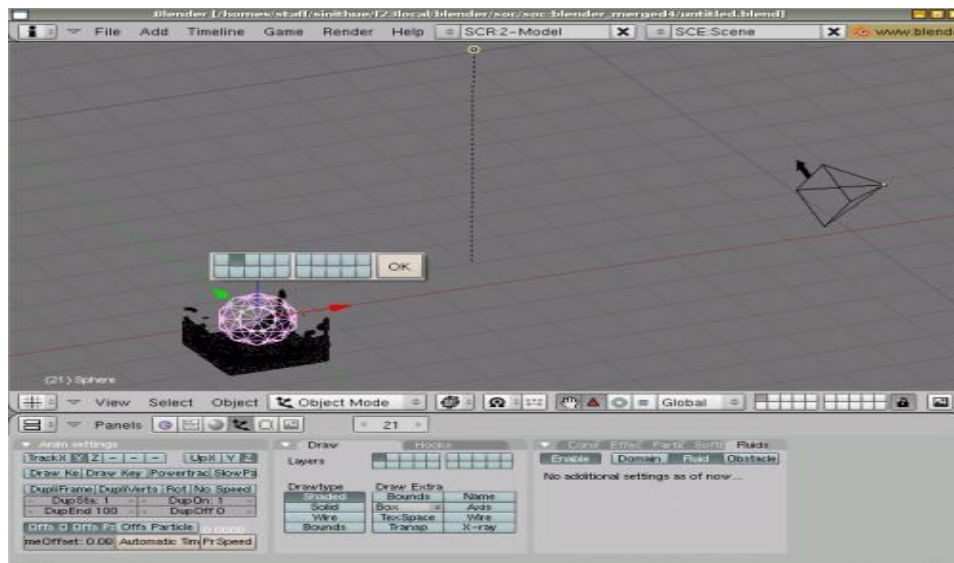
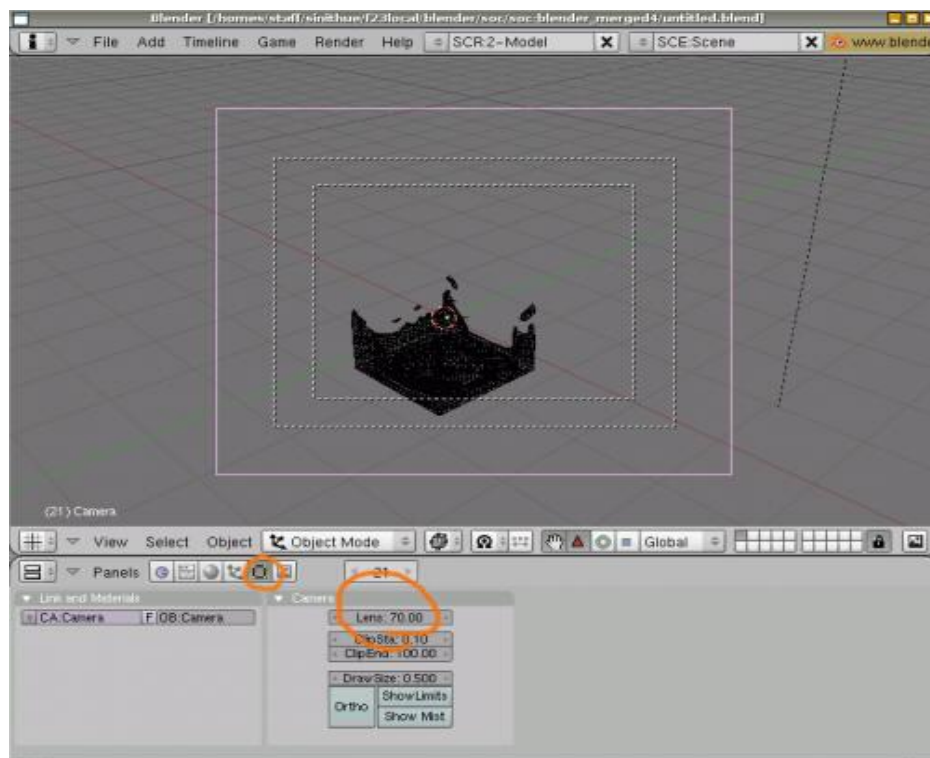


Рис. 14 - Слои объектов

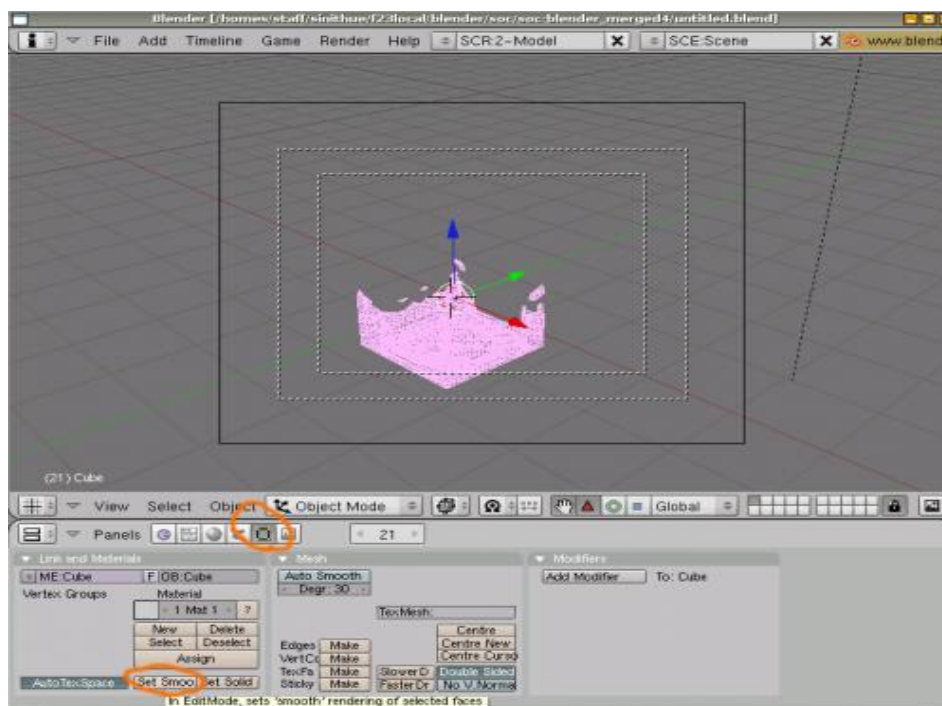
Снова переместите мышку в 3D окно и выберите камеру RMB кнопкой. Нажмите NumPad 0. Это переключит на камеру (NumPad 1, NumPad 3 и NumPad 7 выбирают передний обзор, боковой и вид сверху). Выберите



Editing в панели кнопок, и измените поле Lens на 70 (рис. 15):

Рис. 15 - Работа с камерами

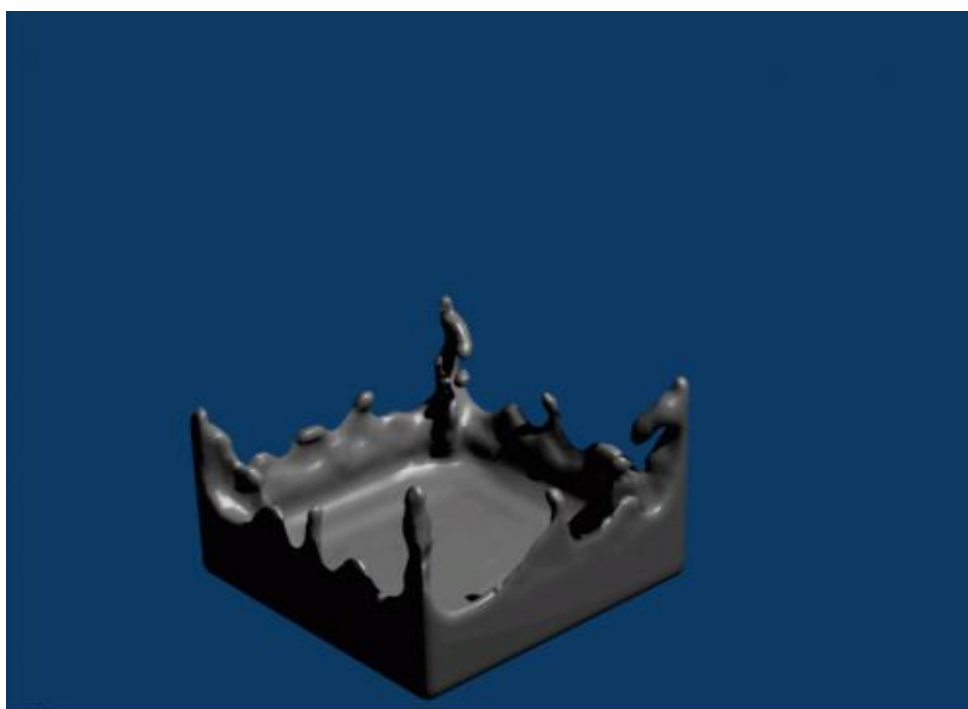
Ну и последний шаг - выберите поверхность воды снова, и нажмите Set



Smooth внизу (рис. 16). Это сделает поверхность воды сглаженной.

Рис. 16 - Параметр сглаживания

В заключение нажмите F12 (или выберите кнопку Scene и нажмите RENDER), чтобы получить изображение текущего кадра. Откроется новое окно, и покажет подобное изображение (рис. 17). Должна получиться серая



капля на синем фоне.

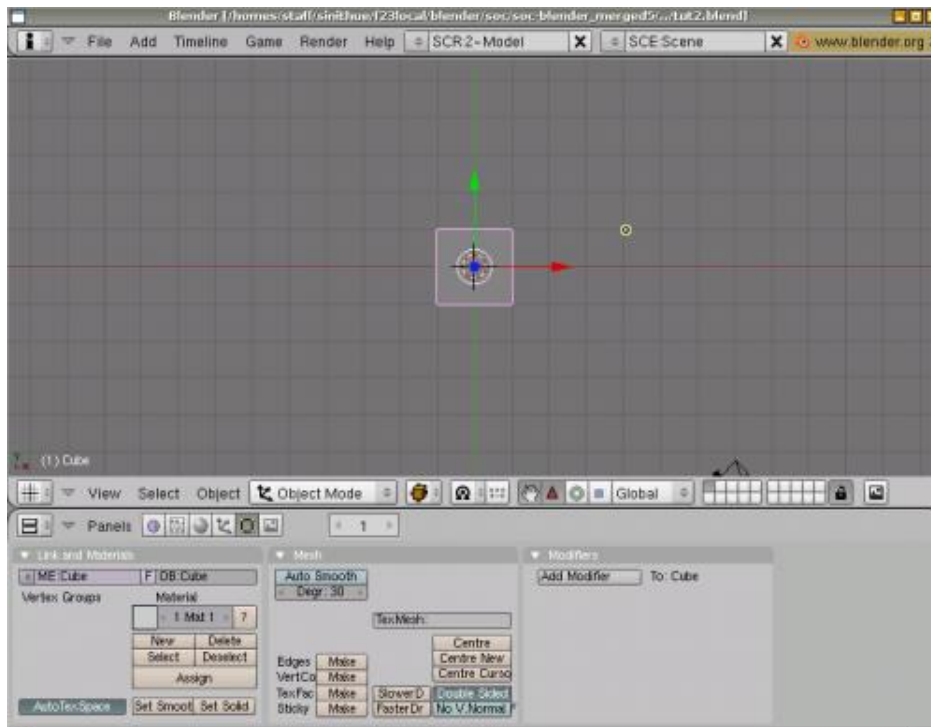
Рис. 17 - Результат рендера

КОНТРОЛЬНЫЕ ВОПРОСЫ:

- 1) Что такое Blender?
- 2) Перечислите функции Blender пакета?
- 3) Можно ли хранить несколько сцен в едином файле?
- 4) Для чего используется средняя кнопка мыши?
- 5) Какие объекты входят в стандартную начальную сцену Blendera?
- 6) Что такое режим скалирования и как туда перейти?
- 7) Что такое Domain?
- 8) Как изменить количество кадров в анимации?
- 9) Как просмотреть созданную анимацию?
- 10) Что такое Set Smooth и где его найти?

Получение простого стекла

Теперь мы смоделируем простое стекло, (очень похоже на кружку) используя один из новых модификаторов холда. Мы начнем с этого стандартного куба, который имеем вначале, при запуске Blendera (рис. 1).



Сразу же сохраним проект, например, под именем "tut2.blend".

Рис. 1 - Начальное окно

Переключимся на вид спереди (NUMPAD 1), перейдем в режим редактирования (TAB) и переименуем объект в "Glass" (рис. 2):

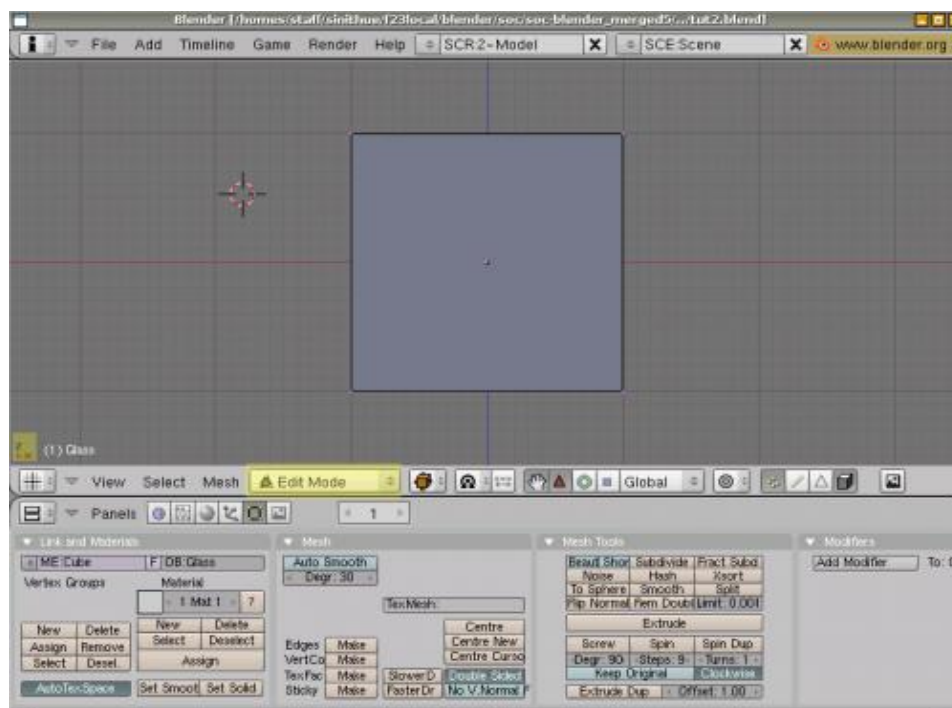
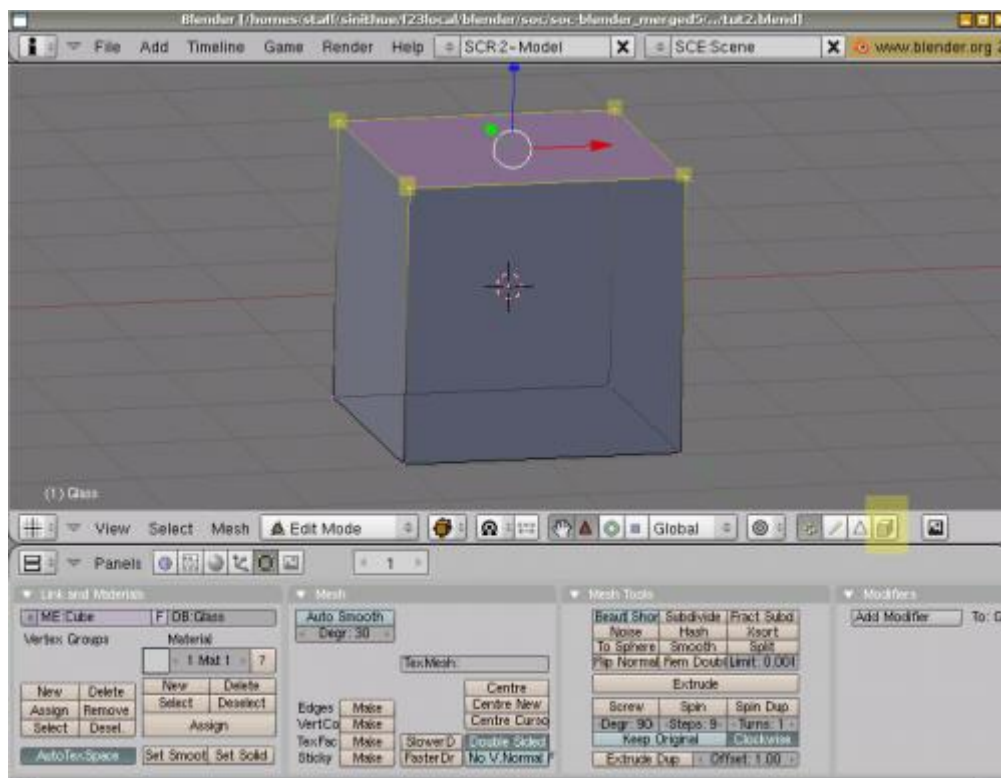


Рис. 2 - Вид куба спереди

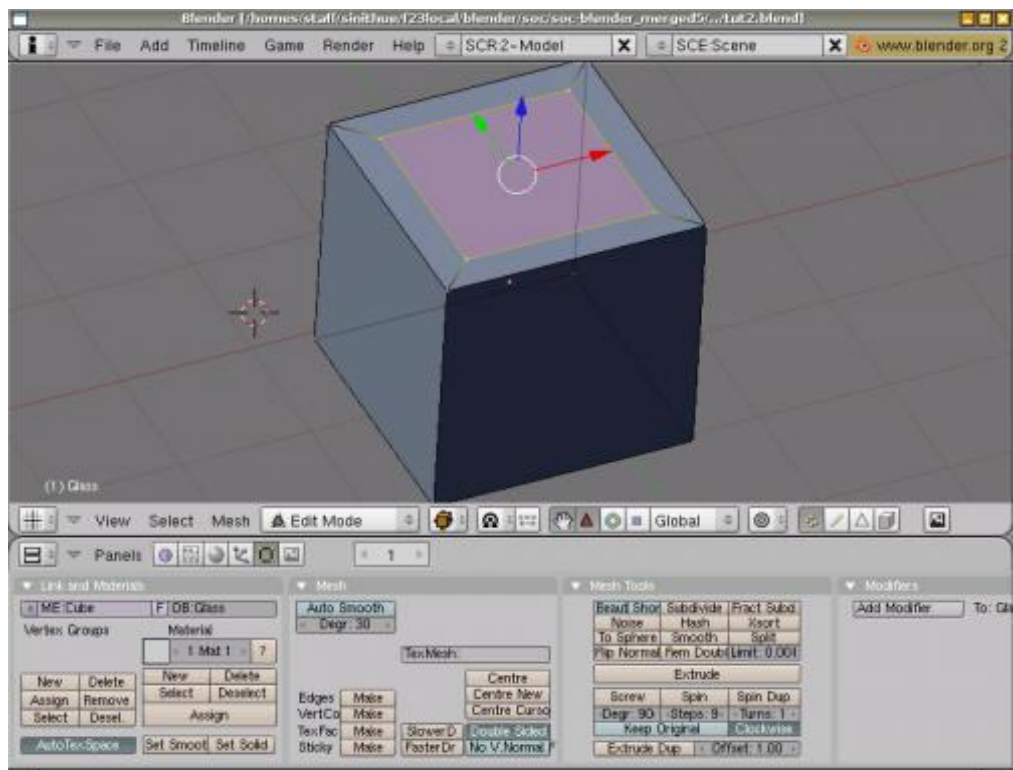
Переключите выбор вида и выберите режим граней клавишей В.



Вращайте обзор для получения лучшего вида (рис. 3):

Рис. 3 - Улучшение обзора

Экструдировать верхнюю поверхность нажав клавишу E, выберите регион и сразу нажмите "enter" для размещения новой поверхности в позиции старой. Теперь с выбранной новой поверхностью нажмем клавишу S и



изменим размер немного внутрь (рис. 4):

Рис. 4 - Добавление новой поверхности

Нажмем опять клавишу E, с этим же регионом, для нового экструдирования, и переместим немного вниз. Переключимся на просмотр сбоку и переместим новую поверхность вниз пока у вас не получится равное расстояние со всех сторон куба (рис. 5):

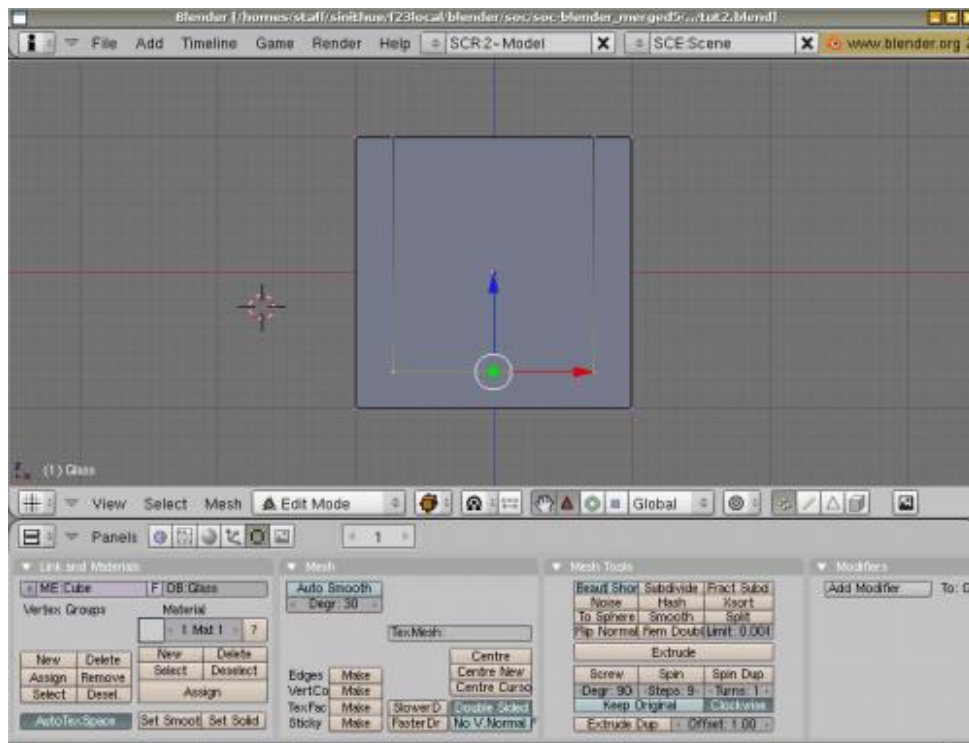
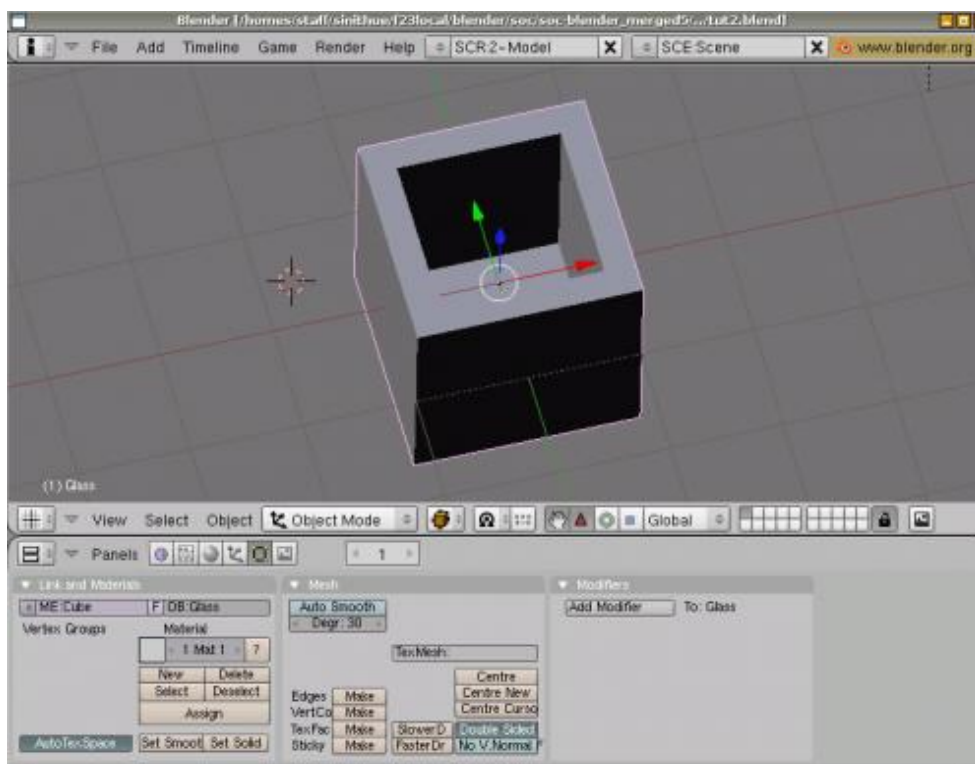


Рис. 5 - Перемещение поверхности в кубе

Покинем режим редактирования нажатием клавиши TAB , у вас есть



теперь отличный полый куб (рис. 6):

Рис. 6 - Результаты экструдирования - полый куб

В панели "Modifiers", нажмем кнопку "Add Modifier" и выберем модификатор "subsurf" (рис. 7):

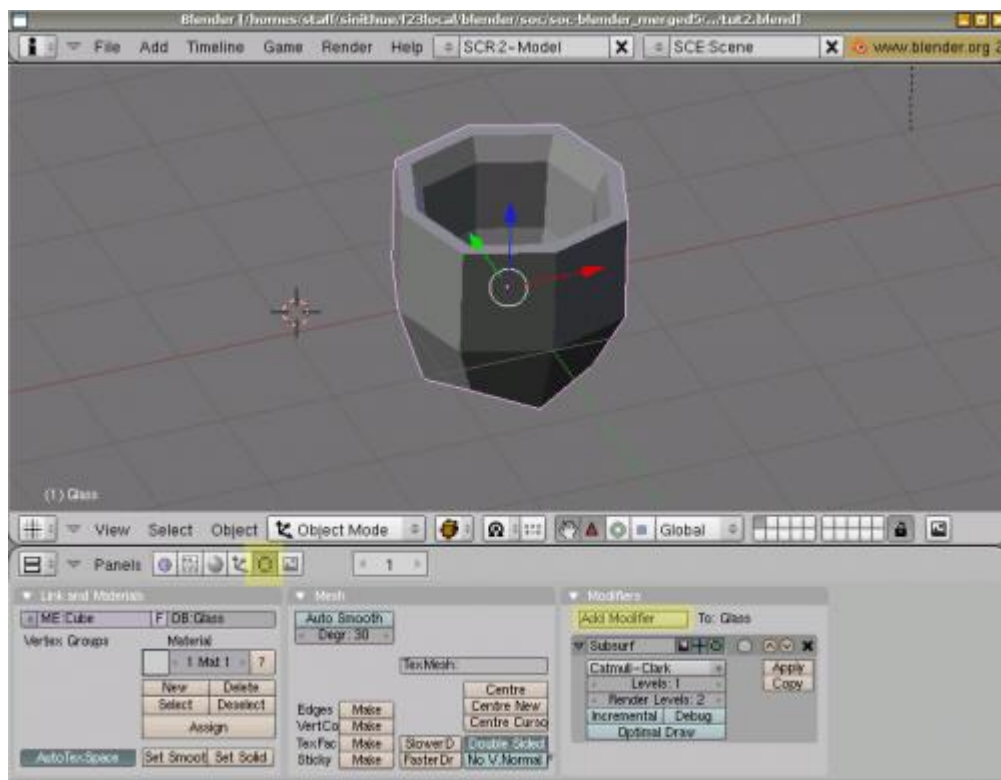


Рис. 7 - Наложение модификатора

Увеличим "Levels" до 2, и "Render Levels" до 4. Также установим отображение поверхности в режим Smooth (рис. 8). Это будет стекло, (в

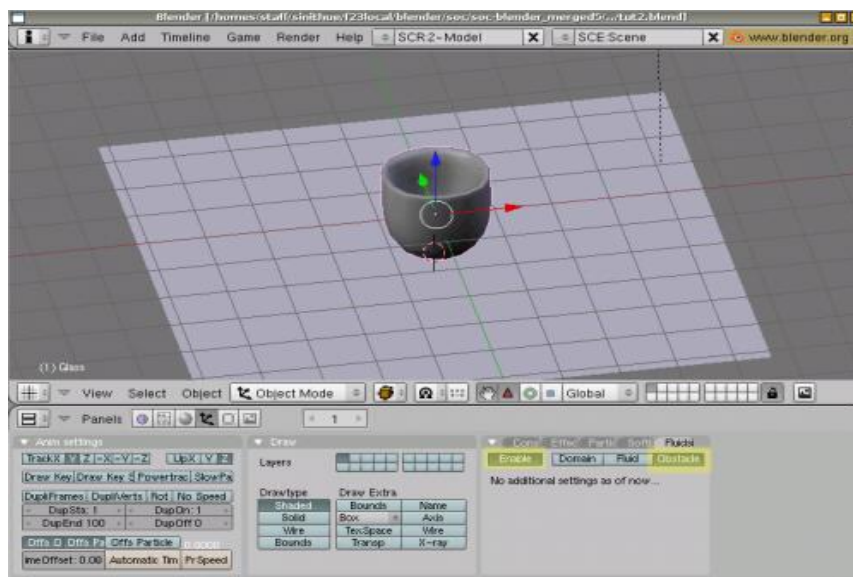


реальности эта вещь конечно упадет) теперь вы можете заполнить её кое-чем.

Рис. 8 - Наложение сглаживания

Установки моделирования жидкости

На закладке "Fluid Simulation" нажмем кнопки "Enable" и "Obstacle" для нашего стекла и добавим меш "plane" ниже объекта "glass" (лучше это делать при виде сверху NUMPAD 7 для более точного выравнивания). Не забывайте помещать красно-белый маркер в нужное место для правильной установки



объектов, и назовем, например, "Floor" (рис. 9):

Рис. 9 - Установка объектов

Переключитесь в теневой режим клавишей Z , и добавьте куб вокруг центра стекла. Масштабируйте до 4 клеток на сторону. В режиме бокового просмотра выровняйте низ куба с полом ("floor"), и назовите "Domain". В заключение, на закладке "Fluid Simulation" нажмите кнопки "Enable" и "Domain", и установите сглаживание "Set Smooth" в "Editing buttons" (рис. 10)(Примечание: в данном случае куб является главным, но чаще всего так не должно быть.)

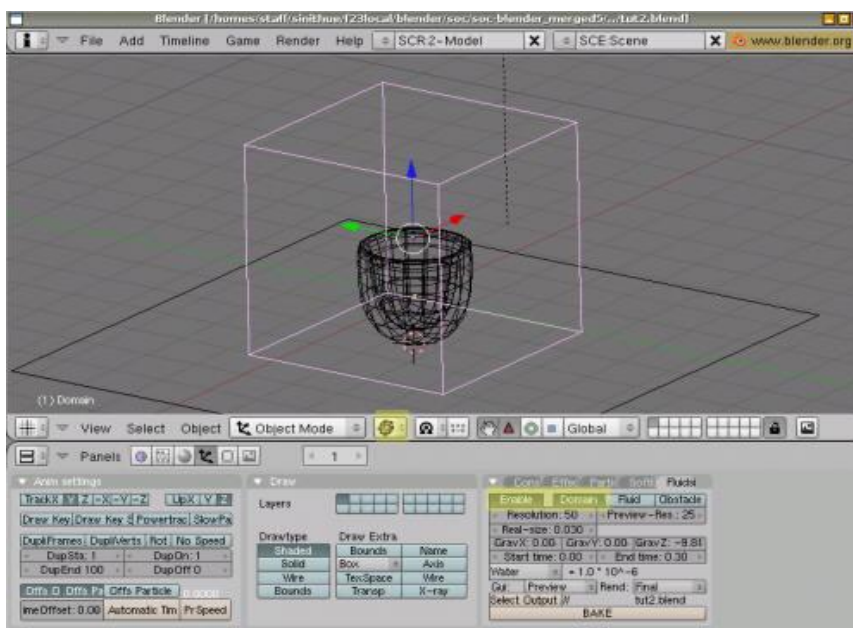
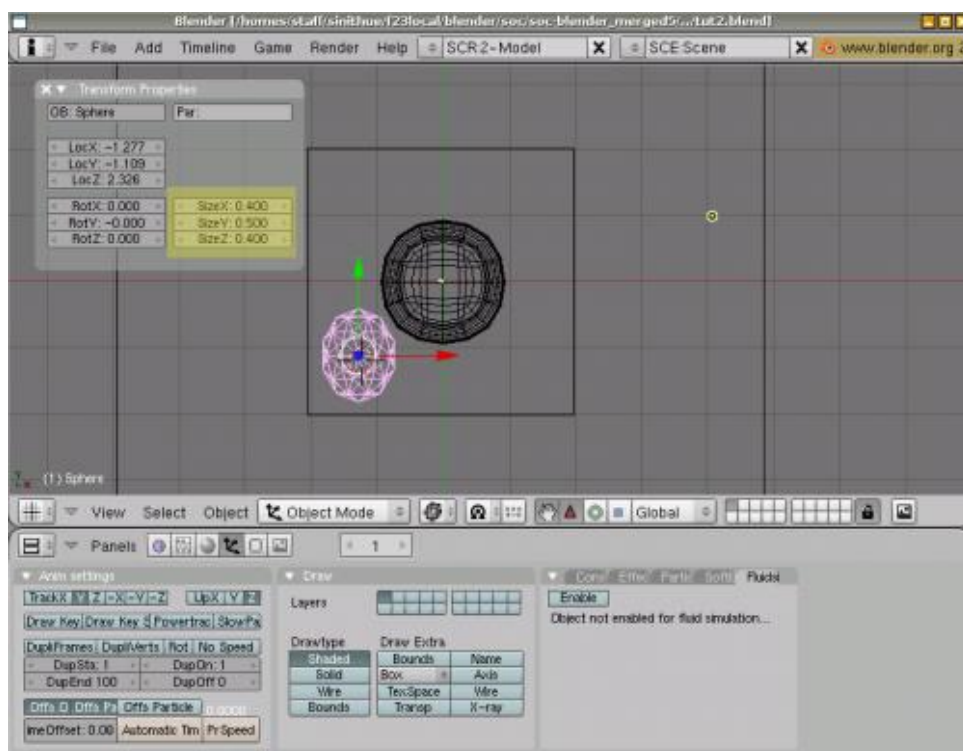


Рис. 10 - Добавление куба

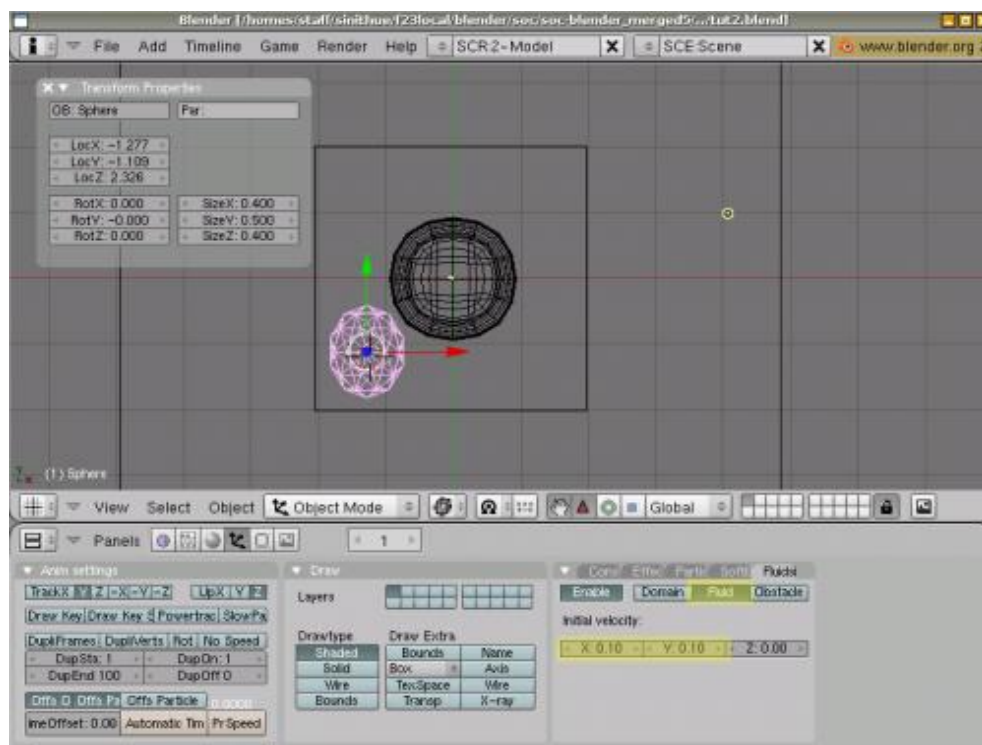
В виде сверху добавьте сферу в нижний левый угол куба. Вызовите окно "Transform Properties" клавишей (N) и установите значение между 0.4 и



0.5 в каждом измерении (рис. 11):

Рис. 11 - Добавление сферы

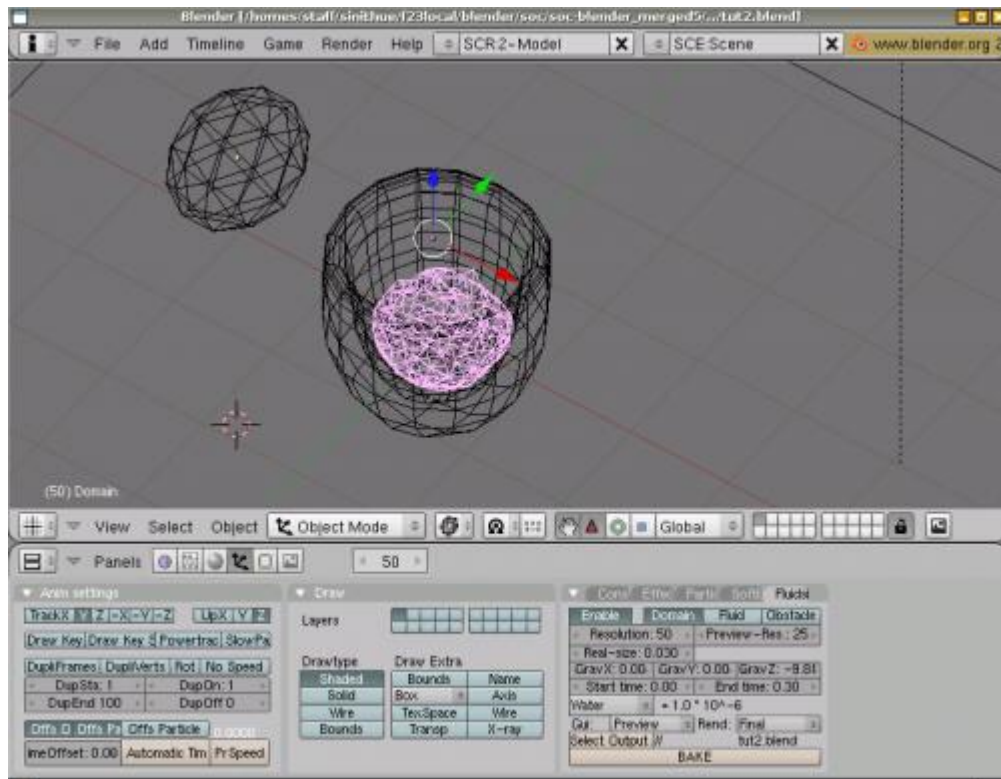
Переименуйте сферу в "Drop" и нажмите кнопку "Fluid" в закладке "Fluid Simulation". Капля ("Drop") должна быть выше стекла ("Glass"), Установите



значения "Initial velocity" - 0.1 для X и Y (Z оставим как есть, рис. 12):

Рис. 12 - Установка параметров

Теперь все готово для моделирования: Для начала установите количество кадров, например, 50. Затем выберите объект "Domain", и нажмите клавишу "Bake" для тестового моделирования, чтобы проверить, действительно ли капля расплескивается в нашем стекле как мы хотим. Проиграйте анимацию клавишами (ALT+A) и рассмотрите под разными углами. Меняйте параметры скорости, пока Вы не получите нужный



результат (рис. 13):

Рис. 13 - Тестовое моделирование анимации

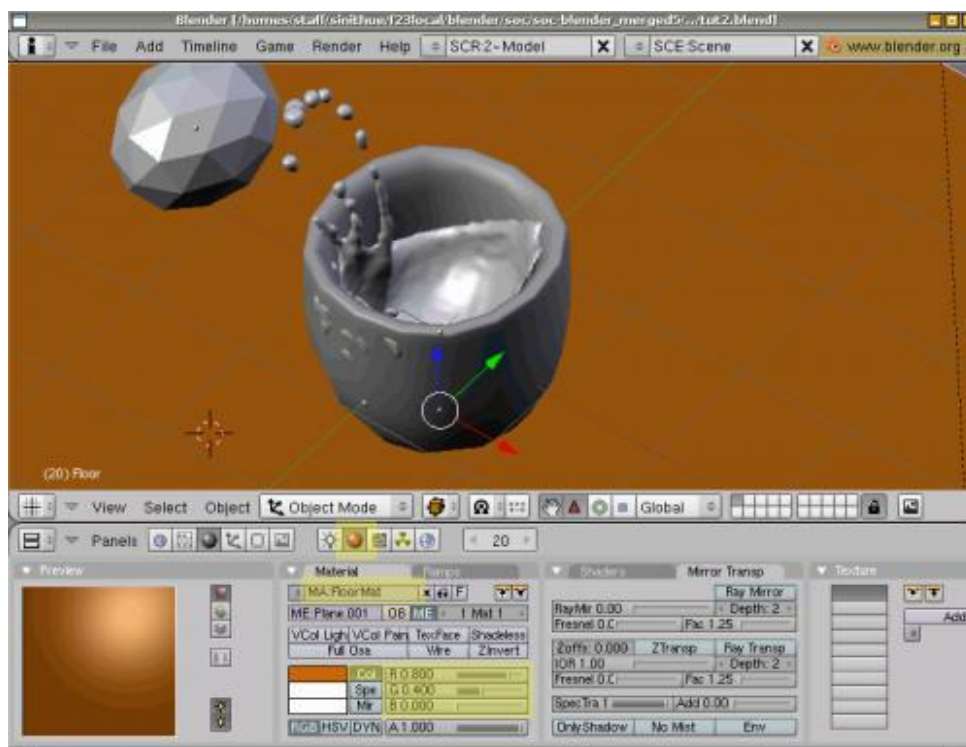
Как только все будет нормально, увеличьте "Resolution" до 128 (это потребует около 300Мб памяти), и возможно также "Preview Resolution" до 30-35 для получения более детализированной картинки. Теперь процесс моделирования может затянуться (вероятно на несколько минут). Проиграйте анимацию снова. Переключитесь в теневой режим клавишей (Z), и посмотрите отдельные кадры, например 20, нажимая →. Выберите в "GUI Display" установку "Final" для объекта "Domain" чтобы проверить как это смотрится при рендеринге. Убедитесь, что "smooth shading" включен для объекта "Domain" в "Editing buttons" (F9). Посмотрите ещё некоторые кадры, затем установите снова режим "Preview". Если Вам не понравился результат, лучше снова уменьшить "resolution" и, например, изменить форму капли и начальную скорость ("initial velocity"), и пробуйте ещё разные варианты моделирования, пока не удовлетворитесь результатом (рис. 14). Не беспокойтесь об оригинальном меше капли, что ещё видна на этом изображении, её не будет видно в конечном результате.



Рис. 14 - Выбор параметров капли

Установки материала

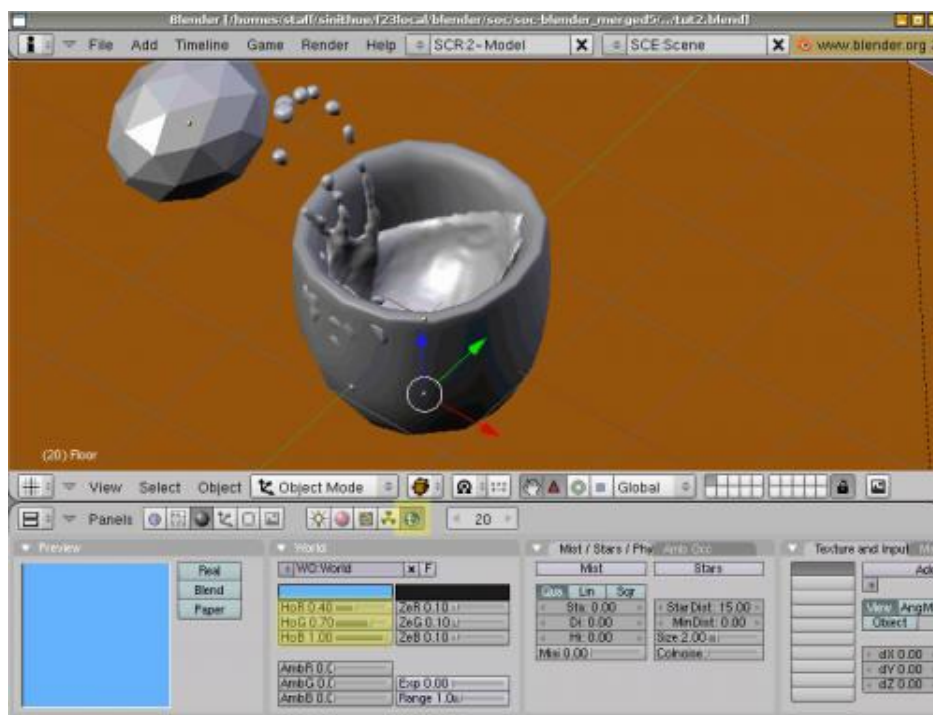
Для следующего, выставьте "Frame" где-нибудь посередине для просмотра и капли, и стекла (например 20). Если Вы отрендерите сцену теперь, это всё будет смотреться изрядно серым и скучным. Выберите пол ("floor"), нажмите кнопку Shading (F5). Добавьте новый материал ("add new"), и измените



"diffuse color" на (0.8, 0.4, 0). Итог будет таким (рис. 15):

Рис. 15 - Добавление материалов

Давайте сделаем фон более ярким, выберем "world buttons" и изменим



цвет как на рисунке в (0.4, 0.7, 1.0). Получится так (рис. 16):

Рис. 16 - Редактирование фона

Выберем объект стекло ("glass"), переключимся обратно на материалы и переименуем этот материал в "Glass" (рис. 17). Активизируем "Ray Mirror", установим величину "Ray Mir" на 0.15 и увеличим "Depth" до 7. Затем активизируем "Ray Transp" установим "IOR" на 1.5 (это правильные физические величины для стекла) и установим ползунок "Alpha" на 0. Примечание - сравнивая картинку, вы заметите, что объект "Domain" перед этим был умышленно забыт.

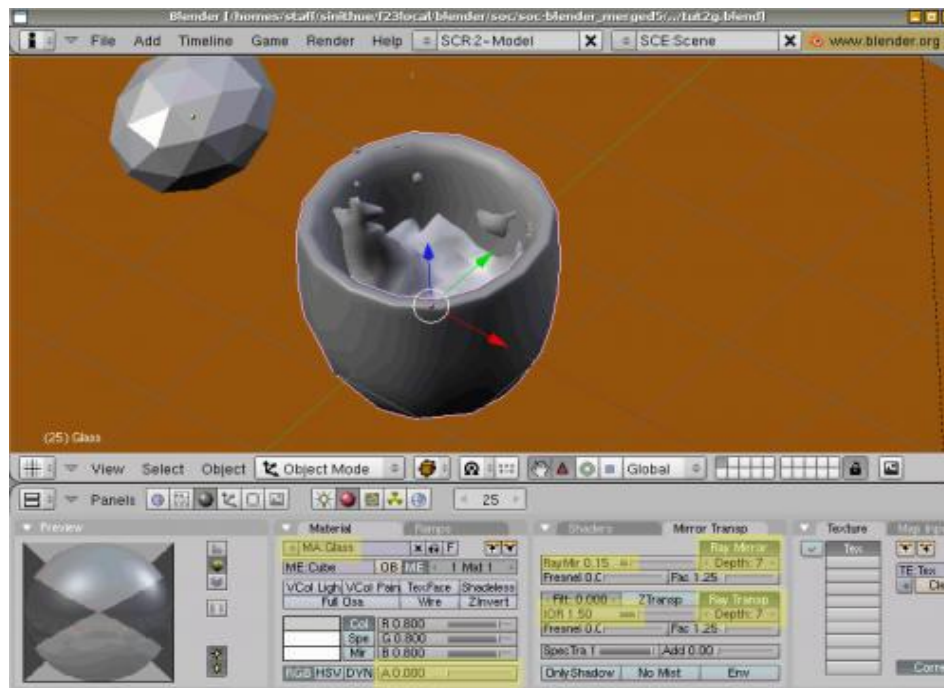
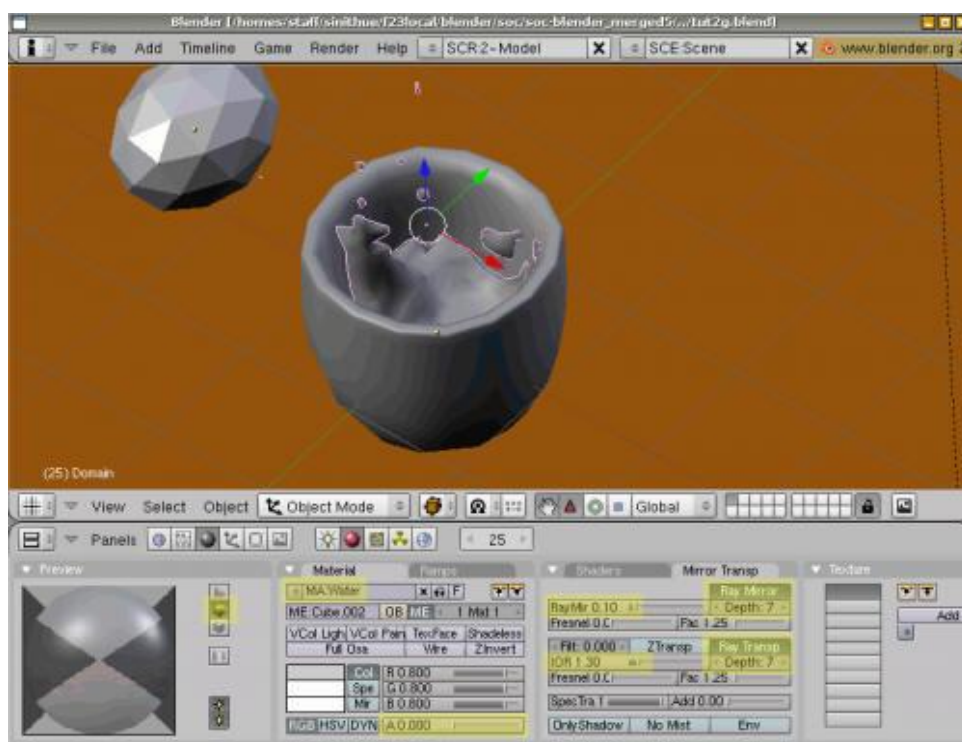


Рис. 17 - Установка стекла

Для жидкости почти то же - создаём новый материал, называем "Water", и устанавливаем те же величины как и у стекла, исключение для "IOR", для



него выставляем 1.3, и можно установить "Ray Mir" на 0.1 (рис. 18):

Рис. 18 - Установка жидкости

При виде сверху, делаем две копии источника света клавишами (SHIFT-D), Вращайте вид немного, и расставьте их вокруг стекла так, как показано на рисунке. Затем включите "Ray Shado" для каждой лампы в "Lamp buttons" (рис. 19):

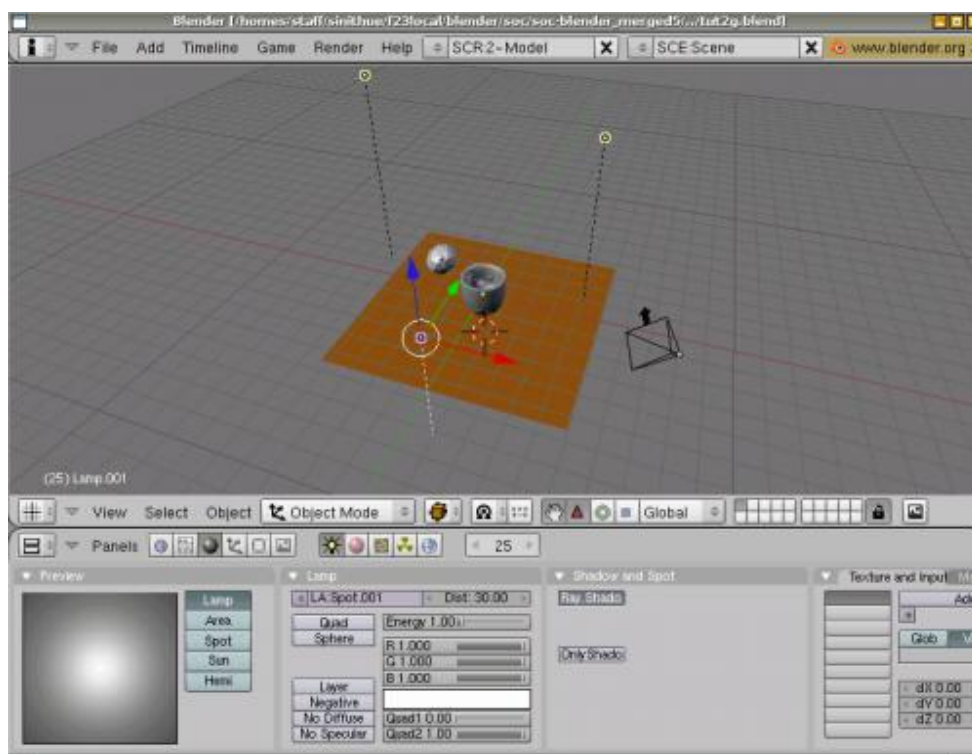
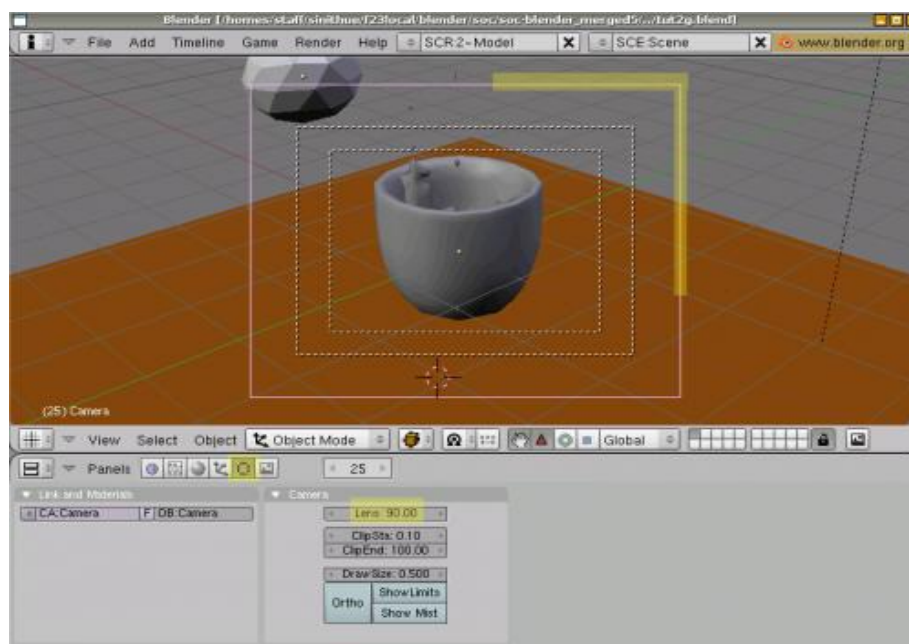


Рис. 19 - Добавление источников освещения

Выберите просмотр камерой, клавишей NUMPAD 0 , и ПКМ по чёрной рамке выберите её. Выставьте размер вида под размер стекла и установите величину "Lens" на 90 в "Editing buttons" (Shift+click на поле для ввода числа), используя клавишу G передвиньте камеру для лучшего обзора, и



оригинальный меш капли больше не будет виден (рис. 20):

Рис. 20 - Работа с визуализацией

Финальный рендеринг близок! Нажмите F12 для рендеринга или перейдите в "Render buttons", отрегулируйте "anti-aliasing" и "resolution" как вы любите, и нажмите кнопку RENDER (рис. 21). Если результат вам нравится, используйте Save Image... (или F3) для сохранения полученной

картинки. С кнопкой "ANIM" вы можете отрендерить также анимацию (для этого Вы может захотите уменьшить разрешение).



Рис. 21 - Результат рендера

Дополнительно - для повышения реализма, вы можете изменить "Point Lamps" на "Area Lights". Установите направление света на стекло вращая их, увеличьте "Size" на 1 или 2, и убавляйте "Engergy" пока сцена не будет освещена правильно (проверьте это делая пробный рендеринг, в GUI освещение сцены отображается неверно). Наконец увеличьте количество "Samples" для каждой лампы, для создания мягких границ тени (рис. 22). Теперь рендер проходит гораздо дольше...

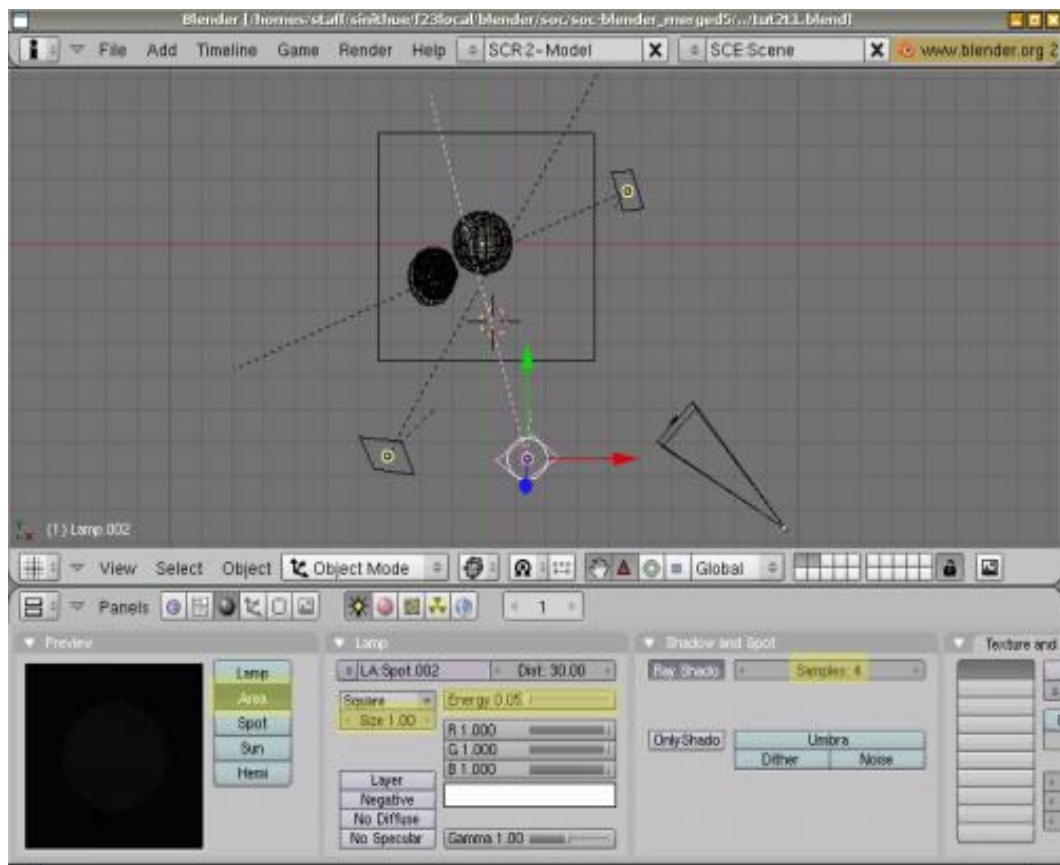


Рис. 22 - Дополнительные параметры освещения

Изменение настроек моделирования

Скажем, Вы решили в некоторый момент, что квадратный стакан и грязь интереснее, чем этот бокал с водой. Для этого, сначала выберите "Glass" и запретите "subsurf" модификатор. Также переключитесь обратно в режим "Set Solid" вместо "Set Smooth". Теперь оригинальная квадратная форма снова видна. Если вы прокрутите анимацию снова, то увидите странности, жидкость будет игнорировать изменения в форме, пока вы снова не нажмете кнопку "Bake" в закладке "Fluid Simulation" (рис. 23):

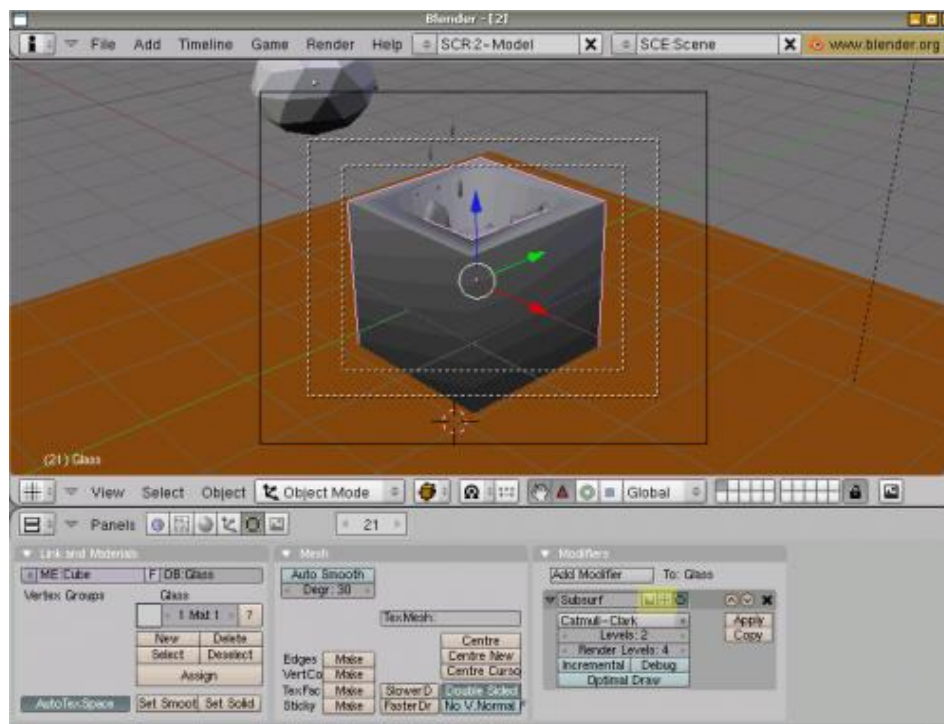
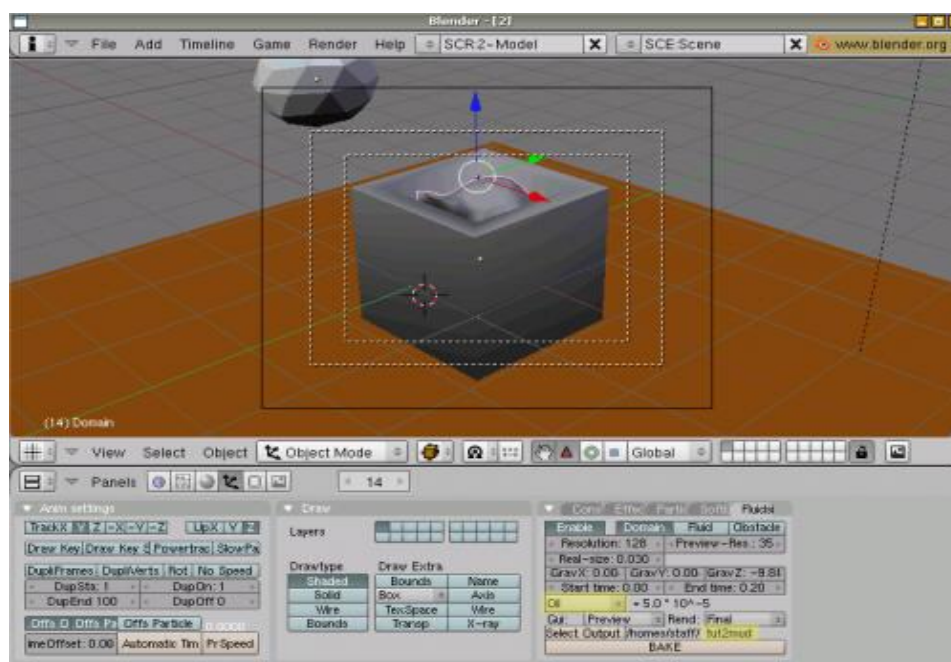


Рис. 23 - Переключение модификаторов

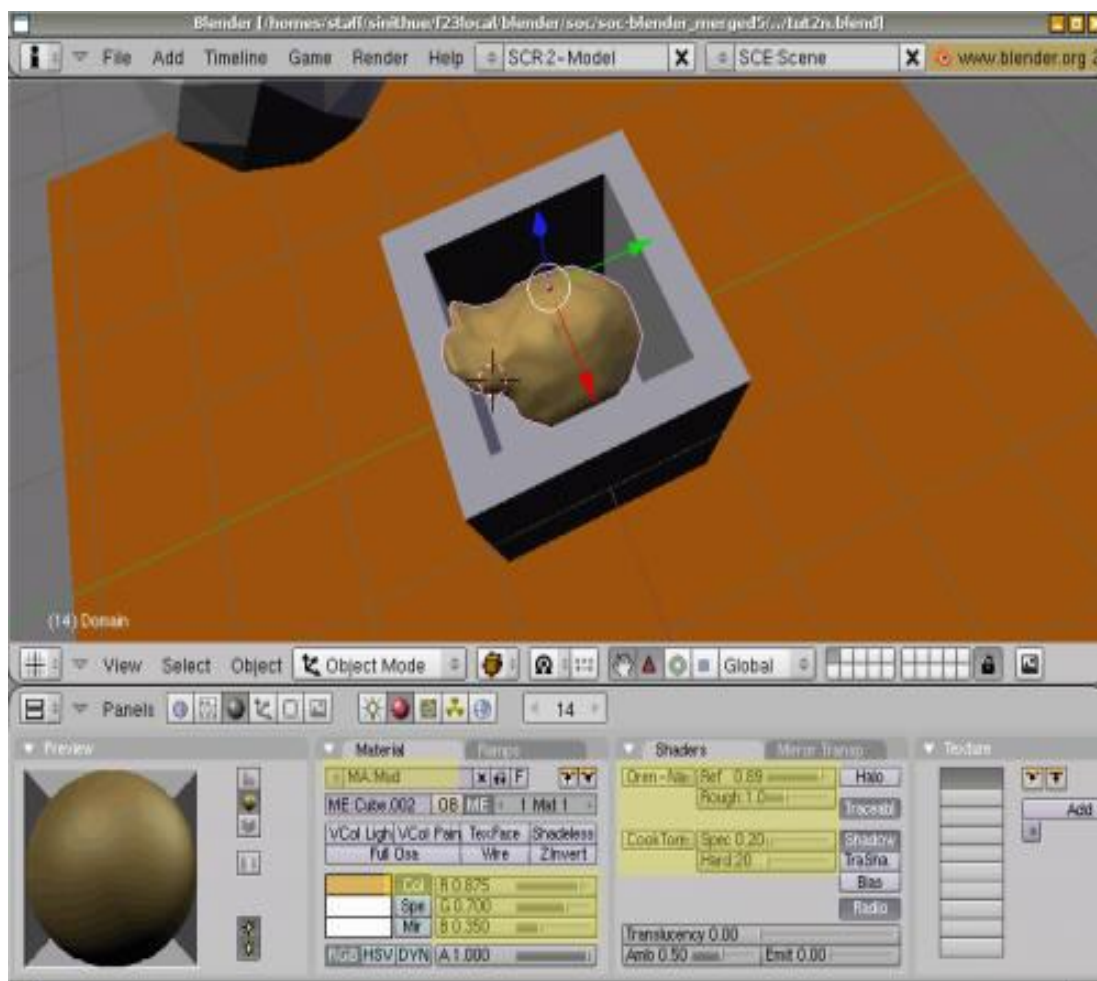
Чтобы сохранить оригинальную, подобную воде жидкость, будут установлены новые данные для грязи. Выберите объект "Domain", и нажмите кнопку Bake Dir. Выберите директорию для сохранения поверхности меша, и введите "tut2mud" в поле "filename", затем нажмите ENTER или Select Directory. Теперь измените вязкость ("Viscosity") для инициализации мёда ("Honey"), и нажмите БАКЕ. Префикс "tut2mud" теперь будет использован для моделирования мешей, и старые данные не будут перезаписаны (рис. 24). Примечание - хорошей идеей будет создание директорий для хранения данных о моделировании жидкостей где-нибудь рядом с вашими .blend



файлами.

Рис. 24 - Сохранение параметров жидкости для грязи

Чтобы сделать вид жидкости грязной, измените настройки материала. Создайте новый материал "Mud", используя выпадающее меню, и установите цвет напоминающий грязь. Установите ползунок "alpha" на 1.0, отключите "ray mirror" и "ray transparency" в случае, если они ещё включены. Вы также можете установить оттенок на тускло-коричневый, как показано на рисунке



25. Теперь вы можете запустить анимацию вновь...

Рис. 25 - Установка параметров грязи

Если вы захотите в некоторый момент вернуться к прежнему варианту с водой, просто выберите объект "domain" для жидкости, нажмите кнопку Bake Dir и выберите любую из оригинальных поверхностей или предпросмотр мешей. Они вероятно названы как "tut2.blend_OVCube_surface_...bobj.gz", или "tut2.blend_OVCube_preview_...bobj.gz". Выбор любого из них, извлечет правильный префикс в поле "filename", и после нажатия Select Directory оригинальный результат моделирования снова станет видимым. Это будет выглядеть как если бы вы включили модификатор "subsurf" снова для стекла (рис. 26):

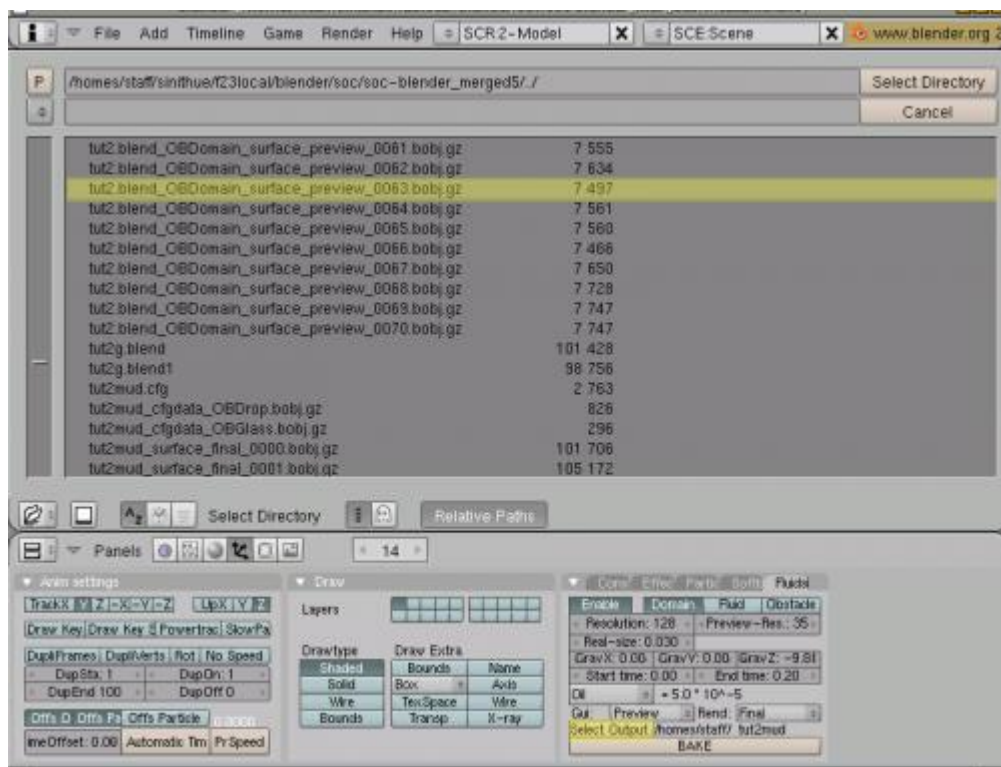


Рис. 26 - Смена результатов моделирования

- Важным эффектом являются видимые язвочки. Если у вас установлен YafRay, вы можете добавить источник "photon light" для исправления язв на изображении.
- YafRay также имеет более реалистичное отражение поверхности, в комбинации с красивым фоном или изображениями HDR, это даст хороший результат.
- Попробуйте установить более сложные препятствия и геометрию жидкости - максимальное разрешение моделирования на текущий момент ограничено опасностью повисания компьютера из-за большого использования памяти, но с самыми высокими установками будет довольно сложно моделировать.
- Выберите для объекта drop ("капля") Inflow вместо Fluid и запустите моделирование снова (возможно придется немного уменьшить размер капли, чтобы не было переполнения).

КОНТРОЛЬНЫЕ ВОПРОСЫ:

- 1) Как переключать выбор вида?
- 2) Что такое операция экструдирования?
- 3) Какой клавишей включается теневой режим?
- 4) Для чего используется кнопка Bake?

- 5) Влияет ли изменение параметра Resolution на изменение требуемого объема памяти?
- 6) Как добавить новый материал?
- 7) Что такое источники света?
- 8) Какие функции выполняет клавиша G?
- 9) Как отрендерить результаты работы?
- 10) Какие существуют способы повышения реализма картины? Влияют ли они на время рендера?
- 11) Какие способы воспроизведения анимации существуют?

Теперь мы поработаем с кривыми и немного с процедурными объектами.

Начиная всего с нескольких фигур, мы создадим анимацию поездки по заброшенной шахте. Используемый метод позволит вам легко менять форму пути, не заботясь о переделывании объектов, подобно созданию движения из ничего.

В виде сбоку добавьте кривую Безье (Bezier). Перейдите в окно EditButtons ([F9]), и нажмите кнопку [3D]. Это позволит вам перемещать вершины не только в пределах одной плоскости, но и во всем пространстве, тем самым сделает кривую действительно трехмерной.

Находясь в режиме редактирования кривой (клавиша [Tab] - все вершины должны быть видны), добавьте к ней несколько точек, кликая левой клавишей мыши с нажатой клавишей [Ctrl]. Когда вы закончите, у вас должно получиться примерно следующее (рис.1):

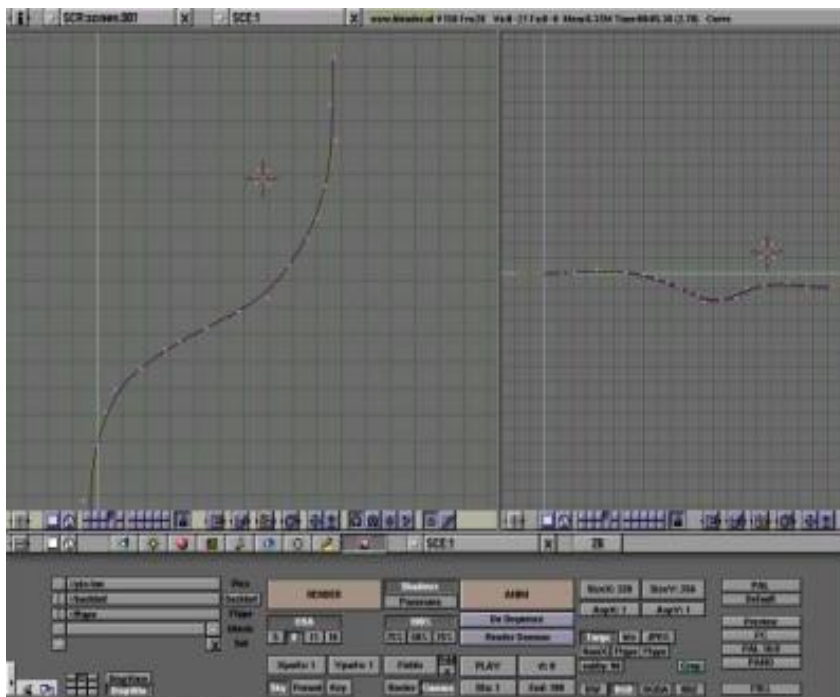


Рис. 1 - Вставка кривой Безье

Теперь мы прикрепим камеру к этому пути. Сначала, выберите камеру, нажмите клавишу [Shift] и выберите путь. Привяжите камеру к пути ([Ctrl]+[P]).

Выделите путь и перейдите в окно AnimButtons ([F7]). Теперь, когда вы нажмете в этом окне кнопку [CurvePath], камера будет двигаться вдоль пути. (Анимируйте, нажав клавиши [Alt]+[A], курсор мыши должен находиться в 3D-окне). Примечание: В зависимости от того, как была нарисована ваша кривая, когда вы нажмёте "CurvePath" камера может сместиться. Просто перетяните ее назад, на место с которого должно начаться движение.

Все это здорово, но не достаточно. Нам нужно, чтобы камера была направлена в правильном направлении! К счастью, для этого тоже имеется кнопка: [CurveFollow]. (Убедитесь, что вектор камеры показывающий ее верх направлен в правильном направлении!)

Примечание: Когда вы нажмете эту кнопку, по некоторым причинам камера повернется на 90 градусов. Как в примечании выше, просто исправьте это поворотом камеры так чтобы она была направлена в нужном направлении, и все будет ОК!

Создание рельсов

Рельсы будут созданы при помощи сечения плюс путь (лофтинг). В виде спереди, создайте форму для рельса - что-то похожее на это (рис. 2):

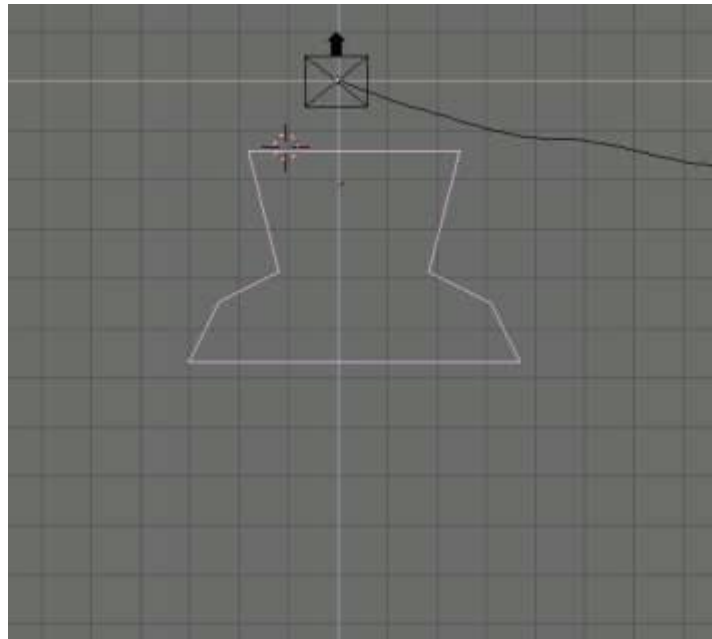


Рис. 2 - Создание рельсов

Уменьшите форму, и скопируйте ее. Получились сечения двух рельсов. Объедините их вместе, выбрав их и нажав [Ctrl]+[J].

На кнопке справа, заголовка окна, вы можете прочесть название объекта. Это вероятно что-то вроде OB:CURVE.001 (по крайней мере у меня так). Теперь выберите главный путь, перейдите в окно EditButtons и введите название объекта в поле BevObj.

Если все нормально, то вы увидите два классных рельса. Возможно, что рельсы повернуты на 90 градусов относительно объектов, которые вы только что нарисовали (рис. 3):

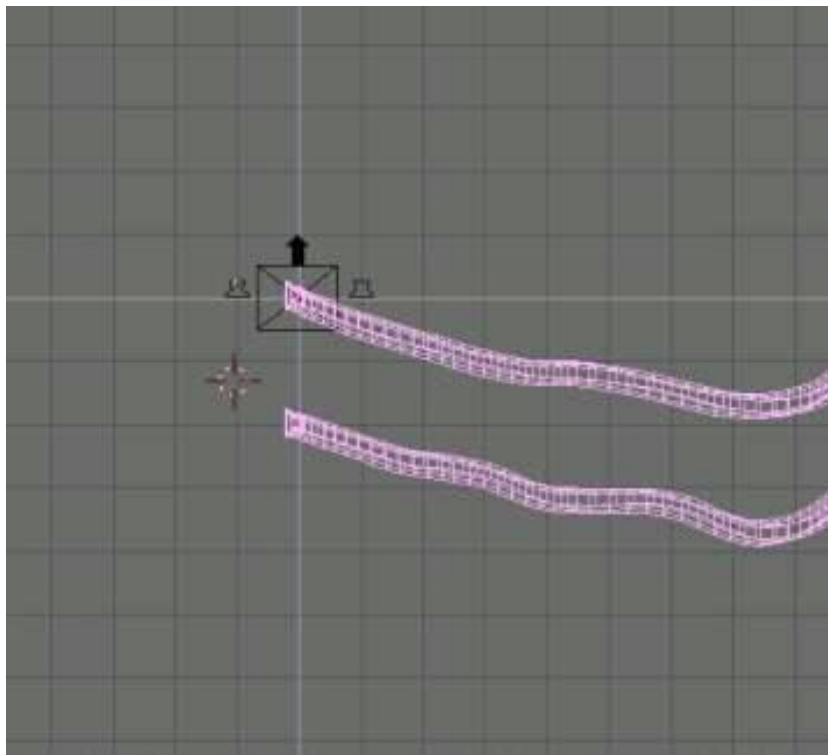
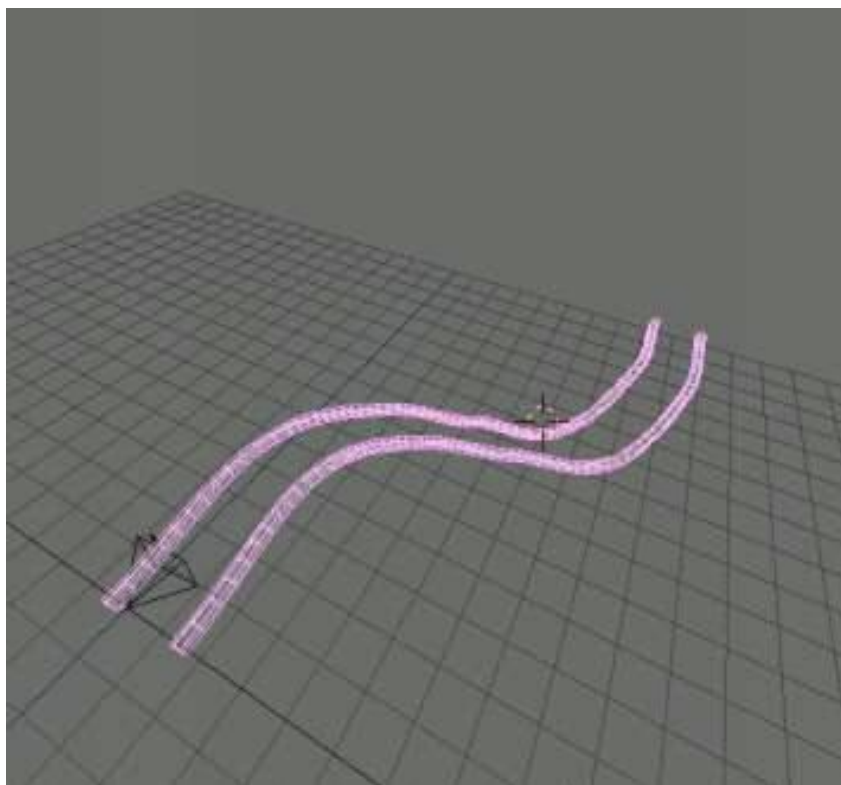


Рис. 3 - Типичная ошибка на этапе создания

Чтобы исправить это, выберите форму рельсов, войдите в режим редактирования, выделите все вершины (клавиша [A]) и поверните их на 90 градусов (Клавиша [R], затем, при движении мышью, удерживайте клавишу [Ctrl]. Ctrl ограничит углы вращения значениями кратными 5 градусам).

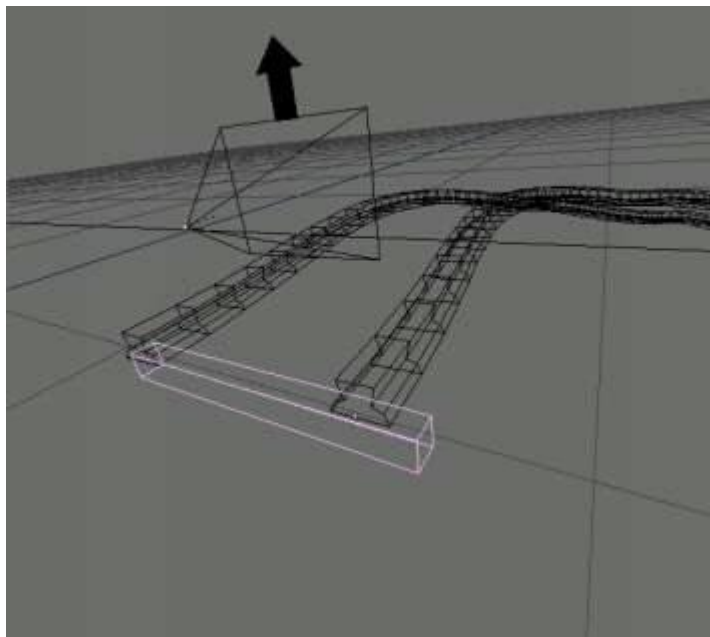


На данный момент у нас должно получиться следующее (рис. 4):

Рис. 4 - Верно построенные рельсы

Процедурные объекты

Теперь начинается забавная часть. Мы будем использовать процедурные объекты, чтобы создать деревянные предметы, которые можно обнаружить в шахтах. Начните с создания блока, немного шире чем два



рельса (рис. 5):

Рис. 5 - Создание блока на рельсах

Привяжите блок к рельсам. (Выберите блок, затем удерживая [Shift], выберите рельсы. Нажмите [Ctrl]+[P]). Теперь, когда Вы проанимируете сцену, вы заметите, что блок тоже следует вдоль рельсов. Выберите блок, и перейдите в окно AnimButtons. Нажмите кнопку [DupliFrames]. Эта опция создаст "виртуальную копию" блока для каждого кадра анимации. Только их получилось слишком много. Увеличивайте значение "DupOff", пока рельсы вам не понравятся. (использовалось значение 2, рис. 6):

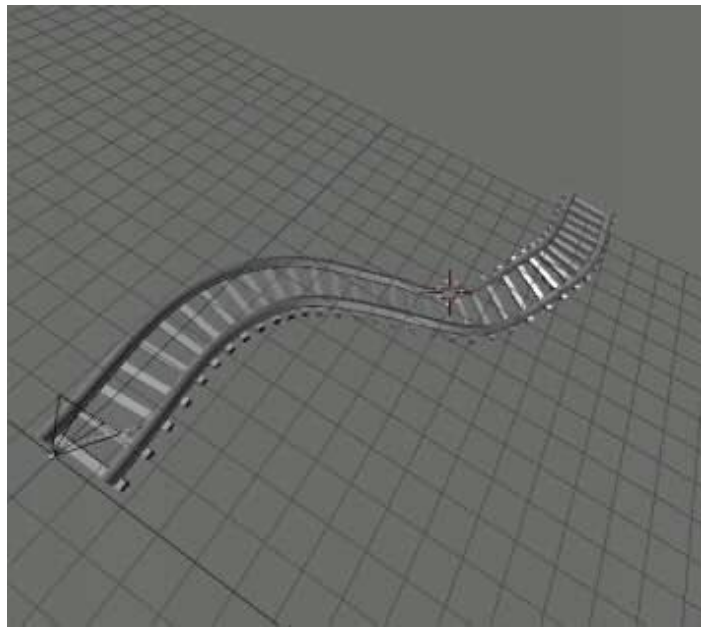
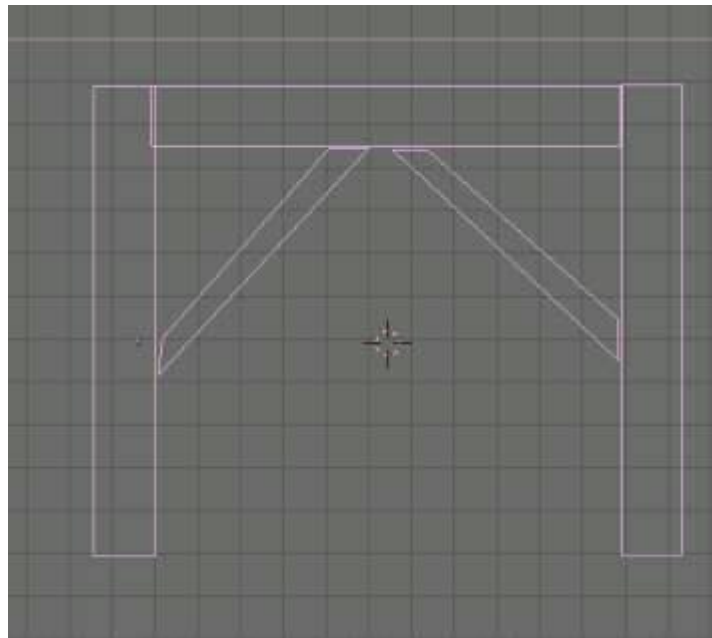


Рис. 6 - Копии блока на всей длине рельс



Теперь, создайте объект, подобный этому (рис. 7):

Рис. 7 - Созданный объект

и придайте ей глубину (установите параметр выдавливания "Ext1:" , если вы использовали кривую, или команду выдавливание "Extrude" если вы используете меш). Снова разместите объект в начале рельсов, привяжите его к рельсам, и нажмите [DupliFrames]. Поиграйте со значением параметра DupOff, пока вам не понравится (рис.8):

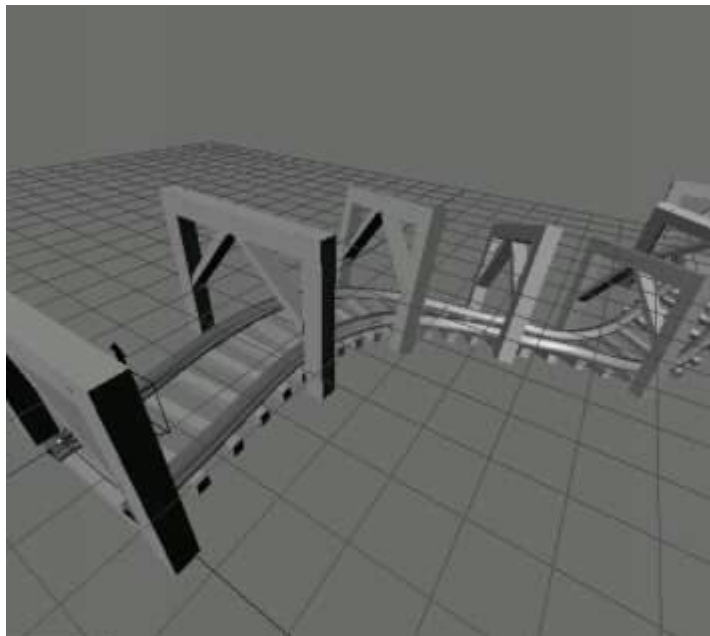


Рис. 8 - Дублирование созданного объекта на всей длине рельс

Для освещения, создайте прожектор (Spot), разместите его над камерой и привяжите его к ней. (Удостоверитесь, что свет направлен вперёд). Он смоделирует свет, устанавливаемый на вашей шахтерской тележке. Однако, так как направление лампы будет точно следовать направлению обзора, освещение будет выглядеть слишком искусственным.

Создайте объект "Empty" и разместите его немного впереди камеры. Привяжите его к рельсам. Выберите вначале лампу и затем "Empty". Нажмите [Ctrl]+[T] (Make Track) чтобы заставить свет двигаться вслед за "Empty". Когда вы это сделаете, направление прожектора вероятнее всего изменится. Исправьте это, вращая прожектор, пока он не будет направлен на "Empty". Выберите "Empty", перейдите в окно AnimButtons, и установите значение TimeOffset (временное смещение) равным небольшой отрицательной величине (например, -3). Это заставит пустой объект передвигаться "перед светом" (во времени). Направление света теперь не совпадает с направлением камеры, и результат от освещения выглядит значительно лучше (рис. 9):

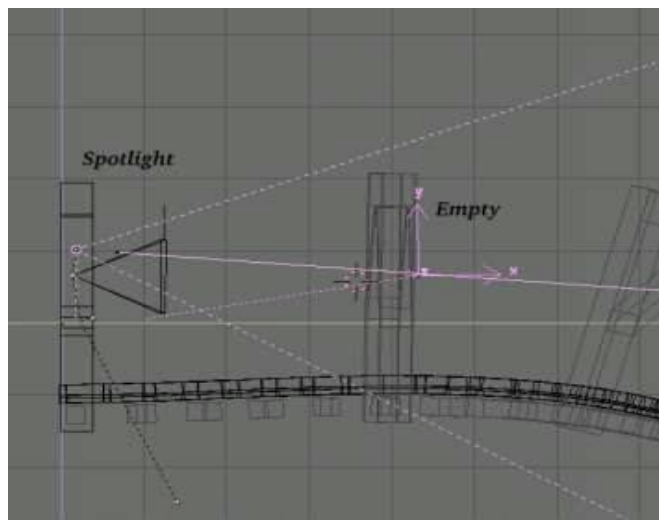


Рис. 9 - Расстановка объектов освещения

Напоследок, создайте каменную стену, производя другое выдавливание (extrusion) вдоль пути. Каждый путь, однако, может иметь только один объект в качестве сечения, так что вы сначала должны продублировать путь рельсов (используя [Shift]+[D], а НЕ [Alt]+[D]!), создайте каменную



стену снаружи и выдавите ее (extrude, [E]). Результат будет таков (рис. 10):

Рис. 10 - Добавление в сечение стены

И наконец результат, после добавления некоторых текстур на ваш выбор и освещения (рис. 11):

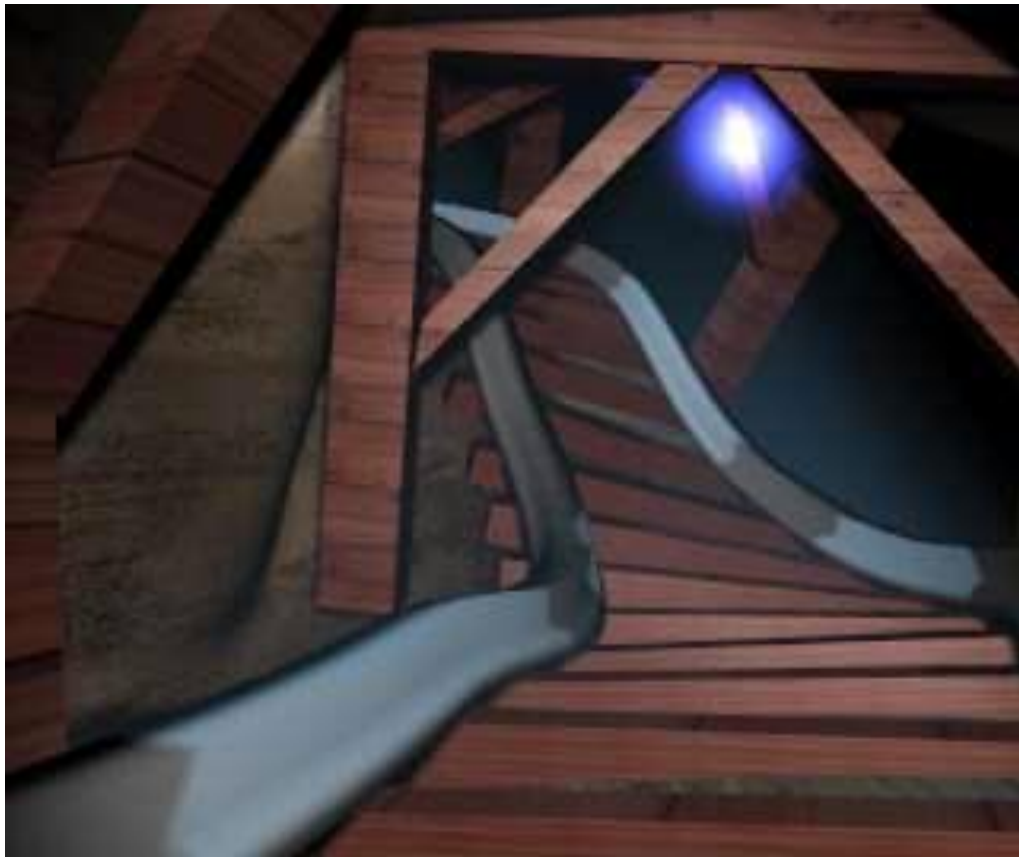


Рис. 11 - Полученный результат рендера

КОНТРОЛЬНЫЕ ВОПРОСЫ:

- 1) Как перейти в режим редактирования кривой Безье?
- 2) Как прикрепляются камеры?
- 3) Как объединить в одно целое несколько объектов?
- 4) Как привязать один объект к другому?
- 5) Как придать созданному объекту глубину?
- 6) Что такое объект Spot и зачем он нужен?
- 7) Как заставить свет двигаться перед объектом?
- 8) Какова функция параметра TimeOffset?
- 9) Как заставить направление света совпадать с направлением камеры?
- 10) Сколько объектов в качестве сечения может иметь каждый путь?

Анимация с помощью ключевых кадров, драйверов и модификаторов

1. Анимация с помощью ключевых кадров

В Blender существует несколько способов анимации. Самый очевидный — анимация при помощи ключевых кадров или кейфреймов (Keyframe). В нижнем окне программы по умолчанию открыт таймлайн — временная линия, на которой можно фиксировать различные параметры объектов в виде ключевых кадров.

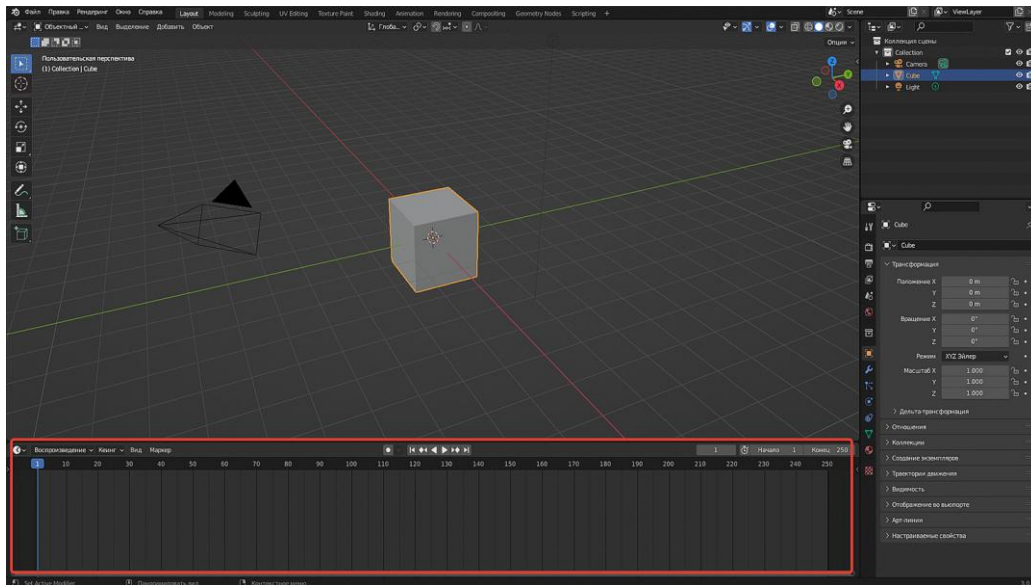


Рис.1 - Расположение таймлайна в Blender 3D.

Попробуем зафиксировать положение куба на кадре 1. Выбираем объект, нажимаем **I** — горячую клавишу **Меню вставки ключевых кадров** (Insert Keyframe Menu) — и выбираем **Положение** (Location). На таймлайне добавится кейфрейм со значением «0, 0, 0».

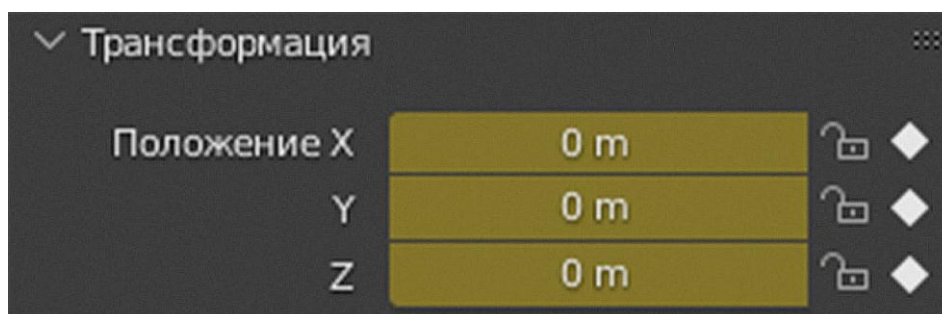


Рис. 2 - Значение положения объекта отобразится на панели «Трансформация» (Transform).

Переходим на 50-й кадр, перемещаем куб куда-нибудь в сторону, ещё раз нажимаем **I** и выбираем **Положение** (Location), фиксируя новые координаты. Запускаем воспроизведение анимации в таймлайне (**Space**) и видим передвижение объекта. Программа самостоятельно высчитывает, как он двигается между двумя заданными ключевыми кадрами.

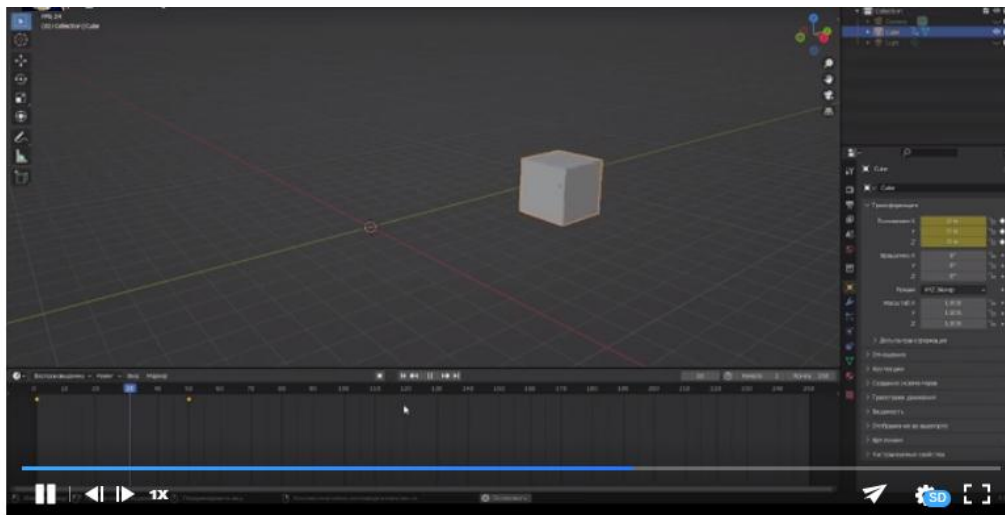


Рис. 3 - Воспроизведение анимации с помощью ключевых кадров

Движение производится плавно, поскольку по умолчанию при добавлении кейфреймов интерполяция между ними представлена в виде кривой Безье. Это визуально отображено в **Редакторе графов (Graph Editor)**.

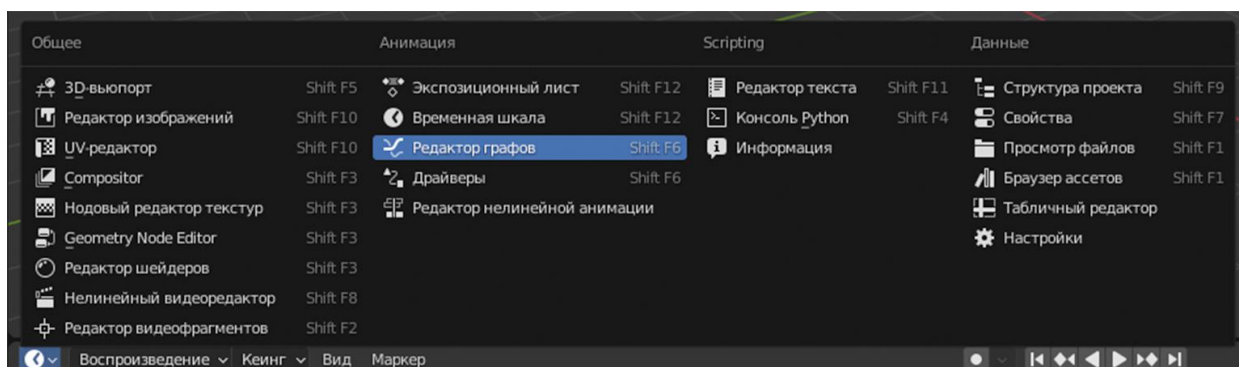


Рис. 4 - Редактор графов во вкладке «Тип редактора»

Редактор графов похож на таймлайн, но в нём отображена более детальная информация о ключевых кадрах и том, как они интерполированы — то есть как кривая отображается на графике.

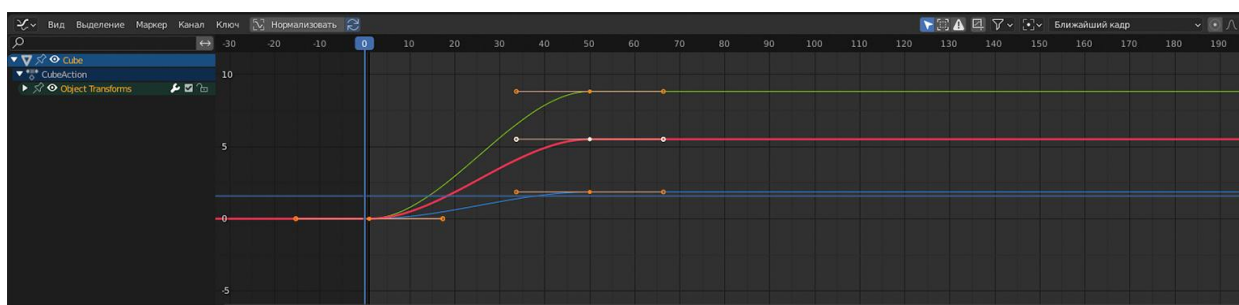


Рис. 5 - Кривая Безье в редакторе графов.

Чтобы масштабировать и перемещать рабочее пространство по своему усмотрению, используйте комбинации Ctrl + СКМ и Shift + СКМ.

В качестве режима интерполяции по умолчанию используется кривая Безье — это означает, что объект медленно начинает движение, разгоняется и постепенно замедляется к концу анимации. Для наглядности

переведём **Центр трансформации (Pivot Point)** в режим **Индивидуальные центры (Individual Centres)** и сделаем кейфреймы чуть шире при помощи клавиши **S**. Это отразится на графике, и распределение скорости движения объекта между кейфреймами изменится. С этими настройками также можно экспериментировать.

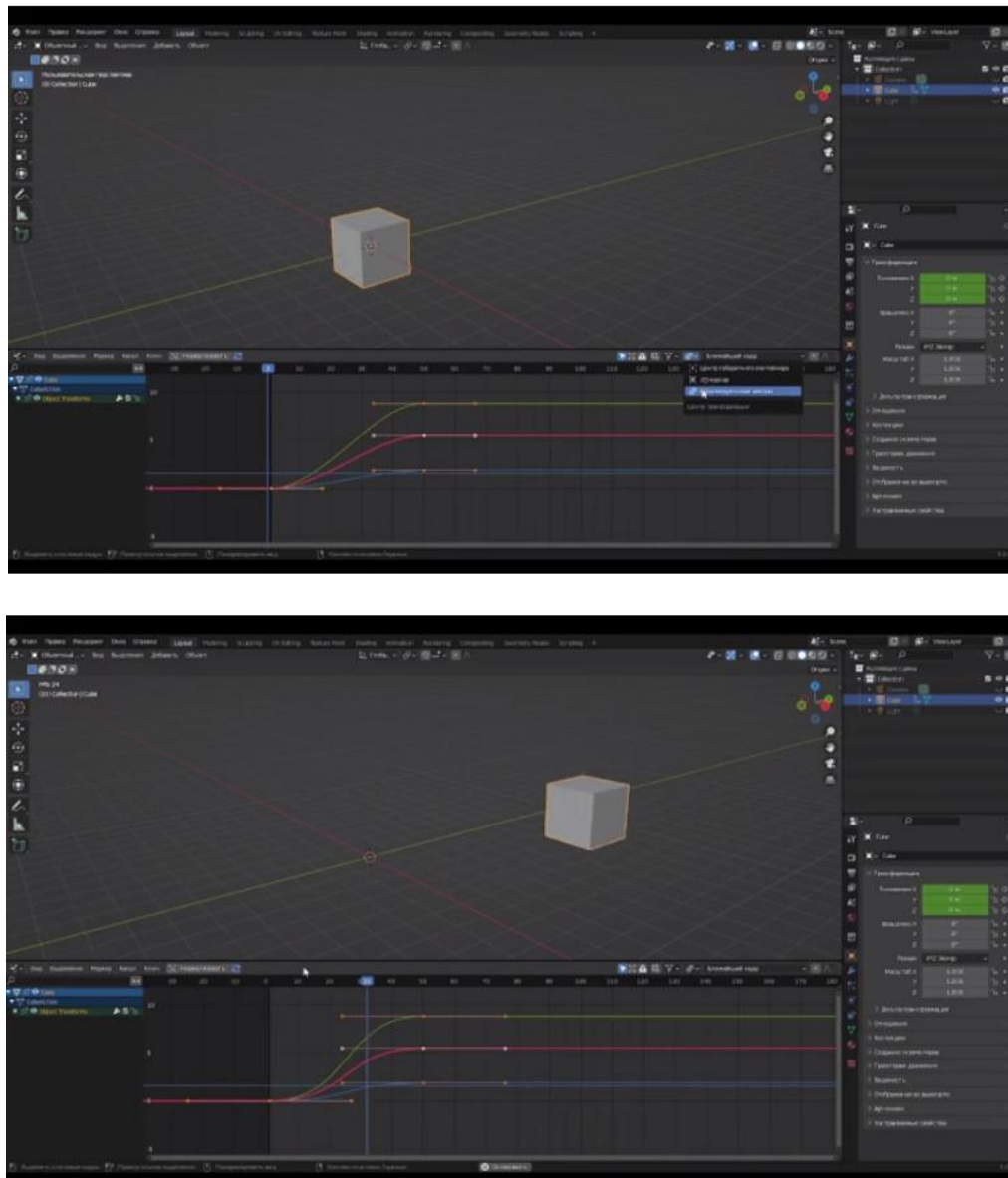


Рис. 6 - Увеличение кейфреймов в редакторе графов

Рассмотрим другие режимы интерполяции. Полный список вызывается клавишей **T**.

Линейно (Linear) — объект движется равномерно, без замедлений и ускорений. Этот режим может оказаться удобным для таких задач, как, например, анимация вращения Земли вокруг своей оси.

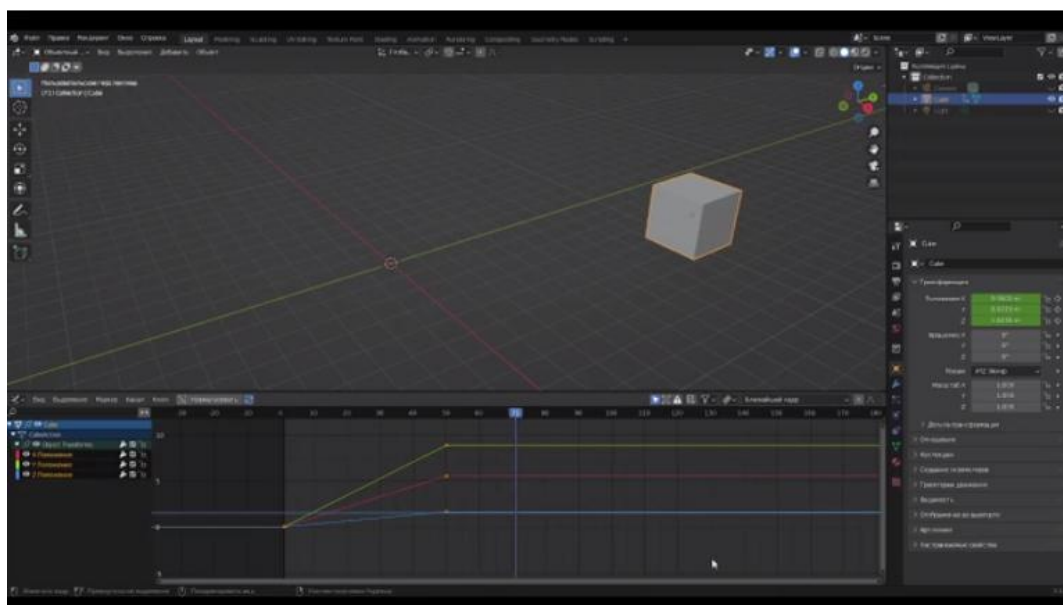
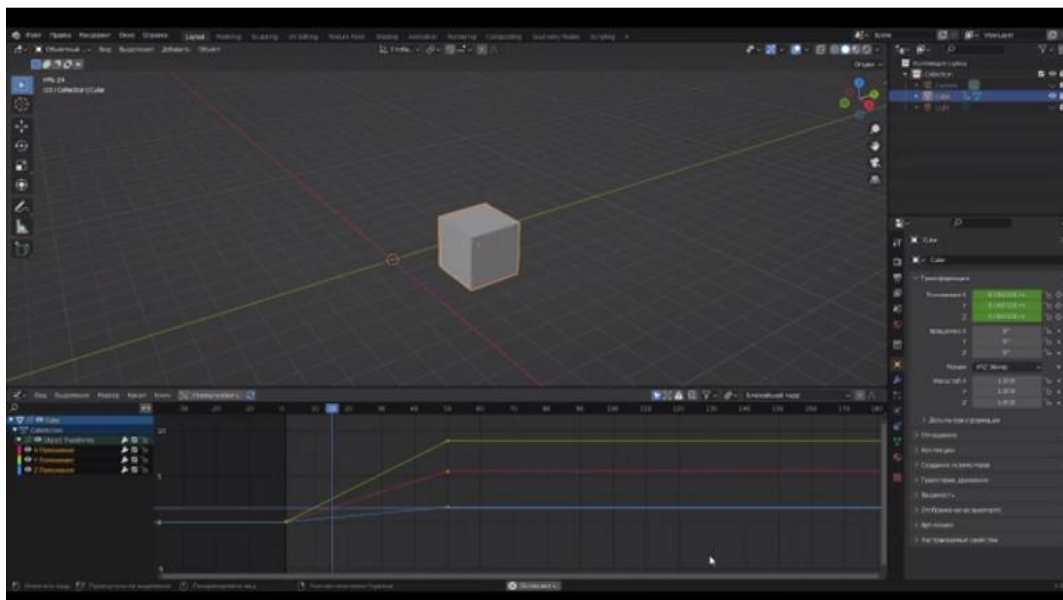


Рис. 7 - Анимация в режиме интерполяции «Линейно»

Постоянно (Constant) — промежуточное значение между кадрами отсутствует. Подойдёт для стоп-моушн-анимации.

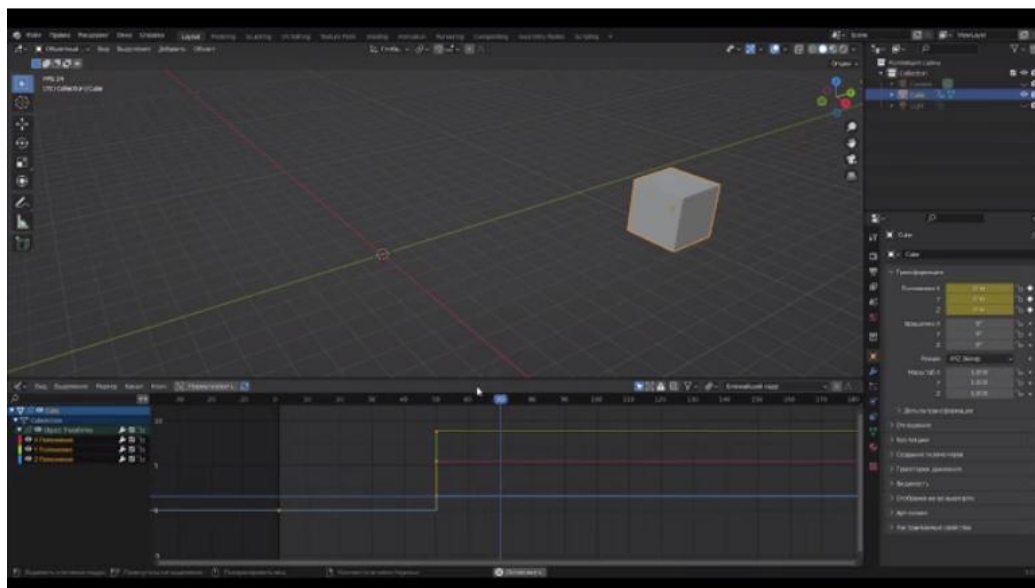
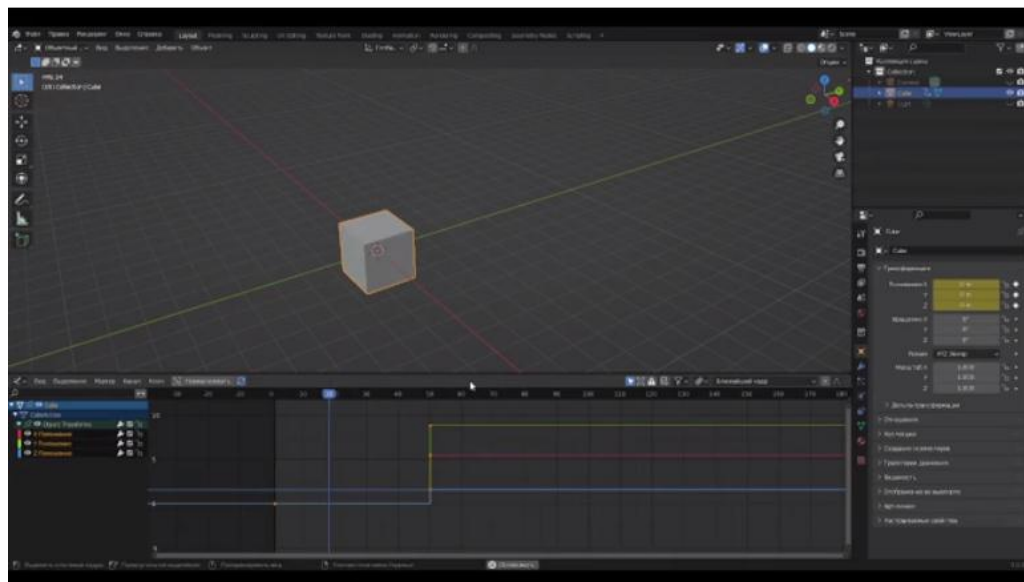


Рис. 8 - Анимация в режиме интерполяции «Постоянно»

Рассмотрим режимы из списка **Динамические эффекты (Dynamic Effects)**.

Отскоки (Bounce) — эффект затухающего подпрыгивания объекта после его падения.

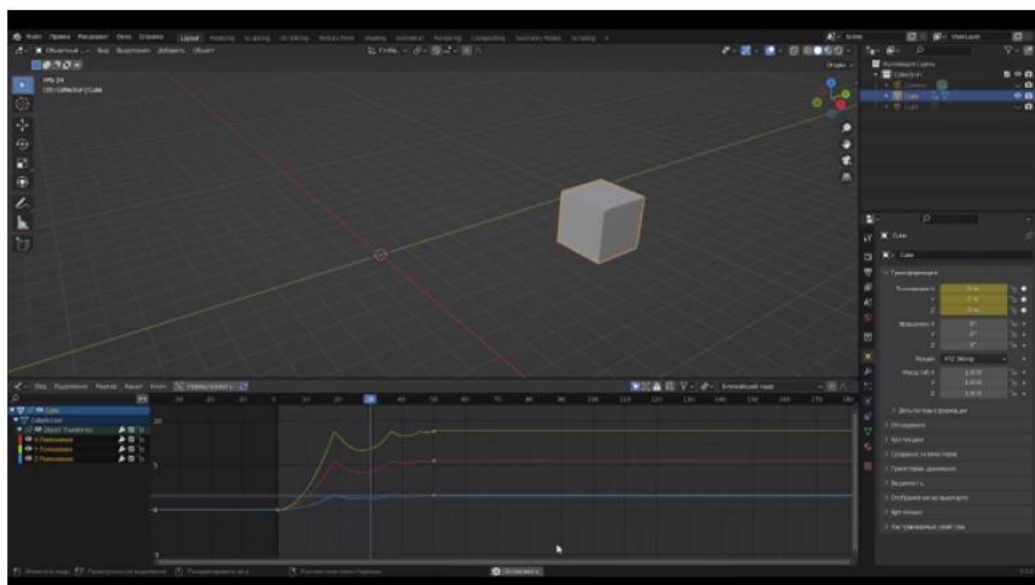
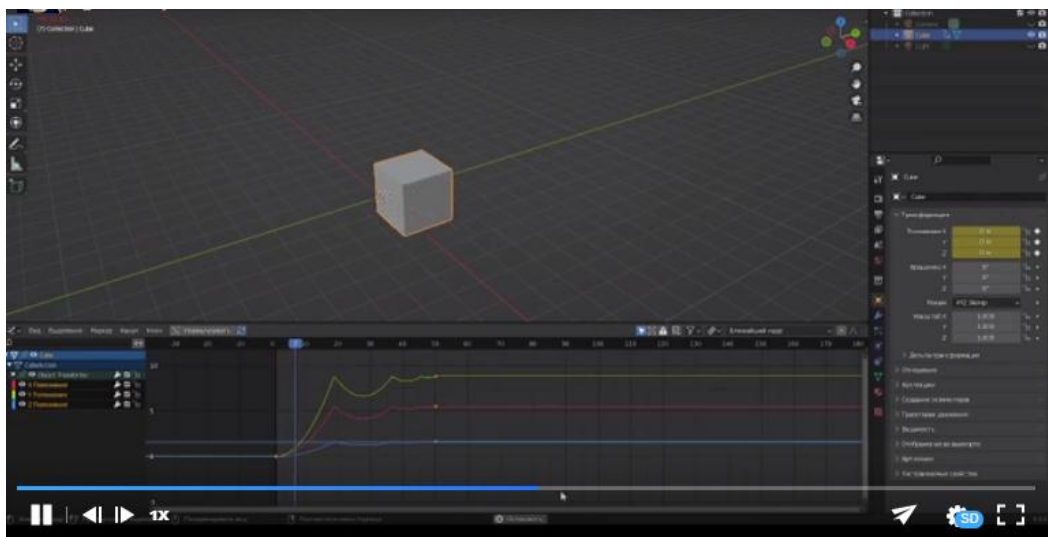


Рис. 9 - Анимация в режиме интерполяции «Отскоки»

Упругое (Elastic) — эффект струны, которую натянули, а затем отпустили.

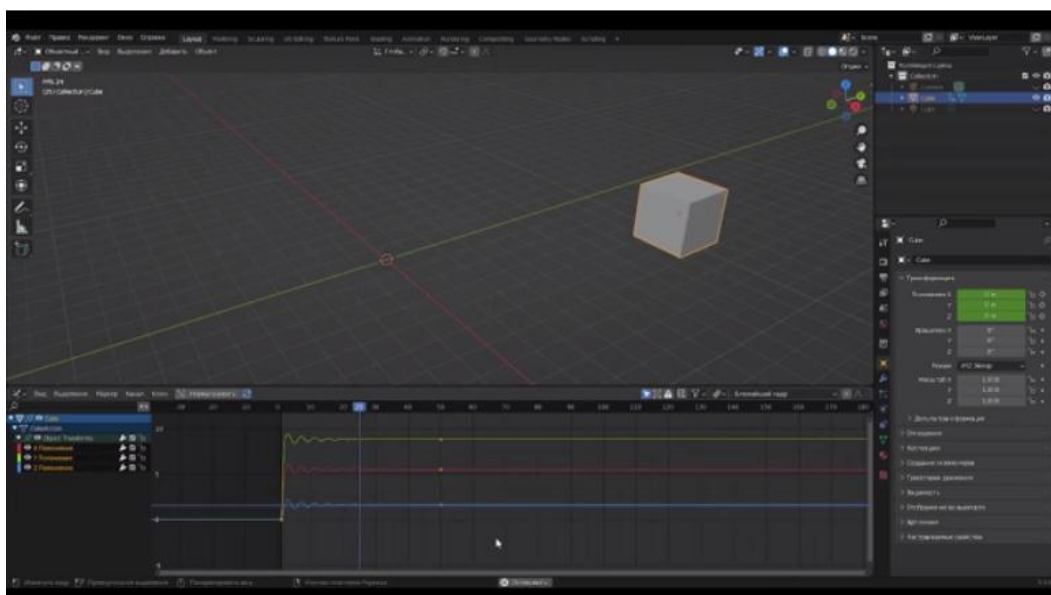


Рис. 10 - Анимация в режиме интерполяции «Упругое»

Сзади (Back) — объект перелетает чуть дальше заданной точки и возвращается к ней.

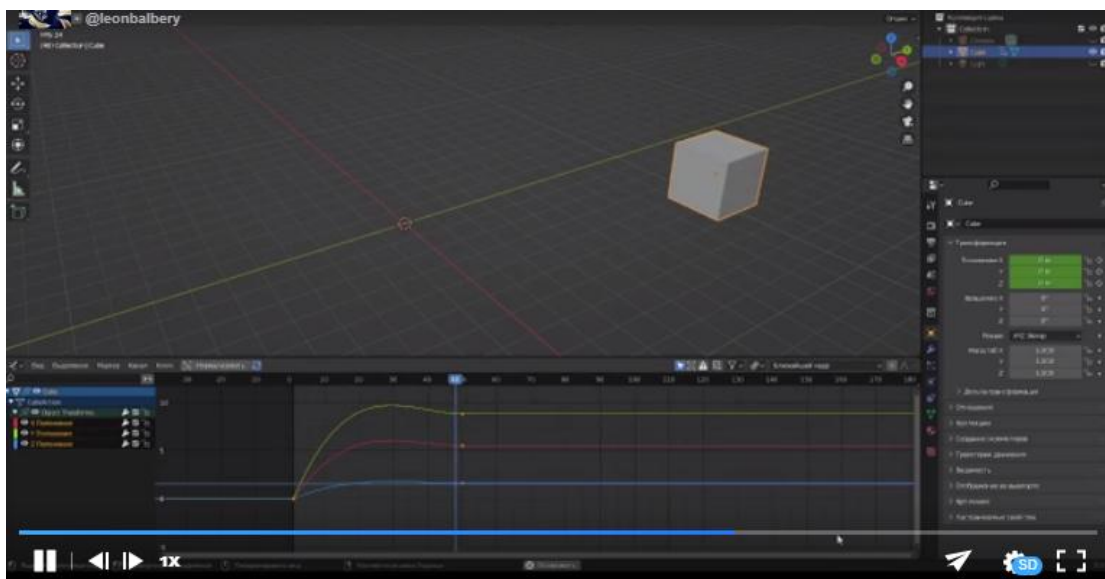


Рис. 11 - Анимация в режиме интерполяции «Сзади»

Попробуем анимировать другие параметры, например вращение. Задаём для объекта исходный угол вращения, вызываем меню через горячую клавишу **I**, выбираем **Вращение (Rotation)**. Затем повторяем то же самое для другого кадра.

ПРИМЕЧАНИЕ

Меню вставки ключевых кадров необязательно вызывать во вьюпорте. Можно навести курсор на нужный параметр во вкладке **Трансформация (Transform)**, при необходимости изменить значение и нажать **I**. Параметр станет жёлтым, оповещая, что изменения зафиксировались в ключевом кадре.

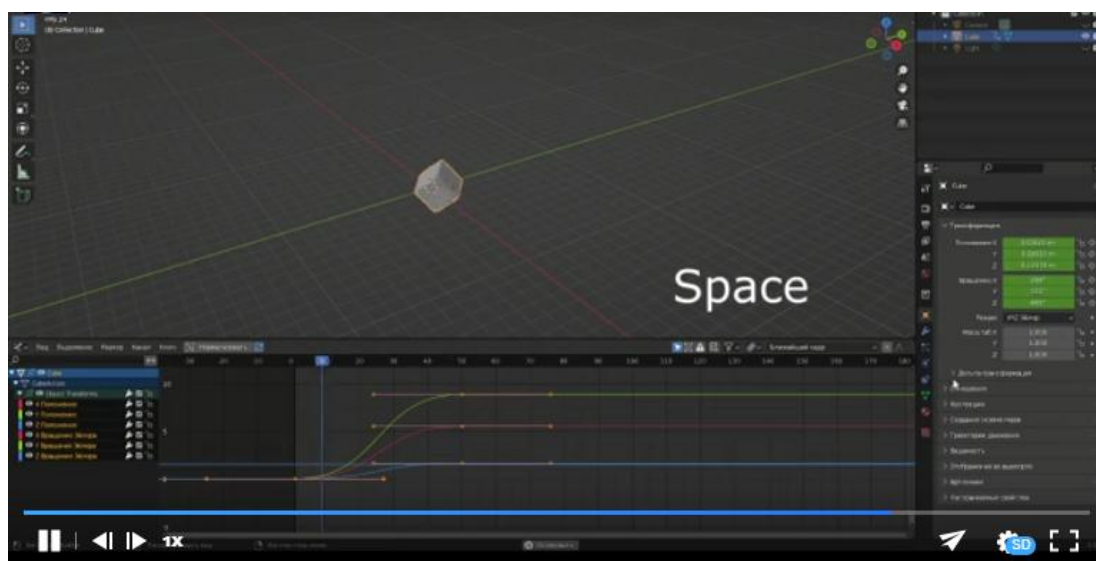


Рис. 12 - Быстрая настройка вращения для ключевого кадра

Анимация с использованием ключевых кадров работает не только со стандартными параметрами объектов (положение, вращение, масштаб), но и с настройками материалов, модификаторов и прочих инструментов программы.

2. Анимация с помощью драйверов

Драйверы в Blender дают возможность задавать зависимости каких-либо параметров одного объекта от определённых параметров другого. Рассмотрим это на практике.

Создаём новый куб и пустышку **Оси** (**Plain Axes**), Перемещаем её в сторону по оси X (**G + X**), выделяем куб и, кликнув правой кнопкой мыши по одному из трёх параметров **Вращения** (**Rotation**) на панели **Трансформация** (**Transform**), выбираем пункт **Добавить драйвер** (**Add Driver**).

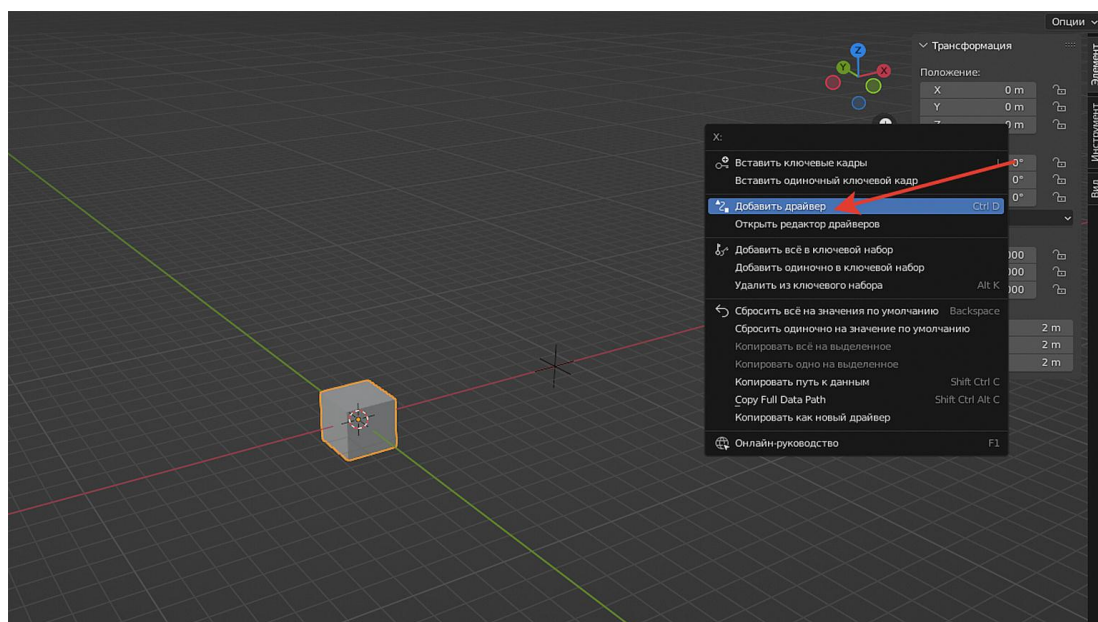


Рис. 13 - Опция «Добавить драйвер» в открывшемся меню

Появится окно с настройками драйвера (всплывающее предупреждение игнорируем). Драйверы поддерживают однострочные выражения на языке Python, но по сравнению с полноценным скриптингом их функциональность сильно ограничена. В ряде случаев использование драйверов может привести к нестабильной работе — например, анимированные с их помощью объекты при рендере останутся статичными или будут двигаться не так, как при воспроизведении анимации во вьюпорте. Тем не менее драйверы удобны для совершения простых операций.

ПРИМЕЧАНИЕ

Функционал Python в драйверах ограничен встроенными методами языка Python и содержанием нескольких импортированных в пространство драйверов модулей: **math**, **mathutils**, **bpy**. Для тех, кто разбирается в Python, выражение, которое мы здесь пишем, — это, по сути, лямбда-функция,

со всеми вытекающими последствиями. Полный список доступных классов и методов можно посмотреть в Консоли Python (Python Console) через автозаполнение в словаре `bpy.app.driver_namespace`. С помощью скриптов Python в этот словарь можно добавлять и другие функции.

У драйверов есть параметр **Тип (Type)**, то есть способ, с помощью которого драйвер вычисляет итоговое значение. В этом уроке мы рассмотрим тип **Скриптовое выражение (Scripted Expression)**.

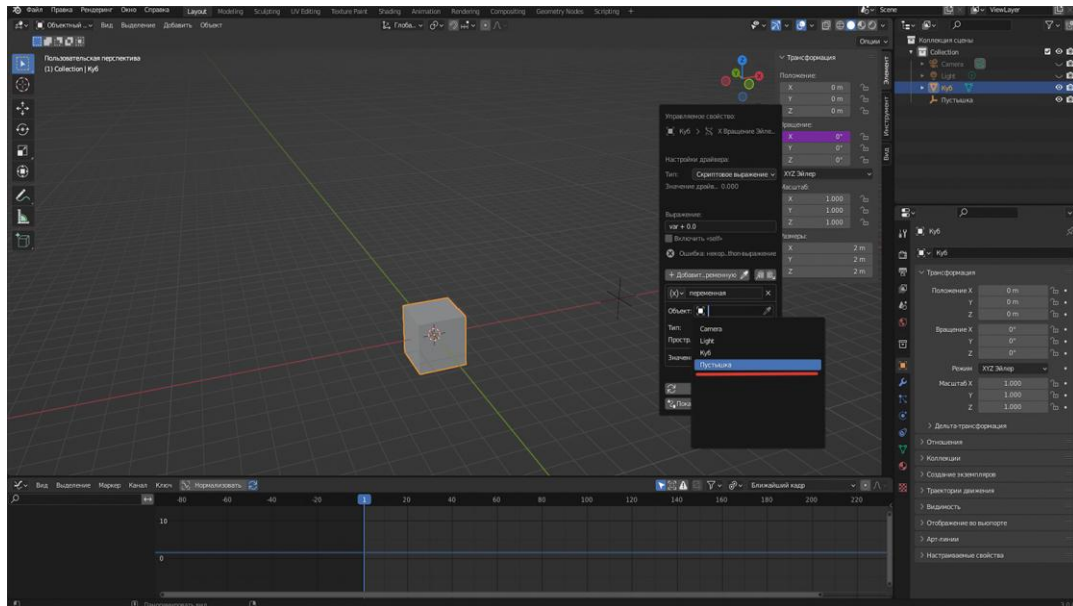


Рис. 14 - Выбор объекта, от параметров которого будет зависеть куб.

Зададим переменную для куба. В качестве **Объекта (Object)** выбираем **Пустышку (Empty)**. Напротив пункта (x) изменяем название с «переменной» (в английской версии *var*) на *empty*. В **Выражении (Expression)** также заменяем *var* на *empty*.

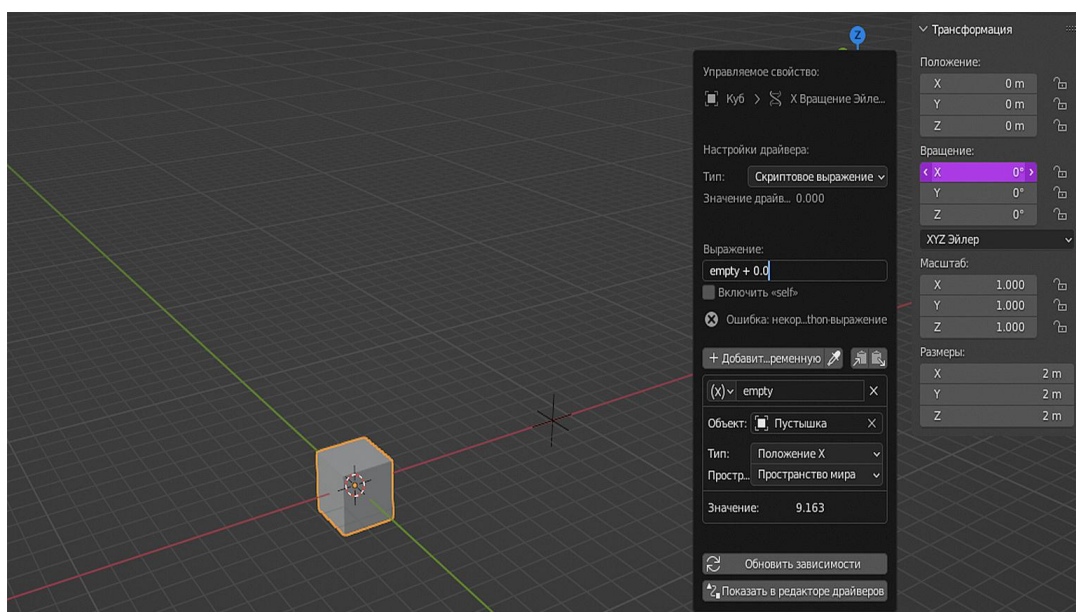


Рис.15 - Замена названия в выражении

Ниже, в пункте **Тип (Type)**, определяем, какой именно параметр пустышки мы будем применять в качестве драйвера для вращения куба. Выберем **Положение X (X Location)** — это означает, что для вращения куба мы используем позицию **Пустышки (Empty)** по оси X. Теперь, если двигать пустышку по этой оси — куб начнёт вращаться.

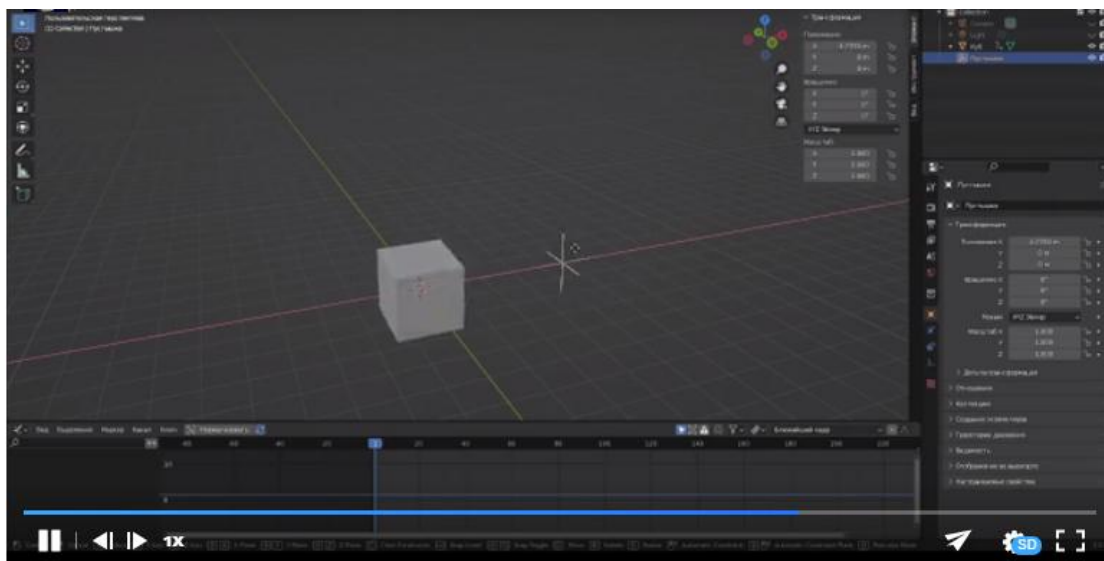


Рис. 16 - Вращение объекта с помощью пустышки

Для анимаций можно писать небольшие формулы. В драйверах они называются выражениями. Возвращаемся в меню, нажав на модифицированный параметр ПКМ и выбрав **Редактировать драйвер (Edit Driver)**. В **Выражении (Expression)** умножаем переменную `empty` на 2 и нажимаем **Enter**.

`empty * 2`

Знак звёздочки (*) означает умножение. Нажимаем **Enter**. Теперь, если перемещать пустышку по оси X — куб вращается в два раза быстрее.

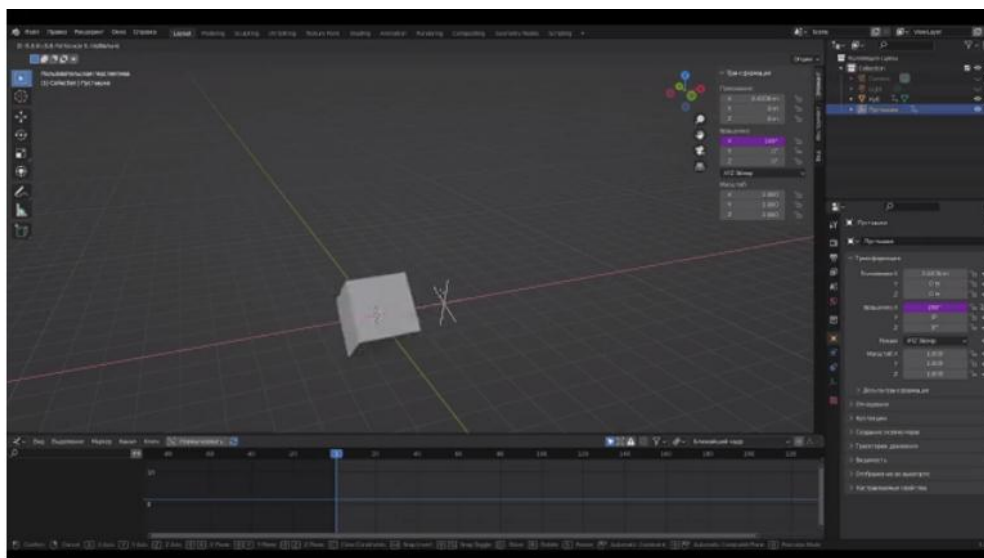


Рис. 17 - Вращение объекта с помощью пустышки при удвоенном значении

Продолжаем эксперимент: дублируем куб (**Shift + D**), выделяем его и чуть сдвигаем в сторону. Заходим в **Редактировать драйвер** (**Edit Driver**) и в строке **Выражение** (**Expression**) пишем:

```
empty * 4
```

Так второй куб будет вращаться уже в четыре раза быстрее, что хорошо заметно в сравнении с первым.

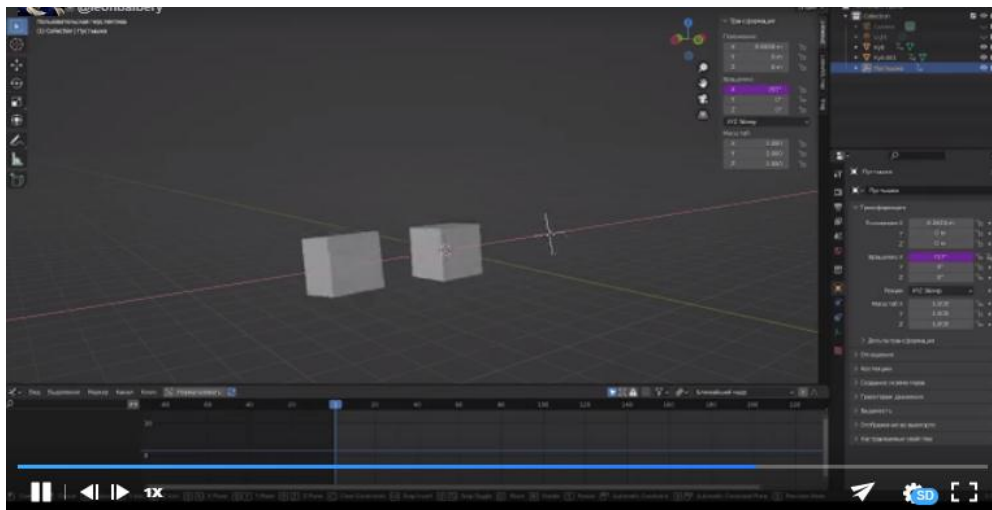


Рис. 18 - Пример вращения двух объектов, имеющих разное скриптовое выражение, с помощью пустышки

Этим методом удобно анимировать механизмы, шестерёнки и различные конструкции, зависимые друг от друга. Во время работы всегда проверяйте, отобразится ли итоговый результат в рендере и при экспорте. **Рекомендуется воздержаться от сложных систем анимаций — использование драйверов не даёт стопроцентной гарантии корректного воспроизведения анимации даже при простых операциях.**

ПРИМЕЧАНИЕ

Важно знать, что в **Выражении** (**Expression**) нельзя импортировать модули **Python**. Например, в теории можно было бы установить случайное число для вращения множества кубов, написав для первого куба выражение `empty * 2 * random.randint (0,100)`. К сожалению, драйвер «не знает» ни о модуле **random**, ни о методе **randint**, поэтому случайные числа можно использовать только непосредственно в **Python**.

Существуют выражения, которые позволяют анимировать объекты без участия других объектов. Удалим данные из первого куба — его переменную и само выражение. Анимация — это изменение параметра во времени, и на каждом кадре у него может быть своё значение. Если написать в **Выражении** (**Expression**) слово **frame** — драйвер на каждом кадре будет

выставлять значение параметра, равное текущему кадру. Так происходит потому, что **frame** в Blender — ключевое слово, обозначающее номер текущего кадра.

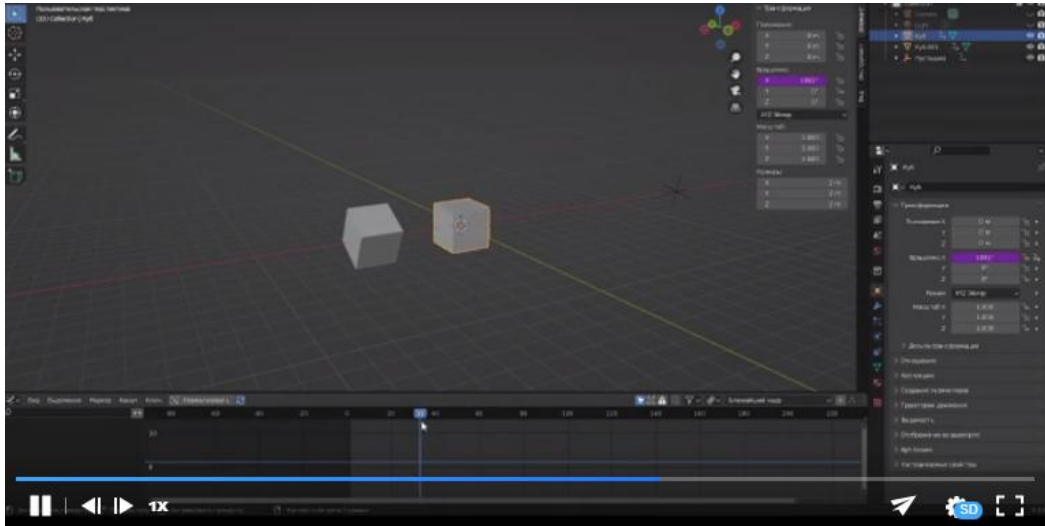


Рис. 19 - Вращение объекта по оси X при использовании frame в качестве значения параметра

ПРИМЕЧАНИЕ

При подключённом драйвере не обязательно переходить в меню **Редактировать драйвер** (Edit driver), чтобы поменять выражение. Достаточно щёлкнуть ЛКМ по параметру, который нужно изменить, прямо во вкладке **Трансформация** (Transform) и набрать выражение там.

Сейчас куб вращается с огромной скоростью — на кадре 9 он поворачивается на 516°. Это объясняется тем, что для параметра вращения счёт идёт в радианах, а не в градусах. Чтобы перевести значение в градусы, напишем выражение.

frame/180*pi

В данном случае **pi** — тоже ключевое слово, которое обозначает математическое число π (3,14159...). Нажимаем **Enter** и видим, что на кадре 9 куб поворачивается на 9°. Если анимацию нужно ускорить — умножаем формулу на нужное число (для наглядности возьмём первую часть выражения в скобки), например 20:

(frame/180*pi) * 20

Теперь куб начнёт вращаться со скоростью 20° в кадр, и на кадре 9 он повернётся на 180°. Чтобы замедлить вращение, уменьшите число.

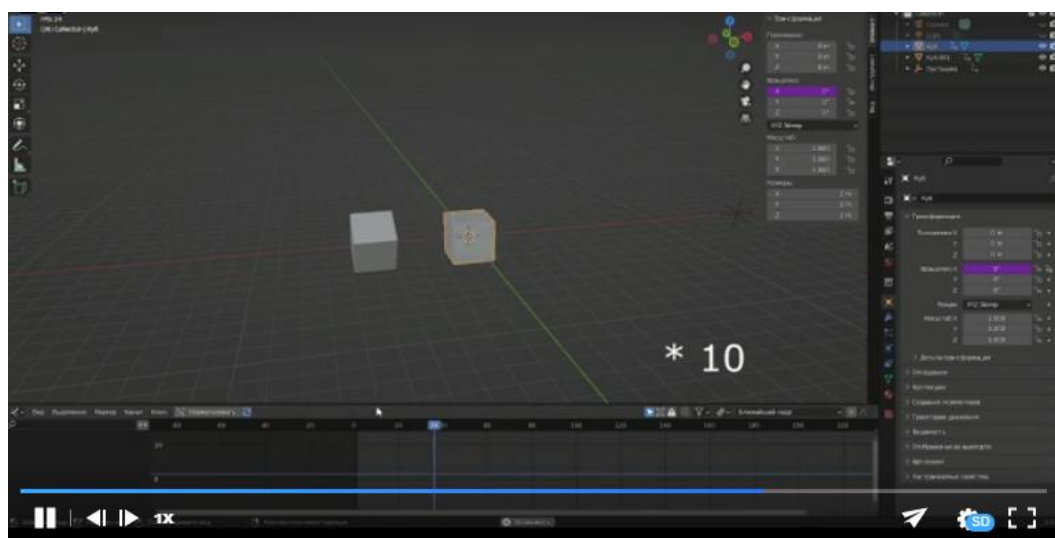
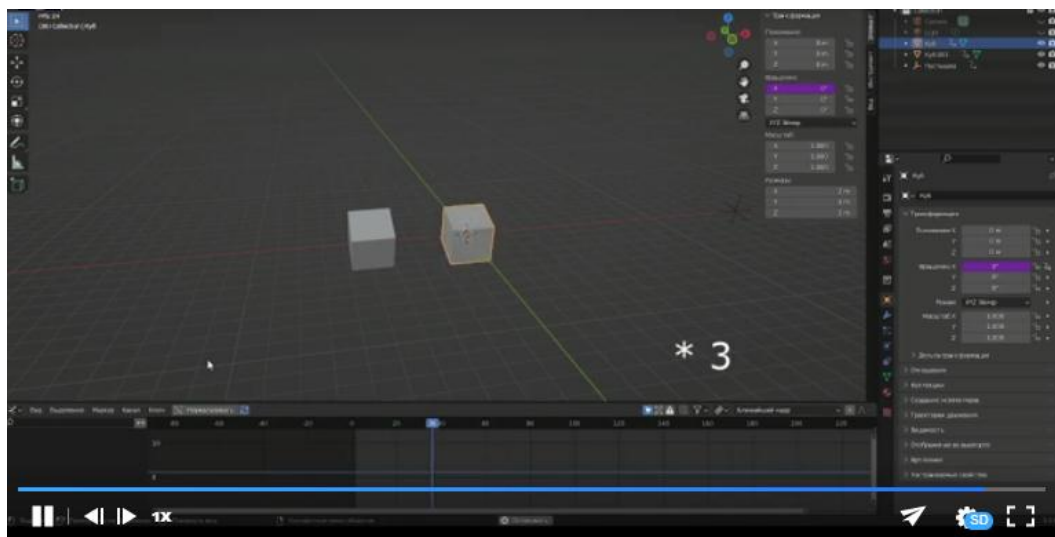


Рис. 20 - Скорость вращения куба при умножении выражения $(\text{frame}/180 \cdot \pi)$ на 20, 10 и 3

Ещё можно написать выражения с использованием синусов ($\sin$), косинусов ($\cos$), тангенсов ($\tan$), арктангенсов (atan) и других функций. Рассмотрим следующий пример:

$\sin(\text{frame}/180 \cdot \pi \cdot 20)$

Если вы помните из уроков математики, на графике синус отображается в виде волны. При таком выражении куб начнёт вращаться из стороны в сторону.

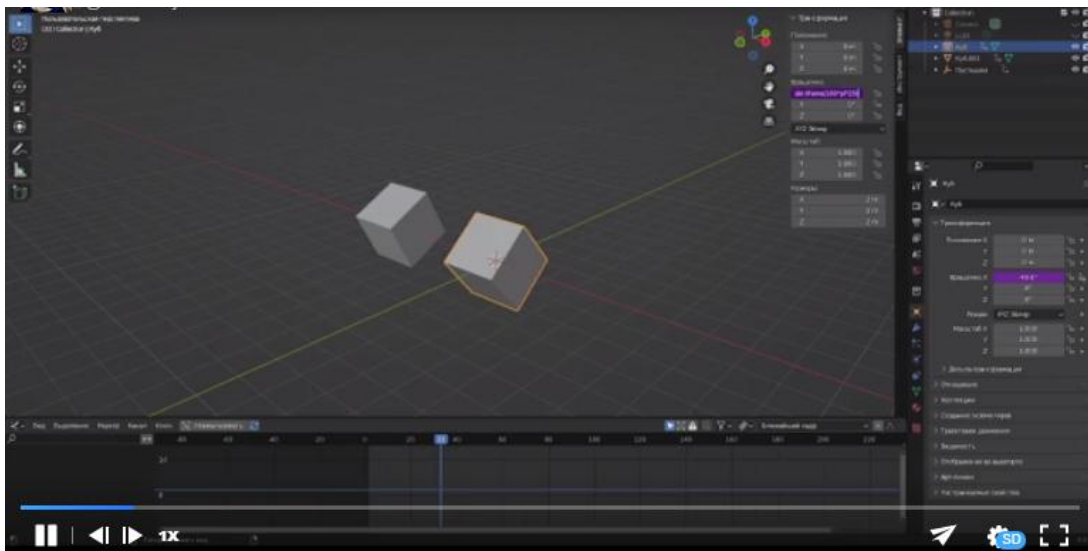


Рис. 21 - Покачивание куба при выражении $\sin (\text{frame}/180*\pi*20)$

Для уменьшения амплитуды можно поделить значение — например опять на 20:

$\sin (\text{frame}/180*\pi*20)/20$

В этом случае куб будет слабо покачиваться.

Такие базовые операции очень удобны для реализации каких-то фоновых событий, например полёта облаков по небу или колыхания травы, — нужно всего лишь «повесить» драйвер на объект, и тот будет двигаться сам по себе. При этом никакие ключевые кадры не нужны.

Чтобы удалить драйвер с объекта, кликните **ПКМ** по модифицированному параметру и нажмите **Удалить драйверы (Delete Drivers)**.

3. Анимация с помощью модификаторов

Ещё один инструмент для анимации — модификаторы в **Редакторе графов (Graph Editor)**. Рассмотрим, как они работают.

Переходим в начало таймлайна, затем на панели **Трансформация (Transform)** кликаем правой кнопкой мыши на **Положении (Location)** объекта по оси X и выбираем **Вставить ключевые кадры (Insert Keyframes)**.

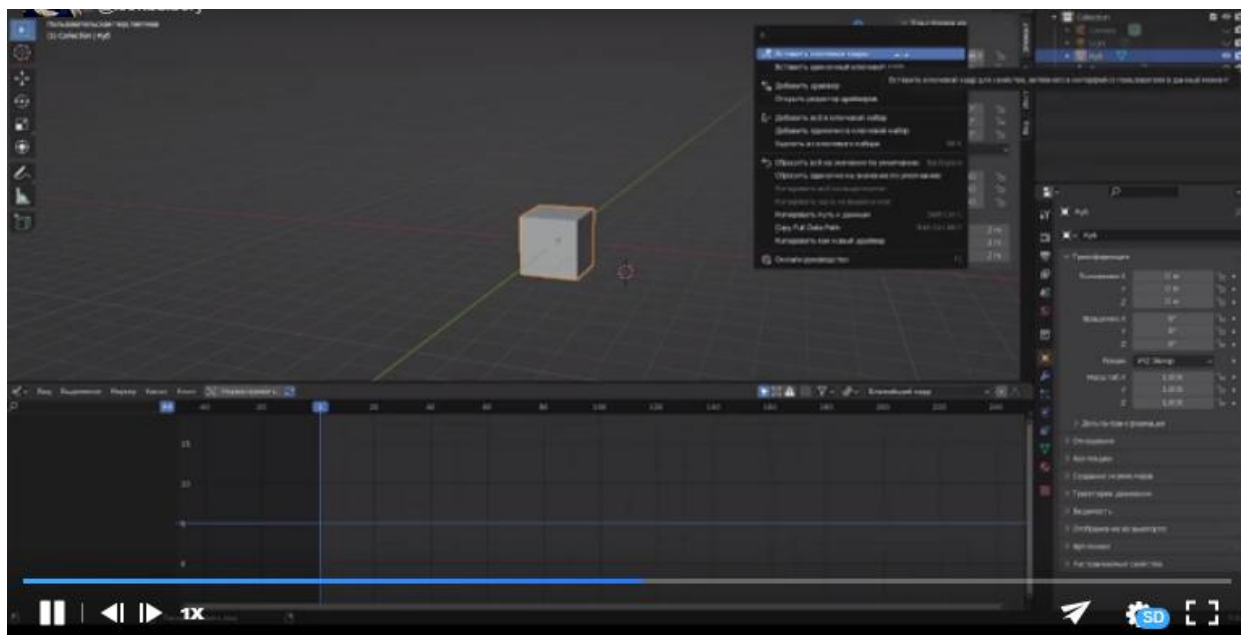


Рис. 22 - Пример вставки ключевого кадра через панель «Трансформация»

Теперь переходим во вкладку **Object Transforms**, которая появилась в левой части **Редактора графов (Graph Editor)**. Выбираем пункт **X Положение (X Location)** и нажимаем N. Справа появится панель со вкладкой **Модификаторы (Modifiers)**.

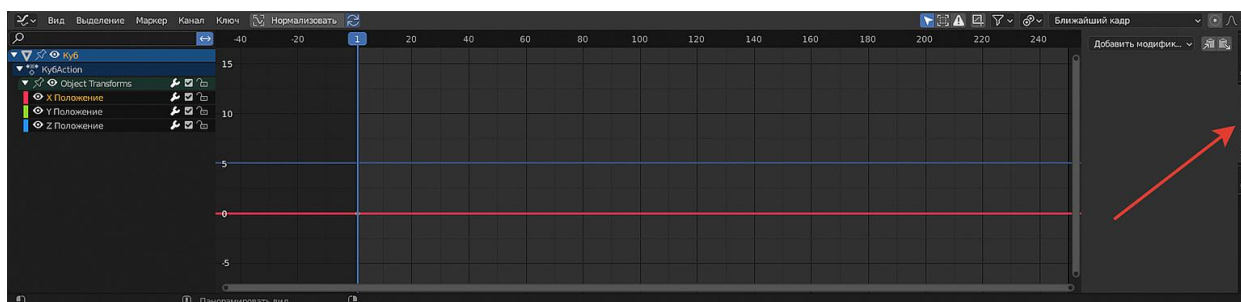


Рис. 23 - Расположение вкладки модификаторов в редакторе графов

В списке **Добавить** модификатор (**Add** **Modifier**) выбираем **Шум** (**Noise**). Он изменяет положение объекта по оси X независимо от ключевых кадров, хотя мы по-прежнему можем изменять это положение на графике кейфреймов. При добавлении модификатора общая траектория графика не меняется, но теперь именно он устанавливает точное положение объекта на основе заданного ключевыми кадрами.

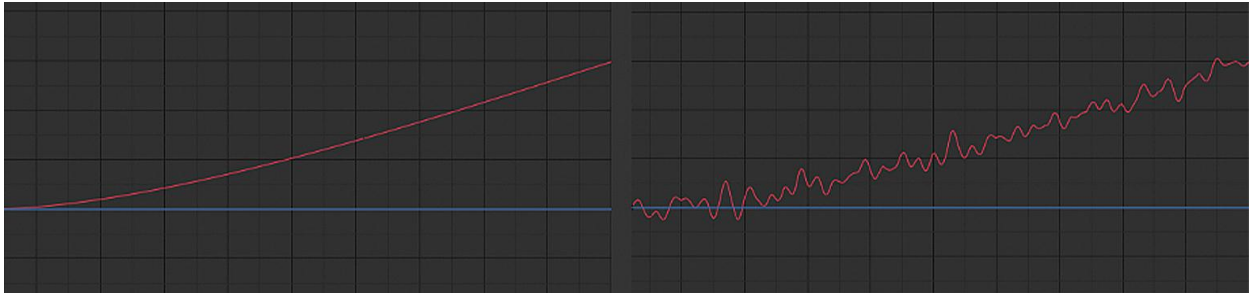


Рис. 24 - График «X Положения» до и после модификатора.

После добавления **Шума** (**Noise**) на панели появится список параметров модификатора. Рассмотрим их.

Масштаб (**Scale**) — отвечает за скорость движения объекта (в данном случае по оси X, потому что выбрана ось X).

Интенсивность (**Strength**) — амплитуда, с которой объект передвигается по оси.

Смещение (**Offset**) — сдвиг огибающей шума назад или вперёд.

Фаза (**Phase**) — создаёт случайный исходный элемент (фазу или противофазу).

Глубина (**Depth**) — добавляет небольшую динамику внутри, как при потряхивании.

Попробуем применить настройки для остальных осей. Копируем модификатор вместе с заданными параметрами, нажав на значок, указанный на скриншоте ниже.

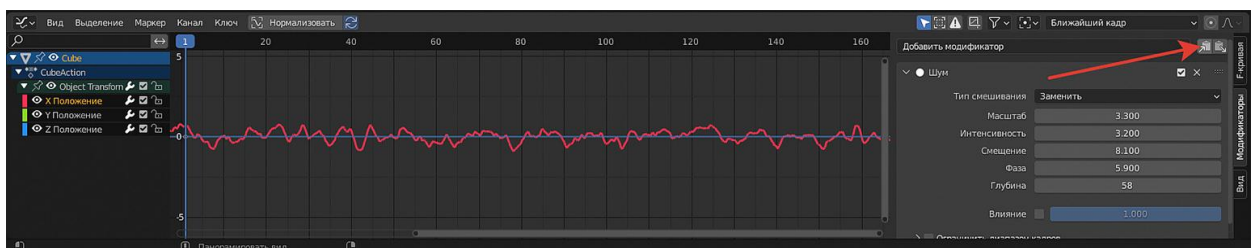


Рис. 25 - Кнопки «Копировать» и «Вставить»

Выбираем другую вкладку, например **Y Положение** (**Y Location**), и вставляем настройки, кликнув на значок, расположенный рядом со значком копирования. Теперь куб движется по двум осям. Для наглядности

мы можем чуть изменить параметр **Смещения** (**Offset**) и увидеть, как объект хаотично передвигается по осям **X** и **Y**.

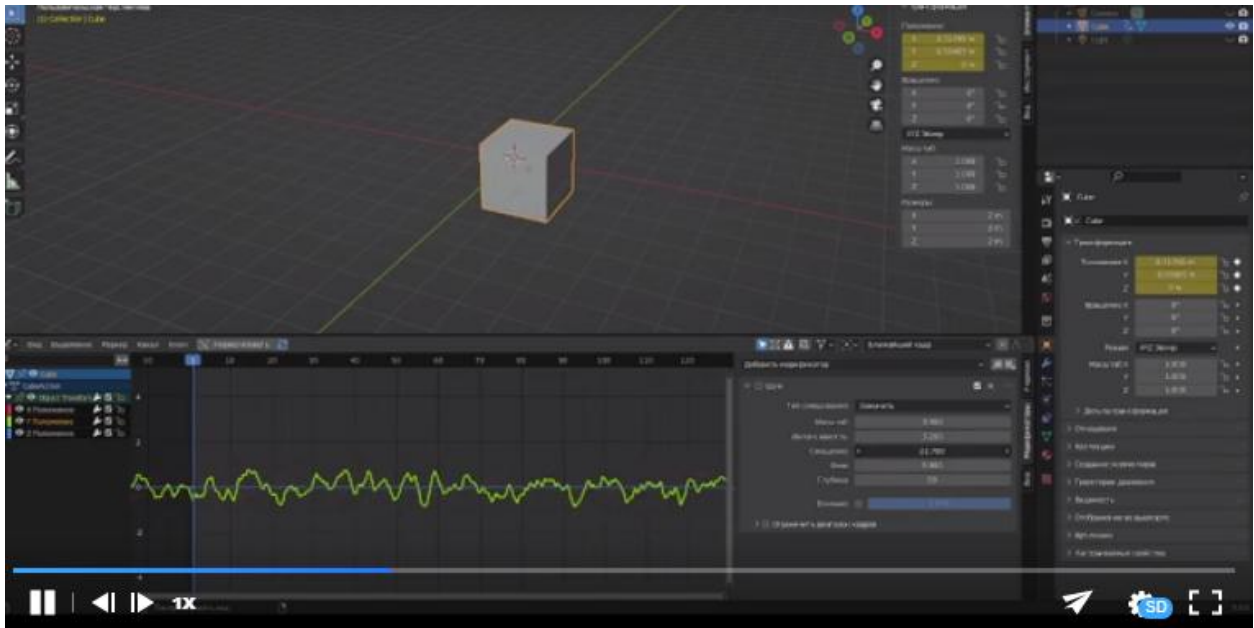


Рис. 26 - Изменение параметра смещения на скопированных настройках и демонстрация хаотичного движения куба по двум осям

Аналогичные манипуляции можно провести и с осью **Z**. Если снова скопировать настройки и изменить параметр **Смещения** (**Offset**), куб начнёт беспорядочно передвигаться уже по трём осям. Модификатор очень удобен, если объектам нужно задать хаотичное движение.

Свойства модификаторов можно ограничивать. Отключим модификатор для осей **Z** и **Y**, нажав на галочку справа.

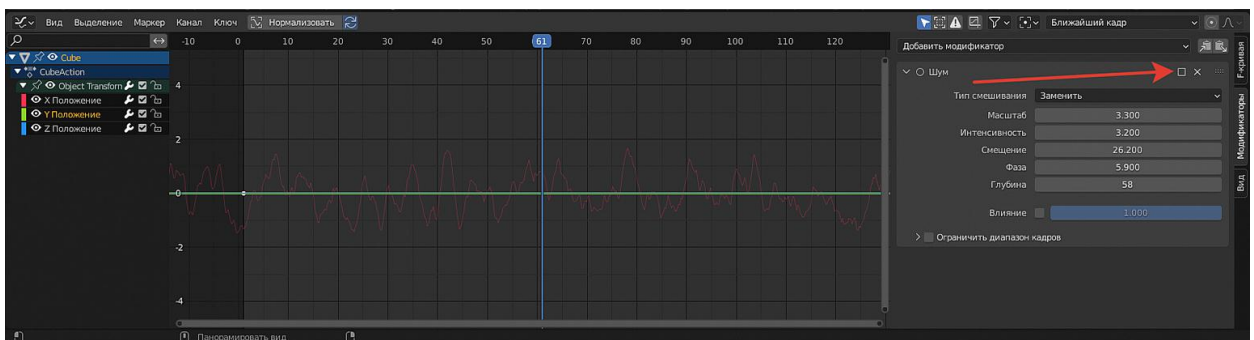


Рис. 27 - Расположение галочки отключения модификатора.

Модификатор для **X Положения** (**X Location**) оставляем включенным. В его настройках выбираем пункт **Ограничить диапазон кадров** (**Restrict Frame Range**) и задаём диапазон — с какого по какой ключевой кадр будет работать модификатор. Для примера возьмём промежуток от 50 до 150. Если нужно плавное затухание в начале и в конце анимации, помогут параметры **Начало смещения** (**In**) и **Выход** (**Out**) — аналоги fade-in/fade-out в звуковых дорожках. Настраиваем их на своё усмотрение и смотрим результат.

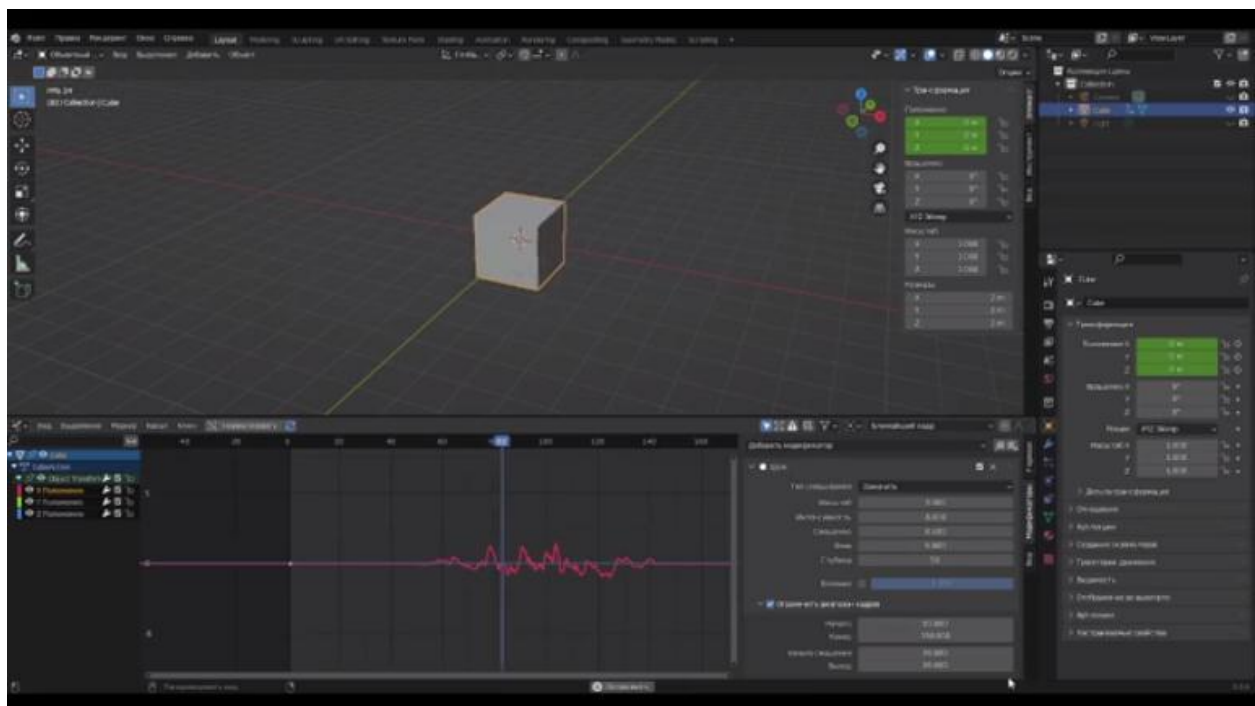


Рис. 28 - Пример использования модификатора «Шум» с ограниченным диапазоном кадров и плавным затуханием

Если одного модификатора недостаточно, можно добавить ещё один **Шум** (**Noise**) и активировать его на другом отрезке таймлайна или наложить на первый — например, чтобы создать «фон».

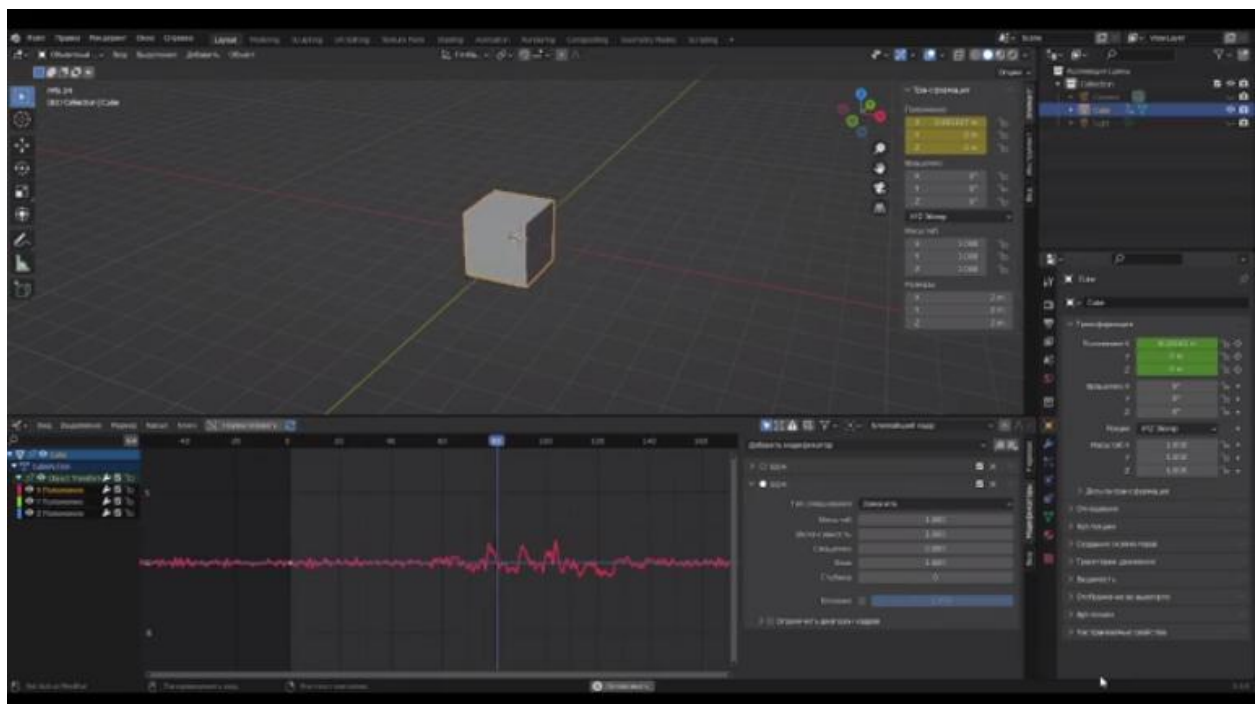


Рис. 29 - Комбинация двух модификаторов «Шум»

Кроме **Шума** (**Noise**) есть и другие модификаторы. Напоследок рассмотрим один из них — **Генератор** (**Generator**). Добавим его для **X Положения** (**X Location**) и включим в настройках **Добавление** (**Additive**) — так его значения будут добавляться к значениям предыдущего модификатора. В противном случае

он их перекроет. Для наглядности в пункте x^1 уменьшим значение до **0,1** — теперь при анимации куб медленно «поедет» по оси X. Так как у нас включён предыдущий модификатор **Шум (Noise)**, с 50-го по 150-й кадр объект будет «потряхивать».

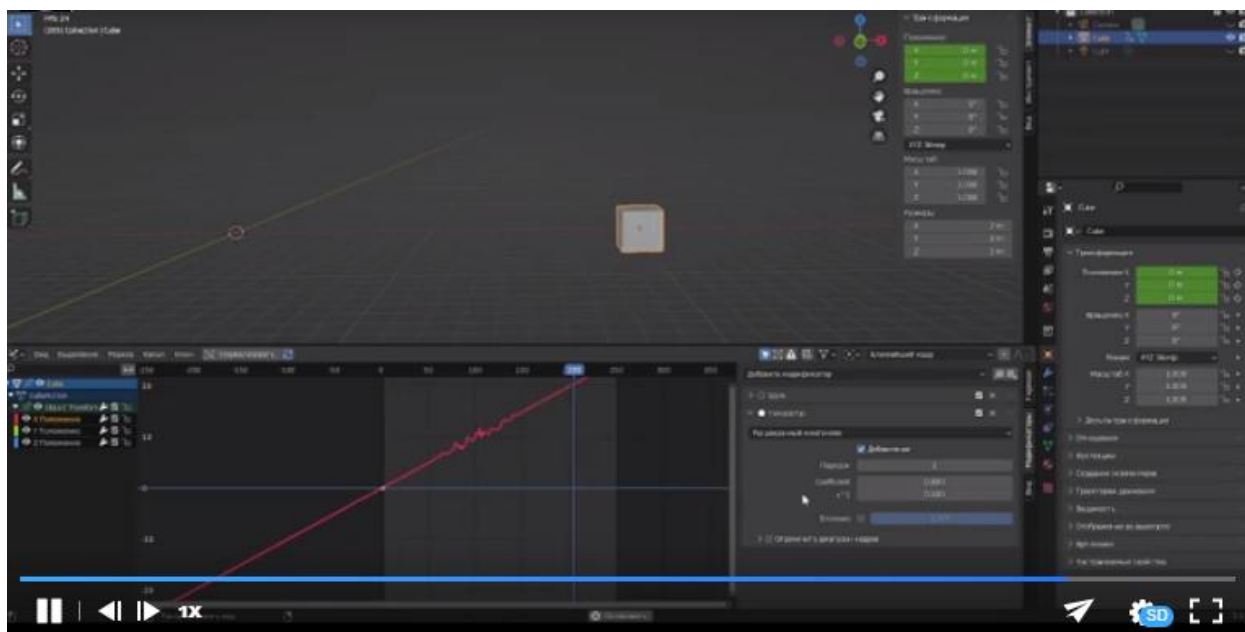


Рис. 30 - Пример комбинации «Шума» и «Генератора» в одном цикле анимации

При этом вручную выставлять ключевые кадры не нужно, как и с драйверами, — вся анимация происходит автоматически. На практике такие колебания, тряски и другие простые движения придают объектам больше жизни и отлично подойдут для анимации в костях арматуры (объекта, схожего с человеческим скелетом), которую мы рассмотрим в следующем уроке.

ПРИМЕЧАНИЕ

Хоть анимация и происходит автоматически, для того, чтобы можно было воспользоваться модификаторами, необходимо установить хотя бы один ключевой кадр на нужном параметре.

Таким образом, мы изучили три основных способа воспроизведения анимации, которые можно использовать для решения любых задач, независимо от сложности проекта. А дальнейшие эксперименты с настройками помогут реализовать все ваши задумки.

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

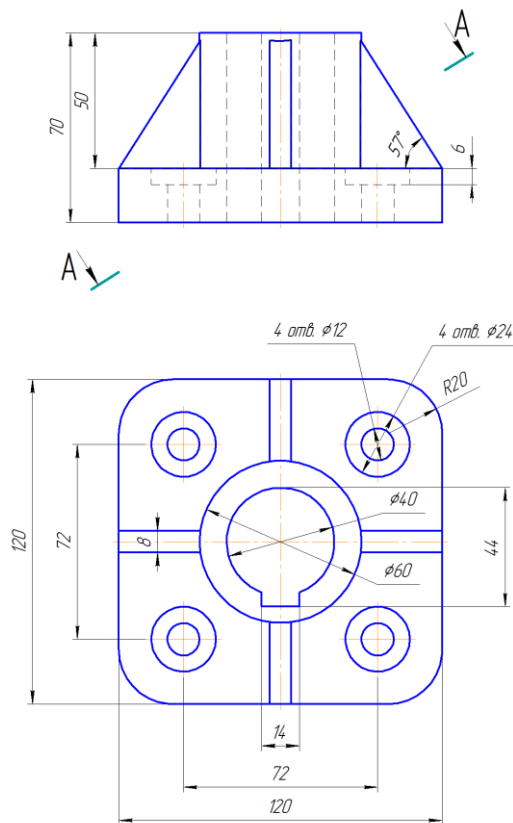
Объект исследования: процесс разработки трехмерной модели в среде Blender.

Предмет исследования: моделирование, проектирование и анимация 3D-модели в Blender по индивидуальному заданию.

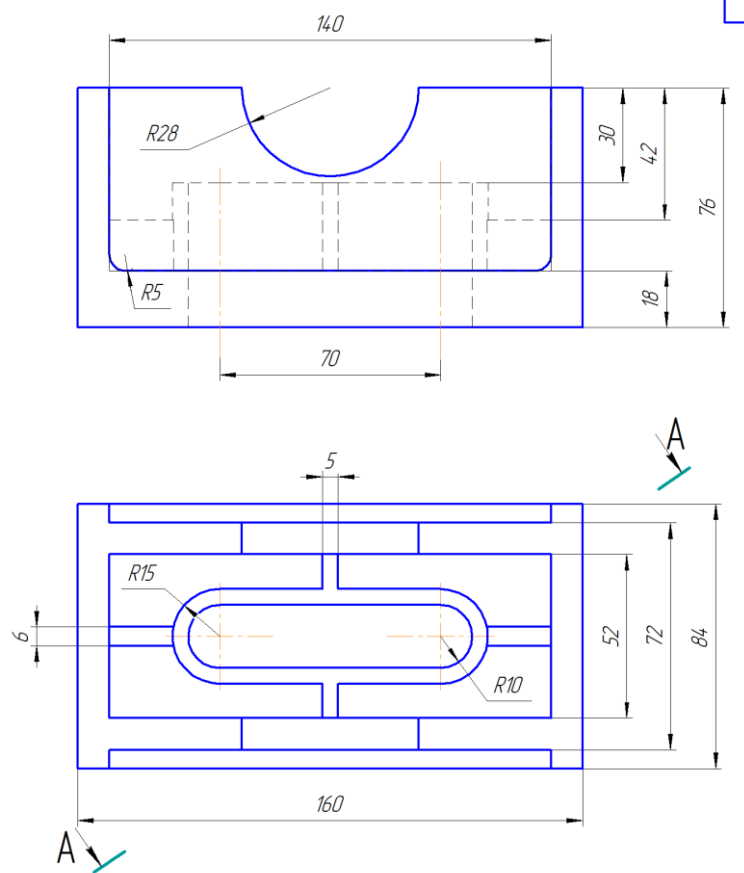
В процессе разработки 3D модели в среде Blender должны быть выполнены следующие задачи:

1. Рассмотреть программы для 3D моделирования.
2. Изучить платформу Blender для разработки 3D модели.
3. Изучение функций выбранной программы.
4. По представленным исходным данным в виде двух проекций детали, построить 3D модель в Blender (**индивидуальный вариант задания по последнему числу номера зачетной книжки**).
5. Создание 3D модели.
6. Анимация 3D модели с помощью ключевых кадров, драйверов и модификаторов.
7. Оформление отчета .

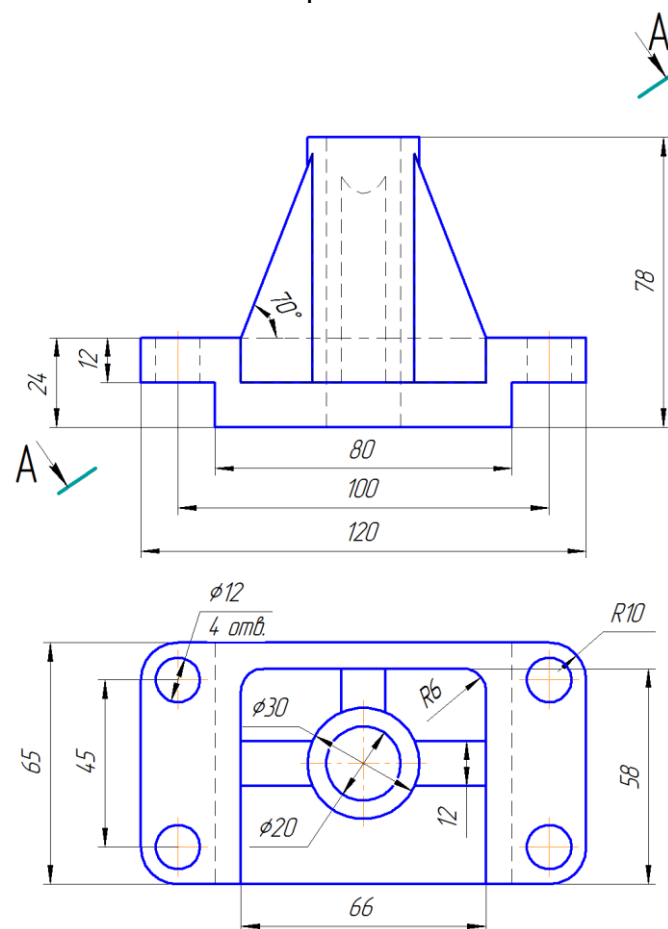
Перечень индивидуальных заданий по вариантам:



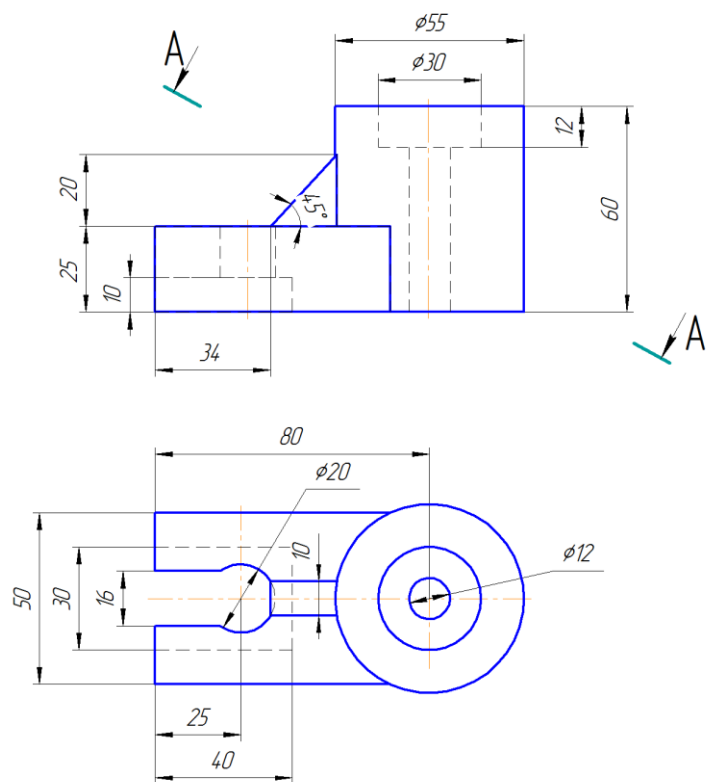
Вариант № 0



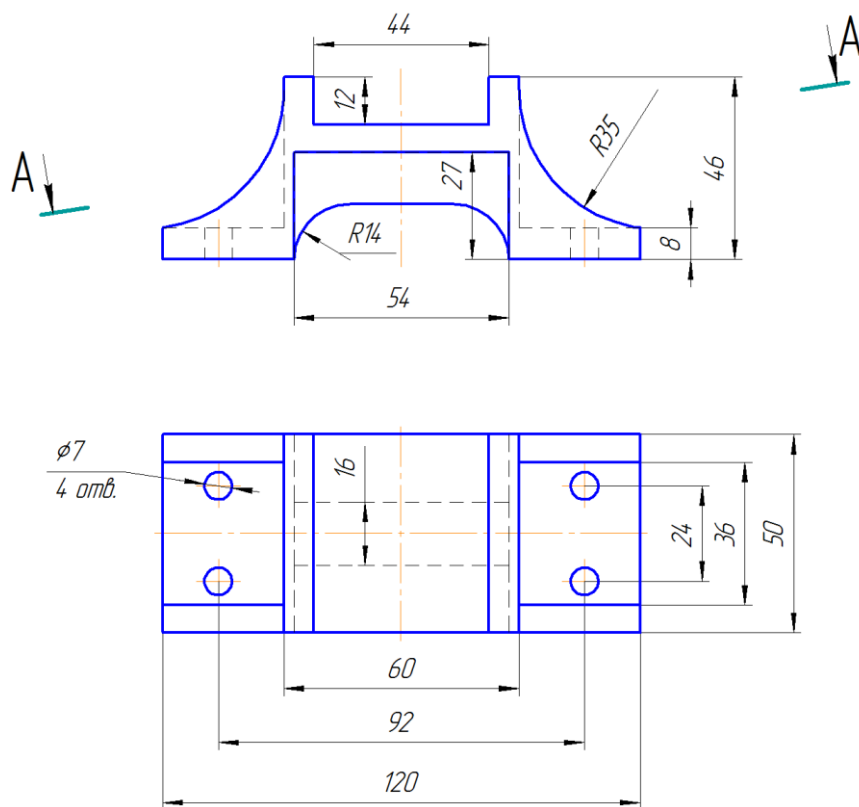
Вариант № 1



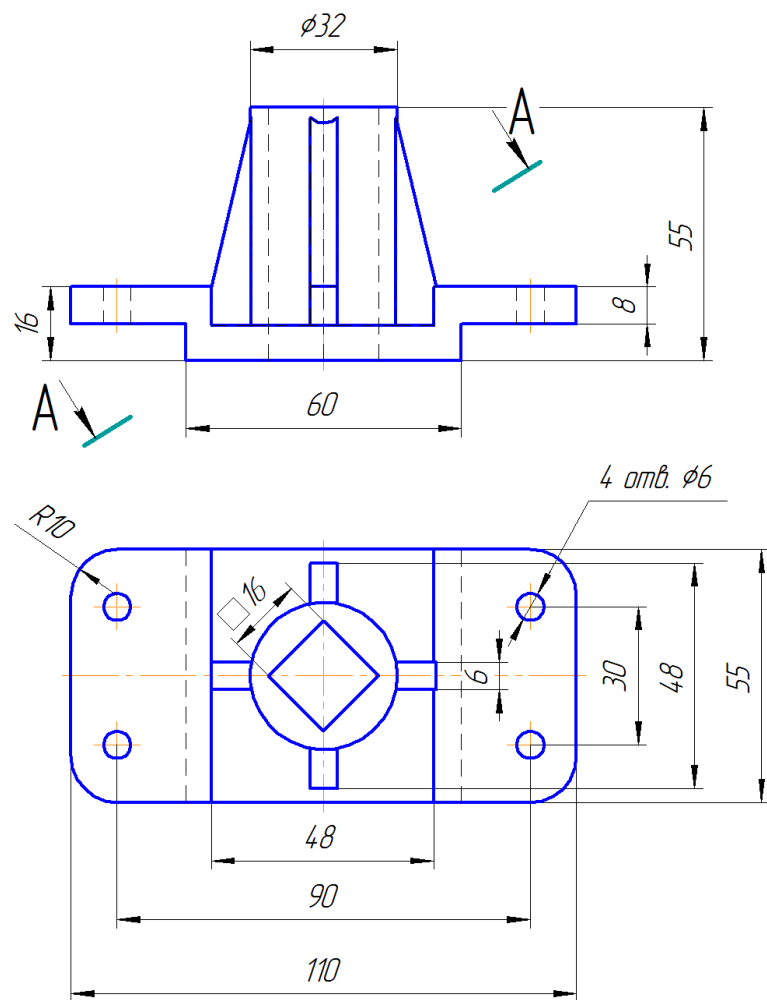
Вариант № 2



Вариант № 3



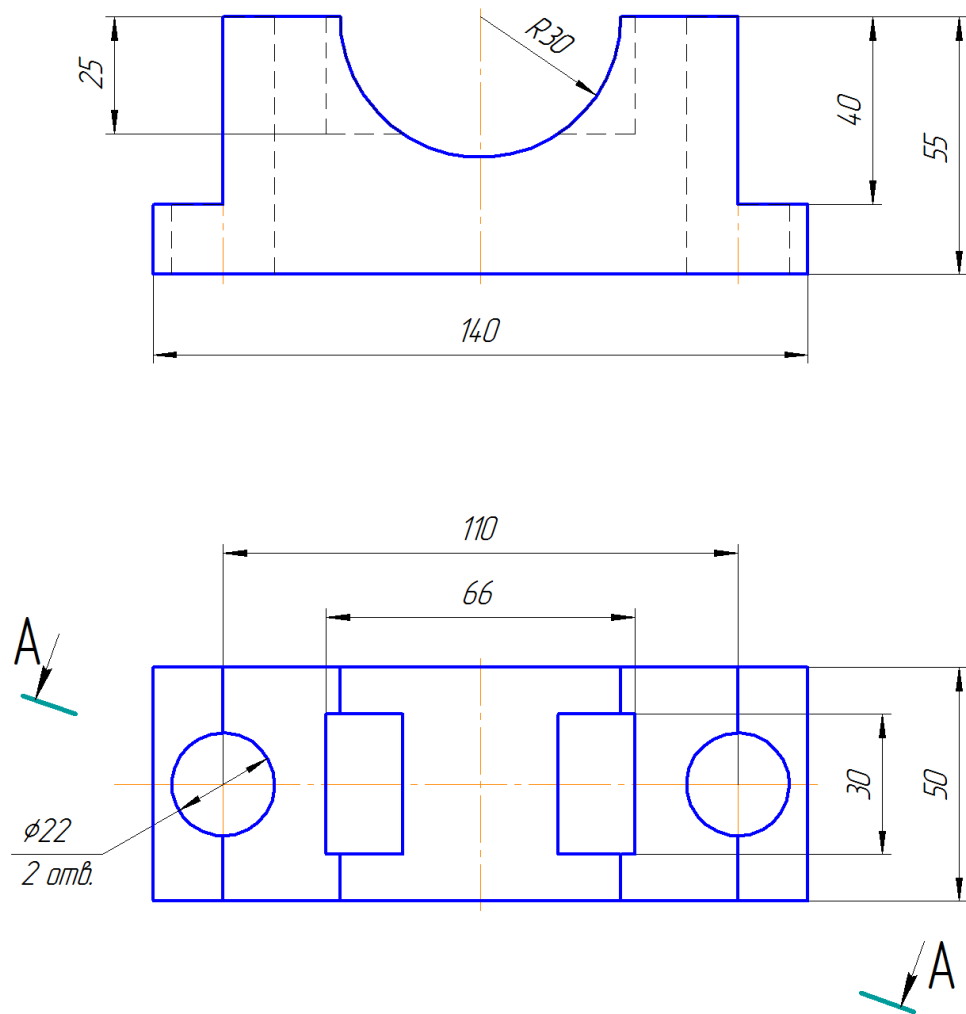
Вариант № 4



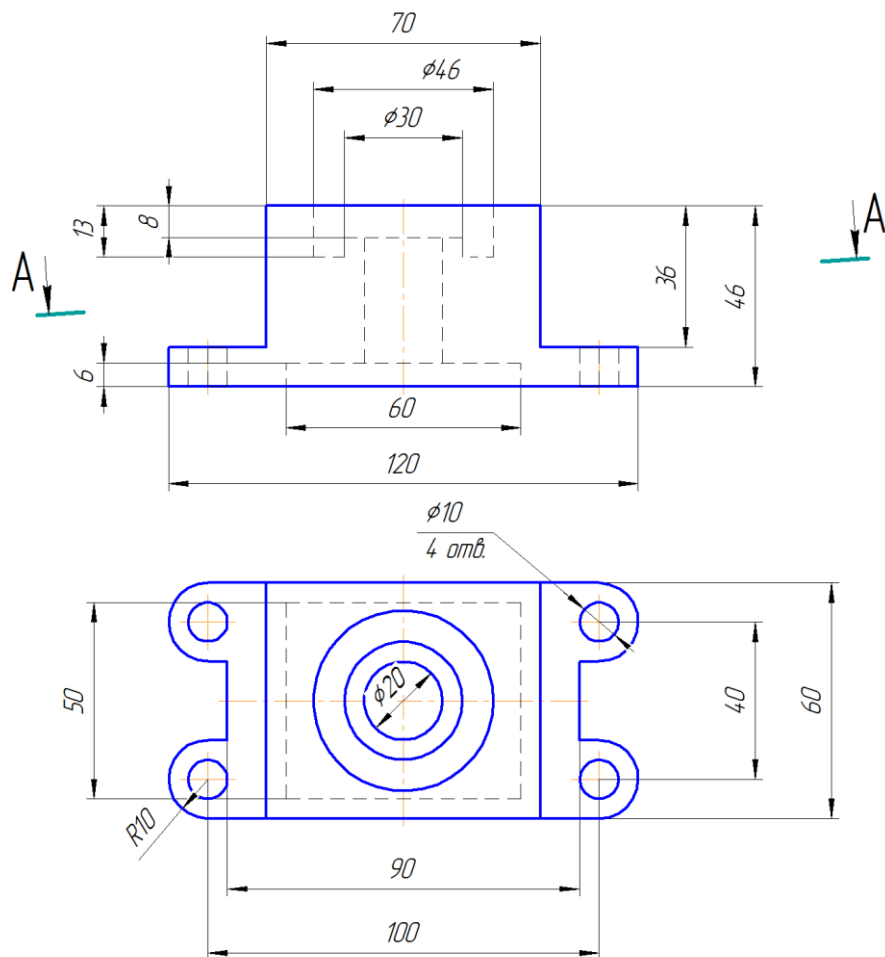
Вариант № 5



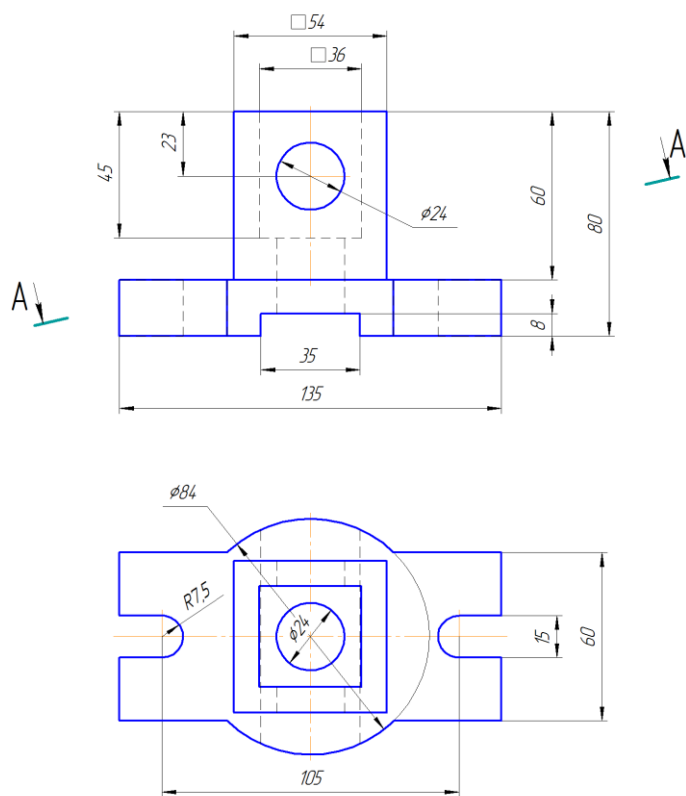
Вариант № 6



Вариант № 7



Вариант № 8



Вариант № 9

СОДЕРЖАНИЕ ОТЧЕТА

Титульный лист.

Бланк задания.

Введение.

1. Теоретические аспекты моделирования реальных объектов.

1.1. Виртуальность как моделирование реального мира.

1.2. Blender - приложение для создание трехмерной графики и анимации.

2. Практическая реализация 3D модели по индивидуальному заданию средствами приложения Blender.

Заключение.

Список литературы.

Приложения. **<файл с моделью загружается вместе с отчетом>**

Список использованных источников

1. Астапчук, В. А. Информационные системы: требования при проектировании: учебное пособие для вузов / В. А. Астапчук, П. В. Терещенко. 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2022. – 113 с. – (Высшее образование). – ISBN 978-5-534-08546-4. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/492141> (дата обращения: 23.03.2022).

2. Боресков, А. В. Основы компьютерной графики: учебник и практикум для вузов / А. В. Боресков, Е. В. Шикин. – Москва: Издательство Юрайт, 2021. – 219 с. – ISBN 978-5-534-13196-3. – Текст: электронный // Образовательная платформа Юрайт – URL: <https://urait.ru/bcode/468914> (дата обращения: 20.04.2022).

3. Грекул, В. И. Проектирование информационных систем: учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. – Москва: Издательство Юрайт, 2019. – 385с. – (Профессиональное образование). – ISBN 978-5-534-12104-9. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/496196> (дата обращения: 22.04.2022).

4. Зализняк, В. Е. Введение в моделирование: учебное пособие для вузов В. Е. Зализняк, О. А. Золотов. – Москва: Издательство Юрайт, 2020. – 133 с. – (Высшее образование). – ISBN 978-5-534-12249-7. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/488304> (дата обращения: 15.03.2022).

5. Зыков, С. В. Программирование: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2020. – 320 с. – (Высшее образование). – ISBN 978-5-534-02444-9. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/489754> (дата обращения: 12.04.2022).

6. Зыков, С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2021. – 155 с. – (Высшее образование). – ISBN 978-5-534-00850-0. –

Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/490423> (дата обращения: 06.05.2022).

7. Зыков, С. В. Программирование. Функциональный подход: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2022. – 164с. – ISBN 978-5-534-00844-9. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/490870> (дата обращения: 22.03.2022).

8. Зыков, С. В. Программирование: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2018. – 320 с. – ISBN 978-5-534-02444-9. – Текст: электронный // Образовательная платформа Юрайт – URL: <https://urait.ru/bcode/469579> (дата обращения: 11.04.2022).

9. Колошкина, И. Е. Компьютерная графика: учебник и практикум для вузов / И. Е. Колошкина, В. А. Селезнев, С. А. Дмитроченко. – 3-е изд., испр. и доп. – Москва: Издательство Юрайт, 2018. – 233 с. – (Высшее образование). ISBN 978-5-534-12341-8. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/490997> (дата обращения: 27.03.2022).

10. Ларин, С. В. Методика обучения математике: компьютерная анимация: учебное пособие для вузов / С. В. Ларин. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2021. – 233 с. – (Высшее образование). – ISBN 978-5-534-08929-5. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/493273> (дата обращения: 22.05.2022).

11. Маликов, Р. Ф. Компьютерное моделирование: учебное пособие для вузов / Р. Ф. Маликов. – Москва: Издательство Юрайт, 2022. – 223 с. – (Высшее образование). – ISBN 978-5-534-14575-5. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/497010> (дата обращения: 01.04.2022).

12. Хворостов, А. С. Художественная обработка дерева: учебник для вузов А. С. Хворостов, Д. А. Хворостов; под общей редакцией А. С. Хворостова. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 248 с. – ISBN 978-5-534-11129-3. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/495131> (дата обращения: 08.04.2022).

13. Хейфец, А. Л. 3D-компьютерная графика в 2 т. Том 2: учебник и практикум для среднего профессионального образования / А. Л. Хейфец, А. Н. Логиновский, И. В. Буторина, В. Н. Васильева ; под редакцией А. Л. Хейфеца. – 3-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2019. – 279 с. – (Профессиональное образование). – ISBN 978-5-534-07974-6. Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/494514> (дата обращения: 22.03.2022).

14. Черпаков, И. В. Теоретические основы информатики: учебник и практикум для вузов / И. В. Черпаков. – Москва: Издательство Юрайт, 2020. 353 с. – ISBN 978-5-9916-8562-7. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/487320> (дата обращения: 22.03.2022).

15. Черпаков, И. В. Теоретические основы информатики: учебник и практикум для вузов / И. В. Черпаков. – Москва: Издательство Юрайт, 2018. 353 с. – ISBN 978-5-9916-8562-7. – Текст: электронный // Образовательная платформа Юрайт – URL: <https://urait.ru/bcode/487320> (дата обращения: 14.05.2022).

16. Чистов, Д. В. Проектирование информационных систем: учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общей редакцией Д. В. Чистова. – Москва: Издательство Юрайт, 2021. – 258 с. – (Профессиональное образование). – ISBN 978-5-534-03173-7. – Текст: электронный // Образовательная платформа Юрайт. – URL: <https://urait.ru/bcode/491568> (дата обращения: 22.03.2022).