

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
«Санкт-Петербургский государственный университет  
аэрокосмического приборостроения»**

---

Методические указания к выполнению лабораторных работ  
по курсу “Проектирование человеко-машинного интерфейса”

Санкт-Петербург  
2017



## Введение

Данное методическое пособие составлено на основе материалов, доступных в сети Internet. При его прочтении рекомендуется посетить сайт <http://blender.org/>, чтобы получить самую свежую версию программного пакета Blender и русской документации к нему. Наиболее полная документация доступна на сайте на английском языке.

## Что такое Blender?

Blender — это интегрированный пакет инструментов, позволяющий создавать 3D контент широкого спектра, обладающий исключительным преимуществом кросс-платформенности и необычайно малым размером скачиваемого файла. Он является самым популярным Open Source приложением для 3D графики во всем мире.

Нацеленный на широкий круг медиа профессионалов и художников, Blender может использоваться для создания 3D визуализации, статических изображений и видео кинематографического качества, а благодаря интегрированному 3D движку реального времени — для создания интерактивного контента для сторонних приложений.

Изначально созданный компанией 'Not a Number' (NaN), сегодня Blender продолжает разрабатываться как 'Свободное ПО' с открытым исходным кодом, доступным по лицензии GNU GPL. В настоящий момент разработкой Blender'a занимается организация Blender Foundation в Нидерландах.

Ключевые особенности:

- Полностью интегрированный пакет разработки, предлагающий широкий выбор инструментов необходимых для создания 3D контента, включая моделирование, uv-маппинг, текстурирование, риджинг, weighting, анимацию, симуляцию частиц и т.п., скриптинг, рендеринг, компоновку, пост-процессинг и создание игр;
- Кросс-платформенность; унифицированный GUI (интерфейс пользователя) для всех платформ за счет использования библиотеки OpenGL; идентичный интерфейс и одинаковая функциональность в Windows (98, NT, 2000, XP), Linux, OS X, FreeBSD, Irix, Sun и множестве других операционных систем;
- Высокое качество 3D архитектуры способствует быстрому и эффективному творческому процессу;
- Небольшой размер исполняемого файла, упрощающий распространение;

Вы всегда можете скачать последнюю версию Blender с сайта <http://blender.org/>.

## История Blender'a

В далеком 1988 году Тон Рузендааль (Ton Roosendaal) стал со-основателем голландской студии анимации *NeoGeo*. NeoGeo быстро стала крупнейшей студией 3D анимации в Нидерландах и одним из лидирующих анимационных домов в Европе. NeoGeo создавала отмеченную призами продукцию (European Corporate Video Awards 1993 и 1995 года) для больших корпоративных клиентов, таких как мульти-национальная компания производитель электроники Philips. Внутри NeoGeo Тон занимался как художественным направлением, так и вопросами разработки внутреннего программного обеспечения. После долгих раздумий Тон решил, что тогдашний набор 3D инструментов слишком устарел и слишком громоздок для дальнейшего обслуживания и разработки, и что необходимо переписать его с нуля. В 1995 году этот процесс начался и со временем стал пакетом 3D моделирования, хорошо известным нам как *Blender*. В процессе того, как NeoGeo продолжала отлаживать и совершенствовать Blender, Тону стало очевидно, что Blender может использоваться в качестве рабочего инструмента и другими художниками, вне компании NeoGeo.

В 1998 году, Тон решает основать новую компанию Not a Number (NaN), как отделение NeoGeo, для дальнейшего продвижения и разработки Blender'a. В сердце NaN жило желание создать компактный, кросс-платформенный пакет 3D моделирования доступный каждому, т.е. распространяемый бесплатно. Тогда как большинство концептуально революционных коммерческих пакетов стоило несколько тысяч американских долларов. NaN надеялась принести инструменты 3D моделирования и анимации профессионального уровня широкому кругу компьютерной публики. Деловая модель NaN строилась на обеспечении коммерческих продуктов и сервисов для Blender'a. В 1999 году NaN собирается на свою первую конференцию Siggraph, с целью представить Blender еще более широкому кругу пользователей. Первая конференция Blender'a на Siggraph 1999 года имела огромный успех и привлекла громадный интерес и прессы, и посетителей. Blender был хитом и его громадный потенциал был подтвержден!

Благодаря успеху Siggraph в начале 2000 года, NaN обеспечила себе финансирование от венчурного капитала в размере €4.5млн. Такое серьезное денежное вливание позволило NaN резко расширить свои возможности. Вскоре NaN насчитывала более пятидесяти сотрудников по всему миру, работающих над улучшением и продвижением Blender'a. Летом 2000 года был выпущен Blender версии 2.0. В этой версии в 3D пакет был добавлен интегрированный игровой движок. А на конец 2000 года число пользователей, зарегистрировавшихся на вебсайте NaN, перевалило за 250.000.

К сожалению, амбиции и возможности NaN не совпали со способностями компании и рыночными реалиями того времени. Такое перерасширение привело к тому, что NaN начала все с начала, с новым инвестором, основав меньшую компанию в апреле 2001 года. Шесть месяцев спустя NaN выпускает свой первый коммерческий программный продукт *Blender Publisher*. Этот продукт был нацелен на зарождающийся рынок интерактивного веб-ориентированного 3D медиа. Но из-за неудовлетворительных продаж и непроходящего тяжелого экономического климата новые инвесторы решают остановить все операции NaN. Что подразумевало, также, прекращение разработки Blender'a. Несмотря на то, что текущая версия Blender'a имела явные недостатки, сложную программную архитектуру, недоработки и нестандартно выполненный пользовательский интерфейс, восторженная поддержка сообщества пользователей и клиентов, купивших Blender Publisher в прошлом, была так сильна, что Тон просто не мог оставить Blender и предать его забвению. Поскольку создать новую компанию с достаточно большой командой разработчиков не представлялось возможным, в марте 2002 года Тон Рузендаль основывает некоммерческую организацию *Blender Foundation*.

Главной целью для Blender Foundation был поиск возможности продолжать разработку и продвижение Blender'a, как общественно-ориентированного Open Source проекта. В июле 2002 года, Тону удается договориться с инвесторами NaN и они соглашаются на план Blender Foundation по переизданию Blender'a с открытым кодом. Кампания "Free Blender" (Освободим Blender) должна была собрать €100.000, чтобы Foundation могла купить права на исходный код Blender'a и права интеллектуальной собственности у инвесторов NaN и, впоследствии, сделать Blender доступным для open source сообщества. Благодаря небольшой группе добровольцев, среди которых были и бывшие сотрудники NaN, был создан специальный фонд для кампании "Free Blender." Ко всеобщему удивлению и восхищению кампания достигла отметки в €100,000 всего за семь недель. И уже в воскресенье 13 октября 2002 года Blender был представлен миру на условиях GNU General Public License (GPL). Разработка Blender'a продолжается и по сей день, силами обширной команды добровольцев со всего мира, под руководством отца-разработчика Blender'a, Тона Рузендаля.

### **О свободном программном обеспечении и GPL**

Когда кто-либо слышит "свободное ПО", то для него это звучит обычно как "бесплатное ПО". Несмотря на то, что это справедливо в большинстве случаев, термин

"свободное ПО", используемый Фондом свободного программного обеспечения (авторами проекта GNU и создателями GNU General Public License), следует понимать как предмет свободы, а не цены (что обычно ассоциируется с "бесплатным мороженым"). Свободное программное обеспечение в этом смысле, это программное обеспечение которое вы можете свободно использовать, копировать, модифицировать, распространять без ограничений. В отличие от лицензий большинства коммерческих пакетов, где вам позволено установить программу на один компьютер, вам позволено сделать ни единой копии и никогда не видеть исходного кода. Свободное программное обеспечение дает удивительную свободу конечному пользователю; к тому же, поскольку исходный код доступен всем, шансы, что баги программы будут отловлены и исправлены, значительно возрастают.

Когда программа лицензирована по лицензии GNU General Public License (the GPL):

- вы имеете право использовать программу с любой целью;
- вы можете изменять программу, и получить доступ к ее исходным кодам;
- вы имеете право копировать и распространять программу;
- вы имеете право улучшать программу, и выпускать свои собственные версии.

Взамен вы несете ответственность, если вы распространяете GPL'ную программу, ответственность предназначенную, чтобы защитить ваши свободы и свободы остальных:

- Вы должны сопровождать программу копией GPL лицензии, чтобы получатель мог ознакомиться со своими правами.
- Вы должны включить исходный код или сделать его общедоступным.
- Если вы вносите модификации в код и распространяете модифицированную версию — вы должны лицензировать ваши модификации лицензией GPL и сделать исходный код ваших модификаций доступным. Вы не можете использовать GPL'ный код как часть проприетарной (платной) программы.
- Вы не можете устрогать лицензирование программы по условиям GPL. (Вы не можете превратить GPL'ную программу в проприетарный продукт.)

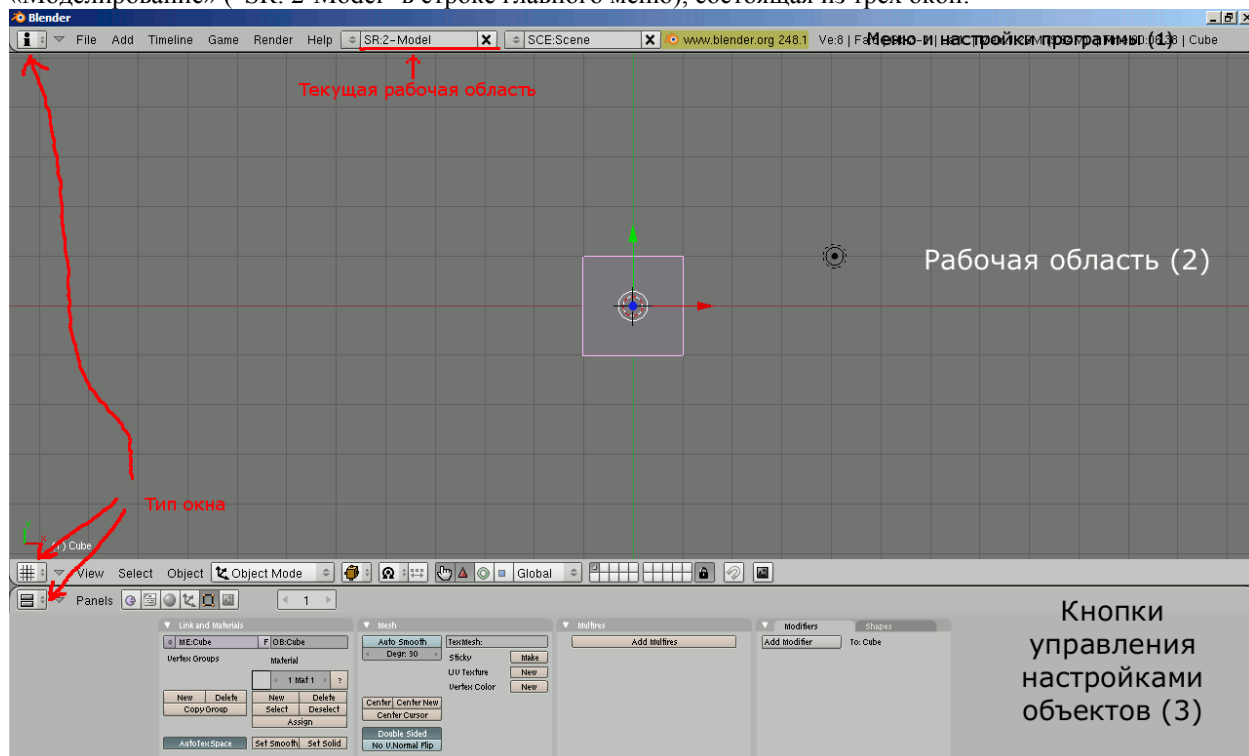
### **Кто использует Blender?**

- Студенты
- люди, для которых изучение мира компьютерной графики (Computer Graphics, CG) и 3D анимации является хобби.
- 2-D художники, которые работают с единичными изображениями или улучшают единичные изображения.
- 2-D художник или целая команда работающая с мультяшными/карикатурными анимациями.
- 3-D художник, который работает сам или в сотрудничестве создает короткие анимационные ролики, возможно, для комбинирования с "живым" видео.
- 3-D команда, занимающаяся производством анимационных (100% CG) фильмов.
- 3-D команда, снимающая классические ("живые") фильмы, что включают в себя элементы компьютерной графики.

Несмотря на то, что Blender предоставляет средства для раскрашивания, они предназначены исключительно для анимации и расцвечивания мешей. Для более детализированных задач, вроде рисования UV текстур или панорамных картин, Blender не соревнуется и не собирается соревноваться в простоте и функциональности со специализированными программами для рисования, как Gimp или Photoshop. Напротив, он беспрепятственно использует генерируемые ими файлы для расцвечивания сцен.

## Работа в пакете Blender

Интерфейс Blender'a состоит из нескольких полноэкранных рабочих областей. Каждая рабочая область в свою очередь состоит из нескольких окон. По умолчанию открывается рабочая область «Моделирование» ("SR: 2-Model" в строке главного меню), состоящая из трех окон:



Можно изменять размер окон перетаскивая разделитель, расположенный между любыми двумя окнами. Также можно добавлять и удалять окна разбивая и объединяя уже существующие. Для этого необходимо нажать правой кнопкой мыши на межоконный разделитель и выбрать действие (split – разбивает окно на два; join – объединяет два окна в одно). Нажав на пиктограмму в левом углу заголовка любого окна, которая обозначает его тип, можно изменить содержимое окна. Таким образом, можно разбить рабочую область на 4 квадрата, указать в меню View в них "Side", "Front", "Top" и "Camera" соответственно, и получить интерфейс, схожий с 3ds Max или Maya.

Полезные сочетания клавиш:

**Ctrl+Up, Ctrl+Down** – окно, над которым находится курсор мыши, разворачивается на весь экран; повторное нажатие сворачивает окно до его прежнего размера;

**Ctrl+Left, Ctrl+Right** – переключение между рабочими областями (всего их 5, но можно создавать дополнительные пользовательские рабочие области);

**MWheel (колесико мыши)** – изменение масштаба рабочей области;

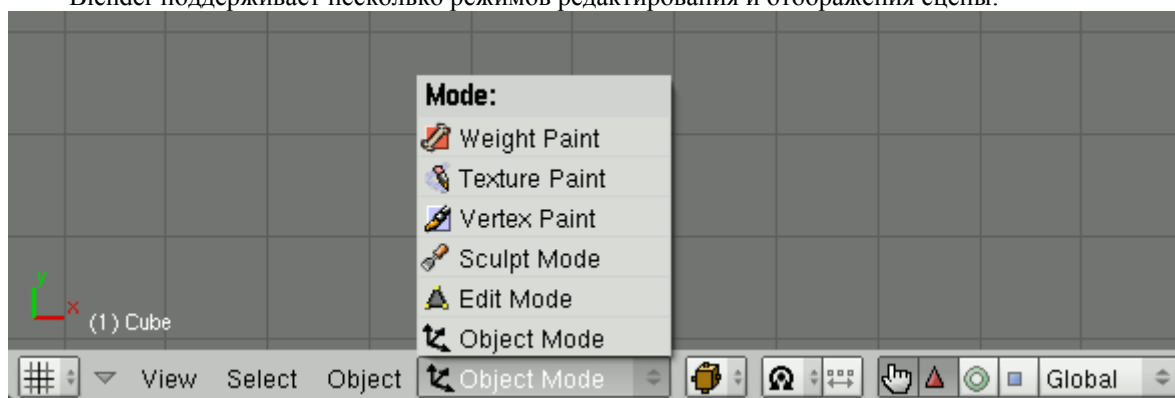
**MMB (Средняя кнопка мыши, нажатие на колесико)** – поворот точки обзора в рабочей области;

**Shift+MMB** – перемещение точки обзора в рабочей области.

**Home** – изменить масштаб рабочей области таким образом, чтобы все объекты поместились на экране;

В меню View в рабочей области можно изменять свойства отображения объектов в рабочей области.

Blender поддерживает несколько режимов редактирования и отображения сцены:



В режиме Object Mode можно оперировать целыми объектами, а режим Edit Mode позволяет оперировать отдельными вершинами. Для переключения между этими двумя режимами можно использовать клавишу **Tab**. Остальные режимы используются при создании анимации, рисовании на полигональной сетке и т.д.

**RMB (правая кнопка мыши)** осуществляет выделение объекта (в Object Mode) или вершины (в Edit Mode).

**Shift+RMB** – множественное выделение.

**B** – прямоугольное выделение.

**A** – выделить все объекты/снять выделение.

**Shift+D** – создать копию выделенного объекта (-ов).

**N** – показать/скрыть текущие трансформации объекта (координаты, углы поворота по осям и т.п.).

**Ctrl+Alt+G** – перемещение выделенного объекта левой кнопкой мыши в рабочей области (сдвиг).

**Ctrl+Alt+R** – вращение выделенного объекта (поворот).

**Ctrl+Alt+S** – изменение размера выделенного объекта (масштабирование).

**Space** – открыть меню редактирования сцены.

## Задания на лабораторные работы

Для выполнения лабораторных работ необходимо установить Blender и Python. Скачать последние версии можно с сайтов <http://blender.org> и <http://python.org> соответственно. Следует обратить внимание на версию python, которую использует скачанная версия Blender, т.к. с более новыми версиями “питона” Blender работать не будет!

### ЛР1. Основы работы в среде трехмерного моделирования Blender.

**Цель работы:** освоение работы со средой трехмерного моделирования Blender, изучение базовых трансформаций, рендеринг.

**Задание:**

1. Создать сцену, состоящую из трех любых объектов.
2. Ознакомиться с инструментами манипулирования сценой:
  - a. выбор объекта,
  - b. применение базовых трансформаций (перенос, поворот, масштабирование),
  - c. рендеринг сцены.
3. Сохранить сцену в файл, загрузить ее из файла.

### ЛР2. Сборка сцены из объектов.

**Цель работы:** освоение сборки сцены из объектов, импорт объектов, работа со шрифтами.

**Задание:**

Создать сцену, состоящую как минимум из 7 различных объектов. **Ни один из объектов сцены не должен быть базовым**, т.е. все объекты должны быть либо загружены из файла (уже готовые сложные объекты), либо получены из базовых объектов Blender (куб, сфера, плоскость и т.п.) путем трансформаций. Добавить в сцену объект-надпись с текстом “<Имя> <Фамилия>, <номер группы>” и задать следующие свойства этого объекта:

Параметр	Значение
Наклон текста	-0.5
Extrude	0.30
Bevel Depth	0.05
Bevel Resolution	4

**Имя и фамилия должны быть написаны русскими буквами**, без транслитерации! Для этого необходимо в свойствах объекта-надписи выбрать любой TrueType шрифт, поддерживающий набор символов Unicode (например, C:\Windows\Fonts\Arial.ttf) и посимвольно ввести имя и фамилию из таблицы символов Unicode, т.к. переключение раскладки в Blender не работает.

### ЛР3. Работа с шейдерами в среде трехмерного моделирования

**Цель работы:** освоение принципов работы с материалами, текстурами, шейдерами, работа с камерой.

**Задание:**

Задать различные свойства шейдеров для объектов собственной сцены. Получить эффекты зеркальных, металлических и стеклянных поверхностей. Наложить текстуры, загруженные из внешних файлов. Создать несколько камер и осуществить рендеринг сцены из разных точек.



## ЛР4. Создание рельефа, рендеринг больших изображений.

**Цель работы:** освоение способов создания рельефа в Blender, изучение возможностей скриптового языка Python, настройка параметров рендеринга.

### Задание:

Создать рельеф двумя различными способами:

1. Сгенерировать рельеф на основе карты высот с помощью модификаторов:
  - а) создать объект Plane;
  - б) применить к нему один или несколько модификаторов SubSurf, чтобы увеличить число вершин с 4 до нескольких сотен или тысяч;
  - в) после SubSurf применить модификатор Displace, указав в поле Texture текстуру, являющуюся картой высот
2. С помощью скрипта Blender World Forge Tool (BWF) сгенерировать рельеф. Добавить кратеры и пики. Для того чтобы воспользоваться скриптом необходимо следующее:
  - а) Установить Python соответствующей Blender'у версии.
  - б) Перезапустить Blender и убедиться, что в консоли появилась строка **Checking for installed Python... got it!**
  - в) Открыть в Blender'е окно типа **Text Editor**
  - г) Alt+O – открыть текстовый файл (выбрать файл скрипта BWF-0.1.0.py)
  - д) Alt+P – выполнить скрипт
  - е) в окне, где был Text Editor, появится новое окошко, в котором необходимо задать настройки генерируемого рельефа и нажать кнопку TERRAFORM

Настроить параметры рендеринга, получить изображение не хуже 1600\*1200 pix, осуществить рендеринг в файл в формате JPEG.

## ЛР5. Использование различных типов освещения.

**Цель работы:** освоение работы с источниками света, настройка общих свойств сцены.

**Задание:** используя сгенерированный рельеф из предыдущей работы, добавить источник света Sun (Солнце). Настроить положение Солнца, камеры и рельефа таким образом, чтобы можно было наблюдать восход, закат, солнечный диск над горами и звездную ночь с силуэтами гор. Осуществить рендеринг и построить ряд изображений одной сцены для разной высоты солнца (см. рисунки).



## ЛР6. Анимация.

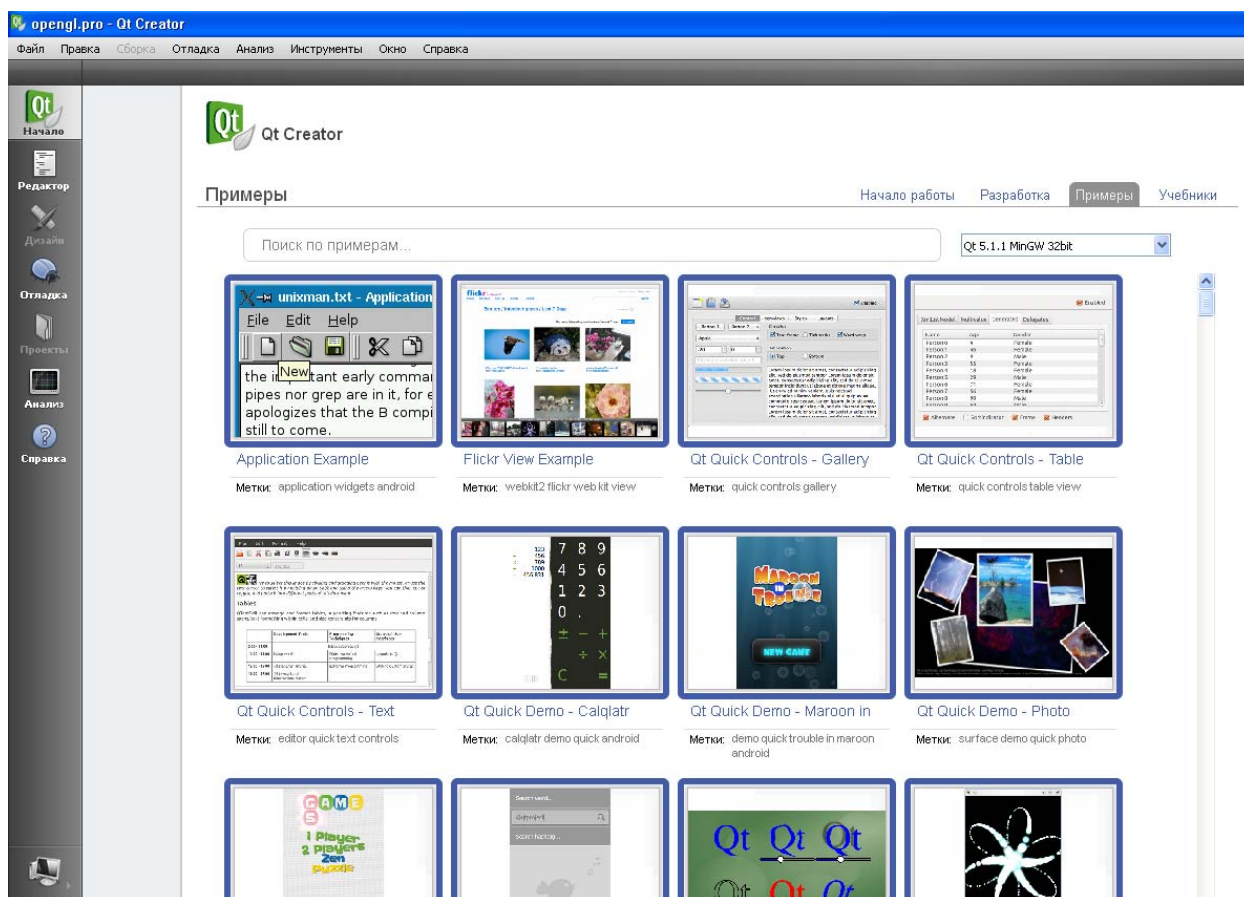
**Цель работы:** создание анимированного изображения, рендеринг видеоролика.

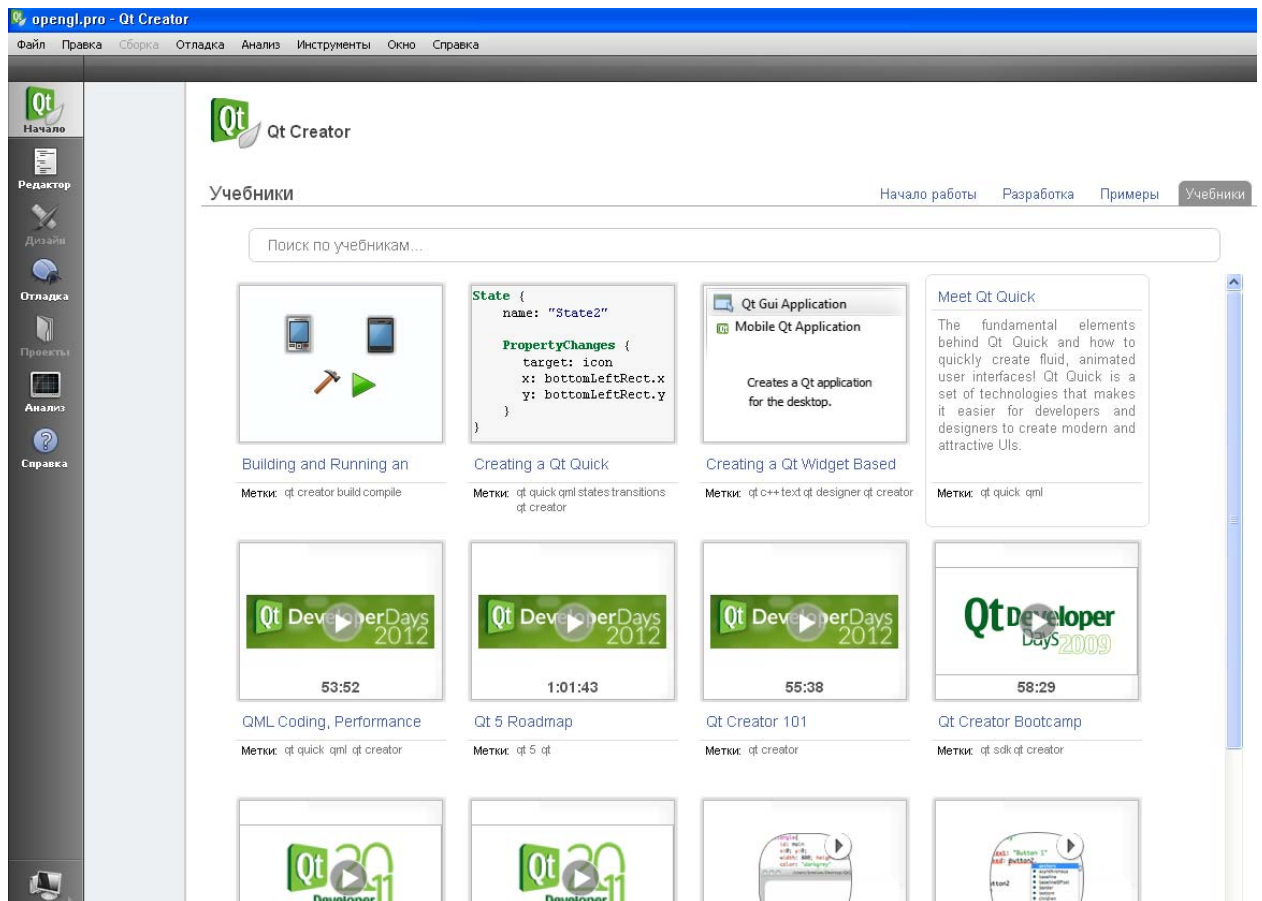
**Задание:** Построить сцену, несколько раз изменить положение или масштаб объектов для получения ключевых кадров видеоряда. Отладить анимацию путем проигрывания видеоряда без рендеринга и сглаживания графика IPO Curve. Осуществить финальный рендеринг видеоролика с использованием выбранного видеоформата, качества изображения и кодека.

### Среда Qt Creator

Qt Creator – кросс-платформенная среда для разработки приложений с двумерным и трехмерным графическим интерфейсом. Среда свободно распространяется вместе с вариантом библиотеки Qt с открытыми исходными текстами в различных вариантах для различных операционных систем.

Для выполнения лабораторных работ рекомендуется использовать вариант Qt Creator IDE for Windows MinGW, который содержит полный комплект необходимых инструментов и библиотек вместе с компилятором C/C++ MinGW. Данный вариант содержит подробную документацию, учебные материалы и большое количество примеров программ, иллюстрирующих самые разные аспекты создания пользовательских интерфейсов программ, включая интеграцию Qt с вызовами трехмерной графической библиотеки Open GL, использование основных и расширенных виджетов Qt для создания двумерного графического пользовательского интерфейса, применение декларативного языка QML для разработки пользовательского интерфейса.



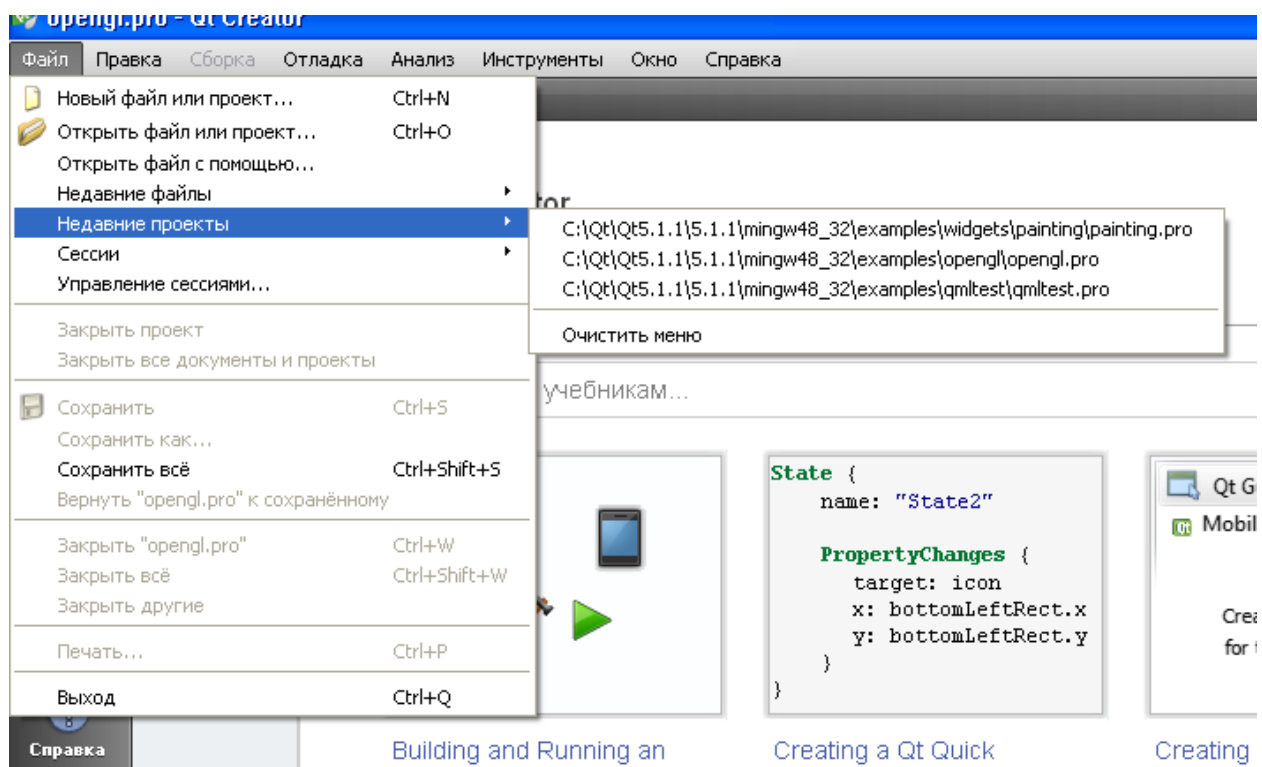


Qt\Qt\*.\*.\*.\*.mingw##\_32\examples - общий каталог с примерами

\*.\*.\* - версия библиотеки Qt

##\_ - версия компилятора MinGW

Например, примеры программ с Open GL находятся в каталоге Qt\Qt\*.\*.\*.\*.mingw##\_32\examples\opengl\



При работе в ОС Windows рекомендуется загрузить и установить off-line инсталлятор свободно распространяемой среды разработки Qt Creator с библиотеками Qt 5.1.1 и Open GL и компилятором MinGW 4.8 со страницы

<http://download.qt.io/archive/qt/5.1/5.1.1/>

Не рекомендуется использовать On-line инсталляторы Qt и Qt-creator. При недостаточной квалификации и неполном выборе или неполной загрузке необходимых компонентов может произойти некорректная установка и неправильная настройка среды.

Другие версии доступны на странице

<http://download.qt.io/archive/qt/>

На наш взгляд, для выполнения лабораторных работ вполне достаточно рекомендуемой версии Qt Creator с библиотеками и компилятором, которые совместимы с 32 и 64-битными версиями Windows (в т.ч. Windows 10)

## **ЛР7. Создание двумерного пользовательского интерфейса в среде Qt Creator**

**Цель работы:** освоение кросс-платформенной интегрированной среды разработки, изучение базовых возможностей разработки и отладки приложения с двумерным пользовательским интерфейсом с использованием библиотеки Qt и компилятора MinGW

**Задание:** Разработать и отладить приложение на языке C++ с графическим пользовательским интерфейсом, использующим основные виджеты Qt

### **Основные виджеты**

Основные виджеты (элементы управления), например, кнопки, выпадающие списки и полосы прокрутки, спроектированы для непосредственного использования.

<a href="#"><u>QCheckBox</u></a>	Флажок с текстовой меткой
<a href="#"><u>QComboBox</u></a>	Сочетание кнопки и всплывающего списка
<a href="#"><u>QCommandLinkButton</u></a>	Командная кнопка-ссылка в стиле Vista
<a href="#"><u>QDateEdit</u></a>	Виджет для редактирования даты, основанный на виджете QDateTimeEdit
<a href="#"><u>QDateTimeEdit</u></a>	Виджет для редактирования даты и времени
<a href="#"><u>QDial</u></a>	Элемент управления, проградуированный по окружности (наподобие спидометра или потенциометра)
<a href="#"><u>QDoubleSpinBox</u></a>	Виджет счетчика, принимающего значения с плавающей точкой
<a href="#"><u>QFocusFrame</u></a>	Рамка фокуса, которая может отображаться вокруг стандартной области прорисовки виджета
<a href="#"><u>QFontComboBox</u></a>	Выпадающий список, который позволяет пользователю выбрать семейство шрифтов
<a href="#"><u>QLCDNumber</u></a>	Отображает число с помощью цифр, имитирующих ЖК-индикатор
<a href="#"><u>QLabel</u></a>	Отображает текст или рисунок
<a href="#"><u>QLineEdit</u></a>	Однострочный редактор текста
<a href="#"><u>QMenu</u></a>	Виджет меню, используемый в панели меню, контекстном меню и других всплывающих меню
<a href="#"><u>QProgressBar</u></a>	Горизонтальный или вертикальный индикатор выполнения
<a href="#"><u>QPushButton</u></a>	Командная кнопка
<a href="#"><u>QRadioButton</u></a>	Радио-кнопка с текстовой меткой
<a href="#"><u>QScrollArea</u></a>	Область прокрутки на другом виджете
<a href="#"><u>QScrollBar</u></a>	Вертикальная или горизонтальная полоса прокрутки
<a href="#"><u>QSizeGrip</u></a>	Область захвата для изменения размера окна верхнего уровня
<a href="#"><u>QSlider</u></a>	Вертикальный или горизонтальный ползунок (slider)
<a href="#"><u>QSpinBox</u></a>	Виджет счетчика

<a href="#"><u>QTabBar</u></a>	Панель вкладок для использования, например, в диалогах со вкладками
<a href="#"><u>QTabWidget</u></a>	Стек виджетов со вкладками
<a href="#"><u>QTimeEdit</u></a>	Виджет для задания времени, основанный на виджете QDateTimeEdit
<a href="#"><u>QToolBox</u></a>	Вертикальный набор элементов виджетов со вкладками
<a href="#"><u>QToolButton</u></a>	Кнопка быстрого доступа к командам или настройкам, обычно используется в QToolBar
<a href="#"><u>QWidget</u></a>	Базовый класс для всех объектов интерфейса пользователя

## **LP8. Создание двумерного пользовательского интерфейса в среде Qt Creator с использованием расширенных виджетов Qt**

**Цель работы:** освоение использования расширенных виджетов Qt для создания пользовательского интерфейса

**Задание:** Разработать и отладить приложение на языке C++ с графическим пользовательским интерфейсом, использующим расширенные виджеты Qt

### **Расширенные виджеты**

Расширенные виджеты графического пользовательского интерфейса предоставляют более сложные элементы управления пользовательским интерфейсом.

<a href="#"><u>QCalendarWidget</u></a>	Виджет помесячного календаря, позволяющий пользователю выбрать дату
<a href="#"><u>QColumnView</u></a>	Реализация идеи модель/представление в виде постолбцового представления
<a href="#"><u>QDataWidgetMapper</u></a>	Отображение области данных модели на виджеты
<a href="#"><u>QDesktopWidget</u></a>	Доступ к информации об экране в многоэкранных системах
<a href="#"><u>QListView</u></a>	Представление списка или пиктограмм в модели
<a href="#"><u>QMacCocoaViewContainer</u></a>	Виджет для Mac OS X, который может использоваться для обертывания произвольных представлений Cocoa (т.е. подклассов NSView) и вставки их в иерархии Qt
<a href="#"><u>QMacNativeWidget</u></a>	Виджет для Mac OS X, который предоставляет способ помещения виджетов Qt в иерархии Carbon или Cocoa, в зависимости от того, как Qt была сконфигурирована
<a href="#"><u>QTableView</u></a>	Реализация по умолчанию модели/представления таблицы
<a href="#"><u>QTreeView</u></a>	Реализация модели/представления по умолчанию для представления дерева
<a href="#"><u>QUndoView</u></a>	Отображает содержимое QUndoStack
<a href="#"><u>QWSEmbedWidget</u></a>	Предоставляет возможность встраивать виджеты верхнего уровня в Qt для встраиваемых Linux-систем
<a href="#"><u>QWebView</u></a>	Виджет, используемый для просмотра и редактирования веб-документов
<a href="#"><u>QX11EmbedContainer</u></a>	Виджет-контейнер XEmbed
<a href="#"><u>QX11EmbedWidget</u></a>	Виджет-клиент XEmbed
<a href="#"><u>Phonon::VideoWidget</u></a>	Виджет, используемый для отображения видео

### **Виджеты упорядочения**

Такие классы, как разделители, панели вкладок, группы кнопок и т.д. используются для упорядочивания и группировки примитивов в более сложных приложениях и диалогах.

<a href="#"><u>QButtonGroup</u></a>	Контейнер для управления группой виджетов-кнопок
<a href="#"><u>QGroupBox</u></a>	Рамка с заголовком для панели-контейнера
<a href="#"><u>QSplitter</u></a>	Реализует виджет-разделитель
<a href="#"><u>QSplitterHandle</u></a>	Обработка функциональности разделителя
<a href="#"><u>QStackedWidget</u></a>	Стек виджетов, в котором в один момент виден только один виджет
<a href="#"><u>QTabWidget</u></a>	Стек виджетов со вкладками

## **Абстрактные классы виджетов**

Абстрактные классы виджетов являются базовыми классами. Они не используются в качестве самостоятельных классов, но предоставляют функциональность при создании их наследников.

<a href="#"><u>QAbstractButton</u></a>	Абстрактный базовый класс виджета кнопки, реализующий общие функции кнопок
<a href="#"><u>QAbstractScrollArea</u></a>	Прокручиваемая область с полосами прокрутки, отображаемыми по требованию
<a href="#"><u>QAbstractSlider</u></a>	Целое значение из диапазона
<a href="#"><u>QAbstractSpinBox</u></a>	Счетчик и строка редактирования для отображения значения счетчика
<a href="#"><u>QDialog</u></a>	Базовый класс для диалоговых окон
<a href="#"><u>QFrame</u></a>	Базовый класс для виджетов, способных иметь рамку

## **ЛР9. Создание двумерного пользовательского интерфейса в среде Qt Creator с использованием QML**

**Цель работы:** освоение QML для создания пользовательского интерфейса

**Задание:** Разработать и отладить приложение с графическим пользовательским интерфейсом, использующим для описания интерфейса язык QML

### **Требования к защите лабораторных работ**

На защите лабораторных работ 1-6 студент демонстрирует подготовленную сцену, отчет и по просьбе преподавателя объясняет и показывает, как выполнять различные манипуляции, использованные при разработке сцены.

На защите лабораторных работ 7-9 студент демонстрирует исходные тексты, работу приложения, отчет и по просьбе преподавателя модифицирует элементы пользовательского интерфейса.

### **Требования к отчету для ЛР 1-6**

Содержание отчета

- 1) титульный лист
- 2) задание
- 3) название и версия используемой среды моделирования
- 4) словесное описание сцены
- 5) Описание технологии создания сцены. Описание должно быть таким, чтобы руководствуясь им можно было получить похожую сцену.
- 6) копии экранов с результатами работы
- 7) выводы, содержащие анализ проделанной работы: эффективность инструментов, удобство работы, качество и общий вид полученной сцены по сравнению с задуманной идеей.

### **Требования к отчету для ЛР 7-9**

Содержание отчета

- 1) титульный лист
- 2) задание
- 3) название используемой среды разработки
- 4) исходный текст приложения
- 5) описание структуры интерфейса
- 6) выводы, содержащие анализ проделанной работы

Отчеты, предоставляются в формате doc, rtf, odg

## **Источники**

<http://www.blender.org/>

<http://python.org>

<http://ru.wikipedia.org/wiki/Blender>

[http://en.wikipedia.org/wiki/Blender\\_\(software\)](http://en.wikipedia.org/wiki/Blender_(software))

<http://www.qt.io/developers/>