

УДК 004.414.2

Баширов А.С. Информационная система распределения учебной нагрузки / **Выпускная квалификационная работа бакалавра** 09.03.04. «Программная Инженерия» / руководитель профессор Нечаев В.В. / кафедра ИППО Института ИТ МИРЭА – с. 50, илл. 21, табл. 12, формул 10, ист. 20.

Выпускная квалификационная работа на тему «Система моделирования интерьеров: исследование и разработка».

Ключевые слова: автоматизация; визуальное моделирование; концептуальная модель.

Final qualifying work "Interiors modeling system: research and development".

Keywords: automation; visual modeling; conceptual model

Целью работы является исследование аспектов визуального моделирования и разработка системы моделирования интерьеров.

В первом разделе были проанализированы теоретические аспекты визуального моделирования.

Во втором разделе была построена концептуальная модель системы.

В третьем разделе был рассмотрен модуль визуализации.

В четвертом разделе была проанализирована экономическая целесообразность системы

МИРЭА: 119454, Москва, пр-т Вернадского, д. 78

кафедра Инструментального и прикладного программного обеспечения (ИППО)

Тираж: 1 экз. (на правах рукописи)

Файл: «090304\_14И0027\_Баширов.doc», исполнитель Баширов А.С.

@ Баширов

## Оглавление

Введение.....	7
Раздел 1. Теоретические основы принципов и механизмов визуального моделирования.....	9
1.1 Создание 2D объектов .....	10
1.2 Создание 3D объектов.....	11
1.3 Создание систем частиц .....	12
1.4 Анимирование 3D объектов.....	12
1.5 Создание источников освещения .....	13
1.6 Моделирование объектов.....	14
1.7 Моделирование систем.....	15
Выводы по разделу .....	16
Раздел 2. Концептуальная модель системы визуально моделирования интерьеров.....	17
2.1 Описание атрибутов и концептуальная схема данных объектов.....	19
2.1 Выбор технологий реализации.....	24
2.2 Архитектура программной системы.....	24
Выводы по разделу .....	25
Раздл 3. Реализация модуля визуализации.....	26
3.1 Алгоритм сохранения интерьера.....	26
3.2 Алгоритм загрузки интерьера.....	26
3.3 Алгоритм создания объекта.....	27
3.4 Алгоритм замены объекта.....	27
3.5 Алгоритм изменения материала объекта.....	27
3.6 Алгоритм перемещения объекта .....	27
3.7 Алгоритм вращения объекта.....	28
3.8 Алгоритм растяжения и сужения объекта.....	28
3.9 Алгоритм удаления объекта.....	29
3.10 Пользовательский интерфейс программы.....	29
3.11 Тестирование разрабатываемого ПО .....	34
Выводы по разделу .....	38
Раздел 4. Экономическая часть.....	39
4.1 Анализ рынка сбыта продукта.....	39
4.2 План производства.....	39

4.3 Организационный план .....	40
4.4 Расчет затрат и договорной цены .....	42
4.5 Материалы и покупные изделия.....	42
4.6 Техническое оборудование .....	43
4.7 Заработная плата персонала.....	43
4.8 Страховые взносы .....	45
4.9 Накладные расходы .....	46
4.10 Себестоимость проекта .....	46
4.11 Договорная цена.....	46
4.12 Обоснование экономической целесообразности проекта.....	46
Выводы по разделу .....	48
Заключение .....	49
Список используемых источников.....	50

## Введение

В рамках выпускной квалификационной работы будут рассмотрены основные аспекты визуального моделирования на примере создания системы моделирования интерьеров.

Цель работы заключается в исследовании аспектов визуального моделирования на примере разработки системы моделирования интерьеров.

В процессе достижения данной цели будут выполнены следующие задачи:

- проведение анализа понятия и принципов визуального моделирования;
- построение концептуальной схемы объектов интерьера;
- разработка системы моделирования объектов интерьера.

Сформулируем определение визуализации.

Визуальное моделирование – рекурсивный процесс формирования визуального представления исходного объекта моделирования или текущего представления объекта с целью получения информации и знаний посредством визуального анализа полученной модели.

Распишем компоненты данного определения:

- ВМ является рекурсивным процессом. Это означает, что операцию визуализации можно применять к полученному ранее её результату;
- результатом ВМ является представление. Представление – совокупность доступных к восприятию человеческим глазом атрибутов исходного объекта;
- ВМ может применяться как к реальному объекту, так и к идее, сформулированной мыслительным процессом человеческого разума с целью получения её визуального представления;
- к полученной модели возможно применение визуального анализа. Данный вид анализа заключается в рассмотрении всех атрибутов представления с целью получения максимума информации об исходном объекте моделирования.

На текущий момент визуальное моделирование применяется практически в любой деятельности. Более того, каждый из имеющихся видов моделирования не может быть рассмотрен без применения визуального моделирования. Можно сказать, что ВМ – является средством реализации моделирования и мета-моделирования.

Визуализация позволяет строить визуальные трёхмерные и двумерные модели объектов произвольных форм и различного уровня сложности. Визуализация решает следующие задачи:

- 1) задача передачи и сохранения информации о форме и других визуальных характеристиках объекта;
- 2) задача создания визуального представления объекта на основе данных о нём;
- 3) задача определения способности сочетания объектов друг с другом;

- 4) задача определения способности восприятия объектов в зависимости от их формы, расположения в пространстве и других визуальных характеристик;
- 5) задача демонстрации объектов;
- 6) задача изучения явлений и объектов путём анализа их визуальной модели.

Основная цель визуального моделирования – донести до пользователя максимум возможной информации о визуальной составляющей объекта моделирования, а также дать пользователю модели возможность изменять конфигурацию одних её компонентов для определения всевозможных реакций других.

Технологии визуального моделирования позволяют выполнять следующие действия:

- 1) создание визуальной модели объекта на основе имеющихся примитивов;
- 2) создание композиции из готовых моделей;
- 3) моделирование освещения;
- 4) моделирование движения отдельных компонентов модели или моделей внутри пространства;
- 5) моделирование физических законов;
- 6) моделирование материалов;
- 7) подбор различных комбинаций цветов и демонстрация полученной композиции;
- 8) создание компьютерной модели на основе реального объекта;
- 9) моделирование явлений, таких как огонь или молния с возможностью регулировки их параметров (интенсивность, цвет и т.д.);
- 10) управление полученной композицией моделей и изменение различных параметров моделей.

Применение визуального моделирования удачно демонстрируется на примере программы создания и использования визуальной модели интерьеров. Модели такого рода представляют собой композицию из объектов интерьера, размещенных внутри трёхмерного пространства. Объектами могут служить предметы мебели, светильники, электроника и т.д.

В последующих разделах рассматривается концептуальная модель процесса визуального моделирования, описывается схема сущностей, из которых создаётся визуальная модель интерьеров, а также демонстрируется пример реализации программного модуля визуализации модели.

## **Раздел 1. Теоретические основы принципов и механизмов визуального моделирования**

Компьютерное визуальное моделирование (КВМ) – процесс формирования визуального представления объекта посредством применения компьютерных технологий [1].

Приведем основные понятия и термины компьютерного моделирования.

Двумерная графика – графика, действие в которой происходит в одной плоскости. Например, пользовательский интерфейс.

Трехмерная графика – визуальное отображение трехмерной сцены или объекта. Для представления трехмерной графики на двумерном устройстве (дисплей) применяют рендеринг.

Цвет – это индивидуальные компоненты белого света, по-разному воспринимаемые человеческим глазом. Цветные мониторы используют три основных компонента цвета, на которые реагирует человеческий глаз: красный, зеленый и голубой. Цвет, который в итоге отображается на экране, образуется в результате смешения этих трех основных цветов.

Компьютерная графика – общее направление, описывающее создание или манипуляцию графическими изображениями и изобразительными данными с помощью компьютера. Может использоваться в САД, анимации, дизайне, архитектуре, деловой графике и т.д. Системы для компьютерной графики обычно являются интерактивными, т.е. отображают изображение на дисплее таким, каким оно создано, или в виде, в который преобразована исходная картинка.

Пиксел – комбинированный термин, обозначающий элемент изображения, являющийся наименьшим элементом экрана монитора.

Изображение на экране состоит из сотен тысяч пикселей, объединенных для формирования изображения. Пиксель является минимальным сегментом растровой строки, которая дискретно управляется системой, образующей изображение. С другой стороны, это координата, используемая для определения горизонтальной пространственной позиции пикселя в пределах изображения. Пиксели на мониторе – это светящиеся точки яркого фосфора, являющиеся минимальным элементом цифрового изображения. Размер пикселя не может быть меньше точки, которую монитор может образовать. На цветном мониторе точки состоят из групп триад. Триады формируются тремя различными фосфорами: красным, зеленым и синим. Фосфоры располагаются вдоль сторон друг друга. Пиксели могут отличаться размерами и формой, в зависимости от монитора и графического режима. Количество точек на экране определяются физическим соотношением ширины к высоте трубки.

Проекция – процесс преобразования трех размерностей в две. Т.е. преобразование видимой части 3D объекта для отображения на двумерном дисплее.

Растреризация – разделение объекта на пиксели.

Рендеринг – процесс создания реалистичных изображений на экране, использующий математические модели и формулы для добавления цвета, тени и т.д.

Текстура – двумерное изображение, хранящееся в памяти компьютера или графического акселератора в одном из пиксельных форматов. В случае хранения в сжатом виде на дисках компьютера, текстура может представлять собой обычный битмап, который мы привыкли видеть в форматах bmp, jpg, gif и т.д. Перед использованием, текстура разворачивается в памяти и может занимать объем в десятки раз больше первоначального размера. Существует порядка двух десятков более или менее стандартизированных пиксельных форматов текстур.

КВМ может быть, как двумерным, так и трёхмерным. Рассмотрим основные виды компьютерного моделирования.

### ***1.1 Создание 2D объектов***

Под созданием 2D объектов понимается «рисование» текстур различных типов [2].

Текстуры могут быть следующих типов:

- текстура, определяющая основной цвет модели;
- текстура, определяющая зеркальную интенсивность и цвет материала;
- текстура, задающая направление нормалей на поверхности. Нормальное направление для каждого пикселя текстуры хранятся в цвета RGB карты нормалей;
- текстура детализации, позволяющая добавить больше деталей к поверхности;
- спрайт – управляемый 2D объект пространства, который представляет собой анимированную текстуру.

Пример текстур изображен на рисунке 1.1.



Рисунок 1.1 Различные типы текстур

## 1.2 Создание 3D объектов

3D объект – геометрическая фигура в пространстве, состоящая из следующих компонентов [3]:

- вершина – это абстрактная геометрическая точка с координатами X, Y и Z. Она является крайней точкой либо замкнутого полигона (плоского многоугольника), либо объёмной фигуры;
- грань – отрезок прямой, соединяющий две вершины. В трёхмерной графике это не самостоятельное нечто, а лишь ограничитель для полигонов;
- полигон – основная функциональная составляющая: плоская многоугольная фигура (обычно трёхмерные редакторы и другие приложения предпочитают оперировать только треугольниками и четырёхугольниками), из множества которых состоит поверхности трёхмерной фигуры.

Любую даже самую сложную фигуру или поверхность можно представить в виде множества простых фигур: такая идеальная фигура, как шар (точнее, сфера), в компьютерной графике в любом случае представляется в виде множества треугольников и четырёхугольников. Чем их будет больше, тем выше степень приближения, то есть, тем более гладкой, тем более сферической будет поверхность.

Пример 3D модели изображен на рисунке 1.2.



Рисунок 1.2 3D модели



### ***1.3 Создание систем частиц***

Система частиц — используемый в компьютерной графике способ представления 3D объектов, не имеющих чётких геометрических границ (различные облака, туманности, взрывы, струи пара, шлейфы от ракет, дым, снег, дождь и т. п.). [4]

Пример системы частиц изображен на рисунке 1.3.

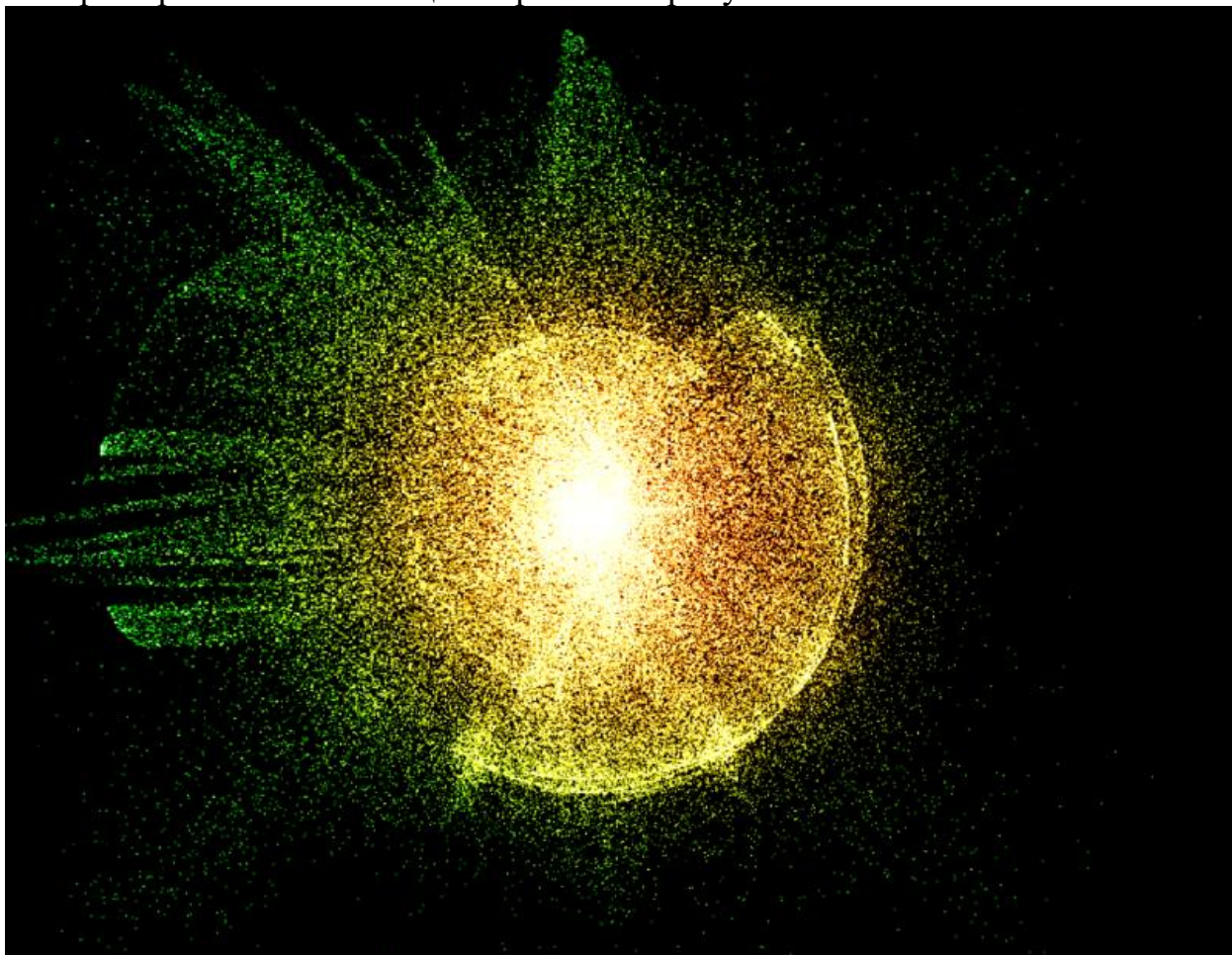


Рисунок 1.3 Система частиц

### ***1.4 Анимирование 3D объектов***

Под анимированием понимается «оживление» 3D моделей, добавление им возможности перемещения в пространстве.

Анимация — это визуальное отображение изменений свойств одного объекта или набора объектов. Анимация сделает ваши модели более живыми, благодаря сохранению действий, которые могут повторяться по вашему выбору. Она помогает визуализировать изменения в перспективе, изменения в свойствах документа и географическую динамику. Используйте анимацию для понимания закономерностей, проявляющихся в данных с течением времени, и для автоматизации процессов, которые должны быть произведены при показе точек, создаваемых только посредством наглядной динамики [5].

Пример создания 3D анимации изображен на рисунке 1.4.

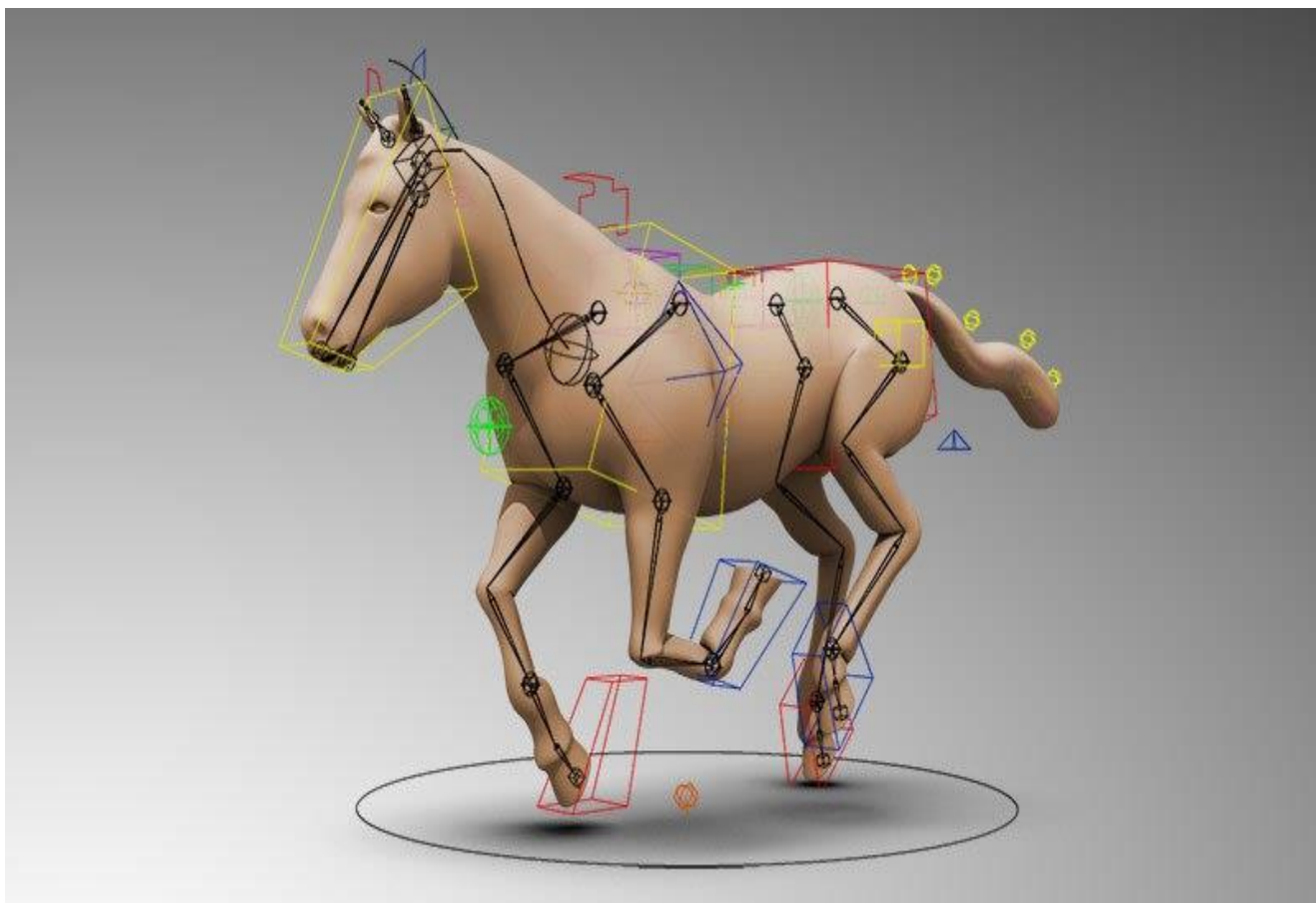


Рисунок 1.4 Пример анимирования 3D модели

### ***1.5 Создание источников освещения***

Компьютерная графика подразумевает возможность создания различных типов источников света. Рассмотрим некоторые из них:

- прямые солнечные лучи. Используются для интерьеров помещений, используемых в дневное время, а также для придания объема экстерьеру или макетам зданий;
- дневное естественное освещение без прямых лучей солнца. Хорошо проявляют себя при использовании для помещений с большими окнами, в то время, как маленькое окно даст недостаточно света, и часть комнаты останется в тени;
- точечный источник света. Используется для визуализации фонарного освещения;
- прожектор. Используется для визуализации освещения в форме конуса. Пример работы источника света изображен на рисунке 1.5.



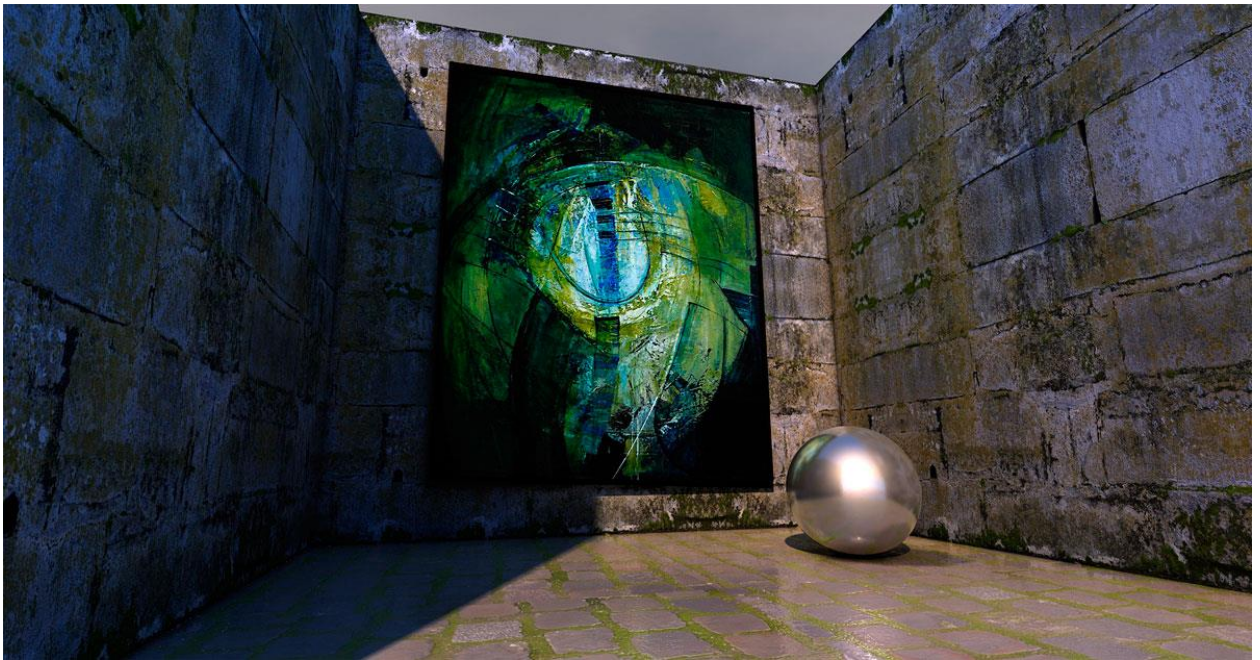


Рисунок 1.5 Пример компьютерного освещения

Процесс 3D моделирования можно разделить на два уровня: микро и макро.

Под микроуровнем мы будем рассматривать моделирование объектов из элементарных частиц – вокселей.

Макроуровень в свою очередь – моделирование композиций объектов в системы внутри трёхмерного пространства, другими словами – визуальное моделирование систем.

Рассмотрим подробнее каждый уровень.

### ***1.6 Моделирование объектов***

Воксель — это элемент объёмного изображения, содержащего значение элемента в трёхмерном пространстве. В качестве значения вокселя в общем случае может выступать что угодно, включая цвет. Что касается формы вокселя, то в общем случае воксели могут быть кубическими, или представлять собой параллелепипед. Координат воксели не хранят, они вычисляются из относительного расположения вокселя.

По сути, воксель является полным аналогом пикселя в 3D. Pixel — элемент изображения, Voxel — элемент объёма. Практически все характеристики пикселя переносятся на воксель, поэтому можно смело проводить аналогии, учитывая размерность. Воксели в свою очередь объединяются в полигоны. В конечном итоге моделирование объектов сводится к полигональному моделированию. Суть его в том, что поверхности представляются в виде простых геометрических двумерных примитивов. В качестве примитивов используют четырехугольники и фигуры с большим кол-вом углов. Эти примитивы, из которых состоит модель, называют полигонами. Но при создании 3D объекта стараются обойтись, как правило,

четырёхугольниками. При необходимости полигоны без проблем превращаются в треугольники при экспорте в систему для композиции объектов [7]. При рассмотрении полигонального моделирования имеет смысл обратить внимание на понятия «сглаживание» и «тесселяция». Если какой-то объект представляется в виде полигонов, то очевидно, что чем меньше размер полигонов, чем их больше, тем более близкой может быть модель к оригиналу. На этом основан метод тесселяции: сначала изготавливают прототип из небольшого кол-ва полигонов, затем применяют операцию тесселяции, при этом каждый полигон делится на 4 части. Таким образом, если полигон четырехугольный, то алгоритмы тесселяции дают более качественный и предсказуемый результат. Также операция сглаживания, схожая с тесселяцией, только с изменением углов на более тупые, при близких к квадрату полигонах, позволяет получить хороший результат.

### ***1.7 Моделирование систем***

Если моделирование 3D объектов подобно лепке скульптуры, то моделирование системы 3D объектов представляет собой последовательность действий:

- размещение 3D объекта в трёхмерном пространстве;
- трансформации 3D объекта;
- конфигурирование параметров 3D объектов;
- связывание 3D объекта с уже существующими в пространстве.

Данных операций хватает для формирования модели системы неподвижных объектов, однако при наличии анимации или поведения объектов, возникает необходимость в дополнительном конфигурировании, заложенной в объекте поведенческой логики.

Таким образом, систему 3D объектов можно описать следующим выражением:

$$S = \langle L, R, Z, M, C, n \rangle, (1)$$

где  $S$  – рассматриваемая система

$L$  – множество координат объектов

$R$  – множество радиан углов поворота объектов, по отношению к нормали пространства

$Z$  – множество множителей растяжения объектов

$M$  – множество материалов объектов

$C$  – множество связей

$n$  – нормаль пространства

### ***Выводы по разделу***

В первом разделе разделе был проведен теоретический анализ основных аспектов визуального моделирования. По результатам анализа можно сделать вывод, что основная концепция ВМ заключается в формировании визуального представления какого-либо объектов посредством средств визуализации.

Наиболее эффективным средством визуализации можно считать инструменты компьютерного визуального моделирования. К ним относятся:

- средства 2D и 3D моделирования;
- средства моделирования освещения и систем частиц;
- средства анимации объектов;

## Раздел 2. Концептуальная модель системы визуального моделирования интерьеров

При рассмотрении визуализации можно выделить два вида формирования визуальной модели, отличающиеся между собой источником данных об объекте моделирования.

Рассмотрим концептуальную модель [6] первого вида – визуальное представление на основе реального объекта. Выделим основные аспекты визуальной составляющей любого объекта или явления реального мира и опишем их в форме кортежей.

$$O = \langle F, M, I, P \rangle, \quad (2)$$

где  $O$  – визуальная сущность объекта,

$F$  – форма объекта,

$M$  – материал объекта,

$I$  – влияние объекта,

$P$  – восприятие объекта.

В свою очередь форма, материал, влияние и восприятия можно декомпозировать следующим образом:

$$F = \langle F_0, F(t) \rangle, \quad (3)$$

где  $F_0$  – начальное состояние формы объекта,  $F(t)$  – закон изменения формы с течением времени.

$$\text{В частном случае } F(t) = F(0) = F_0. \quad (4)$$

Пример статичной формы – стол. Пример динамичной – пар.

$$M = \langle CL, ME, SP, RS, EM, OP, NM, RF \rangle, \quad (5)$$

где каждый компонент кортежа является функцией вида

$$F = F(v, x, y, z, t) \quad (6).$$

В свою очередь  $v$  – значение компонента:

- для  $CL$  – значение цвета;
- для  $ME$  – значение коэффициента блеска материала;
- для  $SP$  – значение коэффициента способности отражения света;
- для  $RS$  – значение коэффициента шероховатости;
- для  $EM$  – значение цвета излучаемого материалом света;
- для  $OP$  – значение коэффициента прозрачности материала;
- для  $NM$  – значение коэффициента поглощения света;
- для  $RF$  – значение коэффициента зеркальности.

Параметры  $x, y, z$  отвечают за координаты точки материала на поверхности формы объекта.  $t$  – значение времени.

$$I = \langle FI, MI \rangle, \quad (7)$$

где  $FI$  – влияние объекта на форму других объектов,

$MI$  – влияние объекта на материал других объектов.

$$P = \langle FP, MP \rangle, (8)$$

где  $FP$  – реакция формы объекта на влияние других объектов,  $MP$  – реакция материала объекта на влияние других объектов.

Стоит заметить, что представленный выше уровень декомпозиции не является финальным, так как при необходимости, каждый компонент можно декомпозировать [8] на более мелкие составляющие.

Схематично концептуальная модель выглядит следующим образом:

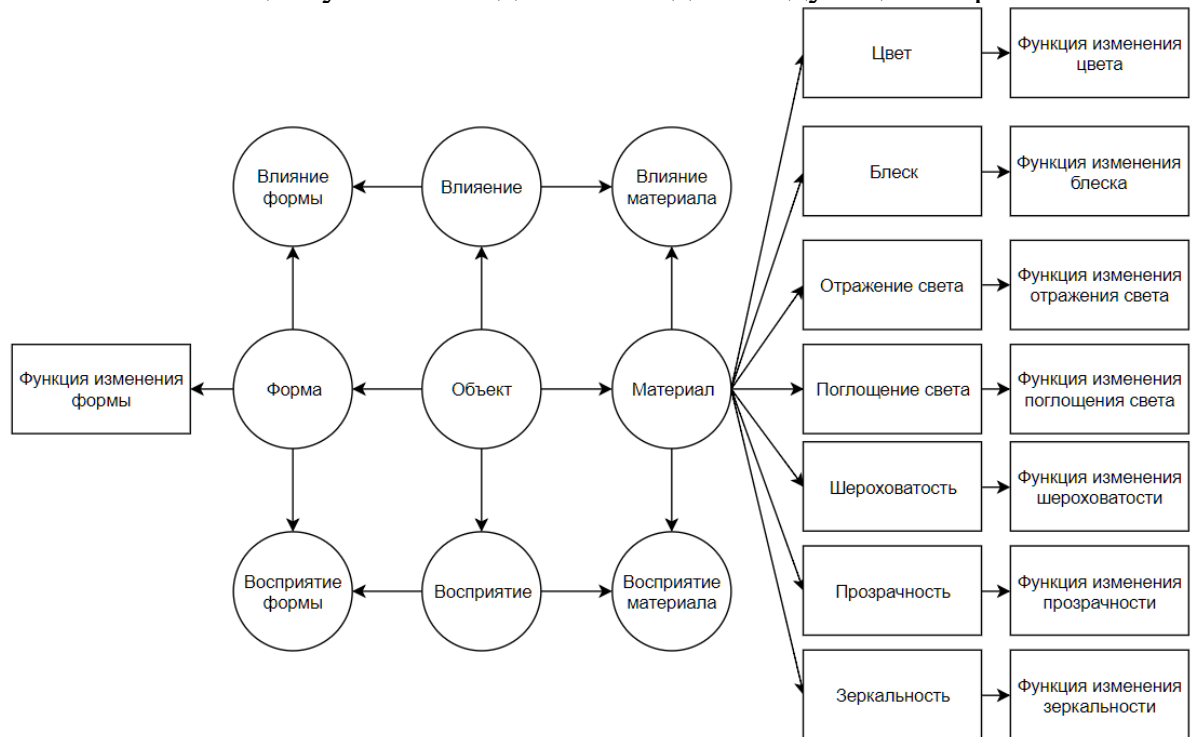


Рисунок 2.1 Концептуальная модель визуального представления реального объекта

Далее рассмотрим концептуальную модель визуализации второго вида – визуализации нереальных объектов, то есть объектов, у которых не имеется реального аналога, они являются продуктами человеческого разума.

Любой продукт разума можно назвать идеей. Таким образом, визуальная модель нереального объекта является по сути визуализацией идеи.

Формула такой концептуальной модели выглядит следующим образом:

$$VC = R(C), (9)$$

где  $VC$  – визуализация концепта (идеи), а  $R(C)$  – функция разума над концептом, определяемая следующим образом:

$$R(C) \rightarrow O, (10)$$

где  $O$  – множество визуальных представлений возможного реального объекта, определяемое разумом.

На практике визуальное моделирование идеальных объектов не отличается от моделирования реальных объектов. Разница состоит лишь в источнике моделирования.

## 2.1 Описание атрибутов и концептуальная схема данных объектов

Далее будет проведен анализ объектов интерьера и построение концептуальной модели сущности данных объектов [9] с целью последующего создания модуля для их визуализации.

С точки зрения атрибутов, объекты интерьера можно разделить на следующие группы:

- источник освещения;
- объект мультимедиа;
- переключатель;
- объект каркаса.

Источником освещения будем называть любой объект, предназначенный для освещения помещения: люстра, торшер, бра и т.д.

Объектом мультимедиа является любой объект, требующий взаимодействия с источником электричества, но при этом не служащий для освещения. Такими объектами являются телевизор, компьютер, звуковые колонки и др.

Переключатель – объект для выключения и включения источников освещения.

Объект каркаса – объекты, которые нельзя трансформировать, они являются частью дома, например, стена, пол, потолок.

Схематично иерархия типов интерьеров выглядит следующим образом:

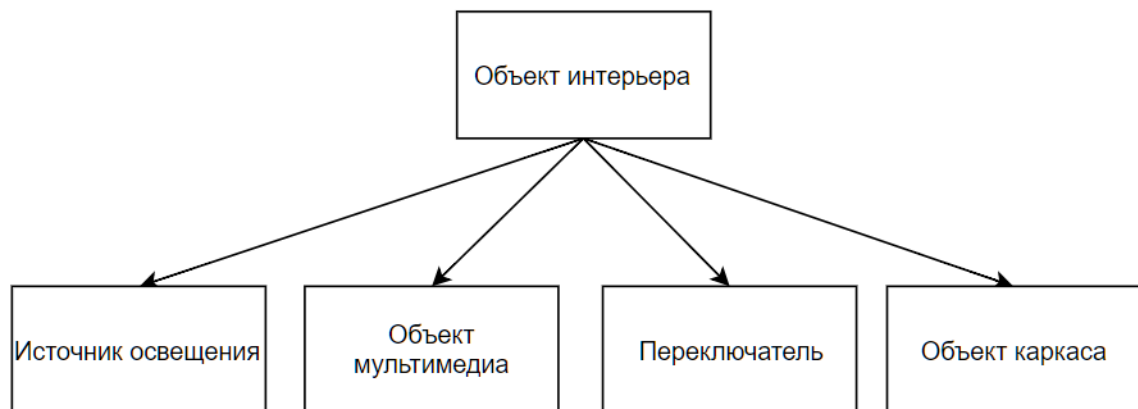


Рисунок 2.2 Типы объектов интерьера.

Опишем атрибуты объектов в соответствии с их типом.

Каждый тип объектов интерьера имеет следующие атрибуты:

- имя объекта;
- имя группы объекта;
- доступные режимы перемещения;
- доступные режимы вращения;
- доступные режимы растяжения;
- тип объекта;



- доступные действия над объектом;
- цвет для визуальной обводки объекта;
- условие прикрепления объекта к поверхности;
- доступные поверхности объекта.

Помимо общих параметров, у некоторых типов объекта имеются уникальные атрибуты.

Источник освещения имеет следующий набор атрибутов:

- компонент освещения;
- состояние (включен/выключен).

У объекта мультимедиа имеются следующие атрибуты:

- компонент мультимедиа.
- состояние (включен/выключен);

Атрибуты переключателя:

- список управляемых источников освещения;

Рассмотрим подробно каждый тип объектов интерьера и опишем концептуальную схему их атрибутов.

Описание общих атрибутов объекта интерьера представлено ниже:

Таблица 2.1 Атрибуты объекта интерьера

Наименование атрибута	Тип атрибута	Описание атрибута	Пример значения атрибута
Имя объекта	Строка	Уникальное наименование объекта	Стол для работы
Имя группы объекта	Строка	Уникальное наименование группы, к которой объект принадлежит	Стол
Доступные режимы перемещения	Перечисление строк: <ul style="list-style-type: none"> <li>– по поверхности;</li> <li>– свободное;</li> <li>– отсутствует.</li> </ul>	Определяет, какие допускаются направления для перемещения объектов	Свободное

Доступные режимы вращения	Перечисление строк: – по поверхности; – свободное; – отсутствует.	Определяет, какие допускаются направления для вращения объектов	Свободное
Доступные режимы растяжения	Перечисление строк: – по поверхности; – свободное; – отсутствует.	Определяет, какие допускаются направления для растяжения объектов	Свободное
Тип объекта	Перечисление строк: – простой объект; – источник освещения; – объект мультимедиа; – переключатель; – объект каркаса.	Указывает на принадлежность объекта к определенному типу	Простой объект
Доступные действия над объектом	Список из перечислений строк: – изменить; – заменить; – переместить; – повернуть; – растянуть; – добавить лампу; – удалить лампу; – отмена.	Список, определяющий возможные операции над объектом	изменить
Цвет для визуальной обводки объекта	Строка	Цвет, отображаемый при выделении объекта в пользовательском интерфейсе	Синий
Условие прикрепления объекта к поверхности	Логическая переменная	Определяет, должен ли объект быть прикреплен к поверхности после создания	ДА

Доступные поверхности объекта	Перечисление строк: – потолок; – пол; – стена; – другой объект;	Определяет, на какие поверхности может быть создан объект	Потолок
Визуальный объект	Ссылка на программный объект	Визуальная составляющая объекта интерьера	Письменный стол Крокер

Описание уникальных атрибутов для источника освещения представлено ниже.

Таблица 2.2 Атрибуты источника освещения.

Наименование атрибута	Тип атрибута	Описание атрибута	Пример значения атрибута
Компонент освещения	Ссылка на программный объект	Визуальный компонент, моделирующий источник света	Точечный источник освещения
Состояние	Перечисление строк: – ВКЛ – ВЫКЛ	Определяет, включен ли источник света или нет	ВКЛ

Описание уникальных атрибутов для объекта мультимедиа представлено ниже.

Таблица 2.3 Атрибуты объекта мультимедиа.

Наименование атрибута	Тип атрибута	Описание атрибута	Пример значения атрибута
Компонент мультимедиа	Ссылка на программный объект	Визуальный компонент, отвечающий за функциональность мультимедиа	Экран телевизора
Состояние	Перечисление строк: – ВКЛ – ВЫКЛ	Определяет, включен ли объект мультимедиа или нет	ВКЛ

Описание уникальных атрибутов для переключателя представлено ниже.

Таблица 2.4 Атрибуты переключателя

Наименование атрибута	Тип атрибута	Описание атрибута	Пример значения атрибута
Список управляемых источников освещения	Список ссылок на программный объект	Определяет, какие источники освещения контролирует переключатель	[люстра на кухне, люстра в гостиной]

Концептуальная схема данных объектов интерьера представлена на рисунке 2.3

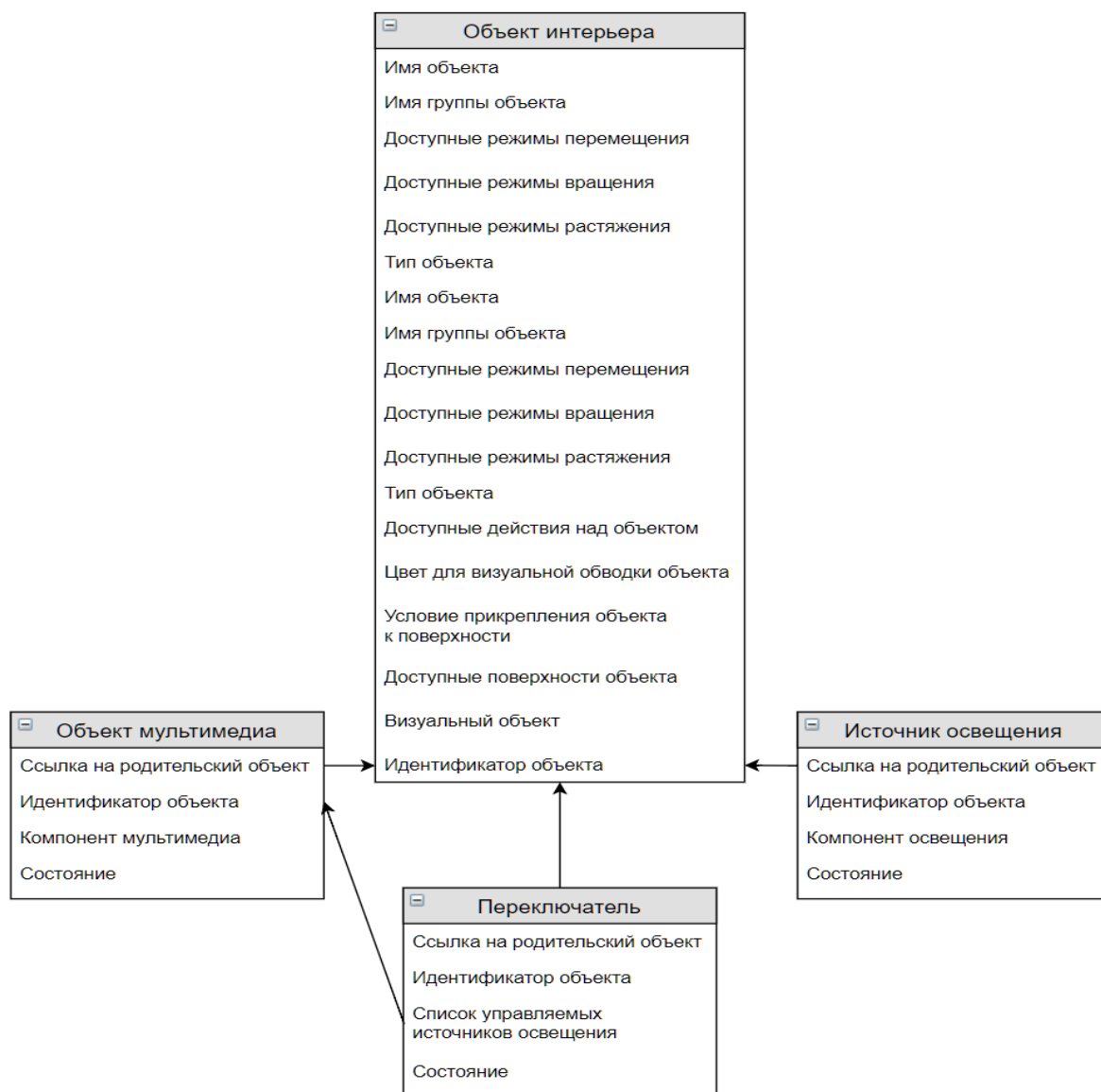


Рисунок 2.3 Концептуальная схема данных объектов интерьера

## ***2.1 Выбор технологий реализации***

Так как система моделирования интерьеров является клиентским графическим приложением, то был проведен анализ следующих наиболее распространённых технологий реализации:

- графический движок CryEngine 3;
- графический движок Unity;
- графический движок Unreal Engine 4;

По результатам анализа наиболее оптимальным и подходящим был выбран графический движок Unreal Engine 4. Основные его плюсы по сравнению с другими технологиями:

- возможность выполнять гибкую разработку на визуальном языке программирования Blueprint;
- возможность выполнять гибкую настройку и конфигурирование материалов и шейдеров с помощью Material Editor;
- удобный и высокотехнологичный редактор трёхмерных пространств;
- простая и гибкая программная модель;
- удобный и качественный редактор пользовательского интерфейса.

## ***2.2 Архитектура программной системы***

В зависимости от сборки система может быть представлена в двух вариантах [10]. В первом случае она – модуль движка UE4, во втором – исполняемое приложение. Концептуальное отличие вариантов состоит в том, что в случае модуля движок присоединяет этот модуль к себе и управляет им извне, а в случае приложения система подключает модули движка и взаимодействует с ними, но при этом жизненный цикл приложения извне никем не управляется [11]. Архитектура системы представлена на рисунке 2.4.

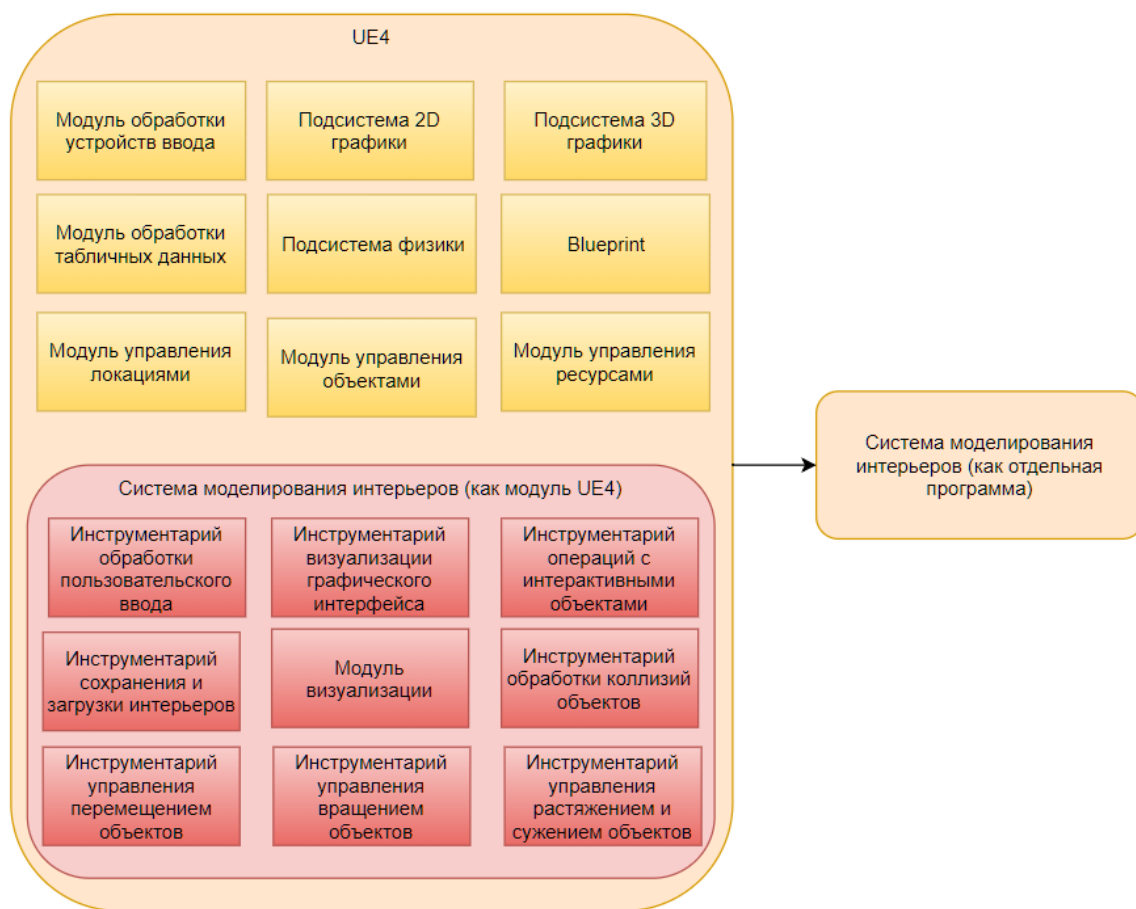


Рисунок 2.4 Общая архитектура системы

### ***Выводы по разделу***

Во втором разделе была рассмотрена концептуальная модель данных объектов интерьера и подробно описан атрибутивный набор сущностей.

Также, была сформулирована общая концептуальная модель визуализации с описанием основных взаимосвязей компонентов визуальной модели и представлена архитектурная модель рассматриваемой программной системы.

По результатам анализа для реализации программной системы наиболее целесообразной и необходимой технологией реализации был выбран графический движок UE4.

### **Раздел 3. Реализация модуля визуализации**

С технической точки зрения ядро реализуемой системы заключено в модуле визуализации. Основная функция модуля состоит в передачи команд от других инструментов и модулей в графический движок. Принцип работы движка состоит в том, чтобы на каждом программном «кадре» сканировать конвейер программных инструкций для графического процесса и выполнять алгоритм отрисовки визуальных объектов. Модуль визуализации на основе данных от других компонентов формирует набор команд для конвейера и передаёт их в графический поток.

В системе можно выделить следующий набор алгоритмов:

- алгоритм сохранения интерьера;
- алгоритм загрузки интерьера;
- алгоритм создания объекта;
- алгоритм замены объекта;
- алгоритм изменения материала объекта;
- алгоритм перемещения объекта;
- алгоритм вращения объекта;
- алгоритм растяжения и сужения объекта;
- алгоритм удаления объекта;

Рассмотрим каждый алгоритм подробнее.

#### ***3.1 Алгоритм сохранения интерьера***

Данный алгоритм реализует сохранение всех объектов созданного интерьера, а также координаты пользователя в файл. Шаги алгоритма:

- 1) создать объект сохраненной сцены интерьера;
- 2) записать в созданный объект текущие параметры пользователя;
- 3) записать каждый объект интерьера в созданный объект сохраненной сцены;
- 4) произвести сериализацию объекта сохраненной сцены.

#### ***3.2 Алгоритм загрузки интерьера***

Данный алгоритм реализует загрузку всех объектов созданного интерьера, а также координат пользователя из файла в систему. Шаги алгоритма:

- 1) создать объект пустой сцены;
- 2) переместить пользователя в объект пустой сцены;
- 3) принудительно сменить режим на «редактор»;
- 4) произвести десериализацию объекта сохраненной сцены;
- 5) создать каждый сохраненный объект;
- 6) изменить параметры игрока на сохраненные.

### ***3.3 Алгоритм создания объекта***

Данный алгоритм реализует создание и инициализацию всех объектов интерьера. Шаги алгоритма:

- 1) по имени объекта получить данные об объекте из базы данных;
- 2) задать параметры создания объекта для использования библиотечной функции;
- 3) выполнить создание объекта средствами UE4 [12];
- 4) произвести инициализацию объекта;
- 5) для лампы установить относительное положение источника света;
- 6) для TV установить медиа-материал;
- 7) для всех объектов задать трёхмерную модель, проверив при этом коллизию, а также задать служебные данные.

### ***3.4 Алгоритм замены объекта***

Данный алгоритм реализует удаление объекта интерьера с последующим созданием нового. Шаги алгоритма:

- 1) временно сохранить параметры текущего объекта;
- 2) удалить объект сцены;
- 3) вычислить параметры создания нового объекта на базе удалённого;
- 4) создать объект;
- 5) очистить кеш параметров.

### ***3.5 Алгоритм изменения материала объекта***

Данный алгоритм реализует изменение материала определенной части объекта интерьера. Шаги алгоритма:

- 1) получить данные о доступных материалах объекта интерьера;
- 2) получить доступ к геометрической модели объекта;
- 3) произвести замену части материала геометрической модели объекта.

### ***3.6 Алгоритм перемещения объекта***

Данный алгоритм реализует изменение координат объекта интерьера. Началом алгоритма считается событие нажатия левой кнопки на объект при активном режиме перемещения, однако проверка нажатия осуществляется на каждом кадре с момента запуска системы. Шаги алгоритма:

- 1) выполнить трассировку и проецирование луча нажатия курсора на поверхность объекта;
- 2) если у объекта тип поверхности – плоскость, то приподнять объект над поверхностью;
- 3) переместить объект в новую координату;
- 4) проверить коллизию объекта;



- 5) если объект пересекается с другим объектом, то переместить его в границу пересечения;
- 6) если объект был приподнят на шаге 2, то опустить его.

### ***3.7 Алгоритм вращения объекта***

Данный алгоритм реализует изменение угла поворота относительно выбранных осей объекта интерьера. Началом алгоритма считается событие нажатия левой кнопки на объект при активном режиме вращения, однако проверка нажатия осуществляется на каждом кадре с момента запуска системы. Шаги алгоритма:

- 1) вычислить дельту вращения;
- 2) если у объекта тип поверхности – плоскость, то приподнять объект над поверхностью;
- 3) повернуть объект вычисленную величину дельты вращения
- 4) проверить коллизию объекта;
- 5) если объект пересекается с другим объектом, то повернуть его в до границы;
- 6) если объект был приподнят на шаге 2, то опустить его.

### ***3.8 Алгоритм растяжения и сужения объекта***

Данный алгоритм реализует растяжение и сужение относительно выбранных осей объекта интерьера. Началом алгоритма считается событие нажатия левой кнопки на объект при активном режиме изменении размера, однако проверка нажатия осуществляется на каждом кадре с момента запуска системы. Шаги алгоритма:

- 1) Вычислить дельту изменения размера;
- 2) если у объекта тип поверхности – плоскость, то приподнять объект над поверхностью;
- 3) увеличить или уменьшить объект в соответствии с дельтой;
- 4) проверить коллизию объекта;
- 5) если объект пересекается с другим объектом, то уменьшить его в до границы пересечения;
- 6) если объект был приподнят на шаге 2, то опустить его.

### 3.9 Алгоритм удаления объекта

Данный алгоритм реализует удаление объекта интерьера. Алгоритм состоит из одного шага: вызов метода уничтожения актера UE4, которым является удаляемый объект.

### 3.10 Пользовательский интерфейс программы

Пользовательский интерфейс программы состоит из композитных виджетов, созданных с помощью средств UMG. Список элементов пользовательского интерфейса:

- виджет перемещения, виджет вращения, виджет изменения размера (представлены на рисунке 3.1);
- диалог создания сцены (представлен на рисунке 3.2);
- диалог выбора сцены (представлен на рисунке 3.3);
- список категорий (представлен на рисунке 3.4);
- список групп (представлен на рисунке 3.5);
- меню выбора объектов (представлено на рисунке 3.6);
- меню действий над простым объектом (рисунок 3.7);
- меню действий над выключателем (рисунок 3.8);
- меню действий над стационарным объектом (рисунок 3.9);
- меню изменения материала объекта (рисунок 3.10).



Рисунок 3.1 Виджеты перемещения, повтора и расширения.

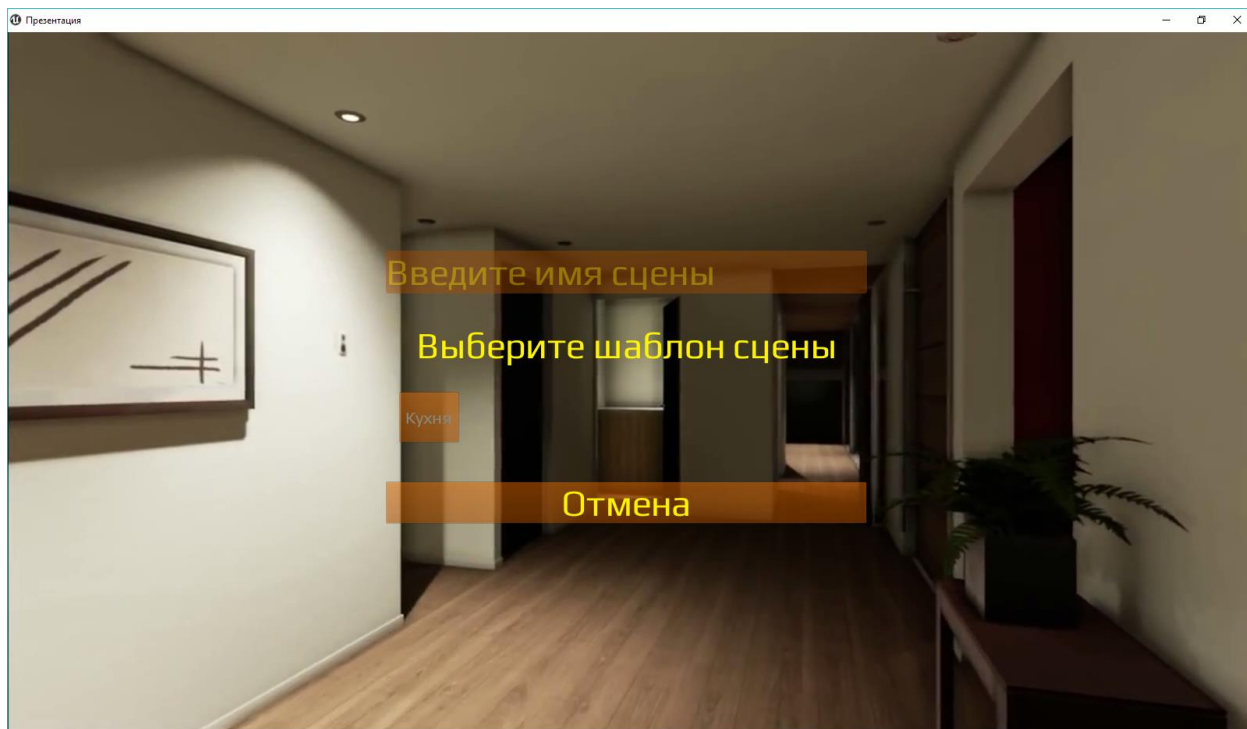


Рисунок 3.2 Диалог создания сцены

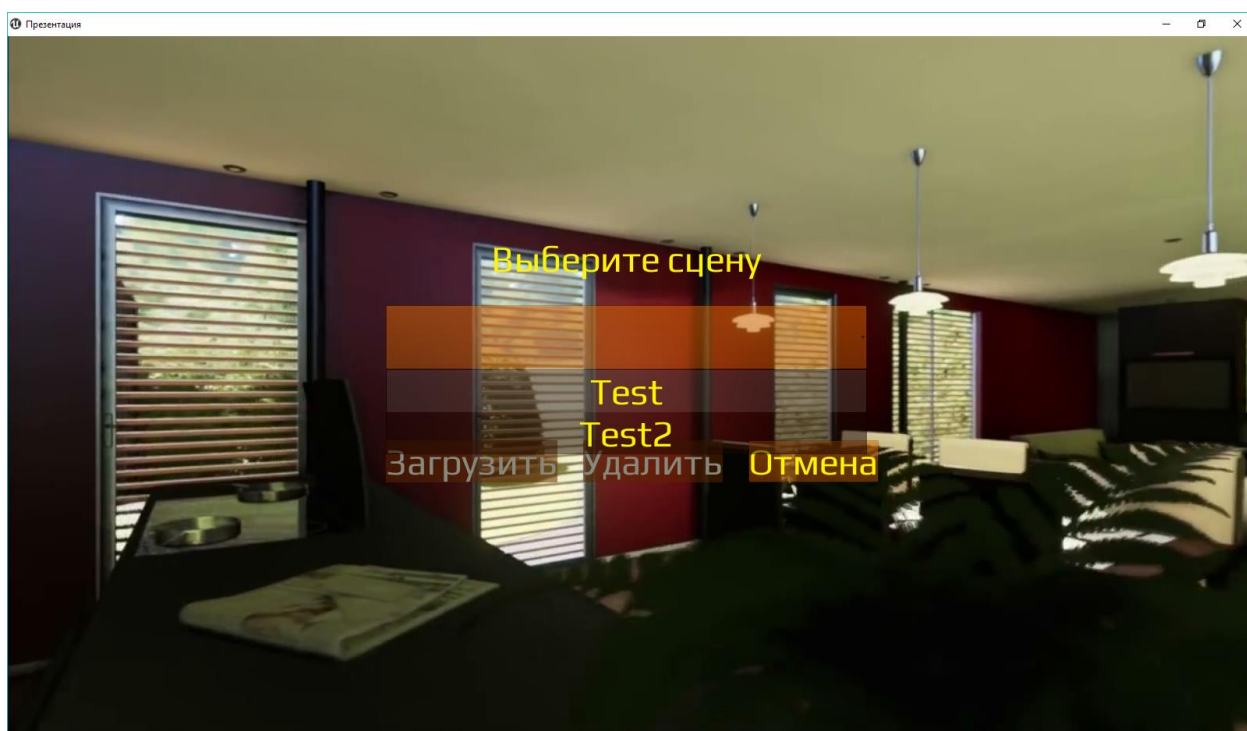


Рисунок 3.3 Диалог выбора сцены



Рисунок 3.4 Список категорий

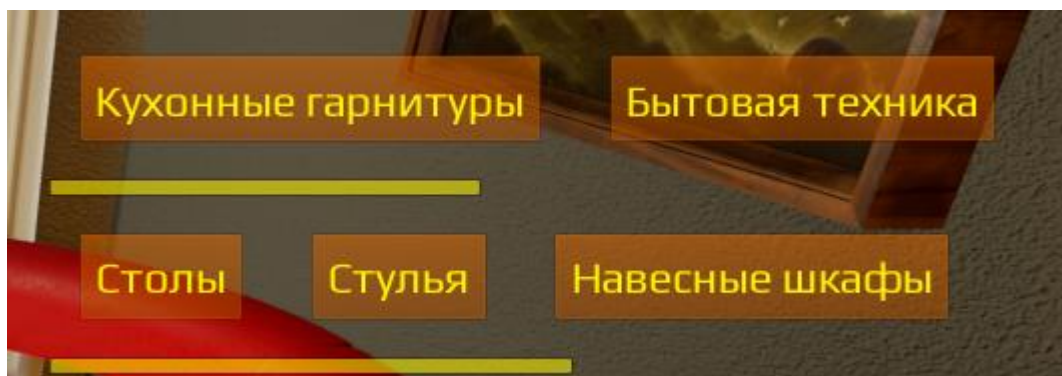


Рисунок 3.5 Список групп

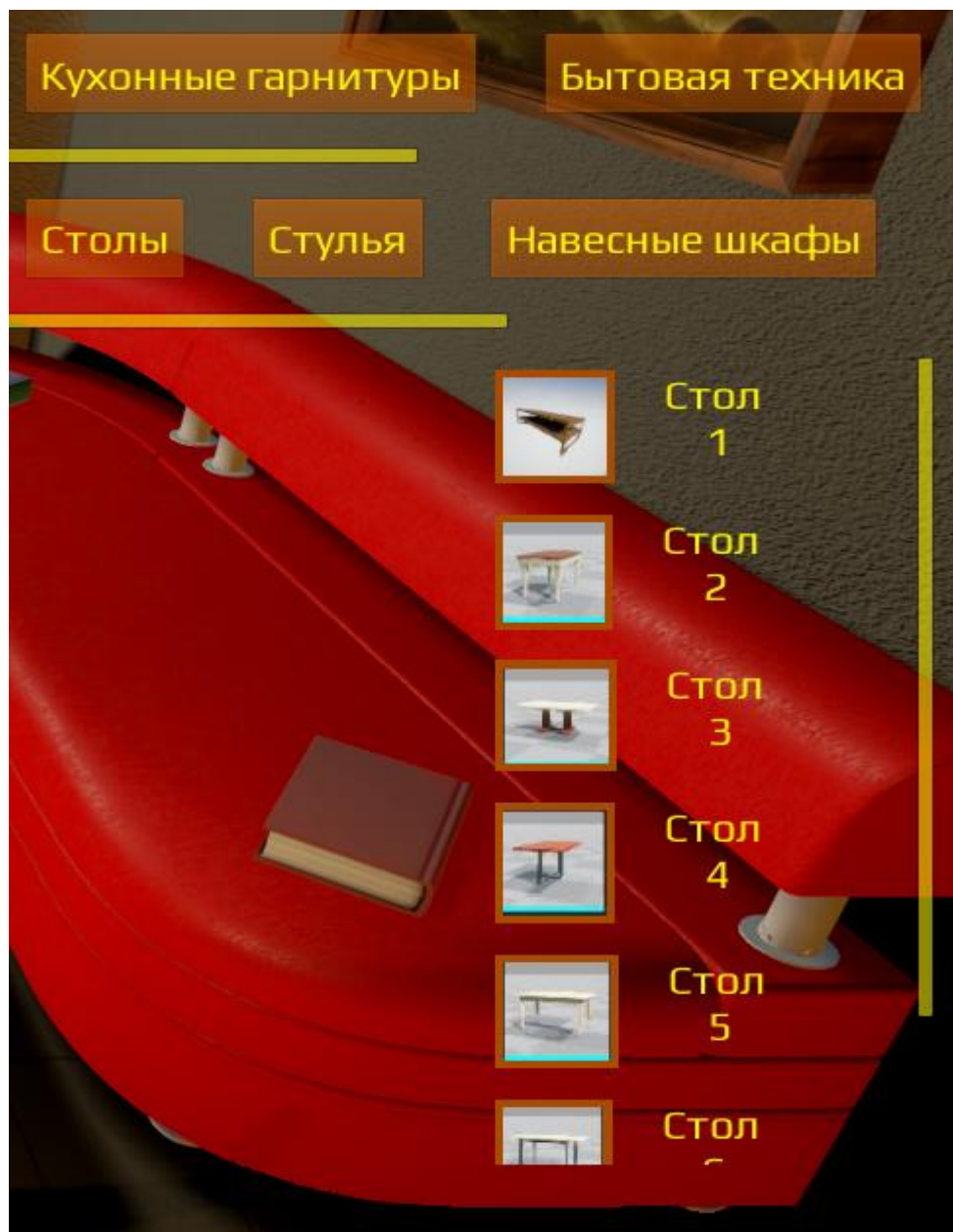


Рисунок 3.6 Меню выбора объектов



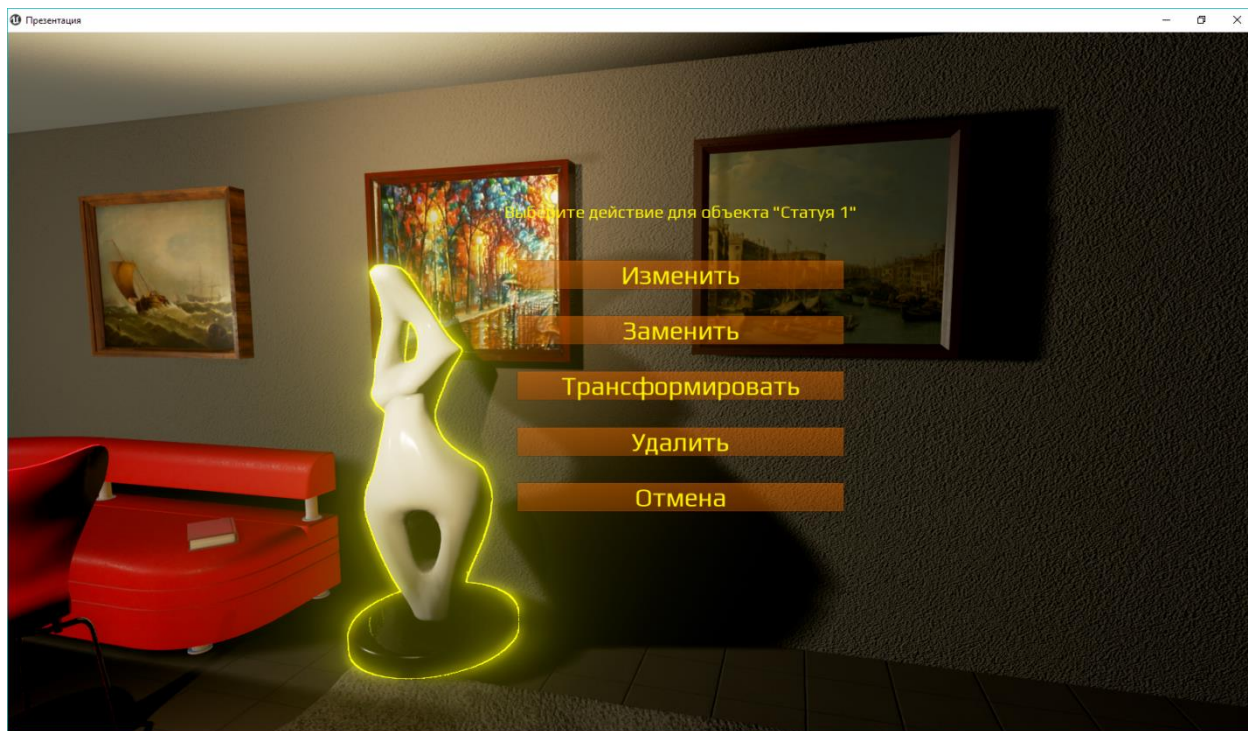


Рисунок 3.7 Меню действий над простым объектом

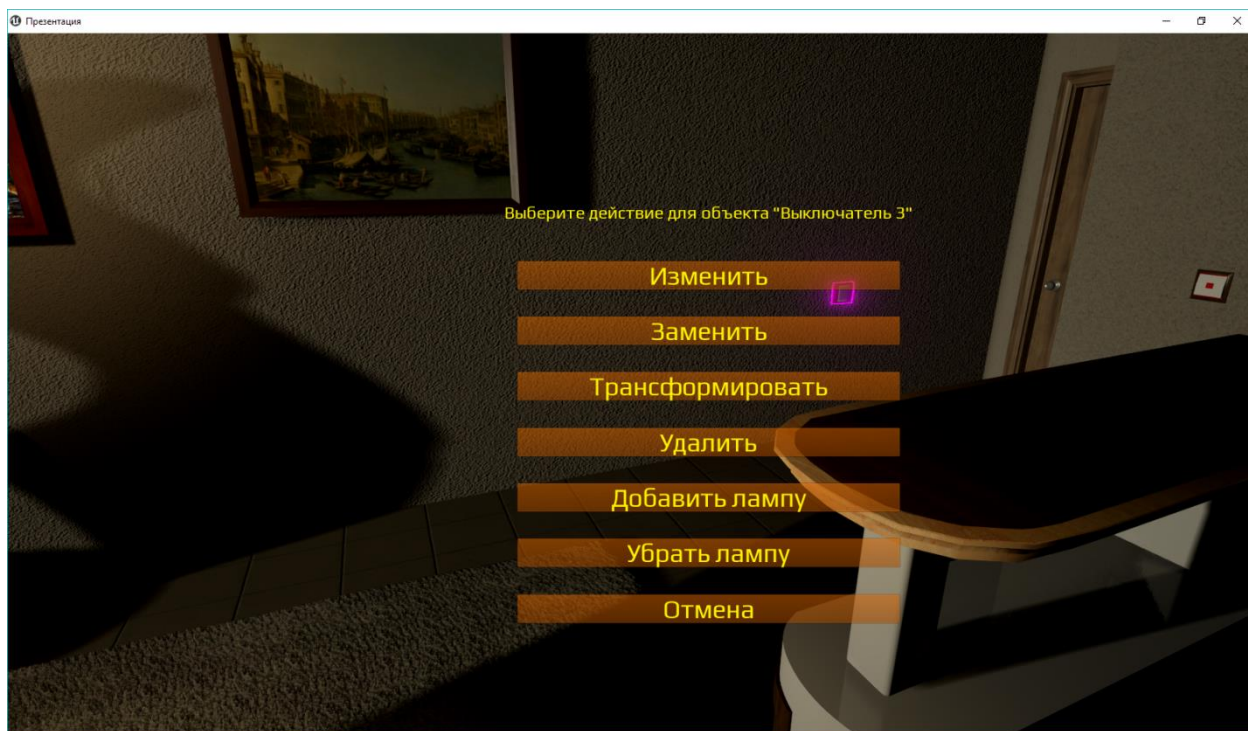


Рисунок 3.8 Меню действий над выключателем

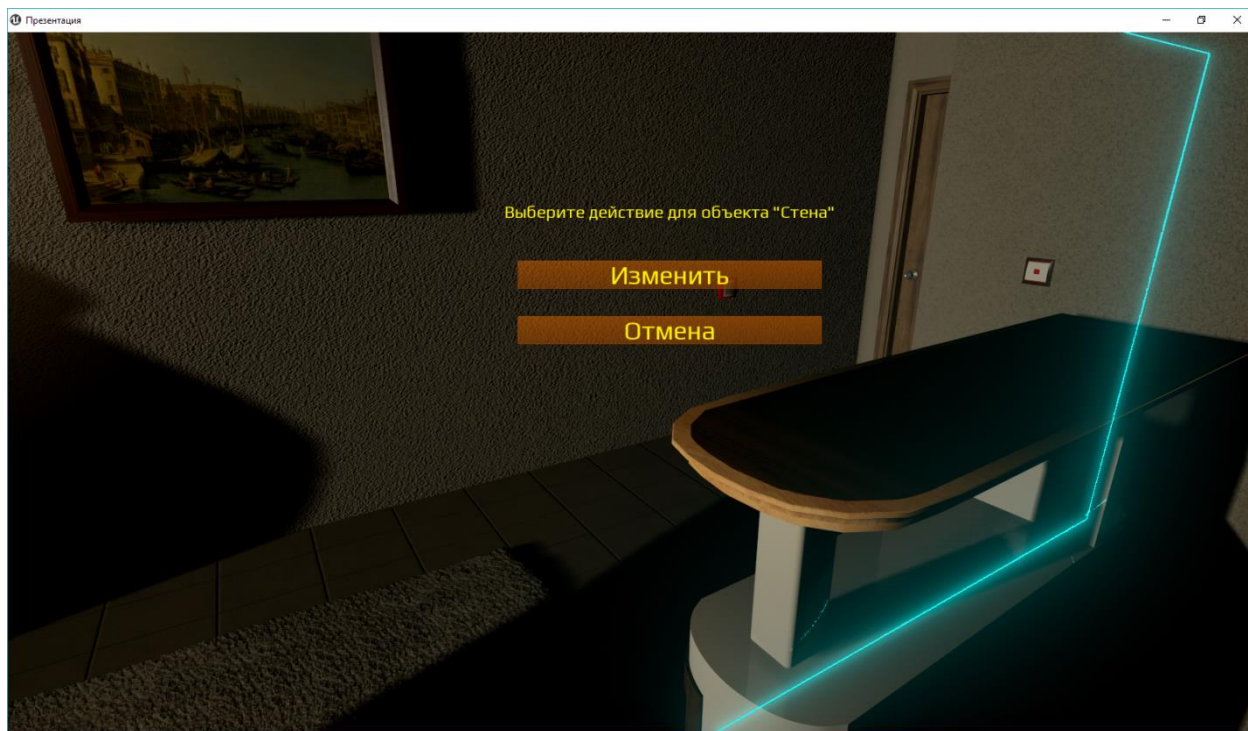


Рисунок 3.9 Меню действий над стационарным объектом



Рисунок 3.10 Меню изменения материала объекта

### 3.11 Тестирование разрабатываемого ПО

Для проверки корректности выполнения всех требований к системе было реализовано множество тестов. Все написанные тесты можно разделить на две категории: юнит тесты и интеграционные тест. Юнит тесты пишутся на функции, или группу функций, решающих одну задачу. Интеграционные тесты пишутся для проверки выполняемых системой сценариев, например, сценарий удаления объекта. Для запуска и мониторинга процесса и результата тестирования использовалась внутренняя система автоматизации тестирования UE4 вместе с Unreal Message Bus.

Юнит тестирование проводится на заранее готовых данных и эмуляторов с целью выявить некорректно работающий код функций. Результаты юнит тестирования представлены в таблице 3.1.

Таблица 3.1 Отчет о результатах Юнит тестирования

Описание теста	Тестируемая функция	Результат
Тестируемый объект должен прикрепиться к поверхности создания	ASimpleSceneObject::SnapToSpawnPlane	Успех. Функция вернула true.
Тестируемый объект должен создаться	AMainGameMode::SpawnSceneObject	Успех. Объект интерьера создан.
Функция должна получить данные из базы	FDatabaseHandler::GetSceneObjectData	Успех. Данные получены
Тестируемая сцена должна быть сохранена	FSceneHandler::SaveScene	Успех. Файл с сохраненной сценой содался

Тестируемая сцена должна загрузиться	FSceneHandler::LoadScene	Успех. Сцена была загружена
Тестируемая сцена должна быть удалена	FSceneHandler::DeleteSavedScene	Успех. Функция вернула true. Файл сцены отсутствует.
Функция должна вернуть массив заготовленных для теста сцен	FSceneHandler::GetSavedScenes	Успех. Массив сцен совпадает с ожидаемым.
Тестируемая сцена должна присутствовать	FSceneHandler::DoesSceneExist	Успех. Функция вернула true. Файл сцены присутствует.
Тестируемая сцена должна отсутствовать	FSceneHandler::DoesSceneExist	Успех. Функция вернула false. Файл сцены отсутствует.



Тестируемый выключатель должен добавить лампу	ASwitch::AddLamp	Успех. Функция вернула true. Указатель на лампу присутствует в массиве переключателя.
Тестируемая лампа должна включиться	ALamp::Interact	Успех. Свет был включен.
Тестируемый телевизор должен включиться	ATV:Interact	Успех. Телевизор был включен
Тестируемый объект должен увеличиться	ASceneObjectBase::ScaleUp	Успех. Параметры после увеличения совпадают с ожидаемыми.
Тестируемый объект должен уменьшиться	ASceneObjectBase::ScaleDown	Успех. Параметры после увеличения совпадают с ожидаемыми.

В качестве интеграционного тестирования был реализован тестовый сценарий, затрагивающий ключевые требования системы. Результаты интеграционного тестирования в таблице 3.2

Таблица 3.2 Отчёт о проведении интеграционного тестирования

<b>№ шага</b>	<b>Описание шага</b>	<b>Реальный результат шага</b>	<b>Ожидаемый результат шага</b>	<b>Результаты идентичны</b>
1	Создание новой сцены	Сцена создана	Сцена создана	Да
2	Создание объекта интерьера	Объект создан	Объект создан	Да
3	Изменение материала объекта	Материал изменился на новый	Материал изменился на новый	Да
4	Перемещение объекта	Координаты объекта стали новыми	Координаты объекта стали новыми	Да
5	Поворот объекта	Углы поворота относительно осей обновились	Углы поворота относительно осей обновились	Да
6	Увеличение объекта	Размер объекта увеличился	Размер объекта увеличился	Да
7	Уменьшение объекта	Размер объекта уменьшился	Размер объекта уменьшился	Да

8	Повтор шагов 5,6,7 с эмулированием коллизии	Объект имеет характеристики в соответствии с допустимой границей коллизии	Объект имеет характеристики в соответствии с допустимой границей коллизии	Да
9	Сохранение сцены	Сцена сохранена	Сцена сохранена	Да
10	Перезапуск системы и загрузка сцены	Сцена загружена	Сцена загружена	Да
11	Удаление объекта	Объект удален	Объект удален	Да

### ***Выводы по разделу***

В третьем разделе было представлено описание реализации модуля визуализации и основного набора исполняемых алгоритмов. В качестве ядра графической подсистемы для реализации механизмов визуализации был использован графический движок Unreal Engine 4 [13]. Особенность реализованной системы заключается в том, что она одновременно является и самостоятельным приложением, и в тоже время – модулем движка. Подобную особенность системы позволила достичь уникальная модель сборки программных модулей движка UE4 [14].

Также были продемонстрированы скриншоты основных экранных форм и предоставлены результаты тестирования системы.

## **Раздел 4. Экономическая часть**

### ***4.1 Анализ рынка сбыта продукта***

Данный продукт в первую очередь будет интересен компаниям, производящим мебель и предметы для дома.

Основная аудитория программы – дизайнеры интерьера, однако данная система может использоваться и в других целях, например, для визуализации произвольного рода помещений. Программа может быть сконфигурирована и оптимизирована под нужды конечного пользователя, в следствие чего рынок сбыта продукта может быть расширен путём небольших модификаций базы данных или конфигурации программы.

### ***4.2 План производства***

Разработка программного продукта производится в компании заказчика – производителя мебельных товаров с использованием новой техники, пригодной для разработки данного проекта.

Для разработки проекта необходимо привлечение следующих людских ресурсов:

- начальник отдела. Данный человек является руководителем всей разработки в целом. Отвечает за организационные процессы отдела;
- технический лидер. Данный человек является руководителем отдела программистов. Отвечает за разработку программной части проекта;
- начальник дизайн-отдела. Данный человек является руководителем отдела дизайнеров. Разрабатывает общий дизайн проекта, основываясь на потребительских вкусах;
- программист графического интерфейса. Данный человек реализовывает графический дизайн проекта, сторону программы, с которой будет работать сам пользователь;
- программист баз данных. Отвечает за реализацию баз данных в проекте.
- логический программист. Разрабатывает и реализовывает внутреннюю программную логику;
- дизайнер 2D графики. Реализовывает дизайн 2D элементов, а также дизайн пользовательского интерфейса;
- дизайнер 3D графики. Данный человек разрабатывает и реализовывает 3D модели, используемые в данном проекте;
- дизайнер материалов. Разрабатывает и реализовывает дизайн материалов для 3D графики.

Структура организационного взаимодействия сотрудников представлена на рисунке 4.1



Рисунок 4.1. Структура организационного взаимодействия сотрудников

### 4.3 Организационный план

При проведении работ по созданию системы выделяются следующие этапы, сведенные в таблицу 4.1

Таблица 4.1. Этапы разработки.

№ этап а	Наименование этапа	Исполнитель	Трудоемкость в чел/день	Продолжительность этапа в днях
1	Анализ требований и составление технического задания	Начальник отдела	1	1
2	Создание технического проекта	Начальник отдела	1	1
		Технический лидер	1	
		Начальник дизайн отдела	1	
3	Создание эскиза проекта	Начальник дизайн отдела	7	7
4	Постановка задачи	Начальник дизайн отдела	1	1
		Технический лидер	1	

5	Планирование сроков реализации	Технический лидер	1	1
		Начальник дизайн-отдела	1	
		Начальник отдела	1	
6	Реализация ресурсов	Дизайнер 2D графики	10	15
		Дизайнер 3D графики	15	
		Дизайнер материалов	10	
7	Оценка приёмка ресурсов	и Начальник отдела	1	1
		Начальник дизайн-отдела	1	
8	Кодирование	Программист графического интерфейса	10	20
		Программист баз данных	20	
		Логический программист	10	
9	Тестирование	Программист графического интерфейса	2	5
		Программист баз данных	5	
		Логический программист	3	
10	Оценка приёмка кода	и Технический лидер	1	1
		Начальник отдела	1	
11	Завершающая проверка и релиз системы	Технический лидер	5	5
		Начальник отдела	2	
		Начальник дизайн-отдела	3	
Итого			115	48

На рисунке 4.2 изображена диаграмма Ганта по этапам разработки проекта.

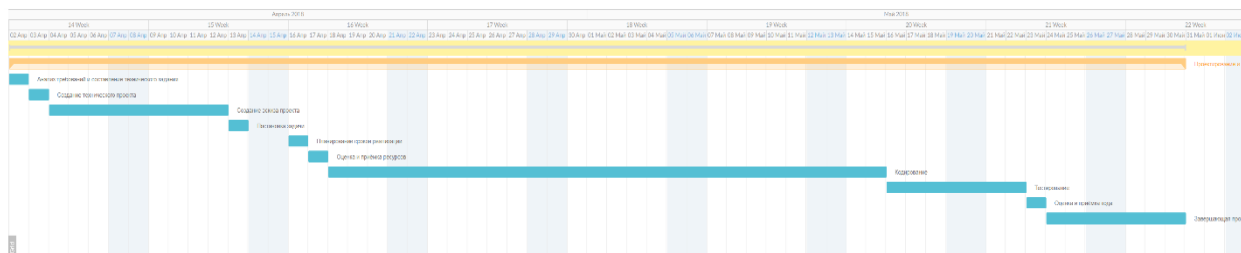


Рисунок 4.2 Диаграмма Ганта

#### 4.4 Расчет затрат и договорной цены

Все расходы представляют собой оправданные экономические затраты, оценка которых выражается в денежной форме с помощью метода калькуляционных затрат. В конечную стоимость проекта включаются все затраты, связанные с его выполнением, независимо от источника финансирования.

Калькуляция стоимости проведения работ составляется по следующим статьям затрат:

- материалы, покупные изделия;
- техническое оборудование;
- заработная плата персонала;
- страховые взносы;
- накладные расходы;

В результате подсчета данных статей производится расчет полной себестоимости разработки. Определяется плановая прибыль проекта и его оптовая цена разработки, учитывая НДС (18%) [15].

#### 4.5 Материалы и покупные изделия

К этой статье относятся расходы на материалы, используемые непосредственно в процессе выполнения проекта. Также к этой статье относят и транспортно-заготовительные расходы. Расчет затрат на материалы и покупные изделия приведен в таблице 4.2

Таблица 4.2. Затраты по первой статье

№	Наименование	Единица измерения	Количество	Цена руб.	Стоимость руб.
1.	Бумага	Упаковка	10	130	1300
2.	Канцелярия	Набор	10	115	1150
3.	Программное обеспечение	Пакет	1	3500	3500
Итого:					5950

Затратами по данной статье является сумма всех вышеизложенных затрат. Транспортно-заготовительные расходы [16] составляют 15% от стоимости материалов:

$$5950 \cdot 0,15 = 892,5 \text{ (руб.)}$$

Суммарные расходы по 1 статье составляют  $S_1 = 5950 + 892,5 = 6842,5$  (руб.).

#### **4.6 Техническое оборудование**

Техническое оборудование – оборудование, с помощью которого будет реализован проект. Расчет затрат на техническое оборудование представлен в таблице 4.3

Таблица 4.3. Затраты по второй статье.

№	Наименование	Единица измерения	Количество	Цена руб.	Стоимость руб.
1	Персональный компьютер	комплект	9	30000	270000
Итого:					270000

Данное техническое оборудование рассчитано на срок использования до 5 лет. Его стоимость будет частично входить в затраты по второй статье. Данная стоимость будет рассчитана по следующей формуле:

$$S = S_{\text{комп}} / X \cdot 12, \text{ где}$$

$S_{\text{комп}}$  – стоимость персональных компьютеров, руб.;

$X$  – срок использования, лет;

12 – количество использованных месяцев в году.

И тогда  $S = 270000 / 5 \cdot 12 = 4500$  руб.

Так как срок разработки проекта – 2 месяца, то сумма затрат по второй статье составит:

$$S_2 = 4500 \cdot 2 = 9000$$

#### **4.7 Заработная плата персонала**

Заработанная плата персонала [17] – это выплаты исполнителям за отработанное время, учитывая количество исполнителей, задействованных в данном проекте. Расчет основной заработной платы производится подсчетом затрат на каждого участника проекта, учитывая месячную оплату труда. В данном случае в одном месяце 22 рабочих дня. Данные расчеты представлены в таблице 4.4



Таблица 4.4. Затраты по третьей статье

№ этапа	Исполнитель	Трудоемкость в днях	Оклад (за 22 раб. дня)	Оплата за 1 день	Оплата за период
1	Начальник отдела	1	38000	1727,3	1727,3
2	Начальник отдела	1	38000	1727,3	1727,3
2	Технический лидер	1	30000	1363,6	1363,6
	Начальник дизайн отдела	1	25000	1136,4	1136,4
3	Начальник дизайн отдела	7	25000	1136,4	7954,8
4	Начальник дизайн-отдела	1	25000	1136,4	1136,4
	Технический лидер	1	30000	1363,6	1363,6
5	Начальник отдела	1	38000	1727,3	1727,3
	Технический лидер	1	30000	1363,6	1363,6
	Начальник дизайн отдела	1	25000	1136,4	1136,4
6	Дизайнер 2D графики	10	20000	909,1	9091
	Дизайнер 3D графики	15	20000	909,1	13636,5
	Дизайнер материалов	10	20000	909,1	9091
7	Начальник отдела	1	38000	1727,3	1727,3
	Начальник дизайн отдела	1	25000	1136,4	1136,4
8	Программист интерфейса	10	18000	818,2	8182
	Программист баз данных	20	20000	909,1	18182
	Логический программист	10	20000	909,1	9091
9	Программист интерфейса	2	18000	818,2	1636,4

	Программист баз данных	5	20000	909,1	4545,5
	Логический программист	3	20000	909,1	2727,3
10	Технический лидер	1	30000	1363,6	1363,6
	Начальник отдела	1	38000	1727,3	1727,3
11	Технический лидер	5	30000	1363,6	6818
	Начальник отдела	2	38000	1727,3	3454,6
	Начальник дизайн отдела	3	25000	1136,4	3409,2
Итого	Начальник отдела	6	10363,8		
	Начальник дизайн отдела	15	17046		
	Технический лидер	9	12272,4		
	Дизайнер 2D элементов	10	9091		
	Дизайнер 3D элементов	15	13636,5		
	Дизайнер материалов	10	9091		
	Программист интерфейса	12	9818,4		
	Программист баз данных	25	22727,5		
	Логический программист	13	11818,3		
	Общее	115	115864,9		

Таким образом, суммарные расходы на выплаты заработной платы равны:

$S_3 = 115864,9$  (руб.)

#### **4.8 Страховые взносы**

Страховые взносы [18] высчитываются как 30% от фонда оплаты труда, равного сумме основной и дополнительной заработной платы персонала. Однако, в данном случае дополнительная заработная плата не предусмотрена. Таким образом,

$S_4 = (S_3 + 0) * 30\%$  (руб.)

И тогда  $S_4 = 115864,9 * 0,30 = 34759,47$  (руб.)

#### **4.9 Накладные расходы**

К накладным расходам [19] относятся: затраты на содержание аппарата управления помещений, использование материалов на хозяйственные нужды, почтово-телеграфные расходы, приобретение необходимой литературы и прочее. В данном случае эти расходы минимальны и будут составлять 50 % от заработной платы:

$S_5 = 115864,9 * 0,5 = 57\,932,45$  (руб.)

#### **4.10 Себестоимость проекта**

На основе полученной выше информации о перечне затрат можно рассчитать себестоимость разработки. Данная информация представлена в таблице 4.5

Таблица 4.5. Определение себестоимости разработки

№	Наименование статей	Сумма(руб.)
1	Материалы и покупные изделия	6842,5
2	Техническое оборудование	9000
3	Заработная плата персонала	115864,9
4	Страховые взносы	39394,1
5	Накладные расходы	57 932,45
Итого себестоимость разработки		229033,95

#### **4.11 Договорная цена**

Плановая прибыль (ПП) [20] от разработки данного проекта составляет 30 % от себестоимости разработки проекта (СР), т.е.

$ПП = СР * 0,3 = 229033,95 * 0,3 = 68710,2$  (руб.)

Оптовая цена разработки (ОЦР) определяется как сумма плановой прибыли (ПП) от разработки и себестоимости разработки (СР):

$ОЦР = ПП + СР = 68710,2 + 229033,95 = 297744,2$  (руб.)

На 2017 год ставка НДС составляет 18% от оптовой цены разработки (ОЦР):

$НДС = ОЦР * 0,18 = 53594$  (руб.)

Цена разработки (ЦР), предлагаемая на рынке, рассчитывается как сумма НДС, плановой прибыли (ПП) и себестоимости разработки (СР):

$ЦР = НДС + ПП + СР = 68710,2 + 297744,2 + 53594 = 420048,4$  (руб.)

#### **4.12 Обоснование экономической целесообразности проекта**

Разработанный проект представляет собой особый товар, имеющий ряд характерных черт и особенностей, в числе которых – специфика труда по

созданию продукта, определение цены на продукт, обоснование затрат на разработки

Применение данного проекта позволяет пользователям экономить на создании собственных средств визуализации. Также созданная система в силу своей простоты не нуждается в привлечение дополнительных специалистов. Помимо экономии программа может быть использована в рекламных целях, позволяя клиентам представить себя в покупаемом помещении.

Для оценки целесообразности проекта необходимо провести сравнительный анализ разрабатываемой системы и решений-аналогов, имеющихся на рынке, о которых уже говорилось выше (табл. 4.6).

Таблица 4.6 – Сравнение систем анализа показателей ПО

<b>Критерий</b>	<b>Разрабатываемая система</b>	<b>Planner 5D</b>	<b>IKEA Homeplaner</b>
Интерактивные объекты	+	-	-
Реалистичные модели моделей	+	-	-
Инструмент перемещения объектов	+	+	+
Инструмент редактирования стен и потолка	+	-	-
Браузерная версия	-	+	-
Высокий уровень оптимизации	+	-	-

<b>Критерий</b>	<b>Разрабатываемая система</b>	<b>Planner 5D</b>	<b>IKEA Homeplaner</b>
Режим просмотра	+	-	-
Привязка выключателей к источникам освещения	+	-	-
Сохранение сцены	+	+	+
Инструмент редактирования материала объекта	+	+	+

### ***Выводы по разделу***

В четвертом разделе был представлен анализ экономической целесообразности системы. Из анализа можно сделать вывод, что проект является целесообразным, так как разрабатываемая система имеет реалистичную графику моделей, интерактивные объекты, режим просмотра и возможность привязки выключателей к конкретным источникам освещения.

Таким образом, фонд оплаты труда составил 115 864 рублей. Полная себестоимость составила 229 033 рублей, а плановая прибыль – 68 710 рублей. Договорная цена разработки системы моделирования интерьеров была определена в размере 420 048 рублей.

## **Заключение**

В выпускной квалификационной работе была описана разработка программной системы моделирования интерьеров с описанием используемых технологий. Проведен анализ предметной области, в процессе которого были рассмотрены основные аспекты и идеи визуального моделирования, и по его результатам была сформулирована концептуальная модель визуального моделирования.

В результате проектирования системы была построена концептуальная схема объектов интерьера, на основе которой была реализована программная база данных сущностей.

В конечном итоге, разработанная программная система моделирования интерьеров успешно прошла все необходимые испытания, соответствует заявленным в задании требованиям, имеет высокое визуальное качество объектов, а также может быть оптимизирована до возможности запуска на платформе HTML5 и использоваться в качестве WEB-приложения.

Также было приведено экономическое обоснование целесообразности применения программной системы.

## Список используемых источников

1. Сайт по 3D моделированию – Autodesk [Электронный источник] <http://www.autodesk.ru> // URL: <https://www.autodesk.ru/solutions/3d-modeling-software> (дата обращения: 04.06.2018)
2. Сайт информационных технологий [Электронный источник] <https://www.inf1.info> // URL: <https://www.inf1.info/modeling> (дата обращения: 04.06.2018)
3. Климачева Т.Н. Трехмерная компьютерная графика и автоматизация проектирования на VBA в AutoCAD: - М.: ДМК, 2007. – 464 с.
4. Корнеев В.И. Интерактивные графические системы: - М.: БИНОМ, 2012. – 232с.
5. Сайт по работе с 3D графикой – Unreal Engine documents [Электронный источник] <https://www.unrealengine.com> // URL: <https://docs.unrealengine.com/latest/INT>. (дата обращения: 04.06.2018)
6. Нечаев В.В. Конфигурационное моделирование: часть 1. Теоретические аспекты: Учебное пособие/ Государственное образовательное учреждение высшего профессионального образования «Московский государственный институт радиотехники, электроники и автоматики (технический университет)». – М.: 2007. – 92 с.
7. Нечаев В.В. Концептуальное метамоделирование структур. // Международная Академия информатизации; Отделение «Математическое и компьютерное моделирование». – М.: Международное изд. «Информация», 1997. – 52 с.; илл. 2; табл. 11; библи. 22 назв.
8. Буч, Гради. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ : Пер. с англ. // Г. Буч .— 2-е изд .— Москва; Санкт-Петербург : Бином : Невский Диалект, 2001 .— 558 с.: ил .— Доп. тит. л. на англ. яз. — Библиогр.: с.479-548 .
9. Создание модульных структур программного продукта – Студопедия [Электронный источник] <http://studopedia.ru> // URL: [http://studopedia.ru/18\\_64582\\_sozdanie-modulnih-struktur-programmnogo-produkta.html](http://studopedia.ru/18_64582_sozdanie-modulnih-struktur-programmnogo-produkta.html) (дата обращения: 04.06.2018)
10. Создание архитектуры программы или как проектировать табуретку – Хабрахабр [Электронный источник] <https://habrahabr.ru> // URL: <https://habrahabr.ru/post/276593> (дата обращения: 04.06.2018)
11. Тьюториал по Unreal Engine. Часть 1: знакомство с движком – Хабрахабр [Электронный источник] <https://habrahabr.ru> // URL: <https://habr.com/post/344394> (дата обращения: 04.06.2018)
12. Архитектурная визуализация в Unreal Engine 4 – Хабрахабр [Электронный источник] <https://habrahabr.ru> // URL: <https://habr.com/post/253503> (дата обращения: 10.06.2018)

13. Wikipedia – Unreal Engine [Электронный ресурс] <https://ru.wikipedia.org> // URL: [https://ru.wikipedia.org/wiki/Unreal\\_Engine](https://ru.wikipedia.org/wiki/Unreal_Engine) (дата обращения: 10.06.2018)
14. Unreal Engine 4 [Электронный ресурс] <https://www.unrealengine.com> (дата обращения: 10.06.2018)
15. Расчёт оптовой цены, прибыли, рентабельности – Студопедия [Электронный источник] <http://studopedia.ru> // URL: <https://studfiles.net/preview/1867908/page:9/> (дата обращения: 04.06.2018)
16. Транспортно-заготовительные расходы (ТЗР) - Бухгалтерский учет. Налоги. Аудит [Электронный ресурс] <https://www.audit-it.ru> // URL: [https://www.audit-it.ru/terms/accounting/transportno\\_zagotovitelnye\\_raskhody\\_tzr.html](https://www.audit-it.ru/terms/accounting/transportno_zagotovitelnye_raskhody_tzr.html) (дата обращения: 10.06.2018)
17. Заработанная плата. Виды, формы, факторы, влияющие на заработную плату – konspekts [Электронный ресурс] <http://konspekts.ru> // URL: <http://konspekts.ru/ekonomika-2/zarabotnaya-plata/zarabotnaya-plata-vidy-formy-factory-vliyayushhie-na-zarabotnuyu-platu/> (дата обращения: 10.06.2018)
18. Страховые взносы: тарифы, ставки, размеры – Главбух [Электронный ресурс] <https://www.glavbuh.ru> // URL: <https://www.glavbuh.ru/rubrika/157> (дата обращения: 14.06.2018)
19. Накладные расходы: расчет и оптимизация – Финансовый директор [Электронный ресурс] <https://fd.ru> // URL: <https://fd.ru/articles/159011-nakladnye-rashody--18-m1> (дата обращения: 14.06.2018)
20. Договорная цена – Мир науки [Электронный ресурс] <http://worldofscience.ru> // URL: <http://worldofscience.ru/jekonomika/72-strahovanie/6091-dogovornaya-tsena.html> (дата обращения: 14.06.2018)