

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.04.01 – Информатика и вычислительная техника
Магистерская программа компьютерное моделирование

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

«_____» _____ 2016 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Компьютерное моделирование общего технического устройства
автокрана Урал-4320

Исполнитель

студент группы 453ом

(подпись, дата)

П.Е. Куксенко

Руководитель

профессор, доктор техн.
наук

(подпись, дата)

И.Е.Еремин

Руководитель

магистерской программы
профессор, доктор техн.
наук

(подпись, дата)

Е.Л. Еремин

Нормоконтроль

доцент

(подпись, дата)

В.В.Еремина

Рецензент

(подпись, дата)

М.Д. ШТЫКИН

Рецензент

(подпись, дата)

Д.В. Фомин

Благовещенск 2016

РЕФЕРАТ

Магистерская диссертация содержит 92 с., 29 рисунков, 7 таблиц, 53 источника.

КОМПЬЮТЕРНАЯ ГРАФИКА, 3D МОДЕЛИРОВАНИЕ, БАЗА ДАННЫХ, САПР, ВИЗУАЛИЗАЦИЯ, 3DS MAX, UNREAL ENGINE, ТЕКСТУРА, ПОЛИГОНАЛЬНОЕ МОДЕЛИРОВАНИЕ, РЕНДЕРИНГ.

Объектом исследования данной работы является подход к созданию и визуализации трехмерных моделей автомобилей..

Цель работы – создание приложения, позволяющего наглядно изучить общее техническое устройство автокрана.

Выполнение работы включает несколько этапов. Первым этапом является изучение предметной области, обзор существующих подходов к решению задачи, выдвигаются общие требования к создаваемому программному продукту. На втором этапе рассматриваются существующие программы, обосновывается выбор среды разработки, предлагается альтернативный подход к решению задачи. Третьим этапом является описание алгоритмических и кодовых процедур реализации программы, детально описывается характеристика структуры и возможностей программы.

Данное приложение позволит наглядно изучить общее техническое устройство автокрана Урал-4320.

					ВКР.145366.090401.ПЗ			
Изм	Лист	№ докум.	Подп.	Дата				
Разраб.		Куксенко П.Е.				Лит.	Лист	Листов
Пров.		Еремин И.Е.			У	2	92	
Рук. маг. прог.		Еремин Е.Л.			АмГУ кафедра ИУС			
Н. контр.		Еремина В.В.						
Зав. каф.		Бушманов А.В.						
					КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ОБЩЕГО ТЕХНИЧЕСКОГО УСТРОЙСТВА АВТОКРАНА УРАЛ-4320			

СОДЕРЖАНИЕ

Введение	6
1 Предметная область компьютерного моделирования	8
1.1 Обзор существующих методов решения задачи	8
1.2 Описание предметной области	18
1.2.1 Этапы трехмерного моделирования	20
1.2.2 Принципы современного трехмерного моделирования	24
1.2.3 Полигональное моделирование	26
1.3 Описание объекта моделирования	30
1.4 Общие требования к создаваемому программному продукту	34
2 Программное и алгоритмическое обеспечение решения задачи	37
2.1 Выбор средств разработки	37
2.1.1 Обзор существующих программных продуктов	37
2.1.2 Обоснование выбора среды разработки	43
2.2 Примеры аналогичных программных продуктов	50
2.3 Построение решения задачи	53
3 Подробное описание разработанного программного продукта	54
3.1 Описание алгоритмических и кодовых процедур реализации программы	54
3.3.1 Описание создания модели	54
3.3.2 Визуализация модели	60
3.3.3 Программная реализация	65
3.3.4 Описание информационной подсистемы	74
3.2 Руководство пользователя	76
Заключение	81
Библиографический список	83
Приложение А	88

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		3

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей магистерской диссертации использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 2.104-68 ЕСКД Основные надписи

ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам

ГОСТ 2.111-68 ЕСКД Нормоконтроль

ГОСТ 7.1-2003 Библиографическое описание документа. Общие требования и правила составления

ГОСТ 19.201-78 ЕСПД Техническое задание. Требования к содержанию и оформлению

ГОСТ 19.401-78 ЕСПД Текст программы. Требования к содержанию и оформлению

ГОСТ 19.402-78 ЕСПД Описание программы.

ГОСТ 19.404-79 ЕСПД Пояснительная записка. Требования к содержанию и оформлению

ГОСТ 19.502-78 Описание применения. Требования к содержанию и оформлению.

ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению.

ГОСТ 34.601-90 КСАС Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания

ГОСТ 34.602-89 КСАС Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы управления

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		4

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

БД	база данных;
ПК	персональный компьютер;
ПО	программное обеспечение
САПР	система автоматизированного проектирования;
3D	3-dimensional (трехмерный);
CAD	Computer Assisted Detection;
ОЗУ	Оперативное запоминающее устройство;
ЭВМ	электронная вычислительная машина.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		5

ВВЕДЕНИЕ

Возможность опредметить любую идею нужна была человечеству издавна. В десятки раз проще объяснить свой замысел большинству, если презентовать его во вполне понятной и доступной форме – материальной. Абстрактные идеи удел многих математиков, теологов, философов. Подавляющее же большинство нуждается в простом и понятном примере, который можно увидеть, потрогать руками, мысленно оценить его высоту, ширину, объем и габариты. Так уже устроен человеческий мозг – оперирование чисто академическими, нематериальными объектами дается ему тяжело.

Воображение стремится все визуализировать, придать форму, размер и цвет. Говоря компьютерным языком, у человека весьма развит навык под названием 3D моделирование. Но процесс представления всех идей из абстрактности в конкретную форму происходит не только в голове у человека. Визуализация имеет и обратный ход, человечество активно использует макеты для презентации своих идей – будь то строительство будущего здания, проектировка концепт-кара или разработка нового крейсера для военно-морских сил. В прошлом веке для этого использовали картон, дерево и массу других материалов, создать качественный макет с сохранением точности в деталях и пропорций – задача не из легких даже для профессионала.

Именно это и призвана делать сфера, что получила название 3D моделирование. За последние несколько лет ее рост стабильно быстр. И это неудивительно, вариантов приложения подобной технологии масса. Чего стоит медицина, там замахнулись на то, чтобы со временем создавать человеческие органы. Печатать их на 3D-принтерах так же, как сейчас печатают макета и прототипы. Безусловно, процедура несколько сложнее обычной, но механизм отлажен. Его необходимо лишь усовершенствовать. А человеческий гений не знает границ, особенно если перед ним стоит конкретная и полезная цель.

Целью проводимого исследования является изучение имеющихся на сегодняшний день способов, для создания трехмерных моделей грузовых специализи-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		6

рованных автомобилей. Предложение своего подхода к созданию модели и дальнейшей визуализации, используя, находящиеся в свободном доступе, современные средства трехмерного моделирования, а именно 3D редактор и игровой движок. Написание интерактивного приложения для изучения общего технического устройства автокрана.

Научная новизна данного подхода заключается в использовании мультимедийных средств, которые применяют в основном для создания компьютерных игр, для написания высоконаглядного обучающего приложения.

Полученный в результате исследования программный продукт имеет практическую значимость, и может быть использован в качестве учебного пособия при подготовке молодых специалистов в автошколе.

Публикации результатов представлены в двух научных журналах:

1) Куксенко, П.Е. Трёхмерное компьютерное моделирование технического устройства автомобиля // XVII региональная научно-практическая конференция «Молодёжь XXI века: шаг в будущее». – Благовещенск : Изд. БГПУ, 2016. – Т.3. – С. 212-214

2) Куксенко, П.Е. Трёхмерное компьютерное моделирование технического устройства автомобиля // научный журнал «Вестник магистратуры». – Йошкар-Ола : ООО «Коллоквиум», 2016. – Т.2. – С. 134-135

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		7

1 ПРЕДМЕТНАЯ ОБЛАСТЬ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

1.1 Обзор существующих методов решения задачи

С самого начала исследований в области трехмерной графики конечной целью исследователей является синтезирование реалистичной графической сцены (визуализируемого трехмерного пространства), подобной реальному изображению. С этой целью выполнялись исследования по традиционным технологиям рендеринга (визуализации) с использованием полигональных моделей, результатом которых стали разработки технологий моделирования и рендеринга, обеспечивающие получение весьма реалистических трехмерных представлений среды. Однако процедура генерирования усложненной модели требует огромных усилий экспертов и больших затрат времени. Кроме того, реалистическое и усложненное представление среды требует очень больших объемов информации и обуславливает снижение эффективности в хранении и передаче.

В настоящее время полигональные модели в типовом случае используются для представления трехмерных объектов в компьютерной графике. Произвольная форма может быть по существу представлена множеством цветных многоугольников, т.е. треугольников. Значительно усовершенствованные алгоритмы программного обеспечения и разработка аппаратных средств графики позволяют визуализировать комплексные объекты и сцены как в значительной мере реалистические полигональные модели неподвижных и движущихся изображений.

Однако в последнее десятилетие весьма активно проводились поиски альтернативных трехмерных представлений. Основные причины этого включают трудность конструирования полигональных моделей для объектов реального мира, а также сложность рендеринга и неудовлетворительное качество формирования сцен истинно фотографического качества.

Необходимые приложения требуют чрезвычайно большого количества многоугольников; например, детальная модель человеческого тела содержит несколько миллионов треугольников, что вызывает трудности в обработке. Хотя последние достижения в методах определения расстояний, например с использованием лазер-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		8

ного сканера расстояний, позволяют получать данные расстояний высокой плотности с допустимыми ошибками, по-прежнему является дорогостоящим и очень трудным получение плавной (без швов) полной полигональной модели объекта в целом. С другой стороны, алгоритмы рендеринга для получения приближенного к фотографии качества требуют сложных вычислений сложными и не обеспечивают рендеринга в реальном времени.

При помощи компьютерных тренажеров, в современном мире, можно получить большой объем практических знаний, что стало возможно благодаря удешевлению вычислительной техники и появлению большого количества сред моделирования 3D реальности, но для того чтобы получить практические знания в более короткие сроки и более качественно необходимо создать тренажеры, которые будут наглядно и достоверно визуализировать учебный процесс. В настоящее время одной из актуальных задач при проектировании и создании интерактивных обучающих комплексов является моделирование трехмерных составных частей объекта, которые бы с точностью повторяли существующие элементы. Поэтому огромную роль играют ключевые элементы, которые будут передавать четкую картинку и иллюстрировать качественно действия рабочей техники в рамках образовательного процесса.

Помимо симуляторов и тренажеров, 3D модели различной техники играют немаловажную роль в процессе обучения, которые придают всем этим приложениям не только увлекательность при использовании, но и наглядность. Таким образом существует необходимость в создании мультимедийных высоконаглядных трехмерных моделей, отображающих общее техническое устройство автомобилей.

Рассмотрим несколько примеров существующие на сегодняшний день методов решения поставленной задачи. Первый метод, это создание моделей с помощью САД средств системы КОМПАС-3D.

КОМПАС-3D – это система твердотельного моделирования. Это значит, что все ее операции по созданию и редактированию трехмерных моделей предназначены только для работы с твердыми телами. Твердое тело – область трехмерного пространства, состоящая из однородного материала и ограниченная замкнутой поверх-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		9

ностью, которая сформирована из одной или нескольких стыкующихся граней. Любое твердое тело состоит из базовых трехмерных элементов: граней, ребер и вершин.

– грань – гладкая (не обязательно плоская) часть поверхности детали, ограниченная замкнутым контуром из ребер. Частный случай – шарообразные твердые тела и тела вращения с гладким профилем, состоящие из единой грани, которая, соответственно, не имеет ребер.

– ребро – пространственная кривая произвольной конфигурации, полученная на пересечении двух граней.

– вершина – точка в трехмерном пространстве. Для твердого тела это может быть одна из точек на конце ребра.

Твердые тела в системе КОМПАС-3D создаются путем выполнения булевых операций над отдельными объемными элементами детали (призмами, телами вращения и т. д.). Другими словами, процесс построения состоит из последовательного добавления и (или) удаления материала детали. Контур формы добавляемого или удаляемого слоя материала определяется плоской фигурой, называемой эскизом, а сама форма создается путем перемещения этого эскиза в пространстве (вращение вокруг оси, выдавливание перпендикулярно плоскости эскиза, перемещение по траектории и пр.). В общем случае любое изменение формы детали (твердого тела) называется трехмерной формообразующей операцией, или просто операцией.

Формировать твердотельные модели в КОМПАС-3D можно в двух типах документов: КОМПАС-Деталь и КОМПАС-Сборка. В отличие от графических документов (чертеж и фрагмент), оба типа трехмерных документов равноценны, среди них нет главного или вспомогательного.

Документ Деталь предназначен для создания с помощью формообразующих операций и хранения модели целостного объекта (чаще всего какого-либо простого изделия, отдельной детали, компонента). Хотя вовсе не обязательно, чтобы модель в документе КОМПАС-Деталь отвечала реальной единичной детали на производстве. Например, никто не мешает представить вам в качестве единой детали трехмерную модель подшипника (в действительности состоящего из нескольких деталей),

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		10

если вам так удобнее использовать его в сборках, параметризировать или редактировать.

В документе Сборка собираются в единый агрегат смоделированные и сохраненные ранее детали: вы сначала размещаете их в пространстве, сопрягаете вместе и фиксируете. Более того, в десятой версии программы функционал по наполнению сборок заметно расширился: теперь вы можете создавать прямо в сборке тела, которые будут принадлежать сугубо сборке (храниться в файле сборки, а не в отдельном файле детали или библиотеке стандартных элементов). Грубо говоря, начиная с десятой версии приложения сборка стала чем-то наподобие документа-детали, в который можно вставлять другие детали из несвязанных документов.

Необходимо также отметить, что в ранних версиях КОМПАС-3D при создании детали существовало жесткое ограничение: в документе КОМПАС-Деталь может быть выполнено только одно твердое тело. Вся геометрия построенной модели детали основывалась на одной базовой формообразующей операции (например, операции вращения или выдавливания), называемой основанием детали. Перед началом формирования 3D-модели, чтобы получить нормальную модель, всегда нужно было выбрать какой-либо элемент в реальном объекте, который бы служил базой для всех построений. Это связано с тем, что все последующие формообразующие операции отталкивались от основания детали, как бы нанизывались на него, и не могли выполняться отдельно. При неудачном выборе базового элемента последующие доработка и редактирование модели оказывались иногда очень затруднительными.

Начиная с КОМПАС-3D V8 Plus это ограничение снято. Теперь в детали, как и в сборке, можно создавать несколько не связанных друг с другом твердых тел (в сборке именно создавать, вставлять и ранее можно было сколько угодно). Такой подход получил название многотельного моделирования. Оно значительно упрощает разработку сложных деталей, снимая ограничения на создание моделей, которые раньше можно было получить лишь в режиме редактирования детали в сборке. Это значит, что булевы операции, которые до этого выполнялись только в сборке, теперь доступны при создании детали.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		11

Многотельность также позволяет создавать модель «с разных сторон». Конструктору теперь необязательно отталкиваться от одной базовой операции в детали или элементов, привязанных к ней (что было не всегда оправдано с точки зрения удобства моделирования и последующего редактирования модели). Сейчас можно формировать модель, начиная с любой ее части, создавая сначала сколь угодно много отдельных тел, свободно размещенных в пространстве, и постепенно объединяя их по мере проектирования (рисунок 1).

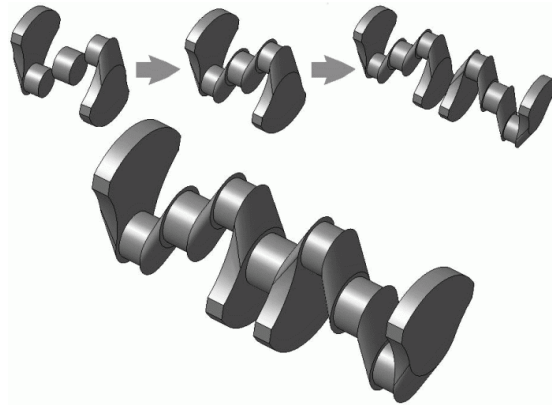


Рисунок 1 – Коленчатый вал: пример многотельного моделирования

При выполнении большинства операций в детали в связи с появлением многотельности добавился выбор нескольких вариантов (режимов) построения:

- при вырезании (удалении материала):
 - вычитание элемента – удаление материала детали происходит внутри замкнутой поверхности, сформированной по заданному эскизу и типу операции (выдавливание, вращение и т. д.);
 - пересечение элементов – удаление материала детали, находящегося снаружи поверхности, которая сформирована в результате операции;
- при «приклеивании» (добавлении материала):
 - новое тело – добавляемый трехмерный элемент формирует в детали новое твердое тело, независимо от того, пересекается он с уже существующими телами или нет. Если создаваемый элемент не имеет пересечений или касаний с существующей геометрией детали, то эта функция включается автоматически;
 - объединение – добавляемый элемент соединяется с твердым телом, с которым он пересекается;

– автообъединение – при этом система автоматически объединяет в одно тело существующий и новый элементы, если они пересекаются, или формирует новое тело, если они не пересекаются.

Результат формообразующей операции выбирается на вкладке Вырезание панели свойств при удалении или Результат операции – при добавлении материала

Очень важное понятие при многотельном моделировании – область применения операции. Представьте себе ситуацию, когда вследствие выполнения той или иной команды создаваемый элемент пересекает несколько твердых тел в модели. Какие действия предпримет система и какой результат будет у этой операции? Чтобы пользователь мог дать конкретный ответ на эти вопросы, и была реализована область применения операции. Например, если элемент выдавливания пересекает два (или более) тела, вы можете указать, с каким из этих тел объединять добавляемый элемент, объединять ли вообще или же формировать изо всех пересекающихся объектов одно твердое тело. Точно так же и при вырезании: настроив область применения операции, вы укажете, какие тела нужно «резать» (удалять часть их материала), а какие оставить нетронутыми. Другими словами, область применения операции – это набор тел, на которые распространяется действие текущей операции. Данный набор формируется простым указанием тел в окне представления модели после нажатия кнопки Ручное указание тел на панели свойств.

При добавлении материала к детали настраивать область применения операции можно только в режиме объединения (это естественно, так как в противном случае создается набор отдельных тел). Для операций удаления материала задать область применения операции можно всегда (конечно, если формообразующий элемент операции пересекается с другими телами модели).

Количество тел в текущей детали отображается в дереве построения в скобках справа от названия детали. При структурном отображении состава модели в дереве построения формообразующие операции, относящиеся к разным телам, показываются в отдельных группах.

Однако, наряду со многими преимуществами многотельного моделирования, способы получения нескольких тел в модели ограничены следующим:

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		13

– Каждое тело в модели детали должно быть неразрывным, из чего следует, что не допускается выполнение таких формообразующих операций, которые разделяют одно или несколько тел на части. Например, нельзя с помощью операции вырезания (или какой-либо другой) разбить тело на несколько нестыкующихся частей. Если вы точно знаете, что в вашей детали будет несколько разрозненных частей, необходимо сразу создавать их как отдельные тела.

– Нельзя перемещать тела в модели (как, например, детали в сборке), кроме как изменяя положения их эскизов.

– Невозможно копировать тела с помощью команд создания массивов. Тело, полученное в результате булевой операции или операции Зеркально отразить тело, также нельзя использовать в массивах. Более того, любые элементы тела, участвовавшего в булевой операции, также не получится размножить.

– При применении массивов в деталях с несколькими твердыми телами копируемые элементы (приклеенные или вырезанные) размещаются на том же теле, что и исходный элемент.

– При наличии пересекающихся, но разных тел в одной детали ассоциативные чертежи могут быть неправильно построены.

Еще один способ создания высоконаглядных моделей это моделирование с помощью средств системы Autodesk AutoCAD. Программа AutoCAD, созданная компанией Autodesk, является сегодня наиболее распространенной программной графической системой автоматизированного проектирования в промышленности, насчитывающей свыше двух миллионов зарегистрированных пользователей. Сама компания Autodesk занимает четвертое место в мировой таблице о рангах среди разработчиков программного обеспечения для персональных компьютеров. Общеизвестным в мире персональных компьютеров сокращением CAD называют как систему конструирования (computer aided design), так и систему технического черчения (computer aided drafting and drawing) с помощью компьютера. (Термин CAD переводится на русский язык как САПР – система автоматизированного проектирования).

Первая версия AutoCAD увидела свет в 1982 году. Работала она в среде DOS. Это была первая по-настоящему работоспособная программа автоматизиро-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		14

ванного проектирования, способная работать на персональном компьютере. В тот период подобного рода системы использовали в качестве технической базы достаточно мощные рабочие станции, а многие системы вообще эксплуатировались на больших универсальных вычислительных машинах. Успех AutoCAD во многом объяснялся принятой при ее разработке концепцией системы с открытой архитектурой. Главная ее особенность состояла в том, что большинство файлов представляли собой обычный текст в формате ASCII и это позволяло легко использовать содержащиеся в них данные в аналогичных пользовательских системах. Другим важным фактором была предусмотренная в AutoCAD возможность использования специализированного языка программирования AutoLISP. В результате широкие функциональные возможности превратили AutoCAD в своего рода стандарт в классе систем автоматизированного технического проектирования и выполнения чертежных работ.

Сейчас AutoCAD – это наиболее гибкая из существующих графическая программная система для персональных компьютеров, способная эффективно работать в различных областях технического проектирования. Уникальная возможность легко адаптироваться к разнообразным «человеческим» языкам, включая и те, что используют нелатинский алфавит, поставили AutoCAD вне конкуренции на международном рынке программных продуктов для САПР. В результате AutoCAD используется более чем в 150 странах мира.

Уникальный опыт, накопленный специалистами Autodesk, позволяет постоянно расширять функциональные возможности системы. Последние версии AutoCAD включают средства проектирования, трехмерного моделирования и визуализации пространственных конструкций, доступа к внешним базам данных, интеллектуальные средства нанесения размеров на чертежи, работы с файлами самых разных форматов и многое другое. Программа AutoCAD является базовой для целого ряда более специализированных САПР, используемых в различных областях техники:

- машиностроительных САПР;
- автоматизированных систем управления ресурсами;

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		15

- архитектурных САПР, по отношению к которым часто используется аббревиатура АЕС (Architectural, Engineering, Construction – архитектура, планировка, строительство);
- географических информационных систем (Geographic Information Systems – GIS);
- САПР в электротехнике и электронике;
- систем мультимедиа.

Есть, однако, и много других, менее известных областей применения AutoCAD, среди которых можно отметить моделирование одежды, промышленный дизайн.

Чертеж в системе AutoCAD – это файл с информацией, которая описывает графическое изображение. Он может быть любого заданного размера, в нем могут быть заданы любые единицы измерения (рисунок 2).

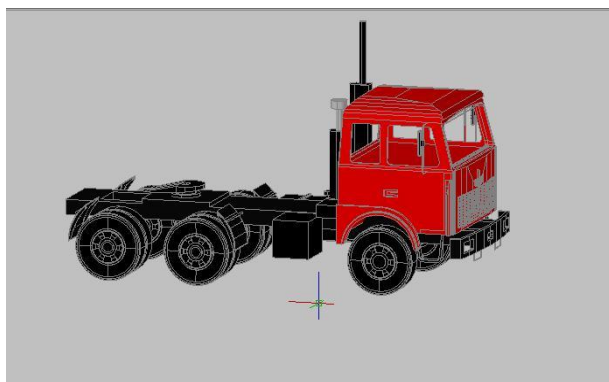


Рисунок 2 – Пример модели в среде AutoCAD

Координаты. Используется мировая система координат и несколько пользовательских систем. Для определения местоположения точек на чертеже используется декартова система координат. Любая точка на чертеже может определяться парой координат X и Y вида (x, y). Ось Z направлена перпендикулярно экрану. Точка (0,0) обычно располагается в нижнем левом углу чертежа), при этом точки могут быть заданы с использованием различных систем координат (абсолютных: декартовых, полярных, сферических, цилиндрических; относительных: декартовых и полярных).

Единицы измерения чертежа. В качестве единицы измерения может быть принята любая единица, которая в данном случае требуется. Это могут быть мет-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		16

ры, миллиметры, дюймы, и т.д. Можно указывать масштабный коэффициент, чтобы каждая единица отражала в точности тот размер, который требуется.

В системе AutoCAD чертеж строится в прямоугольной области. Пределы чертежа – это и есть границы такого прямоугольника, представленные в виде координат чертежа. Расстояние между двумя точками оценивается в единицах. Так, проведенная между точками с координатами (1,1) и (1,2) линия имеет длину в одну единицу.

Объекты в пределах чертежа. Объекты в пределах чертежа могут быть простыми или сложными (составными). Система AutoCAD обеспечивает возможность построения сложного объекта из простых и затем манипулирования этим сложным объектом, как некоторым единичным. Кроме этого, можно распределять различные фрагменты чертежа по различным уровням. Графические примитивы представляют собой заранее определенные элементы, которые можно поместить в чертеж при помощи одной команды. Примитив представляет собой такой графический элемент, как линия, окружность, текстовая строка и т.д. Система AutoCAD для построения чертежей предоставляет в распоряжение пользователя набор графических примитивов. Формы – это небольшие объекты, которые можно определять за пределами системы AutoCAD и вставлять в чертеж в заданную точку, с заданным углом и с заданным масштабом. Блоки представляют собой составные объекты, сформированные из групп других объектов. Уровни, цвета и типы линий. Различные фрагменты чертежа можно распределить по различным уровням рисовать различным цветом и различным типом линий. Можно задавать любое количество уровней. К примеру, файл чертежа может содержать на одном уровне этажный план здания, на другом уровне схему электропроводки, а на третьем уровне водопроводные коммуникации. С каждым уровнем чертежа связываются цвет и тип линий. Цвет представляет собой число от 1 до 255, соответственно которому выбирается реальный цвет, который должны иметь выводимые на экран графического монитора элементы чертежа. Консультативная информация. В систему встроенная справочная служба «HELP» где можно получить любую информацию о системе (форматы команд, опции, параметры и т.д.).

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		17

Возможность обмена графической информацией. Полученный с использованием системы AutoCAD чертеж можно хранить в виде текстового файла в коде ASCII, который может затем легко обрабатываться пользовательскими программами или передаваться в другой компьютер. Так же как и все программы AutoCad имеет свои недостатки и преимущества. К немногочисленным недостаткам программы AutoCad можно отнести сложность привязки информации из базы данных к графическим объектам.

Так как программа работает уже около двух десятков лет, многие элементы программы, которые были актуальны в прошлом, сейчас частично или полностью утратили свою актуальность, но сохранились в интерфейсе программы. Как пример, можно привести экранное меню, которое было актуально во времена, когда компьютер не имел указывающего манипулятора типа «мышь».

Основным достоинством AutoCad является доступность для создания на его базе мощных специализированных расчетно-графических пакетов. Autodesk выпускает две основные линейки продуктов, предназначенных для строителей и архитекторов (Autodesk Architectural Desktop, Autodesk Land, Autodesk Civil Design, Autodesk Map и т. д.) и машиностроителей (Autodesk Mechanical Desktop). Все эти продукты используют AutoCad как основу, поэтому пользователям, чувствующим себя уверенно в AutoCad, не составит большого труда начать работать с любым из них.

Кроме того, ряд сторонних производителей ПО разрабатывают приложения к AutoCAD, предназначенные для решения узких задач. Например, «Русская Промышленная Компания» выпускает три специализированных приложения под AutoCad: WinELSO, AutoСПДС, AutoЕСКД.

1.2 Описание предметной области

Трёхмерная графика – раздел компьютерной графики, посвящённый методам создания изображений или видео путём моделирования объёмных объектов в трёхмерном пространстве. 3D-моделирование – это процесс создания трёхмерной модели объекта. Задача 3D-моделирования – разработать визуальный объёмный образ желаемого объекта. При этом модель может как соответствовать объектам из

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		18

реального мира (автомобили, здания, ураган, астероид), так и быть полностью абстрактной (проекция четырёхмерного фрактала).

Графическое изображение трёхмерных объектов отличается тем, что включает построение геометрической проекции трёхмерной модели сцены на плоскость (например, экран компьютера) с помощью специализированных программ. Однако, с созданием и внедрением 3D-дисплеев и 3D-принтеров, трёхмерная графика не обязательно включает в себя проецирование на плоскость.

Трёхмерная графика активно применяется для создания изображений на плоскости экрана или листа печатной продукции в науке и промышленности, например, в системах автоматизации проектных работ (САПР; для создания твердотельных элементов: зданий, деталей машин, механизмов), архитектурной визуализации (сюда относится и так называемая «виртуальная археология»), в современных системах медицинской визуализации. Самое широкое применение – во многих современных компьютерных играх, а также как элемент кинематографа, телевидения, печатной продукции.

Трёхмерная графика обычно имеет дело с виртуальным, воображаемым трёхмерным пространством, которое отображается на плоской, двухмерной поверхности дисплея или листа бумаги. В настоящее время известно несколько способов отображения трёхмерной информации в объемном виде, хотя большинство из них представляет объёмные характеристики весьма условно, поскольку работают со стереоизображением. Из этой области можно отметить стереочки, виртуальные шлемы, 3D-дисплеи, способные демонстрировать трёхмерное изображение. Несколько производителей продемонстрировали готовые к серийному производству трёхмерные дисплеи. Однако и 3D-дисплеи по-прежнему не позволяют создавать полноценной физической, осязаемой копии математической модели, создаваемой методами трёхмерной графики. Развивающиеся с 1990 годов технологии быстрого прототипирования ликвидируют этот пробел. Следует заметить, что в технологиях быстрого прототипирования используется представление математической модели объекта в виде твердого тела (воксельная модель).

Традиционно рисуют в 2D (по осям X и Y) – на бумаге, холсте, дереве и т.п.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		19

При этом отображают какую-то одну из сторон предмета. Картинка сама по себе плоская. Но если мы хотим получить представление обо всех сторонах предмета, то необходимо нарисовать несколько рисунков. Так поступают в традиционной рисованной анимации. Но, вместе с тем, существует, (кстати, в СССР была довольно хорошо развита) т.н. кукольная анимация. Один раз изготовленную куклу снимают в необходимых позах и ракурсах, получая серию «плоских картинок». 3D (к X и Y добавляется координата глубины Z) визуализация – это те же «куклы», только существующие в цифровом виде. Другими словами, в специальных программах (Blender, 3ds Max, Maya, Cinema 4D и т.п.) создается объемное изображение, например авто.

Преимущество данного метода в том, что в распоряжении, скажем, аниматора есть объемная модель, необходимо лишь поместить ее должным образом в кадр, анимировать (задать траекторию передвижения или рассчитать с помощью симулятора) при необходимости, а уж отображение авто в финальной картинке ложится на специальную программу, называемую визуализатором (render). Еще одно преимущество в том, что модель достаточно нарисовать один раз, а потом использовать в других проектах (скопировав), изменять, деформировать и т.п. по своему усмотрению. Для обычного 2D рисунка, в общем случае, такое невозможно. Третье преимущество – можно создавать практически бесконечно детализированные модели, например смоделировать даже винтики на часах и т.п. На общем плане этот винтик может быть и неразличим, но стоит нам приблизить камеру, программа-визуализатор сама рассчитает, что видно в кадре, а что нет.

1.2.1 Этапы трехмерного моделирования

Для получения трёхмерного изображения на плоскости требуются следующие шаги:

- моделирование – создание трёхмерной математической модели сцены и объектов в ней;

- текстурирование – назначение поверхностям моделей растровых или процедурных текстур (подразумевает также настройку свойств материалов – прозрачность, отражения, шероховатость и пр.);

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		20

- освещение – установка и настройка источников света;
- анимация (в некоторых случаях) – придание движения объектам;
- динамическая симуляция (в некоторых случаях) – автоматический расчёт взаимодействия частиц, твёрдых/мягких тел и пр. с моделируемыми силами гравитации, ветра, выталкивания и др., а также друг с другом;
- рендеринг (визуализация) – построение проекции в соответствии с выбранной физической моделью;
- композитинг (компоновка) – доработка изображения;
- вывод полученного изображения на устройство вывода – дисплей или специальный принтер.

Моделирование сцены (виртуального пространства моделирования) включает в себя несколько категорий объектов:

- геометрия (построенная с помощью различных техник (напр., создание полигональной сетки) модель, например, здание);
- материалы (информация о визуальных свойствах модели, например, цвет стен и отражающая/преломляющая способность окон);
- источники света (настройки направления, мощности, спектра освещения);
- виртуальные камеры (выбор точки и угла построения проекции);
- силы и воздействия (настройки динамических искажений объектов, применяется в основном в анимации);
- дополнительные эффекты (объекты, имитирующие атмосферные явления: свет в тумане, облака, пламя и пр.)

Задача трёхмерного моделирования – описать эти объекты и разместить их в сцене с помощью геометрических преобразований в соответствии с требованиями к будущему изображению.

Назначение материалов: для сенсора реальной фотокамеры материалы объектов реального мира отличаются по признаку того, как они отражают, пропускают и рассеивают свет; виртуальным материалам задается соответствие свойств реальных материалов – прозрачность, отражения, рассеивания света, шероховатость, рельеф и пр. Наиболее популярными пакетами сугубо для моделирования являются

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		21

ся: Pixologic Zbrush, Autodesk Mudbox, Autodesk 3D max, Robert McNeel & Assoc, Rhinoceros 3D, Google SketchUp, Blender.

Для создания трёхмерной модели человека или существа может быть использована как прообраз (в большинстве случаев) скульптура. Текстурирование подразумевает проецирование растровых или процедурных текстур на поверхности трёхмерного объекта в соответствии с картой UV-координат, где каждой вершине объекта ставится в соответствие определенная координата на двухмерном пространстве текстуры.

Как правило, многофункциональные редакторы UV-координат входят в состав универсальных пакетов трёхмерной графики. Существуют также автономные и подключаемые редакторы от независимых разработчиков, например, Unfold3D magic, Deep UV, Unwrella и др.

Освещение заключается в создании, направлении и настройке виртуальных источников света. При этом в виртуальном мире источники света могут иметь негативную интенсивность, отбирая свет из зоны своего «отрицательного освещения». Как правило, пакеты 3D-графики предоставляют следующие типы источников освещения:

- Omni light (Point light) – всенаправленный;
- Spot light – конический (прожектор), источник расходящихся лучей;
- Directional light – источник параллельных лучей;
- Area light (Plane light) – световой портал, излучающий свет из плоскости;
- Photometric – источники света, моделируемые по параметрам яркости свечения в физически измеримых единицах, с заданной температурой накала источника.

Существуют также другие типы источников света, отличающиеся по своему функциональному назначению в разных программах трёхмерной графики и визуализации. Некоторые пакеты предоставляют возможности создавать источники объемного свечения (Sphere light) или объемного освещения (Volume light), в пределах строго заданного объёма. Некоторые предоставляют возможность использовать геометрические объекты произвольной формы.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		22

Одно из главных призваний трёхмерной графики – придание движения (анимация) трёхмерной модели, либо имитация движения среди трёхмерных объектов. Универсальные пакеты трёхмерной графики обладают весьма богатыми возможностями по созданию анимации. Существуют также узкоспециализированные программы, созданные сугубо для анимации и обладающие очень ограниченным набором инструментов моделирования:

- Autodesk MotionBuilder;
- PMG Messiah Studio.

На этапе рендеринга математическая (векторная) пространственная модель превращается в плоскую (растровую) картинку. Если требуется создать фильм, то рендерится последовательность таких картинок – кадров. Как структура данных, изображение на экране представлено матрицей точек, где каждая точка определена, по крайней мере, тремя числами: интенсивностью красного, синего и зелёного цвета. Таким образом рендеринг преобразует трёхмерную векторную структуру данных в плоскую матрицу пикселей. Этот шаг часто требует очень сложных вычислений, особенно если требуется создать иллюзию реальности. Самый простой вид рендеринга – это построить контуры моделей на экране компьютера с помощью проекции, как показано выше. Обычно этого недостаточно, и нужно создать иллюзию материалов, из которых изготовлены объекты, а также рассчитать искажения этих объектов за счёт прозрачных сред (например, жидкости в стакане).

Существует несколько технологий рендеринга, часто комбинируемых вместе, например:

- сканлайн (scanline) – он же Ray casting («бросание луча», упрощенный алгоритм обратной трассировки лучей) – расчёт цвета каждой точки картинки построением луча из точки зрения наблюдателя через воображаемое отверстие в экране на месте этого пикселя «в сцену» до пересечения с первой поверхностью. Цвет пикселя будет таким же, как цвет этой поверхности (иногда с учётом освещения и т.д.);

- трассировка лучей (рейтрейсинг, англ. raytracing) – то же, что и сканлайн, но цвет пикселя уточняется за счёт построения дополнительных лучей (отражён-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		23

ных, преломлённых и т. д.) от точки пересечения луча взгляда. Несмотря на название, применяется только обратная трассировка лучей (то есть как раз от наблюдателя к источнику света), прямая крайне неэффективна и потребляет слишком много ресурсов для получения качественной картинки;

- Z-буфер (используется в OpenGL и DirectX 10);

- глобальное освещение (англ. global illumination, radiosity) – расчёт взаимодействия поверхностей и сред в видимом спектре излучения с помощью интегральных уравнений.

Грань между алгоритмами трассировки лучей в настоящее время практически стёрлась. Так, в 3D Studio Max стандартный визуализатор называется Default scanline renderer, но он считает не только вклад диффузного, отражённого и собственного (цвета самосвечения) света, но и сглаженные тени. По этой причине чаще понятие Raycasting относится к обратной трассировке лучей, а Raytracing – к прямой.

Наиболее популярными системами рендеринга являются: PhotoRealistic RenderMan (PRMan), mental ray, V-Ray, FinalRender, Brazil R/S, BusyRay, Turtle, Maxwell Render, Fryrender, Indigo Renderer, LuxRender, YafaRay, POV-Ray.

Вследствие большого объёма однотипных вычислений рендеринг можно разбивать на потоки (распараллеливать). Поэтому для рендеринга весьма актуально использование многопроцессорных систем. В последнее время активно ведётся разработка систем рендеринга, использующих GPU вместо CPU, и уже сегодня их эффективность для таких вычислений намного выше. К таким системам относятся: Refractive Software Octane Render, AAA studio FurryBall, RandomControl ARION (гибридная), Cycles, Vray-RT, LuxRender, iray.

Многие производители систем рендеринга для CPU также планируют ввести поддержку GPU (LuxRender, YafaRay, mental images iray).

Самые передовые достижения и идеи трёхмерной графики (и компьютерной графики вообще) докладываются и обсуждаются на ежегодном симпозиуме SIGGRAPH, традиционно проводимом в США.

1.2.2 Принципы современного трехмерного моделирования

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		24

Электронное трехмерное моделирование в современном дизайн-проектировании является основным средством реализация художественного замысла формы объекта, от эскизов художественного образа до инженерных чертежей. В настоящее время электронное трехмерное моделирование классифицируется по наличию истории построения объекта (таблица 1) и по элементам построения модели: каркасное, полигональное, поверхностное, твердотельное, конечно-элементное, генеративное (таблица 2).

Таблица 1 – Виды 3D моделирования по наличию истории построения объекта

Виды	Описание	Пример программ
1	2	3
Параметрическое моделирование	Моделирование по набору заданных варьируемых параметров операций	CATIA
Непараметрическое моделирование	Моделирование без сохранения параметров построения (истории построения)	Rhinoceros
Комбинированное моделирование	Историю построения в любой момент можно удалить/отключить	Alias Studio Tools, Rhinoceros + Grasshopper

Каждый из видов моделирования нашел применение в процессе дизайн-проектирования промышленных изделий.

Таблица 2 – Виды 3D моделирования по элементам построения

№	Виды	Элементы построения	Программы	Модель
1	2	3	4	5
1	Полигональное моделирование	Полигон, кривая, (poly, nurbs, mesh)	Alias, 3ds Max, Maya, Rhino	Полигональная
2	Каркасное моделирование	Точка и линия (line & point)	Alias ST, AutoCAD, CATIA, IsemSurf	Каркасная

1	2	3	4	5
3	Поверхностное моделирование	точка, линия, поверхность (surface)	Alias ST, Auto-CAD, CATIA, Isem Surf, Solid Works	Поверхностная
4	Твердотельное моделирование	твердое тело (solid)	Alias ST, Auto-CAD, CATIA, IsemSurf, Solid-Works	Твердотельная
5	Конечно-элементное моделирование	узел, конечный элемент, сетка (point, lines, mesh)	Ansys, Rhino + Kangaroo	Конечно-элементная
6	Генеративное моделирование	компоненты, связи между компонентами	Rhino + Grasshopper	Генеративная

В индустрии дизайна существует постоянная потребность в обновлении форм, в переосмыслении взаимодействия, организации объектов и пространства. Актуальным остается снижение сроков моделирования будущего изделия. Данные задачи позволяет решить внедрение междисциплинарных принципов трехмерного моделирования в процесс проектирования.

1.2.3 Полигональное моделирование

Существует несколько способов моделирования, но самым популярным является полигональное моделирование. Нередко можно увидеть в роликах о 3D или фантастических фильмах как тот или иной объект представляется в виде т.н. сетки. (рисунок 3). Это и есть пример полигонального моделирования. Суть его в том, что поверхности представляются в виде простых геометрических двумерных примитивов. В компьютерных играх это треугольники, для других целей обычно 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, четырехугольниками. При необходимости четырехугольники (полигоны) без проблем пре-

вращаются в треугольники при экспорте в игровой движок, а при необходимости сглаживания или тесселяции модель из четырехугольников получается, как правило, без артефактов.

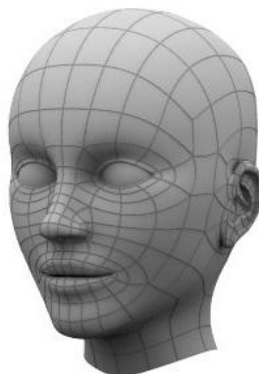


Рисунок 3 – Пример полигонального моделирования

Если какой-то объект представляется в виде полигонов (особенно органические объекты, например человек), то понятно, что чем меньше размер полигонов, чем их больше, тем более близкой может быть модель к оригиналу. На этом основан метод тесселяции: сначала изготавливают грубую болванку из небольшого кол-ва полигонов, затем применяют операцию тесселяции, при этом каждый полигон делится на 4 части. Так вот, если полигон четырехугольный (а еще лучше, близок к квадрату) то алгоритмы тесселяции дают более качественный и предсказуемый результат (рисунок 4). Также операция сглаживания, а это та же тесселяция, только с изменением углов на более тупые, при близких к квадрату полигонах, позволяет получить хороший результат.

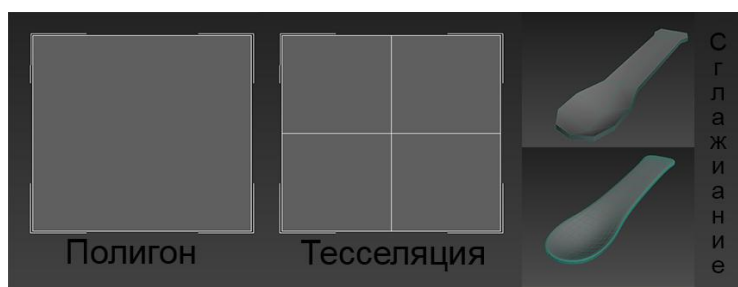


Рисунок 4 – Пример операции тесселяция

Как было сказано выше, чем больше полигонов, тем более модель может (может, потому, что модель должна быть еще похожа на оригинал, а это вопрос мастерства моделера, а не полигонов) походить на оригинал. Но у большого кол-ва полигонов есть обратная сторона: понижение производительности. Чем больше

полигонов, тем больше точек по которым они строятся, тем больше данных приходится обрабатывать процессору. Поэтому 3D графика – это всегда компромисс между детализацией модели и производительностью. В связи с этим даже возникли термины: high poly и low poly, соответственно высоко полигональная модель и низко полигональная модель. В играх применяются низко полигональные модели, так как в них выполняется визуализация в реальном времени. Кстати, модели в играх представлены треугольниками для повышения производительности: графические процессоры умеют на аппаратном уровне быстро обрабатывать сотни миллионов треугольников за секунду.

Как правило, полигональное моделирование относится к пустотелому моделированию, где объект имеет только объем, но внутри пустой. Это означает, что если мы смоделируем куб, а потом удалим одну из стенок, то увидим внутри пустоту. Также имеются программы для твердотельного моделирования, где тот же самый куб представлен в виде монолитного объекта. В таких программах (к примеру, Autodesk Inventor) применяются математические модели отличные от тех, что в полигональном моделировании. Алгоритмы твердотельного моделирования лучше подходят для моделирования механизмов при разработке техники. Программы вроде Autodesk Inventor имеют средства для моделирования с учетом особенностей технологического процесса, как то фаски, сверление отверстий, проставление размеров, допусков и т.п. Получаемые модели можно сразу отправить на подходящий станок для получения изделия в металле или другом материале.

Также существуют так называемые программы 3D лепки (ZBrush, Autodesk Mudbox) в которых моделирование сводится (грубо говоря) к созданию углублений или выпуклостей. Такая техника похожа на то, как скульпторы лепят из глины – убирая ненужное и добавляя необходимое. С помощью таких программ можно добиться реалистичного рельефа поверхности, например морщин на коже или складок ткани. В настоящее время высокополигональные (а для лепки модель должна обладать солидным кол-вом полигонов) реалистичные модели людей и вообще животного мира выполняются, в большинстве своем, с применением программы лепки. Распространена практика, когда заготовка модели создается с помощью полиго-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		28

нального моделирования, а затем в программе лепки тесселируется и добавляются мелкие детали.

Допустим у нас есть готовая модель, скажем, танка. Но на танк, собственно, она не совсем похожа. На данном этапе у нас всего лишь математическая модель содержащая данные только о геометрической форме. Но у реального объекта кроме формы есть еще и цвет, плотность, отражающая способность, и, возможно, запах. Последнее пока в 3D графике не применяется, а вот все остальное можно смоделировать. Придание модели нужного цвета и блеска называют текстурированием, от слова текстура (рисунок 5).

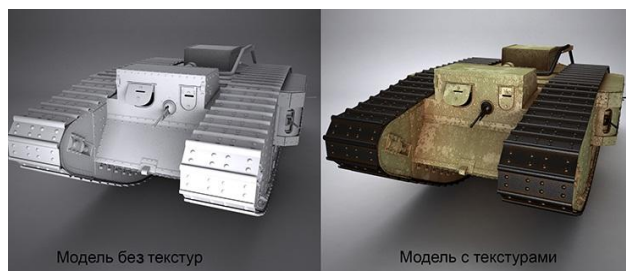


Рисунок 5 – Пример текстурирования

В общем случае текстура – это двумерный рисунок, который накладывается на 3D модель. Текстура может быть как процедурной – сгенерированной при помощи алгоритма, так и нарисованная в графическом редакторе, или фотографией реального объекта. С помощью текстуры задается рисунок и цвет модели, но реальная поверхность обладает и другими параметрами: отражающей способностью, преломлением, рельефом, прозрачностью и т.п. Все эти параметры задаются в свойствах материала. Т.е. материал с точки зрения 3D графики – это некая математическая модель описывающая параметры поверхности. Например, для воды обязательно необходимо указать прозрачность и преломляющую, отражающую способности.

Перед «нанесением» материала на 3D модель необходимо создать ее развертку, т.е. представить все (несколько, одну) поверхности в виде проекции на плоскость. Это необходимо для того, чтобы затем двумерная текстура правильно «легла» на модель.

Между обычным рисунком, скажем, на бумаге, и построением 3D изображения есть существенные различия в самом процессе. Двумерный рисунок, как правило, создается в два этапа: эскиз и раскрашивание. В 3D графике после изготовле-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		29

ния модели ее необходимо поместить в сцену к другим объектам (или в так называемую студию), добавить освещение, камеру и лишь затем можно надеяться получить финальную картинку. Изображение в 3D графике просчитывается на основе физической модели, как правило, это модель распространения луча света с учетом отражения, преломления, рассеивания и т.п. Рисуя красками мы сами отрисовываем тени, блики и т.д., а в трехмерной графике мы подготавливаем сцену с учетом освещения, материалов, геометрии, свойств камеры, программа рассчитывает итоговую картинку сама.

1.3 Описание объекта моделирования

Урал-4320 – грузовой автомобиль повышенной проходимости двойного назначения с колёсной формулой 6×6, производящийся на Уральском автомобильном заводе в Миассе (Россия), в том числе и для использования в вооружённых силах в семействе унифицированных армейских автомобилей «Суша» до 1998 года. На сегодняшний день серия Урал-4320 производится с дизельными двигателями ЯМЗ мощностью 230-312 л.с. экологического класса от Евро-4. Урал-4320 был разработан для транспортировки грузов, людей и трейлеров на всех типах дорог. Обладает значительными преимуществами по сравнению с аналогичными автомобилями: он легко преодолевает заболоченные участки, брод до 1,5 м, канавы до 2 м, рвы, подъёмы до 60 %.

Производство грузовиков и шасси линейки «Урал-4320» было начато 17 ноября 1977 и фактически данная серия производится по настоящее время, хотя и в существенно модернизированном виде.

Основное отличие Урал-4320 от линейки карбюраторного Урал-375Д – экономичный и тяговитый дизельный двигатель. На 1986 год выпущено более миллиона грузовиков. Изначально Урал-4320 оснащался дизельным двигателем КамАЗ-740, но в результате пожара на заводе двигателей КамАЗ 14 апреля 1993 года, поставки данного двигателя сторонним производителям прекратились, и на Урал-4320 (а также всех последующих моделях и модификациях грузовиков Урал) стали устанавливаться двигатели ЯМЗ-236 и ЯМЗ-238 Ярославского моторного завода. Из-за образовавшегося дефицита дизельных двигателей некоторая часть Урал-4320 вы-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		30

пуска 1993–1994 гг. была вынужденно оснащена карбюраторным двигателем ЗИЛ-375, то есть фактически было продлено производство модели Урал-375Д, но с оперением моторного отсека типа 4320. Первоначально модификации с двигателем ЯМЗ-238 отличались внешне более длинным моторным отсеком, а машины с двигателем ЯМЗ-236 сохранили такой же моторный отсек, как у машин с двигателем КАМАЗ-740. С середины 2000-х все машины независимо от модели двигателя выпускаются с удлинённым моторным отсеком. С середины 1990-х годов на Урал-4320 и Урал-5557 появился широкий бампер с фарами, а в крыльях, на старых местах крепления фар, появились пластиковые заглушки. Однако исключительно для нужд Министерства обороны, по специальному заказу, до сих пор поставляются машины с узким бампером и фарами в крыльях. С 1996 года начат выпуск облегченного двухосного (4x4) 4,2-тонного грузовика Урал-43206, конструктивно полностью идентичного трехосному Урал-4320.

В 2009-2014 гг. на автомобили серии 4320 мелкосерийно устанавливали бескапотную кабину типа Ivesco «Р» (производства СП «УралАЗ-ИВЕКО») с оригинальным скругленным интегральным стеклопластиковым капотным оперением (комплектации «-57» и «-58»). В дальнейшем такая кабина начала устанавливаться на шасси Урал-4320 и в бескапотном варианте. С переходом на двигатели стандарта Евро-4 на Урал-4320 начали применяться только современные дизели серий ЯМЗ-6565 и ЯМЗ-536.

В 2014 году с целью повышения надежности узлов и агрегатов шасси и увеличения пробега до капремонта серия Урал-4320 была модернизирована в серию «Урал-М» при сохранении кабины типа 4320. Осенью 2015 года серия «Урал-М» была модернизирована до серии Урал Next путём установки нового поколения современных кабин типа «ГАЗель Next» с оригинальным пластиковым оперением моторного отсека, а также рядом усовершенствованных узлов и агрегатов. В перспективе все гражданские версии Урал-4320 для внутреннего гражданского рынка и частично для поставок силовым ведомствам будут заменены серией «Урал NEXT», при сохранении серии «Урал-М» для поставок на экспорт. Технические характеристики транспортного средства Урал-4320 приведены в таблице 3.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		31

Таблица 3 – Технические характеристики Урал-4320

Наименование	Характеристика
1	2
Базовое шасси	Урал-4320-1958-40
Колесная формула	6х6
Полная масса автомобиля, кг	22 400
Емкость топливного бака, л	300
Дорожный просвет, мм	360
Тип двигателя	ЯМЗ-236НЕ2 дизельный, четырехтактный, шестицилиндровый, с непосредственным впрыском топлива, V-образный.
Рабочий объем, л.	11,15
Номинальная мощность при 2100 1/мин, кВт (л.с.)	169 (230)
Максимальный крутящий момент	882 (90)
Рулевое управление	Со встроенным гидравлическим усилителем двухстороннего действия
Сцепление	ЯМЗ-182, фрикционное, сухое, однодисковое, диафрагменное, втяжного типа
Коробка передач	ЯМЗ-236У, механическая, трехходовая, пятиступенчатая
Рабочая тормозная система	Барабанного типа с пневмогидравлическим приводом
Тип рамы	Клепанная, состоит из двух штампованных лонжеронов, соединенных поперечинами

Автомобильный кран КС-45717-1, грузоподъемностью 25 т предназначен для работы с обычными грузами и грузоподъемностью 20 т для работы с ядовитыми и взрывоопасными грузами, смонтирован на шасси УРАЛ-4320. Привод механизмов автокрана – гидравлический от насоса, приводимого в действие двигателем шасси. Гидропривод обеспечивает легкость и простоту управления краном, плавность ра-

					ВКР.145366.090401.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

боты механизмов, широкий диапазон рабочих скоростей, совмещение крановых операций. Допускается работа на сближенных опорах. Стрела автокрана – телескопическая трехсекционная. Выдвижение секций – гидроцилиндром и полиспадами. Для увеличения подстрелового пространства по особому заказу поставляется легкий решетчатый удлинитель стрелы (гусек). Микропроцессорный ограничитель грузоподъемности автокрана с цифровой индикацией информации позволяет следить за степенью загрузки крана, длиной и вылетом стрелы, высотой подъема оголовка стрелы; показывает фактическую величину груза на крюке и максимальную грузоподъемность на данном вылете, а также автоматически по заданным координатам ограничивает зону действия крана при работе в стесненных условиях или вблизи линии электропередачи. Установленная в ограничителе телеметрическая память фиксирует рабочие параметры, а также степень нагрузки крана в течение всего срока службы.

Основные преимущества автокрана КС-45717-1 «Ивановец»:

- простота конструкции;
- значительный запас прочности;
- отличные грузовые характеристики для стрелы 21м;
- ремонтпригодность;
- скорость работы автокрана;
- отличные вездеходные качества.

В конструкцию автокрана заложена возможность работы с опасными (ядовитыми и взрывчатыми веществами) грузами, при этом его максимальная грузоподъемность ограничивается до 20 тонн и, соответственно, уменьшаются грузовые характеристики на остальных вылетах. Характеристики крановой установки представлены в таблице 4.

Таблица 4 – Технические характеристики крановой установки

Наименование	Характеристика
1	2
Вылет при максимальной грузоподъемности, м	3,20
Вылет стрелы, м	2,00-19,70

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		33

1	2
Высота подъема максимальная, м	21,30
Высота подъема при максимальном вылете, м	8,00
Грузовой момент, тм	75
Грузоподъемность, т	25
Длина стрелы, м	9-21
Количество гидронасосов	1
Количество секций	3
Скорость подъема-опускания груза до 4,5т, м/мин	13,6
Скорость посадки груза, (при 8-и кратной запасовке) м/мин	0,2
Частота вращения поворотной части, об/мин	0,2

1.4 Общие требования к создаваемому программному продукту

В связи со сложившейся ситуацией в сфере специального образования, а именно, необходимости внедрения современных компьютерных технологий и быстрого развития информационных систем, особую актуальность представляет создание трёхмерных моделей грузовых специализированных автомобилей.

Трёхмерное моделирование позволяет увидеть предметы, которых на данный момент не существует, или существуют, но нет возможности их увидеть «вживую». Оно позволяет человеку увидеть объекты в том виде, какими они являются в действительности. Это значит, что такого рода программы дают возможность сэкономить огромное количество средств и времени, поскольку изучение общего технического устройства автомобиля непосредственно на самом автомобиле нецелесообразно, а освоение материала по чертежам и схемам не всегда должным образом информативно. Двумерный рисунок не дает полного представления об объекте и не всегда можно наглядно оценить расположение и принцип работы деталей или узлов.

Наглядность – главное преимущество трёхмерного моделирования и визуализации. Поэтому создание трёхмерной модели автомобиля, его узлов и агрегатов имеет практическую значимость в подготовке молодых специалистов в автошколах.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		34

Главное достоинство этой технологии заключается в том, что она позволяет создавать трехмерные модели с фотографической точностью, дает возможность наглядно показать мельчайшие детали объекта. Используя всю возможную информацию: рисунки, чертежи, план-проекты и текстовые описания, можно с максимальной точностью воссоздать любую модель – от свечи зажигания до целого двигателя. Благодаря 3D, пользователь может буквально «пощупать» то, что он изучает. С появлением 3D технологий больше нет необходимости объяснять «на пальцах» как будет выглядеть та или иная деталь, а используя мультимедийную визуализацию можно продемонстрировать принцип работы, в рамках одного приложения, например, автокрана.

Задача компьютерной графики – визуализация, то есть создание изображения. Визуализация выполняется исходя из описания (модели) того, что нужно отображать. Существует много методов и алгоритмов визуализации, которые различаются между собою в зависимости от того, что и как отображать. Например, отображение того, что может быть только в воображении человека – график функций, диаграмма, схема, карта. Или наоборот, имитация трехмерной реальности – изображения сцен в компьютерных развлечениях, художественных фильмах, тренажерах, в системах архитектурного проектирования. Важными и связанными между собою факторами здесь являются: скорость изменения кадров, насыщенность сцены объектами, качество изображения, учет особенностей графического устройства.

При создании программного продукта основной упор сделан на возможное использование приложения в обучающих целях, для подготовки молодых специалистов. Другими словами, программа должна содержать трехмерную, высоконаглядную модель полностью имитирующую автокран Урал-4320. Модель должна разделяться на элементы, составные части, узлы и агрегаты грузовика. При чем элементы обладали свойством прозрачности, давая возможность детальнее изучить выбранную часть. Для более подробного изучения модели необходимо реализовать скроллинг, вращение вокруг своей оси и приближение-уменьшение. Так же программа должна быть привязана к базе данных, содержащую теоретическую информацию по неисправностям составных элементов, возможные причины появления

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		35

неисправностей и методы устранения таковых. Приложению необходимо включать в себя демонстрационные плакаты, отображающие принцип работы систем автокрана, а также содержать теоретическую часть руководства по эксплуатации автогрузовика.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		36

2 ПРОГРАММНОЕ И АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ РЕШЕНИЯ ЗАДАЧИ

2.1 Выбор средств разработки

2.1.1 Обзор существующих программных продуктов

Выбор программы трехмерного моделирования, которая бы удовлетворяла всем требованиям пользователя, – нелегкая задача, особенно, если все программы, из которых приходится выбирать, одинаково незнакомы пользователю или несравнимы между собой по сложности и глубине предлагаемых возможностей. Например, перечисление возможностей типичного трехмерного приложения старшего класса будет очень неинформативно для начинающего пользователя. Поэтому необходимо ориентироваться на те программы, которые предоставляют действительно необходимые возможности, а окончательное решение о выборе программного пакета должно приниматься с учетом финансовых возможностей и временных затрат на освоение программы. Рассмотрим наиболее распространенные программные продукты:

ARCHI CAD. Мощный и профессиональный продукт для архитектурного моделирования. Довольно детально позволяет представить модель будущего проекта (здания, строения), учитывая всевозможные размеры, нагрузку на конструкции и т.п. Очень широко распространен среди архитекторов. В качестве программы для визуализации и двухмерного представления будущего ландшафта приспособлен мало. Базы растений нет, двухмерный план не отличается наглядностью, а трехмерный вообще впечатления трехмерности не оставляет. Для изучения – сложен. Множество редко нужных, а потому обычно мешающих работе настроек, окон. Хотя и доступно множество различных объектов: здания, деревья, предметы и т.п., но качество визуализации низкое, реалистичность проекта минимальна.

SketchUp – программа для быстрого создания и редактирования трехмерной графики. Проекты SketchUp сохраняются в формате «*.skp». По сравнению со многими популярными пакетами, данный обладает рядом преимуществ, заключающихся, в первую очередь, в почти полном отсутствии окон предварительных настроек.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		37

Все геометрические характеристики задаются с клавиатуры во время или сразу после окончания действия инструмента. Эта особенность позволяет избежать необходимости настраивать каждый инструмент перед его применением, а затем редактировать возможные неучтённые ошибки, но в то же время, это достоинство оборачивается недостатком, когда возникает потребность в массовом изменении геометрии созданных объектов. Тем не менее, такие ситуации возникают редко, а их исправление средствами пакета не составляет большого труда. Данный пакет очень удобен для начинающих, малознакомых с трёхмерным моделированием людей.

Blender. Эта бесплатная программа, позволяющая создавать трехмерную компьютерную графику на основе средств моделирования, анимации, обработки видео и так далее. Функциональные возможности сервиса понравятся и начинающим пользователям, и опытным 3D-модельерам. Программа небольшая по объему, при этом в ней есть все основные функции, текстуры, модели, обработчики событий. Дополнительные возможности обеспечиваются возможностью подключения плагинов. К отличительным особенностям программы Blender относятся: поддержка геометрических примитивов, наличие встроенных механизмов, наличие анимационных инструментов (инверсной кинематики, скелетной анимации, сеточной деформации), оснащение базовыми функциями нелинейного редактирования. Программа Blender востребована художниками и профессиональными 3D-модельерами, поскольку на ее основе можно создавать как 3D-визуализации, так и статические изображения или потоковое видео высокого качества. В интегрированный пакет разработки входит широкий спектр инструментов, открывающий широкие возможности для создания моделей. Кросс-платформенность – залог того, что Blender может быть одинаково эффективно использован на операционной системе.

Возможности:

- полигональное моделирование, сплайны, NURBS-кривые и поверхности;
- режим лепки;
- система частиц;
- динамика твердых и мягких тел: жидкость, шерсть/волосы и т.д.;
- скелетная анимация;

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		38

– встроенные механизмы рендеринга и интеграция со сторонними визуализаторами;

– редактор видео;

– функции создания игр и приложений (Game Blender).

Плюсы: доступность, открытый код, кроссплатформенность, небольшой размер (около 50 мегабайт), широкий функционал, возможность создания игр. Минусы: отсутствие документации в базовой поставке.

Autodesk 123D. Это бесплатная программа, которая пользуется у любителей 3D-моделирования большой популярностью. Ее особенности – в простоте и понятности интерфейса и функций. Продуманность программы – залог того, что каждый пользователь сможет легко разобраться в программе. Процесс проектирования упрощается благодаря тому, что пользователи получают исходные базовые формы и примеры моделей, тематические конструкторы. Приложение работает на базе Windows, Mac и iOS (iPad) и совместимо с 123D Sculpt, 123D Catch, 123D Make посредством общей облачной памяти. Приложение Autodesk 123D Catch позволяет быстро преобразовывать цифровые фотографии в фотореалистичные модели, при этом в основе процесса лежат облачные вычисления. Autodesk 123D Make – это версия технологии, созданная для Mac. На ее основе можно создавать развертки в формате 2D, а впоследствии собирать модели с использованием бумаги, картона, металла, дерева, пластика. 123D Make позволяет заранее представить себе, каким будет прототип модели. Как говорят создатели программы, она позволит даже неопытному пользователю показывать свои творческие умения. Благодаря приложению Autodesk 123D можно воплощать в реальность самые разные идеи, при этом готовые модели будут отличаться высокой точностью и четкостью.

PowerAnimator на SGI. Одна из программ современной трехмерной анимации на SGI - это пакет PowerAnimator фирмы AliasWavefront. Его средства моделирования, поддерживающие работу со сложными иерархическими объектами и поверхностями, представляют собой один из наиболее мощных и удобных комплексов инструментов. Пакет позволяет создавать реалистичные образы, в частности, благодаря отличным возможностям освещения - направленного и рассеянного, с

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		39

использованием бликов и других эффектов. Пакет поддерживает богатые средства затенения и техники придания реалистичности поверхностям, которые позволяют оживить гладкие и жесткие конструкции, неизбежно выдающие свое компьютерное происхождение. Возможен предварительный «быстрый» рендеринг. PowerAnimator содержит богатые инструменты анимации объектов, источников света и камер, отличный инструментарий для работы с частицами и автоматизации анимации.

Maya – промышленный стандарт 3D графики в кино и телевидении. Maya популярна среди крупных студий и масштабных проектов в рекламе, кино, игровой индустрии. Пакет идеален для создания анимации.

Возможности:

- полный набор инструментов для NURBS- и полигонального моделирования;
- мощные средства общей и персонажной анимации;
- развитая система частиц;
- технология Maya Fur (создание меха, волос, травы);
- технология Maya Fluid Effects (моделирование жидкостей, атмосферы);
- динамика твердых и мягких тел;
- широкий набор средств создания динамических спецэффектов;
- UV-текстуры, нормали и цветовое кодирование;
- многопроцессорный гибкий рендеринг.

Плюсы: огромный функционал и возможности. Минусы: длительное и сложное обучение, высокие требования к системе, высокая цена.

Modo – полноценный продукт для моделирования, рисования, анимации и визуализации. Включает также инструменты скульптинга и текстурного окрашивания. Благодаря удобству пользования и высокой производительности, у Modo репутация одного из самых быстрых инструментов моделирования. Modo популярен в сфере рекламы, разработки игр, спецэффектов и архитектурной визуализации.

Возможности:

- полигональное и моделирование SDS;
- современные инструменты анимации;

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		40

- динамика твердых и мягких тел;
- система рисования;
- материал Fur (мех) для создания волос, травы и меха;
- инструменты лепки;
- быстрая и качественная визуализация.

Плюсы: мощный и понятный инструментарий, высокая производительность.

Минусы: мало информации.

Houdini – мощный профессиональный пакет для работы с 3D графикой, в его основе процедурная, нодовая система. Houdini идеально подходит для создания сложной динамики, симуляции: частиц, жидкости, дыма, огня, имитации природных явлений и т.д. А также это отличный инструмент для создания впечатляющих визуальных эффектов. Основная область применения Houdini – киноиндустрия.

Возможности:

- полигональное и NURBS-моделирование,
- анимация (ключевая, процедурная),
- персонажная анимация,
- система частиц,
- динамика твердых и мягких тел, тканей, шерсти/волос, газов и жидкостей,
- работа с объемным звуком,
- мощный рендер движок Mantra,
- встроенный инструмент композитинга.

Плюсы: высококлассные спецэффекты и анимация. Минусы: мало информации, высокая цена.

Рендеринг модели решено было выполнить, используя игровой движок, найти правильный движок – не такая простая задача, как может показаться. Технологические платформы, которые уже доказали свою эффективность «в полевых условиях», станут очевидными фаворитами многих разработчиков. Тем не менее, рынок изменчив, а потому вопрос выбора движка своей актуальности в ближайшее время не утратит.

Рассмотрим наиболее распространенные системы.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		41

Unity 3D – отличный доступный движок. У него есть ряд неоспоримых преимуществ перед конкурентами и, пожалуй, ключевое из них в том, что за лицензию вы платите всего один раз. Неважно, насколько популярной становится игра – если вы выбрали Unity, вам не придется тратиться еще раз. С финансовой точки зрения – это удачное решение, в особенности для стартапов и начинающих разработчиков.

Плюсы:

- выгодная лицензионная политика;
- легкость в использовании;
- совместимость с любой платформой;
- отличное комьюнити;
- низкий порог входа;
- популярен среди разработчиков (это означает, что ошибки быстро находят и исправляют).

Минусы:

- ограниченный набор инструментов (вам, скорее всего, придется разработать некоторые из них самим);
- процесс изготовления игры отнимает много времени.

HeroEngine – этот движок хорошо зарекомендовал себя в создании мультиплеерных игр – взять хотя бы Star Wars: The Old Republic. Лицензия довольно дорогая и вряд ли подойдет начинающим разработчикам, но если ваш проект амбициозен, то данный вариант можно рассмотреть.

Плюсы:

- в наличии несколько карт для создания открытого мира. Есть возможность их «бесшовного» соединения;
- богатый AI;
- удобный набор инструментов для моделирования карт;
- подходит для создания комплексных миссий, крафтинга и собирания ресурсов;
- техподдержка осуществляется при помощи сервиса HeroCloud, что весьма удобно.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		42

Минусы:

- скриптовый движок мощный, но неудобный в управлении;
- HeroEngine вместе с сервисом поддержки клиентов HeroCloud слишком дорого стоит и вряд ли будет доступен начинающим разработчикам;
- высокий порог вхождения.

Project Anarchy – этот мощный игровой движок нравится многим разработчикам за наглядную и понятную документацию. Тем не менее, и у него есть свои недостатки.

Плюсы:

- для платформ iOS, Android и Tizen, лицензия бесплатная;
- мощные инструменты для поиска и устранения багов;
- сильное комьюнити;
- издатель предоставляет четкую, понятную документацию и образцы;
- Fmod для аудио-сопровождения;
- мощный Navok AI.

Минусы:

- отсутствует возможность разрабатывать игру на Mac и Linux;
- нет вводного руководства для начинающих разработчиков;
- дорогая лицензия для ПК.

2.1.2 Обоснование выбора среды разработки

Для создания модели было принято решение использовать полнофункциональную профессиональную программную систему для создания и редактирования трёхмерной графики и анимации Autodesk 3ds Max. Рассмотрим ключевые моменты работы с программой.

3ds Max – большой универсальный пакет с огромными возможностями. Описание всех этих возможностей занимает объемные книги, подчас достаточно сложные для освоения, причем многие из возможностей являются весьма специфическими и нормальному человеку, «не фанату» трехмерной графики, вряд ли придет в голову их использовать, даже если он о них будет знать. С другой стороны, за обилием кнопок и команд иногда теряется суть.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		43

Следует учитывать также, что многое из того, что входит в 3ds Max, на сегодняшний момент не актуально. Например, это некоторые атмосферные и линзовые эффекты, такие, как туман, воздушная перспектива, объемный свет, эффекты «смаза» (motion blur) или глубины резкости (ГРИП, DOF). Все эти эффекты могут быть реализованы значительно быстрее в пакетах для постобработки (Photoshop, Combustion, After Effects и т. д.) при этом с качеством, если и уступающим результатам, которые можно получить при рендеринге этих эффектов, то не намного.

При создании 3D Studio MAX (именно так назывался пакет изначально) его создателями, Томом Хадсоном и Гэри Йостом были введены новаторские на тот момент принципы построения пакета. Кратко рассмотрим основные из них.

– Принцип «все в одном». 3D Studio для DOS, который был предтечей 3ds Max, представлял собой пакет из пяти программ, модулей, каждый из которых выполняла свою собственную функцию, объединенных единой оболочкой. Например, создание плоских форм было сосредоточено в модуле 2D Shaper, анимация – 3D Keyframer и т. д. В 3ds Max весь процесс от моделирования до анимации сосредоточен в рамках одного интерфейса, что позволяет оперативно вносить изменения в сцену. К сожалению, все это хорошо при работе с несложным проектом. Некоторое разделение интерфейса на специализированные рабочие пространства для моделирования, анимации и т. п. не помешало бы. Например, так сделано в Autodesk Maya, Autodesk MotionBuilder и др. При желании, возможно сделать такие шаблоны, так как 3ds Max обладает большими возможностями по настройке интерфейса.

– Объектно-ориентированный принцип построения. Основной структурной единицей в 3ds Max является объект. Тип объекта, его набор параметров определяет все возможные операции, которые можно с ним произвести, и набор модификаторов, которые можно применить к этому объекту. Логика работы с объектами весьма жесткая, в отличие от 3D Studio, что дает возможность эффективно работать даже в том случае, когда объектов в сцене много. С другой стороны, эта жесткая логика иногда приводит к излишней усложненности тех или иных действий. Например, невозможно работать на уровне подобъектов сразу с несколькими объектами, не применив предварительно необходимый модификатор, и др.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		44

– Модификаторы и стек модификаторов (Modifiers Stack). Модификаторы – это набор процедур, применяемых к объектам и наделяющих объект новыми свойствами. Аналогом модификаторов можно считать фильтры в пакетах растровой графики, таких, как Adobe Photoshop. Но в отличие от указанных фильтров, модификаторы являются недеструктивными, т. е. позволяют вернуться к исходному объекту в любой момент, так как не приводят к необратимым последствиям (аналогия – фильтры в After Effects или Smart Filters в Adobe Photoshop CS3).

– Модульность и расширяемость. С самого начала 3ds Max построен так, чтобы дать возможность без труда расширить возможности пакета за счет подключаемых модулей (или «плагинов», plugins). Практически, пакет представляет собой ядро и большое количество модулей, входящих в поставку (рисунок 6).

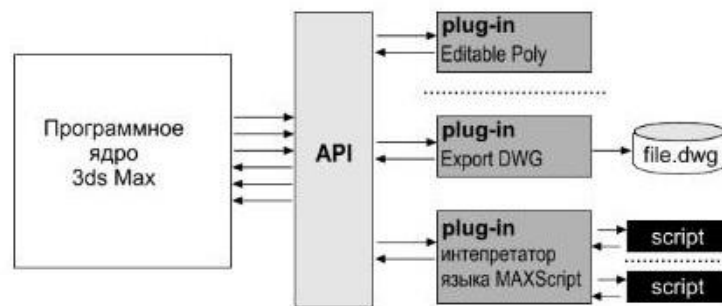


Рисунок 6 – Схема построения 3ds Max

Несмотря на то, что первые версии 3ds Max были далеки от совершенства, 3ds Max завоевал популярность во многом и потому, что в поставке с самого начала присутствовал SDK – Software Development Kit. Благодаря этому уже в начале развития 3ds Max последовал целый вал дополнительных модулей, расширяющих возможности пакета. На сегодняшний момент количество их превышает полутысячу, причем в этом списке присутствуют как большие коммерческие проекты, так и бесплатные разработки, зачастую написанные для внутреннего использования. Многие из них сейчас входят в поставку 3ds Max. К сожалению, такая архитектура имеет и негативные стороны. Чтобы обеспечить совместимость проектов «снизу-вверх», при выпуске каждой новой версии разработчикам приходится перекомпилировать все старые модули. Достаточно медленным является интерпретатор языка MAXScript уже по той причине, что не встроен в ядро, а также является плагином.

Универсальность 3ds Max – это одно из самых важных его достоинств. Но именно это и является одновременно и самым большим недостатком, и не только 3ds Max, но и любого другого универсального пакета. В 3ds Max реализовано очень много методов моделирования, анимации и т. д., но, к сожалению, не всегда лучшим образом. Например, если вас интересует моделирование с помощью NURBS (Non-UniformRationalB-Spline – неоднородные рациональные B-сплайны), то возможно лучшим выбором являются пакет McNeal Rhinoceros или Eovia Amari Pro, полигонов – то Luxology Modo или DAZ Hexagon. Если вы серьезно занимаетесь «ручной» анимацией персонажей, то, возможно, лучшими решениями являются SEGA Animanium и Messiah studio, если используете Мосар (файлы захвата движения) – то Autodesk Motion Builder. Эти пакеты, которые относятся к специализированным, ориентированы на узкий круг задач, но сделано это значительно эффективнее, нежели чем в универсальном пакете.

Такая специализация имеет и негативную сторону. Необходимо позаботиться о совместимости пакетов в технологической цепочке, так как получается ситуация, при которой моделировать приходится в одном пакете, анимировать – в другом, визуализировать – в третьем (кстати, для визуализации очень часто используется как раз 3ds Max в связке с V-Ray). При использовании 3ds Max этих проблем, как правило, нет, так как все сконцентрировано в одном пакете. В результате поливариантности методов моделирования в 3ds Max, вы получаете ситуацию, когда для достижения одной и той же цели можно избрать несколько путей. Например, для создания сиденья скамейки, вы можете использовать примитив Chamfer Box и придать нужную форму при помощи модификатора Bend, а можете воспользоваться сплайном нужной формы и придать ему объем при помощи модификатора Extrude, а закругления сделать при помощи инструмента Chamfer в модификаторе Edit Poly. В качестве текстуры на поверхности вы можете использовать изображение из внешнего файла, а можете применить и настроить процедурную текстуру. Возможно пойти и комбинированным методом – начать моделирование одним методом, например, создав объект на базе лоскутов, а затем доработать его на полигональном уровне. Создать процедурную текстуру, сконвертировать ее в растровый файл, до-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		46

рисовать нужное в редакторе растровой графики и вновь применить его в качестве текстуры в 3ds Max. И таких комбинаций в 3ds Max возможно очень много.

Несмотря на то, что 3ds Max активно применяется для технической и дизайнерской визуализации, следует учитывать, что 3ds Max не конструкторский и не архитектурный пакет. Не стоит требовать от него той точности, к которой, возможно, привыкли, используя пакеты CAD/CAM (Computer Aided Design/Computer-Aided Manufacturing). Основная задача при работе с 3ds Max – это получение красивых и реалистичных картинок или роликов, а отнюдь не конструкторской документации.

С другой стороны, «фамильные черты» линейки продуктов AutoCAD накладывают свой отпечаток, и традиционно 3ds Max в большей степени применяется для технической и архитектурной визуализации, в игровой индустрии, и в меньшей степени для создания видео и кинопродукции, в отличие, например, от Autodesk Maya или Softimage XSI, с которыми ситуация обратная. Но это совершенно не значит, что 3ds Max не подходит для создания спецэффектов для кино, существует несколько студий, как у нас в стране, так и за рубежом, активно использующих 3ds Max для создания «полного метра».

Для визуализации полученной в среде 3ds Max модели был выбран игровой движок Unreal Engine. Приведем некоторые особенности работы с этим движком.

Unreal Engine – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games. Написанный на языке C++, движок позволяет создавать игры для большинства операционных систем и платформ: Microsoft Windows, Linux, Mac OS и Mac OS X; консолей Xbox, Xbox 360, PlayStation 2, PlayStation 3, PSP, PS Vita, Wii, Dreamcast, GameCube и др., а также на различных портативных устройствах, например, устройствах Apple (iPad, iPhone), управляемых системой iOS и прочих. Впервые работа с iOS была представлена в 2009 году, в 2010 году продемонстрирована работа движка на устройстве с системой webOS.

Для упрощения портирования движок использует модульную систему зависимых компонентов; поддерживает различные системы рендеринга (Direct3D, OpenGL, Pixomatic; в ранних версиях: Glide, S3, PowerVR), воспроизведения звука

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		47

(EAX, OpenAL, DirectSound3D; ранее: A3D), средства голосового воспроизведения текста, распознавание речи), модули для работы с сетью и поддержки различных устройств ввода. Для игры по сети поддерживаются технологии Windows Live, Xbox Live, GameSpy и прочие, включая до 64 игроков (клиентов) одновременно. Таким образом, движок адаптировали и для применения в играх жанра MMORPG (один из примеров: Lineage II).

Все элементы игрового движка представлены в виде объектов, имеющих набор характеристик, и класса, который определяет доступные характеристики. В свою очередь, любой класс является «дочерним» классом object. Среди основных классов и объектов можно выделить следующие:

Актёр (actor) – родительский класс, содержащий все объекты, которые имеют отношение к игровому процессу и имеют пространственные координаты.

Пешка (pawn) – физическая модель игрока или объекта, управляемого искусственным интеллектом. Название происходит от англ. pawn — тот, кем манипулируют (или пешка, поэтому такой объект без какой-либо модели выглядит как пешка). Метод управления описан специальным объектом, такой объект называется контроллером. Контроллер искусственного интеллекта описывает лишь общее поведение пешки во время игрового процесса, а такие параметры как «здоровье» (количество повреждений, после которых пешка перестаёт функционировать) или, например, расстояние, на котором пешка обращает внимание на звуки, задаются для каждого объекта отдельно.

Мир, уровень (world, game level) – объект, характеризующий общие свойства «пространства», например, силу тяжести и туман, в котором располагаются все актёры. Также может содержать в себе параметры игрового процесса, как, например, игровой режим, для которого предназначен уровень.

Для работы с простыми и, как правило, неподвижными элементами игрового пространства (например, стены) используется двоичное разбиение пространства – всё пространство делится на «заполненное» и «пустое». В «пустой» части пространства располагаются все объекты, а также только в ней может находиться «точка наблюдения» при отрисовке сцены. Возможность полного или частичного

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		48

помещения объектов в «заполненную» часть пространства не исключается, однако может привести к неправильной обработке таких объектов (например, расчёт физического взаимодействия) или неправильной отрисовки в случае помещения туда «точки наблюдения» (например, эффект «зала зеркал»). Все пешки, попадающие в «заполненную» часть пространства, сразу «погибают».

Зонирование. В камеру не попадает ни один портал (пунктирная линия) красной зоны, поэтому объекты в ней не обрабатываются вовсе.

Поверхность (surface) является основным элементом двоичного дерева пространства. Эти элементы создаются на грани пересечения между «заполненной» и «пустой» частями пространства. Группа элементов двоичного дерева пространства называется **нодом (node, рус. узел)**. Этот термин, как правило, употребляется в контексте **node count** – количество нодов на экране или в игровом пространстве вообще. Количество нодов, одновременно видимых на экране влияет на производительность при прорисовке сцены. Если какой-то нод не попадает на экран или перекрывается целиком другими нодами, он не обсчитывается – это служит для повышения производительности, особенно в закрытых пространствах. Разбиение всего пространства на группы нодов называется **зонированием**.

Для этого иногда используются порталы – невидимые поверхности, которые служат для того чтобы вручную разделить крупный нод на два меньших (в версии движка Unreal Engine 3 ввели поддержку аддитивной геометрии, что позволило отказаться от зонирования). Кроме порталов используются антипорталы.

Описание «заполненных» и «пустых» частей пространства выполняется с помощью набора замкнутых трёхмерных объектов, составленных из не пересекающихся поверхностей – **брашей (brush, рус. кисть)**. Этот принцип построения пространства называется **конструктивной сплошной геометрией**. Геометрия может быть «аддитивной» (всё пространство изначально «пустое») и «вычитательной» (изначально заполненное материей пространство).

Браши делятся на три типа:

– **сплошные (solid)** – полноценно участвуют в двоичном разбиении пространства.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		49

- аддитивные (additive) – «заполняют» двоичное пространство.
- вычитательные (subtractive) – «вырезают» объёмы в пространстве.
- полу-сплошные (semi-solid) – не влияют напрямую на двоичное дерево пространства, однако влияют на её физическую модель. Могут только «заполнять» пространство. Служат для создания «невидимых» препятствий, а также снижения числа полигонов и нодов.

- пустые (non-solid) – только создают поверхности, не влияют на двоичное дерево пространства. Используются преимущественно для создания объёмов (volume) – часть пространства, которая обладает свойствами, отличными от свойств игрового мира. Объёмы имеют приоритет, свойства объёма с большим приоритетом применяются к находящимся в нём актёрам. Игровой мир всегда имеет минимальный приоритет. При помощи объёмов можно изменить гравитацию, вязкость, туман и тому подобное. Объёмы, начиная с версии движка Unreal Engine 2, используются для создания воды (но не водной поверхности). Движок комплектуется инструментарием (SDK), в том числе и редактором UnrealEd.

Несмотря на кажущуюся сложность работы, редактор Unreal Editor 4 выглядит приветливым. А с ростом производительности компьютерного оборудования GPU рендер может изменить традиционный подход к работе визуализаторов, дизайнеров и архитекторов.

2.2 Примеры аналогичных программных продуктов

В качестве примера из существующих, аналогичных программных продуктов можно выделить следующие:

- Модели созданные в рамках программы компас-3D. А в частности модель двигателя ЯМЗ-238 (рисунок 7), устанавливаемого на автокран Урал-4320.

Данная система, имея большое количество положительных сторон таких как: очень легка в освоении, причем даже для конструкторов, не имеющих опыт общения с 3D редакторами; имеет большое количество библиотек элементов стандартизированных по ГОСТ; является продуктом отечественных разработчиков, а потому не существует никаких проблем с ее локализацией; несложный для обучения и довольно удобный интерфейс, обладает двумя существенными недостатками:

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		50

- система платная;
- возникают проблемы при импорте/экспорте моделей из других САПР.

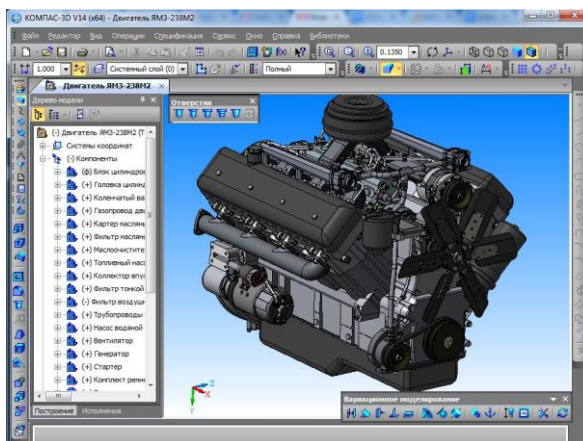


Рисунок 7 – Пример модели, выполненной в программе компас-3D

– «Мультимедийное руководство по ремонту и обслуживанию автомобиля Ваз 2101». Данный программный продукт обладает следующими особенностями: материал четко структурирован, обширное руководство по эксплуатации автомобиля; сотни иллюстраций, показывающих органы управления и отдельные этапы работ; понятный, удобный интерфейс (рисунок 8).



Рисунок 8 – Интерфейс программы «Мультимедийное руководство по ремонту и обслуживанию автомобиля Ваз 2101»

При всех преимуществах программы можно выделить один недостаток, наглядная трехмерная модель выполнена только экстерьера автомобиля, а все детали и технические особенности представлены в виде иллюстраций, что не совсем информативно.

– Модели созданные в рамках программы AutoCAD. А в частности модель двигателя автогрузовика КамАЗ (рисунок 9).

					ВКР.145366.090401.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

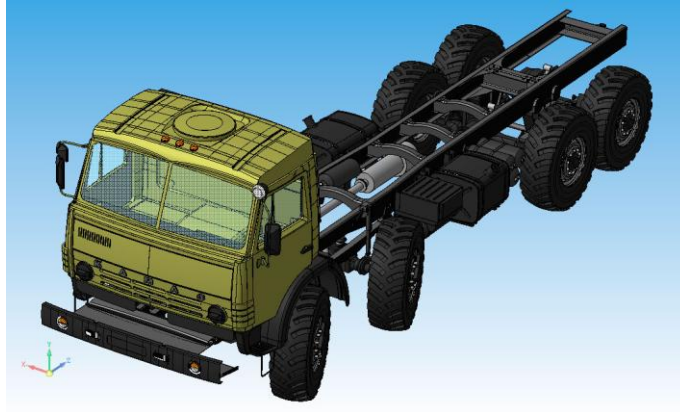


Рисунок 9 – Пример модели, выполненной в программе AutoCAD

Итак, к ряду несомненных достоинств построения трехмерных моделей в AutoCAD можно отнести:

- удобство работы с такой моделью: вы можете рассмотреть ее со всех сторон и с любой стороны;
- в автокаде есть функция автоматической генерации на плоскость основных и дополнительных видов;
- есть возможность построения сечений на плоскость;
- благодаря подавлению скрытых линий и реалистичному тонированию, вы получаете отличное изображение;
- можно проверить взаимодействия;
- вы можете экспортировать модель, сделанную в АвтоКАДе в анимированное приложение;
- в AutoCAD есть возможность проведения инженерного анализа.

Недостатки:

- высокие системные требования;
- дорогая цена за профессиональное издание.

Autodesk является таким же крупным игроком на рынке софта, как и Adobe или Corel. Поэтому выход любой новой версии AutoCAD воспринимается как нечто должное и привычное. Очень удобно использовать ее для чертежа моделей (2D и 3D). Благодаря мощному функционалу здесь очень легко сделать сечение объекта в соответствии с правилами начертательной геометрии. Из-за высокой стоимости программы большинство людей знакомится с AutoCAD на практических занятиях в

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		52

ВУЗе. Но это свойственно всем профессиональным программам. Отличий в 2013 не так уж и много. В основном они касаются облачных сервисов, а весь инструментарий и интерфейс остался на месте.

2.3 Построение решения задачи

На данный момент все технические разработки создаются в рамках САПР (система автоматизированного проектирования). Все эти системы разрабатываются в виде приложения к таким программным продуктам как AutoCAD и работают с учетом установленного программного обеспечения, что не всегда рентабельно. С другой стороны, принимая во внимание наглядность разрабатываемой модели, можно создавать необходимый программный продукт, вскрывающий техническое устройство систем автомобиля, необходимых для преподавания тех или иных дисциплин. Существующие модели максимально детализированы. С одной стороны, это удобно, но информация становится избыточной и как следствие программный продукт становится дорогостоящим. Альтернативным решением упрощения данной ситуации является создание высоконаглядных детализированных трёхмерных моделей, которые будут визуализироваться с помощью игрового графического процессора Unreal Engine.

На сегодняшний день компьютерное моделирование является одним из актуальных направлений для изучения общего технического устройства автомобильной техники. Общее положение дел имеющихся компьютерных моделей сводится к тому, что модели являются избыточными по наличию информации и дорогостоящими. Поэтому целесообразно использовать более доступные программные средства, для построения менее детализированных, но не уступающих по объёму информации компьютерных моделей.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		53

3 ПОДРОБНОЕ ОПИСАНИЕ РАЗРАБОТАННОГО ПРОГРАММНОГО ПРОДУКТА

3.1 Описание алгоритмических и кодовых процедур реализации программы

3.3.1 Описание создания модели

Создание модели начинаем с поиска чертежей автомобиля. Хороших чертежей мало. В основном попадаются подделки. Некоторые из них можно отличить на глаз. Обычно такие подделки можно определить по второй паре колес или по виду спереди. Несоответствие с оригиналом может быть очень заметно. К тому же работать по таким чертежам очень сложно. Обязательно находим фотографии автомобиля в разных ракурсах. А также фото различных частей автомобиля крупным планом. Например, салон, кресла, диски, а главное фотографии фар. Все эти детали на чертежах изображены очень плохо. Поэтому, чем больше разрешение фотографий, тем лучше. Прежде чем начать моделирование нужно запомнить – периодически, на разных этапах, сохранять работу в отдельный файл. К примеру, если испортили какую-нибудь деталь. В таком случае эту деталь не обязательно делать заново или исправлять, её можно взять из предыдущего сохранения. Открываем меню file, выбираем merge, жмём на предыдущее сохранение, в появившемся окне выбираем эту деталь. И разумеется нельзя забывать про hold – аналог возврата действия. Перед тем как сделать какое-нибудь сомнительное действие, жмём edit/hold, и если действие, которое задумывали, нас разочаровало то, тогда опять открываем edit и жмём fetch, который вернёт нас на то место, где нажали hold. Кстати hold можно открыть как сохранение. В той же папке autoback жмём на файл с именем maxhold.mx.

Теперь можно приступать к моделированию. Начинаем с расстановки чертежей, лучше всего работать с прозрачными (рисунок 10):

- 1) В фотошопе меняем цвета и запоминаем размеры чертежей;
- 2) Делаем box по размерам чертежей;
- 3) Удаляем две плоскости;
- 4) Выделяем остальные плоскости, нажимаем flip normalis;

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		54

3) Ставим галочку «Do not show this message again»;

4) Во всплывающем окне выбираем «connect only».

Затем применяем модификатор surface. Обязательно treshold переставляем на ноль, что бы он не сваривал близлежащие точки. Steps тоже переставляем на ноль. Затем следует модификатор symmetry. Иногда он может размещать центр не там где надо. В таком случае поможет инструмент Align, который совмещает центры объектов.

Следующим шагом нажимаем на Mirror в symmetry, а потом в икс вводим ноль. Сразу же добавляем и MeshSmooth. В нём во вкладке subdivision amount в первом Iterations всегда ставим двойку. Iterations это количество сглаживаний, по этому нельзя ставить туда какие-то большие цифры. Жмём на последнем модификаторе правую кнопку и выбираем Collapse all, сворачиваем line и surface. После этого увидим модификатор editable poly, в котором и будет проходить вся работа. В editable poly во вкладке polygon properties отключаем первую группу сглаживания. Начинаем проводить дополнительные линии и вырисовывать на кузове все подробности. Лини тянуть можно с помощью кнопки Cut в editable poly. Можно провести прямую линию кнопкой quick Slice. Образовавшаяся линия не выйдет за пределы выделенных полигонов. Избавляться от линий можно с помощью кнопки target Weld, которая присоединит точку к одной из соседних. Ещё можно выделить линию и нажать кнопку remove. Только после такого удаления, останутся точки которые лежали на этой линии. После того как выделили линию, жмём правую кнопку, выбираем convert to vertex, а потом уже возвращаемся в режим редактирования линий и удаляем эту линию. Потом заходим в режим редактирования точек. Видим, что точки после удаления линии остались выделенными, и остаётся только нажать remove.

Таким образом формируем более или менее правильную сетку. И пока что приблизительно выравниваем весь кузов по чертежам. Группы сглаживания – это когда одним полигонам присваивается одна группа, а вторым другая. Линия которая является границей разделяющей эти группы, после сглаживания останется острой (рисунок 12).

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		56

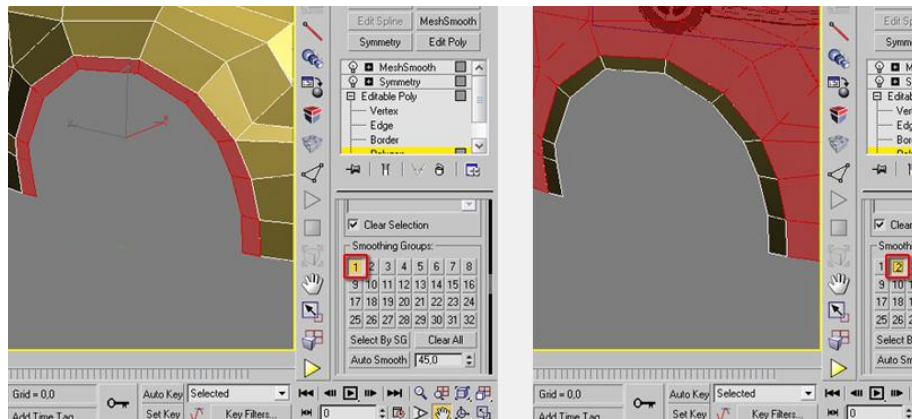


Рисунок 12 – Группы сглаживания

Но использовать группы повсюду нельзя. Во-первых, такая линия будет идеально острой. А во-вторых, часто острые линии на кузове должны плавно исчезать, то есть переходить в плоскость. Способ с группами сглаживания в таком случае не годится. Чтобы линия после mesh smooth осталась острой, обычно её раздваивают с помощью Chamfer. Делаем Chamfer не меньше чем 0,03, а то с линией будет неудобно работать. Чем ближе соседние линии, тем острее будет раздвоенная. После того, как раздвоили все необходимые линии, начинаем окончательно формировать сетку.

После того, как определились со всеми линиями, начинаем окончательно выравнивать кузов. Некоторые места в автомобиле могут состоять из абсолютно прямых линий. Их можно выровнять с помощью кнопок X, Y, Z в editable poly (рисунок 13). Если, во вкладке Edit Geometry напротив constraints в выпадающем меню выбрать режим edge, то точка будет двигаться строго по линиям, которые от неё отходят.

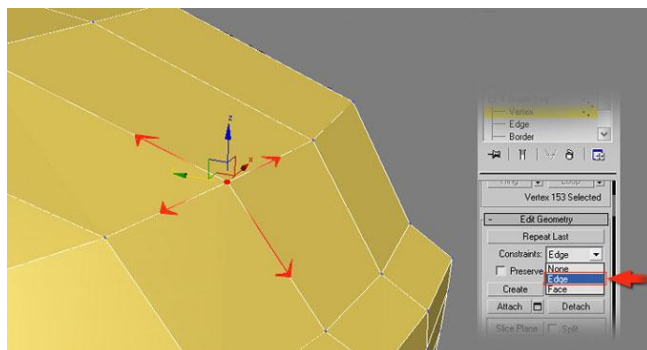


Рисунок 13 – Выравнивание кузова

В constraints выбираем edge, выделяем все точки этой линии, и просто тянем их к соседней линии до тех пор, пока все точки не сравняются со своими соседями,

а потом обязательно отпускаем левую кнопку мыши. После такого действия, точки сбрасывают информацию о изначальном своём расположении. Дальше, не снимая выделения с этих точек, тянем их обратно вверх. Линия выровнялась за счёт соседней, и теперь абсолютно параллельна ей.

При помощи `shape merge` можно отделить все – стёкла, решётки, фары и некоторые другие детали. Но, есть определённые правила. Например, `Shape merge` не работает с не замкнутым сплайном. Также нельзя разрывать замкнутый сплайн.. Ещё если повернуть сплайн, `shape merge` выдаст неправильный результат. Если не понравилось расположение контура на кузове, его можно легко переместить. Чтобы кузов окончательно был готов, осталось сделать только швы. Для этого, используем сплайны и `boolean`. Он не заменим в случаях, когда нужно сделать дырку или углубление в виде сложной формы. `Boolean` не работает после режима `Cut`. То есть, если использовали в `boolean` режим `Cut`, то второй раз `boolean` с этим объектом работать будет не правильно. Ещё `boolean` не распознаёт сплайны. Имеется ввиду сплайн настройки которого переставили так, чтобы он имел вид трубки. Чтобы `boolean` с ним работал, сплайн нужно конвертировать в `editable poly` или применить к нему модификатор `edit poly`. Теперь можно приступить к швам. Для этого сплайнами выкладываем линии которые повторяют линии швов по всем четырём чертежам (рисунок 14).

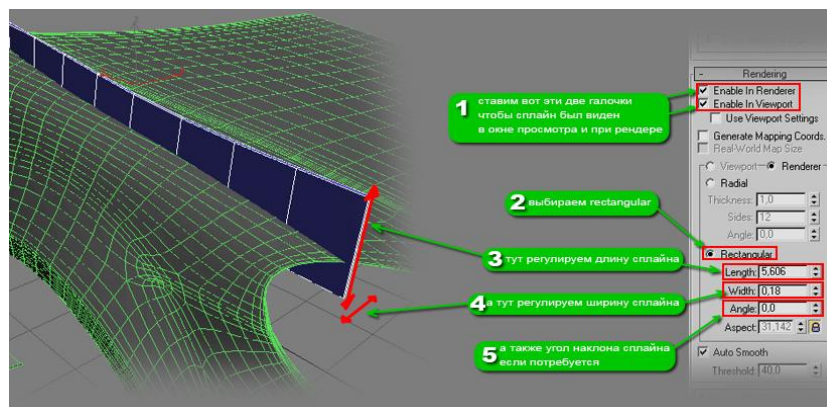


Рисунок 14 – Настройки сплайнов

В этих сплайнах меняем вот эти настройки:

- 1) Выбираем пункты «Enable in Renderer» и «Enable in Viewport»;
- 2) Выбираем «Rectangular»;
- 3) Регулируем длину сплайна;

					<i>Лист</i>
					ВКР.145366.090401.ПЗ
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	58

- 4) Регулируем ширину сплайна;
- 5) Регулируем угол наклона сплайна.

Все сплайны размещаются так, чтобы они наполовину заходили в кузов. Необходимо сначала сделать сплайн только для одного шва и сразу определиться с его толщиной. Таким образом, все следующие сделанные сплайны будут такой же толщины как и первый, и не надо будет выставлять толщину отдельно у каждого. То есть каждый шов – это отдельный сплайн. В некоторых сплайнах во вкладке interpolation можно значение steps сделать меньше шести. Так просто сплайны остаются на кузове меньше точек, и вероятность появления различных артефактов уменьшается.

Затем конвертируем один из сплайнов в editable poly и присоединяем к нему с помощью attach остальные сплайны. Перед тем, как приступить к boolean, нажимаем на всякий случай HOLD. Теперь нажимаем на наш кузов, заходим в boolean, выбираем в нём режим Cut/refine, жмём на кнопку pick operand b, и нажимаем на наши сплайны. Потом конвертируем в editable poly. Зайдя в нём в режим редактирования полигонов, мы увидим, что все полигоны внутри швов выделились сами.

Также в некоторых местах могут быть большие скопления ненужных точек. Для этого мы, не снимая выделения с полигонов, нажимаем правую кнопку, выбираем convert to vertex, и видим, как полигоны конвертировались в точки. После этого применяем Weld со значением 0,01. Возвращаемся в режим редактирования полигонов. Полигоны швов по прежнему выделены. Дальше делаем следующее.

- 1) Применяем «Bevel Polygons», для сглаживания краев швов (рисунок 15);

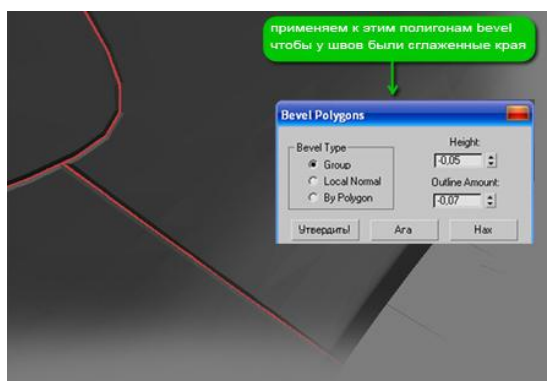


Рисунок 15 – Применение «Bevel polygons»

- 2) Применяем «Extrude Polygons» для углубления полигонов (рисунок 16).

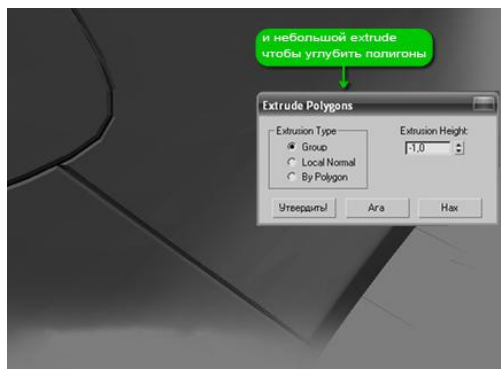


Рисунок 16 – Применение «Extrude Polygons»

На этом моделинг кузова закончен. Далее идёт моделинг всех остальных деталей. Но перед тем как начать делать, определимся с фоном, ракурсом и настройками камеры. Например в некоторых ракурсах могут быть не видны задние кресла, руль, внутренняя часть двери и, возможно, некоторые другие детали салона. То есть, сначала ставим фоновое изображение на вид из камеры. Скрываем стёкла и делаем рендер. Смотрим какие части салона видны, а какие нет. Почти всегда на чертежах не обозначены кресла, по этому эту деталь салона придётся делать на глаз из примитива box. Очень сложно угадать с пропорциями кресел, для этого необходимо найти фотографии салона.

2.3.2 Визуализация модели

Первым шагом при экспорте будет подготовка автомобиля, потому как можно столкнуться с рядом проблем в работе физики автомобиля из-за неправильного экспорта. 3ds Max использует правостороннюю систему координат с осью Z, направленной вверх. Такую систему координат используют почти все САПР (рисунок 17).

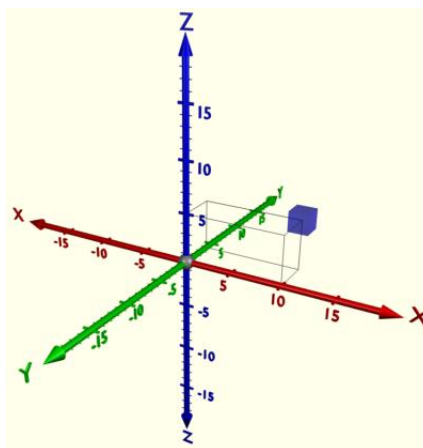


Рисунок 17 – Система координат 3ds Max

					ВКР.145366.090401.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

Unreal Engine как почти любой другой игровой движок использует левостороннюю систему координат. Когда Y направлен вверх, X – вбок, Z- вперед (рисунок 18).



Рисунок 18 – Система координат Unreal Engine

Следовательно сначала для правильного экспорта автомобиля, мы должны понять, что Z+ это будет направление автомобиля движения вперед, (Z-) – направление когда автомобиль движется назад. (X+ -) это направления поворота вбок – но это в Unreal Engine, которая использует левостороннюю систему координат. А в правосторонней системе координат 3ds Max Y+ это будет направление когда автомобиль смотрит вперед и (X+ -) это направления вбок. Все параметры вращения должны стоять на 0 для дальнейшей работы и для правильной генерации Меша (Mesh), надо Scale установить по умолчанию. Для этого мы сделаем следующие действия в 3ds Max:

- 1) Выделим все объекты в сцене (клавиша A);
- 2) Object>Apply>Rotate & Scale.

Теперь все параметры вращения приняты и установлены на 0. А параметры масштабирования (Scale) установлены на по умолчанию.

Теперь можем делать «правильный экспорт» из правосторонней системы координат в левостороннюю систему координат.

- 1) Выделим все объекты (клавиша A);
- 2) Жмем последовательность R>X>-90;
- 3) Object>Apply>Rotate;
- 4) Жмем последовательность R>X>90.

Обязательно делаем в данной последовательности. Далее экспортируем модель в Engine. File>Export>FBX.

После экспорта модели, идем в папку с этой моделью и копируем папку Textures (папку с текстурами) и наш FBX файл. В папку Assets нашего проекта. Это позволит Engine импортировать модель прямо с текстурами, которые были настроены в 3ds Max.

Далее настроим автомобиль. Создадим Prefab и добавляем объект в Prefab:

- 1) Вкладка Assets>Create>Prefab;
- 2) Переименуем new prefab в Car;
- 3) Перетащим методом D&D наш Ассет Ural_4320_mesh в наш Prefab с именем Car;
- 4) Перетащим методом D&D наш Prefab с именем Car на сцену.

Теперь добавим физику автомобилю.

- 1) Выделяем наш объект – автомобиль (Car).
- 2) Перейдем на вкладку Inspector нажмем кнопку Add component> Physics> Rigidbody.
- 3) Заходим в компонент Rigidbody и делаем настройки (желательно более приближено к оригинальной модели).

В нашем случае полная масса автомобиля 15205 кг. Mass=15205. Все остальные характеристики ставим на 0, т.к. они отвечают за сопротивление. А за сопротивление у нас будет отвечать отдельный скрипт «AerodynamicResistance.cs».

Настройки Rigidbody выставляем следующими (рисунок 19):

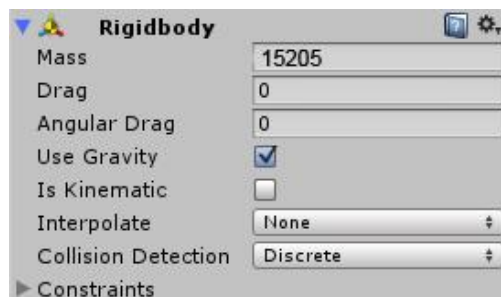


Рисунок 19 – Настройки Rigidbody

Далее добавим тени. Направленный свет сделает у автомобиля хорошую blob тень на дороге. Для этого используем проектор, который проецирует Blob тень, на-

поминающую форму автомобиля на дороге непосредственно под машиной. Проектор встроенный в компонент, который работает так же, как настоящий проектор. Указываем текстуру, которую хотим передавать, и на основе настроек для проектора и (расстояния до цели), тень будет опираться на те текстуры объектов, которые находятся на пути света проектора.

- В иерархии, создайте пустой игровой объект и перетащите его на автомобиль «Car», чтобы сделать его дочерним объектом, игрового объекта «Car»;
- Задаем имя объекту «Blob shadow projector»;
- Добавим компонент (Projector) проектора на Blob тень этого проектора (Component->Renderer->Projector);
- Установим на проекторах настройки Near Clip Plane = 0,1, Far Clip Plane = 50, Field of View = 30;
- Назначаем материал Blob_shadow для слота материала;
- В «Ignore Layers» выберите «Everything», а затем снимите флажок со слоя «Road», в результате чего проектор отбросит Blob тень на дорогу;
- Добавляем скрипт «BlobShadowController.js» (в папке Scripts/CSharpScripts в окне проекта)

Положение и вращение компонента (Projector) проектора обновляется каждый кадр в BlobShadowController.js скрипте который является довольно простым. Короче говоря, он находится в 10 метрах над автомобилем и получает его координаты вращения на основе вращения машины. Есть возможность взглянуть на скрипт и попытаться изменить значение, если хотим что бы Blob тень автомобиля на дороге выглядела иначе.

Создадим центр тяжести:

- 1) Правый клик мыши по Prefabs автомобиля > Create Empty;
- 2) Переименовываем пустой объект с именем GameObject на CoG;
- 3) Задаем координаты в компоненте Transform объекта CoG.

Центр тяжести примерно должен быть впереди чуть выше дна автомобиля, но в принципе для каждого автомобиля индивидуально настраивается. Всё это будет отлаживаться во время тестирования.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		63

Завершаем настройку автомобиля добавлением скриптов (рисунок 20):

- 1) Выделяем prefab автомобиля;
- 2) На вкладке inspector жмем кнопку Add Component>Script>Car Controller;
Добавится скрипт CarController и зависимый от него скрипт Drivetrain
- 3) Нажимаем кнопку Add Component>Script>Sound Controller;
- 4) Нажимаем кнопку Add Component>Script>Anti Roll bar;
- 6) Нажимаем кнопку Add Component>Script>Aerodynamic Resistance;
- 7) Нажимаем кнопку Add Component>Script>Traction Helper.

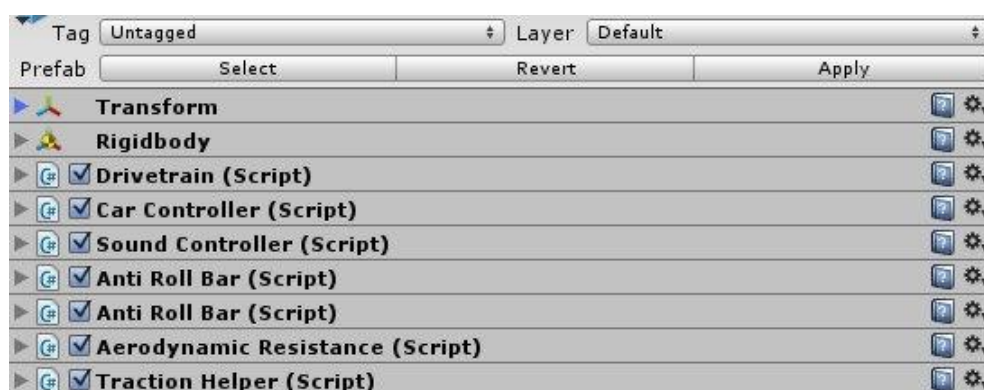


Рисунок 20 – Добавление скриптов

Последняя часть, делаем настройки камеры:

- 1) Нажимаем кнопку Add Component>Script>Car Camera.

Делаем следующую настройку, как показано на рисунке 21:

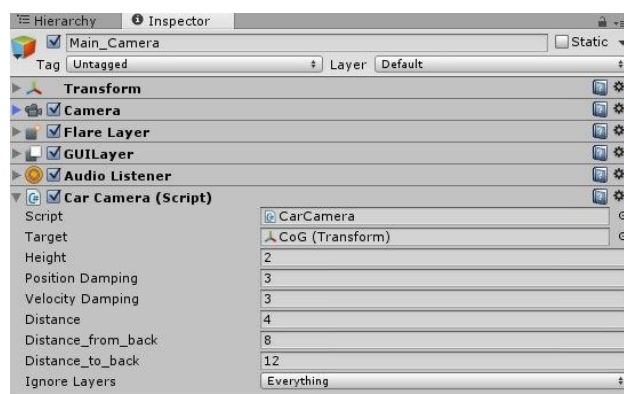


Рисунок 21 – Настройка камеры

- Target – объект, на который будет ориентироваться камера;
- Height – высота камеры;
- Distance_to_back – дистанции от объекта до камеры;
- Distance_from_back – дистанции от объекта до камеры после разгона автомобиля (эффект приближения).

2.3.3 Программная реализация

Программа была написана на языке программирования Action Script 2 с реализацией графического пользовательского интерфейса. Для начала создадим основной запускаящий файл, где собственно весь интерфейс и управление. В нем делаем фон, надписи, кнопки, клипы со списками чертежей и описаний, задается прозрачность деталям. Каждой кнопке прописываем действия – изменяем значение переменной видимости детали на заданной значение – 0, 15 %, и так далее.

```
on(release){
this._visible=0;
_root.part=11;
_parent.ref.gotoAndStop("p11");
//_parent.ref._visible=1;
_root.ramaV=100;//прозрачность рамы
_root.kolesaV=30;
_root.ressoraV=30;
_root.kabinaV=30;
_root.rulevoeV=15;
_root.motorV=15;
_root.transmisV=15;
_root.vyhlopV=15;
_root.strelaV=15;
_root.kranKrukV=15;
_root.kranBaseV=15;
_root.kranPlatformaV=15;
_root.kranKabinaV=15;
_root.oporaV=15;
_root.kranStoykaV=15;
_root.diskiV=0; //торможные диски
_root.tormoznayaV=0;//тормозная система
_root.rulMehanizmV=0;
```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		65

```

_root.rulKardanV=0;
_root.barankaV=0;
_root.pnevmoUsiliteI=0;
_root.kompressorV=0;
_root.ballonsV=0;
_root.ohlagdenieV=0;
_root.smazkaV=0;
_root.toplivoV=0;
_root.dvsV=0;
_root.kardansV=0;
_root.korobkaPeredachV=0;
_root.mostyV=0;
_root.razdatochnayaV=0;}

```

Эти переменные в руте, то есть в корне программы. Они будут видимы всем процедурам, в том числе если эти процедуры в загруженном файле с деталями прописаны. Плюс в каждой кнопке строка изменяет координаты шарика, так чтобы он указывал на нужное значение видимости детали. Для каждой детали своя копия шарика со своим именем объекта.

```

main.tutors._visible=0;
xStep=50;
yStep=50;
main.plakats._visible=0;
angle1=3;
rulevoeV=100;
motorV=100;
transmisV=100;
vyhlopV=100;
strelaV=100;
kranKrukV=100;
kranBaseV=100;

```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		66

```

kranPlatformaV=100;
kranKabinaV=100;
oporaV=100;
kranStoykaV=100;
diskiV=0;
tormoznayaV=0;
_root.rulMehanizmV=0;
_root.rulKardanV=0;
_root.barankaV=0;
_root.pnevmoUsiliteIV=0;
_root.kompressorV=0;
_root.ballonsV=0;
_root.ohlagdenieV=0;
_root.smazkaV=0;
_root.toplivoV=0;
_root.dvsV=0;
_root.kardansV=0;
_root.korobkaPeredachV=0;
_root.mostyV=0;
_root.razdatochnayaV=0;

```

В меню описаний и чертежей на каждом элементе списка делаем прозрачную кнопку, которая загружает в клип растровую картинку с чертежом или текстовым описанием соответственно. Загружаем растровые картинки одной командой с указанием имени файла и места назначения. В данном случае имеем для чертежей свой клип, для описаний свой. Такой подход удобнее против включения изображений в файл программы, поскольку экономит размеры файла и позволяет легко изменить сами картинки в любой момент, сохраняя их имена.

```

loadMovie("detalsLayers5_1-9.swf", main.ural);
loadMovie("detalsLayers5_10-21.swf", main.ural2);
loadMovie("detalsLayers5_22-24.swf", main.ural3);

```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		67

```

loadMovie("detalsLayers5_sub.swf", main.ural4);
main.ural._x=-100;
main.ural._y=-300;
main.ural._xscale=70;
main.ural._yscale=70;
main.ural2._x=-100;
main.ural2._y=-300;
main.ural2._xscale=70;
main.ural2._yscale=70;
main.ural3._x=-100;
main.ural3._y=-300;
main.ural3._xscale=70;
main.ural3._yscale=70;
main.ural4._x=-100;
main.ural4._y=-300;
main.ural4._xscale=70;
main.ural4._yscale=70;

```

Подготавливаем картинки чертежей и описаний. Выбираем некий предельный размер, чтобы картинка примерно умещалась в экран приложения. В данном случае для плакатов это около тысячи пикселей. Некоторые до 1200 доходят. В общем делается исходя из того, чтобы на картинке можно было нормально все разглядеть и прочитать, но чтобы она не сильно вылезала за экран и было проще ее двигать. По туторам технология отдельная. Там хранятся исходные тексты. Они переводятся в jpg следующим образом – сначала делается экспорт текста в PDF формат. То есть как бы «печать в файл». Далее этот pdf можно загрузить в фотешоп, там на входе некоторые настройки просит для растеризации, размер картинок, какие страницы выбрать и т.п. В итоге в зависимости от размера текста получается несколько файлов-страниц. Первый файл увеличиваем размер поля чтобы можно было добавить другие страницы. Скидываем все страницы и размещаем по полю до трех-четырёх рядов и столбцов. Затем после выравнивания все в один слой и смот-

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		68

рим чтобы не выходило за пределы 3200 пикселей. Если больше, то просто уменьшается пропорционально. В среднем получается файлы до 2000-2400 пикселей по большей стороне. Некоторые до 3200. Это видно в папке tutors. В основном файле для перетаскивания плакатов и туторов по экрану делаем для клипа Drag-and-drop технологию. Для этого просто делается кнопка прозрачная поверх всего поля клипа и там прописываются две строки на событие нажатия и отпускания кнопки соответственно команды `startDrag()` и `stopDrag()`; При этом клипы с плакатами и описаниями должны находиться в самом верхнем слое перекрывающем все остальные. Кодовая реализация представлена в приложении А.

Аналогично делаем слой для изображения машины, где так же делаем невидимую кнопку и записываем несколько строк программы для вращения и масштабирования машины. Логика примерно такова – при нажатии кнопки фиксируются координаты курсора мыши в глобальных переменных. При событии движения мышки считаем произошло ли достаточное изменение от предыдущего. Если скажем оно больше 100 пикселей, или 200, подбирается экспериментально, то делается изменение значения угла поворота модели, или ее масштаба, если движение по вертикали мышью. Это тоже все в глобальных переменных. Если там достигнут максимальный угол, то скидываем его на ноль и так далее. Если вращение в другую сторону, то вместо прибавления вычитание.

```
function rotateRight(){
  if(angle1<11){
    angle1+=1;
  }else{
    angle1=1; }
  //main.ural.wheels.gotoAndStop(angle1);
  // main.ural.rama.gotoAndStop(angle1);
  //main.ural.kabina.gotoAndStop(angle1);
  //main.ural.kran.gotoAndStop(angle1);
  main.ural.gotoAndStop(angle1);}
function rotateLeft(){
```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		69

```

if(angle1>1){
angle1-=1;
}else{
angle1=11; }
//main.ural.wheels.gotoAndStop(angle1);
// main.ural.rama.gotoAndStop(angle1);
// main.ural.kabina.gotoAndStop(angle1);
//main.ural.kran.gotoAndStop(angle1);
main.ural.gotoAndStop(angle1);}

```

Далее выравниваем в слоях все загруженные детали. Для этого в нижний слой для референса подгружаем изображение всего крана. По нему с точностью до пикселя двигаем и ровняем все элементы. Далее смотрим перекрытие по слоям деталей в зависимости от угла поворота. К сожалению, недостаточно просто иметь для каждой детали отдельный слой. Приходится некоторые из их слоев перемещать в другие и там элементы дополнительно устанавливаются по слоям уже через сортировку расположения обычным путем как в других многих – выше других элементов, ниже, в низ слоя и т.п. На каждом ракурсе нужно контролировать все элементы. Каждая деталь в каждом кадре при этом превращается в клип из просто растровой картинки.

```

sm=150;
xx=_xmouse;
yy=_ymouse;
if(_root.mDown==1){
dx=xx-lastX;
dy=yy-lastY;
if(Math.abs(dx)>xStep){
if(dx<0){
rotateRight();}
if(dx>0){
rotateLeft();}

```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		70

```

lastX=xx;}
if(Math.abs(dy)>yStep){
if(dy<0 and main.ural._xscale<300){
main.ural._xscale+=20;
main.ural._yscale+=20;
main.ural._x-=sm;
main.ural._y-=sm;
main.ural2._xscale+=20;
main.ural2._yscale+=20;
main.ural2._x-=sm;
main.ural2._y-=sm;
main.ural3._xscale+=20;
main.ural3._yscale+=20;
main.ural3._x-=sm;
main.ural3._y-=sm;
main.ural4._xscale+=20;
main.ural4._yscale+=20;
main.ural4._x-=sm;
main.ural4._y-=sm;}
if(dy>0 and main.ural._xscale>20){
main.ural._xscale-=20;
main.ural._yscale-=20;
main.ural._x+=sm;
main.ural._y+=sm;
main.ural2._xscale-=20;
main.ural2._yscale-=20;
main.ural2._x+=sm;
main.ural2._y+=sm;
main.ural3._xscale-=20;
main.ural3._yscale-=20;

```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		71

```

main.ural3._x+=sm;
main.ural3._y+=sm;
main.ural4._xscale-=20;
main.ural4._yscale-=20;
main.ural4._x+=sm;
main.ural4._y+=sm;}
lastY=yy;}}

```

Чтобы контролировать видимость деталей, для каждой из них надо прописать контролирующий код. Во-первых, каждую деталь приписываем к одной из шести групп видимости, которые в нашем приложении существуют. Это шесть значений переменной, заведующей группами. Далее в клипе каждой детали делается замкнутый по циклу код, который в каждом кадре проверяет какое значение видимости в процентах для деталей этой группы, а точнее просто устанавливает для данного клипа детали это значение видимости. от 0 до 100. Нужно для каждой детали в каждом кадре в клип скопировать нужный код. В итоге в процессе работы каждая деталь 12 раз в секунду проверяет свое состояние и обновляет свое значение видимости-прозрачности.

```

sm=150;
xx=_xmouse;
yy=_ymouse;
if(_root.mDown==1){
dx=xx-lastX;
dy=yy-lastY;
if(Math.abs(dx)>xStep){
if(dx<0){
rotateRight();}
if(dx>0){
rotateLeft();}
lastX=xx;}
if(Math.abs(dy)>yStep){

```

					ВКР.145366.090401.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		72


```

if(dy<0 and main.ural._xscale<300){
main.ural._xscale+=20;
main.ural._yscale+=20;
main.ural._x-=sm;
main.ural._y-=sm;}
if(dy>0 and main.ural._xscale>20){
main.ural._xscale-=20;
main.ural._yscale-=20;
main.ural._x+=sm;
main.ural._y+=sm;}
lastY=yy;}}

```

Для того чтобы при переходе из меню второго уровня к меню первого уровня слои не накладывались друг на друга необходимо прописать функцию обнуления.

```

function resetTrans(){
_root.rulMehanizmV=0;
_root.rulKardanV=0;
_root.barankaV=0;
_root.pnevmoUsiliteIV=0;
_root.kompressorV=0;
_root.ballonsV=0;
_root.ohlagdenieV=0;
_root.smazkaV=0;
_root.toplivoV=0;
_root.dvsV=0;
_root.kardansV=0;
_root.korobkaPeredachV=0;
_root.mostyV=0;
_root.tormozV=0;
_root.kabinaV=0;

```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		73

_root.razdatochnayaV=0;}

Далее остается только сделать публикацию файла с деталями машины и прописать загрузку этого файла в основное приложение.

2.3.4 Описание информационной подсистемы

Для хранения информации по неисправностям и способам их устранения узлов и механизмов автокрана, была спроектирована реляционная база данных. Анализ предметной области позволяет определить набор сущностей, необходимых для удовлетворения требований к системе. Набор всех сущностей представлен в таблице 3.

Таблица 3 – Формирование сущностей

Название сущности	Описание сущности
1	2
Узлы	Содержит информацию о узлах автомобиля
Механизмы	Содержит информацию о механизмах автомобиля
Неисправности	Содержит информацию о неисправностях механизмов автомобиля
Причины	Содержит информацию о причинах и способах устранения неисправностей механизмов автомобиля

Путем анализа предметной области было выделено набор атрибутов для каждой сущности. Они отображены в таблицах 4 – 7.

В сущности «Узлы» ключевым атрибутом является «id_узла».

Таблица 4 – Атрибуты сущности «Узлы»

Название атрибута	Описание атрибута	Диапазон значений	Единицы измерения	Пример
1	2	3	4	5
id_узла	Идентификатор узла	–	–	1
Название узла	Наименование узла автомобиля	–	–	Трансмиссия

В сущности «Механизмы» ключевым атрибутом является «id_механизма».

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		74

Таблица 5 – Атрибуты сущности «Механизмы»

Название атрибута	Описание атрибута	Диапазон значений	Единицы измерения	Пример
1	2	3	4	5
id_механизма	Идентификатор механизма	–	–	1
Название механизма	Наименование механизма автомобиля	–	–	Система смазки
Название узла	Наименование узла автомобиля	–	–	Силовые установки

В сущности «Неисправности» ключевым атрибутом является «id_неисправности».

Таблица 6 – Атрибуты сущности «Неисправности»

Название атрибута	Описание атрибута	Диапазон значений	Единицы измерения	Пример
1	2	3	4	5
id_неисправности	Идентификатор неисправности	–	–	1
Название неисправности	Наименование неисправности автомобиля	–	–	Двигатель стучит
Название механизма	Наименование механизма автомобиля	–	–	Система охлаждения
Название узла	Наименование узла автомобиля	–	–	Несущая конструкция

В сущности «Причины» ключевым атрибутом является «id_причины».

Таблица 7 – Атрибуты сущности «Причины»

Название атрибута	Описание атрибута	Диапазон значений	Единицы измерения	Пример
1	2	3	4	5

1	2	3	4	5
id_причины	Идентификатор причины неисправности	–	–	1
Причина	Наименование неисправности автомобиля	–	–	Износ поршневых колец
Способ устранения	Наименование механизма автомобиля	–	–	Заменить кольца
Название неисправности	Наименование узла автомобиля	–	–	Двигатель стучит

Для того, чтобы БД сохраняла целостность при любых операциях над ней, все отношения в ней нормализованы, БД приведена к третьей нормальной форме. Итоговая физическая модель приведена на рисунке 22.

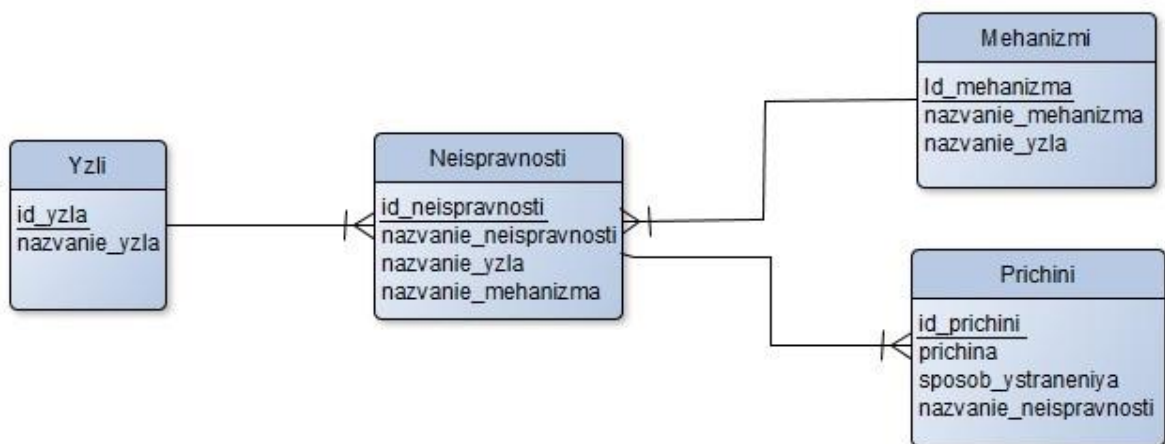


Рисунок 22 – Итоговая физическая модель БД

3.2 Руководство пользователя

Приложение предназначено для визуального изучения общего технического устройства автокрана Урал-4320, а также его узлов и агрегатов. Позволяет просмотреть теоретическую часть по эксплуатации автомобиля, выдает информацию по неисправностям выбранных механизмов, возможен поиск по неисправностям с помощью ключевого слова.

Для запуска программы необходимо запустить файл «ural4320.exe». После чего запустится главное рабочее окно (рисунок 23).

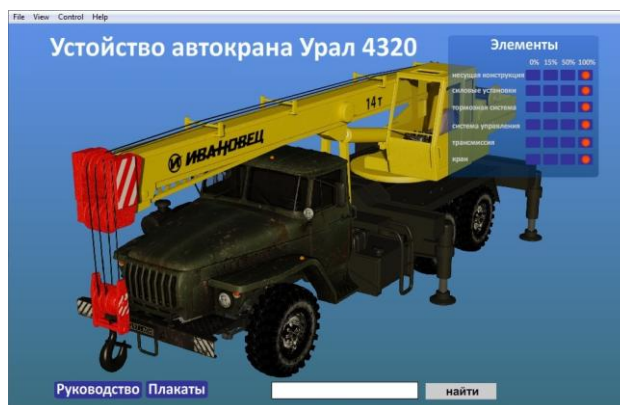


Рисунок 23 – Главное окно

В правом верхнем углу расположено меню первого уровня, содержащее основные узлы автокрана. Для всех элементов предусмотрен выбор прозрачности (рисунок 24).

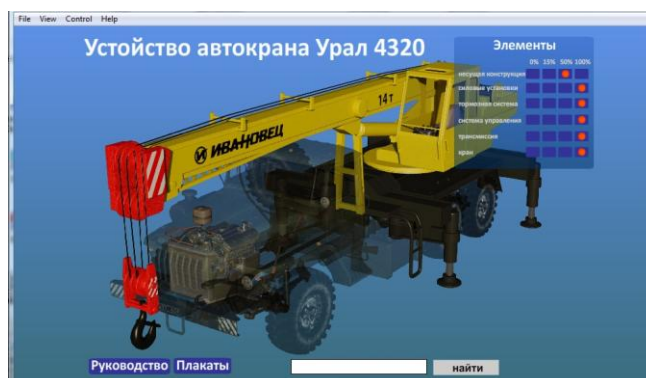


Рисунок 24 – Выбор прозрачности элементов

При нажатии на один из пунктов первого уровня появляется меню второго уровня с возможностью выбрать механизм для более детального рассмотрения, при этом непрозрачность механизма 100%, выбранного узла из меню первого уровня 50%, остальных частей 15% (рисунок 24).

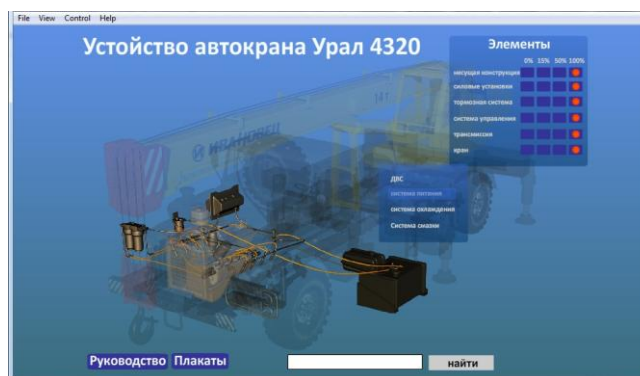


Рисунок 24 – Меню второго уровня

При выборе одного из пунктов меню второго уровня появляется таблица с информацией по неисправностям, причинам и способам их устранения, выбранного механизма (рисунок 25).

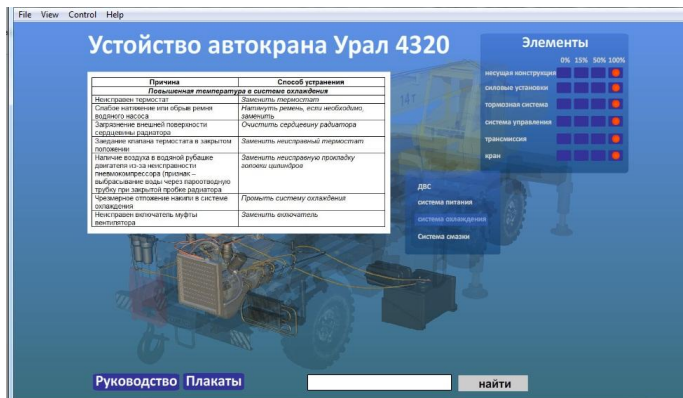


Рисунок 25 – Вывод таблицы

При нажатии кнопки «Руководство» появляется меню с выбором главы теоретической части руководства по эксплуатации автокрана (рисунок 26).

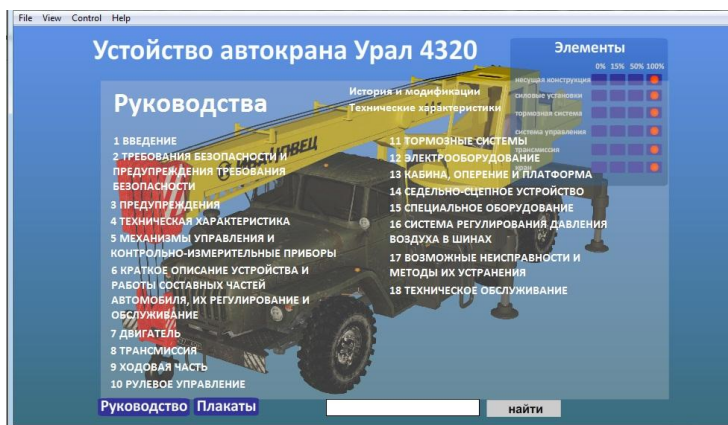


Рисунок 26 – Меню руководство по эксплуатации автокрана

При выборе одной из глав руководства по эксплуатации автокрана на экран выводится теоретическая часть в виде отдельного окна (рисунок 27).

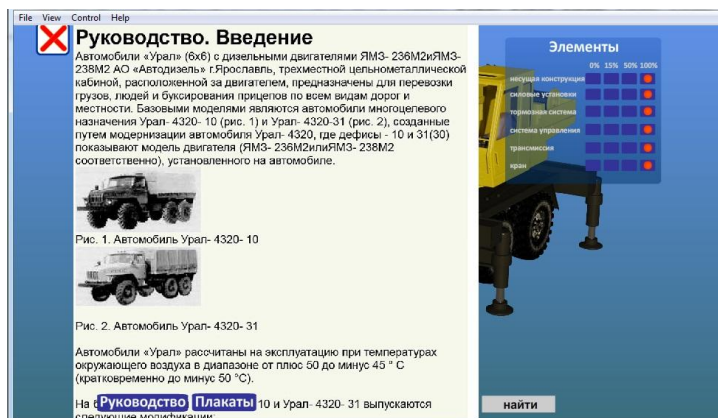


Рисунок 27 – Окно теоретической части руководства

При нажатии кнопки «Плакаты» появляется меню с выбором плакатов отображающих принцип работы основных систем автокрана (рисунок 27).

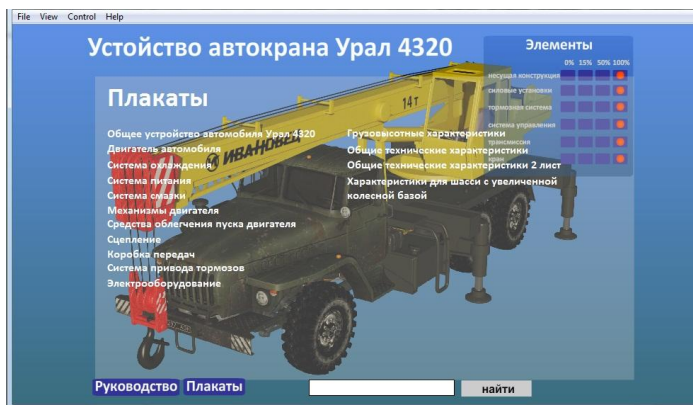


Рисунок 27 – Меню плакаты

При выборе одного из пунктов списка плакатов на весь экран выводится окно с изображением плаката (рисунок 28).

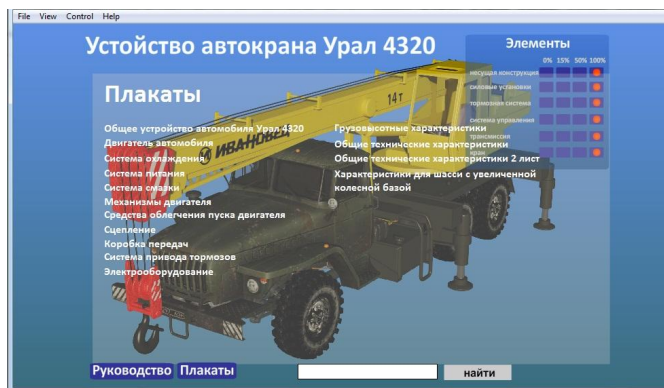


Рисунок 28 – Окно с изображением плаката

Для поиска информации по неисправностям, причинам и способам их устранения, выбранного механизма, необходимо в строку поиска ввести ключевое слово, в результате программа выдаст таблицу с интересующей информацией (рисунок 29).

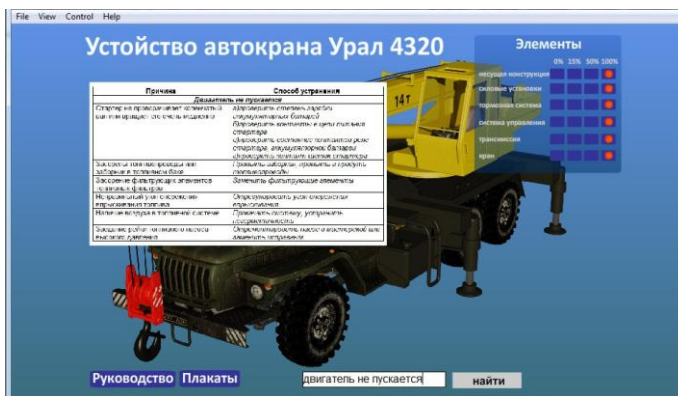


Рисунок 29 – Поиск по неисправностям

Минимальные требования к техническим характеристикам рабочих станций следующие:

- одноядерный процессор с тактовой частотой 1 ГГц;
- объем оперативной памяти от 512 Мбайт;
- видеоадаптер с объемом графической памяти не менее 256 Мбайт;
- размер дискового пространства от 10 Гбайт.

Из ключевых особенностей программы можно выделить следующие:

- вращение модели, а также выбранного механизма или узла вокруг своей оси;
- добавление прозрачности (15 %, 50 %, 100 %) выбранным элементам на общей модели автокрана;
- просмотр плакатов, отображающих принцип работы основных систем автокрана;
- просмотр теоретической части руководства по эксплуатации автокрана;
- поиск по неисправностям, причинам и способам их устранения, выбранного механизма.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		80

Заключение

В ходе диссертационного исследования была изучена предметная область трехмерного компьютерного моделирования. Рассмотрены имеющиеся подходы к созданию модели и существующие аналогичные программные продукты. Сделан вывод о том, что все модели разработаны в рамках систем автоматизированного проектирования, работают согласно предустановленного программного обеспечения, являются избыточными и как следствие дорогостоящими.

Предложен альтернативный подход к созданию модели и ее визуализации с помощью игрового процессора. В итоге разработано приложение обучающего направления.

Разработанный в ходе исследования программный продукт, позволяет детально изучить общее техническое устройство автокрана Урал-4320. Созданная модель делится на подсистемы, состоящие из узлов и агрегатов автомобиля. Благодаря функции скроллинга, модель можно увеличить и вращать вокруг своей оси. Приложение позволяет выбрать интересующий механизм из всей модели, для более детального изучения, а благодаря настройкам прозрачности сделать это еще удобнее.

Так же программа привязана к базе данных, при выборе конкретного механизма, из меню второго уровня, система выдаст теоретическую информацию по неисправностям составных элементов, возможные причины появления неисправностей и методы устранения таковых. Так же для большего удобства организован поиск по неисправностям. Приложение включает в себя демонстрационные плакаты, отображающие принцип работы систем автокрана, а также содержит теоретическую часть руководства по эксплуатации автогрузовика.

В связи со сложившейся ситуацией в сфере специального образования, а именно, необходимости внедрения современных компьютерных технологий и быстрого развития информационных систем, особую актуальность представляет создание трёхмерных моделей грузовых специализированных автомобилей. При создании программного продукта основной упор сделан на возможное использование

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		81

приложения в обучающих целях, для подготовки молодых специалистов. Другими словами, программа содержит трехмерную, высоконаглядную модель полностью имитирующую автокран Урал-4320.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		82

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Атре, Ш. Современные методы 3D моделирования / Ш. Атре. – М. : Моделирование и проектирование, 2003. – 234 с.
- 2 Бесплатные игровые движки для разработки 2D/3D игр / QIQER.RU URL:<http://qiqer.ru/besplatnye-igrovye-dvizhki-dlya-razrabotki-2d3d-igr.html> (дата обращения 11.04.16).
- 3 Благодатских, В.А. Разработка и использование программного обеспечения ЭВМ / В. А. Благодатских, М. А. Енгибарян, Е. В. Ковалевская. – М. : Проектирование и программирование, 2012. – 210 с.
- 4 Божко, А.Н. Компьютерная графика : учеб. пособие для вузов / А. Н. Божко, Д. М. Жук, В. Б. Маничев. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2009. – 392 с
- 5 Большаков, В.П. Инженерная и компьютерная графика : практикум / В. П. Большаков. – СПб. : БХВ-Петербург, 2014. – 592 с.
- 6 Большаков, В.П. Инженерная и компьютерная графика: Учебное пособие / В. П. Большаков, В. Т. Тозик, А. В. Чагина. – СПб. : БХВ-Петербург, 2013. – 288 с.
- 7 Важность 3D-моделирования // TSN.NET.RU URL:<http://www.tsn.net.ru/knowbase/1711--3d-> (дата обращения 29.03.2016).
- 8 Ведерников, А.А. Урал-4320-10/ 4320-31 и их модификации. Руководство по эксплуатации / А. А. Ведерников. – Миасс : ОАО «Автомобильный завод Урал», 2003. – 219 с.
- 9 Вендров, А.М. CASE – технологии. Современные методы и средства проектирования информационных систем / А. М. Вендров. – М. : Проектирование и программирование, 2008. – 176 с.
- 10 Верстак, В.И. 3dsMax 9. Секреты мастерства / В. И. Верстак, О. И. Нессонов. – СПб. : Питер, 2009. – 320 с.
- 11 Габидулин, В.М. Трехмерное моделирование в AutoCAD 2014 / В. М. Габидулин. – М. : ДМК Пресс, 2014. – 280с.
- 12 Гайкович, В. Основы работы с графическими редакторами / В. Гайкович,

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		83

А. Першин. – М. : Единая Европа, 2014. – 328 с.

13 Галкин, Н.С. Создание трехмерной модели местности в Unity3d // Инновационные технологии: теория, инструменты, практика / Н. С. Галкин, Е. А. Ромин. 2014. №1. С. 311-316

14 Голованов, Н.Н. Геометрическое моделирование / Н. Н. Голованов. – М. : Изд-во Физико-математической литературы, 2012. – 472 с.

15 Голованов Н.Н. Компьютерная геометрия : учеб. пособие для студ. вузов / Н. Н. Голованов [и др.]. – М. : Академия, 2010. – 512 с.

16 Дегтярев, В.М. Инженерная и компьютерная графика: Учебник для учреждений высшего профессионального образования / В. М. Дегтярев. – М.: Академия, 2011. – 240 с.

17 Десятка лучших движков для создания своих собственных игр / НАВРА-НАВР.RU URL:<https://habrahabr.ru/post/257019/> (дата обращения 11.03.16).

18 Денискин, Ю.И. Современные методы геометрического моделирования в обеспечении качества жизненного цикла наукоемкой продукции / Ю. И. Денискин. – М. : МАИ, 2007. – 268 с.

19 Залогова, Л.А. Компьютерная графика. Элективный курс: Учебное пособие / Л. А. Залогова. – М. : БИНОМ. ЛЗ, 2009. – 213 с.

20 Зеленко, Л.С. Виртуальная дистанционная обучающая среда: особенности реализации и применения в учебном процессе // ученые записки ИСГЗ / Л. С. Зеленко, Д. А. Загуменнов. – 2013. №1 (11). С. 302-305.

21 Зеленко, Л.С. Основы построения виртуальной информационно-образовательной среды / Л. С. Зеленко, Д.А. Загуменнов. – Вестник Самарского Государственного Аэрокосмического Университета им. академика С.П. Королева (Национального Исследовательского Университета). 2012. №7 (38). С. 46-53.

22 Кашев, Л.Е. Армейские автомобили Урал-375, 4320 / Л. Е. Кашев. – М. : Russian Motor Books, 2000, – 20 с.

23 Кащеев, Л.Б. Урал-4320/ 5323/ 6301 - военные грузовики (Военные машины №65) / Л. Б. Кащеев. – Киров : Кировское общество Любителей военной техники и моделизма, 2004. – 64 с.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		84

24 Кудрявцев, Е.М. Компас-3D. Основы работы в системе / Е. М. Кудрявцев. – М. : ДМК Пресс, 2008. – 528 с.

25 Курс для студентов – дизайнеров. Шпаргалка-напоминалка по 3ds Max // STUDFILES.RU URL:<http://www.studfiles.ru/preview/4437280> (дата обращения 20.12.2015).

26 Ли, К. Основы САПР (CAD/CAM/CAE) / Кунву Ли. – СПб. : Питер, 2004. – 560 с.

27 Логиновский, А.Н. Инженерная 3D-компьютерная графика: Учебное пособие для бакалавров / А. Н. Логиновский. – М. : Юрайт, 2013. – 464 с.

28 Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite / С. В. Маклаков. – М.: Диалог-МИФИ, 2009. – 432 с.

29 Миронов, Д.Ф. Компьютерная графика в дизайне: Учебник / Д. Ф. Миронов. – СПб. : БХВ-Петербург, 2008. – 560 с.

30 Молодёжь XXI века: шаг в будущее: трёхмерное компьютерное моделирование технического устройства автомобиля / Н. С. Зуев, П. Е. Куксенко. Благовещенск : БГПУ, 2016. Т. 3. С. 212-214

31 Немцова, Т.И. Практикум по информатике. Компьютерная графика и Web-дизайн. Практикум: Учебное пособие / Т. И. Немцова. – М. : ИД ФОРУМ, НИЦ ИНФРА-М, 2013. – 288 с.

32 Никулин, Е.А. Компьютерная геометрия и алгоритмы машинной графики / Е. А. Никулин. – СПб. : Петербург, 2015. – 576 с.

33 О значении 3D моделирования // HSPLINE.COM URL:<http://hspline.com/o-znachenii-3d-modelirovaniya.html> (дата обращения 11.04.2016).

34 Орлов, А. AutoCAD 2013 // А. Орлов.. – СПб.: Питер, 2013. – 384 с.

35 Основы 3D моделирования / В. В. Баронов [и др.]. – М. : ИНФРА-М, 2009. – 432 с.

36 Пантюхин, П.Я. Компьютерная графика. В 2-х т. Т. 1. Компьютерная графика: Учебное пособие / П. Я. Пантюхин. – М. : ФОРУМ, НИЦ ИНФРА-М, 2012. – 88 с.

37 Полищук, Н.Н. Самоучитель AutoCAD 2014 / Н. Н. Полищук.– СПб.: БХВ

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		85

–Петербург, 2014. – 464 с.

38 Роджерс, Д. Алгоритмические основы машинной графики : пер. с англ. / Д. Роджерс ; под ред. Ю. М. Баяковского и В. А. Галактионова. – М. : Мир, 2009. – 512 с.

39 Романченко, А.А. Дизельные автомобили «Урал» / А. А. Романченко, Н. Н. Чиненов, В. Т. Иванов. – М. : Транспорт, 1984. – 208 с.

40 Руководство к Car Tutorial (Unity3d) // НАВРАНАВР.RU <https://habrahabr.ru/post/247667/> (дата обращения 21.02.2016).

41 Сакулина, Ю.В. Компьютерная графика как средство формирования профессиональных компетенций / Ю. В. Сакулина, И. В. Рожина. – Педагогическое образование в России. 2012. № 6. С. 76-80.

42 Титоренко, Г.А. Автоматизированные информационные технологии. Учебник. / Г. А. Титоренко. – М.: ЮНИТИ, 2011. – 189 с.

43 Топ-10 игровых движков: выбери свой // APP2TOP.RU URL:http://app2top.ru/game_development/top-10-igrovyy-h-dvizhkov-vy-beri-svoj-45170.html (дата обращения 10.04.16).

44 Трёхмерная графика // WIKIPEDIA.ORG URL:https://ru.wikipedia.org/wiki/Трёхмерная_графика (дата обращения 5.02.16).

45 Чекмарев А.А. Инженерная графика (машиностроительное черчение) : учеб. : рек. НМС/ А.А. Чекмарев. – М.: ИНФРА–М, 2009. –396 с.

46 Чепак, Л.В. Методические указания к выполнению курсовой работы по дисциплине: «Базы данных». Практикум / Л. В. Чепак, А. Г. Масловская. – Благовещенск: Амурский гос. ун-т, 2010.

47 Черепашков, А.А. Компьютерные технологии, моделирование и автоматизированные системы в машиностроении / А. А. Черепашков, Н. В. Носов. – Волгоград : Ин-Фолио, 2009. – 640 с.

48 Чои, Джае-Джин. Моделирование и анимация персонажей в Maya / Джае-Джин Чои. – М. : НТ Пресс, 2010. – 760 с.

49 Хейфец, А.Л. Инженерная компьютерная графика. AutoCAD: учеб. пособие: рек. Мин. обр. РФ/ А. Л. Хейфец. – СПб. : БХВ-Петербург, 2007. – 316 с.

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		86

50 Шикин, Е.В. Компьютерная графика. Динамика, реалистические изображения / Е. В Шикин, А. В. Боресков.. – М.: Диалог-МИФИ, 2005. – 365 с.

51 3d моделирование и визуализация // KOLORO.UA URL<http://koloro.ua/3d-modelirovanie-i-vizualizaciya.html> (дата обращения 12.05.2016).

52 Autodesk 3ds Max // WIKIPEDIA.ORG https://ru.wikipedia.org/wiki/Autodesk_3ds_Max (дата обращения 15.03.2016).

53 Unreal Engine // WIKIPEDIA.ORG https://ru.wikipedia.org/wiki/Unreal_Engine (дата обращения 5.02.16).

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		87

ПРИЛОЖЕНИЕ А

Кодовая реализация клипов для плакатов

```
function openPlakatMenu(){
if(main.plakatsMenu._visible==0){
main.plakatsMenu._visible=1;
main.tutorsMenu._visible=0;
main.tutors._visible=0;
main.plakats._visible=0;
}else{
main.plakatsMenu._visible=0;}}
function openTutorsMenu(){
if(main.tutorsMenu._visible==0){
main.tutorsMenu._visible=1;
main.plakatsMenu._visible=0;
main.tutors._visible=0;
}else{
main.tutorsMenu._visible=0;}}
function openPlakat(page:Number){
if(main.plakats._visible==0){
main.plakats._visible=1;
main.tutors._visible=0;
main.plakats._x=0;
main.plakats._y=0;
if(page==1){
main.plakats.list2.loadMovie("plakats/01_URAL4320.jpg");
main.plakatsMenu._visible=0;}
if(page==2){
main.plakats.list2.loadMovie("plakats/02_URAL4320.jpg");
main.plakatsMenu._visible=0;}}
```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		88

Продолжение ПРИЛОЖЕНИЯ А
Кодовая реализация клипов для плакатов

```
if(page==3){
  main.plakats.list2.loadMovie("plakats/03_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==4){
  main.plakats.list2.loadMovie("plakats/04_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==5){
  main.plakats.list2.loadMovie("plakats/05_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==6){
  main.plakats.list2.loadMovie("plakats/06_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==7){
  main.plakats.list2.loadMovie("plakats/07_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==8){
  main.plakats.list2.loadMovie("plakats/08_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==9){
  main.plakats.list2.loadMovie("plakats/09_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==10){
  main.plakats.list2.loadMovie("plakats/10_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
if(page==11){
  main.plakats.list2.loadMovie("plakats/11_URAL4320.jpg");
  main.plakatsMenu._visible=0;}
```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		89

Продолжение ПРИЛОЖЕНИЯ А
Кодовая реализация клипов для плакатов

```
if(page==21){
main.plakats.list2.loadMovie("plakats/12.jpg");
main.plakatsMenu._visible=0;}
if(page==22){
main.plakats.list2.loadMovie("plakats/13.jpg");
main.plakatsMenu._visible=0;}
if(page==23){
main.plakats.list2.loadMovie("plakats/14.jpg");
main.plakatsMenu._visible=0;}
if(page==24){
main.plakats.list2.loadMovie("plakats/15.jpg");
main.plakatsMenu._visible=0;}
}else{
//main.plakats._visible=0;}}
function openTutor(page:Number){
if(main.tutors._visible==0){
main.tutors._visible=1;
main.tutorsMenu._visible=0;
main.plakats._visible=0;
}else{
main.tutors._visible=0;
main.tutors.list1._y=0;}
if(page==1){
main.tutors.list1.loadMovie("tutor/t01.jpg");}
if(page==2){
main.tutors.list1.loadMovie("tutor/t02.jpg");}
if(page==3){
```

					ВКР.145366.090401.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		90

Продолжение ПРИЛОЖЕНИЯ А
Кодовая реализация клипов для плакатов

```
main.tutors.list1.loadMovie("tutor/t03.jpg");}
if(page==4){
main.tutors.list1.loadMovie("tutor/t04.jpg");}
if(page==5){
main.tutors.list1.loadMovie("tutor/t05.jpg");}
if(page==6){
main.tutors.list1.loadMovie("tutor/t06.jpg");}
if(page==7){
main.tutors.list1.loadMovie("tutor/t07.jpg");}
if(page==8){
main.tutors.list1.loadMovie("tutor/t08.jpg");}
if(page==9){
main.tutors.list1.loadMovie("tutor/t09.jpg");}
if(page==10){
main.tutors.list1.loadMovie("tutor/t10.jpg");}
if(page==11){
main.tutors.list1.loadMovie("tutor/t11.jpg");}
if(page==12){
main.tutors.list1.loadMovie("tutor/t12.jpg");}
if(page==13){
main.tutors.list1.loadMovie("tutor/t13.jpg");}
if(page==14){
main.tutors.list1.loadMovie("tutor/t14.jpg");}
if(page==15){
main.tutors.list1.loadMovie("tutor/t15.jpg");}
if(page==16){
main.tutors.list1.loadMovie("tutor/t16.jpg");}
```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		91

Продолжение ПРИЛОЖЕНИЯ А
Кодовая реализация клипов для плакатов

```
if(page==17){
main.tutors.list1.loadMovie("tutor/t17.jpg");}
if(page==18){
main.tutors.list1.loadMovie("tutor/t18.jpg");}
if(page==21){
main.tutors.list1.loadMovie("tutor/hystoriMod1.jpg");}
if(page==22){
main.tutors.list1.loadMovie("tutor/tehHaract1.jpg");}
if(page==23){
main.tutors.list1.loadMovie("tutor/tehHaract1.jpg");}
main.tutors.list1._x=0;
main.tutors.list1._y=0;
main.tutors._x=0;
main.tutors._y=0;
main.tutorsMenu._visible=0;}
```

					ВКР.145366.090401.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		92