

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Доцент департамента
программной инженерии
факультета компьютерных наук
канд. техн. наук

_____ А. Р. Закиевна
« ____ » _____ 2016 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»

_____ В. В. Шилов
« ____ » _____ 2016 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

ПРОГРАММА СКЕЛЕТНАЯ АНИМАЦИЯ

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.509000-01 ТЗ 01-1-ЛУ

Исполнитель
студент группы 151 ПИ
_____/А. М. Абрамов /

« ____ » _____ 2016 г.

2016

ПРОГРАММА СКЕЛЕТНАЯ АНИМАЦИЯ

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.500900 ТЗ 01-1-ЛУ

Листов 16

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Содержание

1 Введение	2
1.1 Наименование	2
1.2 Краткая характеристика	2
1.3 Документы, на основании которых ведется разработка	2
2 Назначение разработки	3
2.1 Функциональное назначение	3
2.2 Эксплуатационное назначение	3
3 Технические характеристики	4
3.1 Постановка задачи на разработку программы	4
3.2 Описание алгоритма и функционирования программы	4
3.2.1 Изменение местоположения точек в 3-х мерном пространстве	4
3.2.2 Работа с OpenGL	4
3.3 Обоснование выбора алгоритма решения задачи	5
3.4 Метод организации входных и выходных данных	5
3.4.1 Описание метода входных и выходных данных	5
3.4.2 Обоснование выбора метода организации входных и выходных данных	5
3.5 Выбор состава технических средств	6
3.5.1 Состав технических и программных средств	6
3.6 Обоснование выбора технических и программных средств	6
4 Техничко-экономические показатели	7
4.1 Ориентировочная экономическая эффективность и годовая потребность	7
4.2 Экономические преимущества разработки	7
5 Источники, используемые при разработке	8
5.1 Список используемой литературы	8
6 Приложение 1. Терминология	9
6.1 Терминология	9

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

1. Введение

1.1. Наименование

Наименование: «Программа скелетная анимация»

1.2. Краткая характеристика

Программа предназначена для быстрого просмотра анимационных файлов созданных в пакетах для 3-х мерного моделирования.

1.3. Документы, на основании которых ведется разработка

Приказ НИУ ВШЭ No 6.18.1-02/1112-19 от 11.12.2015 в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

2. Назначение разработки

2.1. Функциональное назначение

Функциональным назначением приложения является предоставление пользователю возможности быстро загрузить 3D анимацию из файла, проиграть ее, показывать анимацию учитывая связанные с ней материалы и нормали, просмотреть мешы и кости входящие в состав анимации, перейти к любому моменту времени в анимационном файле.

2.2. Эксплуатационное назначение

Программа наглядно демонстрирует содержание файла экспортированного из пакетов для 3-х мерного моделирования. Она может использоваться в процессе отладки приложений использующих анимацию и в работе дизайнера 3D моделей. В силу простоты интерфейса она подходит для использования людям не очень хорошо знакомым с более мощными и трудными в использовании пакетами для 3-х мерного моделирования, которые можно использовать для просмотра содержимого файла анимации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

3. Технические характеристики

3.1. Постановка задачи на разработку программы

Разработанный программный продукт обязан:

1. Выполнять загрузку данных из специально отформатированного файла коллада (collada или .dae).
2. Проигрывать записанную в данном файле анимацию.
3. Предоставлять пользователю возможность перейти к любому моменту времени в анимации.
4. Отрисовывать кости модели.
5. Иметь возможность вкл./выкл. учитывание нормалей и материалов во время отрисовки.
6. Поддерживать два вида камер в OpenGL, первый вид это камера движение которой сковано орбитой вокруг модели и другой тип это камерадвигающаяся совершенно свободно.

3.2. Описание алгоритма и функционирования программы

3.2.1. Изменение местоположения точек в 3-х мерном пространстве

Алгоритм скелетной анимации основан на наличии следующей информации:

1. Простого меша, который называется скелетом.
2. Позиции этого меша в определенные моменты времени.
3. Комплексного меша модели.
4. Связь каждой вершиной модели с «костью» из скелета.

Для реализации алгоритма был написан класс применяющий к костям скелета деформацию соответствующую заданному моменту во времени. Деформация - это матрица переводящая старую пару координат, которые определяют кость, в новую пару. Однако она нам не известна. Так как позиции костей в скелете известны нам только в ключевые моменты времени, необходимо использовать алгоритмы интерполяции для нахождения матрицы деформации для данного кадра.

```
1 print "hello world from ApplyArmature.cs"
```

Каждая матрица деформации находится относительно матрицу деформации кости-родителя. Далее матрица деформации должна быть переведена из системы координат кости-родителя в систему координат меша скелета. Это позволяет применять их на вершинах в произвольном порядке, что необходимо для увеличения производительности. Ниже приведен код переводящий деформации кости в глобальные координаты:

```
1 print "hello world from ApplyArmature.cs"
```

Каждый кадр необходимо для каждой вершине комплексного меша находить кости, которые имеют право на нее воздействовать и применять к вершине такую же деформацию, что была применена и к костям. Ниже приведен код занимающийся поиском костей влияющих на вершину и применяющий к ней матрицу деформации каждой кости в зависимости от веса кости:

```
1 print "hello world from Entity.cs"
```

3.2.2. Работа с OpenGL

Графический драйвер поддерживающий OpenGL, использует архитектуру наподобие клиент-сервер. Данные о вершинах, материалах и нормалях необходимо загружать в буферы памяти расположенные на видеокарте. Ниже приведен код для загрузки этих данных:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```
1 print "hello world from DrawMesh.cs"
```

Для создания эффекта движения необходимо менять каждый кадр содержимое буферов расположенных на графической карте. А именно необходимо менять координаты вершин и направления нормалей к каждой вершине (для корректного отображения света/тени). Для этого необходимо послать запрос к драйверу OpenGL и получить указатель на память с загруженными данными. Так как эта память не принадлежит GCCollector, (объекту .Net Framework отвечающему за контролируруемую память) то все манипуляции с ней должны производиться в блоке «unsafe». Далее приведен код модифицирующий данные в буфере для следующего кадра.

```
1 print "hello world from DrawMesh.cs and Entity.cs"
```

3.3. Обоснование выбора алгоритма решения задачи

Выбор алгоритма скелетной анимации продиктован постановкой задачи, а именно пунктом о загрузке данных из специально отформатированного файла коллада (collada или .dae).

Перевод матриц деформации из системы координат кости-родителя в систему координат меша скелета позволяет применять их на вершинах в произвольном порядке, что необходимо для увеличения производительности.

Данные о вершинах, материалах и нормалях необходимо загружать в буферы памяти расположенные на видеокарте для того что бы обеспечить приложению скорость не менее чем в 20 кадров в секунду для моделей состоящих более чем из 1,300,000 треугольников.

Модифицировать буфер памяти необходимо после каждого кадра, в OpenGL существует специальный тип буферов для таких случаев создаваемый с помощью флага STREAM_WRTIE. Далее для удобства и скорости создается словарь сопоставляющий индексу вершины, влияющие на нее кости и их камулятивную матрицу деформации, это позволяет сэкономить на поиске подходящей кости в скелете.

3.4. Метод организации входных и выходных данных

3.4.1. Описание метода входных и выходных данных

Входными данными является файл в формате коллада, в котором в обязательном порядке должны присутствовать следующие элементы:

1. Один меш с соответствующим ему тегом <Name=«Mesh»>
2. Один трэк анимации.
3. Один скелет связанный с мешем, и с соответствующим ему тегом <Name=«Armature»>

Выходными данными является отображение анимации на экране.

3.4.2. Обоснование выбора метода организации входных и выходных данных

Если не выполнены условия на наличие меша, трэка анимации и связанного с мешем скелета то у программы не хватит информации для воспроизведения анимации. Наименования объектов, экспортируемых из 3-х мерного пакета для моделирования, необходимы для того чтобы различать меш и скелет для меша, так как при экспорте они получают одинаковую структуру внутри файла.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

3.5. Выбор состава технических средств

3.5.1. Состав технических и программных средств

Для возможности запустить приложение необходимо учесть следующие системные требования:

1. Компьютер, оснащенный:
 - (а) Обязательно 64-разрядный (x64) процессор с тактовой частотой 1 гигагерц (ГГц) или выше;
 - (б) 512 мегабайт (ГБ) оперативной памяти (ОЗУ);
 - (с) 2 ГБ (для 64-разрядной системы) пространства на жестком диске;
 - (д) графическое устройство OpenGL с драйвером версии 3.1 или выше.
2. Монитор
3. Видеокарта
4. Мышь
5. Клавиатура

Также необходимо учесть следующие программные требования:

1. Поддержка OpenGL версии не менее 3.1
2. 64-битная операционная система Windows 7.
3. .NET Framework версии не ниже 4.5.1
4. Библиотека Assimp версии не ниже 3.1
5. Библиотека OpenTK версии не ниже 1.1.4

3.6. Обоснование выбора технических и программных средств

Программа была протестирована и отлажена на версии OS Windows 7 с использованием .Net Framework 4.5.1, OpenTK версии 1.1.4 и Assimp версии 3.1.

Качество и корректность работы программы при других версиях не проверялось.

Программа использует буферы графической памяти типа STREAM_WRITE и функции glMapData и glSubBufferData которые в OpenGL официально поддерживаются лишь с версии 3.1

64-х разрядный компьютер необходим для того чтобы упростит логику взаимодействия в блоках кода unsafe с внешней памятью выделяемой графическим драйвером.

Технические требования к памяти и периферии не превышают технических требований к операционной системе Windows 7 с установленным на ней .Net Framework 4.5.1

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

4. Техничко-экономические показатели

4.1. Ориентировачная экономическая эффективность и годовая потребность

В силу простоты интерфейса она подходит для использования людям не очень хорошо знакомым с более мощными и трудными в использовании пакетами для 3-х мерного моделирования, которые можно использовать для просмотра содержимого файла анимации. Предполагается, что программа будет использоваться пользователем несколько раз в неделю, на протяжении коротких периодов времени, т. е. количество сеансов на одном рабочем месте в год составит примерно 48 сеансов. Она может использоваться в процессе отладки приложений использующих анимацию и в работе дизайнера 3D моделей.

4.2. Экономические преимущества разработки

Экономические преимущества разработки в сравнении с лучшими отечественными и зарубежными аналогами рассчитаны на январь 2016 года. Существующими аналогами данного приложения являются пакеты для 3-х мерного моделирования и анимации. В силу того что данное приложение распространяется бесплатно, единственным экономически выгодным аналогом к нему будет программа Blender. Однако Blender гораздо более сложен в использовании и потребляет намного больше системных ресурсов (жесткой памяти, ОЗУ, времени процессора).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

5. Источники, используемые при разработке

5.1. Список используемой литературы

1. OpenGL Superbible: Comprehensive Tutorial and Reference (7th Edition) Graham Sellers (Author), Richard S Wright Jr. (Author), Nicholas Haemel (Author) ISBN-13: 978-0672337475
2. Порев В.Н. Компьютерная графика. – СПб.: БХВ-Петербург, 2002. – 432 с.: ил.
3. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению // Единая система программной документации. -М.:ИПК Издательство стандартов, 2001.
4. ГОСТ 19.101-77 Виды программ и программных документов //Единая система программной документации. -М.: ИПК Издательство стандартов, 2.: 001.
5. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. -М.: ИПК Издательство стандартов, 2001.
6. Требования к системе для .NET Framework 4.5. [Электронный ресурс]// URL: [https://msdn.microsoft.com/ru-ru/library/8z6watww\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/8z6watww(v=vs.110).aspx) (Дата обращения: 23.02.2016, режим доступа: свободный).
7. Системные требования ОС Windows 7. [Электронный ресурс]// URL: <http://windows.microsoft.com/ru-ru/windows7/products/system-requirements> (Дата обращения: 20.08.2016, режим доступа: свободный).
8. Документация OpenGL 3.3 [Электронный ресурс] // <https://www.opengl.org/sdk/docs/man/> (Дата обращения: 21.10.2016, режим доступа: свободный)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

6. Приложение 1. Терминология

6.1. Терминология

Корневая вершина (англ. root node) Самый верхний узел дерева.

Полигональная сетка (жарг. меш от англ. polygon mesh) Совокупность вершин, рёбер и граней, которые определяют форму многогранного объекта в трехмерной компьютерной графике и объёмном моделировании. Гранями являются треугольники.

Дерево Связный ациклический граф. Связность означает наличие путей между любой парой вершин, ацикличность — отсутствие циклов и то, что между парами вершин имеется только по одному пути.

Степень вершины Количество инцидентных ей (входящих/исходящих из нее) ребер.

Интерполяция, интерполирование анимации Способ нахождения промежуточных значений состояния анимации по имеющемуся дискретному набору известных значений.

Z-буферизация В компьютерной трёхмерной графике способ учёта удалённости элемента изображения. Представляет собой один из вариантов решения «проблемы видимости»

Z-конфликт (англ. Z-fighting) Если два объекта имеют близкую Z-координату, иногда, в зависимости от точки обзора, показывается то один, то другой, то оба полосатым узором.

OpenGL (Open Graphics Library) Спецификация, определяющая независимый от языка программирования платформонезависимый программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику. На платформе Windows конкурирует с Direct3D.

Рендеринг (англ. rendering — «визуализация») Термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы.

Текстура Растровое изображение, накладываемое на поверхность полигональной модели для придания ей цвета, окраски или иллюзии рельефа. Приблизительно использование текстур можно легко представить как рисунок на поверхности скульптурного изображения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.509000 ТЗ 01-1-ЛУ				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата