# **Министерство образования Российской Федерации**

**Профессиональный колледж №**

**Специальность**:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## КУРСОВОЙ ПРОЕКТ

**Дисциплина \_\_Технология разработки программных продуктов\_**

**Тема \_\_\_\_\_Игровая программа «Автомобильные гонки»\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**( КР. 2203. 04. 35. ПЗ )**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Руководитель \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата Личная подпись Расшифровка подписи

Нормоконтролер \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Зав. кафедрой \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил студент группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Новокузнецк, 2005**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СОДЕРЖАНИЕ ВВЕДЕНИЕ 3 1. ОБЩАЯ ЧАСТЬ 4  1.1 ЯЗЫК ПРОГРАММИРОВАНИЯ DELPHI 4  1.2 Delphi. Основные характеристики продукта 4  1.3 Компилятор в машинный код 6  1.4 Объектно-Ориентированная модель программных компонентОВ 6  1.5 Масштабируемые средства для построения баз даннЫх 8  1.6 Некоторые особенности Delphi 8  1.7 Библиотека визуальных компонентов 9  1.8 Структурное объектно-ориентированное программирование 9  1.9 Поддержка OLE 2.0, DDE и VBX 10  **2. Специальная часть 10  2.1 Постановка задачи 10  2.2 Минимальные технические требования и ОС 10  2.3 Проектирование пп 11  2.4 Основные алгоритмы для событий на форме и компонентах 12 3. разработка 12  3.1 Кратко описания программной среды разработки 12  3.2 Реализация основных алгоритмов в среде разработки 12 4. Документирование 13  4.1 Инструкция пользователя по пп 13 5. Тестирование 15  5.1 тестирование функций пп 15  5.2 тестирование надёжности 15 Приложение 16** | | | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | | | |
|  |  |  |  |  |
| Изм | Лист | № докум | Подп | Дата |
| Разраб. | |  |  |  | **Игровая программа «Автомобильные гонки»** | Лит | Лист | Листов |
| Пров. | |  |  |  |  | 2 | 24 |
| Н. контр | |  |  |  |  | | |
| Утв. | |  |  |  |
|  | |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ВведениеДанный программный продукт предназначен для развлекательных целей. Эта программа арканный симулятор гонок. Действие которого разворачивается на дороге.Цель игры проехать расстояние 20 километров.Во время игры на дороге будут встречаться:Другие машины (противники) препятствующие прохождению, при столкновении отнимающие у вас 1 единицу топлива;Значок в виде бочки (топливо), восполняющий потери при столкновении или при продолжительной езде;Нитро, элемент ускорения помогающий развить на некоторое время большую скорость. Игра разрабатывалась в среде Delphi 5 с использованием библиотеки OpenGL версия 1.3.4582 помогающей ускорить вывод графического движка, Pixia версия 2.4g для создания и редактирования текстур, Image Editor версия 3.0 для создания иконок, 3D-Stydio Max версия 5.0 для создания моделей машин (игрока). | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 3 |
| Изм | Лист | № докум | Подп. | Дата. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1. Общая часть** **1.1 Язык программирования Delphi.**  Delphi - это греческий город, где жил дельфийский оракул. И этим именем был назван новый программный продукт с феноменальными характеристиками.  Hадо отметить, что к моменту выхода продукта обстановка вокруг компании Borland складывалась не лучшим для нее образом. Поговаривали о возможной перепродаже компании, курс акций компании неудержимо катился вниз. Сейчас уже можно без всяких сомнений утверждать, что период трудностей позади. Hеверно, конечно, было бы говорить, что только Delphi явился причиной восстановления компании.  Тем не менее, именно Delphi стал тем продуктом, на примере которого стало ясно, что у Borland есть еще долго не уйдет с ранка, и что один единственный продукт может настолько удачно сочетать несколько передовых технологий.  **1.2 Delphi. Основные характеристики продукта**  Delphi - это комбинация нескольких важнейших технологий:  · Высокопроизводительный компилятор в машинный код  · Объектно-ориентированная модель компонент  · Визуальное (а, следовательно, и скоростное) построение приложений из программных прототипов  · Масштабируемые средства для построения баз данных. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 4 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1.3 Компилятор в машинный код** Компилятор, встроенный в Delphi, обеспечивает высокую производительность, необходимую для построения приложений в архитектуре “клиент-сервер”. Этот компилятор в настоящее время является самым быстрым в мире, его скорость компиляции составляет свыше 120 тысяч строк в минуту на компьютере 486DX33. Он предлагает легкость разработки и быстрое время проверки готового программного блока, характерного для языков четвертого поколения (4GL) и в то же время обеспечивает качество кода, характерного для компилятора 3GL. Кроме того, Delphi обеспечивает быструю разработку без необходимости писать вставки на Си или ручного написания кода (хотя это возможно).  В процессе построения приложения разработчик выбирает из палитры компонент готовые компоненты как художник, делающий крупные мазки кистью. Еще до компиляции он видит результаты своей работы - после подключения к источнику данных их можно видеть отображенными на форме, можно перемещаться по данным, представлять их в том или ином виде. В этом смысле проектирование в Delphi мало чем отличается от проектирования в интерпретирующей среде, однако после выполнения компиляции мы получаем код, который исполняется в 10-20 раз быстрее, чем то же самое, сделанное при помощи интерпретатора. Кроме того, компилятор компилятору рознь, в Delphi компиляция производится непосредственно в родной машинный код, в то время как существуют компиляторы, превращающие программу в так называемый p-код, который затем интерпретируется виртуальной p-машиной. Это не может не сказаться на фактическом быстродействии готового приложения. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 5 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1.4 Объектно-ориентированная модель программных компонентОВ** Основной упор этой модели в Delphi делается на максимальном не использовании кода. Это позволяет разработчикам строить приложения весьма быстро из заранее подготовленных объектов, а также дает им возможность создавать свои собственные объекты для среды Delphi. Никаких ограничений по типам объектов, которые могут создавать разработчики, не существует. Действительно, все в Delphi написано на нем же, поэтому разработчики имеют доступ к тем же объектам и инструментам, которые использовались для создания среды разработки. В результате нет никакой разницы между объектами, поставляемыми Borland или третьими фирмами, и объектами, которые вы можете создать.  В стандартную поставку Delphi входят основные объекты, которые образуют удачно подобранную иерархию из 270 базовых классов. Для начала - неплохо. Но если возникнет необходимость в решении какой-то специфической проблемы на Delphi, советуем, прежде чем попытаться начинать решать проблему “с нуля”, просмотреть список свободно распространяемых или коммерческих компонент, разработанных третьими фирмами, количество этих фирм в настоящее время превышает число 250, хотя, возможно, я не обо всех знаю. Скептики, возможно, не поверят мне, когда я скажу, что на Delphi можно одинаково хорошо писать как приложения к корпоративным базам данных, так и, к примеру, игровые программы. Тем не менее, это так. Во многом это объясняется тем, что традиционно в среде Windows было достаточно сложно реализовывать пользовательский интерфейс. Событийная модель в Windows всегда была сложна для понимания и отладки. Но именно разработка интерфейса в Delphi является самой простой задачей для программиста. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 6 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Среда Delphi включает в себя полный набор визуальных инструментов для скоростной разработки приложений (RAD - rapid application development), поддерживающей разработку пользовательского интерфейса и подключение к корпоративным базам данных. VCL - библиотека визуальных компонент, включает в себя стандартные объекты построения пользовательского интерфейса, объекты управления данными, графические объекты, объекты мультимедиа, диалоги и объекты управления файлами, управление DDE и OLE. Единственное, что можно поставить в вину Delphi, это то, что готовых компонент, поставляемых Borland, могло бы быть и больше. Однако, разработки других фирм, а также свободно распространяемые программистами freeware-компоненты уже восполнили этот недостаток. Постойте, - скажете вы, ведь это уже было. Да, это было в Visual Basic.Соответствующий стандарт компонент назывался VBX. И этот стандарт так же поддерживается в Delphi. Однако, визуальные компоненты в Delphi обладают большей гибкостью. Вспомним, в чем была проблема в VB. Прикладной программист программировал, вообще говоря, в среде языка бэйсик. А компоненты в стандарте VBX готовили ему его коллеги-профессионалы на С++.VBX’ы приходили, “как есть”, и ни исправить, ни добавить ничего было нельзя. А для изготовления VBX надо было осваивать “кухню” языка C++. В Delphi визуальные компоненты пишутся на объектном Паскале, на том же Паскале, на котором пишется алгоритмическая часть приложения. И визуальные компоненты Delphi получаются открытыми для надстройки и переписывания. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 7 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1.5 Масштабируемые средства для построения баз данных** Объекты БД в Delphi основаны на SQL и включают в себя полную мощь Borland Database Engine. В состав Delphi также включен Borland SQL Link, поэтому доступ к СУБД Oracle, Sybase, Informix и InterBase происходит с высокой эффективностью. Кроме того, Delphi включает в себя локальный сервер Interbase для того, чтобы можно было разработать расширяемые на любые внешние SQL-сервера приложения в офлайновом режиме. Разработчик в среде Delphi, проектирующий информационную систему для локальной машины (к примеру, небольшую систему учета медицинских карточек для одного компьютера), может использовать для хранения информации файлы формата .dbf (как в dBase или Clipper) или .db (Paradox). Если же он будет использовать локальный InterBase for Windows 4.0 (это локальный SQL-сервер, входящий в поставку), то его приложение безо всяких изменений будет работать и в составе большой системы с архитектурой клиент-сервер. 1.6 Некоторые особенности Delphi Локальный сервер InterBase - следует заметить, что этот инструмент предназначен только для автономной отладки приложений. В действительности он представляет из себя сокращенный вариант обработчика SQL-запросов InterBase, в который не включены некоторые возможности настоящего сервера InterBase. Отсутствие этих возможностей с лихвой компенсируется преимуществом автономной отладки программ.  Team Development Support - средство поддержки разработки проекта в группе. Позволяет существенно облегчить управление крупными проектами. Это сделано в виде возможности подключения такого продукта как Intersolve PVCS 5.1 непосредственно к среде Delphi. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 8 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1.7 Библиотека визуальных компонентов** Эта библиотека объектов включает в себя стандартные объекты построения пользовательского интерфейса, объекты управления данными, графические объекты, объекты мультимедиа, диалоги и объекты управления файлами, управление DDE и OLE.  **1.8 Структурное объектно-ориентированное программирование**  Delphi использует структурный объектно-ориентированный язык (Object Pascal), который сочетает с одной стороны выразительную мощь и простоту программирования, характерную для языков 4GL, а с другой стороны эффективность языка 3GL. Программисты немедленно могут начать производить работающие приложения, и им не придется для этого изучать особенности программирования событий в Windows. Delphi полностью поддерживает передовые программные концепции включая инкапсуляцию, наследование, полиморфизм и управление событиями. 1.9 Поддержка OLE 2.0, DDE и VBX Это очень важная особенность для разработчиков в среде Windows, поскольку в уже существующие Windows-приложения программист может интегрировать то, что разработает при помощи Delphi. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 9 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2. Специальная часть****2.1 Постановка задачи** Функции ПП  Данный программный продукт выводит изображение на форму средствами графической библиотеки OpenGL. Во время работы приложения происходит три этапа:   1. Загрузка текстуры (картинки) из файла; 2. Подготовка изображения, перестановка цветовой палитры (особенность OpenGL); 3. Вывод созданной текстуры на полигон и прорисовка в своей очереди.   **Входные и выходные данные.**  Входными данными в данном случае будут действия производимые пользователем на клавиатуре (управление движением автомобиля).  Выходные данные отображение автомобиля вследствие изменившихся координат, скорости столкновение и разгон. 2.2 Минимальные технические требования и ОС  * Процессор Intel (R) Pentium (R) III 900 MHz или выше; * ОЗУ 32MB SDRAM или выше; * Видео карта с поддержкой OpenGL v. 1.3.4582 т. е. Подойдет даже встроенная но главное с установленными драйверами; * Звуковая карта AC’ 97 или лучше; | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 10 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2.3 Проектирование пп** Главная форма предназначена для вывода графических действий. Во время работы приложения появляется пункты меню в которых можно выбрать четыре действия. Данная форма называется SpeedGL и на ней находится два компонента:   1. MediaPlayer; 2. ActionList.   **MediaPlayer**  Служит для загрузки звуков (музыки) формата \*.mp3, \*.wav и в дальнейшем проигрывания ее.  **ActionList**  Служит для выполнения кода на ту клавишу на которую у него прописанно действие. Пример: действие на сочетание клавиш Alt+F4. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 11 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2.4 Основные алгоритмы для событий на форме и компонентах** Самый главный алгоритм происходит на форме на событии OnPaint;  Событие OnPaint происходит примерно каждые 0,010 сек. Оно вызывается специальным таймером, который подключается из модуля mmsystem. Если произошло событие OnPaint то происходят следующие действия:   * Проверка главного условия нахождения в меню;   Проверка состоит в следующем:  если переменная Selected\_Menu\_item=’Start’ то тогда игра начнется и автоматически будут загружены нужные текстуры и начнет прорисовываться загруженный уровень, в других случаях если:  Selected\_Menu\_item=’’;  Selected\_Menu\_item=’Car’;  Selected\_Menu\_item=’Actors’;  Selected\_Menu\_item=’Exit’, то это все относится к меню и небольшим манипуляциям с текстурами. 3. разработка 3.1 Кратко описания программной среды разработки  Разработка велась в программной среде Delphi 5 т. к. это наиболее современная и наиболее совместимая с будущими версиями Delphi. Начиная с этой версии Delphi и выше велась поддержка OpenGL.  **3.2 Реализация основных алгоритмов в среде разработки**  В данной программе реализован механизм зацикливания на событии таймера. Когда происходит это событие специальный алгоритм просматривает состояние клавиатуры и если нажата клавиша на которую у него прописано действие происходит определённые вычисления. | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 12 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| После выполнения процедуры таймера автоматически вызывается процедура перерисовки окна OnPaint. В этой процедуре основываясь на изменённых и подсчитанных данных в процедуре таймера прорисовываются по соответствующим координатам текстуры. Так происходит взаимодействие пользователя с программой. Так как событие таймера происходит достаточно часто примерно 100 раз в секунду то прорисовка окна и реакция программы достаточно высока. Конечно здесь не маловажную роль играет процессор т. к. чем выше его частота тем чаще и быстрее будет происходить событие таймера. 4. Документирование **4.1 Инструкция пользователя по пп**  Так как в моём проекте речь идет про игру то я опишу основные функциональные клавиши управления.  **Клавиши действующие в меню:**   * «Стрелка вверх» и «Стрелка вниз» передвижение курсора выделения вверх и вниз; * «Esc» выход из игры и возвращение в главное меню; * «Стрелка вправо» и «Стрелка влево» действуют если находишься в подменю выбора машины; при нажатии «Esc» выбирается та машина которая была показана; * «Enter» используется для входа в подменю и начала игры; | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 13 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Клавиши действующие в игре:**  * «Стрелка вверх» и «Стрелка вниз» разгон и торможение автомобиля; * «Стрелка влево» и «Стрелка вправо» поворот налево и поворот направо; * «N» (англ.) используется для временного ускорения; * «P» или «Pause» пауза в игре, при повторном нажатии пауза отключается; * «Esc» возвращение в главное меню;  **Клавиши действующие и в меню и в игре:**«K» (англ.) используется для отображения координат нахождения курсора мыши на форме, результат выводится в заголовок формы;«F» (англ.) используется для отображения количества кадров в секунду, результат выводится в заголовок формы;  * «A» (англ.) используется для проверки поддержки OpenGL, если поддерживается то в заголовке формы будет написано версия OpenGL и видео карта через которую осуществляется вывод; | | | | | | |
|  |  |  |  |  | КП. 2203. 04. 35. ПЗ | Лист |
|  |  |  |  |  | 14 |
| Изм | Лист | № докум | Подп. | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **5. тестирование** 5.1 тестирование функций пп  Выполнение функций в данном проекте идет отличное так как небольшой сбой бал бы сразу замечен при работе программы.  **5.2 тестирование надежности**  Функция по считыванию клавиш продуманна таким образом что только реагирует на те клавиши на которые есть код выполнения, остальные игнорируются. | | | | | | |
|  |  |  |  |  | **КП. 2203. 04. 35. ПЗ** | Лист |
|  |  |  |  |  | 15 |
| Изм | Лист | № докум | Подп. | Дата |

приложение

**Событие таймера:**

procedure FNTimeCallBack(uTimerID, uMessage: UINT;dwUser, dw1, dw2: DWORD) stdcall;

var

word:DWORD;

begin

Inc(CountTickTimer);

if (((PressedKey[19]=true)or

(PressedKey[80]=true))and(CountTickTimer>100))and(CountFuel>0) then

if(not Pause) then begin

Pause:=true;

CountTickTimer:=0;

end

else

begin

Pause:=false;

CountTickTimer:=0;

end;

if not Pause then begin

if Selected\_Menu\_item='Start' then //Нажата клавиша Start

begin

if SpeedGL.MediaPlayer1.FileName<>'Sound\Race.mp3' then begin

SpeedGL.MediaPlayer1.FileName:='Sound\Race.mp3';

SpeedGL.MediaPlayer1.Open;

SpeedGL.MediaPlayer1.Play;

end;

if (PressedKey[27]=true)and(CountTickTimer>100) then

begin

Selected\_Menu\_item:='';

SpeedGL.MediaPlayer1.Stop;

SpeedGL.MediaPlayer1.Close;

SpeedGL.MediaPlayer1.FileName:='';

Menu\_item:=2;

CountTickTimer:=0;

Speed:=0.0;

sound:=false;

CountFuel:=8;

CountKM:=0;

end;

//торможение

if PressedKey[40]=true then

begin

Speed:=Speed+0.00002;

GasDeley:=1;

end;

if (CountFuel>0)and(CountKM<20) then if PressedKey[38]=true then Speed:=Speed-0.000005;//разгон

if (PressedKey[39]=true)and(Speed<-0.00001) then X\_KordCar:=X\_KordCar+0.00005-Speed/5;//поворот вправо с учётом скорости

if (PressedKey[37]=true)and(Speed<-0.00001) then X\_KordCar:=X\_KordCar-0.00005+Speed/5;//поворот влево с учётом скорости

if (PressedKey[40]=False)and(PressedKey[38]=False) then begin Speed:=Speed+0.000003; GasDeley:=1; end;//торможение инерцией

if Speed>0 then Speed:=0;//скорость торможения

//////////////////////////////////////////////////////////

if (Gas)and(CountFuel<>0) then

begin

GasDeley:=GasDeley+0.0005;

if GasDeley>=1 then Speed:=Speed+0.00005 else Speed:=-0.025;

if Speed>-0.01 then

begin

Speed:=-0.01;

Gas:=false;

GasDeley:=1;

end;

end

else

if Speed<-0.01 then Speed:=-0.01;//ограничение скорости

//////////////////////////////////////////////////////////

Y\_Kord:=Y\_Kord+Speed;//наращивание скорости

if EnymiGoToWay then begin

Y\_Enymi:=Y\_Enymi+Speed+0.0085;

X\_Enymi:=X\_Enymi+(0.001\*LR);

if X\_Enymi>=Right\_death\_dot then LR:=-1;

if X\_Enymi<=Left\_death\_dot then LR:=1;

if (Y\_Enymi>1.2)or(Y\_Enymi<-1.2)then EnymiGoToWay:=false;

end;

if FuelGoToWay then

Y\_KordFuel:=Y\_KordFuel+Speed; //Перемещение элементов

if (Speed<>0)and(Speed<-0.00002)and(CountFuel>0) then

CountFuel:=CountFuel+MinusFuel/(Speed\*50); //Уменьшение топлива

if (CountFuel<=0)or(CountKM>=20) then begin Gas:=false; GasDeley:=1; Speed:=Speed+0.000003; end;

if (Speed<-0.005)and(MinusFuel=0.00002) then MinusFuel:=MinusFuel\*10 else

if MinusFuel=0.0002 then MinusFuel:=MinusFuel/10;

if Y\_Kord<=-2.0 then Y\_Kord:=0;//перестановка кадров

if X\_KordCar>=Right\_death\_dot then

begin

X\_KordCar:=Right\_death\_dot;//ограничение справа

if Speed<-0.001 then begin Speed:=Speed+0.00002; GasDeley:=1; end;//Уменьшение скорости при прикосновении к бардюру справап

end;

if X\_KordCar<=Left\_death\_dot then

begin

X\_KordCar:=Left\_death\_dot;//ограничение слева

if Speed<-0.001 then begin Speed:=Speed+0.00002; GasDeley:=1; end;//Уменьшение скорости при прикосновении к бардюру слева

end;

end;

end

else

if (PressedKey[27]=true)and(CountTickTimer>100) then

begin

Selected\_Menu\_item:='';

Menu\_item:=2;

CountTickTimer:=0;

Speed:=0.0;

Pause:=false;

sound:=false;

CountFuel:=8;

CountKM:=0;

end;

if Selected\_Menu\_item='' then //Нажата клавиша Esc в игре или ничего не выбрано

begin

if SpeedGL.MediaPlayer1.FileName<>'Textures\Video\SL\_Start\_Logo.avi' then

if SpeedGL.MediaPlayer1.FileName<>'Sound\Main\_Menu.mp3' then begin

SpeedGL.MediaPlayer1.FileName:='Sound\Main\_Menu.mp3';

SpeedGL.MediaPlayer1.Open;

SpeedGL.MediaPlayer1.Play;

end;

if CountTickTimer>100 then begin

if PressedKey[27]=true then

begin

SpeedGL.MediaPlayer1.Stop;

SpeedGL.MediaPlayer1.Close;

SpeedGL.MediaPlayer1.FileName:='';

Selected\_Menu\_item:='Exit';

CountTickTimer:=0;

end;

if PressedKey[40]=true then

begin

Inc(Menu\_item);

PlaySound('Sound\Select\_MainMenu.wav',0,SND\_ASYNC);

CountTickTimer:=0;

end; //вниз

if PressedKey[38]=true then

begin

Dec(Menu\_item);

PlaySound('Sound\Select\_MainMenu.wav',0,SND\_ASYNC);

CountTickTimer:=0;

end;//вверх

if CountTickTimer>1000 then CountTickTimer:=110;

end;

if Menu\_item>4 then Menu\_item:=1;//ограничение

if Menu\_item<1 then Menu\_item:=4;// в пределах 4 пунктов

if PressedKey[13]=true then begin

PlaySound('Sound\Choose\_MainMenu.wav',0,SND\_ASYNC);

case Menu\_item of

1:begin

Selected\_Menu\_item:='Start';

Score:=0;

FuelGoToWay:=false;

NextKM\_GenerFuel:=1;

CountKM:=0;

CountFuel:=8;

CountNitro:=0;

MinusFuel:=0.00002;

Y\_KordFuel:=1.5;

FuelGoToWay:=false;

GasDeley:=1;

Gas:=false;

NextKmEnymy:=0.3;

Y\_Enymi:=1.2;

end;

2:begin Selected\_Menu\_item:='Car'; end;

3:begin Selected\_Menu\_item:='Actors'; end;

4:begin Selected\_Menu\_item:='Exit'; end;

end;

end;

end;

if CountTickTimer>100 then begin

if Selected\_Menu\_item='Actors' then //Нажата клавиша Sound

begin

if SpeedGL.MediaPlayer1.FileName<>'Sound\Actor.mp3' then begin

SpeedGL.MediaPlayer1.FileName:='Sound\Actor.mp3';

SpeedGL.MediaPlayer1.Open;

SpeedGL.MediaPlayer1.Play;

end;

if PressedKey[27]=true then

begin

Selected\_Menu\_item:='';

SpeedGL.MediaPlayer1.Stop;

SpeedGL.MediaPlayer1.Close;

SpeedGL.MediaPlayer1.FileName:='';

CountTickTimer:=0;

end;

end;

if Selected\_Menu\_item='Car' then //Нажата клавиша Car

begin //Car

if SpeedGL.MediaPlayer1.FileName<>'Sound\Garage.mp3' then begin

SpeedGL.MediaPlayer1.FileName:='Sound\Garage.mp3';

SpeedGL.MediaPlayer1.Open;

SpeedGL.MediaPlayer1.Play;

end;

if PressedKey[27]=true then

begin

Selected\_Menu\_item:='';

SpeedGL.MediaPlayer1.Stop;

SpeedGL.MediaPlayer1.Close;

SpeedGL.MediaPlayer1.FileName:='';

CountTickTimer:=0;

OldSelectMenyItem:='Car';

end;

if PressedKey[37]=true then

begin

Dec(CarMenuItem);

CountTickTimer:=0;

LeftButton:=true;

PlaySound('Sound\Select\_CarMenu.wav',0,SND\_ASYNC);

end;

if PressedKey[39]=true then

begin

Inc(CarMenuItem);

CountTickTimer:=0;

RightButton:=true;

PlaySound('Sound\Select\_CarMenu.wav',0,SND\_ASYNC);

end;

if CarMenuItem>4 then CarMenuItem:=1;

if CarMenuItem<1 then CarMenuItem:=4;

end; //Car

end;//CountTickTimer

if Selected\_Menu\_item='Exit' then //Нажата клавиша Exit в гланом меню

begin

SpeedGL.MediaPlayer1.Free;

Application.Terminate;

Application.ProcessMessages;

end;

InvalidateRect(SpeedGL.Handle, nil, False);//перерисовка

if (Symvol=0)and

(PressedKey[78]=true)and

(CountNitro>0)and

(GasDeley=1) then

begin

Gas:=true;

CountNitro:=CountNitro-1;

GasDeley:=0;

PlaySound('Sound\Start\_Nitro.wav',0,SND\_ASYNC);

end;

end;

**событие OnPaint:**

procedure TSpeedGL.FormPaint(Sender: TObject);

var

ps:TPaintStruct;

begin

BeginPaint(dc,ps);

if Selected\_Menu\_item='' then

begin

case Menu\_item of

1:begin LeftTiresMeny:=-0.849; RightTiresMeny:=0.8; TopTiresMeny:=0.76; end;

2:begin LeftTiresMeny:=-1.1; RightTiresMeny:=1.0; TopTiresMeny:=0.4; end;

3:begin LeftTiresMeny:=-0.8; RightTiresMeny:=0.8; TopTiresMeny:=0.0; end;

4:begin LeftTiresMeny:=-0.8; RightTiresMeny:=0.8; TopTiresMeny:=-0.58; end;

end;

if OldSelectMenyItem='Car' then begin

if RemoveMeny>=0.0 then begin RemoveMeny:=0.0; OldSelectMenyItem:='' end else

RemoveMeny:=RemoveMeny+0.018;

glCallList(Background);

glPushMatrix;

glTranslatef(RemoveMeny,0.0,0.0);

glCallList(StringMeny);

glPopMatrix;

end

else begin

glCallList(Background);//Загрузка фона

glCallList(StringMeny);//Загрузка надписей в меню

end;

if (Selected\_Menu\_item<>'Car')and(RemoveMeny=0.0) then begin

Angel:=Angel+1;

if Angel=360 then Angel:=0;

glCallList(8);//Загрузка колеса

glPushMatrix;

glScalef(0.5,0.66,0.5);

glTranslatef(LeftTiresMeny,TopTiresMeny,0.0);

glRotatef(Angel,0.0,0.0,1.0);

glCallList(Tires);

glPopMatrix;

glPushMatrix;

glScalef(0.5,0.66,1.0);

glTranslatef(RightTiresMeny,TopTiresMeny,0.0);

glRotatef(Angel,0.0,0.0,1.0);

glCallList(Tires);

glPopMatrix;

end;

end;

if Selected\_Menu\_item='Start' then //Нажата клавиша Start игра началась

begin //Start

// glClear(GL\_COLOR\_BUFFER\_BIT); //медленное стирание

glPushMatrix;

glTranslatef(0.0,Y\_Kord,0.0);

glCallList(WayDesert\_1);

glPopMatrix;

glPushMatrix;

glTranslatef(0.0,Y\_Kord+1.99,0.0);

glCallList(WayDesert\_1);

glPopMatrix;

if not Pause then

if Speed<>0 then CountKM:=CountKM+abs(Speed/27);//количесто пройденых километров

// Caption:=FloatToStrF(CountKM,ffNumber,6,3);

//Генерирование Бензина или Нитро на дороге

if (CountKM>NextKM\_GenerFuel)and(FuelGoToWay=False)and(speed<>0) then

begin

if Random(2)=1 then X\_KordFuel:=Random\*0.5 else X\_KordFuel:=-Random\*0.5;

FuelGoToWay:=true;

Y\_KordFuel:=1.5;

Inc(NextKM\_GenerFuel,1);

if not FuelOrNitro then FuelOrNitro:=true else FuelOrNitro:=false;

end;

if (FuelGoToWay)and(Y\_KordFuel<-1.2) then FuelGoToWay:=False;

//Прорисовка Бензина или Нитро

if not KordObjectItems(X\_KordFuel,Y\_KordFuel,0.05,0.065,X\_KordCar,Y\_KordCar,CarXPogresh,CarYPogresh) then

begin // KordObject

glPushMatrix;

glTranslatef(X\_KordFuel,Y\_KordFuel,0.0);

if FuelOrNitro then glCallList(ItemFuelList) else glCallList(ItemNitroList);

glPopMatrix;

end//KordObject

else

begin

Y\_KordFuel:=1.5;

FuelGoToWay:=False;

PlaySound('Sound\Get\_Nitro\_Fuel.wav',0,SND\_ASYNC);

if FuelOrNitro then if CountFuel<20 then CountFuel:=CountFuel+1;

if not FuelOrNitro then if CountNitro<=10 then CountNitro:=CountNitro+1;

end;

//////////////////////////////////////////////////////

{Caption:=FloatToStrF(CountFuel,ffNumber,6,3)+' '+

FloatToStrF(CountNitro,ffNumber,6,3)+' '+

FloatToStrF(CountKM,ffNumber,6,3)+' '+

FloatToStrF(GasDeley,ffNumber,6,3);}

//////////////////////////////////////////////////////

if (CountKM>NextKmEnymy)and(EnymiGoToWay=False)and(speed<>0) then

begin

NumbrCar:=random(9)+1;

EnymiGoToWay:=true;

NextKmEnymy:=NextKmEnymy+0.1;

Y\_Enymi:=1.2;

end;

if (EnymiGoToWay)and(Y\_Enymi<-1.2) then EnymiGoToWay:=False;

if not KordObjectEnimy(X\_Enymi,Y\_Enymi,0.08,0.130,X\_KordCar,Y\_KordCar,CarXPogresh-0.05,CarYPogresh-0.05) then

begin // KordObject

glPushMatrix;

glTranslatef(X\_Enymi,Y\_Enymi,0.0);

glCallList(EnymiCars[NumbrCar]);

glPopMatrix;

end//KordObject

else

begin

Y\_Enymi:=1.2;

EnymiGoToWay:=False;

GasDeley:=1;

Gas:=false;

Speed:=-0.001;

PlaySound('Sound\Denger\_Sound.wav',0,SND\_ASYNC);

if CountFuel>0 then CountFuel:=CountFuel-1;

if CountFuel<0 then CountFuel:=0;

end;

//рисование машины

if ChooseCar='ShotWarKiller' then begin

glPushMatrix;

glTranslatef(X\_KordCar,Y\_KordCar,0.0);

glCallList(WarKiller);

glPopMatrix;

end;

if ChooseCar='ShotFlash' then begin

glPushMatrix;

glTranslatef(X\_KordCar,Y\_KordCar,0.0);

glCallList(Flash);

glPopMatrix;

end;

if ChooseCar='ShotGembic' then begin

glPushMatrix;

glTranslatef(X\_KordCar,Y\_KordCar,0.0);

glCallList(Gembic);

glPopMatrix;

end;

if ChooseCar='ShotGlass' then begin

glPushMatrix;

glTranslatef(X\_KordCar,Y\_KordCar,0.0);

glCallList(Glass);

glPopMatrix;

end;

//Проигрыш и выигрышь в игре

if (CountFuel<=0)and(CountKM<20) then begin glCallList(GameOver); if not sound then sound:=true; end;

if CountKM>=20 then begin glCallList(YouWin); if not sound then sound:=true; end;

if ((CountFuel<=0)or(CountKM>=20))and(Speed>=0)then begin

glPushMatrix;

glTranslatef(0.0,-0.4,0.0);

glCallList(PressEsc);

glPopMatrix

end;

//Игровое меню

glPushMatrix;

glTranslatef(0.7,0.58,0.0);

glScalef(1.5,1.5,1.0);

glCallList(GameMenu);

glPopMatrix;

//Надписи на меню

glPushMatrix;

glTranslatef(0.64,0.68,1.0);

glCallList(FontList);

glPopMatrix;

//

if Pause then glCallList(PauseToGame);

end; //Start

if Selected\_Menu\_item='Car' then //Нажата клавиша Car

begin //Car

if RemoveMeny<=-0.5 then RemoveMeny:=-0.5 else RemoveMeny:=RemoveMeny-0.018;

glCallList(Background);

glPushMatrix;

glTranslatef(RemoveMeny,0.0,0.0);

glCallList(StringMeny);

glPopMatrix;

if RemoveMeny<=-0.5 then

begin

//Меню

glPushMatrix;

glTranslatef(0.5,0.2,0.0);

glCallList(SubMenuInCar);

glPopMatrix;

DeleyButtonInCarMeny;

//Выбирается машина

case CarMenuItem of

1:begin

//ShotWarKiller

glPushMatrix;

glTranslatef(0.5,0.12,0.0);

glCallList(ShotWarKiller);

glPopMatrix;

ChooseCar:='ShotWarKiller';

CarXPogresh:=XPogreshWarKiller;

CarYPogresh:=YPogreshWarKiller;

end;

2:begin

//ShotFlash

glPushMatrix;

glTranslatef(0.5,0.12,0.0);

glCallList(ShotFlash);

glPopMatrix;

ChooseCar:='ShotFlash';

CarXPogresh:=XPogreshFlash;

CarYPogresh:=YPogreshFlash;

end;

3:begin

//ShotGembic

glPushMatrix;

glTranslatef(0.5,0.12,0.0);

glCallList(ShotGembic);

glPopMatrix;

ChooseCar:='ShotGembic';

CarXPogresh:=XPogreshFlash;

CarYPogresh:=YPogreshFlash;

end;

4:begin

//ShotGlass

glPushMatrix;

glTranslatef(0.5,0.12,0.0);

glCallList(ShotGlass);

glPopMatrix;

ChooseCar:='ShotGlass';

CarXPogresh:=XPogreshFlash;

CarYPogresh:=YPogreshFlash;

end;

end;//Case

end;//RemoveMeny

end; //Car

if Selected\_Menu\_item='Actors' then //Нажата клавиша Actors

begin

glCallList(Actor);

end;

if Selected\_Menu\_item='Exit' then //Нажата клавиша Exit

begin

end;

SwapBuffers(DC);

EndPaint(DC,ps);

{if Selected\_Menu\_item='Start'then begin

Canvas.Brush.Color:=clWhite;

Canvas.TextOut(500,40,'Бензин: '+FloatToStrF(CountFuel,ffNumber,6,3)+'л');

Canvas.TextOut(500,55,'Нитро: '+FloatToStrF(CountNitro,ffNumber,6,3)+'кг/см');

Canvas.TextOut(500,70,'К.метры: '+FloatToStrF(CountKM,ffNumber,6,3)+'км');

end;}

//

Caption:=StatisticCaption;

//Поддержка аппаратного ускорителя и версия OpenGL клавиша 'a'

if PressedKey[65]=true then

Caption:='OpenGL: '+

StrPas(glGetString(GL\_VERSION))+

' '+

StrPas(glGetString(GL\_VENDOR))+

' '+

StrPas(glGetString(GL\_RENDERER));

if PressedKey[70]=true then //Показывать Fps -- клавиша 'f'

begin

newCount:=GetTickCount;

Inc(frameCount);

If abs(newCount - lastCount) > 1000 then

begin

StatisticCaption:='FPS: '+FloatToStr(frameCount);

lastCount := newCount;

frameCount := 0;

end;

end;

if PressedKey[75]=true then

StatisticCaption:='X='+FloatToStrF(xpos,ffNumber,4,3)+

' '+'Y='+FloatToStrF(ypos,ffNumber,4,3);

if (PressedKey[70]=false)and

(PressedKey[75]=false)and

(PressedKey[65]=false)then StatisticCaption:='SpeedLine';

end;

procedure TSpeedGL.FormDestroy(Sender: TObject);

begin

//glDeleteLists(1,1);

timeKillEvent(uTimerID);

wglMakeCurrent (0, 0);

wglDeleteContext (hrc);

ReleaseDC (Handle, DC);

end;

procedure TSpeedGL.FormMouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

begin

//////////////////////////////////////////////

xpos:=2\*X/ClientWidth-1;

ypos:=2\*(ClientHeight-Y)/ClientHeight-1;

//////////////////////////////////////////////

if Selected\_Menu\_item='' then begin //Selected\_Menu\_item

if (xpos>-0.3)and(ypos>0.4)and

(xpos<0.3)and(ypos>0.4)and

(xpos<0.3)and(ypos<0.6)and

(xpos>-0.3)and(ypos<0.6)then begin

if Menu\_item<>1 then

PlaySound('Sound\Select\_MainMenu.wav',0,SND\_ASYNC);

Menu\_item:=1;

end;

if (xpos>-0.3)and(ypos>0.15)and

(xpos<0.3)and(ypos>0.15)and

(xpos<0.3)and(ypos<0.35)and

(xpos>-0.3)and(ypos<0.35)then begin

if Menu\_item<>2 then

PlaySound('Sound\Select\_MainMenu.wav',0,SND\_ASYNC);

Menu\_item:=2;

end;

if (xpos>-0.3)and(ypos>-0.1)and

(xpos<0.3)and(ypos>-0.1)and

(xpos<0.3)and(ypos<0.1)and

(xpos>-0.3)and(ypos<0.1)then begin

if Menu\_item<>3 then

PlaySound('Sound\Select\_MainMenu.wav',0,SND\_ASYNC);

Menu\_item:=3;

end;

if (xpos>-0.3)and(ypos>-0.5)and

(xpos<0.3)and(ypos>-0.5)and

(xpos<0.3)and(ypos<-0.3)and

(xpos>-0.3)and(ypos<-0.3)then begin

if Menu\_item<>4 then

PlaySound('Sound\Select\_MainMenu.wav',0,SND\_ASYNC);

Menu\_item:=4;

end;

{if Selected\_Menu\_item='Start' then

begin

if

end;}

end; //Selected\_Menu\_item

end;