|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | Инженерный бизнес и менеджмент |
| КАФЕДРА | Промышленная логистика |

**Домашнее задание**

**по дисциплине   
«Парадигмы и конструкции языков программирования»**

***НА ТЕМУ:***

|  |
| --- |
| ***Создание игр на языке C++*** |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИБМ3-34Б |  |  |  | Н.Новачлы |
|  | (группа) |  | (подпись, дата) |  | (И.О. Фамилия) |
| Преподаватель |  |  |  |  | Гапанюк Ю.Е.  Нардид А.Н. |
|  |  |  | (подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |  |

Москва, 2024 г.

Оглавление

[Введение 3](#_Toc184669886)

[1.Основыне этапы при подготовке к написаю игры 4](#_Toc184669887)

[3.Скелет приложения 13](#_Toc184669888)

[4.Моделирование 16](#_Toc184669889)

[5. Популярные игры, созданные на C++ 19](#_Toc184669890)

[Заключение 21](#_Toc184669891)

# Введение

 C ++ это как волшебный инструмент для разработки игр поскольку он позволяет создавать графику и механику. Независимо от того, нацелены ли вы на блокбастер, такой как "Counter-Strike", или на инди-хит, такой как "Undertale", освоение C ++ очень важно. В этой реферате мы рассмотрим C ++ в разработке игр, но учтем, что существует множество языков, таких как C #, Java и другие также используются в области разработки игр, и у каждого есть свои уникальные преимущества.

Создание мира азарта и веселья внутри компьютера похоже на разработку игр! Чтобы оживить виртуальные миры, эта многопрофильная профессия объединяет элементы дизайна, программирования, искусства, аудио и многого другого. Они начинают с отличной идеи, придают ей потрясающий вид с помощью картинок и звуков, пишут компьютерные инструкции, чтобы все это заработало, а затем следят за тем, чтобы все прошло гладко. Когда все готово, перед игроками словно открывается дверь в новое приключение.

Разработка игр часто кажется захватывающим путешествием, и в этой ситуации [C ++](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/c-plus-plus/) становится надежным спутником разработчика. C ++ стал наиболее подходящим языком для разработки игр, требующих высочайшей производительности, потому что C ++ в основном славится своей скоростью и гибкостью, а также возможностью прямого взаимодействия с оборудованием. Создаете ли вы визуально потрясающий 3D мир или дорабатываете игровой процесс для обеспечения бесперебойного взаимодействия, C ++ предоставляет вам инструменты и возможности для воплощения ваших идей в захватывающий игровой процесс.

# 1.Основыне этапы при подготовке к написаю игры

Разработка игр предполагает создание интерактивного взаимодействия с помощью программного обеспечения. Понимание основных концепций технически очень важно.

Игровой цикл: Это основная структура игры, в которой игра непрерывно обрабатывает вводимые пользователем данные, обновляет состояние игры и визуализирует игровой мир.

Рендеринг: Рендеринг обычно включает отображение графики на экране определенных устройств. Сюда входят методы рисования 2D или 3D графика, работа с анимацией и управление визуальными эффектами.

Физика: Игры часто имитируют физические взаимодействия, такие как гравитация, столкновения и движение объектов. Понимание основных физических концепций помогает создавать реалистичную игровую механику.

Обработка входных данных: Игры получают входные данные от игроков, такие как клавиатура, мышь или входы для контроллера. Эффективная обработка этих входных данных необходима для создания адаптивного игрового процесса.

2. Изучите игровой движок

Игровые движки предоставляют инструменты и библиотеки для упрощения разработки игр. Некоторые популярные движки:

Unity: Представьте[Unity](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/game-development-with-unity-introduction/) в роли дружелюбного соседа, который всегда готов протянуть руку помощи. В общих чертах там написано [C #](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/introduction-to-c-sharp/) и хорошо известен тем, что с ним легко ознакомиться, что в конечном итоге делает его отличным выбором для разработчиков, которые хотят создавать игры, которыми можно пользоваться на различных устройствах.

Unreal Engine: Unreal Engine подобен техническому гуру, разбирающемуся в визуальных эффектах. По сути, в дополнение к C ++ предлагается язык визуальных сценариев, который называется Blueprints, что также делает его очень популярным выбором для разработчиков, которые хотят создавать визуально привлекательные игры.

CryEngine: CryEngine похож на художника из группы, который известен своими потрясающими визуальными эффектами и вниманием к деталям. Как правило, это отличный выбор для разработчиков, которые в основном хотят сосредоточиться на создании визуально привлекательных игр.

Lumberyard: Lumberyard похож на универсальный инструментарий, который в основном предлагает широкий спектр инструментов и функций для разработки игр. Это очень надежный и хороший вариант для разработчиков игр, которые хотят масштабировать свои проекты благодаря его взаимодействию с [Amazon Web Services (AWS)](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/aws-tutorial/).

Начните с ознакомления с интерфейсом движка, основными функциями и рабочим процессом. Оба движка имеют обширную документацию и руководства для начинающих.

3. Освоите C ++ для разработки игр

[C ++](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/cpp-tutorial/) является широко используемым языком в разработке игр благодаря своей производительности и гибкости. Чтобы освоить C ++ для разработки игр:

Управление памятью: это похоже на навык управления инвентарем вашего персонажа — речь идет об эффективном обращении с ресурсами. Понимание [указателей](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/cpp-pointers/), распределение памяти, [утечки памяти](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/what-is-memory-leak-how-can-we-avoid/) can поможет вашей игре работать бесперебойно, не тратя впустую ценные ресурсы.

Объектно-ориентированное программирование (ООП): Думайте о [ООП](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/object-oriented-programming-in-cpp/) как о специальных движениях вашего персонажа. По сути, это основа структурирования вашего кода и данных таким образом, чтобы это имело смысл для вашей игры. Освоение нескольких концепций OOPS, таких как классы, Объекты, [наследование](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/inheritance-in-c/), и [полиморфизм](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/cpp-polymorphism/)предоставляет вам мощные инструменты для создания сложных игровых систем.

Оптимизация производительности: Это похоже на ловкость и скорость вашего персонажа. Чтобы гарантировать бесперебойную работу вашей игры, вам в основном необходимо оптимизировать свой код. Такие методы, как профилирование, оптимизация алгоритмов и управление ресурсами, подобны повышению уровня ваших навыков, чтобы ваша игра работала наилучшим образом.

Структуры данных и алгоритмы (DSA): Это похоже на стратегии и тактики, которые ваш персонаж использует в сражениях. Использование [DSA](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/learn-data-structures-and-algorithms-dsa-tutorial/) помогает эффективно решать сложные задачи, что очень важно для создания продвинутых игровых механик и систем.

Чтобы перейти на следующий уровень владения C ++, пожалуйста, посетите сайт- [Освоите программирование на C ++ - от начального до продвинутого](https://translated.turbopages.org/proxy_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/gfgcdn.com/tu/Q6B/)

4. Изучайте математику для разработки игр

Математика необходима для различных аспектов разработки игр:

Линейная алгебра: Это похоже на волшебство, стоящее за перемещением и изменением объектов в трехмерном пространстве. Он помогает вам с такими вещами, как увеличение или уменьшение размеров объектов (масштабирование), их вращение (rotation) или перемещение из одного места в другое (transition).

Тригонометрия: Рассматривайте это как руководство по пониманию углов и поворотов. Это как иметь компас, который помогает вам определять, где что находится как в 2D, так и в 3D мирах.

Математический анализ: это как секретный соус для создания реалистичной физики и продвинутых визуальных эффектов. Как правило, это помогает вам понять, как все меняется со временем, что очень важно для придания играм ощущения реальности.

Изучив эти темы и навыки, вы сможете создавать игры, в которые в принципе интересно играть и которые заставят игроков почувствовать себя реалистами.

5. Изучите графическое программирование

Графическое программирование включает в себя создание визуальных элементов в играх. Ключевые области для изучения включают:

Графические API: это что-то вроде инструментов, которые помогают вашей игре взаимодействовать с графическим оборудованием вашего компьютера. По сути, они предоставляют способ создания визуальных элементов и эффектов в вашей игре.

Языки шейдеров: Думайте о них как о специальных языках, которые помогают вам создавать пользовательские визуальные эффекты. Как правило, они дают вам возможность придать вашей игре уникальный вид.

Методы рендеринга: это своего рода профессиональные приемы, позволяющие сделать визуальные эффекты вашей игры действительно эффектными. Методы рендеринга в основном включают в себя такие вещи, как то, как работает свет в вашей игре, как создаются тени, как двигаются игроки и как добавлять интересные эффекты, такие как размытие или свечение.

6. Специализируйтесь в области разработки игр

Разработка игр предлагает различные специализации, и вы можете выбрать специализацию в таких областях, как:

Программирование игрового процесса: по сути, эта роль заключается в том, чтобы оживить игру, работая над множеством вещей, таких как то, как и когда персонажи обычно перемещаются и как игроки взаимодействуют с игровым миром.

Программирование искусственного интеллекта: здесь основное внимание уделяется созданию разумного поведения персонажей, неподконтрольных игрокам, например, врагов или неигровые персонажи (NPC).

Графическое программирование: Эта роль связана с тем, чтобы игра выглядела великолепно, от создания классных визуальных эффектов до обеспечения бесперебойной работы.

Сетевое программирование: это включает в себя обеспечение хорошей работы игры, когда люди играют вместе онлайн, управление такими вещами, как взаимодействие игры с серверами и другими игроками.

Каждая роль очень важна для разработки хорошей игры, и все они объединяются, чтобы сделать конечный продукт приятным для игроков.

7. Практика работы с игровыми проектами:

Когда начинаешь заниматься разработкой игр, обычно лучше всего начинать с простых проектов. Начните с разработки небольших игр, чтобы вы могли изучить весь процесс, от разработки игры до тестирования и устранения любых возникающих проблем.

По мере продвижения в разработке игр вы повышаете уровень в своей любимой игре. Экспериментировать с различными типами игр, такими как платформеры, головоломки или шутеры, все равно что исследовать уникальные квесты в игровом мире. Каждый жанр или категория представляет собой свой собственный набор задач и добавляет глубины вашему путешествию. Создать сильное портфолио из разнообразных игр - это все равно что создать собственное игровое приключение, демонстрирующее ваш рост и преданность делу как разработчика игр.

8. Будьте в курсе тенденций отрасли

Разработка игр - быстро развивающаяся отрасль, и важно постоянно обновляться:

Следите за новостями отрасли: Мы всегда в курсе последних разработок игровых движков, технологий и передового опыта.

Читайте блоги и статьи: Следите за многочисленными блогами и веб-сайтами по разработке игр, чтобы узнать больше от экспертов отрасли и оставаться в курсе новых тенденций.

Участвуйте в онлайн-сообществах: Также присоединяйтесь к форумам по разработке игр, субредактам и группам в социальных сетях, чтобы общаться с другими разработчиками и делиться знаниями.

9. Создайте портфолио

Портфолио демонстрирует ваши навыки и проекты потенциальным работодателям. В индустрии разработки игр ваше портфолио подобно вашему резюме — оно необходимо для того, чтобы произвести отличное впечатление. Демонстрируя свои проекты и разнообразные навыки разработки игр, вы действительно можете выделиться из толпы. Это все равно что сказать потенциальным работодателям и клиентам (если вы работаете фрилансером): "Посмотрите, на что я могу!".

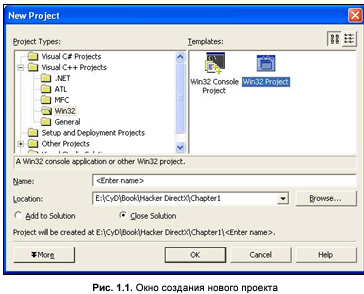
10. Создание сетей и совместная работа

Нетворкинг необходим для карьерного роста в сфере разработки игр:

Присоединяйтесь к сообществам разработчиков игр: чтобы развиваться в области разработки игр, присоединяйтесь к онлайн-форумам, посещайте встречи и вступайте в группы в социальных сетях. Общайтесь с множеством специалистов в этой области, узнавайте об их опыте и трудностях и всегда будьте в курсе тенденций отрасли.

Совместная работа над проектами: Работа с другими разработчиками над игровыми проектами может дать вам ценный опыт и поможет наладить сеть контактов в отрасли, что очень важно для роста в этой области.

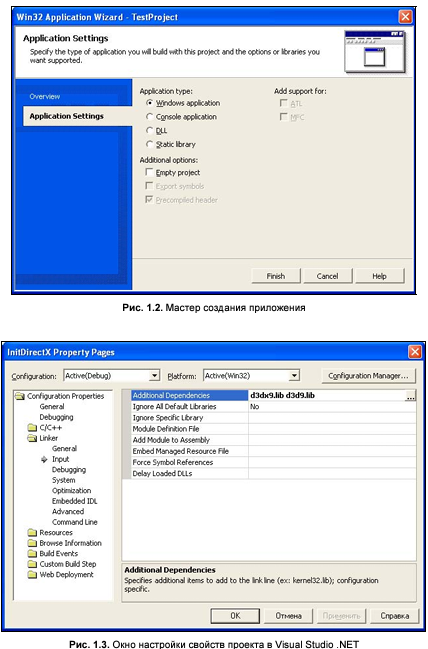
2.Подготовка к созданию приложения

Оптимизировать игры сложнее. Если в Demo-роликах используется только графика и звук и весь код можно реализовать на чистом языке С, то в играх без объектов обойтись сложно. Объекты необходимы, но это не значит, что вы должны использовать библиотеку MFC. Из возможностей этой библиотеки и WinAPI в приложении, применяющей DirectX, необходимо только создание окна. Больше ничего из API Windows нам не поможет, и будут использоваться только интерфейсы DirectX или функции языка С. Поэтому скелет программы не должен использовать объектов. Давайте создадим скелет будущего игрового приложения. Итак, создайте пустое приложение File | New | Project (Файл | Новый | Проект) и в появившемся окне выберите в дереве Project Type (Тип проекта) пункт Visual C++ Projects | Win32 (Проекты Visual C++ | Win32). Чтобы наши программы были небольшими и быстрыми, мы не будем использовать MFC, поэтому выбираем пункт Win32 Project (Проект Win32) (рис. 1.1). В поле Name (Имя) укажите имя проекта, а в раскрывающемся списке Location (Месторасположение) — путь, по которому будет сохранен проект

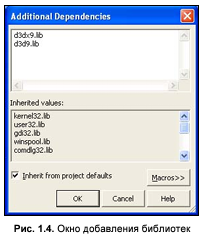
После нажатия кнопки OK перед нами открывается мастер создания приложения. В разделе Application Settings (Настройки приложения) (рис. 1.2) установите тип приложения, выбрав переключатель Windows application (Приложение Windows) в группе переключателей Application type (Тип приложения), а все прочие флажки должны остаться по умолчанию.

Теперь наше базовое приложение готово. Не забудьте в свойствах проекта подключить библиотеки Direct3D. Для этого выберите меню Project | Properties (Проект | Свойства). В дереве свойств выбираем раздел Configuration Properties | Linker | Input (Свойства конфигурации | Сборщик | Входящие).

Перед вами откроется окно, как на рис. 1.3.



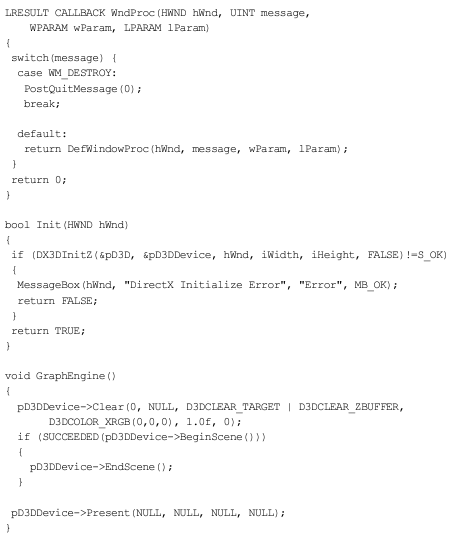
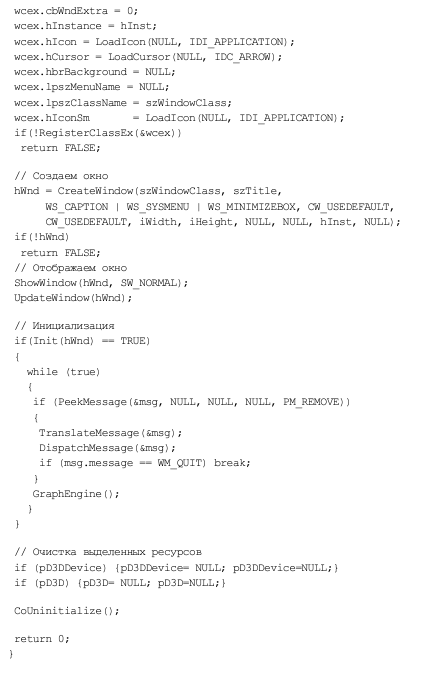
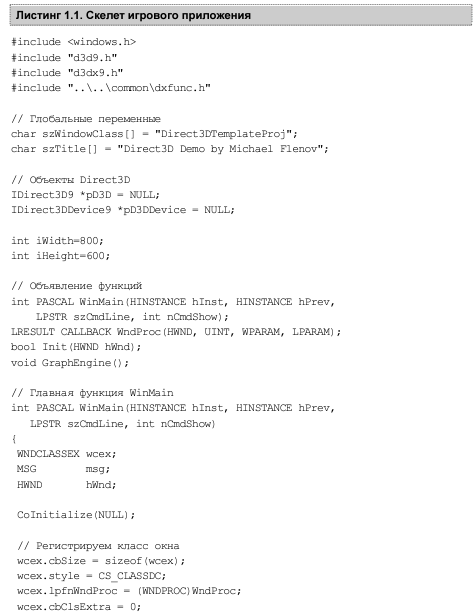
В строке Additional Dependencies (Дополнительные зависимости) необходимо указать библиотеки, которые нужно подключить во время сборки проекта. Выделите эту строку и щелкните по кнопке с изображением трех точек. Перед вами появится окно, в котором можно указать дополнительные библиотеки (рис. 1.4)



Для большинства проектов из данной книги необходимо как минимум указать библиотеки d3dx9.lib и d3d9.lib. Укажите их каждую в отдельной строке и нажмите ОК для сохранения изменений. Добавить библиотеки нужно для обеих конфигураций: Release и Debug. Последовательно выберите в выпадающем меню Configuration (Конфигурация) оба пункта и добавьте модули. Помимо этого, в разделе Configuration Properties | C\C++ | Precompiled Header (Свойства конфигурации | C\C++ | Предварительно скомпилированные заголовочные файлы) для обеих конфигураций установите в параметре Create/Use precompiled Header (Создавать/Использовать предварительно скомпилированные заголовочные файлы) параметр Automatically Genera-te (/YX) (Автоматически генерировать).

# 3.Скелет приложения

Даже для простого приложения Win32 мастер создает слишком много лишнего. Например, меню и диалоговое окно с информацией о программе. Все это ненужно, если игра будет работать только в полноэкранном режиме. В большинстве игр меню и диалоговые окна Windows не используются, поэтому и мы не будем этого делать. Удаляем содержимое файла, который сгенерировал мастер, и заменяем его кодом из листинга 1.1.



# 4.Моделирование

В играх сложно обойтись без объектов сеток. Создавать объекты игры вручную из треугольников достаточно утомительная задача. Намного проще смоделировать объекты с помощью 3D-редактора и потом загрузить их в своей программе. Хороший 3D-редактор с помощью визуальных средств позволяет легко и быстро создавать достаточно сложные модели. Я предпочитаю использовать 3D Studio Max (http://www.discreet.com). Да, это дорогой продукт, но универсальный и позволяет создать одну и ту же фигуру разными способами.

Создав сетку персонажа или другого объекта игры, скелет и анимацию, все это необходимо сохранить для последующей загрузки из собственной программы. Вот тут возникает проблема выбора — как сохранить? Формат MAX позволяет хранить всю необходимую информацию, но он слишком сложный для использования в собственных программах. Можно использовать старый формат 3DS, но тут возникает проблема с сохранением информации о костях и самой скелетной анимации. Эти проблемы можно обойти, но проблематично. Придется писать собственные анализаторы и загрузки данных, а потом еще создавать классы или функции для поддержки всего загруженного в программе. Зачем выдумывать себе проблемы, когда можно использовать X-формат файла, который разработан Microsoft и является открытым, универсальным и легко расширяемым. В предыдущем разделе мы увидели, как легко можно загрузить и отобразить сетку на экране. В дальнейшем вы убедитесь, что X-формат файла удобен и для создания анимации, такой как скелетная.

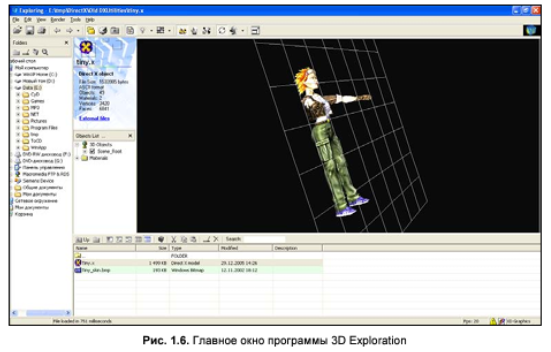
Итак, мы можем преобразовать созданную модель в старый формат 3DS и потом конвертировать ее в X-файл. Для этого в Интернете можно найти множество различных утилит, от простых и бесплатных до сложных и дорогих. Для простых сцен я использую утилиту conv3ds.exe (она идет в поставке с DX SDK). Для конвертирования достаточно выполнить команду: conv3ds.exe filename.3ds

И в результате мы получим в том же каталоге файл filename.x. Но иногда эта простая утилита конвертирует не корректно, поэтому приходится прибегать к более сложной утилите 3D Exploration (http://www.xdsoft.com/explorer/), которая поддерживает большинство распространенных графических форматов. Главное окно программы можно увидеть на рис. 1.6.

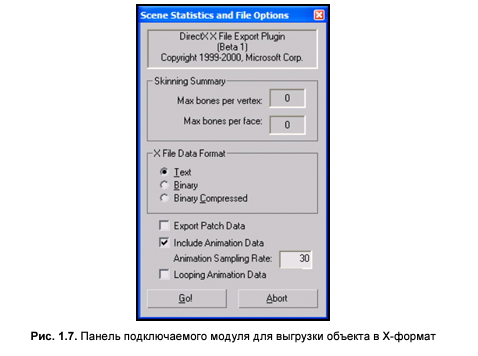
Сразу же бросается в глаза, что панель управления у программы скудная. Да, возможностей не так уж и много. Основанные функции программы это:

* просмотр 3D-файлов. Программа поддерживает все распространенные форматы, в том числе и игровые, такие как файлы моделей для игр Quake и Quake 2. Поддержка игровых форматов очень полезна для нас, как для разработчиков 3D-игр и аниматоров;
* конвертирование из одного формата в другой. Для решения этой задачи у программы есть множество настроек. Количество форматов файлов, в которые вы можете конвертировать 3D-сцену, намного меньше, но основные форматы есть.

Функции по редактированию сцены очень скудные, да и не это главное. Если я не ошибаюсь, то корни у этой программы русские. Если корни действительно русские, то стоит гордиться людьми, которые могут создавать такие хорошие проекты и чего-то добиваться своим трудом.



Расширять возможности файла нам не понадобится, потому что все необходимое для хранения сетки и костей в нем уже есть. Сама 3ds Max не может сохранять или экспортировать данные в Х-формат, но можно установить специальный подключаемый модуль (plug-in), разработанный самой Microsoft. Этот plug-in можно найти в составе DX SDK, причем в исходных кодах. Нам остается только скомпилировать его и установить. На рис. 1.7 можно увидеть, как выглядит этот подключаемый модуль в программе 3DS. Хватит рекламировать 3D Studio Max, к тому же Autodesk (владелец торговой марки) не платит мне за это. Это только в кино, когда показывают какой-то продукт, то это скорей всего является скрытой рекламой. А мы этим заниматься не будем, поэтому выбирайте то, что удобно и по карману именно вам. Я только высказал свои предпочтения и показал, как можно конвертировать проекты 3D Studio Max в Х-формат. Итак, с подготовительными "телодвижениями" покончено, давайте переходить к более интересным вещам, а именно — к работе с графикой.



# 5. Популярные игры, созданные на C++

World of Warcraft

World of Warcraft (WoW) — это одна из самых известных MMORPG (Massively Multiplayer Online Role-Playing Game), разработанная компанией Blizzard Entertainment. WoW использует C++ для создания высокопроизводительных игровых систем и сложных алгоритмов, которые обеспечивают плавный игровой процесс для миллионов игроков по всему миру. Игра включает в себя огромный открытый мир, множество квестов и событий, а также сложную систему взаимодействия между игроками. Это делает WoW не только технически сложным проектом, но и примером того, как C++ может быть использован для создания масштабных и детализированных игровых миров.

Counter-Strike — это культовый шутер от первого лица, который был изначально создан как модификация для Half-Life. Игра написана на C++ и использует движок GoldSrc, что позволяет ей быть одной из самых популярных и долговечных игр в своем жанре. Counter-Strike известен своим быстрым и напряженным игровым процессом, а также высокой степенью реализма. Благодаря использованию C++, разработчики смогли создать игру, которая не только выглядит и ощущается реалистично, но и работает плавно на различных системах.

The Elder Scrolls V: Skyrim

Skyrim — это ролевая игра с открытым миром, разработанная Bethesda Game Studios. Игра написана на C++ и использует движок Creation Engine, что позволяет создавать огромные, детализированные миры с богатым игровым процессом. В Skyrim игроки могут исследовать обширные территории, выполнять разнообразные квесты и взаимодействовать с множеством персонажей. Использование C++ позволяет игре обрабатывать сложные алгоритмы и физические модели, что делает игровой процесс более реалистичным и захватывающим.

Doom

Doom — это легендарный шутер от первого лица, разработанный id Software. Игра написана на C++ и использует движок id Tech, который стал основой для многих других игр в жанре. Doom известен своими быстрыми и динамичными боями, а также атмосферным дизайном уровней. Благодаря использованию C++, разработчики смогли создать игру, которая не только выглядит впечатляюще, но и работает плавно даже на менее мощных системах.

Fortnite

Fortnite — это популярная игра в жанре королевской битвы, разработанная Epic Games. Игра написана на C++ и использует движок Unreal Engine, который также создан на C++. Это позволяет Fortnite обеспечивать высокую производительность и качественную графику. В Fortnite игроки могут строить сооружения, сражаться с другими игроками и участвовать в различных игровых режимах. Использование C++ позволяет игре обрабатывать большое количество данных и обеспечивать плавный игровой процесс даже при большом количестве игроков на одном сервере.

# Заключение

Разработка игр на C++ — это увлекательное и сложное занятие, которое требует глубоких знаний и навыков. Начните с изучения основ языка и простых проектов, постепенно переходя к более сложным задачам. Используйте популярные библиотеки и движки, такие как Unreal Engine и SFML, чтобы упростить процесс разработки и сосредоточиться на создании качественного игрового контента.

Не забывайте также изучать примеры кода и участвовать в сообществах разработчиков, чтобы получать советы и поддержку от более опытных коллег. Участие в форумах, чтение блогов и просмотр обучающих видео помогут вам быстрее освоить все тонкости разработки игр на C++. Также полезно изучать исходный код популярных игр и анализировать, как они реализованы. Это поможет вам лучше понять, как использовать возможности C++ для создания собственных проектов.

Кроме того, не забывайте о важности тестирования и оптимизации вашего кода. Игры должны работать плавно и без ошибок, поэтому уделяйте внимание тестированию на различных устройствах и платформах. Оптимизация кода поможет вам добиться высокой производительности и улучшить игровой процесс.

В заключение, разработка игр на C++ — это сложный, но увлекательный процесс, который требует постоянного обучения и практики.

Используемые источники:

1. Популярные игры созднанные на языке C++, <URL:https://sky.pro/wiki/profession/populyarnye-igry-na-c-chto-mozhno-sozdat/>
2. Михаил Фленов, «Искусство программирования игр на C++», Санкт-Петербург «БХВ-Петербург» 2006
3. C ++ для разработки игр: полное руководство, URL:https://translated.turbopages.org/proxy\_u/en-ru.ru.14670d8b-6756b121-3bac4584-74722d776562/https/www.geeksforgeeks.org/cpp-for-game-development/