Областное бюджетное профессиональное

образовательное учреждение

«Курский техникум связи»

# РАЗРАБОТКА ИГРЫ «FAIRY TAIL» В ЖАНРЕ ПЛАТФОРМЕР НА UNITY

Дипломная работа

по направлению подготовки 09.02.03 Программирование в компьютерных системах

Курск 2024

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc167636659)

[1 Исследование понятия «компьютерная игра» и обзор технологий разработки компьютерных игр 6](#_Toc167636660)

[1.1 Исследование предметной области 6](#_Toc167636661)

[1.2 Формирование требований к программному изделию 10](#_Toc167636662)

ВВЕДЕНИЕ

Компьютерные игры стали неотъемлемой частью современной культуры и развлечений. Все больше людей обращаются к играм как к способу расслабиться, отвлечься от повседневных забот и погрузиться в увлекательные миры виртуальной реальности. Они становятся не просто развлечением, но и инструментом, который может использоваться для обучения, тренировки навыков и даже развития креативности и воображения.

В России, как и во многих странах, игровая индустрия развивается не так быстро, как хотелось бы. Несмотря на значительный интерес к играм со стороны пользователей, количество компаний-разработчиков в нашей стране остается недостаточным для полноценной конкуренции с западными производителями. Это создает определенные вызовы и возможности для развития индустрии в России и стимулирует исследования в этой области.

Цель дипломной работы заключается в разработке игры «Fairy Tail» в жанре 2D-платформер на платформе Unity. Проект направлен на создание увлекательного и захватывающего игрового опыта для пользователей, где каждый игрок сможет погрузиться в мир волшебства, приключений и фэнтези. Разработка игры «Fairy Tail» представляет собой уникальную возможность для изучения процесса создания компьютерных игр, от выбора жанра и платформы до реализации анимации, и создания прототипа.

Исследование особенностей и состояния компьютерной индустрии в России поможет понять текущие тенденции и проблемы, с которыми сталкиваются разработчики игр в нашей стране. Анализ жанров, видов и платформ для игр даст возможность выбрать оптимальные решения для проекта «Fairy Tail» и создать интересный игровой опыт для пользователей. Разработка сценария и концепции игры позволит определить основные элементы, которые сделают игру увлекательной и уникальной.

Таким образом, разработка игры «Fairy Tail» не только представляет собой возможность создания захватывающего продукта для пользователей, но и способствует развитию игровой индустрии в России, стимулирует творческий подход к разработке игр и открывает новые перспективы для будущих проектов в этой области.

Объектом исследования является разработка компьютерных игр.

Предмет исследования: технологии разработки компьютерной игры жанра 2D-платформер.

Цель дипломной работы заключается в разработке прототипа компьютерной игры в жанре 2D-платформер с использованием средств Unity.

Для достижения поставленной цели необходимо выполнить следующие задачи:

– изучить особенности и состояние компьютерной индустрии России;

– выбрать жанр, вид и платформу для компьютерной игры;

– разработать сценарий, концепцию основных элементов;

– выбрать и изучить средство реализации;

– подготовить необходимые для игры анимации;

– реализация прототипа игры.

ГЛАВА 1 Предпроектное обследование

1.1 Исследование предметной области

Понятие компьютерной игры (иногда также называемой видеоигрой) обозначает программу, которая служит для организации игрового процесса (геймплея), связи с партнёром по игре, или сама выступающая в качестве партнёра. Часто компьютерные игры создаются на основе сторонних источников, таких как фильмы или книги, но в последнее время наблюдается обратное явление, когда на основе известной игровой серии выпускаются дополнительные материалы, расширяющие мир игры.

Существуют специально разработанные игры, которые могут использоваться в образовательных целях или для научных исследований. Хотя такие игры редко попадают в широкое распространение. Некоторые игры становятся предметом любительских и профессиональных соревнований, известных как киберспорт.

Влияние компьютерных игр на общество настолько значительно, что в последнее время в области информационных технологий наблюдается тенденция к геймификации неигровых приложений. Например, в некоторых европейских школах игра «Minecraft» используется для обучения. С 2011 года компьютерные игры официально признаны правительством США и Национальным фондом США отдельным видом искусства, наряду с театром и кино. В России киберспорт был официально признан видом спорта.

Все это свидетельствует о том, что компьютерные игры тесно взаимодействуют с нашей современной жизнью. Их область применения значительно расширилась за последние десять лет: теперь игры используются не только для развлечения, но и для обучения, проведения научных исследований и других целей.

Компьютерные игры классифицируют по нескольким основным признакам:

– жанр;

– количество игроков;

– визуальное представление;

– платформа.

Жанр игры определяется целью и основной механикой игры. Основные жанры представлены в таблице 1.

Таблица 1 – Жанровая классификация

|  |  |  |
| --- | --- | --- |
| Жанр | Особенности | Основные поджанры |
| Аркада (Arcade) | Игры с нарочно примитивным игровым процессом | Платформер  Файтинг  Скролеры |
| Приключенческая игра (Adventure) | Главной частью игры является история. | Квест  Визуальная новелла |
| Ролевая игра (RPG) | Главная особенность игры, игрок практически не ограничен в выборе игровых предметов, напарников и диалогов. | Тактическая  Экшен RPG  Японская (Западная) |
| Экшен (Action) | Игра, характеризующая частым и активным нажатием кнопок управления. | Beat’em Up  Шутер  Слешер |
| Стратегическая игра (Strategy) | Основа игры – необходимость игроку делать нетривиальный выбор. | Пошаговая/Реального времени  Глобальная  Варгейм  Экономические |
| Компьютерный симулятор (Simulator) | Игрок делает множество упражнений и оттачивает свою технику. | Технические  Симуляторы жизни |
| Головоломка (Puzzle) | Требует аналитического мышления. |  |
| Обучающая игра  (Educational) | Игрок обучается во время выполнения какихлибо действий в игре. |  |
| Игрушки (Toys) | Программы, взаимодействуя с которыми, игрок получает удовольствие. |  |

Существует много игр, которые не подпадают под четкую классификацию, так как в последнее время появляются игры с уникальными жанрами, которые можно отнести как к одному из существующих, так и к собственному, выделяющемуся.

По количеству игроков игры делятся на две категории:

– однопользовательские;

– многопользовательские;

С точки зрения визуализации компьютерные игры могут быть разделены на следующие типы:

– текстовые – минимальное графическое представление, общение с игроком проходит с помощью текста;

– 2D, где все элементы отрисованы в виде двумерной графики (спрайтов);

– 3D, где все элементы отрисованы в виде трехмерной графики (3D – модели).

В данном перечне отсутствуют категории 2,5D и псевдо-3D, так как они являются вариациями представленных категорий, и технологии их создания практически не отличаются от основных.

По типу платформы:

– персональные компьютеры;

– игровые приставки/консоли;

– мобильные телефоны;

К сожалению, данная классификация не является исчерпывающей и может быть дополнена. Например, можно добавить музыкальные игры, где акцент делается на музыке и звуках, а многопользовательские игры могут быть разделены на различные подкатегории. Важно понимать, что данная классификация достаточна для определения большинства существующих игр, но не является исчерпывающей из-за разнообразия игровых форматов. Игры могут сочетать несколько жанров, выпускаться на разных платформах и предлагать, как одиночный, так и многопользовательский режимы игры.

Возникновение игровой индустрии произошло относительно недавно, по сравнению с другими развлекательными сферами, и является причиной того, что она продолжает эволюционировать и расширяться. Несмотря на это, представленная классификация подходит для определения большинства существующих компьютерных игр на сегодняшний день. Исходя из данной классификации, было принято решение разработать двухмерный однопользовательский платформер для персональных компьютеров. Платформер был выбран из-за своей простой и интуитивно понятной механики, что делает его идеальным выбором для начинающих разработчиков игр. Кроме того, создание двухмерных игр проще по сравнению с трехмерными, что ускоряет и делает проще процесс производства.

1.2 Формирование требований к программному изделию

Процесс разработки компьютерной игры включает в себя ряд этапов, которые схожи с процессом разработки других программных продуктов. Основные этапы разработки игры включают в себя следующие этапы:

– планирование;

– разработка;

– тестирование и выпуск.

На этапе планирования определяются цели игры и выбираются инструменты для ее создания.

При определении целей игры выделяются основная концепция, жанр и атмосфера игры. Концепция игры – это то, что вдохновляет игроков на участие в игре, и она тесно связана с выбранным жанром. Например, основной концепцией RPG может быть возможность игрока создать и развивать своего персонажа, в то время как для шутеров – участие в стрельбах и боевых действиях. Определение основной концепции игры помогает выбрать подходящий жанр.

После выбора жанра и концепции игры следующим шагом будет определение атмосферы игры. Атмосфера – это окружение, в котором разворачивается игровой процесс. Она включает в себя место, время и условия игры. Выбор атмосферы может значительно облегчить создание сценария для игры, поэтому важно выбрать атмосферу заранее, учитывая предпочтения целевой аудитории.

При разработке компьютерных игр ключевыми средствами являются программный код и игровой движок. От правильного выбора этих компонентов зависит как скорость разработки, так и качество конечного продукта. Выбор программного кода зависит от платформы, на которой будет запускаться игра. Например, для игр в браузере часто используются языки программирования Java или Flash, в то время как для компьютерных игр наиболее подходящим может быть язык C#.

Игровой движок отвечает за низкоуровневое описание физики объектов, визуализацию графики и другие аспекты. При выборе игрового движка важно учитывать его доступность и совместимость с выбранным языком программирования. Например, Unity позволяет создавать игры на C# и предоставляет бесплатный доступ при определенных условиях.

После выбора целей игры и инструментов разработки начинается следующий этап – разработка. Этот этап является наиболее объемным и трудоемким, включая множество шагов, необходимых для создания качественного и работоспособного продукта.

На первом этапе разработки компьютерной игры необходимо определить сценарий и игровую механику. Игровая механика опирается на цель игры и устанавливает правила и объекты, с которыми игрок будет взаимодействовать.

Параллельно с разработкой игровой механики проводится работа над сюжетом игры. Сюжет играет важную роль, поскольку определяет уровень интереса игрока к проекту. Сюжет представлен двумя вариантами: литературным и режиссерским сценариями. Литературный сценарий описывает основные события и персонажей игры, в то время как режиссерский сценарий подробно раскрывает уровни игры и события, происходящие на них.

Также на данном этапе начинается разработка графического дизайна игры. На основе сюжета и заранее обсужденного дизайна создаются концептуальные иллюстрации, которые послужат основой для разработки общего вида игры и персонажей.

После завершения разработки сюжета и игровой механики начинается ключевой этап – активное создание игры.

На основе сюжета и концептуальных иллюстраций начинается создание персонажей и игровых объектов, параллельно с этим происходит разработка игровых уровней. При проектировании игровых уровней сначала создается их упрощенный план, на котором схематически изображены элементы уровня и предметы, с которыми игрок будет взаимодействовать.

В начале создается первая версия игрового уровня, которая обычно представляет собой минимально оборудованную локацию с необходимыми элементами для прохождения. Эта версия служит для проверки проходимости уровня.

Вскоре после создания первых уровней разрабатывается альфа-версия игры, первый прототип, который позволяет проверить основные игровые механики и их соответствие требованиям. В альфа-версии игры объекты могут быть представлены без текстур или в виде абстрактных форм.

При успешном прохождении альфа-тестирования начинается проработка механики и объектов игры. На этом этапе уровни и игровые механики усовершенствуются, а в игру добавляются первые элементы сюжета, такие как видеоролики, диалоги и кат-сцены. Также исправляются найденные ошибки и недочеты в коде игры.

После этого приступают к созданию второго прототипа игры, известного как бета-версия. Бета-версия позволяет провести тестирование игры на ошибки и недоработки, и фактически представляет собой почти готовую игру. В бета-версии могут отсутствовать незначительные элементы, не влияющие на геймплей. При тестировании бета-версии проверяется весь функционал игры. Часто для тестирования бета-версии приглашают обычных игроков, особенно если игра мультиплеерная, что помогает ускорить процесс тестирования и снизить нагрузку на команду разработчиков.

Если игра проходит бета-тестирование, она отправляется на окончательную доработку и исправление критических ошибок, после чего идет сборка финальной версии игры и следом наступает релиз игры.

После выпуска игры следует ее последующая поддержка. Поддержка включает в себя выпуск патчей, то есть исправлений ошибок в уже выпущенном продукте. Кроме того, для продления жизненного цикла игры выпускают дополнительный контент в виде DLC (Downloadable Content), который добавляет новые предметы и возможности для игроков.

Дополнительный контент может включать в себя новые уровни, персонажей, сюжетные линии, предметы экипировки и другие игровые элементы, которые обогащают игровой опыт и продлевают интерес к игре. DLC позволяет разработчикам расширить возможности игры и удовлетворить запросы игроков на новый контент.

Таким образом, становится очевидным, что этапы разработки компьютерных игр во многом схожи с этапами разработки других программных продуктов. Поддержка и выпуск дополнительного контента после релиза игры являются важными этапами для обеспечения долгосрочного успеха и удовлетворения аудитории игры.

ГЛАВА 2 Техническое задание

2.1 Постановка задачи

Целью данного дипломного проекта является разработка игры «Fairy Tail» в жанре платформер на платформе Unity. Игра будет представлять собой приключенческую игру, в которой игроку предстоит управлять персонажем и преодолевать различные препятствия, собирать предметы и бороться с врагами.

Для достижения поставленной цели необходимо решить ряд задач. В первую очередь, требуется разработать дизайн уровней, включая локации, препятствия, платформы и секретные места, чтобы создать увлекательное и разнообразное игровое пространство. Далее, необходимо создать анимации для персонажей, включая анимации бега и прыжков, чтобы придать игре динамичность и реалистичность.

Важным шагом является реализация логики игрового процесса, которая включает в себя управление персонажем, взаимодействие с объектами на уровне, бои с врагами и сбор предметов. Это позволит создать интересный и увлекательный геймплей, который будет привлекать игроков. Кроме того, необходимо разработать систему сохранения прогресса игры и управление звуками и музыкой, чтобы обеспечить комфортное игровое восприятие.

Важным этапом является тестирование игры на наличие ошибок и багов, а также оптимизация ее производительности. Только после успешного завершения всех этапов разработки и тестирования игра будет готова к выходу.

Для реализации поставленных задач рекомендуется использовать стандартные программные средства, такие как Unity для разработки игры, C# для написания скриптов и программирования игровой логики, графические редакторы для создания спрайтов и текстур, программы для создания анимаций, а также Unity Asset Store для поиска и использования готовых ассетов.

Таким образом, постановка задачи для разработки игры «Fairy Tail» в жанре платформер на Unity предполагает определение целей и задач проекта, логику их реализации, а также рекомендации по использованию стандартных программных средств для достижения поставленной цели.

2.2 Концепция информационной базы и её структура

Концепция информационной базы в программировании относится к организации и хранению данных, необходимых для работы программного продукта. Информационная база представляет собой структурированный набор данных, который используется программой для выполнения своих функций. Она включает в себя различные типы информации, такие как текст, числа, изображения, аудиофайлы и другие.

Структура информационной базы включает в себя определение типов данных, их атрибутов и связей между ними. Например, в базе данных структура может быть представлена таблицами с определенными полями, связанными ключами или отношениями. В других случаях, информационная база может быть представлена в виде дерева каталогов или файловой системы.

Важно правильно спроектировать структуру информационной базы в программировании, чтобы обеспечить эффективное хранение, доступ и обработку данных программой. Хорошо спроектированная информационная база позволяет улучшить производительность программы, обеспечить надежность данных и удобство работы с ними.

При разработке игры ключевым этапом является определение информационной базы проекта. Это важный шаг, который позволяет структурировать данные, определить объекты сущностей и связи между ними.

В контексте игры «Fairy Tail» можно выделить несколько основных объектов сущностей, которые играют важную роль в игровом процессе. Первым и, пожалуй, одним из наиболее значимых объектов является персонаж. Это игровой герой, которым управляет игрок, а также враги, с которыми персонаж взаимодействует. Другим важным объектом являются уровни игры, которые включают в себя локации, препятствия, предметы и врагов. Предметы, анимации, звуки и музыка также играют существенную роль в создании атмосферы игры и взаимодействии с игроком.

Для каждой сущности определяются атрибуты, которые характеризуют ее свойства и параметры. Например, у персонажей могут быть атрибуты, такие как имя, уровень здоровья, сила атаки и так далее. Для уровней игры атрибутами могут быть сложность, количество врагов, расположение предметов и другие параметры, влияющие на игровой процесс.

После выделения атрибутов происходит их агрегация в рамках каждой сущности. Например, атрибуты персонажа, такие как имя, уровень здоровья и сила атаки, объединяются в одну сущность «Персонаж». Это позволяет более компактно и системно представить информацию о каждой сущности.

Для обеспечения минимальной достаточности атрибутов и устранения избыточности информации происходит нормализация отношений между сущностями. Это позволяет уменьшить дублирование данных и обеспечить их целостность. На основе выделенных сущностей, атрибутов и отношений строится инфологическая модель задачи, отображающая структуру информационной базы проекта и оптимизирующая процессы взаимодействия между сущностями.

Таким образом, концепция информационной базы игры «Fairy Tail» в жанре платформер на Unity предполагает определение объектов сущностей, выделение атрибутов, их агрегацию, нормализацию отношений и построение инфологической модели для эффективной организации информационных процессов в игре.

2.3 Неформальное описание алгоритма работы ПП

Неформальное описание алгоритма работы программного продукта (ПП) представляет собой словесное описание глобальных шагов функционирования программы без привязки к конкретным техническим деталям или языкам программирования. Этот этап проектирования программы направлен на общее понимание логики работы продукта, выделение ключевых этапов выполнения задачи и определение основных подсистем, которые будут взаимодействовать для достижения целей программы.

В неформальном описании алгоритма работы ПП обычно описывается общий принцип работы программы, последовательность действий, взаимодействие различных компонентов, основные функциональные возможности и ожидаемые результаты. Это помогает разработчикам, заказчикам и другим участникам проекта лучше понять суть программного продукта, определить его основные характеристики и обеспечить успешное выполнение поставленных задач.

Неформальное описание алгоритма работы ПП является важным этапом проектирования перед тем, как приступать к более детальной разработке программы, так как позволяет четко определить цели и задачи продукта, структуру его функционирования и основные принципы взаимодействия компонентов.

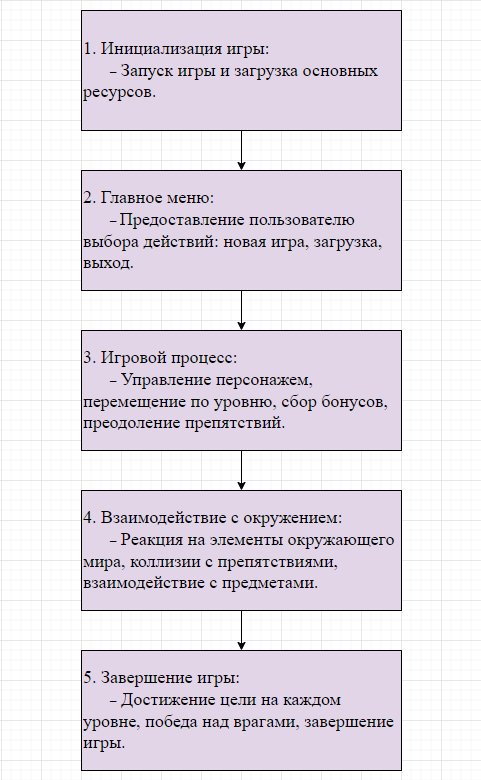
Алгоритм работы игры «Fairy Tail» в жанре платформера начинается с загрузки главного меню, где игроку предоставляется возможность начать новую игру. После выбора новой игры происходит загрузка уровня, который представляет собой платформы, препятствия, врагов и другие игровые элементы.

Игровой процесс включает управление персонажем, который может перемещаться по уровню, выполнять прыжки, атаковать врагов и взаимодействовать с окружающей средой. Физика окружения определяет поведение объектов на экране, их взаимодействие друг с другом и с персонажем. Система коллизий отвечает за обнаружение столкновений между объектами и определение реакции на них.

Подсистемы игры «Fairy Tail» включают управление, искусственный интеллект врагов, анимации персонажей, звуковое сопровождение, систему сохранения прогресса и другие. Каждая подсистема имеет свою роль в общем функционировании игры и требует дальнейшей детализации для полноценной реализации.

Для наглядного представления общей структуры алгоритма работы программного продукта рекомендуется создать укрупненную блок-схему, описанную на рисунке 1, которая позволит визуализировать взаимосвязи между основными шагами и подсистемами игры. Это поможет разработчикам и заинтересованным сторонам лучше понять логику работы игры «Fairy Tail» и обеспечит успешную реализацию проекта. Тут уже можно добавить каких-то скриншотов, о том о чем говоришь!

Рисунок 1 – Блок-схема структуры алгоритма работы программного продукта Такой большой рисунок добавлять нельзя, его нужно выносить в приложения! Либо делать как-то меньше, либо дробить на этапы. Сделать в виде USE-case например



2.4 Функции подсистем (модулей)

При разработке игры «Fairy Tail» были выделены различные подсистемы, каждая из которых отвечает за определённую функциональность игры. Ниже приведены основные функции каждой подсистемы:

Геймплейная подсистема:

– отслеживание передвижения игрового персонажа;

– обработка действий игрового персонажа (прыжки, атаки);

– реализация взаимодействия игрового персонажа с окружающим миром (взаимодействие с объектами, столкновения).

Анимационная подсистема:

– управление анимацией игрового персонажа (бег, прыжки);

– переключение между анимацией в зависимости от действий игрока;

– создание плавных переходов между анимацией.

Графическая подсистема:

– загрузка и отображение игровых ресурсов (спрайты, текстуры);

– управление интерфейсом игры (отображение жизней, очков);

– реализация эффектов визуализации (спецэффекты, анимации);

Звуковая подсистема:

– воспроизведение фоновой музыки и звуковых эффектов.

– переключение между различными аудио дорожками в зависимости от игровых событий.

Каждая подсистема выполняет свою чётко определённую функцию и вместе обеспечивают работу игры. Связь между подсистемами осуществляется с помощью вызова функций и обмена данными. В результате каждая подсистема работает как независимый модуль, который взаимодействует с другими модулями для обеспечения полноценного функционирования игры.

2.5 Оценка экономической эффективности от внедрения программы

Эти расчеты необходимо взять в методичке или в примере что я скидывал.

Для проведения оценки затрат на разработку игры на Unity, необходимо учитывать следующие критерии: оценка размера программного изделия, оценка трудоемкости, оценка продолжительности, оценка стоимости.

Оценка размера программного изделия:

Игра «Fairy Tail» в жанре платформер была разработана одним разработчиком. При разработке размер программного изделия по объему кода определяется примерно в 1500 строк кода, что соответствует небольшому проекту.

Оценка трудоемкости:

Учитывая, что игра была разработана одним разработчиком, трудоемкость можно оценить как среднюю. Основное время на разработку игры ушло на создание уровней, персонажей, логики игрового процесса и тестирование.

Оценка продолжительности:

Продолжительность разработки «Fairy Tail» в жанре платформер на Unity составила приблизительно 6 месяцев.

Оценка стоимости:

Все ресурсы для разработки игры были бесплатными, включая Unity для создания игры, ассеты, звуковые эффекты и так далее. Не использовались услуги дизайнера, программиста или других специалистов, все работы были выполнены самостоятельно. С учетом этого, можно сделать вывод о том, что затраты на разработку проекта ограничивались только временем и усилиями одного человека.

Таким образом, с учетом оценки размера программного изделия, трудоемкости, продолжительности и стоимости, затраты на разработку игры «Fairy Tail» в жанре платформер на Unity могут быть оценены как минимальные и ограничиваются временем и усилиями одного разработчика.