

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего
образования

«Алтайский государственный технический университет
им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Специальность (направление, профиль) Программная инженерия

Курсовой проект

защищен с оценкой _____

_____ В.С. Троицкий

(подпись руководителя проекта) (инициалы, фамилия)

“ ____ ” _____ 2024 г.

КУРСОВОЙ ПРОЕКТ

Разработка компьютерной игры-квест в жанре платформер
(тема курсового проекта)

Пояснительная записка

по дисциплине Программирование – 3 семестр

КП 09.03.04.1.000 ПЗ

(обозначение документа)

Студенты группы ПИ-22 Гартман С. А. _____

Карелин Н. В. _____

Чепрасов А. В. _____

(фамилия, имя, отчество)

(подпись)

21.01.2024

(дата)

Руководитель

Проекта доцент, к.т.н. _____ В.С. Троицкий

(должность, ученое звание)

(подпись)

(инициалы, фамилия)

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Алтайский государственный технический университет
имени И.И. Ползунова»

Факультет информационных технологий

Кафедра «Прикладная математика»

З А Д А Н И Е

на курсовой проект по дисциплине «Программирование»

студентам группы ПИ-22 Чепрасову Андрею Вячеславовичу, Гартману Степану Андреевичу, Карелину Никите Витальевичу

Тема курсового проекта: «Разработка компьютерной игры-квест в жанре платформер».

Календарный план работы:

№ этапа	Содержание этапа	Недели семестра
1	Получение задания	1
2	Постановка задачи. Работа с документацией	2
3	Проектирование программы	3-4
4	Реализация программы	5-13
5	Оформление пояснительной записки	14
6	Защита курсового проекта	15-16

Руководитель проекта _____ Троицкий В. С., доцент
подпись

Дата выдачи задания «04» сентября 2023 г.
число месяц год

Задание приняли к исполнению _____ Гартман С. А.
_____ Карелин Н. В.
_____ Чепрасов А. В.
подпись

Содержание

Введение	1
1. Обзор предметной области и постановка задачи.....	2
1.1. Обзор предметной области.....	2
1.2. Постановка задачи.....	2
2. Проектирование	4
2.1. Алгоритм решения	4
2.2 Структура данных	6
3. Реализация	7
3.1. Выбор средств реализации	7
3.2. Структура программы.....	8
3.3. Состав программы.....	9
Заключение	9
Список использованных источников	13
Приложение А Исходный код программы	14
Приложение Б Результат работы программы	15

Введение

В настоящее время разработка видеоигр стала одной из наиболее популярных областей программирования. Создание гейм-движков, написание игр и управление её ресурсами - всё это требует глубокого понимания своего дела и не обходится без применения объектно-ориентированного программирования (ООП)

Цель данного курсового проекта - разработать 2D пиксельную приключенческую видеоигру с использованием принципов ООП. Этот проект предоставит возможность применить умения и знания ООП на практике, создавая игровые объекты, персонажей, сцены и игровую логику.

В данном отчете будут рассмотрены основные этапы разработки игры, включая проектирование классов, представленных в используемом движке блюпринтами (от слова blueprint), их взаимодействие и реализацию игровых механик.

Использование ООП в разработке видеоигр позволяет упростить поддержку кода, повысить его читаемость и расширяемость, что делает данную тему актуальной и интересной для изучения и практической реализации.

1. Обзор предметной области и постановка задачи

1.1. Обзор предметной области

Игровой движок — базовое программное обеспечение компьютерной игры. Разделение игры и игрового движка часто расплывчато, и не всегда студии проводят чёткую границу между ними. Но в общем случае термин «игровой движок» применяется для того программного обеспечения, которое пригодно для повторного использования и расширения, и тем самым может быть рассмотрено как основание для разработки множества различных игр без существенных изменений. Он предоставляет все инструменты и материалы, необходимые для проработки игрового мира и его последующего функционирования, поэтому серьезно экономит время и деньги.

Игровой движок в играх отвечает практически за всё. Благодаря работе игрового движка на экране не только двигаются объекты, но и происходит расчет освещения, влияние действий на игровой мир, обработка команд, поступающих с контроллера игрока, переключение музыки и множество других действий.

Движок игры работает на настолько глубоком уровне, что именно он обеспечивает появление на экране необходимых моделей персонажей, задает им анимации и поведение, а также включает озвучку, музыку и звуковые эффекты, соответствующие происходящим событиям. Часто игровые движки используют наравне с обычными языками программирования ещё и визуальные.

Блочное кодирование - это язык программирования, который позволяет пользователям создавать программы, манипулируя программными элементами графически, а не указывая их как текст.

Блюпринт— система визуального программирования в Unreal Engine 4 и Unreal Engine 5 на основе нодов с данными: событиями и функциями. Их можно связывать между собой и формировать элементы геймплея. Различают несколько видов блюпринтов, рассчитанных на определённые задачи — от создания события на уровне до интерфейсов и макросов, которые можно использовать как основу для другого блюпринта.

Блюпринты не полная альтернатива программированию на C++. Всё зависит от самого проекта и возможностей разработчика. Кто-то делает игры целиком на C++, другие используют только блюпринты. Иногда применяется комплексный подход: прототип собирают на блюпринтах, а непосредственно сам проект пишут на C++.

В дискуссиях программисты часто упоминают, что проекты на C++ получаются более оптимизированными и в них проще делать отладки. К тому же рефакторинг блюпринтов реализован в несколько раз хуже, что критично для масштабных проектов.

1.2. Постановка задачи

Разработать игру в жанре квест с видом сбоку и пиксельной графикой. В игре представлена одна многоуровневая локация – общежитие с интерактивным внутренним миром.

Игра должна включать в себя:

1) Игрового персонажа, которым будет управлять игрок и проходить основной квест, также у персонажа должны быть сделаны анимации, чтобы картинка не выглядела статичной.

2) Игровая локация, которую будет исследовать игрок: в нашем случае – общежитие и его постоянные жильцы. Общежитие не должно быть пустым, иначе игроку будет неинтересно исследовать локацию.

3) Понятный интерфейс, чтобы игрок мог легко ориентироваться в игровом меню и побочных меню, которые будут в игре.

4) Полноценный квест, иначе игроку будет нечего делать на локации и пропадёт смысл исследовать её.

5) Защита от неожиданных сценариев, иначе игрок не поймёт изначальной задумки и пропустит весь сюжет.

Цель состоит в том, чтобы показать человеку, который не жил в общежитии, ситуации, с которыми может столкнуться первокурсник, который только заселился в общежитие. Наша игра погружает в дружественную атмосферу студенческого общежития, показывает важность помощи и общения в жизни человека.

2. Проектирование

2.1. Алгоритм решения

Программа работает в оконном и полноэкранном режимах, является полноценным приложением.

При входе в программу пользователь видит главное меню в оконном режиме, где он осуществляет выбор для дальнейшей работы:

- Начать играть;
- Полноэкранный режим;
- Выход.

При выборе меню «Начать играть», осуществляется переход непосредственно к самой игре.

При выборе меню «Полноэкранный режим» программа меняет режим, переходит из оконного в полноэкранный режим.

При выборе меню «Выход», вызывается меню, в котором у пользователя спрашивается, хочет ли он покинуть игру и предоставляется выбор:

- Да;
- Нет.

Соответственно, при выборе ответа «Да», программа завершает свою работу, при выборе ответа «Нет», программа возвращает пользователя в главное меню.

Как сказано было выше, при выборе в главном меню «Начать играть», осуществляется переход непосредственно к самой игре. Игра представляет собой многоуровневую локацию “Общежитие” с визуальным интерфейсом и открытым для взаимодействия миром. Пользователь является основным героем игры, которому необходимо взаимодействовать со второстепенными персонажами для прохождения игры. Целью игры является знакомство с жильцами общежития (второстепенными персонажами) посредством главного квеста, задачей которого является сбор подписей. Взаимодействие происходит путем общения главного и второстепенных персонажей.

При отказе выполнения квеста, пользователь досрочно заканчивает прохождение и перед ним появляется меню с выбором:

- Начать заново;
- Выйти в меню.

При выборе меню «Начать заново», осуществляется перезапуск игры, и пользователь начинает прохождение сначала. Соответственно, при выборе «Выйти в меню», пользователь попадает в главное меню.

При успешном диалоге персонажей, пользователю открывается доступ к главному квесту по сбору подписей. Во время прохождения квеста, пользователь может рассмотреть обстановку общежития, познакомиться с его жильцами и соответственно выполнит главный квест.

Перемещение между этажами локации осуществляется через меню, которое вызывается при подходе к определенной двери в конце этажа с помощью клавиши Enter. В данном меню пользователь может сделать выбор, куда ему отправиться “вверх” или “вниз”, в зависимости от того, на каком этаже находится пользователь.

Пользователь может в любой момент игрового процесса вызывать меню паузы нажатием клавиши Tab, в котором предоставляется выбор:

- Вернуться в игру;
- Выйти в меню.

При выборе меню «Вернуться в игру», пользователь возвращается в игру. Соответственно, при выборе «Выйти в меню», пользователь попадает в главное меню.

Как говорилось выше, основной целью игры является выполнение главного квеста. При успешном выполнении данного квеста, пользователь заканчивает прохождение игры. Перед пользователем открывается победное меню с поздравлением и секретной наградой, как и в других меню, пользователю доступен выбор:

- Начать заново;
- Выйти в меню.

При выборе меню «Начать заново», осуществляется перезапуск игры, и пользователь начинает прохождение сначала. Соответственно, при выборе «Выйти в меню», пользователь попадает в главное меню.

При выборе определённой функции программа ждёт выполнения условий от пользователя и даёт соответствующую реакцию.

Выполненная работа:

- Гартман Степан – построение локации, написание меню, настройка анимаций, настройка персонажей;
- Карелин Никита – диалоговая система, квест и внутреннее оформление локации, оформление виджетов;
- Чепрасов Андрей – физика перемещения, концовки, пауза, настройка виджетов и действий виджетов.

2.2 Структура данных

Для решения поставленной задачи реализованы следующие модули:

- MainTileMap – Карта, построенная из плиток;
- MainHero – Главный персонаж, которым управляет игрок;
- Menu – Карта меню запуска игры;
- MainMap – Игровая карта, основная часть игры;
- 2DSideScrollerGameMode – Игровой режим, который хранит информацию о игроке
- BT_CharacterResponse – Класс, отвечающий за ответы игрока;
- BT_EndDialog – Класс, отвечающий за выход из диалога
- BT_NPCSpeak – Класс, отвечающий за речь второстепенных героев
- BT_ResponseDecorator – Класс, отвечающий за проверку ответа героя
- PauseMenuWB – Виджет, отвечающий за меню паузы
- Quest_Item – Игровой предмет, который игрок должен поднимать
- Parent – Родительский класс для неиграбельных героев типа Child
- Child_Adam – Персонаж Командир ССБ
- Child_Alex – Персонаж Мотя
- Child_Amelia – Персонаж Психолог
- Child_Bob – Персонаж Охранник
- DialogController – Класс, отвечающий за поведение персонажа при диалоге
- DialogBehaviorTree – Класс, отвечающий за реплики персонажа Охранник
- DialogBehaviorTreeAdam – Класс, отвечающий за реплики персонажа Командир ССБ
- DialogBehaviorTreeAlex – Класс, отвечающий за реплики персонажа Мотя
- DialogBehaviorTreeAmelia – Класс, отвечающий за реплики персонажа Психолог
- DialogComponent_Adam – Класс, отвечающий за вызов диалога с персонажем Командир ССБ
- DialogComponent_Alex – Класс, отвечающий за вызов диалога с персонажем Мотя
- DialogComponent_Amelia – Класс, отвечающий за вызов диалога с персонажем Психолог
- DialogComponent_Bob – Класс, отвечающий за вызов диалога с персонажем Охранник

3. Реализация

3.1. Выбор средств реализации

Данное ПО реализовано на языке Blueprint с использованием игрового движка Unreal Engine 4.27.2. Blueprint — система визуального программирования в Unreal Engine 4 и Unreal Engine 5 на основе нодов с данными: событиями и функциями.

Для полноценного функционирования программы требуется:

- ОС: Windows 10 версии 1909 или новее, Windows 11 версии 21H2 или новее;
- Оперативная память: не менее 8 ГБ ОЗУ;
- Жёсткий диск: не менее 20 ГБ;
- Процессор: минимальное число ядер – 4, 2,5 ГГц или выше;
- Видеоадаптер: DirectX 11 или DirectX 12 совместимые графические карты;
- Наличие клавиатуры, мыши, монитора.

3.2. Структура программы

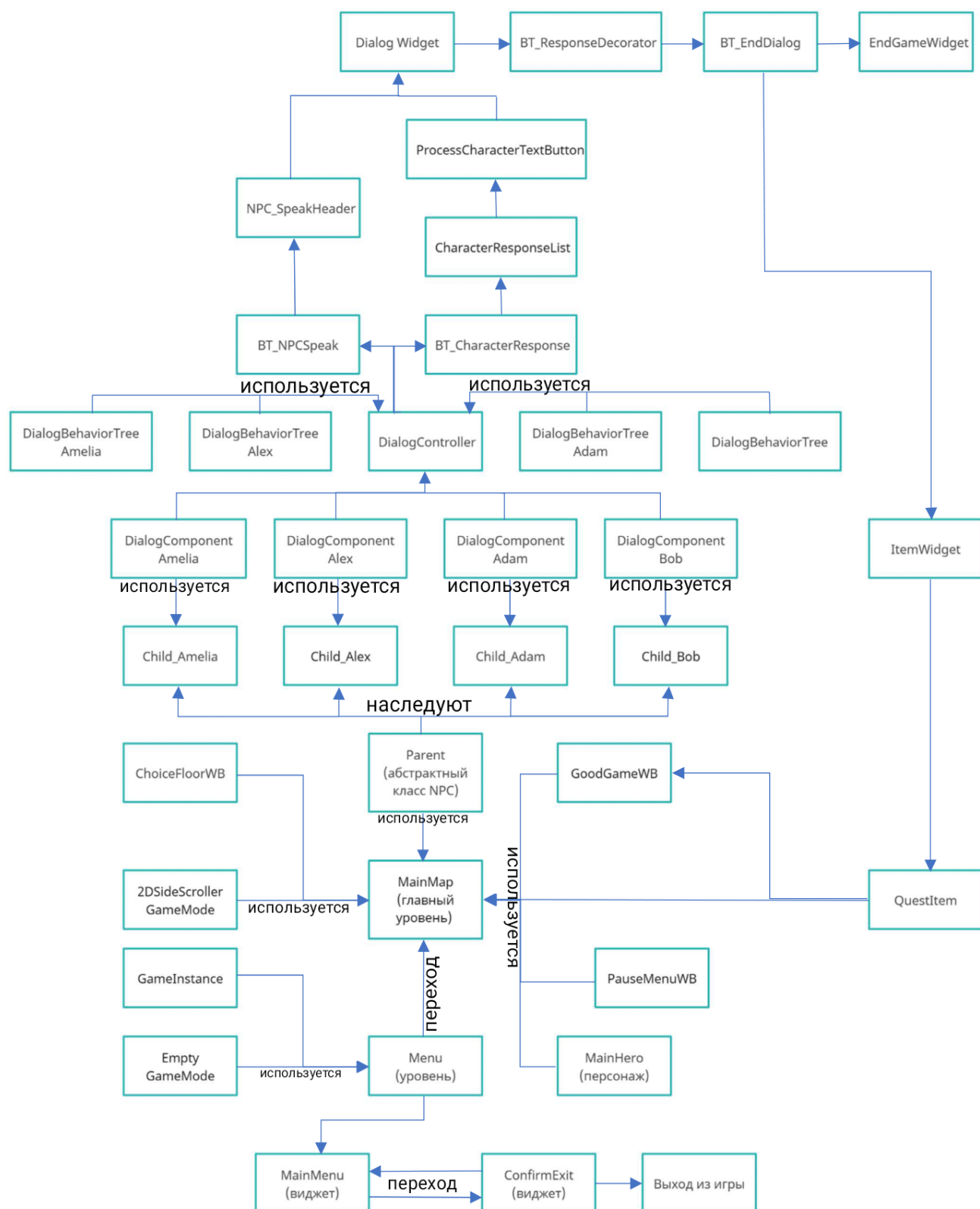


Рисунок 1 – Структура программы в виде иерархии модулей

3.3. Состав программы

Модули (функции) программы:

- MainTileMap – Карта, построенная из плиток. Создана с помощью TileSet, в которых находятся плитки, каждая плитка содержит в себе текстуру;
- MainHero – Главный персонаж, которым управляет игрок. Содержит в себе функции: передвижение героя, обновление анимаций, анимация передвижения, прыжок, взятие квеста, работа дверей, вызов меню выбора этажа, вызов меню паузы;
- MainMap – Игровая карта, основная часть игры. При запуске вызывает запуск музыки;
- 2DSideScrollerGameMode – Игровой режим, который хранит информацию о игроке – основное управление и героя, которым управляет игрок и который является главным.
- Quest_Item – Игровой предмет, который игрок должен поднимать, содержит в себе функцию счёта выполненных миссий и вызов победного меню при сборе всех предметов.
- Parent – Родительский класс для неиграбельных героев типа Child, содержит в себе текстуру, куда будет помещаться спрайт персонажа.
- Child_Adam – Персонаж Командир ССБ, содержит в себе коробку, входя в которую появляется надпись, содержащая имя персонажа.
- Child_Alex – Персонаж Мотя, содержит в себе коробку, входя в которую появляется надпись, содержащая имя персонажа.
- Child_Amelia – Персонаж Психолог, содержит в себе коробку, входя в которую появляется надпись, содержащая имя персонажа.
- Child_Bob – Персонаж Охранник, содержит в себе коробку, входя в которую появляется надпись, содержащая имя персонажа.
- PauseMenuWB – Виджет, отвечающий за меню паузы, содержит в себе две кнопки, первая отвечает за возобновление игры, вторая – за выход в главное меню.
- ItemWidget – Виджет, который показывает количество подобранных Quest_Item.
- ChoiceFloorWB – Виджет, который вызывается при подходе к двери. Имеет в себе две кнопки, работает только на 2 и 3 этаже. Выбор кнопки переносит на этаж ниже или выше соответственно.
- EndGameWB – Виджет, который запускается при выборе неверного диалога и провале миссии. Содержит в себе 2 кнопки – запуск уровня сначала и выход в

главное меню. При выборе первого варианта запускается MainMap, при выборе второго – Menu.

- GoodGameWB - Виджет, который запускается при сборе всех Quest_Item и победе в игре. Содержит в себе 2 кнопки – запуск уровня сначала и выход в главное меню. При выборе первого варианта запускается MainMap, при выборе второго – Menu. Также содержит в себе секретный подарок.
- DialogWidget – Виджет, который вызывается при диалоге со второстепенными персонажами. Он содержит в себе панель с текстовый блок с фразами второстепенного персонажа (Text block - NPC_SpeakHeader), также присутствует блок с вариантами ответов пользователем (CharacterResponseList), каждый вариант ответа является кнопкой (CharacterTextButton), в зависимости от выбора которой происходит дальнейший диалог. Все фразы и ответы наследуются от соответствующего дерева поведения – BehaviorTree, которое выбирается при взаимодействии со второстепенным персонажем игры.
- BT_CharacterResponse – Класс, отвечающий за ответы игрока. После взаимодействия с определенным второстепенным персонажем вызывается дерево поведения – BehaviorTree. Из дерева поведения достаются реплики-ответы для главного персонажа и отправляются в CharacterResponseList.
- BT_EndDialog – Класс, отвечающий за выход из диалога. Данный класс является флагом-проверки негативных реплик-ответов главного героя, при выборе которых завершается диалог и вызывается виджет EndGameWB.
- BT_NPCSpeak – Класс, отвечающий за речь второстепенных героев. После взаимодействия с определенным второстепенным персонажем вызывается дерево поведения – BehaviorTree. Из дерева поведения достаются реплики для второстепенного персонажа и отправляются в NPC_SpeakHeader.
- BT_ResponseDecorator – Класс, отвечающий за проверку ответа героя. Данный класс является флагом проверки реплик-ответов главного героя, для дальнейшего диалога.
- DialogController - Класс, отвечающий за поведение персонажа при диалоге.
- DialogComponent_Bob – Класс, вызываемый персонажем Child_Bob, отвечающий за вызов диалога с персонажем Охранник. Вызов диалога осуществляется путем привязки данного модуля программы к Child_Bob. Данный класс вызывает диалог с персонажем Охранник, после нажатия пользователем клавиши Е. Событие обрабатывается через универсальный DialogController, с последующем вызовом диалога через DialogBehaviorTree. Диалог вызывается через виджет DialogWidget.

- DialogBehaviorTree - Класс, отвечающий за реплики персонажа Охранник. Данный класс отправляет заготовленные реплики персонажа Охранник и варианты ответа пользователя в DialogWidget.
- DialogComponent_Amelia – Класс, вызываемый персонажем Child_Amelia, отвечающий за вызов диалога с персонажем Психолог. Вызов диалога осуществляется путем привязки данного модуля программы к Child_Amelia. Данный класс вызывает диалог с персонажем Психолог, после нажатия пользователем клавиши Е. Событие обрабатывается через универсальный DialogController, с последующим вызовом диалога через DialogBehaviorTreeAmelia. Диалог вызывается через виджет DialogWidget.
- DialogBehaviorTreeAmelia- Класс, отвечающий за реплики персонажа Психолог. Данный класс отправляет заготовленные реплики персонажа Психолог и варианты ответа пользователя в DialogWidget.
- DialogComponent_Adam – Класс, вызываемый персонажем Child_Adam, отвечающий за вызов диалога с персонажем Мотя. Вызов диалога осуществляется путем привязки данного модуля программы к Child_Adam. Данный класс вызывает диалог с персонажем Мотя, после нажатия пользователем клавиши Е. Событие обрабатывается через универсальный DialogController, с последующим вызовом диалога через DialogBehaviorTreeAdam. Диалог вызывается через виджет DialogWidget.
- DialogBehaviorTreeAdam- Класс, отвечающий за реплики персонажа Мотя. Данный класс отправляет заготовленные реплики персонажа Мотя и варианты ответа пользователя в DialogWidget.
- DialogComponent_Alex – Класс, вызываемый персонажем Child_Alex, отвечающий за вызов диалога с персонажем Командир ССБ. Вызов диалога осуществляется путем привязки данного модуля программы к Child_Alex. Данный класс вызывает диалог с персонажем Командир ССБ, после нажатия пользователем клавиши Е. Событие обрабатывается через универсальный DialogController, с последующим вызовом диалога через DialogBehaviorTreeAlex. Диалог вызывается через виджет DialogWidget.
- DialogBehaviorTreeAlex- Класс, отвечающий за реплики персонажа Мотя. Данный класс отправляет заготовленные реплики персонажа Мотя и варианты ответа пользователя в DialogWidget.

Заключение

В ходе работы была создан программный продукт, который погружает первокурсника в атмосферу общежития после изучения документации. Реализованы защиты на излом игры, построена локация и персонажи, настроены диалоги сделано вертикальное перемещение по локации.

В процессе выполнения задания был подробно изучены движок Unreal Engine 4 и визуальный язык программирования Blueprint.

Преимущества приложения:

1. Мощная платформа. Unreal Engine 4 на данный момент является одним из самых продвинутых движков, предлагающих широкий спектр инструментов и функций. Он позволяет реализовать практически любую идею и возможность, что делает разработку игры более продуктивной.
2. Обширная поддержка сообщества. Unreal Engine не заканчивается только на движке. За ним стоит активное сообщество, которое может предложить использовать уже готовые решения, информационные ресурсы и форумы, а компания Ubisoft этому только способствует.

Недостатки приложения:

- Зависимость проектирования визуального интерфейса игры от готовых спрайт-изображений для объектов игры;
- Отсутствие выбора настроек качества рендера;
- Ограниченность одной игровой локацией;
- Отсутствие настроек и подсказок управления.

Возможны следующие усовершенствования приложения со временем:

- Создание и проработка новых локаций с большими активностями;
- Возможность сохранения результатов и загрузки предыдущих результатов;
- Возможность настройки качества рендера;
- Добавление настройки управления;

Список использованных источников

1. Habr : сообщество IT-специалистов / Тьюториал по Unreal Engine. - Кипр, 2006 - . - URL: <https://habr.com/ru/articles/344446/> (дата обращения: 19.01.2024).
2. Unreal Engine 4 Documentation. - Соединённые Штаты Америки, 2020 - . - URL: <https://docs.unrealengine.com/4.27/en-US/> (дата обращения: 20.01.2024).
3. Leaf Corcoran, Itch corp. - Соединённые Штаты Америки, 2013.- Top Game assets -.- URL: <https://itch.io/game-assets> (дата обращения: 20.01.2024).
4. Youtube : бесплатный видеохостинг / Канал MakeYourGame!. - Соединённые Штаты Америки, 2006 - . - URL: <https://www.youtube.com/@makeyourgame2210/> (дата обращения: 19.01.2024).

Приложение А

Исходный код программы

Весь код программы находится на платформе GitHub по ссылке:

Unreal Engine 4: <https://github.com/DemboDev/DormCourseWork>

Готовая сборка:

<https://drive.google.com/file/d/1zgqTFDSGEISrZBPIKEnnJ3sZtxz2bEEb/view?usp=sharing>

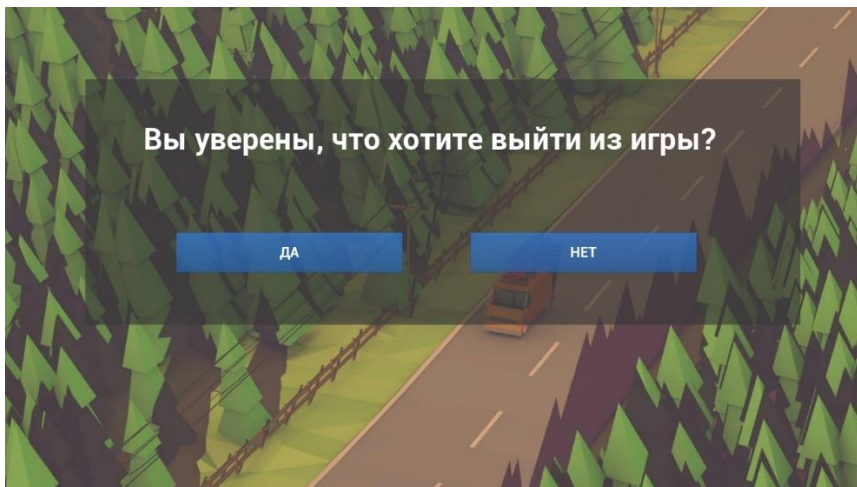
Приложение Б

Результат работы программы

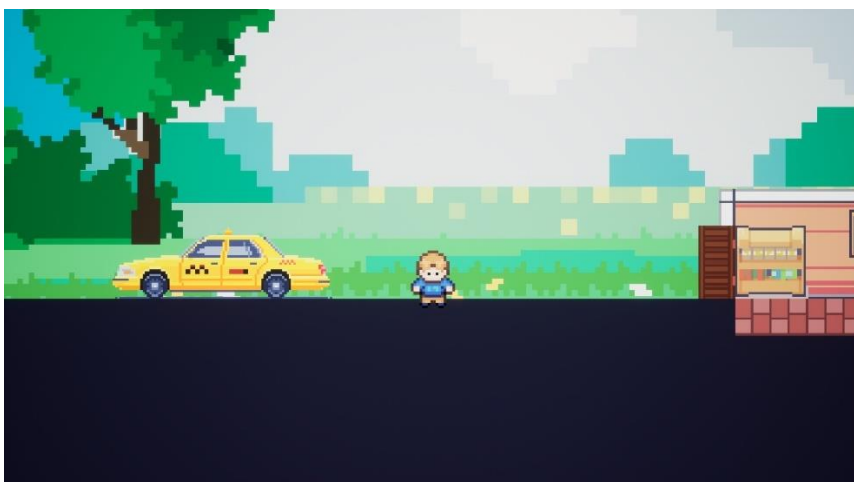
- Главное меню



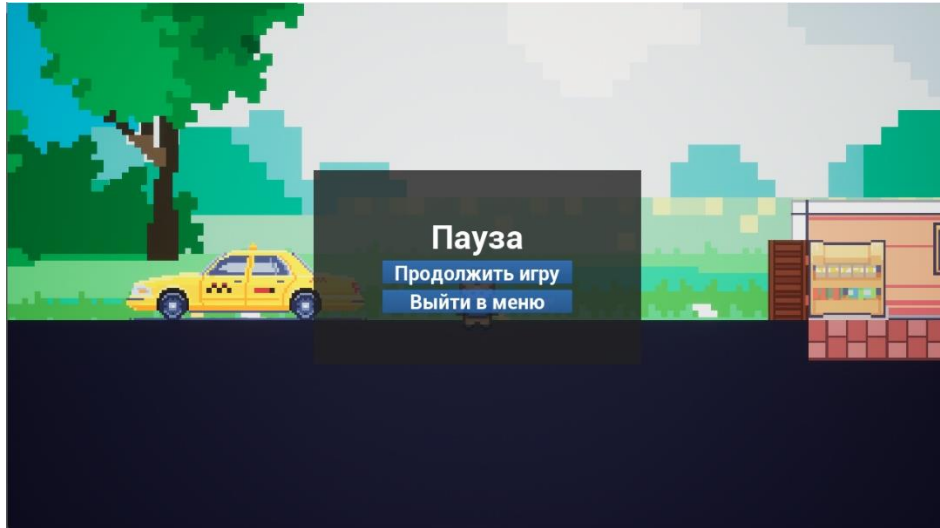
- Меню выхода из программы



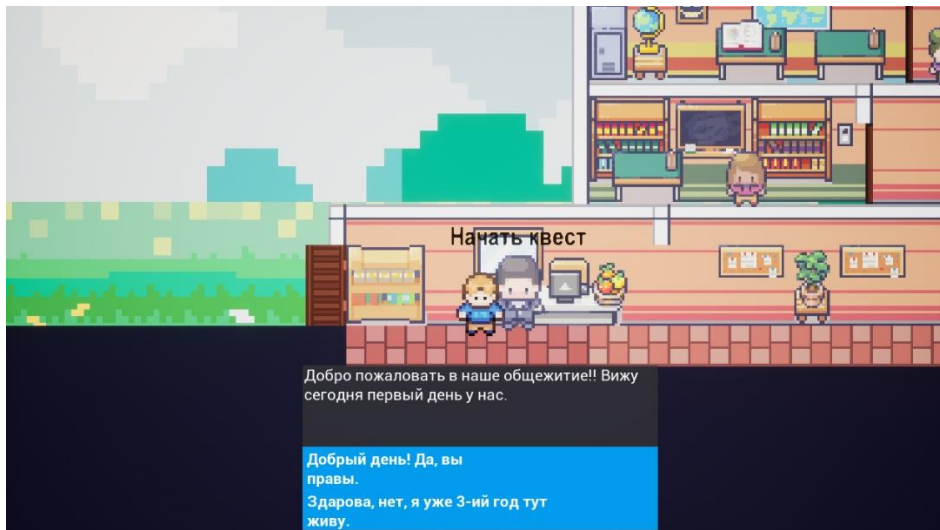
- Стартовая позиция игрока, после нажатия “Play”



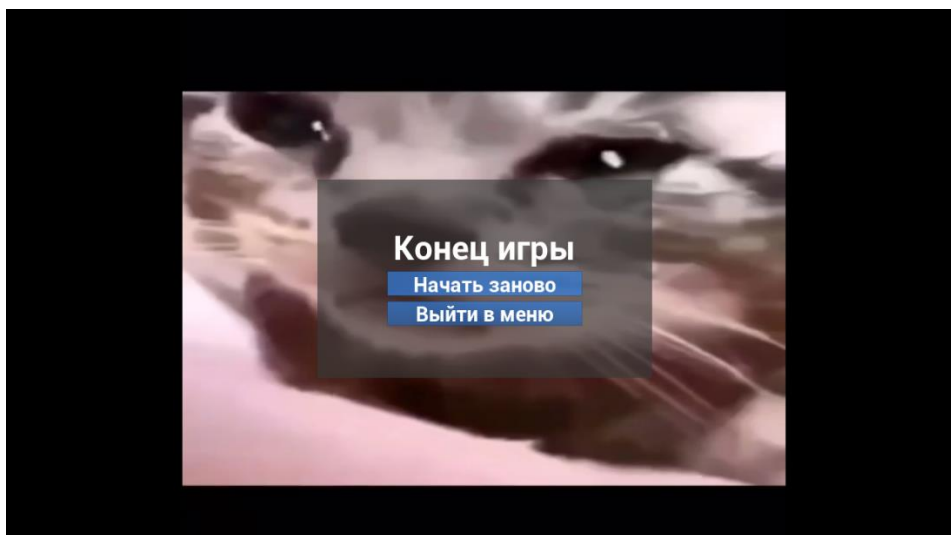
- Меню паузы



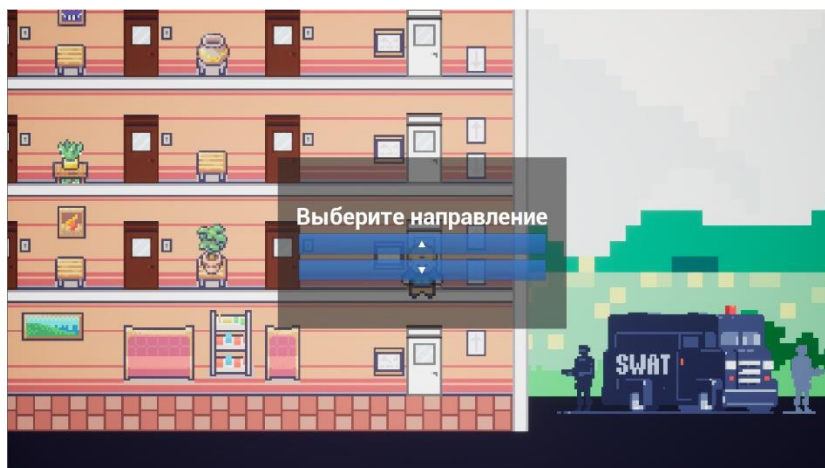
- Начало квеста. Диалоговая система



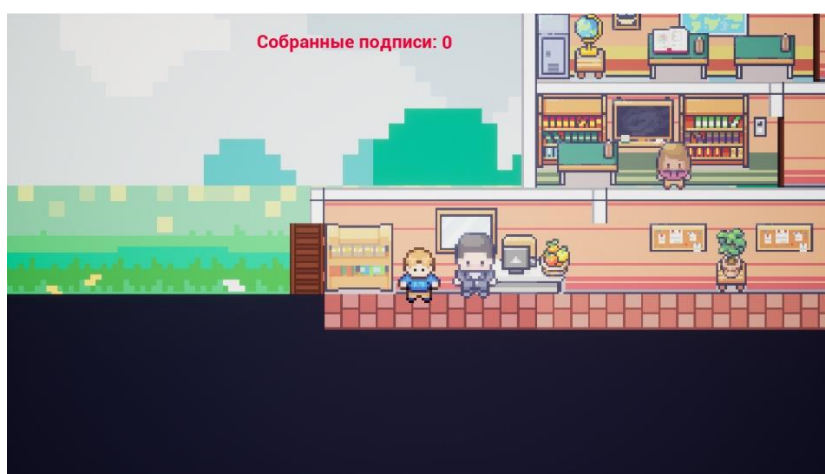
- Отклонение квеста. Меню проигрыша



- Меню выбора этажа



- Взятие квеста. Отображение выполнения



- Выполнение квеста





- Завершение квеста. Меню победы

