

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Направление подготовки
09.03.01 Информатика и вычислительная техника

Направленность (профиль)
«Технологии разработки программного обеспечения»

Выпускная квалификационная работа

Использование инструментария Godot Engine для решения задач геймдева

Обучающегося 4 курса
очной формы обучения
Степаненко Артёма Кирилловича

Руководитель выпускной
квалификационной работы:
доктор педагогических наук, профессор
кафедры информационных технологий и
электронного обучения
Власова Елена Зотиковна

Санкт-Петербург
2024

Содержание

Введение.....	3
Глава 1. Теория игровой разработки.....	6
1.1. История развития индустрии видеоигр.....	6
1.2. Структура компьютерных игр.....	10
1.3. Основные принципы работы игровых движков.....	14
1.4. Анализ функционала Godot Engine.....	18
Выводы по главе 1.....	20
Глава 2. Создание игрового проекта.....	21
2.1. Разработка идеи для компьютерной игры.....	21
Выводы по разделу 2.1.....	23
2.2. Создание визуальных элементов.....	24
Выводы по разделу 2.2.....	29
2.3. Реализация первого уровня видеоигры.....	30
Выводы по разделу 2.3.....	36
2.4. Реализация второго уровня видеоигры.....	38
Выводы по разделу 2.4.....	38
2.5. Разработка главного меню.....	39
Выводы по разделу 2.5.....	39
2.6. Внедрение звуковых эффектов.....	40
Выводы по разделу 2.6.....	41
2.7. Экспорт проекта.....	42
Выводы по разделу 2.7.....	42
Заключение.....	43
Библиография.....	44

Введение

Компьютерные игры – это с каждым годом набирающий большую популярность вид развлечений, взаимодействие с которым происходит при помощи соответствующих электронных устройств. По данным Newzoo, на сегодняшний момент в видеоигры играют около 3 миллиардов человек.

Данная выпускная квалификационная работа представляет собой комплексный проект, охватывающий различные направления разработки программного обеспечения, включает в себя все основные аспекты создания компьютерных игр и является продуктом индустрии компьютерных развлечений.

Актуальность выпускной квалификационной работы заключается в потребности большого количества IT-специалистов, занимающихся созданием компьютерных игр, поскольку постепенно разработка каждой видеоигры становится масштабным процессом.

Предметом исследования является Godot Engine – один из новых игровых движков, который имеет большие перспективы развития благодаря своей доступности.

Теоретическая значимость заключается в возможности начинающим разработчикам легко стать частью игропрома благодаря изучению данного инструментария.

Практическая значимость заключается в перспективе развития данного направления в Российской Федерации с целью создания конкуренции на мировом рынке IT-технологий и внедрения отечественных идей в игровую индустрию.

Целью выпускной квалификационной работы стала разработка демонстрационной версии компьютерной игры с помощью инструментария Godot Engine.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Детально проработать сюжет и персонажей компьютерной игры.
2. Продумать идею, как игра должна проходиться и на какие уровни (главы, миссии) она будет разделена.

3. Разработать дизайн персонажей и окружения, звуковые эффекты.
4. Реализовать идеи игры с помощью игрового движка.

Результатом бакалаврской выпускной квалификационной работы является готовая демонстрационная версия компьютерной игры, разработанная при помощи инструментария Godot Engine.

Структура бакалаврской выпускной квалификационной работы.

Выпускная квалификационная работа состоит из введения, двух глав, заключения и списка литературы.

Introduction

Computer games are a type of entertainment that is becoming increasingly popular every year and is interacted with using appropriate electronic devices.

This final qualifying work is a comprehensive project covering various areas of software development, includes all the main aspects of creating computer games and is a product of the computer entertainment industry.

The relevance of the final qualifying work lies in the need for a large number of IT specialists involved in the creation of computer games, since gradually the development of each video game becomes a large-scale process.

The subject of the study is the Godot Engine, one of the new game engines that has more development prospects due to its accessibility.

The theoretical significance lies in the ability for novice developers to easily become part of the gaming industry by studying this toolkit.

The practical significance lies in the prospects for the development of this area in the Russian Federation with the aim of creating competition in the global IT technology market and introducing domestic ideas into the gaming industry.

The goal of the final qualifying work is the development of the demo version of the computer game using the Godot Engine tools.

To achieve this goal, it is necessary to solve the following **tasks**:

1. Work out in detail the plot and characters of the computer game.
2. Think over an idea of how the game should be played and into what levels (chapters, missions) it will be divided.
3. Develop character and environment design, sound effects.
4. Implement game ideas using a game engine.

The result of the final qualifying work is the demo version of the computer game developed using the Godot Engine tools.

Structure of final qualifying work. The final qualifying work consists of an introduction, two chapters, a conclusion and a list of references.

Глава 1. Теория игровой разработки

1.1. История развития индустрии видеоигр

Компьютерные игры за несколько десятилетий смогли стать неотъемлемой частью досуга для огромного количества людей. Данный вид развлечений активно начал зарождаться в середине XX века, в период после Второй мировой войны.

Первопричинами появления геймдева принято считать два знаменательных события, произошедших в тот промежуток времени. 14 февраля 1946 года был произведён первый в мире программируемый компьютер – ENIAC [12, стр. 146]. Незадолго после этого, в 1948 году, британскими математиками Дэвидом Чемперноуном и Аланом Тьюрингом был разработан алгоритм программы Turochamp, который мог позволить искусственному интеллекту победить реального человека в шахматах. Однако технологиям тех времён не удавалось воспроизвести эту сложную инструкцию.

Несмотря на неудачу эксперимента, данное исследование мотивировало других учёных продолжать развитие в направлении разработки цифровых игр. Одним из первых успешных примеров в этой области является Spacewar! – космический симулятор, разработанный студентами Массачусетского технологического института в 1961 году. Правила игры заключаются в следующем: два участника берут на себя роль космических кораблей и цель каждого – уничтожить своего оппонента, уворачиваясь от вражеских снарядов вокруг звезды в центре экрана (Рисунок 1.1).

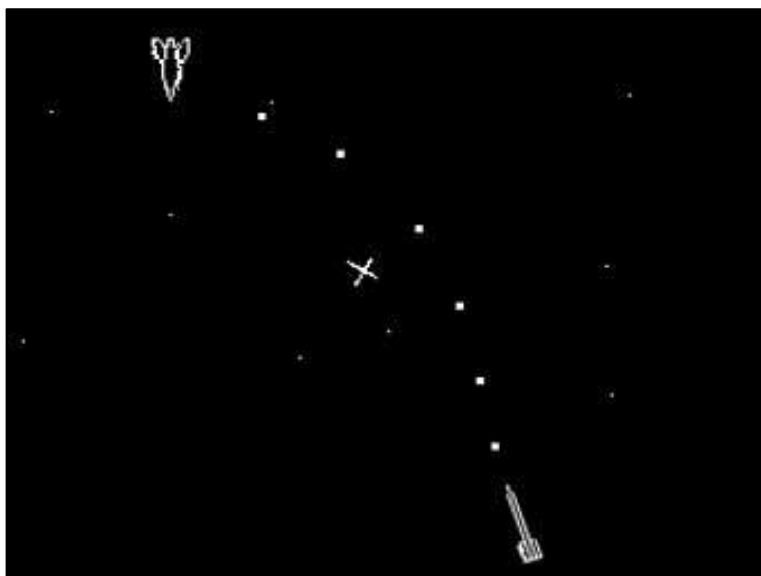


Рисунок 1.1. Изображение видеоигры Spacewar!

С тех пор видеоигры начали совершенствоваться с максимальной эффективностью. Появились особые электронные девайсы, специализирующиеся на работе интерактивных развлечений – игровые приставки [19, стр. 248]. Основоположителем подобного типа оборудования является коммерческая видеоигровая система для домашнего использования Magnavox Odyssey (Рисунок 1.2). Позже технологии данных систем стали адаптироваться для персональных компьютеров.



Рисунок 1.2. Первая в мире игровая консоль Magnavox Odyssey и запущенная на ней виртуальная игра в пинг-понг

Благодаря стремительному прогрессу геймдевелопмент получил широкое распространение, аппаратное обеспечение стало более универсальным, а производство игр заметно упростилось. Появилось понятие игровых движков: если в ранние годы каждая видеоигра представляла собой уникальную программную сущность, то с появлением в 1990-е годы нового ключевого инструмента разработчики имели возможность создавать игры-клоны – проекты, обладающие идентичной основой [5, стр. 1888]. Одним из ранних таких примеров является шутер от первого лица «Heretic» (Рисунок 1.3), созданный с помощью инструментария id Tech 1 и обладающий наработками компьютерной игры «Doom» (Рисунок 1.4).



Рисунок 1.3. Изображение видеоигры «Heretic»



Рисунок 1.4. Изображение видеоигры «Doom»

1.2. Структура компьютерных игр

Основное отличие проектов, связанных с разработкой видеоигр, от других в области информационных технологий заключается в том, что результат должен передать пользователю уникальный опыт [8, стр. 92].

Для успешного достижения данной цели одной работы программистов окажется недостаточно. В создании игры могут принимать участие представители широкого спектра специальностей:

- художники;
- аниматоры;
- писатели;
- математики;
- композиторы;
- звуковые дизайнеры;
- маркетологи.

Список может включать в себя даже профессии, которые на первый взгляд не имеют ничего общего с описанной деятельностью. Например, умения архитекторов позволят детально спроектировать сооружения виртуальных миров; знания антропологии и психологии дадут понять, какие элементы создаваемой компьютерной игры могут удовлетворить геймеров; навыки актёрского мастерства смогут лучше погрузить игроков в сюжет проекта и заставить их проникнуться переживаниями главных героев [15, стр. 19-22].

Однако самым главным связующим звеном игровой разработки является геймдизайнер. Это специалист, обладающий основными качествами упомянутых ранее профессий. При этом он имеет относительно других членов команды наиболее важные задачи: объединить усилия участников разработки, достичь гармонии между исполнителями, издателем и целевой аудиторией, и создать игру, которая позволит людям получить определённый опыт с помощью различных составляющих данного продукта.

Цифровые развлечения состоят из четырёх важных компонентов:

1) Игровая механика

Отличительной чертой компьютерных игр относительно других средств для получения опыта (например, фильмов или книг) является интерактивность, которая достигается благодаря данной составляющей. Когда пользователь с помощью определённых манипуляторов даёт конкретные команды действующим лицам, расположенных на экране монитора, телевизора или даже мобильного устройства, это способствует максимальному погружению в события виртуального мира и позволяет геймеру почувствовать себя их активным участником.

2) История

Многие люди ошибочно полагают, что данный элемент является самым необязательным для проектов данного типа. Однако именно благодаря наличию истории игрок может активно вовлечься в игровой процесс [15, стр. 81]. Приведём пример: стратегическая игра «This is The Police» от белорусской компании Weappy Studio повествует о немолодом шефе полиции Джеке Бойде, которого через полгода собираются отправить в отставку. Цель главного героя – заработать в течение этого периода полмиллиона долларов любыми способами. Подобный синопсис мотивирует пользователя успешно выполнить поставленную задачу и пройти игру полностью, чтобы узнать, чем закончится сюжет и для чего протагонисту нужна была данная сумма денег.

3) Художественный стиль

Вселенная видеоигры не ограничивается лишь одной историей. Чтобы игроку поверить в происходящее на экране, необходима визуальная составляющая, соответствующая повествованию. В упомянутой ранее игре «This is The Police» основную часть дисплея занимает макет карты криминального города Фрибург, который насыщен тёмно-серыми деталями – такое цветовое решение позволяет грамотно передать атмосферу мрачного сюжета. Кроме того, нижнюю часть интерфейса занимает работающий в конкретную смену отряд

полиции: все представители имеют карточку со своим обликом (Рисунок 1.5). Это придаёт каждому члену вашей команды индивидуальности, и заставляет пользователя действовать более осторожно при отправке исполнителей на опасные вызовы.



Рисунок 1.5. Изображение видеоигры «This is The Police»

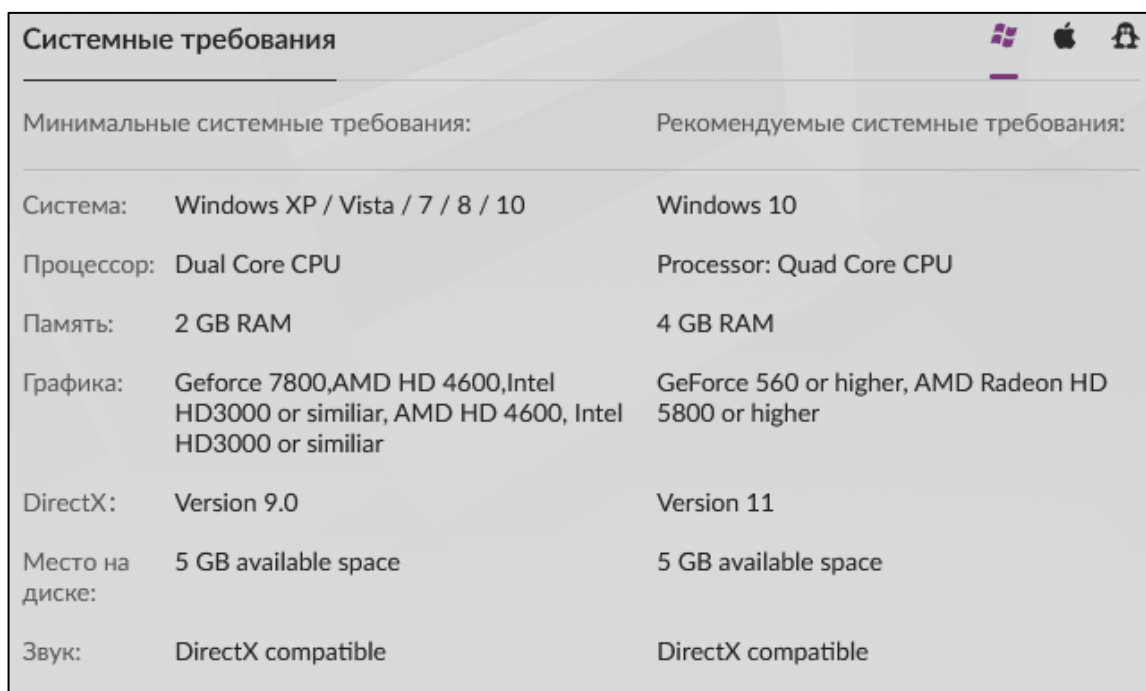
Данный компонент игровой структуры также включает в себя звуковое и музыкальное сопровождение. Постоянное звучание сирен отправившихся на задания полицейских машин и играющий на фоне джаз в стиле 50-х годов в полной мере способствуют игрокам представить себя в роли пожилого начальника полицейского департамента, находящегося на грани отставки.

4) Технология

Четвёртый элемент, формирующий компьютерную игру, является малозаметным для конечного пользователя, но при этом также играет немаловажную роль для получения им нового опыта. Инновационные инструменты для создания видеоигр, а также современное оборудование, предназначенное для их запуска, могут предоставить разработчикам обширные возможности для реализации задуманных игровых механик и завлекающей

подачи истории с помощью продвинутой графики и высококачественных звуковых эффектов.

Технологичность игры зависит от платформы, на которую она будет выпущена [9, стр. 1]. Проекты, созданные для веб-сайтов или мобильных устройств, являются примитивными с точки зрения каждой игровой составляющей. Игровые консоли и персональные компьютеры предназначены для крупномасштабных игр, хотя могут также включать в себя и более простые разработки. Для ПК характерны системные требования – список составляющих аппаратного обеспечения, необходимых для поддержания правильной работы приложения (Рисунок 1.6).



Системные требования				
Минимальные системные требования:		Рекомендуемые системные требования:		
Система:	Windows XP / Vista / 7 / 8 / 10	Windows 10		
Процессор:	Dual Core CPU	Processor: Quad Core CPU		
Память:	2 GB RAM	4 GB RAM		
Графика:	Geforce 7800,AMD HD 4600,Intel HD3000 or similiar, AMD HD 4600, Intel HD3000 or similiar	GeForce 560 or higher, AMD Radeon HD 5800 or higher		
DirectX:	Version 9.0	Version 11		
Место на диске:	5 GB available space	5 GB available space		
Звук:	DirectX compatible	DirectX compatible		

Рисунок 1.6. Системные требования для видеоигры «This is The Police»

1.3. Основные принципы работы игровых движков

Другим существенным фактором, влияющим на технологичность компьютерных игр, являются игровые движки, на которых они разрабатываются.

Данные инструментарии существенно упрощают производство видеоигр и представляют из себя наборы библиотек, которые поддерживают работу с графической, звуковой и физической составляющей создаваемого продукта, управлением с помощью устройств ввода данных, искусственным интеллектом, интерфейсом, а также сетевыми элементами [5, стр. 1888].

Конструирование игровых движков – сложный и трудоёмкий процесс, требующий высоких бюджетов и участие высококвалифицированных специалистов в области программирования. До середины 2000-х годов присутствие данного этапа разработки было зачастую распространённым явлением среди геймдевелоперов, поскольку в тот период приобретение сторонних лицензий имело огромную стоимость.

Так, для создания видеоигры Mafia: The City of Lost Heaven чешской студией Illusion Softworks был разработан инструментарий LS3D Engine. Отличительными элементами данного программного обеспечения являлись поддержка динамических теней и текстур высокого разрешения, а также наличие технологии обработки открытых пространств большого размера (Рисунок 1.7). Движок позже был использован для последующих проектов компании.



Рисунок 1.7. Изображение видеоигры «Mafia: The City of Lost Heaven»

Ситуация кардинально изменилась во второй половине первого десятилетия XXI века. Появились игровые движки Torque и Unity, имеющие необходимые элементы для создания полноценных проектов и доступные по цене для начинающих разработчиков. Одновременно с этим такие компании, как Epic Games и Crytek, изменили свою ценовую политику таким образом, чтобы большее количество пользователей могли беспрепятственно работать с инструментариями Unreal Engine и CryEngine.

Выбор движка зависит от концепции создаваемой игры. Одни инструменты предназначены для работы над 2D-проектами, в которых графика реализована с помощью спрайтов – неподвижных двухмерных картинок. Другое программное обеспечение лучше всего подходит для разработки трёхмерных видеоигр, включающих в себя анимирование объёмных объектов [3, стр. 73].

Управление в играх реализуется с помощью функций, реагирующих на нажатие определённых кнопок клавиатуры, мыши, геймпада, джойстика, руля или иных устройств ввода.

Интерфейс видеоигры создаётся с применением специальных объектов,

которые перекрывают изображение непосредственно игрового процесса. В процессе проектирования UI разработчик должен учитывать размер экрана устройств, для которых приложение выпускается.

Симуляция физических процессов осуществляется с помощью соответствующих компонентов, которые применяются на игровые объекты. Например, технология Havok Physics имеет высокооптимизированную библиотеку обнаружения столкновений и позволяет отслеживать динамику твёрдых тел в трёх измерениях.

Описание логики игровых элементов происходит посредством написания высокоуровневого кода, содержащего множество условий. Однако некоторые движки также позволяют использовать встроенные редакторы, которые имеют готовые скрипты типичных геймплейных ситуаций [5, стр. 1892]. Пример подобной реализации – система Blueprints, являющаяся частью Unreal Engine (Рисунок 1.8).

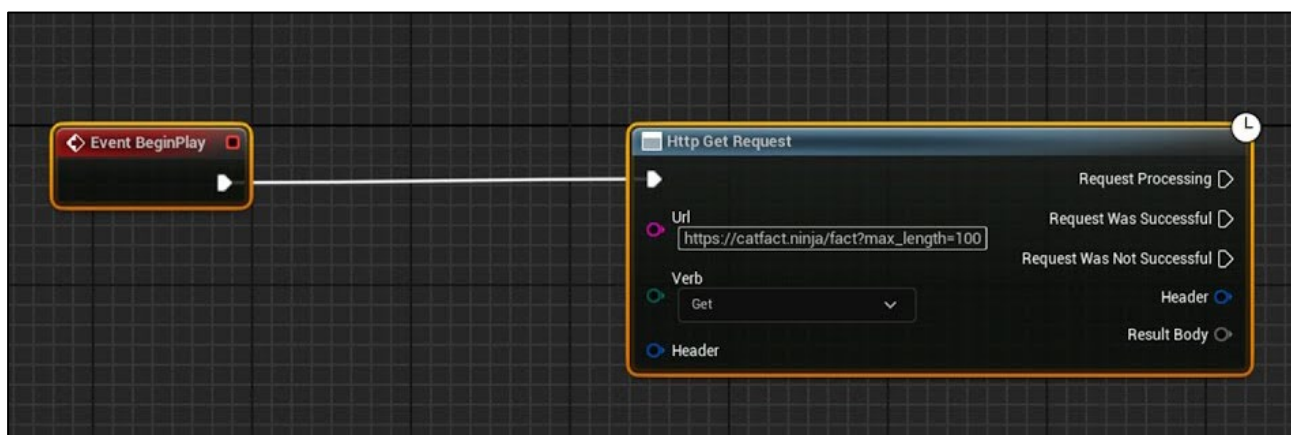


Рисунок 1.8. Интерфейс Blueprints

Игровые пространства представляют собой сцены, на которых выстраиваются различные объекты, взаимодействующие друг с другом во время геймплея [5, стр. 1893]. Главным объектом является камера, благодаря которой происходящие события демонстрируются на дисплее игровой платформы (Рисунок 1.9).

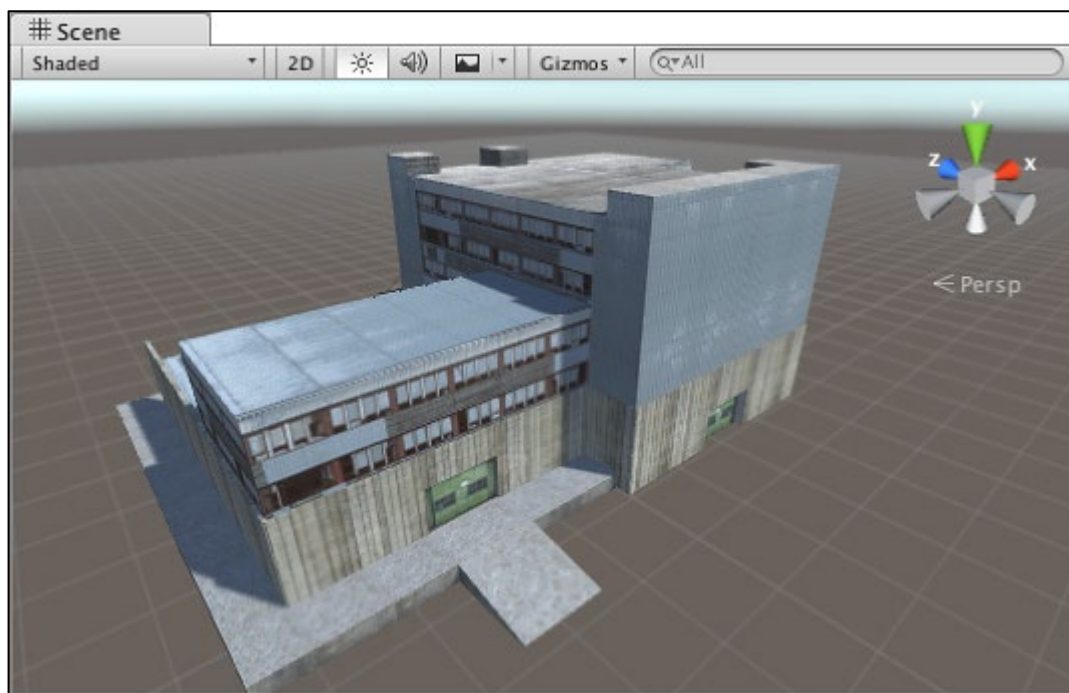


Рисунок 1.9. Пример сцены, созданной на игровом движке Unity

Опциональной частью разработки компьютерных игр является внедрение сетевых элементов, позволяющих пользователям взаимодействовать друг с другом в игровом процессе. В частности, для реализации многопользовательской составляющей в проектах, создаваемых с помощью инструментария Unity, используется библиотека Mirror, основанная на архитектуре одноранговой сети.

1.4. Анализ функционала Godot Engine

Godot Engine является современным бесплатным игровым движком, включающим в себя полный набор общих инструментов для создания двухмерных и трёхмерных видеоигр. Реализованные с его помощью проекты можно экспортировать на различные платформы, в том числе на мобильные устройства и игровые консоли.

Инструментарий разрабатывался с 2007 года двумя программистами из Аргентины – Хуаном Линецким и Ариэлем Манзуром – и первоначально являлся частным программным продуктом, который использовался в нескольких компаниях. Спустя 7 лет исходный код данного ПО под лицензией MIT был выложен на веб-сервис GitHub и игровой движок стал доступен для открытого пользования [6, стр. 57].

В настоящее время Godot Engine активно развивается. В 2016 году была выпущена версия 2.0, которая улучшила элементы для разработки трёхмерных игр, а также решила множество проблем, связанных с совместимостью. Основным нововведением Godot Engine 3.0 стало внедрение физически корректного рендеринга для 3D-объектов, который позволяет более достоверно отображать элементы игры с помощью моделирования реалистичного светового потока. Четвёртая версия игрового движка является последней на момент 2024 года: в ней был реализован новый модуль рендеринга на основе API Vulkan, обеспечивающий максимальную производительность графического процессора [18, стр. 14]. Также было осуществлено множество других изменений, улучшающих взаимодействие пользователя с инструментарием.

Godot Engine поддерживает работу с языками программирования C# и C++. Кроме того, для данного движка был разработан специальный высокоуровневый динамически типизированный язык GDScript, обеспечивающий написание простого структурированного кода и быстрое выполнение алгоритма. Его синтаксис имеет сходства с Python, при этом GDScript является более оптимизированным для работы с элементами игрового инструментария.

Ключевым элементом архитектуры данного программного обеспечения являются сцены. Их структура представляет собой деревья узлов, состоящие из функциональных единиц с уникальными свойствами и наименованием [6, стр. 59]. К сценам могут относиться как отдельные персонажи или предметы, так и полноценные уровни (Рисунок 1.10). Все сцены объединяются между собой иерархической связью, порядок их выполнения организуется непосредственно разработчиком проекта.

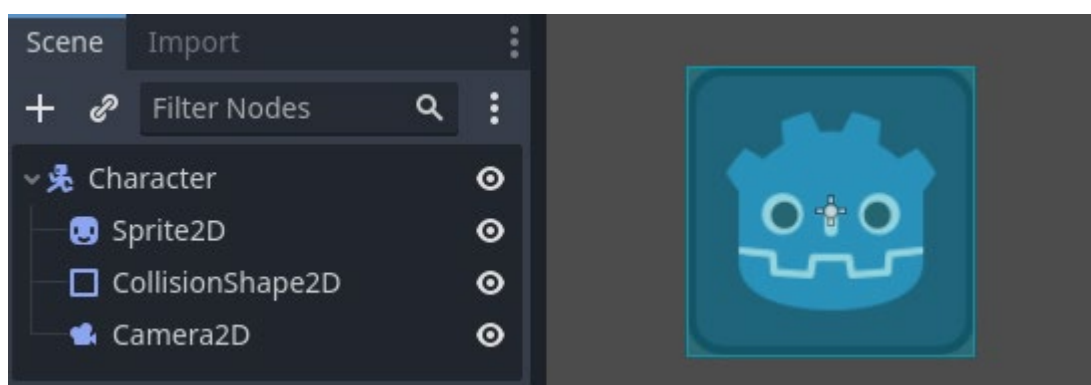


Рисунок 1.10. Сцена для игрового персонажа, за которым следует камера

Движок имеет встроенные редакторы для работы с различными компонентами компьютерных игр, таких как анимация объектов и звуковые эффекты. Например, Godot Engine имеет инструмент, позволяющий сделать скелет игрового персонажа. Его использование обеспечивает простоту в создании новых движений героя (Рисунок 1.11).

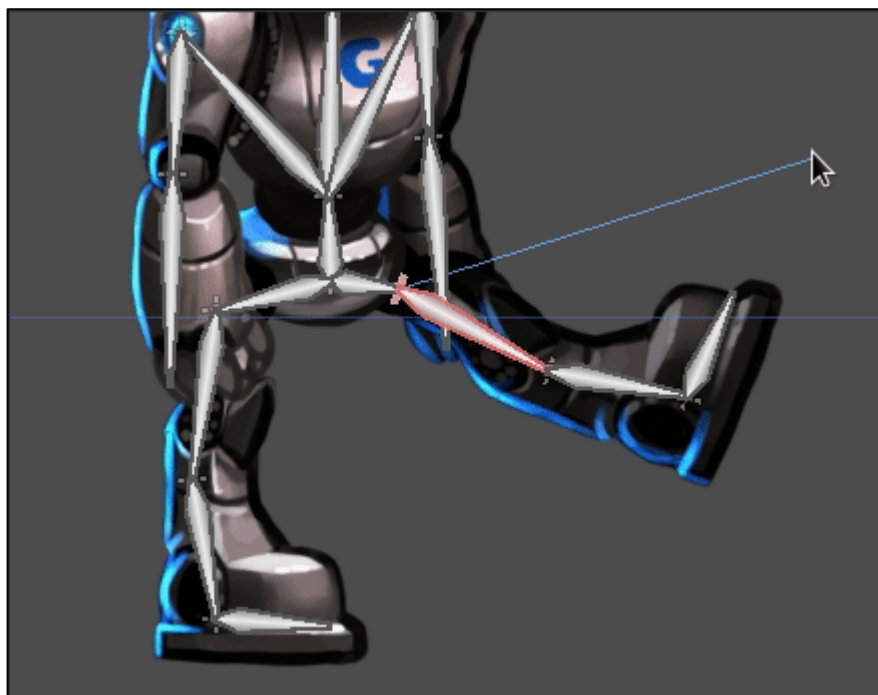


Рисунок 1.11. Пример скелетной анимации в Godot Engine

Немаловажной особенностью Godot Engine является его модульная структура. Пользователи могут внедрять в игровой движок новый функционал с помощью модулей, сконструированных другими разработчиками или созданных самостоятельно.

Выводы по главе 1

1. Изучены главные особенности геймдевелопмента. Выделены основные области знаний, необходимые для создания цифровых развлечений. Рассмотрена роль геймдизайнера в разработке видеоигр. Описаны основные компоненты игровых приложений.
2. Исследованы ключевые аспекты игровых движков. Был проанализирован их базовый функционал с демонстрацией конкретных примеров.
3. Проведён анализ инструментария для создания игр Godot Engine. Были выделены и изучены его основные отличительные черты относительно других игровых движков.

Глава 2. Создание игрового проекта

2.1. Разработка идеи для компьютерной игры

Видеоигра, реализуемая в рамках выпускной квалификационной работы, имеет жанр adventure. Суть игровых проектов данного типа заключается в рассказе истории, на развитие которой игрок влияет посредством осуществления конкретных команд с помощью определённых манипуляторов.

Проект представляет собой «симулятор ходьбы» – поджанр приключенческих игр, в котором история и художественный стиль имеют первостепенное значение. Отличительными чертами видеоигр данной категории являются приоритет в беспрепятственном перемещении по виртуальному миру и намеренно упрощённый игровой процесс. В этих играх, как правило, невозможно проиграть, а управление не предполагает использование большого количества кнопок и выполнение сложных комбинаций на устройствах ввода.

Визуальная составляющая приложения будет выполнена в двухмерном формате. Основу 2D-игр составляют спрайты – графические элементы, представляющие собой растровые изображения.

Наблюдение за игровым процессом осуществляется видом сбоку. Проекты, выполненные в данном стиле, называются сайд-скроллерами: каждое графическое содержимое в них передвигается в вертикальном и горизонтальном направлении.

Видеоигра предназначена для персональных компьютеров. Взаимодействие с игровым процессом осуществляется с помощью клавиатуры и мыши.

Компьютерная игра имеет название «Жизненный путь». Её сюжет основан на рассказе М. А. Шолохова «Судьба человека». Демонстрационная версия продукта включает в себя два уровня, представляющих собой адаптированные под интерактивное взаимодействие начальные эпизоды произведения.

В первом игровом сегменте пользователь берёт под собственный контроль главного героя, призванного на фронт. Задачи данного уровня представляют собой последовательность следующих действий: дойти до поезда вместе с женой

протагониста, обнять на прощание супругу героя и отправиться на войну, сев в вагон (Рисунок 2.1).

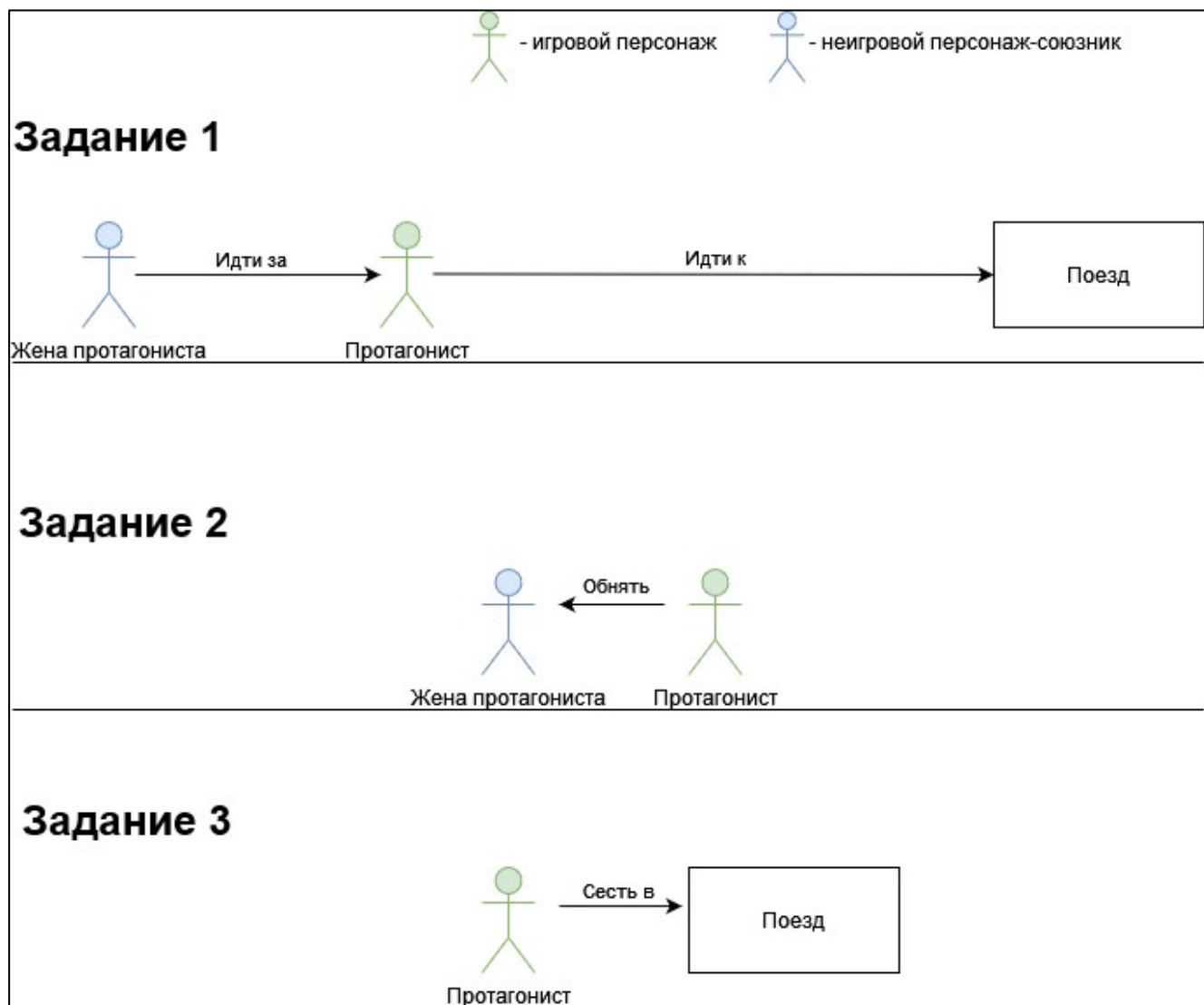


Рисунок 2.1. Концепт первого уровня игры

В следующем эпизоде игрок управляет автомобилем и его цель заключается в том, чтобы отвезти боеприпасы артиллерийской батарее. Однако миссия прервётся из-за взрыва снаряда, упавшего рядом с главным героем (Рисунок 2.2). В конце уровня активируется внутриигровой видеоролик, в котором около потерявшего сознание протагониста стоят два вооружённых немецких солдата.

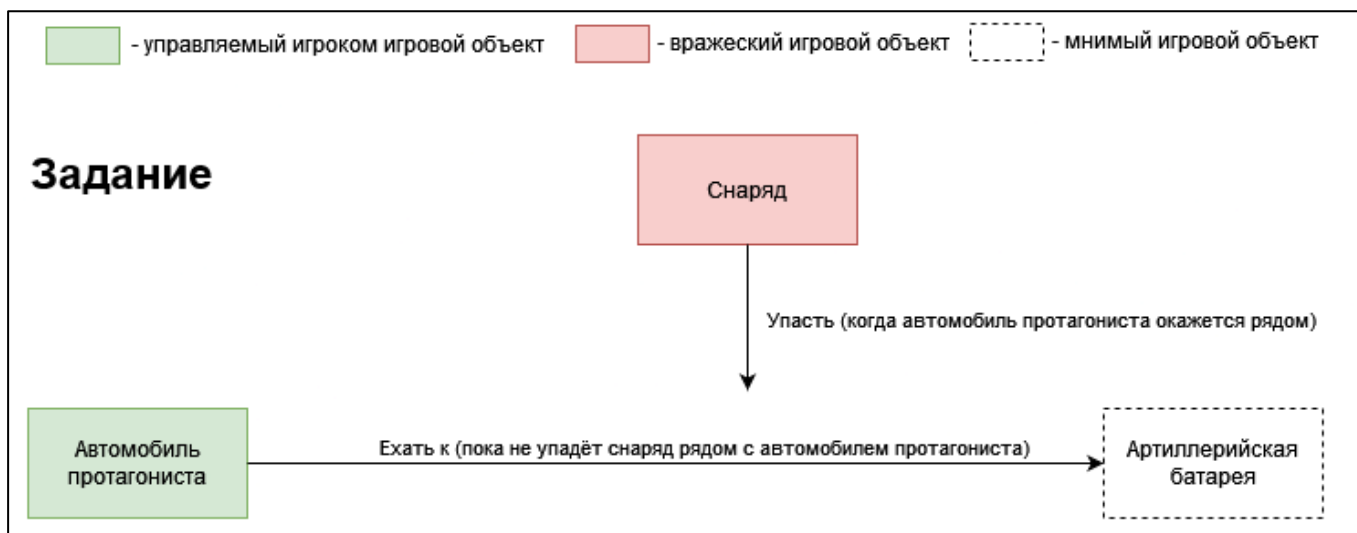


Рисунок 2.2. Концепт второго уровня игры

Выводы по разделу 2.1

1. Определён жанр разрабатываемой компьютерной игры.
2. Перечислены основные черты визуальной части создаваемого проекта.
3. Обозначена игровая платформа, для которой будет реализовано приложение.
4. Приведено подробное описание игрового процесса разрабатываемого продукта с демонстрацией концептов каждого уровня.

2.2. Создание визуальных элементов

Рисование спрайтов для компьютерной игры будет реализовано с помощью программного обеспечения Aseprite (Рисунок 2.3). Отличительной чертой данного инструмента является возможность анимирования объекта по его отдельным частям. Например, для анимации передвигающегося человека имеется возможность покадрово настроить изменение положения каждой ноги и руки.

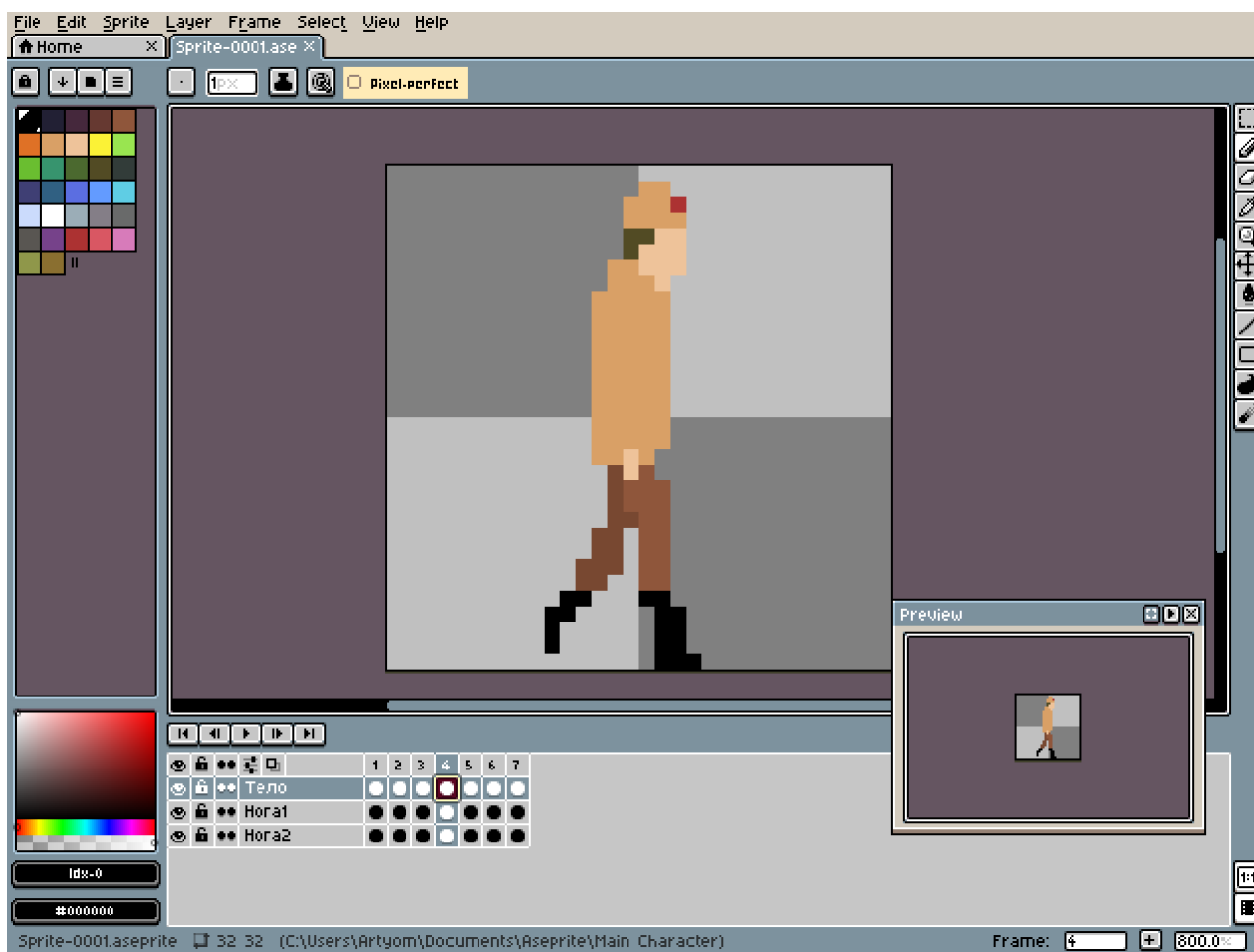


Рисунок 2.3. Интерфейс программы Aseprite

Для главного героя потребуется реализовать следующие движения: Walk (используется во время перемещения протагониста) и Hug (используется во время события, когда персонаж обнимает свою жену). Они экспортируются в виде листа спрайтов – изображения, состоящего из всех кадров конкретной анимации (Рисунки 2.4-2.5).

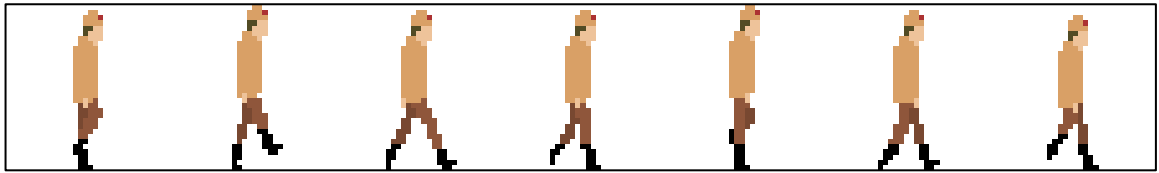


Рисунок 2.4. Анимация «Walk» для главного героя

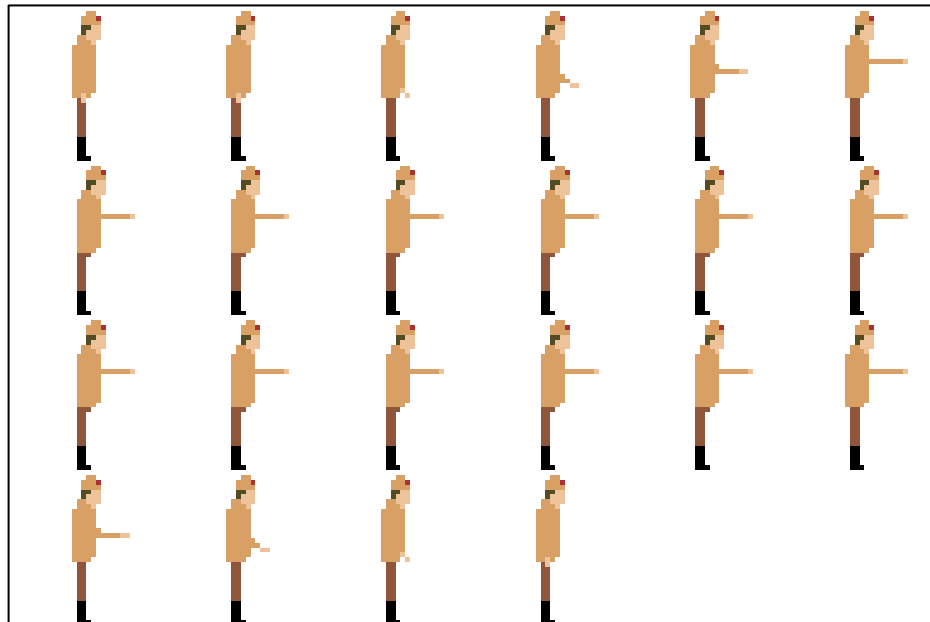


Рисунок 2.5. Анимация «Hug» для главного героя

Аналогичные анимации будут использоваться для действий жены протагониста (Рисунки 2.6-2.7).

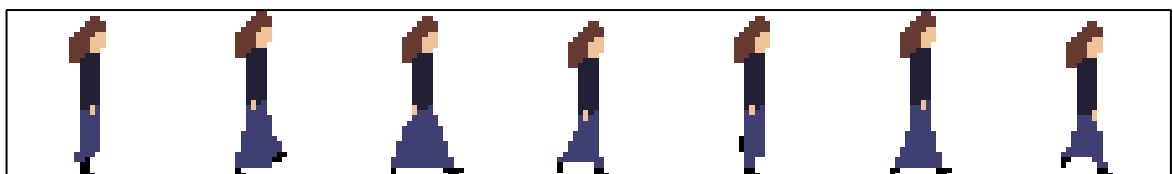


Рисунок 2.6. Анимация «Walk» для жены главного героя

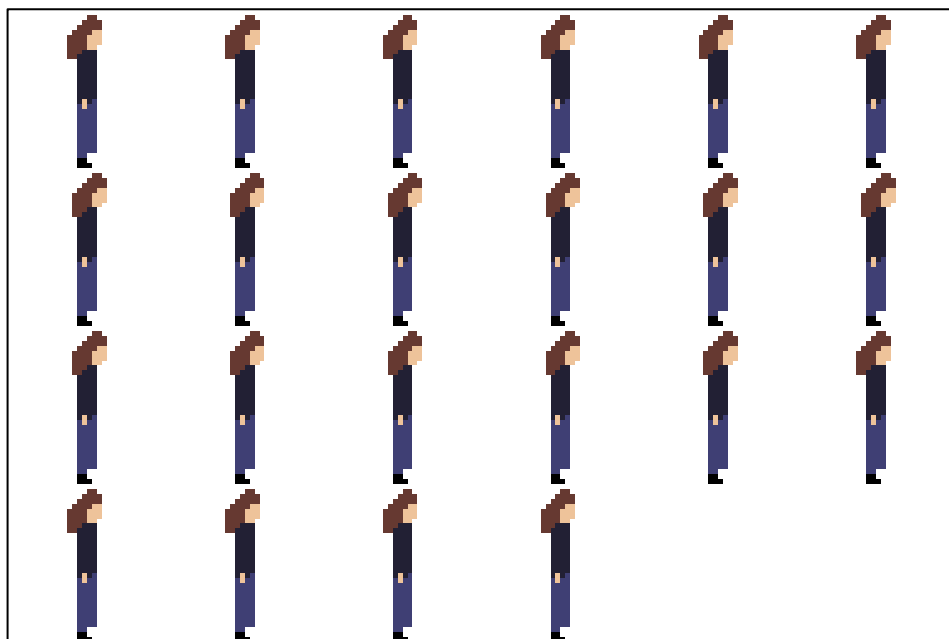


Рисунок 2.7. Анимация «Hug» для жены главного героя

Поскольку на втором уровне игры главному герою предстоит ехать на автомобиле, потребуется реализация отдельной анимации для транспортного средства (Рисунок 2.8).

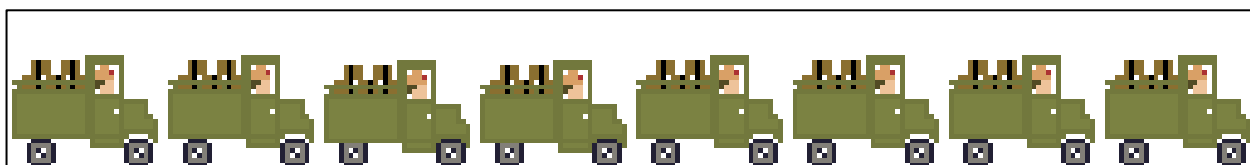


Рисунок 2.8. Анимация «Drive» для транспортного средства главного героя

Кроме того, в игре будут нужны статичные изображения персонажей и объектов, которые будут использоваться во время их бездействия (Рисунки 2.9-2.13).

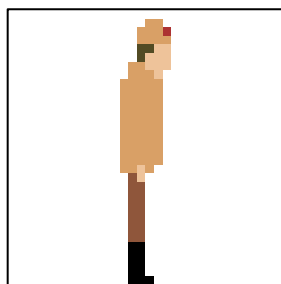


Рисунок 2.9. Изображение главного героя в бездействии

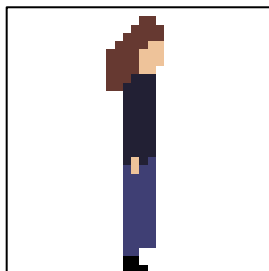


Рисунок 2.10. Изображение жены главного героя в бездействии

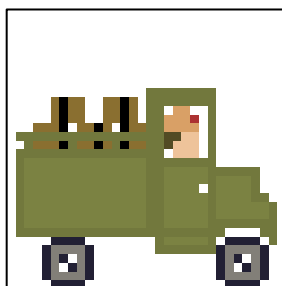


Рисунок 2.11 Изображение автомобиля главного героя в бездействии

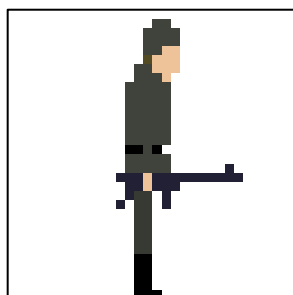


Рисунок 2.12. Изображение врага в бездействии

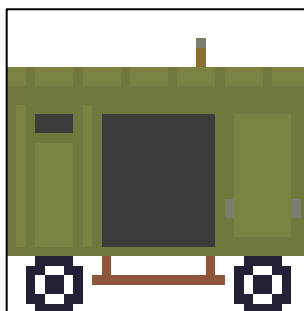


Рисунок 2.13. Изображение вагона поезда в бездействии

Важной составляющей для разработки визуальной части компьютерной игры являются тайлы – повторяющиеся изображения, из которых складывается вид игровых локаций. В проекте будут представлены четыре вида данного типа графических элементов (Рисунки 2.14-2.17).



Рисунок 2.14. Тайл «Земля с травой»



Рисунок 2.15. Тайл «Земля с рельсами на поверхности»



Рисунок 2.16. Тайл «Земля»



Рисунок 2.17. Тайл «Трава»

Выводы по разделу 2.2

1. С помощью программы Aseprite нарисованы спрайты персонажей и объектов игры.
2. Реализованы анимации некоторых персонажей и объектов игры.
3. Созданы несколько видов тайлов для конструирования уровней видеоигры.

2.3. Реализация первого уровня видеоигры

В параметрах создания проекта на движке Godot Engine был выбран метод рендеринга «Совместимость». Данный режим, несмотря на наименее продвинутую графику, позволит разработанным играм без затруднений работать даже на самых бюджетных устройствах.

Используется размер проекта 640x360. Данная настройка позволит масштабировать экран игры до разрешений 1920x1080 и 2560x1440, которые являются самыми распространёнными для современных персональных компьютеров согласно статистике Steam 2024 года.

Внедрим в файловую систему проекта нарисованные ранее спрайты (Рисунок 2.18).

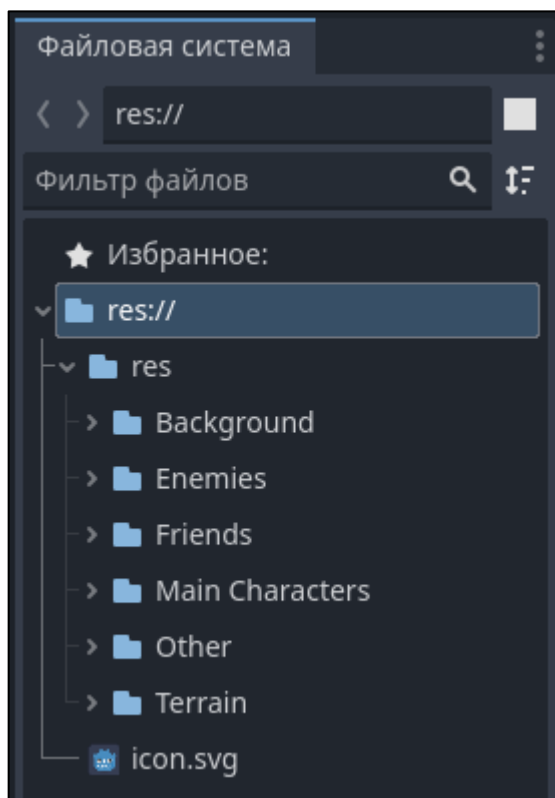


Рисунок 2.18. Добавление директории со спрайтами в проект

Создадим сцену `CharacterBody2D` (далее будет иметь название `Player`), которая будет отвечать за параметры, связанные с игровым персонажем. В ней должен содержаться узел `CollisionShape2D`, позволяющий взаимодействовать с

другими объектами игры.

Для анимирования протагониста добавим узел AnimatedSprite2D и добавим в него кадры выполнения соответствующих действий (Рисунок 2.19).

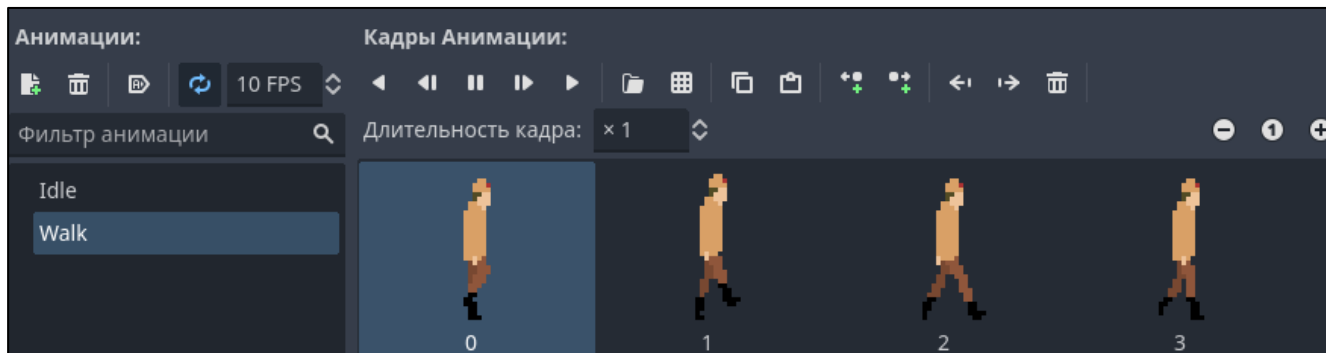


Рисунок 2.19. Интерфейс для работы с кадрами анимации

Подставим в форму узла CollisionShape2D изображение главного героя игры в бездвижном состоянии (Рисунок 2.20). Для обеспечения наилучшего качества отображения объектов в настройках проекта выбрана режим фильтрации текстур «Nearest».

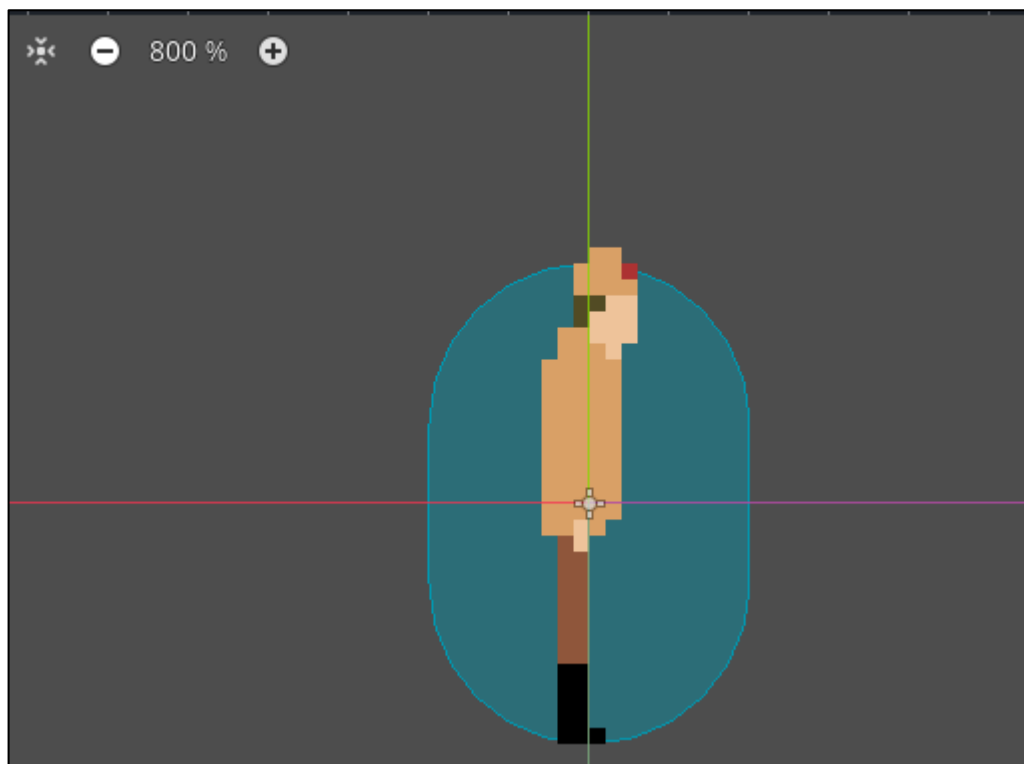


Рисунок 2.20. Создание оболочки игрового персонажа

Скрипт управляемого игроком героя включает в себя константу *SPEED*, хранящую в себе значение скорости движения персонажа, а также переменные *anim* и *gravity*, содержащие соответственно анимации и встроенные в движок настройки гравитации (Рисунок 2.21).

```
1  extends CharacterBody2D
2
3  const SPEED = 20.0
4  @onready var anim: AnimatedSprite2D = $AnimatedSprite2D
5
6  var gravity = ProjectSettings.get_setting("physics/2d/default_gravity")
```

Рисунок 2.21. Константы и переменные скрипта игрового персонажа

В коде содержатся функции *_ready* (отвечает за команды, выполняющиеся перед запуском игры), *_physics_process* (отвечает за команды, связанные с поведением протагониста в игровом мире) и *update_animation* (отвечает за команды, связанные с проигрыванием соответствующих анимаций во время игрового процесса) (Рисунок 2.22).

```
8  func _ready() -> void:
9      anim.play('Idle')
10
11  func _physics_process(delta):
12      if not is_on_floor():
13          velocity.y += gravity * delta
14
15      var direction = Input.get_action_raw_strength("ui_right")
16      if direction:
17          velocity.x = direction * SPEED
18      else:
19          velocity.x = move_toward(velocity.x, 0, SPEED)
20
21      update_animation()
22      move_and_slide()
23
24  func update_animation():
25      if velocity.x:
26          anim.play("Walk")
27      else:
28          anim.play("Idle")
```

Рисунок 2.22. Функции скрипта игрового персонажа

Чтобы настроить управление с помощью устройств ввода для создаваемой игры, используем окно «Список действий», расположенное в настройках проекта (Рисунок 2.23).

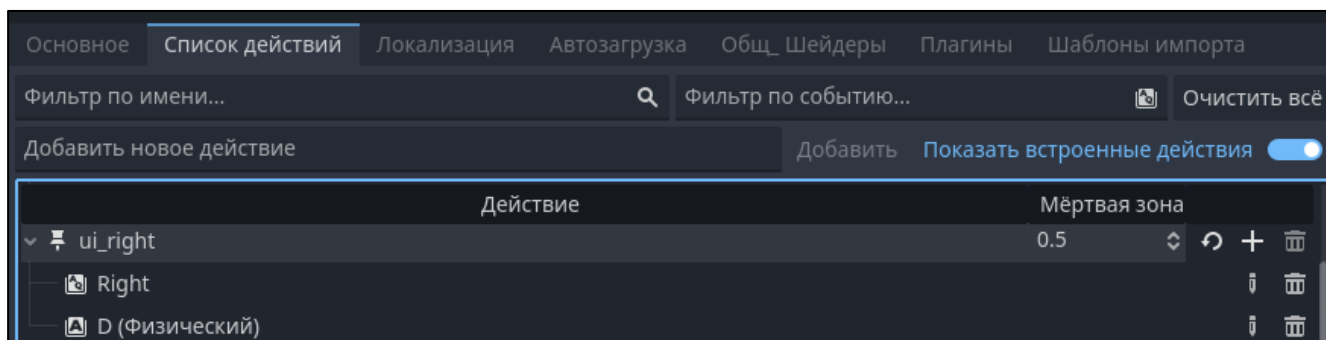


Рисунок 2.23. Интерфейс для настройки управления в компьютерной игре

Создадим сцену первого уровня игры. Добавим в неё игрового персонажа и узел TileMap, в который будут добавлены спрайты платформ, по которым игрок будет передвигаться. Настроим для данных объектов коллизию с помощью свойств Collision и Physics Layers, затем разместим их на главном окне. Также добавим невидимые стены путём добавления узла StaticBody2D по краям уровня, чтобы игрок не смог выйти за пределы сцены (Рисунок 2.24).

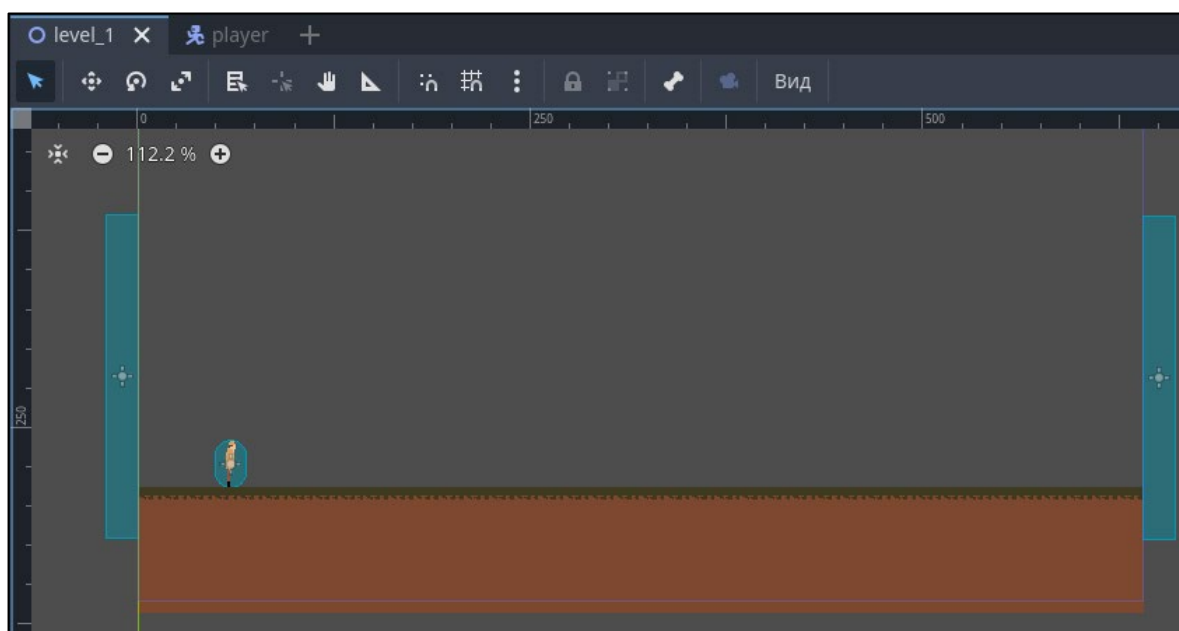


Рисунок 2.24. Базовая версия сцены Level_1

Дополним сцену Player узлом Camera2D – он позволит игровой камере следовать за игроком. Настроим параметры Zoom для приближения объектива к персонажу и Limit, чтобы камера не смогла выходить за пределы сцены (Рисунок 2.25).

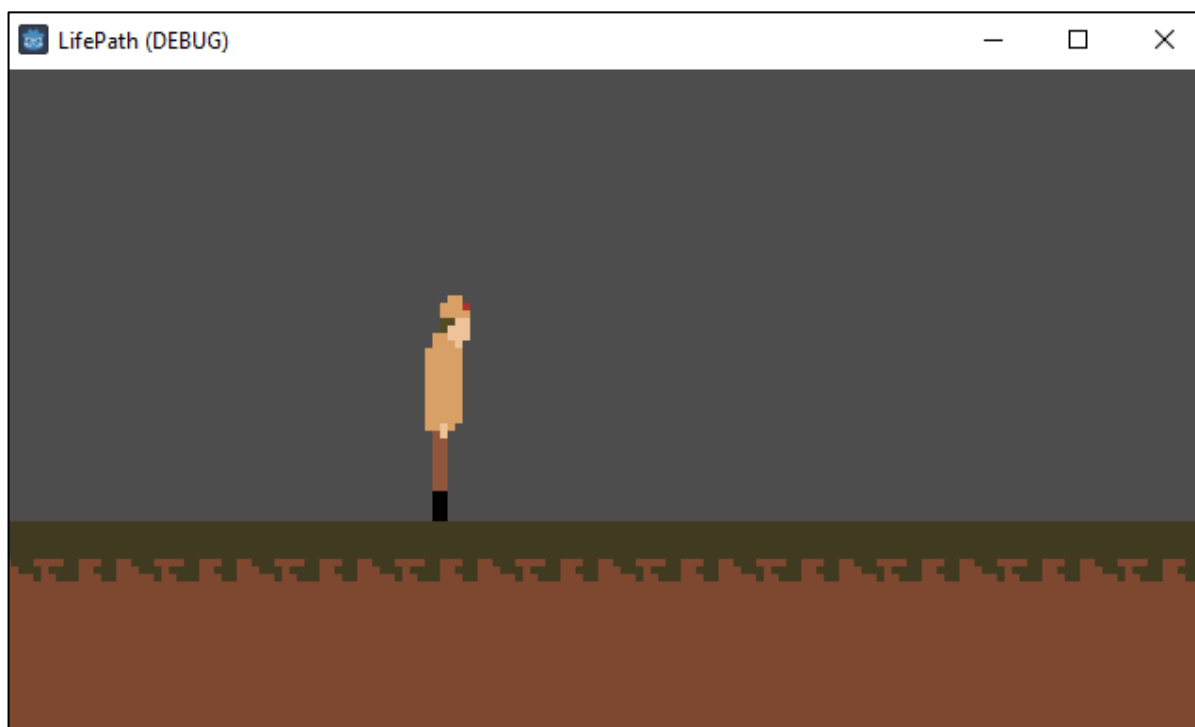


Рисунок 2.25. Вид компьютерной игры с настроенной камерой

Добавим движущийся фон игровому уровню. Для этого на сцене Level_1 создадим ParallaxBackground, который будет содержать ParallaxLayer с заготовленной текстурой для заднего плана игры. Изменим значения параметров Scale и Mirroring, чтобы фон двигался медленнее персонажа и при этом мог повторяться (Рисунок 2.26).

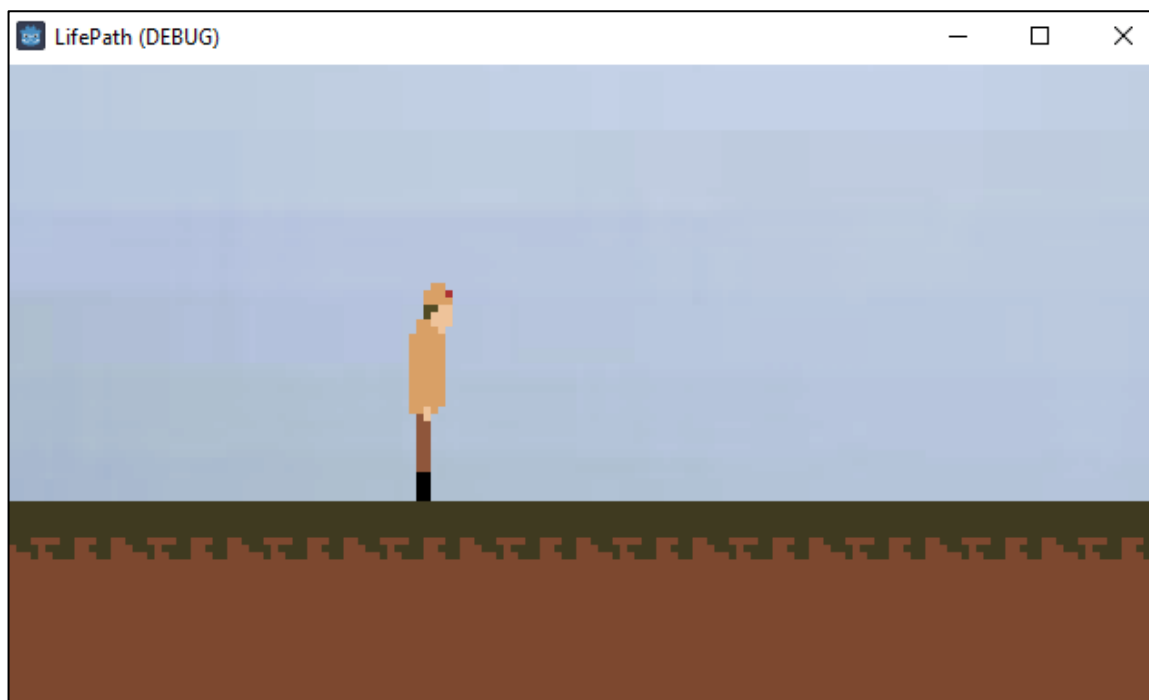


Рисунок 2.26. Вид компьютерной игры с добавленным задним фоном

Создадим дружеского неигрового персонажа – жену главного героя. Структура её сцены также будет включать в себя узлы `CollisionShape2D` и `AnimatedSprite2D`. Основа скрипта будет идентичной коду протагониста: таким образом, она будет перемещаться по игровому миру в тот момент, когда будет двигаться игрок (Рисунок 2.27).

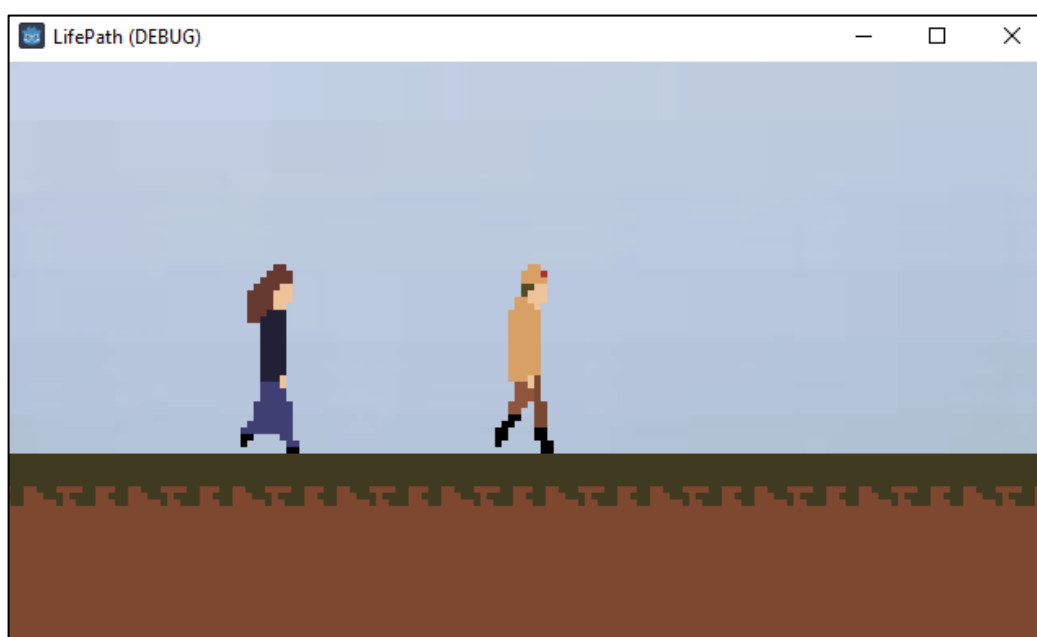


Рисунок 2.27. Демонстрация следования неигрового персонажа за пользователем

Уровень будет завершаться посадкой в вагон поезда. Добавим данный объект на игровой уровень и внедрим в него триггер – механизм, который проверяет присутствие игрока в заданном поле и в зависимости от этого активирует определённое событие (Рисунок 2.28).

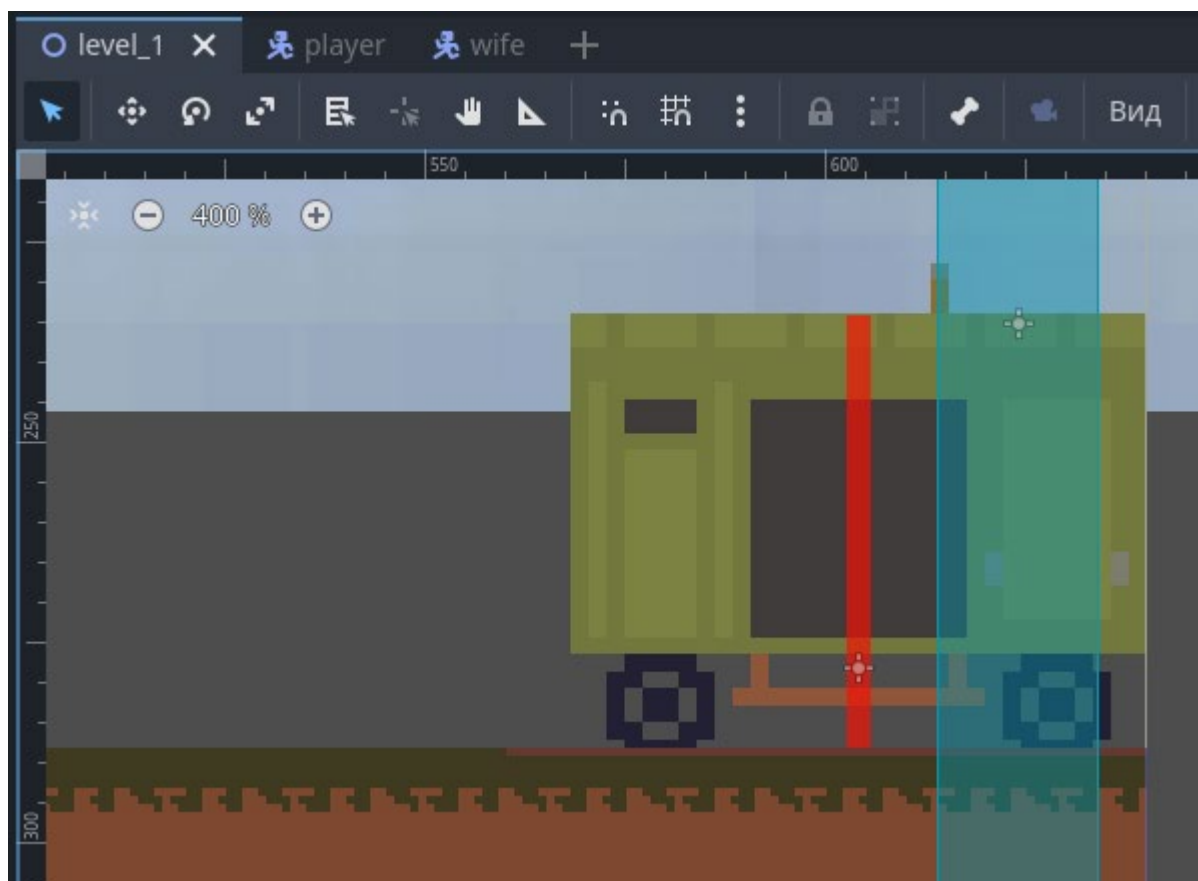


Рисунок 2.28. Объект «Вагон» с полем активации события

(Далее в этом разделе будет описан процесс реализации внутриигрового видео и переход к нему через данный триггер)

Выводы по разделу 2.3

1. Произведена настройка проекта в Godot Engine.
2. В игру были встроены спрайты и анимации, созданные с помощью Aseprite.

3. Написаны скрипты и настроены параметры, которые позволяют персонажам и объектам взаимодействовать друг с другом в игровом мире.
4. Была встроена игровая камера, которая следует за движениями игрока.
5. Создан задний фон, движущийся параллельно перемещению пользователя по игровому миру.
6. Реализован первый уровень компьютерной игры.

2.4. Реализация второго уровня видеоигры

(В данном разделе будет представлен процесс наследования игровых уровней, реализация езды на автомобиле и создание завершающей катсцены для демонстрационной версии проекта)

Выводы по разделу 2.4

1. С

2.5. Разработка главного меню

(В данном разделе будет представлена разработка главного меню проекта, с помощью которого можно начать игру, выбрать определённый уровень и выйти из игры)

Выводы по разделу 2.5

1. С

2.6. Внедрение звуковых эффектов

Музыкальное сопровождение компьютерной игры написано с использованием онлайн-сервиса Apronus.com. Данное веб-приложение позволяет пользователю играть на виртуальном пианино посредством клавиатуры и мыши, а также записывать создаваемую мелодию (Рисунок 2.). Музыку имеется возможность сохранить на компьютер в формате .mid.

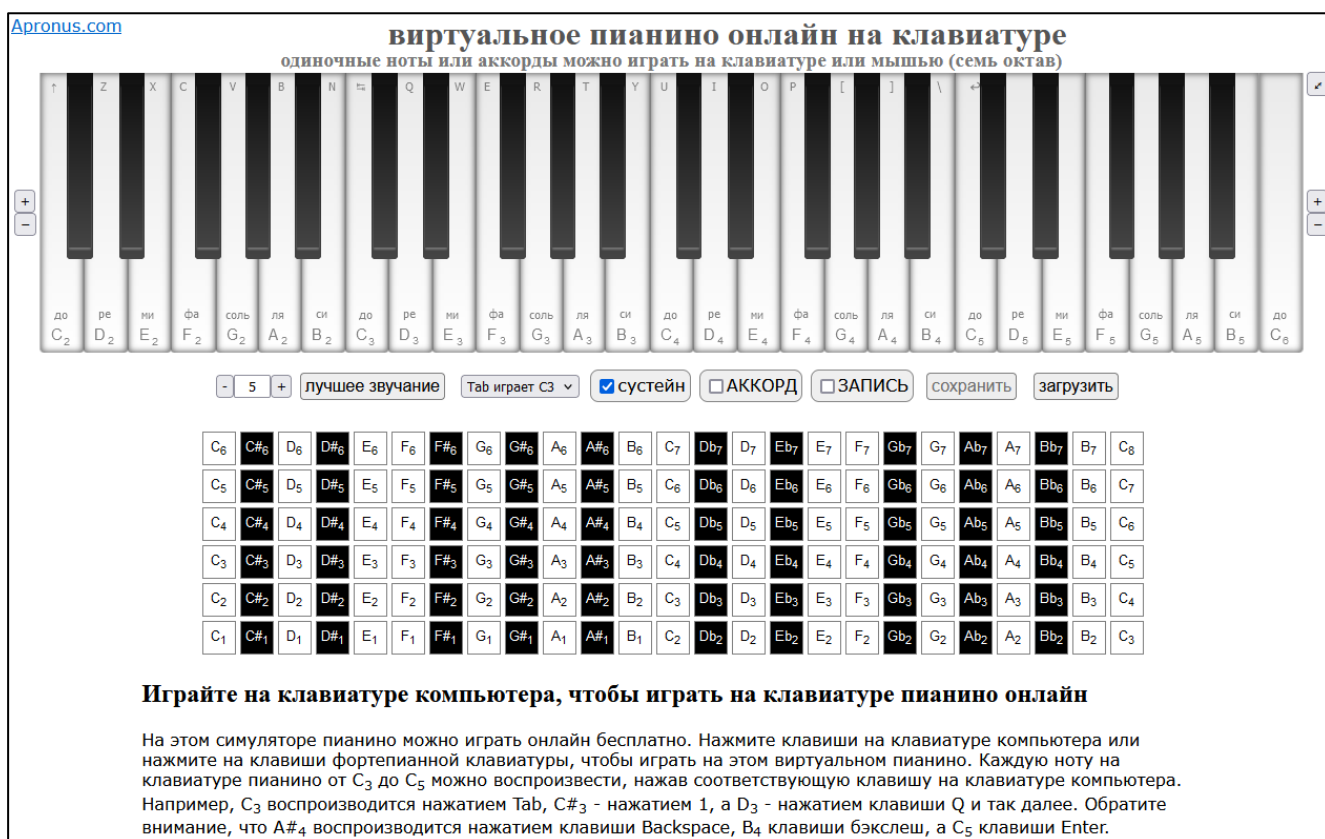


Рисунок 2.. Интерфейс онлайн-сервиса Apronus.com

Саундтрек проекта включает в себя спокойные и меланхоличные композиции, которые были сыграны на высоких нотах. Они позволяют передать угнетающую атмосферу страны, окутанной страшной войной, и настроение людей, до последнего верящих в благополучное будущее. Также есть мелодии на низких нотах, которые проигрываются во время динамичных и напряжённых эпизодов сюжета.

Для полного погружения в мир игры в проект должны быть внедрены

аудио-эффекты, воспроизводящие звуки окружения, например, шаги людей или шумы транспортных средств.

Были использованы бесплатные звуковые эффекты из следующих электронных ресурсов:

- zapsplat.com;
- fesliyanstudios.com;
- partnersinrhyme.com.

(Далее будет представлен ход действий по встраиванию музыкальных композиций в игровой процесс и катсцены, а также по добавлению аудио-эффектов к игровым объектам)

Выводы по разделу 2.6

1. С

2.7. Экспорт проекта

(В данном разделе будет представлен процесс экспортирования компьютерной игры и её запуск на уровне пользователя)

Выводы по разделу 2.7

1. С

Заключение

Цель выпускной квалификационной работы заключалась в разработке демонстрационной версии компьютерной игры на движке Godot Engine.

Для достижения данной цели были решены следующие задачи:

- проанализированы главные компоненты, из которых состоят видеоигры;
- изучены особенности игровых движков;
- исследованы возможности инструментария Godot Engine;
- составлен концепт компьютерной игры, описывающий жанр проекта, сценарий, визуальное оформление и поддерживаемые платформы;
- разработаны визуальная часть и звуковое составляющее видеоигры с помощью компьютерных технологий;
- с использованием Godot Engine созданы игровые уровни, реализующие разработанный концепт.

Таким образом, все поставленные задачи были выполнены и цель выпускной квалификационной работы была успешно достигнута.

Библиография

1. Грузин Н. А. Сравнение движков для разработки игр: Godot Engine и Unity // Modern science. – 2021. – С. 440-444.
2. Дёминов С. В. Создание игры-платформера в среде Godot engine // Северо-Восточный федеральный университет имени М.К. Аммосова. – 2022. – С. 220-225.
3. Дмитриев В. С., Зоткина А. А. Сравнительный анализ мультиплатформенных движков для разработки игр // Современные технологии: актуальные вопросы, достижения и инновации. – 2019. – С. 71-73.
4. Згонникова А. О., Прокопенко А. А. Разработка игр // Наука, образование, инновации: актуальные вопросы и современные аспекты. – 2023. – С. 37-39.
5. Колесников В. Д., Бондаренко Т. В. Разработка компьютерных игр с помощью игровых движков // Белгородский государственный технологический университет им. В. Г. Шухова. – 2020. – С. 1888-1894
6. Ломайкин А. С. Godot Engine в современной игровой индустрии // E-Scio. – 2023. – С. 56-63.
7. Малкова А. Д., Белозерова С. И. Основы разработки игр на кроссплатформенном игровом движке «Godot Engine» // Дальневосточный государственный университет путей сообщения. – 2022. – С. 114.
8. Мататов А. Д., Карпеня Д. А. Отличия разработки игр от разработки «обычных» ИТ-проектов. // Белорусский государственный университет информатики и радиотехники – 2019. – С. 92.
9. Моисеенков Д. М. Геймдев в современном мире // Россия молодая. – 2022. – С. 1-4.
10. Окорочков Э. С., Сивцев Н. Н., Протоद्याконова Г. Ю. Разработка инди игр // Вестник современных исследований. – 2018. – С. 634-636.

- 11.Перевощикова К. Ф., Попова А. Г., Иванчатенко А. А. Обзор программного обеспечения для разработки компьютерных игр // Дни науки. – 2021. – С. 65-70.
- 12.Прихач Я. А. Анализ рынка игровой индустрии и игр приключенческого жанра. // Белорусский государственный университет – 2023. – С. 146-148.
- 13.Телега Е. Д., Шерстобитов К. В., Мозговенко А. А. Дизайн главного героя 2D-платформера, разработанного в инструментарии Godot // Мелитопольский государственный университет. – 2023. – С. 138-145.
- 14.Турантаев А. В., Голиков Г. Г. Программирование игр, алгоритмы и технологии // Инновации. Наука. Образование. – 2021. – С. 1183-1192.
- 15.Шелл Д. Геймдизайн: Как создать игру, в которую будут играть все // Альпина Паблишер. – 2019. – 640 с.
- 16.Bradfield C. Godot Engine Game Development projects: Build five cross-platform 2D and 3D games with Godot 3.0 // Packt Publishing Ltd. – 2018. – 298 с.
- 17.Manzur A., Marques G. Godot Engine Game Development in 24 Hours, Sams Teach Yourself: The Official Guide to Godot 3.0 // Sams Publishing. – 2018. – 432 с.
- 18.Salmela T. Game development using the open-source Godot Game Engine // Tampere University of Applied Sciences. – 2022. – 67 с.
- 19.Wolf M. J. P. Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming [3 volumes] // Bloomsbury Publishing USA. – 2021. – С. 248-252.