**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»**

Высшая школа цифровых технологий

Кафедра математики и прикладных информационных технологий

**ДОПУСТИТЬ К ЗАЩИТЕ**

Зав. кафедрой математики и

прикладных информационных

технологий

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Барбаков О. М.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

**РАЗРАБОТКА ИГРЫ НА БАЗЕ ДВИЖКА GODOT С ДОБАВЛЕНИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К бакалаврской работе

|  |  |
| --- | --- |
| **НОРМОКОНТРОЛЕР:**  ст. преподователь  *\_\_\_\_\_\_\_\_\_\_\_ Михайлова О.В.* | **РУКОВОДИТЕЛЬ:**  *доцент, к.п.н.*  *\_\_\_\_\_\_\_\_\_\_\_ Спирин И.С.* |
|  | **РАЗРАБОТЧИК:**  *обучающийся группы РИСб-20-1*  *\_\_\_\_\_\_\_\_\_\_\_\_ Никитин М.Д.* |
|  | Бакалаврская работа  защищена с оценкой \_\_\_\_\_\_\_\_\_\_\_  Секретарь ГЭК  \_\_\_\_\_\_\_\_\_\_\_\_\_*Кондратьева Н.А.* |

Тюмень, 2024

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»**

Высшая школа цифровых технологий

Кафедра математики и прикладных информационных технологий

**ДОПУСТИТЬ К ЗАЩИТЕ**

Зав. кафедрой математики и

прикладных информационных

технологий

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Барбаков О. М.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

**ЗАДАНИЕ**

На выпускную квалификационную работу (бакалаврскую работу)

Ф.И.О. обучающегося: Никитин М.Д.

Ф.И.О. руководителя ВКР: Спирин И. С.

Тема ВКР: «Разработка игры на базе движка godot с добавлением искусственного интеллекта»

утверждена приказом по Высшей школе цифровых технологий от \_\_.\_\_.2024 г. №\_\_

Срок предоставления завершенной ВКР на кафедру «\_\_» июня 2024 г.

Исходные данные к ВКР получены при изучении деятельности «\_\_\_\_\_\_\_\_\_\_».

Содержание пояснительной записки

|  |  |  |
| --- | --- | --- |
| Наименование главы, раздела | % от объема ВКР | Дата выполнения |
| 1 Анализ предметной области |  |  |
| 2 Создание визуальной концепции интерфейса и схемы работы игры |  |  |
| 3 Разработка геймплейной составляющей |  |  |

Дата выдачи задания \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Аннотация**

Целью данной выпускной квалификационной работы является разработка игры на движке Godot.

Объектом работы является игра на движке Godot.

Предметом исследования является процесс разработки игры на движке Godot.

Теоретической основой работы послужили учебные издания и публикации отечественных и зарубежных авторов в области разработки игр в целом и на движке Godot в частности.

Информационная база исследования: электронные ресурсы, статьи, учебные пособия и техническая документация в области разработки игр на движке Godot.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованных источников, приложения.

**Abstract**

The purpose of this final qualifying work is to develop a game on the Godot engine.

The object of the work is a game made in the Godot engine.

The subject of the study is the process of developing a game using the Godot engine.

The theoretical basis of the work was educational publications and publications of domestic and foreign authors in the field of game development in general and on the Godot engine in particular.

Research information base: electronic resources, articles, tutorials and technical documentation in the field of game development on the Godot engine.

The final qualifying work consists of an introduction, three chapters, a conclusion, a list of sources used, and application.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc163784219)

[Теоретические аспекты разработки игр 8](#_Toc163784220)

[1.1 Основные понятия 8](#_Toc163784221)

[1.2 Этапы разработки видеоигры 11](#_Toc163784222)

[1.3 Анализ предметной области 15](#_Toc163784223)

[ЗАКЛЮЧЕНИЕ 18](#_Toc163784224)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 19](#_Toc163784225)

# ВВЕДЕНИЕ

С развитием информационных технологий и ростом популярности видеоигр, их разработка становится все более актуальной и востребованной областью в ИТ-индустрии. Видеоигры одновременно являются средством развлечения, популяризации новых технологий, а также обучения и образования.

Неотъемлемой частью разработки видеоигр стали игровые движки. Они являются специализированной средой разработки и позволяют разработчикам сосредоточиться на создании контента и геймплея, а не о написании функционала с нуля. Одним из самых быстрорастущих игровых движков в индустрии, на данный момент, является Godot. Имея полностью бесплатную модель пользования, открытый код, свой собственный язык программирования, предназначенный специально для разработки игр и разработчиков, активно прислушивающихся к мнению пользователей, данный движок набирает популярность как среди разработчиков, так и среди инвесторов.

Программисты Хуан Линьетски (англ. Juan Linietsky) и Ариель Манзур (англ. Ariel Manzur) начали разработку движка в 2007 году, и через некоторое время он уже использовался как закрытый программный продукт в некоторых компаниях. В феврале 2014 года разработчики приняли решение сделать технологию открытой, и выложили исходники движка под лицензией MIT на GitHub под названием Godot.

15 декабря 2014 движок достиг версии 1.0, первой стабильной версии нового продукта. В ней были добавлены дополнительные шейдеры, navimesh и lightmapping.

4 ноября 2015 проект Godot присоединился к организации Software Freedom Conservancy.

23 февраля 2016 вышла стабильная версия 2.0. Основные акценты в её разработке ставились на существенное повышение удобства работы в движке через рабочее окружение.

22 июня 2016 проекту Godot от Mozilla Open Source Support (MOSS) «Mission Partners» присуждена сумма в $20,000 для того, чтобы добавить в проект поддержку WebSocket, WebAssembly и WebGL 2.0.

# Анализ предметной области

## Основные понятия

Компьютерная игра – компьютерная программа, служащая для организации игрового процесса (геймплея), связи с партнёрами по игре, или сама выступающая в качестве партнёра.

Геймплей – компонент игры, отвечающий за взаимодействие игры и игрока. Геймплей описывает, как игрок взаимодействует с игровым миром, как игровой мир реагирует на действия игрока и как определяется набор действий, который предлагает игроку игра.

Игровой движок – базовое программное обеспечение компьютерной игры.

Игровой ассет или игровой ресурс – цифровой объект, преимущественно состоящий из однотипных данных, неделимая сущность, которая представляет часть игрового контента и обладает некими свойствами. Понятие «игрового ассета» используется при разработке компьютерных игр по отношению к тем элементам контента, которые обрабатываются ресурсной системой как неделимые (атомарные, элементарные) сущности.

Игровой искусственный интеллект – набор программных методик, которые используются в компьютерных играх для создания иллюзии интеллекта в поведении персонажей, управляемых компьютером. Игровой ИИ, помимо методов традиционного искусственного интеллекта, включает также алгоритмы теории управления, робототехники, компьютерной графики и информатики в целом.

Игровая механика – набор правил и способов, реализующий определённым образом некоторую часть интерактивного взаимодействия игрока и игры. Все множество игровых механик игры формируют конкретную реализацию её игрового процесса или геймплея.

Godot – это универсальный 2D и 3D игровой движок, спроектированный для поддержки всех видов проектов.

GDScript – это высокоуровневый, объектно-ориентированный, императивный язык программирования с постепенной типизацией, созданный для Godot. Он использует синтаксис на основе отступов, аналогичный таким языкам, как Python. Его цель – оптимизация и тесная интеграция с Godot Engine, обеспечивая большую гибкость при создании и интеграции контента.

Aseprite – это редактор изображений с доступным исходным кодом, предназначенный в первую очередь для рисования и анимации в стиле пиксель-арт. Он работает на Windows, macOS и Linux и содержит различные инструменты для редактирования изображений и анимации, такие как слои, кадры, поддержка тайловых карт, интерфейс командной строки и другие.

Тайловая, плиточная или знакоместная графика (от англ. tile – плитка) – метод создания больших изображений (как правило, уровней в компьютерных играх) из маленьких фрагментов одинаковых размеров.

ChipTone – бесплатный инструмент для генерации звуковых эффектов, распространяющийся по лицензии CC0, что означает, любые коммерческие и некоммерческие проекты могут выполняться без разрешения автора приложения.

Основными элементами в Godot являются:

Узел (Или нода) – самый маленький строительный блок игры. В качестве свойств могут иметь имя, принадлежность к потоку обработки, свойства обработки (Обрабатывается ли узел только во время паузы, только когда нет паузы, всегда, никогда или наследует это свойство от родителя). К каждому узлу можно прикрепить не больше одного скрипта. Существует огромное количество узлов под самые разные задачи.

Скрипт – файл, содержащий в себе определенный код. В Godot скрипты обрабатываются только если они прикреплены к узле или загружены отдельно путем подключения автозагрузки (Принцип проектирования «Одиночка» иди «Singleton»).

Одиночка – порождающий шаблон проектирования, гарантирующий, что в однопоточном приложении будет единственный экземпляр некоторого класса, и предоставляющий глобальную точку доступа к этому экземпляру.

Сцена – многократно используемый шаблон. В сцена может быть чем угодно, начиная от персонажа или оружия и заканчивая интерфейсом, домом или отдельным уровнем. В Godot сцены очень гибки и одновременно выполняют роль шаблонов и сцен из других игровых движков. Также следует упомянуть что одной из сцен присваивается титул главной, и именно она загружается первой при запуске игры.

Дерево сцены – все сцены игры собираются в дереве сцен, буквально дереве из сцен. И, так как сцены – деревья узлов, дерево сцены также является деревом узлов.

Сигнал – это шаблон «Наблюдателя» в исполнении движка Godot.

Шаблон «Наблюдателя» – это поведенческий шаблон проектирования. Также известен как «подчинённые». Реализует у класса механизм, который позволяет объекту этого класса получать оповещения об изменении состояния других объектов и тем самым наблюдать за ними.

Шина – в программирование, класс "Издатель" из паттерна "Издатель-подписчик". При возникновении события в «издателе», вызываются методы «подписчика(ов)» на это событие.

Ресурс – обобщенное название всех контейнеров данных. Ресурсом может быть изображение, скрипт, анимация, звуковой файл, сцена и т.д. Также Godot позволяет создавать свои шаблоны ресурсов, содержащие в себе другие ресурсы.

Хитбокс – в компьютерных играх тело, которое проверяется на столкновение.

## Этапы разработки видеоигры

Как и любое другое приложение, видеоигра проходит несколько различных этапов в процессе разработки:

1. Концептирование. На этом первом шаге команда придумывает концепцию игры, и проводит начальную проработку игрового дизайна. Главная цель данного этапа – это геймдизайнерская документация, в нашем случае, дизайн-документ. Среди ключевых принципов формирования продуктовой документации стоит отметить: структурированность, защищенность от разночтений, полное описание продукта, регулярную актуализацию.
2. Прототипирование. Важный этап проектирования любой игры – это создание прототипа. То, что хорошо выглядит «на бумаге», совершенно не обязательно будет интересно в реальности. Прототип реализуется для оценки основного игрового процесса, проверки различных гипотез, проведения тестов игровых механик, для проверки ключевых технических моментов. Очень важно на этапе создания прототипа реализовывать только то, что нужно проверить и в сжатые сроки. Прототип должен быть простым в реализации, т.к. после достижения поставленных перед ним целей, он должен быть «выкинут». Серьёзная ошибка начинающих разработчиков – нести временную инфраструктуру и «костыли» реализации кода в основной проект.
3. Вертикальный срез. Цель Вертикального среза – получить минимально возможную полноценную версию игры, включающую в себя полностью реализованный основной игровой процесс. При этом высокое качество проработки обязательно нужно воплотить только для тех игровых элементов, которые существенно влияют на восприятие продукта. При этом все базовые фичи игры присутствуют как минимум в черновом качестве. Реализован минимальный, но достаточный для воплощения полноценного игрового процесса набор контента (один уровень или одна локация).
4. Производство контента. На этом этапе производится достаточное количество контента для первого запуска на внешнюю аудиторию. Реализуются все фичи, запланированные к закрытому бета-тестированию. Это наиболее продолжительный этап, который может занимать, для крупных клиентских проектов год и более. Задействуется наибольшее количество специалистов, которые занимаются производством всего основного наполнения игры. Художники создают все графические ресурсы, геймдизайнеры настраивают баланс и заполняют конфиги, программисты реализуют и полируют все фичи.
5. Закрытое бета тестирование. На этапе закрытого бета тестирования продукт впервые демонстрируется достаточно широкой публике, хотя и лояльной продукту или компании. Среди наиболее важных задач на этом этапе выступают: поиск и исправление гейм-дизайнерских ошибок, проблем игровой логики и устранение критических багов. На этом этапе в игре присутствуют уже все ключевые фичи, создано достаточно контента для полноценной игры продолжительное время, настроены сбор и анализ статистики. Тестирование идет по тест-плану, проводятся стресс-тесты уже с привлечением реальных игроков.
6. Открытое бета тестирование. На этом этапе продолжается тестирование игры, но уже на широкой аудитории. Происходит заморозка разработки фич, программисты перестают реализовывать что-то новое, а полностью переключаются на модификацию и улучшение имеющегося функционала.
7. Релиз. На этом этапе должно быть полностью отлажено оперирование продукта (техническая поддержка, работа с сообществом), соблюдаются маркетинговые и финансовые планы, ведутся работы по улучшению финансовых показателей, активно отрабатываются каналы по привлечению трафика. Команда разработки на этом этапе занимается исправлением технических багов, выявляемых в процессе эксплуатации и оптимизацией продукта. Геймдизайнеры занимаются тонкой настройкой геймплея под реальную ситуацию в игровом мире (особенно актуально для ММО проектов). Также реализует различные внутриигровые функции, поддерживающие новые монетизационные схемы. Также идет разработка и интеграция в продукт нового контента, поддерживающего интерес игроков.

Соответственно, самым первым официально документируемым этапом разработки является создание дизайн-документа. Цель дизайн-документа заключается в том, чтобы однозначно описать коммерческие аспекты игры, целевую аудиторию, игровой процесс, графику, дизайн уровней, историю (сюжет), персонажей, пользовательский интерфейс, средства и т. д. Для осуществления поставленной цели, каждое требование к разработке определенной части игры должно быть достаточно подробно описано для соответствующих разработчиков (дизайнеров, программистов и т. д.). Документ намеренно разделен таким образом, чтобы разработчики игры могли поддерживать отдельные его части.

Дизайн-документ – это план работы от начала и до конца проекта. А это значит, что все основные задачи должны в нем находиться, и не только задачи, а и приблизительные методы решения. В частности:

1. Схема игры. Что должен делать игрок, какова конечная цель, что мешает ее достижению.
2. Интерфейс. Подробно описанная функциональная часть (что можно делать, каким образом – меню, мышь, горячие клавиши, кнопки и т.д.).
3. Игровая механика. Как устроен игровой мир, какие характеристики есть у его объектов, формулы движения, боя и всего остального, ролевая система, физика – по вкусу.
4. Программные механизмы и алгоритмы. Какими характеристиками будут обладать графический движок, ИИ, сетевой код, интерфейс, редактор карт, звук.
5. Графика. Сколько и каких вам понадобится моделей, анимаций, двумерной графики, роликов, обоев (да, и их тоже стоит запланировать заранее). Здесь крайне желательны (может, лучше сказать – необходимы) хоть какие-то наброски, концепт арты, по которым можно почувствовать визуальный стиль игры.
6. Звуки и музыка. Темы, вид и способ отображения звуков, набор звуковых эффектов.
7. Сюжет. Общая сюжетная канва, план кампаний, основные задания и т. п. – в зависимости от жанра. Каждая из предполагаемых карт должна быть запланирована здесь.
8. Игровой мир. Основные персонажи / монстры / виды войск с параметрами и примерным расположением / способом добычи и производства.
9. Сотрудники, зарплаты, сроки и план работы.

## Анализ предметной области

Перед созданием видеоигры необходимо ответить на ряд важных вопросов. Данные ответы предоставят достаточно полную информацию для написания дизайн-документа, что, соответственно, значительно ускорит и упростит процесс разработки.

Определение целевой аудитории. Как самый большой сегмент геймеров, целевой аудиторией были выбраны мужчины в возрасте от 16 до 24 лет, владеющие небольшим количеством свободного времени.

Далее предстоял выбор платформы. Ввиду материальных ограничений, разработка консольных игр не является доступным вариантом, соответственно остается мобильный и ПК гейминг. Исходя из статистики, хоть мобильный гейминг и растет, спрос на ПК гейминг все еще сохраняет свои позиции. Однако, разработка на ПК является более простым и быстрым вариантом ввиду гораздо большего разброса производительности, разрешения экрана и объема памяти на мобильных устройствах. Также, в России, ПК гейминг все еще является приоритетным. Исходя из этих факторов, а также учитывая личные предпочтения и доступность людей для будущего закрытого бета тестирования, платформой разработки был выбран ПК.

Далее необходимо было определить продолжительность игрового процесса. Это является настолько же важным параметром, как и платформа, ввиду того что разные целевые аудитории предпочитают разную продолжительность игры. В целях ускорения тестирования было решено остановиться на длительности игры в 5 минут. Такая продолжительность игры позволяет сократить время, необходимое на тестирование баланса игры, а также позволяет войти в нишу игр, доступных в короткий период времени, например обеденный перерыв или 15-ти минутная разгрузка в процессе выполнения домашнего задания.

Определение игрового процесса. Игровой процесс должен быть динамичным и полноценным, соответственно стратегические игры и игры жанра Idle не подходят. Пазлы также не являются хорошим вариантом, ввиду необходимости расслаблять человека, а не напрягать. Мой выбор пал на жанр шутеров (стрелялок). Данный жанр являлся популярным на протяжении всей истории видеоигр и все еще не начал терять популярности. Для поддержания вовлеченности игроков дополнительным жанром был выбран rogue-lite. Суть этого жанра заключается в случайных улучшениях, получаемых во время каждой отдельной игровой сессии, также называемой «забегом», улучшением характеристик между «забегами» и процедурной генерации врагов и локаций. Основной игровой цикл был выбран следующий: Игрой заходит в игру, выбирает персонажа и оружие, после чего на протяжении 5 минут старается пережить нападение постоянно увеличивающихся волн врагов, параллельно получая опыт и улучшения. В конце он получает валюту для улучшений между «забегами», тратит эту валюту и повторяет цикл пока не закончится время или игра не наскучит ему.

Определение графики. Выбор графики был совершен исключительно исходя из личных умений ввиду того, что в наше время больше ценится стиль, а не фотореализм в графике. Создание графики также должно было занимать минимальное количество времени при сохранении приемлемых результатов. Соответственно, идеальным вариантом стала графика в стиле пиксель-арт и разрешением 32 на 32 пикселя. При необходимости допускается возможность увеличения размера холста или размера самих пикселей.

Определения дизайна уровня. Дизайн уровней должен быть простым и не отвлекать игрока от геймплея. Создание атмосферы планируется за счет использования визуальных шейдеров и эффектов в целях экономии времени на создание ассетов уровней. Оптимальным вариантом в таком случае являются подземные локации, где большая часть уровня будет скрыта темнотой.

Определение истории (Сюжета). Исходя из выбора подземной локации и механики получения улучшений, главный герой должен обладать причиной быть под землей и навыками для модификации своего оружия. Таким образом была выбрана роль инженера-шахтера, решено посадить главного героя внутрь робота, а врагов сделать инопланетными подземными существа ввиду отсутствия подземных хищников на нашей планете. Также, исходя из того, что игрок остается один против множества врагов, было решено, что по сюжету, он упал в пещеру уровнем ниже и ждет подмоги. Приходом подмоги, по совместительству, можно объяснить ограничение по времени.

Определение персонажей. Персонажи в игре должны соответствовать работе инженера-шахтера. Соответственно, они должны быть изобретательными, старательными, возможно грубые или недовольные условиями труда.

Определение пользовательского интерфейса. Пользовательский интерфейс должен четко отражать нужную информацию, но при этом оставлять как можно больше пространства на экране не занятым. Оптимальным размещением стали 3 шкалы в левом верхнем углу и одна шкала в центре снизу.

# Создание визуальной концепции интерфейса и схемы работы игры.

## Создание схемы переходов между меню и их наполнения

Игра должна состоять из семи меню и одного интерфейса. Первое меню (Или так называемое «Главное меню») отображается первым после запуска игры и из него совершается переход во все остальные меню.

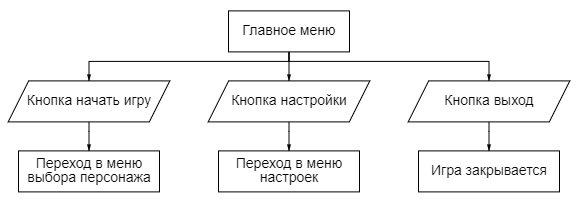


Рисунок . – Схема главного меню

Вторым меню является меню настроек. Оно состоит из трех вкладок: настроек видео, аудио и управления. Кнопки сохранения, отмены и сброса до стандартных настроек присутствуют во всех трех вкладках. Меню настроек может вызываться сразу из двух мест: из главного меню и из меню паузы во время игры. При закрытии открывается то меню, из которого открывалось меню настроек.

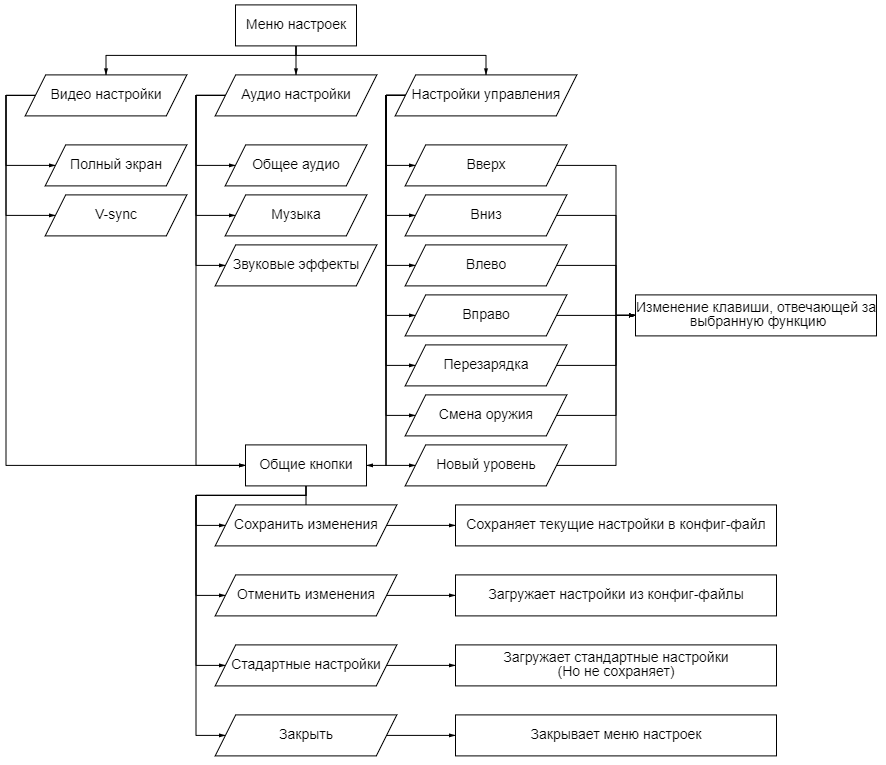


Рисунок . – Схема меню настроек

Третьим меню является меню выбора персонажа. Здесь игроку предоставлены 3 списка, первый список позволяет выбрать своего персонажа, второй оружие, расходующее патроны, третий оружие, расходующее топливо. При выборе персонажа, его характеристики, изображение и дрон-помощник появляются в правой части экрана. Также отсюда игрок может перейти обратно в главное меню, в меню покупки постоянных улучшений или начать игру.

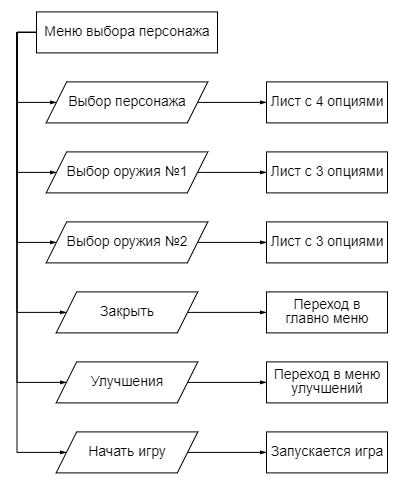


Рисунок . – Схема меню выбора персонажа

Четвертым меню является меню покупки улучшений. В левой части экрана находятся 6 кнопок, рядом с которыми написан текущий уровень характеристики, максимальный уровень характеристики и цена улучшения. При нажатии на кнопку уровень повышается, и валюта вычитается из хранилища игрока. В правой верхней части показывается оставшееся число валюты. Игрок может сбросить все уровни улучшений до 0 и вернуть себе потраченную валюту или сохранить изменения и вернуться в меню выбора персонажа.

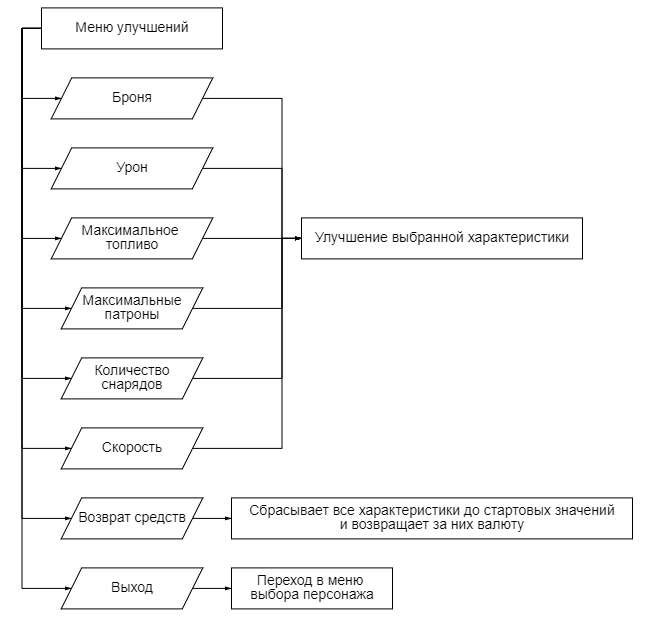


Рисунок . – Меню покупки улучшений

Следующим по логической цепочке идет игровой интерфейс. Во время игры вся информация отображается по краям экрана, с целью не мешать видимости игрока. Слева в несколько строк идут показатели: здоровья игрока, оставшееся количество патронов, оставшееся количество топлива. Сверху в центре экрана находится секундомер, ведущий счет времени, прошедшего с начала игры. Снизу в центре экрана находится полоска опыта и, при наличии достаточного количества опыта, уведомление о возможности повысить уровень. Также, вокруг персонажа периодически появляются стрелки, указывающие на местоположение ящиков с припасами и залежей руды (В схеме именуемые «Точки интереса»).

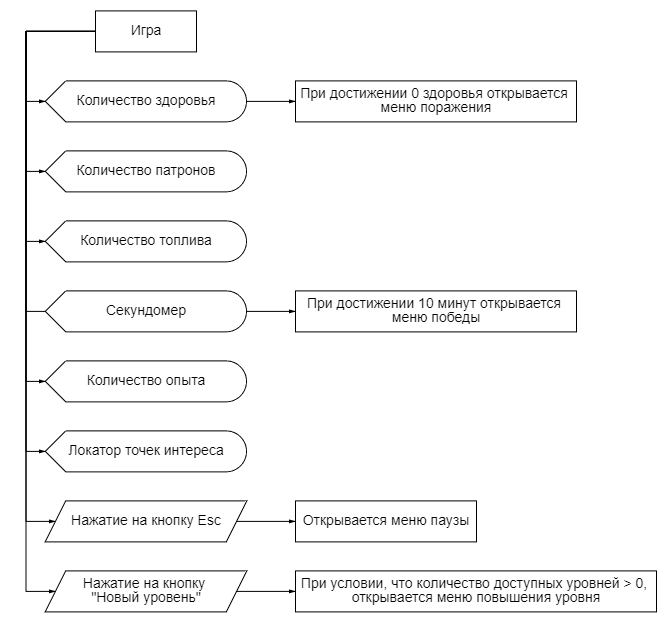


Рисунок . – Схема игрового интерфейса

Пятым меню является меню паузы. Из меню паузы игрок может вернуться к игре, зайти в меню настроек, выйти в главное меню или полностью закрыть игру.

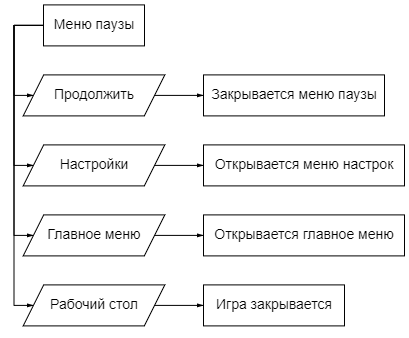


Рисунок . – Схема меню паузы

Шестым является меню повышения уровня. Открыть его можно в том случае, когда игрок набрал достаточно опыта. При открытии игроку даются на выбор до трех разных улучшений, в зависимости от того, сколько улучшений осталось в массиве.

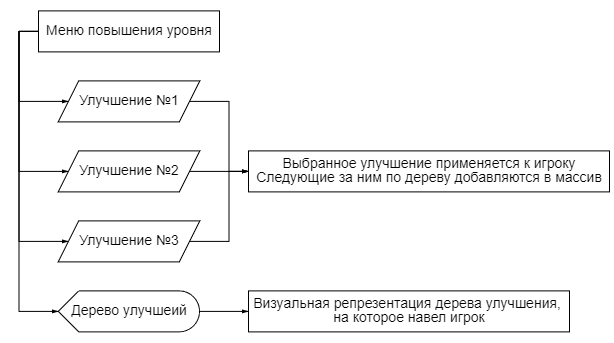


Рисунок . – Схема меню повышения уровня

При наведении на улучшение, в правой половине экрана появляется дерево улучшений, состоящее из одного снизу, соединенного с двумя выше него, слева и справа и еще одного сверху, соединенного с двумя ниже.

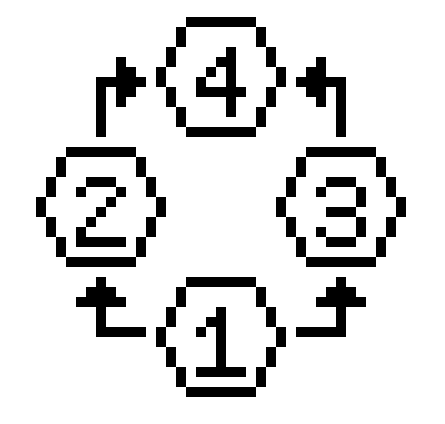


Рисунок . – Пустое дерево улучшений

Последним, седьмым меню является меню победы/поражения. Оно показывается всякий раз, когда выполняется одно из условий. Если здоровье игрока упало до нуля, тогда это меню появляется с надписью «You lost». Если же игроку удалось дожить до десяти минут, это меню появляется с надписью «You won!».

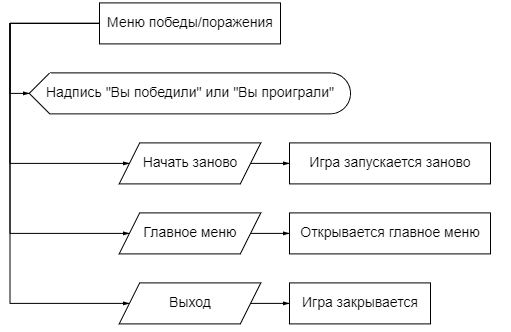


Рисунок . – Схема меню победы/поражения

Находясь в этом меню, игрок может начать заново, пропуская выбор персонажа, выйти в главное меню или же закрыть игру.

Весь интерфейс и меню создается с использованием нарисованных вручную текстур в стиле пиксель-арт и бесплатного шрифта под названием «Pixeloid-sans».

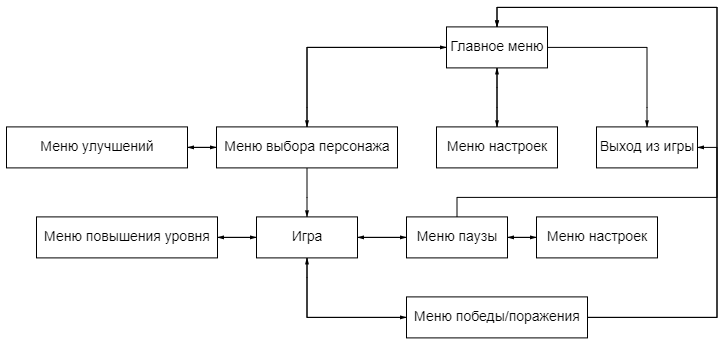


Рисунок . – Схема переходов между меню

После завершения создания отдельных схем меню, была создана полная схема переходов между меню. Две копии меню настроек являются идентичными и разделены в схеме с целью избежать неправильной интерпретации.

## Создание макета интерфейсов

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Книга. В алфавитном порядке фамилия авторов
2. Журнал.
3. Статья.