МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТЮМЕНСКИЙ индустриальный университет»

Институт сервиса и отраслевого управления

Кафедра Бизнес-информатики и математики

КУРСОВОЙ ПРОЕКТ

Дисциплина: «Интеллектуальные системы и их проектирование»

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

Выполнил:

Студент 4 курса группы РИСб-20-1

направления 45.03.04 «Интеллектуальные

системы в гуманитарной сфере»

Никитин М.Д.

Руководитель:

Доцент, канд. пед. наук

Спирин И.С.

Тюмень, 2024 г.

Содержание

Введение 3

1 Теоретические аспекты разработки игр 4

1.1 Основные понятия 4

1.2 Этапы разработки видеоигры 7

1.3 11

Заключение 12

Используемые источники 13

# 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. 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. В данной работе будет рассмотрена разработка видеоигры на движке Godot, включающая в себя элементы искусственного интеллекта.

Целью – изучение процесса разработки видеоигр на движке Godot и создание прототипа игры, отражающего полученные знания и навыки.

Объект – игра, созданная на движке Godot

Предмет – процесс разработки игры

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать возможности и ограничения движка Godot
2. Изучить процесс планирования и разработки игр
3. Создать дизайн-документ
4. Подобрать набор инструментов для создания ассетов
5. Создать ассеты
6. Разработать альфа-версию игры

# Теоретические аспекты разработки игр

## 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.

## 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 отладку и тюнинг имеющихся фичей.
7. Релиз. На этом этапе должно быть полностью отлажено оперирование продукта (техническая поддержка, работа с комьюнити), 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 фичи, поддерживающие новые монетизационные схемы. И конечно 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.

# Аналитический аспект разработки игры

## Выбор ПО, используемого в разработке

Самым главным средством разработки является движок, соответственно, с него выбор и начался. Самыми крупным на данный момент являются Unity, Unreal Engine и Godot.

Unity – кроссплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Плюсы Unity:

* самое большое сообщество из трех вариантов;
* встроенная реклама и аналитика;
* возможность разработки на множество платформ;
* бесплатная версия для игр с доходом меньше ста тысяч долларов;
* самый большой выбор готовых ассетов;
* самый большой выбор обучающих материалов;
* язык программирования C#;
* большие возможности самого движка.

Минусы Unity:

* не удобная структура и интерфейс;
* высокий вес файлов на выходе;
* ошибки, не исправленные на протяжении многих версий;
* несмотря на количество обучающего материала и ассетов, очень сложный для новичка движок;
* 3D графика приемлема, но уступает Unreal Engine.

Unreal Engine – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games. Первой игрой на этом движке был шутер от первого лица Unreal, выпущенный в 1998 году. Хотя движок первоначально был предназначен для разработки шутеров от первого лица, его последующие версии успешно применялись в играх самых различных жанров, в том числе стелс-играх и файтингах. Плюсы Unreal Engine:

* лучшая графика из трех вариантов;
* удобные инструменты моделирования уровней;
* большое сообщество и количество обучающего материала;
* встроенный визуальный скриптинг;
* бесплатный;
* готовых ассетов меньше количественно, но лучше по качеству.

Минусы Unreal Engine:

* большой вес и программы и финальной игры;
* очень высокий порог входа;
* интерфейс хоть и удобный, но очень объемный;
* плохая производительность.

Godot – открытый кроссплатформенный 2D и 3D игровой движок под лицензией MIT, который разрабатывается сообществом Godot Engine Community. До публичного релиза в виде открытого ПО движок использовался внутри некоторых компаний Латинской Америки. Среда разработки запускается на Android, HTML5, Linux, macOS, Windows, BSD и Haiku и может экспортировать игровые проекты на ПК, консоли, мобильные и веб-платформы. Плюсы Godot:

* лучшая документация из трех вариантов;
* бесплатный;
* размер сообщества уступает только двум другим движкам;
* открытый исходный код, быстрое исправление багов;
* работает на всех платформах и позволяет экспортировать почти на все платформы;
* производительный, игры имеют малый вес;
* удобная организация структуры файлов;
* поддержка GDScript, неполная поддержка C#, C++;
* возможность поиска документации внутри приложения.

Минусы Godot:

* относительно молодой движок, существует необходимость устанавливать дополнительные модули;
* плохая производительность с большим количеством объектов;
* небольшое количество обучающих материалов, частая смена версий и последующее устаревание старых материалов;
* малое количество успешных игр и общая популярность движка.

Unreal Engine данному проекту не подходит ввиду объема и сроков, данный движок больше подходит для крупных проектов, создаваемых командой профессионалов. Unity же хоть и является одним из самых популярных движков, последние несколько лет страдает от плохого руководства и постепенно теряет популярность ввиду этого. Также, мой личный опыт работы с Unity оставил впечатление гораздо хуже, чем Godot. Исходя из всего этого, выбор движка пал на Godot.

Выбор языка программирования соответственно, был сделан при выборе движка, ведь Godot полностью построен вокруг использования языка GDScript и использование других языков хоть и увеличит производительность, но значительно увеличит время разработки.

Выбор программы для рисования ассетов имеет значение только в трех критериях: удобство, стоимость и лицензия на использование. Ввиду предыдущего опыта использования Aseprite, их лицензии на свободное использование созданных ассетов в разработке игр, а также низкой цены (435 рублей на момент написания курсовой), был выбран именно он. Для создания звуков было решено использовать единственный удобный полностью бесплатный редактор ChipTone, распространяемый на itch.io по лицензии, позволяющей выполнять любые манипуляции не запрашивая разрешения автора.

## Выбор прочего ПО, используемого в разработке

# Заключение

В ходе выполнения данной работы был закреплен материал, изученный в процессе обучения и выполнены все поставленные ранее задачи:

* Рассмотрены основные понятия веб-разработки
* Рассмотрены этапы создания веб-сайтов
* Рассмотрены технологии разработки веб-страниц
* Оформлено техническое задание на разработку веб-сайта
* Разработан дизайн веб-сайта
* Разработан веб-сайт

# Используемые источники

1. Портрет русского геймера 2020