

**муниципальное бюджетное общеобразовательное учреждение
города Ростова-на-Дону «Школа № 6
имени Героя Советского Союза Самохина Н.Е.»**

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

на тему:

«Программирование компьютерной игры на Unity»

Выполнил учащийся 11 класса:
Гайдин Даниил Михайлович
Научный руководитель:
учитель высшей категории
Казарова Лаура Варужановна
Допуск к защите: _____

г. Ростов-на-Дону 2024 год

Оглавление

Введение	3
Глава 1.....	6
Предпродакшен	6
Шаги разработки игры.....	6
Глава 2.....	8
Изучение требований и концепции игры.....	8
Проектирование архитектуры программного обеспечен	8
Разработка игровых механик	8
Реализация и тестирование функциональности	9
Сотрудничество с другими специалистами.....	10
Глава 3.....	12
Шаги разработки моей игры	12
Планирование	12
Разработка игровых ресурсов	12
Написание кода	12
Где найти игру?.....	12
Заключение.....	13
Список источников	14

Введение

Игра — это форма развлечения, обучения, активности или досуга, погружающая участника в виртуальный мир, основанная на участии игрока, который следует определенным правилам и стремится достичь конкретной цели. Игра может быть одиночной, соревновательной или кооперативной.

Проблема: Создание компьютерной игры на Unity - задача, которая может вызывать не мало трудностей. Необходимо учесть аспекты наподобие, программирования и тестирования игровых механик. Кроме того, для создания привлекательной игры необходимо уделять должное внимание созданию увлекательного и сбалансированного геймплея. Несомненно, разработка компьютерной игры — это творческий и сложный процесс, который требует множество умений и знаний.

Объект проекта включает в себя разработку программного обеспечения, которое обеспечивает функционирование и взаимодействие всех элементов игрового мира. Этот процесс включает в себя следующие ключевые шаги и задачи:

1. Проектирование архитектуры: определение структуры кода, компонентов и взаимодействий между ними для реализации игровых механик и функциональности.
2. Выбор языка программирования: определение языка программирования, который наилучшим образом соответствует потребностям проекта и целям разработки.
3. Написание кода игровых механик: разработка кода для реализации различных игровых механик, таких как управление персонажем, искусственный интеллект, физика, коллизии и т. д.
4. Реализация пользовательского интерфейса: создание кода для отображения информации на экране, включая меню, интерфейс пользователя, текстовые элементы и другие визуальные компоненты.
5. Интеграция графики и аудио: встраивание графических и звуковых ресурсов в игровой движок с использованием кода для их загрузки, воспроизведения и управления.
6. Тестирование и отладка: проведение тестирования кода для обнаружения и устранения ошибок, а также проверка на соответствие требованиям и ожиданиям игры.
7. Оптимизация и производительность: оптимизация кода для повышения производительности и эффективности работы игры на различных устройствах и платформах.
8. Документация и комментирование: создание документации и комментариев к коду для облегчения понимания и поддержки кодовой базы проекта.
9. Поддержка и обновления: после выпуска игры продолжение поддержки кода, выпуск обновлений и исправлений ошибок, а также добавление нового контента и функциональности.

Объект проекта является ключевым компонентом разработки игрового программного обеспечения и важным этапом в создании увлекательного и интересного игрового опыта для пользователей.

Целью работы по созданию компьютерной игры на Unity является разработка увлекательной и качественной игровой механики, приятной визуальной и звуковой составляющей, а также интересного и динамичного

игрового сюжета. В результате работы планируется создание игры, которая будет способна заинтересовать и увлечь игроков, предлагая им уникальные игровые возможности и вызывая положительные эмоции. Работа будет выполняться с использованием современных технологий и инструментов разработки, а также внимательного анализа и исправления ошибок для достижения наилучшего качества игры.

Предметом проекта для проекта "Программирование компьютерной игры на Unity" является разработка полноценной компьютерной игры с использованием игрового движка Unity. Этот проект включает в себя создание игровых механик, персонажей, уровней, а также реализацию графики, звуков и других аспектов игровой среды. Основной целью проекта является создание увлекательной и качественной игры, способной привлечь внимание игроков и обеспечить им приятный игровой опыт.

Задачи исследования:

- Разработка игровых механик: определение основных правил и механик игры, таких как система управления, система взаимодействия игрока с объектами, прогрессия и уровни сложности.
- Имплементация игровой логики: программирование игровой логики, включающей в себя обработку пользовательского ввода, управление объектами в игре, алгоритмы искусственного интеллекта и т.д.
- Тестирование и отладка: проверка игры на наличие ошибок, исправление багов и тестирование на соответствие требованиям и ожиданиям пользователей.
- Оптимизация производительности: улучшение производительности игры, так чтобы она работала с минимальными задержками и лагами.

Методы, предлагаемые для создания проекта в области написания кода для игры, включают в себя следующий набор:

1. Определение концепции игры: первый шаг в создании проекта - определение концепции игры. Это включает в себя разработку идеи игры, определение её жанра, основных механик, целевой аудитории и других ключевых аспектов.
2. Планирование: затем необходимо спланировать проект. Это включает в себя создание плана разработки, определение этапов разработки, распределение задач между участниками команды (если проект командный) и разработку графика работы.
3. Выбор технологий: на этом этапе определяются технологии и инструменты, которые будут использоваться при создании проекта. Это может быть игровой движок (например, Unity, Unreal Engine), язык программирования, библиотеки и другие инструменты разработки.
4. Настройка среды разработки: после выбора технологий необходимо настроить среду разработки. Это включает установку необходимого программного обеспечения, настройку проекта в выбранном игровом движке или среде программирования, создание проектной структуры и настройку рабочего пространства.
5. Проектирование архитектуры: следующим шагом является проектирование архитектуры программного обеспечения. Это включает определение структуры проекта, разделение функциональности на модули, проектирование классов и интерфейсов, определение взаимодействия между компонентами и другие архитектурные решения.

6. Написание кода: после проектирования архитектуры приступают к написанию кода игры. Это включает разработку игровых механик, создание игровых объектов, обработку пользовательского ввода, управление анимациями, обработку столкновений и другие аспекты разработки.
7. Тестирование и отладка: после написания кода проект подвергается тестированию и отладке. Это включает проверку работоспособности игры, обнаружение и исправление ошибок, тестирование на соответствие требованиям и корректировку дизайна.
8. Оптимизация: после завершения разработки проекта производится его оптимизация. Это включает улучшение производительности, оптимизацию использования ресурсов, устранение узких мест и другие меры для повышения качества и эффективности проекта.
9. Документация: наконец, создается документация проекта, включающая описание функциональности, инструкции по использованию, руководства для разработчиков и другие документы, необходимые для поддержки и дальнейшего развития проекта.

Это общий набор методов, которые могут быть использованы при создании проекта в области написания кода для игры. Конкретные методы могут различаться в зависимости от требований проекта, его масштаба и особенностей разработки.

Практическая значимость проекта в области написания кода для игры проявляется в нескольких аспектах:

1. Развлекательный аспект: Игры являются источником развлечения для миллионов людей по всему миру. Создание качественного игрового программного обеспечения позволяет предложить пользователям интересный и увлекательный игровой опыт, который помогает им отдохнуть, развлечься и провести время с удовольствием.
2. Образовательный аспект: Разработка игр требует разнообразных навыков и знаний, включая программирование, дизайн, анимацию, звуковое проектирование и другие. Процесс создания игрового программного обеспечения может стать практическим опытом для разработчиков, помогая им развивать свои навыки и получать новые знания.
3. Экономическая значимость: Игровая индустрия является одной из самых быстрорастущих и прибыльных отраслей в мире. Создание успешной игры может привести к значительным финансовым выгодам для разработчиков, издателей и других участников индустрии.
4. Социальное влияние: Игры имеют значительное социальное влияние, способствуя формированию культуры, влияя на мировоззрение и общественное сознание. Создание качественных игр с положительными ценностями и сообщениями может оказать положительное влияние на общество.
5. Исследовательский аспект: Разработка игр позволяет исследовать новые технологии, методы и подходы к программированию, анимации, искусственному интеллекту и другим областям. Это способствует развитию науки и техники в целом.

Глава 1

Предпродакшен

Создание игры с точки зрения программирования требует учета нескольких факторов и применения различных технологий.

1. Выбор платформы: сначала нужно определиться, для какой платформы (компьютер, консоль, мобильное устройство) будет разрабатываться игра.
2. Выбор игрового движка: игровой движок — это программное обеспечение, которое обеспечивает основные функциональные возможности игры, такие как графика, физика, звук и управление. В зависимости от выбранного языка программирования, вы можете использовать существующие игровые движки (например, Unity, Unreal Engine).
3. Язык программирования: в зависимости от выбранного движка нужно выбрать язык программирования, но некоторые движки работают только с одним языком программирования.
4. Реализация графики: для создания визуальной составляющей игры потребуется работа с графическими библиотеками и игровыми движками, которые предлагают набор инструментов для работы с 2D и 3D графикой. Здесь могут использоваться спрайты, текстуры, эффекты освещения, анимации и другие элементы.
5. Реализация физики: при создании игры часто требуется взаимодействие объектов виртуального мира с учетом законов физики. Для этого можно использовать встроенные инструменты движка.
6. Реализация игровой логики: это часть программы, которая определяет правила и механику игры. Здесь мы будем создавать и управлять объектами, обрабатывать ввод пользователя (например, с клавиатуры или геймпада) и определять условия победы или поражения.
7. Работа со звуком: чтобы создать атмосферу и добавить эффектов в игру, нужно работать с инструментами, которые позволяют воспроизводить звуки, музыку и звуковые эффекты в соответствии с действиями игрока или состоянием игры.
8. Тестирование и отладка: важная часть процесса разработки игры - тестирование игрового процесса, выявление и исправление ошибок и недочетов. Это может включать в себя ручное тестирование, автоматическое тестирование или бета-тестирование среди пользователей.

Таким образом, создание игры с точки зрения программирования требует понимания всех аспектов разработки игр - от выбора платформы и языка программирования до оптимизации и развертывания игры. Это всеобъемлющий процесс, который требует знания и опыта в программировании и разработке игр.

Шаги разработки игры

1. Планирование и дизайн: на этом этапе определяются основные элементы игры, такие как сюжет, уровни, персонажи и интерфейс. Также проводится дизайн игровых механик и систем игры.

2. Разработка игровых ресурсов: это включает создание графических и звуковых ресурсов игры, таких как спрайты, текстуры, модели персонажей, анимации, музыка и звуковые эффекты.
3. Написание кода: на этом этапе программисты приступают к написанию кода игры на выбранном языке программирования. Они создают классы и функции для управления объектами, обработки ввода, реализации игровых правил и т.д.
4. Интеграция ресурсов и кода: после написания кода игры, программисты интегрируют созданные ресурсы в игру, связывают их с соответствующими объектами и реализуют их функциональность.
5. Тестирование и отладка: на этом этапе разработчики проводят тестирование игры, чтобы обнаружить и исправить ошибки, недочеты и непредвиденные ситуации. Они проверяют игровую механику, графику, физику, звук и другие аспекты игры.
6. Полировка и оптимизация: после исправления ошибок и недочетов разработчики переходят к полировке игры. Они улучшают графику, звук, игровую механику, добавляют эффекты и другие детали, чтобы сделать игру более привлекательной для игроков. Они также оптимизируют код и ресурсы игры для повышения производительности.
7. Развертывание: после завершения разработки и тестирования игры она готова к развертыванию на целевых платформах. Разработчики могут создать установочные файлы или загрузочные пакеты для целевых платформ, чтобы игроки могли скачивать и устанавливать игру.
8. Монетизация: разработчики также могут выбрать модель монетизации игры, например, платная загрузка, внутриигровые покупки, реклама или подписка. Это позволяет разработчикам зарабатывать деньги на своей игре.
9. Обновления и поддержка: после выпуска игры разработчики могут продолжать работать над игрой, выпуская обновления и исправляя ошибки, а также поддерживая игру и взаимодействуя с сообществом игроков.

Все эти шаги требуют технических знаний в области программирования, графики и звука, а также творческий подход и умение работать в команде. Создание игры — это сложный и многогранный процесс, но с правильным подходом и усилиями можно создать увлекательное и качественное игровое приложение.

Глава 2

Изучение требований и концепции игры

Для программиста, изучение требований и концепции игры является важным этапом перед началом разработки. Вот несколько шагов, которые следует выполнить в процессе изучения требований и концепции игры:

- 1. Изучение документации.** Разработчики игры должны внимательно изучить документацию, предоставленную дизайнерами игры, чтобы понять ее основные механики, функциональные требования и цель.
- 2. Анализ конкурентов.** Изучение аналогичных игр на рынке может помочь определить особенности и требования, которые могут быть востребованы у пользователей. Это также позволит избежать повторения ошибок, сделанных другими разработчиками.
- 3. Создание концепции.** На основе изученных требований и анализа конкурентов, программисты могут помочь разработать концепцию игры, которая объединит функциональные возможности и цели проекта.
- 4. Разработка технических требований.** На основе концепции игры, программисты должны разработать технические требования, которые включают в себя описание основных систем, платформы, языки программирования и другие детали, необходимые для разработки игры.
- 5. Разработка прототипа.** Программистам рекомендуется создавать прототип игры, чтобы проверить, насколько эффективно работает разработанная концепция и требования. Прототип может быть базовым вариантом игры, созданным для демонстрации основных механик или структуры игры.
- 6. Контроль и тестирование.** Во время разработки игры программисты должны постоянно проверять соответствие требованиям и концепции, а также проводить тестирование для обнаружения и исправления ошибок и неполадок.

Проектирование архитектуры программного обеспечен

- 1. Определение требований:** понимание функциональных и нефункциональных требований к системе.
- 2. Разделение на подсистемы:** разбивка системы на подсистемы (модули, компоненты) для упрощения процесса разработки и управления сложностью.
- 3. Определение интерфейсов:** разработка четких интерфейсов между компонентами системы для обеспечения их взаимодействия; обеспечение удобства использования и обратной совместимости интерфейсов.

Разработка игровых механик

Разработка игровых механик — это процесс создания внутриигровых правил, интерактивных элементов и систем, которые определяют способы взаимодействия игрока с игрой. Хорошие игровые механики формируют интересный и увлекательный геймплей, способствуют вовлеченности игрока и создают уникальный опыт. Вот некоторые ключевые шаги и принципы, которые могут помочь в разработке игровых механик:

- 1. Определение целей игры:** чётко сформулируйте цели, которые игроки должны достигнуть в игре. Это может быть завершение уровня, победа в сражении, достижение определенного результата и т.д.
- 2. Исследование жанра:** изучите игры в том же жанре, что и ваша игра. Определите успешные механики и элементы, которые могли бы быть приспособлены к вашему проекту.
- 3. Определение основных механик:** определите основные механики, которые будут определять геймплей. Это могут быть элементы управления, взаимодействие с окружением, системы боя, головоломки и т.д.
- 4. Балансировка механик:** соблюдайте баланс между различными механиками, чтобы избежать перекосов в геймплеи. Гармоничное взаимодействие различных механик создает увлекательный опыт.
- 5. Интуитивность управления:** обеспечьте интуитивность управления. Игроки должны легко освоить основные механики и элементы управления.
- 6. Эмоциональная привлекательность:** работайте над тем, чтобы механики вызвали эмоциональные реакции у игроков. Это может включать в себя чувство достижения, волнение, удивление и др.
- 7. Итеративная разработка:** проводите тестирование игровых механик среди целевой аудитории и осуществляйте итеративные изменения на основе обратной связи.
- 8. Сценарии использования:** разработайте сценарии использования, определяющие, как игроки будут взаимодействовать с механиками в различных ситуациях.
- 9. Тематика и стиль:** учтите тематику и стиль вашей игры при разработке механик. Они должны соответствовать общей атмосфере и концепции игры.

Разработка игровых механик — это творческий и исследовательский процесс. Важно привлекать целевую аудиторию и внимательно слушать их отзывы, чтобы создать интересный и увлекательный игровой опыт.

Реализация и тестирование функциональности

Реализация и тестирование функциональности — это важные этапы в разработке программного обеспечения. В этом процессе команда разработчиков преобразует проектные спецификации и требования в реальный код, а затем тестирует его, чтобы убедиться в правильности и надежности функциональности. Вот общий план действий для реализации и тестирования функциональности:

Реализация функциональности:

- 1. Программирование:** команда разработчиков начинает создавать код в соответствии с проектными спецификациями.
- 2. Контроль версий:** используйте системы управления версиями, такие как Git, чтобы отслеживать изменения в коде и обеспечить совместную работу разработчиков.
- 3. Интеграция кода:** регулярно интегрируйте новый код в общий репозиторий для предотвращения конфликтов и обеспечения единого источника правды.

4. **Тестирование на уровне кода (Unit Testing):** разработчики создают и запускают модульные тесты для проверки правильности отдельных компонентов кода.
5. **Отладка:** выявляйте и устраняйте ошибки, которые могут возникнуть в процессе написания кода.

Тестирование функциональности:

1. **Тестирование компонентов (Component Testing):** проверьте, как работают отдельные компоненты или модули в изоляции от других частей системы.
2. **Интеграционное тестирование (Integration Testing):** проверьте, как взаимодействуют между собой различные компоненты и модули системы.
3. **Системное тестирование (System Testing):** тестируйте систему в целом, чтобы убедиться, что она соответствует общим требованиям и работает согласно функциональным спецификациям.
4. **Приемочное тестирование (Acceptance Testing):** проверьте соответствие продукта конечным требованиям заказчика. Это может включать функциональные, пользовательские и производительные тесты.
5. **Документация тестирования:** создайте документацию тестирования, включающую в себя результаты тестов, шаги воспроизведения ошибок и другую полезную информацию.
6. **Итерационный процесс:** повторяйте процесс реализации и тестирования в случае необходимости, особенно если вы вносите изменения или добавляете новые функции.

Сотрудничество с другими специалистами

Сотрудничество с другими специалистами является ключевым аспектом успешного разработки программного обеспечения. В большинстве проектов участвуют разнообразные специалисты с разными навыками и областями экспертизы. Вот несколько важных принципов и рекомендаций по сотрудничеству в команде:

1. **Обмен идеями и информацией:** активно обменивайтесь идеями, мнениями и информацией с другими членами команды. Это помогает создать общее понимание задач и целей проекта.
2. **Четкое понимание ролей и обязанностей:** убедитесь, что каждый член команды понимает свои роли и обязанности. Это помогает избежать дублирования работы и обеспечивает эффективное распределение задач.
3. **Открытая коммуникация:** поддерживайте открытую и прозрачную коммуникацию внутри команды. Это включает в себя регулярные совещания, обратную связь и открытость к обсуждению проблем.
4. **Работа в инкрементах:** используйте методологии разработки, такие как Agile, чтобы работать над проектом в коротких циклах и регулярно взаимодействовать с другими членами команды.
5. **Уважение к экспертизе:** уважайте уникальные навыки и знания каждого участника команды. Признавайте и цените их экспертизу в своих областях.
6. **Регулярные обзоры кода:** проводите регулярные обзоры кода, чтобы обеспечить соответствие стандартам кодирования и обменяться знаниями между разработчиками.

7. **Совместная работа в режиме реального времени:** используйте средства для совместной работы в режиме реального времени, такие как средства для онлайн-чата и коллаборации, чтобы команда могла быстро реагировать на изменения и проблемы.
8. **Про активное решение конфликтов:** решайте конфликты прозрачно и про активно. Стремитесь к поиску компромиссов и общему пониманию.
9. **Обучение и развитие:** поощряйте обучение и развитие членов команды. Это может включать в себя проведение внутренних семинаров, обучение новым технологиям и обмен опытом.

Сотрудничество и командная работа играют ключевую роль в успехе проекта. Понимание и применение этих принципов способствует эффективному взаимодействию различных специалистов в команде разработки.

Глава 3

Шаги разработки моей игры

Планирование

1. Платформа для игры была выбрана компьютер так, как для этой платформы есть больше общедоступной информации по разработке в интернете.
2. Движок для разработки был выбран Unity потому, что он прост в усвоение, имеет множество информации по работе с ним и у меня уже был опыт работы в нем. Язык программирования был выбран C# та как Unity работает только с ним.
3. Для игры была выбрана пиксельная графика, 2D.

Разработка игровых ресурсов

1. Физика и физические свойства были взяты с уже существующих встроенных в движок.
2. Звуки и музыка были взяты из интерната и настроены в движке.
3. Спрайты персонажей и уровней были созданы моими коллегами.

Написание кода

Написание кода для игрового проекта – это ключевой этап в разработке игрового программного обеспечения. Этот процесс требует не только технических навыков, но и креативного подхода, чтобы превратить концепцию игры в интерактивный и увлекательный игровой опыт.

Основные шаги:

Анализ требований: первым шагом является анализ требований игры. В игре запрограммировали перемещения, прыжок, систему здоровья и переход меж уровнями и скрипт для клавиш.

Написание кода: при написании кода был созданы игровые объекты, системы управления, визуальных и звуковых эффектов, логики игровых механик и других компонентов игры.

Тестирование: после написания кода провести тестирование, чтобы убедиться в правильности работы игры. Это включает в себя тестирование функциональности, производительности, стабильности и совместимости на разных устройствах.

Итерации и улучшения: процесс разработки обычно включает в себя несколько итераций. После тестирования были выявлены ошибки и возможности для улучшения, и код игры пересматривался.

Написание кода для игры – это творческий и технически сложный процесс, который требует внимательного планирования, умения работать в команде и стремления к постоянному улучшению. Эффективная разработка кода игры позволяет создать уникальный и захватывающий игровой опыт для пользователей.

Где найти игру?

Погрузитесь в захватывающий мир моей новой игры! Скачайте её бесплатно и отправьтесь в увлекательное приключение! Вы можете ее скачать на сайте GameJolt по ссылке <https://gamejolt.com/games/Maria/872814>

Заключение

В ходе разработки данного проекта по созданию игры мы с командой сделали значительные успехи в достижении наших целей. Моя команда в лице разработчика, левел-дизайнера и арт-дизайнера продемонстрировала высокий уровень профессионализма, творческое мышление и способность эффективно работать в команде.

Мы успешно реализовали все функциональные требования, предусмотренные спецификацией проекта, создав увлекательную игровую среду с интересными механиками. Наш код отличается чистотой, эффективностью и высоким уровнем производительности, что гарантирует приятный пользовательский опыт.

Кроме того, мы уделили должное внимание тестированию и отладке, что позволило нам выявить и устранить потенциальные проблемы и ошибки еще на ранних этапах разработки. Это позволило нам создать стабильную и надежную игру, которая будет радовать пользователей без сбоев и неполадок.

В целом, мы уверены в качестве нашей работы и убеждены, что наш проект принесет удовлетворение и радость нашим пользователям. Мы постоянно работаем над улучшением нашей игры и готовы предложить вам еще больше захватывающего контента и исправлений багов после её релиза. Ваше участие и обратная связь помогут нам сделать игру еще лучше!

Мы также стремились к тому, чтобы наш проект имел практическую значимость не только для развлечения, но и для обучения. Создание игры позволило нам применить на практике знания и навыки, полученные в процессе обучения, и расширить свой опыт в области разработки программного обеспечения.

Представляя нашу игру общественности, мы надеемся вдохновить других на творческую и инновационную работу в области игровой индустрии. Мы уверены, что успешное завершение проекта послужит мотивацией для нас и других разработчиков, продолжить исследование и создание увлекательных игр.

Наша цель - не только развлечь, но и вдохновить, обучить и поддержать наших пользователей. Мы готовы принять вызовы, учиться на ошибках и совершенствоваться дальше вместе с нашими пользователями. Благодаря вашей поддержке и участию, мы уверены, что наш проект будет иметь долгосрочное положительное воздействие на игровое сообщество.

Мы уверены, что наш проект по созданию игры не только приносит радость и развлечения нашим пользователям, но и имеет широкую практическую значимость. Сочетание высокого уровня профессионализма, креативности и коллективной работы позволило нам достичь успешных результатов в разработке увлекательной игровой среды.

Мы гордимся чистотой и эффективностью нашего кода, обеспечивающего стабильную и плавную работу игры. Наше внимание к тестированию и отладке помогло создать надежное и безупречное игровое приложение.

Более того, мы стремимся вдохновить других разработчиков на творческий подход к созданию игр и обучить новые поколения разработчиков, демонстрируя путь к успешному завершению проекта.

Список источников

1. Сайт DFT статья «С чего начать делать игру? Пошаговая инструкция» <https://dtf.ru/gamedev/849688-s-chego-nachat-delat-igru-poshagovaya-instrukciya> 22.07.2023
2. Сайт Хабр статья «Геймдизайнер с нуля: как начать делать игры без опыта» <https://habr.com/ru/companies/netologyru/articles/420689/> 22.07.2023
3. Сайт Proglib статья «GameDev: с чего начать разработку игр?» <https://proglib.io/p/gamedev-s-chego-nachat-razrabotku-igr-2019-11-28> 24.07.2023
4. Сайт Хабр статья «Диздок-десятистраничник – шаблон по книге Скота Роджерса с примером на игру от Pixonic» <https://habr.com/ru/articles/469971/> 01.08.2023
5. Видео «Организация проекта в Unity. Иерархия папок» <https://www.youtube.com/watch?v=OtrH-Ws1D5M> 09.08.2023
6. Видео «Всё про Tilemaps в Unity 3D» <https://www.youtube.com/watch?v=6-jlx2QVTuY> 10.08.2023
7. Сайт Proglib статья «Создание 2D–игру на Unity: инструкция для новичка» <https://proglib.io/p/sozdaem-2d-igru-na-unity-instrukciya-dlya-novichka-2020-09-01> 15.08.2023
8. Сайт Хабр статья «Про создание платформера на Unity. Часть вторая, звездная» <https://proglib.io/p/sozdaem-2d-igru-na-unity-instrukciya-dlya-novichka-2020-09-01> 16.08.2023
9. Книга П.В.Евдокимова «С# на примерах. Практика, практика и только практика» напечатана 2022 году
10. Книга А.В.Корнилов «Unity. Полное руководство. + Виртуальный диск с примерами, проектами и ассетами.» напечатана 2020 году