

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
«ЦЕНТР ДЛЯ ОДАРЕННЫХ ДЕТЕЙ «ПОИСК»

РЕКОМЕНДОВАНА

педагогическим советом
Протокол №7 от «4» апреля 2025 г.



УТВЕРЖДАЮ

Заведующая филиалом
Т.В. Ларина

Дополнительная общеобразовательная общеразвивающая
программа технической направленности

**«UNITY GAME JAM: СОЗДАНИЕ ИГРЫ В ЖАНРЕ HORROR
SURVIVAL SHOOTER»**

Направление: техническое

Возраст обучающихся: 12-16 лет

Объем программы: 72 часа

Срок освоения: 1 год

Форма обучения: очная

Авторы программы: Журавлев Алексей Борисович, педагог
дополнительного образования ЦЦО «IT-куб»

Михайловск, 2025

ОГЛАВЛЕНИЕ

| | |
|---|----|
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА..... | 3 |
| Направленность программы | 4 |
| Актуальность программы | 4 |
| Новизна | 5 |
| Цель программы..... | 6 |
| Задачи программы | 6 |
| Категория обучающихся | 7 |
| Структура программы | 7 |
| Форма реализации программы | 8 |
| Формы организации деятельности обучающихся..... | 9 |
| Ожидаемые результаты..... | 9 |
| Способы определения результативности | 11 |
| Формы подведения итогов реализации программы..... | 11 |
| УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН КУРСА..... | 12 |
| СОДЕРЖАНИЕ КУРСА | 14 |
| СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ | 28 |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Став социальным феноменом в конце XX века, компьютерные игры прочно вошли в ряд активных занятий, предпочитаемых современными людьми для повседневного времяпровождения, а доступность и мобильность цифровой техники обеспечили играм широкое распространение. Такое положение дел является следствием высокого технического и технологического развития общества в совокупности с необходимостью в удовлетворении индивидами собственных потребностей.

Компьютерная игра – это не просто технический продукт. Она представляет собой уникальное сочетание технологий, искусства, психологии и даже философии.

На первом уровне, игры разрабатываются с использованием сложных технологий. Программирование, графический дизайн, создание звукового сопровождения – все это требует высокой квалификации и глубоких знаний в своих областях. Но за этой «технической оболочкой» скрывается также творческий процесс: игровые разработчики выступают не только как программисты, но и как рассказчики. Каждая игра имеет свою историю, персонажей, уникальный мир, который захватывает воображение игрока. Это требует художественного чутья, умения создавать эмоциональные связи и разрабатывать интересные сюжетные повороты.

Кроме того, разработка игр включает в себя элементы психологии. Понимание мотивации и поведения игрока позволяет создать более увлекательный игровой процесс. Также многие аспекты концепции и дизайна имеют культурные и социальные корни, что делает игры важной частью современного искусства и социокультурной среды.

Таким образом, компьютерная игра представляет собой сложное и многогранное произведение, в которое встроены не только технологии, но и элементы искусства, психологии и культуры. Она оставляет заметный след в жизни каждого игрока и общества в целом.

Направленность программы

Программа носит междисциплинарный характер и позволяет решить задачи развития у учащихся технико-технологических и гуманитарных компетенций.

Актуальность программы

В настоящий момент игровая индустрия продолжает расти и развиваться, являясь одной из самых быстрорастущих рыночных отраслей, предоставляющих множество карьерных возможностей.

Проектирование игры – это обычно командный процесс, который требует взаимодействия сотрудников различных специальностей.

Обучение разработке игр помогает улучшать навыки общения и работы в команде, сочетает в себе элементы креативности и технологий, помогает развивать творческое и критическое мышление, применять инновационные подходы к решению задач, требующих анализа, планирования и решения проблем, что полезно не только в игровой индустрии, но и в других областях.

Во время обучения внимание уделяется таким технологическим навыкам как программирование, создание компьютерной графики, звукозаписи и другим, поэтому специалисты в области разработки игр могут зарабатывать довольно высокие заработные платы и имеют возможность работать в различных компаниях и студиях по всему миру в профессиях, связанных с проектированием, программированием, тестированием и маркетингом игр.

Таким образом, обучение разработке компьютерных игр не только открывает массу возможностей, но и способствует личному и профессиональному развитию, что делает его важным и перспективным направлением.

Новизна

Программа построена на принципах, применяемых при организации мероприятий в формате Game Jam.

Game Jam представляет собой формат мероприятий, в рамках которого разработчики игр, художники, музыканты и другие творческие личности собираются вместе, чтобы создать игру за ограниченный период времени (обычно от 24 до 48 часов). Этот формат представляет собой уникальный подход к разработке игр и имеет ряд отличительных особенностей, таких как:

- **Ограничение во времени.** Ограниченное время способствует быстрому принятию решений и генерированию идей. Участники вынуждены работать быстро, что приводит к инновационным решениям и нетипичным игровым механикам. Это создает возможность для экспериментов, которые невозможно реализовать в традиционных условиях разработки.

- **Командное взаимодействие.** В ходе Game Jam формируются многопрофильные команды, что позволяет участникам обмениваться опытом и знаниями. Это взаимодействие часто приводит к неожиданным и оригинальным результатам, когда разные специализации объединяются для достижения общей цели.

- **Фокус на прототипировании.** Game Jam поощряет создание прототипов, а не завершенных продуктов. Участники могут сосредоточиться на основных механиках игры и концепции, что снижает давление по сравнению с созданием «идеального» продукта. Это позволяет сосредоточиться на креативности и готовности к изменениям в процессе разработки.

- **Формирование сообщества и обмен опытом.** Game Jam способствует формированию сообщества разработчиков, художников и музыкантов. Участники могут познакомиться с единомышленниками, обмениваться знаниями и опытом. Это создает возможности для будущих коллабораций и долговременных профессиональных связей.

- **Поиск новых идей и тем.** В рамках Game Jam участники часто

исследуют новые темы и жанры, которые могут быть не так популярны в обычных условиях разработки. Это способствует созданию игр, которые выделяются на фоне традиционных проектов, и зачастую могут стать основой для успешных продуктов в будущем.

В целом, формат разработки игр Game Jam представляет собой свежий и инновационный подход, способствующий креативному процессу, сотрудничеству и быстрому прототипированию, что делает его привлекательным для разработчиков, а учебная программа, построенная на этих принципах, способствует развитию «мягких» (soft skills) и «твёрдых» (hard skills) навыков, необходимых широкому кругу современных технических специалистов.

Цель программы

Создание условий для изучения технологий, необходимых для разработки компьютерных и мобильных игр, и разработка игры с применением изученных технологий в жанре «Horror survival shooter» с опорой на принципы Game Jam.

Задачи программы

Образовательные:

- Ознакомить со специальным терминами и понятиями;
- Ознакомить с платформой для создания игр и интерактивного контента Unity 3D (далее Unity) и инструментами разработки игр, которые предоставляет Unity, сформировать представление о возможностях и ограничениях Unity;
- Ознакомить со специальным программным обеспечением, применяющимся для разработки игр, в том числе Blender 3D,
- Сформировать навыки практического применения технологий программирования на языке C# при решении поставленных технических задач;
- Ознакомить с проектной деятельностью: структурой и дизайном проекта, подходами к его разработке, включая ознакомление с главными этапами решения задач;

- Сформировать навыки работы в коллективе при реализации комплексных проектов;

- Сформировать навыки организации и распределения задач в команде, навыки достижения поставленных целей в срок.

Воспитательные:

- воспитывать интерес к занятиям информационными технологиями в целом и разработке игр в частности;

- воспитывать культуру общения между учащимися, культуру работы в команде;

- воспитывать культуру безопасного труда при работе за компьютером.

Развивающие:

- способствовать развитию критического, системного, алгоритмического и творческого мышления;

- развивать внимание, память, наблюдательность, познавательный интерес;

- развивать умение работать с компьютерными программами и дополнительными источниками информации;

- развивать навыки планирования проекта, умение работать в команде.

Категория обучающихся

Программа предназначена для детей, проявляющих интерес к технологиям разработки компьютерных и мобильных игр, к информационным технологиям в целом, а также для детей, стремящихся к саморазвитию.

Возраст обучающихся: 12 – 16 лет.

Структура программы

Программа курса состоит из 15 тем, выполняемых в формате мастер-классов с применением подготовленных материалов и информационных ресурсов.

Тема 1. Знакомство с платформой Unity 3D. Основы работы.

Тема 2 Настройка окружающей среды и создание уровня.

Тема 3. Создание игрового персонажа: трёхмерная модель и анимация

Тема 4. Создание игрового персонажа: программирование управления.

Тема 5. Настройка игровой камеры. Перспективная и ортогональная проекции. Программирование камеры.

Тема 6. Создание и настройка объекта неигрового персонажа. Система частиц. Физические объекты и столкновения.

Тема 7. Создание простого искусственного интеллекта для неигровых персонажей на основе NavMesh.

Тема 8. Создание игрового интерфейса. Знакомство с UI. Интерфейс для отображения здоровья персонажа.

Тема 9. Программирование логики нанесения урона. Атака противника и здоровье персонажа.

Тема 10. Программирование логики причинения вреда врагам. Оружие главного героя. RayCast. Состояние здоровья врагов. Убийство врагов.

Тема 11. Программирование системы подсчета очков.

Тема 12. Создание системы порождения врагов. Создание разнообразных типов врагов. Префабы.

Тема 13. Экран окончания игры. Проигрыш. Создание анимации для интерфейса средствами Unity.

Тема 14. Создание нескольких уровней и переход между ними.

Тема 15. Украшение игры и работа со звуком. Игра в целом (FMOD studio).

Форма реализации программы

Программа предполагает очную форму реализации с использованием электронного обучения.

Под электронным образованием понимается реализация образовательных программ с использованием информационно - образовательных ресурсов, информационно-коммуникационных технологий, технических средств, а также

информационно-телекоммуникационных сетей, обеспечивающих передачу информационно-образовательных ресурсов и взаимодействие участников образовательного пространства.

Формы организации деятельности обучающихся

При изучении тем программа предусматривает использование фронтальной, индивидуальной учебной работы обучающихся:

- фронтальная форма - для изучения нового материала, информация подаётся всей группе до 11 человек;
- индивидуальная форма - самостоятельная работа учащихся, педагог может направлять процесс в нужную сторону.
- практическая форма – закрепление полученных знаний и навыков.

Типы занятий: теоретические, практические в форме мастер-класса.

Режим занятий: один раз в неделю два часа.

Ожидаемые результаты

Основным результатом обучения является достижение определенного уровня информационно-коммуникационной компетентности учащегося в области разработки игр.

В процессе занятий обеспечивается целенаправленная работа на достижение личностных, метапредметных и предметных результатов обучения, обозначенных ФГОС.

Предметные результаты:

- знание принципов построения проектов в Unity, а также формы представления и управления информацией в них;
- понимание структуры Unity проектов и умение использовать компонентный подход Unity для создания внутри игровых объектов
- знание основ игрового дизайна, его ключевых особенностей и элементов
- умение создавать и модифицировать 3D-модели и текстуры, используя современные программные инструменты.

- умение создавать логику игр и игрового процесса и использованием языка программирования C#
- умение создавать и интегрировать анимации персонажей и объектов в игру
- умение разрабатывать игровые уровни и игровую среду с учетом механик игры и целевых взаимодействий с игроком
- умение тестировать игру, выявлять и исправлять ошибки в игровом процессе
- способности к совместной работе в команде над проектом, следуя принципам Agile или другим методологиям
- умение давать конструктивную обратную связь по вопросам разработки игр

Метапредметные результаты:

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учебе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- умение использовать техники ведения проектной деятельности и принципы тайм-менеджмента;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- владение основами самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать учебное сотрудничество и совместную деятельность с участниками образовательного процесса;
- умение работать индивидуально и в команде;
- умение формулировать, аргументировать и отстаивать свое мнение.

Личностные результаты:

- готовность и способность к самообразованию, саморазвитию, личностному и профессиональному самоопределению;
- наличие мотивации к обучению и целенаправленной познавательной деятельности;
- наличие системы значимых социальных и межличностных отношений, ценностно-смысловых установок, отражающих личностные и гражданские позиции в деятельности;
- умение ставить цели и строить жизненные планы;
- способность к осознанию российской идентичности в поликультурном социуме.

Способы определения результативности

Результативность выполнения программы определяется посредством педагогического наблюдения, педагогического анализа результатов выполнения текущих задач и публичного представления результатов деятельности.

Виды контроля: программа не предусматривает контрольно-измерительных мероприятий.

Формы подведения итогов реализации программы

По окончании обучения проводится итоговая аттестация в форме публичного представления результатов деятельности. Документальной формой подтверждения итогов аттестации является документ об образовании установленного Центром «Поиск» образца.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН КУРСА

«Unity game jam: создание игры в жанре horror survival shooter»

| № | Наименование кейса, темы | Количество часов | | |
|---|---|------------------|----------|-------|
| | | Теория | Практика | Всего |
| 1 | Тема 1. Знакомство с платформой Unity 3D. Основы работы. | 1 | 1 | 2 |
| 2 | Тема 2 Настройка окружающей среды и создание уровня. | 0 | 2 | 2 |
| 3 | Тема 3. Создание игрового персонажа: трёхмерная модель и анимация | 1 | 1 | 2 |
| 4 | Тема 4. Создание игрового персонажа: программирование управления. | 0 | 2 | 2 |
| 5 | Тема 5. Настройка игровой камеры. Перспективная и ортогональная проекции. Программирование камеры. | 1 | 1 | 2 |
| 6 | Тема 6. Создание и настройка объекта неигрового персонажа. Система частиц. Физические объекты и столкновения. | 1 | 3 | 4 |
| 7 | Тема 7. Создание простого искусственного интеллекта для неигровых персонажей на основе NavMesh. | 1 | 3 | 4 |
| 8 | Тема 8. Создание игрового интерфейса. Знакомство с UI. Интерфейс для отображения здоровья персонажа. | 1 | 1 | 2 |
| 9 | Тема 9. Программирование логики нанесения урона. Атака противника и здоровье персонажа. | 0 | 2 | 2 |

| № | Наименование кейса, темы | Количество часов | | |
|----|---|------------------|-----------|-----------|
| | | Теория | Практика | Всего |
| 10 | Тема 10. Программирование логики причинения вреда врагам. Оружие главного героя. RayCast. Состояние здоровья врагов. Убийство врагов. | 1 | 2 | 3 |
| 11 | Тема 11. Создание системы порождения врагов. Создание разнообразных типов врагов. Префабы. | 1 | 2 | 3 |
| 12 | Тема 12. Программирование системы подсчета очков. | 1 | 1 | 2 |
| 13 | Тема 13. Экран окончания игры. Проигрыш. Создание анимации для интерфейса средствами Unity. | 0 | 2 | 2 |
| 14 | Тема 14. Создание нескольких уровней и переход между ними. | 0 | 2 | 2 |
| 15 | Тема 15. Украшение игры и работа со звуком. Игра в целом (FMOD studio). | 0 | 2 | 2 |
| | Итого | 9 | 27 | 36 |

СОДЕРЖАНИЕ КУРСА

«Unity game jam: создание игры в жанре horror survival shooter»

Тема 1. Знакомство с платформой Unity 3D. Основы работы.

Теория

Концепция игрового движка Unity 3D и его применение. Основные особенности и преимущества использования Unity 3D. Обзор главных элементов интерфейса: сцена, инспектор, проект, консоль. Настройка рабочего пространства под свои нужды. Создание и управление игровыми объектами.

Практика

Пошаговое руководство по установке Unity и необходимых компонентов. Создание первого проекта настройка его параметров. Добавление игровых объектов на сцену: примитивы, камеры, источники света, настройка их свойств в инспекторе. Запуск тестовой сцены в редакторе и проверка работы созданных объектов и скриптов. Внесение корректировок на основе тестирования.

Обсуждение результатов работы, выводы и планы на следующее занятие.

Тема 2. Настройка окружающей среды и создание уровня.

Практика

Создание новой сцены, Инициализация новой сцены в Unity и настройка ее параметров, Импорт объектов, Импортировать готовые 3D модели из внешних источников (например, Asset Store), Настройка объектов окружения, Размещение объектов на сцене и настройка их материалов, работа с масштабом и ориентацией объектов, Добавление и настройка различных источников света на сцене для достижения реалистичной атмосферы.

Оценка итогового уровня, внесение необходимых изменений и подготовка к тестированию.

Тема 3. Создание игрового персонажа: трёхмерная модель и анимация.

Теория

Принципы создания трёхмерных моделей: полигональная сетка, текстуры и материалы и проведён обзор программы для 3D моделирования Blender 3D. Процесс экспорта 3D модели из программы моделирования и её импорт в Unity. Настройка параметров импорта для корректного отображения. Принципы создания костной структуры для модели. Основы анимации в Unity, работа с Animator. Создание анимационных клипов и использование переходов между анимациями.

Практика

Использование программы моделирования для создания простого трёхмерного персонажа. Настройка текстуры и материалов для модели. Экспортировать созданную модель в подходящем формате и импортировать её в Unity. Настройка параметров импортирования для корректной работы с моделью. Добавить костную структуру к модели и создать риг персонажа. Создать анимационные клипы для различных действий (например, ходьба, прыжок). Настройка Animator controller и добавление анимации.

Запуск сцены для проверки движений и анимаций персонажа, внесение необходимые корректировки.

После выполнения практического задания будет организовано обсуждение результатов, где ученики смогут поделиться своими подходами и задать вопросы по трудностям, возникшим в процессе работы.

Тема 4. Создание игрового персонажа: программирование управления.

Практика

Создание скрипта управления: написание скрипта на C# для управления движением персонажа; реализация перемещения по осям X и Z с использованием WASD или стрелок. Интеграция ввода: настройка системы ввода для реагирования на команды от пользователя; реализация прыжков и других действий персонажа. Использование физики: компонент Rigidbody для персонажа для работы с физическим движением; настройка параметров физики (массы, гравитации) для достижения желаемого поведения. Состояния и

анимация: связывание управления персонажем с анимациями (например, переходы между походкой и бегом). Создание системы для определения текущего состояния (например, проверка на приземление после прыжка).

Тестирование и отладка: запуск и тестирование скрипта управления в игровой сцене. устранение возможных багов и улучшение игрового взаимодействия.

После завершения практического задания обучающиеся обсудят увиденные проблемы и варианты их решения, а также поделятся своим опытом и успешными подходами к программированию управления игровым персонажем.

Тема 5. Настройка игровой камеры. Перспективная и ортогональная проекции. Программирование камеры.

Теория

Роль камеры в игре: общее назначение камеры и ее влияние на восприятие игрового мира.

Типы проекций: перспективная проекция: реализация ощущения глубины, важна для 3D-игр. Ортогональная проекция: используется в 2D играх и для изометрической графики, где размеры объектов не изменяются с расстоянием.

Настройка камеры в Unity: основные параметры камеры: положение, угол обзора, клиппинг (параметры ближнего и дальнего отображения). Инструменты для настройки камеры в Unity.

Программирование поведения камеры: Написание скриптов для динамического управления камерой. Основные паттерны управления камерой (следование за персонажем, фиксированный угол обзора и т.д.).

Практика

Настройка камеры: Создание новой сцены в Unity и настройка камеры для перспективной и ортогональной проекций. Изучение и изменение основных

параметров камеры.

Скрипт управления камерой: программирование скрипта для создания эффекта следования камеры за игровым персонажем. Реализация плавных переходов между движениями камеры и персонажем.

Тестирование: протестировать реализованные скрипты в сцене, удостоверившись в корректности работы камеры, обсудить полученные результаты и возможности улучшения реализации.

Тема 6. Создание и настройка объекта неигрового персонажа. Система частиц. Физические объекты и столкновения.

Теория

Неигровые персонажи: NPC в игре: их влияние на сюжет, взаимодействие с игроком и формирование атмосферы. Моделирование поведения NPC (основные механики, такие как патрулирование, поиск игрока). Система частиц: Введение в систему частиц в Unity: основные компоненты и возможности. Создание эффектов частиц, таких как взрывы, искры, дым и другие визуальные эффекты.

Практика

Создание NPC: создание 3D модели или использование стандартного ассета для NPC. Настройка компонентов NPC: добавление анимации, коллайдеров и Rigidbody для взаимодействия с физическим миром.

Программирование поведения NPC: написание скриптов для реализации базового поведения NPC (например, патрулирование и реагирование на игрока). Использование триггеров для активации определённых действий NPC.

Создание системы частиц: Изучение компонента Particle System в Unity и создание простых эффектов частиц. Настройка параметров частиц: скорость, размер, цвет и продолжительность.

Работа с физическими объектами: добавление коллайдеров к игровым объектам и их настройка (Box Collider, Sphere Collider и т.д.). Настройка параметров Rigidbody для управления физическим поведением объектов при

столкновениях.

Тестирование и отладка: Проверить взаимодействия NPC с игровым миром: столкновения, поведение при взаимодействии с игроком. Тестирование созданных эффектов частиц в игровом процессе, чтобы убедиться в их корректной работе.

Тема 7. Создание простого искусственного интеллекта для неигровых персонажей на основе NavMesh.

Теория

Искусственный интеллект: Понятие ИИ в играх: основные цели и задачи, роль в игровом процессе. Принципы работы ИИ для NPC: паттерны поведения, навигация, реагирование на действия игрока.

NavMesh: Введение в систему NavMesh: что это такое и как она упрощает навигацию AI в игровом мире. Процесс создания NavMesh: настройка поверхности и параметры.

Патрулирование и следование: основные механики патрулирования NPC с использованием NavMesh. Реализация поведения слежения за игроком и других объектами.

Практика

Создание NavMesh: Определение игровых объектов, по которым может перемещаться NPC (настройка NavMesh Static). Построение NavMesh для игрового уровня, проверка его корректности с помощью NavMesh Agent.

Настройка NPC: добавление компонента NavMesh Agent к NPC и настройка его параметров (скорость, радиус, высота). Изучение различных методов, таких как SetDestination(), для управления перемещением NPC.

Программирование поведения: разработка скриптов для патрулирования NPC по заданным точкам с использованием NavMesh. Реализация логики следования за игроком с использованием методов NavMesh Agent.

Тестирование и отладка: Проверка и отладка поведения NPC: корректное патрулирование и слежение за игроком. Настройка реакции NPC на

взаимодействия с игроком или другими объектами, например, изменение паттерна поведения при выполнении определённых условий.

Итерации и улучшения: Обсуждение полученных результатов и выявление проблем в поведении NPC. Рекомендации по улучшению сложности ИИ, например, добавление различных состояний поведения (погоня, укрытие).

Тема 8. Создание игрового интерфейса. Знакомство с UI. Интерфейс для отображения здоровья персонажа.

Теория

Основы игрового интерфейса: понятие UI в играх: функции и значение интерфейса для пользователя. Элементы UI: кнопки, панели, текстовые метки и их роли в взаимодействии с игроком.

Системы UI в Unity: введение в систему Canvas: создание и настройка интерфейса. Различные компоненты UI в Unity: Image, Text, Slider и их применение. Отображение здоровья игрока: Как измерять и визуализировать здоровье персонажа. Обработка событий, связанных со здоровьем, и обновление интерфейса в реальном времени.

Практика

Создание интерфейса: Canvas для игрового интерфейса. Добавление необходимых элементов UI: создание панели здоровья (Health Bar) с использованием Image и Slider.

Настройка элементов интерфейса: Изменение свойств элементов (размер, цвет, поведение) для отражения здоровья персонажа. Создание текстовой метки для отображения текущего уровня здоровья.

Программирование взаимодействия с интерфейсом: создание скрипта, который будет управлять значением здоровья персонажа. Использование событий и методов (например, Update(), OnHealthChange()) для обновления Health Bar в зависимости от состояния персонажа.

Тестирование и отладка: проверка работы интерфейса: корректное отображение изменения здоровья при получении урона или исцелении. Отладка

взаимодействия между логикой персонажа и интерфейсом, чтобы убедиться в актуальности отображаемых данных.

Итерации и улучшения: обсуждение возможных улучшений интерфейса: добавление анимаций, звуковых эффектов или дополнительной информации. Рассмотрение лучших практик по дизайну интерфейса, чтобы улучшить пользовательский опыт.

Тема 9. Программирование логики нанесения урона. Атака противника и здоровье персонажа.

Практика

Создание логики атаки: разработка скрипта для противника, который будет определять, когда и как атаковать игрока. Определение условий для атаки: расстояние до цели, период времени между атаками и тип атаки.

Нанесение урона: Реализация метода, который будет отвечать за нанесение урона игроку при атаке противника. Ввод параметров, таких как сила атаки, чтобы определить, сколько урона будет нанесено.

Обновление состояния здоровья: разработка системы, которая будет реагировать на юмор и обновлять здоровье персонажа. Добавление обратной связи для игрока: звуковые и визуальные эффекты при получении урона.

Тестирование механик: проверка логики атаки, чтобы убедиться, что противник корректно реагирует на персонажа. Отладка обновления состояния здоровья и отображения изменений (например, применение эффекта потери здоровья).

Анализ и доработка: обсуждение возможных улучшений и дополнительных механик, таких как критические удары или специальные способности. Рассмотрение внедрения системы исцеления и других способов взаимодействия игрока с механиками игры.

Тема 10. Программирование логики причинения вреда врагам. Оружие главного героя. RayCast. Состояние здоровья врагов. Убийство врагов.

Теория

Принципы причинения вреда врагам: определение механизмов, через которые игрок может причинить вред врагу. Роль оружия главного героя в механиках атаки.

Использование RayCast: понятие RayCast: как он работает, его применение для обнаружения объектов в игре. Применение RayCast для отслеживания попадания атак игрока по врагам.

Состояние здоровья врагов: структура здоровья врагов: параметры текущего и максимального здоровья. Реакция врагов на получение урона: различные состояния, такие как урон и смерть.

Логика убийства врагов: определение условий, при которых враг считается "убитым". Механизмы награды и обратной связи для игрока при убийстве врага.

Практика

Разработка логики оружия: создание скрипта для оружия главного героя, который будет обрабатывать атаку и нанесение вреда врагам. Определение параметров, таких как типы оружия, скорость, сила атаки и дальность.

Реализация RayCast: написание кода для использования RayCast для определения попадания атаки в противника. Учёт размеров объектов и расстояния при расчёте попадания.

Управление состоянием здоровья врагов: разработка системы, отвечающей за хранение и обновление здоровья врагов. Реакция врагов на получение урона: изменение состояния, визуальные и аудиоэффекты, уведомляющие игрока о повреждениях.

Логика убийства врагов: определение условий, при которых враг считается убитым (например, здоровье меньше или равно нулю). Обработка уничтожения врагов: анимация смерти, сбор ресурсов, награды игрока.

Тестирование механик: проверка работоспособности механики RayCast и логики причинения вреда. Обсуждение полученных результатов: стабильность и производительность скриптов, возможные баги и их устранение.

Анализ и улучшение: рассмотрение дополнительных механик, таких как

комбинированные атаки или специальные способности оружия. Идеи по улучшению визуальной и звуковой обратной связи для игрока.

Тема 11. Программирование системы подсчета очков.

Теория

Понятие системы подсчета очков: зачем нужна система подсчета очков в игре. Различные способы начисления очков: за убийства врагов, выполнение заданий, сбор предметов и т.д.

Структура системы очков: хранение текущего количества очков, максимальных достижений и рекордов. Использование переменных для хранения значений.

Обновление очков: логика начисления очков: условия и триггеры. Обработка различных событий, влияющих на счет игрока.

Отображение очков: способы вывода очков на экран: интерфейс и его элементы. Использование текстовых полей и графических элементов для представления информации.

Система достижений и наград: определение ключевых состояний для наград (например, за накопление определенного количества очков). Реализация системы уведомлений о достижениях.

Практика

Создание системы подсчета очков: написание скрипта, который будет управлять очками: начислением, обновлением и сохранением состояния.

Логика начисления очков: реализация методов для начисления очков за убийства врагов, выполнение миссий или сбор предметов. Определение триггеров для начисления очков (например, события столкновения).

Отображение очков в интерфейсе: разработка интерфейса для отображения текущих очков на экране. Написание кода для актуализации интерфейса при изменении очков.

Уведомления о достижениях: реализация системы появляющихся уведомлений при достижении игроком определенного количества очков.

Настройка графики и звуковых эффектов, связанных с достижениями.

Тестирование системы подсчета очков: проверка корректности начисления очков за различные действия. Обсуждение возможных ошибок или недочетов в реализации.

Оптимизация и улучшение: рассмотрение дополнительных возможностей: системы мультипликаторов очков или временных бонусов. Обсуждение идей для улучшения интерфейса и информативности системы.

Тема 12. Создание системы порождения врагов. Создание разнообразных типов врагов. Префабы.

Теория

Система порождения врагов: определение понятий: что такое порождение врагов и его роль в игре. Объяснение механики появления врагов на игровом поле: временные интервалы, условия и триггеры.

Типы врагов: обзор различных типов врагов: слабые, сильные, стремительные, с дальним атакующим режимом и т.д. Особенности поведения и искусственного интеллекта (ИИ) каждого типа.

Префабы: Что такое префаб в игровой разработке: преимущества использования префабов для врагов. Процесс создания и настройки префабов в игровом движке (например, Unity).

Разработка AI для врагов: основы создания логики поведения врагов: движение, атака, уклонение. Использование состояний и тактических решений в действиях врагов.

Оптимизация системы порождения: рассмотрение важных аспектов: производительность и управление памятью. Создание пула объектов для обработки врагов вместо постоянного создания и разрушения.

Практика

Создание системы порождения врагов: реализация скрипта для генерации врагов через определенные временные интервалы или события. Настройка начальных параметров, таких как количество врагов и зоны их появления.

Проектирование различных типов врагов: создание нескольких типов префабов врагов с разными характеристиками: здоровье, скорость, урон и поведения. Настройка параметров и их визуализация на игровом поле.

Разработка искусственного интеллекта: написание логики поведения для каждого типа врагов: патрулирование, преследование игрока, атака. Создание состояний для врагов: ожидание, атака, уклонение или уничтожение.

Создание и использование префабов: создание префабов для врагов в игровом движке: настройка визуальных и функциональных компонентов. Реализация возможности изменения параметров префаба без необходимости редактирования каждого объекта.

Тестирование системы: проверка работоспособности системы порождения и поведения врагов в различных сценариях. Обсуждение результатов тестирования, выявление недочетов и возможные направления для улучшений.

Оптимизация и расширение: обсуждение методов оптимизации системы порождения врагов. Идеи для добавления новых типов врагов с уникальными способностями и механиками.

Тема 13. Экран окончания игры. Проигрыш. Создание анимации для интерфейса средствами Unity.

Практика

Создание экрана окончания игры: разработка интерфейса с использованием UI-элементов Unity (текст, кнопки, изображения). отображаемой информации на экране окончания (например, финальный счет и сообщения о проигрыше).

Реализация логики проигрыша: написание скрипта для отслеживания условий проигрыша и вызова экрана окончания. Создание перехода на экран окончания при наступлении проигрыша.

Анимация интерфейса: создание анимаций для плавного появления и исчезновения элементов экрана окончания. Использование Animator и

анимационных кривых для настройки анимаций кнопок и сообщений.

Создание и настройка кнопок: добавление кнопок "Повторить" и "Выход в меню" на экран окончания. Настройка действий кнопок для повторного запуска игры или возвращения в главное меню.

Тестирование системы: проведение тестирования для проверки работы экрана окончания: проверка корректности отображаемой информации и функциональности кнопок. обсуждение результатов тестирования, выявление возможных недостатков в интерфейсе.

Оптимизация интерфейса: настройка экрана окончания для различных разрешений и аспектов экранов, тестирование на различных устройствах. Обсуждение возможных доработок и улучшений в интерфейсе для повышения удобства использования.

Тема 14. Создание нескольких уровней и переход между ними.

Практика

Создание нескольких уровней: разработка нескольких сцен в Unity, каждая из которых представляет отдельный уровень с уникальными особенностями и сложностями.

Настройка объектов на уровне: добавление игровых объектов, препятствий и врагов на каждом уровне. Настройка взаимодействий и триггеров для активации событий.

Реализация механики перехода: написание скриптов для перехода между уровнями: загрузка следующей сцены по нажатию кнопки или достижению определенной точки. Использование методов `OnTriggerEnter` для активации перехода и контроля состояния игры.

Создание меню между уровнями: моделирование промежуточного экрана или меню, позволяющего игроку перейти на следующий уровень или возвратиться назад. Настройка необходимых кнопок и их функциональности.

Сохранение прогресса: реализация системы сохранения, позволяющей игрокам продолжать с последнего достигнутого уровня. Настройка хранения

данных о текущем уровне с помощью PlayerPrefs.

Тестирование уровней: проведение тестирования созданных уровней на наличие ошибок и быстрой проверки механики перехода. Обсуждение с группой проблемы, выявленные в ходе тестирования, и предложения по улучшению.

Обсуждение и завершение: презентация разработанных уровней: демонстрация переходов и особенностей каждого уровня. Обсуждение возможных улучшений в дизайне уровней и механике игры.

Тема 15. Украшение игры и работа со звуком. Игра в целом (FMOD studio).

Практика

Создание визуальных элементов: проектирование и применение графических элементов на сцены: текстуры, анимации и эффекты. Настройка освещения для создания нужной атмосферы.

Работа в FMOD Studio: Импорт звуковых файлов и создание звуковых событий: эффектов окружения, звуков объектов и музыкальных тем. Настройка параметров звука: громкость, панорамирование и обработка.

Интеграция FMOD в Unity: Импорт FMOD банка звуков в проект Unity и настройка связи между событиями FMOD и объектами в игре. Программирование триггеров для активации звуков в зависимости от действий игрока.

Создание динамических звуковых эффектов: настройка переменных в FMOD для изменения звука в зависимости от игровых событий, например, изменение темпа музыки при изменении уровня здоровья персонажа. Использование пространственного звука для создания эффекта присутствия.

Тестирование звука и графики: проведение тестирования на предмет синхронизации звука и действия. Обсуждение результатов тестирования с группой, выявление проблем и предложение улучшений.

Презентация и обратная связь: демонстрация финальной версии игры с реализованными визуальными и звуковыми эффектами. Обсуждение отзывов однокурсников и преподавателя для дальнейшего улучшения.

СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ

1. Фернандес, Ж. Unity in Action: Multiplatform Game Development in C# / Ж. Фернандес. — Изд. 2-е. — М.: БХВ-Петербург, 2018. — 500 с.
2. Катя, П. Beginning Unity 5 Game Development / П. Катя. — М.: ИД "Питер", 2016. — 400 с.
3. Фум, Дж. The Unity Game Development Scripting Masterclass / Дж. Фум. — Изд. 1-е. — Лондон: Packt Publishing, 2018. — 300 с.
4. Каджур, А. Unity 2021 By Example / А. Каджур. — 3-е изд. — Лондон: Packt Publishing, 2021. — 350 с.
5. Нистром, Р. Game Programming Patterns / Р. Нистром. — Изд. 1-е. — С. Фрэнсиско: Genever Benning, 2014. — 200 с.
6. Матук, С. Unity 3D Game Development for Beginners / С. Матук. — М.: Издательство «Сон»; Питер, 2017. — 250 с.
7. Профи, Р. Learn Unity 2019: Beginner to Advanced / Р. Профи. — Изд. 1-е. — М.: ИД "БХВ-Петербург", 2019. — 480 с.
8. Нодер, М. Unity Game Optimization: Enhance and extend your game performance / М. Нодер. — Изд. 1-е. — Лондон: Packt Publishing, 2020. — 400 с.
9. Шелл, Дж. The Art of Game Design: A Book of Lenses / Дж. Шелл. — Изд. 2-е. — М.: ИД "Олимп", 2019. — 500 с.
10. Брант, К. Unity 2D Game Development / К. Брант. — Изд. 1-е. — М.: ИД "Питер", 2018. — 300 с.