

ИНДИВИДУАЛЬНЫЙ ПРЕДПРИНИМАТЕЛЬ
ВИХАРЕВА МАРЬЯ АЛЕКСАНДРОВНА



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Курс создания видеоигр на Unity»

возраст обучающихся: 12 – 14 лет

срок реализации: 1 год (72 часа)

наполняемость группы: 6 – 12 человек

Автор-составитель: Вихарева М.А.

Мирный, 2025 год

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.

Программа составлена на основе программы «Курс создания видеоигр на Unity» (А. Выговская, М. Липчанская) Международной школы программирования и математики «Алгоритмика» в соответствии с федеральными нормативными правовыми актами в области дополнительного образования, государственными требованиями к образовательным программам системы дополнительного образования детей:

- Федеральный Закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Концепция развития дополнительного образования детей до 2030 года (распоряжение Правительства РФ от 31.03.2022 г. № 678-р);
- Приказ Министерства просвещения Российской Федерации от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Приказ Министерства просвещения Российской Федерации от 03.09.2019 № 467 «Об утверждении целевой модели развития региональных систем дополнительного образования»;
- Приказ Минобрнауки России от 23.08.2017 № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки РФ от 18.11.2015 г. №09-3242);
- Методические рекомендации по реализации образовательных программ начального общего, основного общего, среднего общего образования, образовательных программ среднего профессионального образования и дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий от 20.03.2020 г.;
- Постановление Главного государственного санитарного врача РФ от 28.01.2021 г. №2 «Об утверждении СанПиН 1.2.3.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

Направленность программы: техническая.

Уровень программы – базовый.

Новизна программы заключается в технической направленности обучения, которое основывается на взаимодействии с современными цифровыми технологиями, что способствует развитию информационной культуры.

Актуальность программы заключается в:

- необходимости расширения кругозора школьника;
- формировании навыков планирования деятельности;
- формирование навыков работы с англоязычным интерфейсом;

- развитию пространственного мышления через формирование навыка создания 3D моделей различных объектов;
- развитию функциональной грамотности;
- развитию эмоционального интеллекта;
- развитию алгоритмического, пространственного и креативного мышления, творчества через создание собственных проектов с использованием цифровых ресурсов.

Педагогическая целесообразность программы заключается в развитии технического и творческого мышления у детей среднего школьного возраста через разработку игр, работу с информацией, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформы или интерфейса, формирование первичных представлений об информационной безопасности и правовых аспектах работы с информацией. Также развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, развитию сотрудничества и коллективной работы в цифровой среде.

Развитие навыков геймдизайна является одним из способов формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, критического, пространственного и креативного мышления.

Пространственное мышление влияет на формирование личности через такие психические процессы как внимание, восприятие, память, представление, воображение. Оно необходимо в практической и теоретической деятельности. Пространственное мышление является одним из главных показателей развития интеллектуальных и творческих способностей. С помощью него происходит постоянное перекодирование образов, т. е. переход от пространственных образов реальных объектов к их условно-графическим изображениям, от трехмерных изображений к двухмерным и обратно.

Одним из наиболее эффективных современных способов развития пространственного мышления является взаимодействие учащихся с 3D моделями и формирование навыка создания графических 3D моделей, важнейшей отличительной чертой которых является то, что при работе с ними можно в любой момент произвольно изменить ракурс изображения.

Цель программы – способствовать формированию пространственного и алгоритмического мышления у учащихся с помощью устойчивого интереса к процессу разработки игр.

Задачи:

Обучающие:

- познакомить с основными понятиями геймдизайна;
- познакомить с базовыми инструментами движка Unity;
- сформировать навык разработки игр с помощью Unity;
- сформировать навык использования языка программирования C# при разработке игр;
- сформировать навык понимания разных игровых стратегий и умения их проектировать;
- сформировать навык использования итерационного подхода при решении различных задач;
- формировать навык разработки сюжетной линии игр;
- формировать навык продвижения игр на платформе и привлечения новых пользователей;

- формировать навык тестирования и оптимизации созданных проектов;
- формировать навык кроссплатформенной разработки.

Развивающие:

- формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление;
- формировать навык публичного выступления и презентации;
- формировать навык планирования деятельности и основ тайм-менеджмента;
- формировать чувство цвета и стиля при оформлении материалов (эстетического вкуса);
- формировать и расширять словарный запас на английском языке;
- расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания;
- совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

Воспитательные:

- воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект;
- формировать навык работы в команде;
- формировать умение давать качественную обратную связь и реагировать на нее;
- формировать информационную культуру.

Отличительной особенностью программы является использование обучающих программ и приложений. На каждом занятии ученики работают в среде разработки несколькими способами: фронтально и индивидуально, что позволяет первоначально обсудить все сложности, которые могут возникнуть в процессе самостоятельной разработки. Каждый модуль завершается парным тестированием проектов.

В продолжении курса ученики осваивают все большее количество инструментов, создают разнообразные проекты на основе получаемых знаний и развивающихся умений. Эволюция компьютеров и программного обеспечения привела к достаточной простоте их освоения для самых неподготовленных пользователей, в том числе школьников.

Важным моментом подготовки детей к жизни в мире информационных технологий является развитие у них навыков работы с информацией в цифровой среде — поиск, анализ, создание и управление. Ребенок овладевает разными способами получения и обработки информации, и меняет свое отношение к новому классу техники и к миру предметов в целом.

В процессе разработки игр ученики узнают основные принципы работы игрового движка и игровой физики, навыки программирования и базовые принципы моделирования объектов.

Использование компьютерных технологий в работе с учащимися среднего школьного возраста является стремительно развивающейся образовательной методикой в мире. С ее помощью можно более эффективно решать образовательные задачи, которые будут способствовать качественному улучшению обучения ребенка в школе.

Адресат программы: обучение ведется в группах, которые комплектуются из обучающихся 12-14 лет. Отбор на курс предусмотрен только по возрасту.

Формы организации образовательного процесса:

Дети занимаются в кабинете с педагогом (групповая форма занятий), самостоятельно работая за компьютером (не более 20 минут за все занятие), занятия разделены перерывом.

Формы обучения:

- обучение от общего к частному (дедуктивный метод) и от частного к общему (индуктивный);
- поощрение вопросов и свободных высказываний по теме;
- уважение и внимание к каждому ученику;
- создание мотивационной среды обучения;
- создание условий для дискуссий и развития мышления учеников при достижении учебных целей вместо простого одностороннего объяснения темы преподавателем.

Занятие состоит из следующих блоков:

- повторение и определение целей на урок (5 минут);
- изучение нового материала (20 минут);
- практика (15 минут);
- разминка (10 минут);
- изучение нового материала (10 минут);
- практика (20 минут);
- рефлексия: подведение итогов занятия (10 минут).

Объем и срок освоения программы: дополнительная общеобразовательная общеразвивающая программа «Курс создания видеоигр на Unity» рассчитана на 1 год (72 часа; 362 занятий).

Режим занятий: 1 занятие в неделю по 2 академических часа.

Ожидаемые результаты.

Достижение цели и задач образовательной программы предполагает получение следующих результатов:

Планируемые результаты	Способ достижения	Критерий достижения образовательного результата
Предметные навыки		
Сформированность понятийного аппарата в области разработки игр.	Выполнение учебных проектов и работа на платформе, использование понятий в устной речи.	Ученик понимает, может объяснить и правильно использует основные понятия курса: геймдизайн, геймдизайнер, дизайн-документ, сценарий игры, скрипт, система координат, игра-платформа, виды компьютерных игр, пользователь и т. д.
Сформированность навыков разработки игр с помощью игрового движка Unity.	Выполнение учебных проектов и работа на платформе.	Ученик использует различные инструменты редактора Unity при самостоятельной разработке игр разных видов.

Сформированность навыка составления дизайн-документа и формального описания проекта.	Выполнение учебных проектов и работа на платформе.	Ученик умеет самостоятельно составлять дизайн-документы с описанием игры (умеет создавать сценарий для будущей игры, создает описание для каждого уровня и карту переходов между ними, продумывает дизайн уровней и оформляет их, работать по заранее составленному сюжету).
Сформированность навыка тестирования продукта.	Выполнение учебных проектов и работа на платформе.	Ученик умеет самостоятельно тестировать игры, создавать тест-кейсы, находить элементы, нарушающие целостность игры, исправлять и дорабатывать их.
Развитие умений применять C# для разработки логики видеоигры.	Выполнение учебных проектов и работа на платформе.	Ученик самостоятельно пишет логику видеоигры на языке C#.
Усвоение основных принципов продвижения игр.	Выполнение учебных проектов и работа на платформе.	Ученик умеет использовать доступные инструменты для продвижения игры, создавать CG-ролики.
Усвоения принципов кроссплатформенной разработки.	Выполнение учебных проектов и работа на платформе.	Ученик умеет самостоятельно настраивать скрипты для кроссплатформенной разработки, в т. ч. мобильной.
Личностные навыки		
Освоение социальной роли обучающегося и формирование личностного смысла учения.	Демонстрация связи между способностью выполнить интересную задачу и наличием/отсутствием соответствующих знаний.	За отведенное время ученик пытается не только выполнить базовые уровни, но и приступить к бонусным.
Развитие навыков сотрудничества со взрослыми и сверстниками и умения находить выходы из спорных ситуаций.	Использование программирования как способа показать преимущества работы в команде.	Ученик не боится просить помощь и сам пытается помогать одноклассникам и учителю.
Овладение начальными навыками адаптации в динамично изменяющемся и развивающемся мире.	Обсуждение научно-популярных материалов, демонстрирующих необходимость не просто овладения предметными	При работе над проектом ученик сначала пытается найти информацию в Интернете или теоретической справке на платформе и только при неудаче задает вопрос учителю.

	навыками, но и развития умения учиться.	
Метапредметные навыки		
Развитие и формирование учебных действий.	Создание благоприятных условий для участия в диалоге, в коллективном обсуждении. Строится продуктивное взаимопонимание со сверстниками и взрослыми в процессе коллективной деятельности.	Ученик легко общается, не боится просить помощь или оказать ее другим. Ученик способен кооперироваться, чтобы достичь цели.
Умение презентовать свою работу.	Презентация индивидуальных проектов. Учитель и другие ученики дают обратную связь. Учитель также дает советы, каким образом это лучше делать.	Ученик во время презентации своих проектов пользуется вниманием аудитории.
Развитие критического и алгоритмического мышления учеников.	Выполнение логических операций: сравнение, анализ, синтез, обобщение, установление аналогий на уроке. Анализ любой получаемой информации, составление плана действий для решения задачи.	Ученик строит логическую цепь рассуждений. Управляет своей деятельностью.
Развитие творческих способностей учеников.	Планирование и создание проекта по сценарию. Использование инструментов цифрового сервиса для воплощения своего проекта.	Ученик придумывает, обсуждает, планирует и реализует свой проект.

Формы аттестации и оценочные материалы. Текущий контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов на нескольких уровнях:

- **на каждом занятии:** опрос, выполнение заданий на платформе, реализация проектов, взаимоконтроль учеников в парах, самоконтроль ученика;
- **в конце каждого модуля:** проведение презентации (по желанию) финальных проектов модуля и их оценка.

Для контроля сформированности результатов освоения программы с помощью выполнения заданий на платформе используются следующие механики:

- классический тест (выбор одного или нескольких правильных ответов),
- заполнение пропусков,
- классификация,
- сопоставление,
- сортировка,
- ответ в свободной форме, в т. ч. прикрепление файла.

На курсе возможны следующие **формы деятельности** учащихся: учебная дискуссия, занятие-проект, практическое занятие. При решении одной задачи ученики могут работать индивидуально, затем в парах, а при проверке задачи – в паре с учителем.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН.

№ п/п	Наименование раздела/темы	Количество часов		
		всего	теория	практика
<i>Модуль 1. Основы Unity (8 часов).</i>				
1	Знакомство с Unity.	2	1	1
2.	Работа с игровыми объектами и префабами.	2	1	1
3.	Работа с материалами и текстурами.	2	1	1
4.	Создание прототипа игрового уровня.	2	1	1
<i>Модуль 2. Дизайн 3D-уровней (8 часов).</i>				
5.	Начало работы с ландшафтом.	2	1	1
6.	Детализация ландшафта.	2	1	1
7.	Настройка света.	2	1	1
8.	Визуальные и звуковые эффекты.	2	1	1
<i>Модуль 3. Программирование видеоигр на языке C# (14 часов).</i>				
9	Введение в язык C#.	2	1	1
10.	Переменные и типы данных.	2	1	1
11.	Условные конструкции.	2	1	1
12	Циклы.	2	1	1
13.	Объекты и методы.	2	1	1
14.	Классы.	2	1	1
15.	Пользовательский ввод.	2	1	1
<i>Модуль 4. Анимация 3D-объектов (10 часов).</i>				
16.	Введение в анимацию.	2	1	1
17.	Работа с аниматором. Анимация гуманоида.	2	1	1
18.	Работа с аниматором. Анимация других объектов.	2	1	1
19.	Создание видеоигрового трейлера. CG-ролик.	4	2	2
<i>Модуль 5. Специфика 2D-игр. Мобильная платформа (8 часов).</i>				
20.	Введение в 2D-игры.	2	1	1

21.	Анимация в 2D-играх.	2	1	1
22.	Физика в 2D-играх.	2	1	1
23.	Сборка под Android и Play Market.	2	1	1
Модуль 6. Создание гоночной 3D-игры (10 часов).				
24.	Знакомство с жанром гоночных игр.	2	1	1
25.	Дизайн уровней для гоночной игры.	4	2	2
26.	Программирование логики гоночной игры. Часть 1.	2	1	1
27.	Программирование логики гоночной игры. Часть 2.	2	1	1
Модуль 7. Создание ролевой 3D-игры (10 часов).				
28.	Знакомство с жанром RPG.	2	1	1
29.	Дизайн уровней для RPG-игры.	4	2	2
30.	Программирование логики RPG-игры. Часть 1.	2	1	1
31.	Программирование логики RPG-игры. Часть 2.	2	1	1
Модуль 8. Итоговая аттестация (4 часа).				
32.	Создание портфолио Unity-разработчика.	2	1	1
33.	Тестирование. Итоговая аттестация.	2	1	1
Всего:		72	36	36

3. СОДЕРЖАНИЕ ПРОГРАММЫ.

Модуль 1. Основы Unity (8 часов)

1.1. Знакомство с Unity.

Теория. Геймдизайнер. Опыт игрока. Игровой движок. Интерфейс редактора Unity. Управление камерой сцены. Виды примитивов. Компоненты объектов. Редактирование положения и формы примитивов — компонент Transform. Игровые объекты и ассеты.

Практика. Создание игровой сцены с несколькими 3D-объектами.

1.2. Работа с игровыми объектами и префабами.

Теория. Отличие 3D-пространства от 2D. Координатная система сцены и игрового объекта. Вложенные объекты. Иерархия игровых объектов. Префабы и экземпляры. Принцип наследования.

Практика. Настройка иерархии игровых объектов сделанной ранее сцены и её расширение за счёт экземпляров префабов.

1.3. Работа с материалами и текстурами.

Теория. 3D-рендеринг. Пиксели. Растровый рисунок. 3D-модель объекта. Полигональная сетка. Триангуляция. Низкополигональные и высокополигональные 3D-модели. Параметры материала 3D-объекта. Шейдер. Текстура.

Практика. Создание визуального стиля сделанной ранее сцены.

1.4. Создание прототипа игрового уровня.

Теория. Цифровой прототип. Цикл создания видеоигр. Дизайн-документ. Линейный и разветвлённый типы игровых уровней. Потайные места и хабы. Схема игрового уровня. Импорт ассетов из Unity Asset Store. Тестирование прототипа. Создание WebGL-билда проекта.

Практика. Создание прототипа игрового уровня по сформированному дизайн-документу и его тестирование.

Модуль 2. Дизайн 3D-уровней (8 часов).

2.1. Начало работы с ландшафтом.

Теория. Левел-дизайн. Объект Terrain. Формирование ландшафта с помощью карты высот (Heightmap). Настройка кистей для работы с ландшафтом.

Практика. Создание ландшафта для игрового уровня.

2.2. Детализация ландшафта.

Теория. Работа с Terrain Layer. Текстурирование ландшафта — инструмент Paint Texture. Добавление и настройка природных объектов (травы, деревья, ветра) — инструменты Paint Trees, Paint Details и Terrain Setting.

Практика. Детализация созданного ранее ландшафта.

2.3. Настройка света.

Теория. Атмосфера в видеоигре. Основные виды источника света: точечный (point), направленный (spot) и глобальный направленный (directional). Свойства разных видов источника света. Приёмы светового дизайна.

Практика. Создание определённой атмосферы видеоигры с помощью средств светового дизайна.

2.4. Визуальные и звуковые эффекты.

Теория. Система частиц. Свойства модулей системы частиц (компонент Particle System). Окно Particle Effect. Работа с Curve Editor. Компонент Audio listener. Настройка компонента Audio source (источник звука). Добавление звуковых файлов (audio clips) к источнику звука.

Практика. Усиление выбранной атмосферы видеоигры с помощью световых и звуковых эффектов. Тестирование и презентация финальной версии проекта.

Модуль 3. Программирование видеоигр на языке C# (14 часов).

3.1. Введение в язык C#.

Теория. Механики 3D-платформера. Понятия «язык программирования», «алгоритм», «программа», «скрипт». Интегрированная среда разработки Visual Studio. Синтаксис языка C#. Структура шаблона скрипта в Unity. Создание скрипта в Unity и его добавление к игровому объекту. Функции Start() и Update(). Вывод информации в консоль редактора Unity (функция print()). Написание комментариев.

Практика. Создание и тестирование скриптов, осуществляющих вывод данных.

3.2. Переменные и типы данных.

Теория. Данные. Типы данных. Переменная. Числовой тип данных. Оператор присваивания. Арифметические операции с переменными. Конкатенация.

Практика. Создание и тестирование скриптов, обрабатывающих переменные числового типа и арифметические операции с ними.

3.3. Условные конструкции.

Теория. Логический тип данных. Утверждение. Условие. Условный оператор. Синтаксис условного оператора. Операторы сравнения. Коллайдеры и триггеры (функции `OnTriggerEnter()`, `OnTriggerExit()`, `OnTriggerStay()`).

Практика. Создание и тестирование скриптов, содержащих условный оператор.

3.4. Циклы.

Теория. Цикл. Необходимость применения циклов. Типы циклов. Демонстрация синтаксиса цикла `while()`. Разбор примеров скриптов с использованием цикла. Цикл со счетчиком. Цикл с накоплением результата вычислений в переменной.

Практика. Создание и тестирование скриптов, содержащих цикл.

3.5. Объекты и методы.

Теория. Объекты. Методы и свойства объектов. Создание объектов. Написание методов. Изменение параметров 3D-объектов (функция `GetComponent()`).

Практика. Создание и тестирование скриптов, меняющих свойства 3D-объектов.

3.6. Классы.

Теория. Классы. Основные классы в Unity (`GameObject`, `Vectors`, `Time`). Конструктор класса. Наследование. Класс-наследник. Использование готовых свойств и методов объекта. Добавление новых методов.

Практика. Разработка скриптов, осуществляющих обработку классов.

3.7. Пользовательский ввод.

Теория. События. Обработка взаимодействия пользователя и видеоигры. Обработка событий мыши (функции `OnMouseDown()`, `OnMouseUp()`, `OnMouseExit()`). Создание пользовательского интерфейса (UI).

Практика. Создание логики и интерфейса видеоигры в жанре «3D-платформер». Тестирование финальной версии проекта.

Модуль 4. Анимация 3D-объектов (10 часов).

4.1. Введение в анимацию.

Теория. 3D-анимация. Этапы создания 3D-анимации в Unity. Настройка анимации с помощью окна `Animation`. Запись анимации — инструмент `Record Mode`. Создание плавных переходов в анимации — инструмент `Curves Editor`.

Практика. Настройка и запись анимации примитивов.

4.2. Работа с аниматором. Анимация гуманоида.

Теория. Подготовка 3D-модели для анимации. Скелет (`rig`) 3D-модели. Система `Mecanim`. Работа с объектом `Animator Controller`. Создание переходов между состояниями персонажа в окне `Animator`.

Практика. Создание анимации нескольких состояний персонажа видеоигры.

4.3. Работа с аниматором. Анимация других объектов.

Теория. Анимация игровых объектов. Написание скриптов для анимации объектов в зависимости от игрового события.

Практика. Создание анимации игровых объектов под конкретные события в видеоигре.

4.4. Создание видеоигрового трейлера. CG-ролик.

Теория. Примеры CG-роликов. Процесс создания CG-роликов. Планирование записи видеоигрового трейлера.

Практика. Создание нескольких катсцен с главным героем видеоигры. Презентация роликов.

Модуль 5. Специфика 2D-игр. Мобильная платформа (8 часов).

5.1. Введение в 2D-игры.

Теория. Основные отличия 2D-игр от 3D-игр. Жанровые особенности 2D-игр. Спрайты. Особенности жанра «шутер». Планирование разработки шутера.

Практика. Создание дизайн-документа и первого прототипа 2D-шутера.

5.2. Анимация в 2D-играх.

Теория. Особенности 2D-анимации. Создание спрайт-листа (sprite sheet) для покадровой анимации — инструмент Sprite Editor. Работа со слоями спрайтов.

Практика. Создание анимации спрайтов для 2D-шутера.

5.3. Физика в 2D-играх.

Теория. Особенности физики в 2D-играх. RigidBody 2D. Коллайдеры 2D-объектов. Механики на основе физики 2D-объектов.

Практика. Написание логики 2D-шутера.

5.4. Сборка под Android и Play Market.

Теория. Особенности мобильной платформами. Создание билда под мобильную платформу Android.

Практика. Создание финальной версии 2D-шутера для мобильной платформы Android. Тестирование и презентация проекта.

Модуль 6. Создание гоночной 3D-игры (10 часов).

6.1. Знакомство с жанром гоночных игр.

Теория. Особенности жанра «гоночная игра». Планирование разработки гоночной игры.

Практика. Создание дизайн-документа для гоночной 3D-игры.

6.2. Дизайн уровней для гоночной игры.

Теория. Принципы дизайна уровней для гоночной игры.

Практика. Создание игровых уровней для гоночной игры.

6.3. Программирование логики гоночной игры. Часть 1.

Теория. Основные механики гоночной игры.

Практика. Создание скриптов для реализации основных механик гоночной игры.

6.4. Программирование логики гоночной игры. Часть 2.

Теория. Дополнительные механики гоночной игры. Особенности создания пользовательского интерфейса для гоночной игры.

Практика. Создание финальной версии гоночной 3D-игры. Тестирование и презентация проекта.

Модуль 7. Создание ролевой 3D-игры (10 часов).

7.1. Знакомство с жанром RPG.

Теория. Особенности жанра «ролевая игра». Планирование разработки ролевой игры.

Практика. Создание дизайн-документа для ролевой 3D-игры.

7.2. Дизайн уровней для RPG-игры.

Теория. Принципы дизайна уровней для ролевой игры.

Практика. Создание игровых уровней для ролевой игры.

7.3. Программирование логики RPG-игры. Часть 1.

Теория. Основные механики ролевой игры.

Практика. Создание скриптов для реализации основных механик ролевой игры.

7.4. Программирование логики RPG-игры. Часть 2.

Теория. Дополнительные механики ролевой игры. Особенности создания пользовательского интерфейса для ролевой игры.

Практика. Создание финальной версии ролевой 3D-игры. Тестирование и презентация проекта.

Модуль 8. Итоговая аттестация (4 часа).

8.1. Создание портфолио Unity-разработчика. Тестирование. Итоговая аттестация.

Теория. Правила создания портфолио разработчика видеоигр. Инструменты создания цифрового портфолио.

Практика. Создание цифрового портфолио с описанием всех созданных на курсе проектов. Тестирование.

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ.

Материально-техническое обеспечение:

Для каждого обучающегося:

- канцелярия: бумага А4, ручки/карандаши;
- персональный ноутбук;
- персональный логин и пароль для доступа на платформу;
- платформа Алгоритмики, открывается через браузер Google Chrome;
- десктопное приложение Unity Hub, разработка игр происходит с помощью редактора Unity и интегрированной среды разработки Visual Studio;
- компьютеры (ноутбуки) должны быть подключены к единой сети с доступом в Интернет.

Для преподавателя:

- презентационное оборудование (проектор с экраном/телевизор с большим экраном) с возможностью подключения к компьютеру (ноутбуку);
- флипчарт с комплектом листов / маркерная доска, соответствующий набор письменных принадлежностей.

Информационное обеспечение:

- поурочные методические рекомендации к занятиям (платформа Алгоритмика);
- тематические презентации (платформа Алгоритмика);
- задания (платформа Алгоритмика);
- бонусные задания (платформа Алгоритмика);
- программы и цифровые сервисы, в которых ученики будут создавать проекты;
- справочный теоретический материал, размещенный на платформе;

- задачи со свободным решением для реализации в среде разработки на базе платформы «Алгоритмика».

Кадровое обеспечение

Уровень образования среднее профессиональное или высшее.

5. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ.

Основным методом организации учебной деятельности по программе является метод кейсов.

Кейс – описание проблемной ситуации понятной и близкой обучающимся, решение которой требует всестороннего изучения, поиска дополнительной информации и моделирования ситуации или объекта, с выбором наиболее подходящего.

Преимущества метода кейсов:

- практическая направленность. Кейс-метод позволяет применить теоретические знания к решению практических задач.;
- интерактивный формат. Кейс-метод обеспечивает более эффективное усвоение материала за счёт высокой эмоциональной вовлечённости и активного участия обучаемых. Участники погружаются в ситуацию с головой: у кейса есть главный герой, на место которого ставит себя команда и решает проблему от его лица. Акцент при обучении делается не на овладение готовым знанием, а на его выработку;
- конкретные навыки. Кейс-метод позволяет совершенствовать soft-skills, которым уделяют мало внимания в классических образовательных учреждениях, но которые оказываются крайне необходимы на протяжении всей жизни.