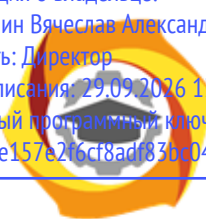


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Лапин Вячеслав Александрович
Должность: Директор
Дата подписания: 29.09.2026 13:31:55
Уникальный программный ключ:
df48b51be157e2f6cf8adf83b04ff59abaacac



ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
УГМК



Негосударственное частное образовательное учреждение высшего образования
«Технический университет УГМК»
(НЧОУ ВО «ТУ УГМК»)



УТВЕРЖДАЮ
Директор НЧОУ ВО «ТУ УГМК»

В.А. Лапин
(инициалы, фамилия)

« 1 » сентября 2026 г.

**Дополнительная общеобразовательная общеразвивающая
ПРОГРАММА**

«Разработка игр на Unity»

(наименование программы)

Направленность: информатика и вычислительная техника

Возраст обучающихся: 12-15 лет

Срок реализации: 8 месяцев

Автор-разработчик:
Антипин Алексей Александрович

Верхняя Пышма, 2026 год

Оглавление

1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ.....	3
1.1. Пояснительная записка.....	3
1.2. Актуальность.....	4
1.3. Цель и задачи Программы	5
1.4. Формы и режим занятий	6
1.5. Учебный (тематический) план	6
1.6. Содержание учебного (тематического) плана	8
1.7. Планируемые результаты	13
1.8. Календарный учебный график	16
2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ	17
2.1. Условия реализации программы	17
2.2 Форма аттестации/контроля и оценочные материалы.....	17
3. СПИСОК ЛИТЕРАТУРЫ.....	18

1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

1.1. Пояснительная записка

Направленность общеобразовательной программы:

Современный мир предъявляет новые требования к молодому поколению, вступающему в жизнь, так как будущее сегодняшних детей — это информационное общество, в котором индустрия компьютерных игр и интерактивных развлечений занимает всё более заметное место. Информатизация образования открывает новые возможности для развития методов и организационных форм воспитания и обучения детей. В современных условиях родители и педагоги должны быть готовы к тому, что ребёнок будет ежедневно взаимодействовать с информационными технологиями, поэтому важно научить ребёнка осознанному и созидательному взаимодействию с цифровым миром.

Разработка игр в Unity — это образовательный инструмент, который позволяет учащимся напрямую взаимодействовать с профессиональной сферой геймдева. Работая в кроссплатформенном игровом движке Unity, учащиеся знакомятся с инструментами 3D- и 2D-разработки, основами программирования на языке C#, принципами левел-дизайна, анимации и работы над завершённым игровым проектом.

Для успешного обучения в школе и жизни в мире информационных технологий важен не столько набор знаний, сколько развитое мышление, умение получать знания, использовать имеющиеся навыки для решения различных учебных задач. Большие возможности при этом открываются при работе с компьютером и специализированным программным обеспечением.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии технического и творческого мышления у детей среднего школьного возраста через разработку игровых проектов: создание трёхмерных и двухмерных уровней, программирование игровой логики, работу с анимацией, светом и визуальными эффектами. Также программа формирует первичные представления об архитектуре программных систем, работе с системами контроля версий (GitHub) и правовых аспектах работы с готовыми цифровыми ассетами.

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

Актуальность общеобразовательной программы

Индустрия компьютерных игр — одна из наиболее динамично растущих отраслей современной экономики, а профессии геймдизайнера, программиста и технического художника становятся всё более востребованными. Игровой движок Unity является одним из самых распространённых инструментов индустрии: на нём создаются как учебные прототипы, так и коммерческие проекты для персональных компьютеров, консолей и мобильных устройств.

Курс «Разработка игр в Unity» позволит учащимся познакомиться с полным циклом создания игрового проекта — от первой сцены и левел-дизайна до программирования игровой механики, анимации персонажа и публикации итогового результата. Работая над модульными проектами, учащиеся объединят полученные знания в законченные игры: трёхмерный уровень, небольшую

игру на C#, анимированную кат-сцену, 2D RPG и финальный 3D-проект с презентационным GIF-роликом.

Важным моментом подготовки детей к жизни в мире информационных технологий является развитие у них навыков работы со сложными программными системами — планирование, декомпозиция задачи, отладка и итерационное улучшение результата. Использование игрового движка в работе с детьми среднего школьного возраста — стремительно развивающаяся методика в образовании во всём мире, позволяющая совмещать техническое и творческое развитие ребёнка.

Актуальность программы заключается в:

- необходимости расширения кругозора школьника в области информационных технологий и геймдизайна;
- формировании основных навыков разработки игр в профессиональном игровом движке;
- формировании базовых навыков алгоритмического мышления и программирования на языке C#;
- формировании навыков командной и системной работы с использованием инструментов контроля версий;
- формировании понимания принципов левел-дизайна, композиции и работы со светом;
- развитии цифровой и функциональной грамотности;
- развитии алгоритмического, пространственного и креативного мышления через создание собственных игровых проектов.

Новизна программы

Новизна программы заключается в технической направленности обучения, основанной на взаимодействии с профессиональным инструментарием разработки игр. Авторское воплощение замысла курса состоит в том, что ученики с первого занятия работают непосредственно в редакторе Unity и на языке программирования C#, постепенно переходя от простых сцен из примитивов к завершённым игровым проектам в 3D и 2D. Программирование вводится не абстрактно, а через практические задачи внутри игровых сцен, а работа с системой контроля версий GitHub формирует у учащихся начальные представления о профессиональной разработке программного обеспечения.

В продолжение курса ученики осваивают всё большее количество инструментов движка — системы Terrain, освещение, анимацию, Timeline, Tilemap, — и на их основе создают разнообразные проекты по мере развития знаний и умений.

Адресат: учащиеся средних классов школы в возрасте от 12 до 15 лет включительно.

Режим занятий: занятия проходят 1 раз в неделю. Дети занимаются в компьютерном классе с преподавателем, самостоятельно работая за компьютером в редакторе Unity.

Занятия проводятся в группах не более 15 человек и индивидуально. Язык обучения-русский.

Объём: 2 академических часа (1 академический час-45 минут).

Форма обучения:

- Игровая, задачная и проектная.
- Обучение от общего к частному: от простых сцен из примитивов к завершённым игровым проектам.
- Поощрение вопросов и свободных высказываний по теме.
- Уважение и внимание к каждому ученику.
- Создание мотивационной среды обучения.
- Создание условий для обсуждения игровых решений и развития мышления учеников вместо простого одностороннего объяснения темы преподавателем.

Основные формы работы — индивидуальная и групповая. По итогам каждого модуля учащиеся защищают итоговый проект модуля перед группой.

1.2. Цель и задачи общеразвивающей программы

Цель курса — способствовать формированию у учащихся устойчивого интереса к разработке компьютерных игр и созданию собственных завершённых игровых проектов в Unity.

Задачи:

Обучающие:

1. Познакомить с интерфейсом и основными инструментами редактора Unity.
2. Познакомить с основными понятиями левел-дизайна и работы с 3D-сценой.
3. Сформировать навык программирования игровой логики на языке C#: переменные, условные конструкции, циклы, методы, классы и массивы.
4. Сформировать навык работы с системой контроля версий GitHub.
5. Сформировать навык создания и настройки анимации 3D- и 2D-объектов.
6. Сформировать навык разработки как трёхмерных, так и двухмерных игровых проектов.
7. Сформировать навык записи и оформления игрового процесса для презентации проекта.

Развивающие:

1. Сформировать и развивать логическое, алгоритмическое и критическое мышление.
2. Сформировать навык публичного выступления и презентации собственного проекта.
3. Расширять кругозор, развивать память, внимание, пространственное воображение и абстрактно-логические виды мышления.
4. Совершенствовать диалогическую речь учащихся: умение слушать собеседника, задавать вопросы и отвечать на них.
5. Формировать эстетическое чувство при оформлении игровых уровней и сцен.

Воспитательные:

1. Воспитывать у учащихся потребность в сотрудничестве и взаимодействии со сверстниками при работе над групповыми проектами.
2. Формировать умение давать качественную обратную связь и реагировать на неё.
3. Формировать информационную культуру и уважение к результатам чужого труда, в том числе к готовым цифровым ассетам.

1.3. Содержание общеразвивающей программы

Учебно-тематический план

№ п/п	Наименование раздела/темы	Всего	Теория	Практика	Форма аттестации (контроля)
	Модуль 1. Введение в Unity	4,5	1,5	3	
1	Знакомство с Unity и создание первой сцены	1,5	0,5	1	Практическая работа с проверкой преподавателем
2	Материалы, текстуры и префабы	1,5	0,5	1	Практическая работа с проверкой преподавателем
3	Творческий проект: создание уровня	1,5	0,5	1	Презентация проекта
	Модуль 2. Дизайн 3D-уровней	6	2	4	
4	Введение в левел-дизайн и система Terrain	1,5	0,5	1	Практическая работа с проверкой преподавателем
5	Детализация ландшафта: деревья и трава	1,5	0,5	1	Практическая работа с проверкой преподавателем
6	Световой дизайн и освещение	1,5	0,5	1	Практическая работа с проверкой преподавателем
7	Визуальные и звуковые эффекты	1,5	0,5	1	Презентация проекта
	Модуль 3. Программирование на C#	16,5	5,5	11	
8	Основы программирования и работа с GitHub	1,5	0,5	1	Практическая работа с проверкой преподавателем
9	Хранение данных. Переменные	1,5	0,5	1	Практическая работа с проверкой преподавателем
10	Условные конструкции	1,5	0,5	1	Практическая работа с проверкой преподавателем
11	Циклы	1,5	0,5	1	Практическая работа с проверкой преподавателем
12	Методы и функции	1,5	0,5	1	Практическая работа с проверкой преподавателем

№ п/п	Наименование раздела/темы	Всего	Теория	Практика	Форма аттестации (контроля)
13	Практика: методы и функции	1,5	0,5	1	Практическая работа с проверкой преподавателем
14	Классы и взаимодействие объектов	1,5	0,5	1	Практическая работа с проверкой преподавателем
15	Практика: классы и системы	1,5	0,5	1	Практическая работа с проверкой преподавателем
16	Массивы, списки и UI	1,5	0,5	1	Практическая работа с проверкой преподавателем
17	Итоговый проект модуля. Часть 1	1,5	0,5	1	Практическая работа с проверкой преподавателем
18	Итоговый проект модуля. Часть 2	1,5	0,5	1	Презентация проекта
	Модуль 4. Анимация 3D-объектов	7,5	2,5	5	
19	Введение в анимацию	1,5	0,5	1	Практическая работа с проверкой преподавателем
20	Анимация персонажа. Часть 1	1,5	0,5	1	Практическая работа с проверкой преподавателем
21	Контроллер движения персонажа	1,5	0,5	1	Практическая работа с проверкой преподавателем
22	Анимация персонажа. Часть 2: Blend Trees	1,5	0,5	1	Практическая работа с проверкой преподавателем
23	Завершение проекта. Кат-сцены и Timeline	1,5	0,5	1	Презентация проекта
	Модуль 5. Создание 2D-игры	7,5	2,5	5	
24	Введение в 2D-разработку. Начало проекта	1,5	0,5	1	Практическая работа с проверкой преподавателем
25	2D-физика и движение персонажа	1,5	0,5	1	Практическая работа с проверкой преподавателем
26	Tilemap и создание уровней	1,5	0,5	1	Практическая работа с проверкой преподавателем

№ п/п	Наименование раздела/темы	Всего	Теория	Практика	Форма аттестации (контроля)
27	Игровая механика RPG	1,5	0,5	1	Практическая работа с проверкой преподавателем
28	Завершение 2D RPG-проекта	1,5	0,5	1	Презентация проекта
	Модуль 6. Создание 3D-игры	6	2	4	
29	Побег из лаборатории. Создание прототипа уровня	1,5	0,5	1	Практическая работа с проверкой преподавателем
30	Препятствия и левел-дизайн. Физика и анимация	1,5	0,5	1	Практическая работа с проверкой преподавателем
31	Противники и финальная механика побега	1,5	0,5	1	Практическая работа с проверкой преподавателем
32	Создание GIF-анимации и запись геймплея	1,5	0,5	1	Презентация проекта
	Всего часов:	48	16	32	

В течение курса ученики создадут не менее 6 индивидуальных и групповых проектов, объединяющих знания и умения соответствующего модуля.

1.4. Содержание учебного плана

Модуль 1. Введение в Unity

1.1. Знакомство с Unity и создание первой сцены

Игровой движок. Интерфейс редактора Unity: Scene, Game, Hierarchy, Inspector, Project. Игровой объект (GameObject), компонент, трансформация (Transform). Создание новой сцены и первых игровых объектов из примитивов.

Создание новой независимой сцены, размещение и настройка нескольких примитивов, знакомство с панелями редактора и сохранение сцены проекта.

1.2. Материалы, текстуры и префабы

Материал (Material), текстура, канал Albedo, нормали (Normal map). Библиотека материалов. Префаб (Prefab) как переиспользуемый шаблон объекта, вложенные префабы.

Создание и настройка нескольких материалов, применение текстур к объектам сцены, оформление уличных объектов (фонари, лавочки, клумбы) и сохранение их как префабов для многократного использования.

1.3. Творческий проект: создание уровня

Компоновка сцены (blockout), зонирование пространства, повторное использование префабов при застройке уровня.

Самостоятельная сборка небольшого городского уровня из примитивов и ранее созданных префабов: дороги, здания, деревья, освещение и центральная композиционная точка. Презентация готового уровня группе.

Модуль 2. Дизайн 3D-уровней

2.1. Введение в левел-дизайн и система Terrain

Левел-дизайн, ландшафт (Terrain), скульптинг рельефа, слои текстур ландшафта. Создание объекта Terrain, лепка возвышенностей и впадин, покраска слоями (трава, камень, песок), размещение водоёма.

2.2. Детализация ландшафта: деревья и трава

Детализация ландшафта (Terrain Details), кисти деревьев и травы, вариативность растительности. Добавление деревьев и травяного покрова на ландшафт, создание естественных зон и точек обзора уровня.

2.3. Световой дизайн и освещение

Источники света (Directional Light, Point Light, Spot Light), запечённое и реалтайм-освещение, туман (Fog), пробники отражений. Настройка времени суток, тёплого и холодного освещения, добавление точечных источников и атмосферного тумана.

2.4. Визуальные и звуковые эффекты

Система частиц (Particle System): огонь, дым, водопад, листва. Источник звука (Audio Source) и слушатель (Audio Listener), объёмный звук. Добавление визуальных эффектов и фоновых/точечных звуков на уровень.

Групповой проект: оформление совместно созданного уровня светом, эффектами и звуком, презентация результата.

Модуль 3. Программирование на C#

3.1. Основы программирования и работа с GitHub

Язык программирования C#, скрипт (Script) как компонент, консоль (Console). Система контроля версий, репозиторий, коммит. Создание первого скрипта, вывод отладочных сообщений, сохранение проекта в GitHub.

3.2. Хранение данных. Переменные

Переменная, типы данных (int, float, bool, string), публичные и приватные поля, отображение переменных в инспекторе. Создание объекта со скриптом, изменение значений переменных из инспектора и наблюдение результата в игре.

3.3. Условные конструкции

Условный оператор if / else, логические выражения, сравнение значений. Программирование объектов, поведение которых зависит от условия: запёртая дверь, изменение состояния объекта по условию.

3.4. Циклы

Циклы for и while, итерация, счётчик. Программное создание рядов и сеток объектов на сцене с помощью циклов, вывод результатов цикла в консоль.

3.5. Методы и функции

Метод (функция), параметры, возвращаемое значение, вызов метода. Вынесение повторяющегося кода в собственные методы, программирование простых игровых действий персонажей и объектов.

3.6. Практика: методы и функции

Закрепление темы через самостоятельную практику: применение методов для расчёта игровых показателей и организации взаимодействия объектов сцены.

3.7. Классы и взаимодействие объектов

Класс, объект как экземпляр класса, поля и методы класса, взаимодействие компонентов между собой (GetComponent). Создание собственных классов для описания игровых объектов и организации их взаимодействия на сцене.

3.8. Практика: классы и системы

Закрепление темы через практическую работу: проектирование небольшой системы из нескольких взаимодействующих классов (например, объекты-ловушки и триггеры).

3.9. Массивы, списки и UI

Массив, список (List), интерфейс пользователя (UI): Canvas, Text, элементы управления. Хранение наборов игровых данных и вывод их на экран через пользовательский интерфейс: счётчик очков, полоса здоровья.

3.10. Итоговый проект модуля. Часть 1

Проектирование и начало разработки небольшой игры на C#, объединяющей переменные, условия, циклы, методы, классы и пользовательский интерфейс.

3.11. Итоговый проект модуля. Часть 2

Завершение разработки, тестирование и отладка игры, презентация итогового проекта модуля перед группой с публикацией кода в GitHub.

Модуль 4. Анимация 3D-объектов

4.1. Введение в анимацию

Кадр анимации, частота кадров (FPS), ключевой кадр, окно Animation. Создание простых анимаций объектов (кнопка, ворота) и настройка их переключения через Animator Controller.

4.2. Анимация персонажа. Часть 1

Риггинг, скелет персонажа, аватар (Avatar), Animator Controller для гуманоидного персонажа. Импорт и настройка готовой 3D-модели персонажа, добавление анимаций ожидания, ходьбы и бега.

4.3. Контроллер движения персонажа

Компонент Rigidbody, векторы (Vector3), физическое перемещение объекта. Написание скрипта управления персонажем с клавиатуры с использованием физического движка.

4.4. Анимация персонажа. Часть 2: Blend Trees

Дерево смешивания анимаций (Blend Tree), параметры смешивания, плавные переходы между состояниями. Настройка плавного перехода между анимациями ожидания, ходьбы и бега в зависимости от скорости персонажа.

4.5. Завершение проекта. Кат-сцены и Timeline

Инструмент Timeline, виртуальная камера (Cinemachine), монтаж кат-сцены. Создание короткой вступительной кат-сцены для проекта модуля с движением камеры и анимацией персонажа. Презентация готовой сцены.

Модуль 5. Создание 2D-игры

5.1. Введение в 2D-разработку. Начало проекта

Спрайт (Sprite), отличия 2D-проекта от 3D, слои сортировки (Sorting Layer). Создание нового 2D-проекта, импорт спрайтов, настройка порядка отрисовки объектов сцены.

5.2. 2D-физика и движение персонажа

Компонент Rigidbody2D, Collider2D, нормализация движения, аниматор персонажа для вида сверху. Написание скрипта перемещения персонажа, настройка коллайдеров и анимации движения.

5.3. Tilemap и создание уровней

Тайловая карта (Tilemap), палитра тайлов (Tile Palette), несколько слоёв карты. Создание уровня из тайлов: базовый слой, декоративные детали, отдельные тематические зоны локации.

5.4. Игровая механика RPG

Паттерн Singleton для игрового менеджера, пользовательский интерфейс полосы здоровья, архитектура игровых систем. Программирование базовой RPG-механики: здоровье персонажа, интерфейс состояния, взаимодействие систем через единый менеджер.

5.5. Завершение 2D RPG-проекта

Импорт и подключение звуков, озвучивание боевых действий и окружения. Финальная доработка и полировка 2D RPG-проекта, добавление звукового сопровождения, презентация готовой игры.

Модуль 6. Создание 3D-игры

6.1. Побег из лаборатории. Создание прототипа уровня

Блокаут уровня (blockout), маршрут игрока, компоновка помещений. Совместное проектирование и сборка прототипа уровня «Побег из лаборатории»: коридоры, комнаты, размещение игрока и камеры.

6.2. Препятствия и левел-дизайн. Физика и анимация

Физические препятствия (трамплины, качели), анимационные препятствия (двери, ловушки), OnTriggerEnter. Создание препятствий на основе физики и анимации, программирование их взаимодействия с игроком.

6.3. Противники и финальная механика побега

Простой искусственный интеллект NPC, патрулирование по маршруту, сбор предметов (ключей), таймер обратного отсчёта. Расстановка противников, программирование сбора ключей, таймера и условия победы/поражения.

6.4. Создание GIF-анимации и запись геймплея

Запись видео в OBS Studio, монтаж в видеоредакторе, конвертация видео в GIF. Запись фрагмента прохождения готового уровня, монтаж короткого ролика и подготовка GIF-презентации итогового проекта курса.

1.5. Планируемые результаты

Планируемый результат	Способ достижения	Критерий достижения образовательного результата
Личностные результаты обучения		
Освоение социальной роли обучающегося и формирование личностного смысла учения.	Демонстрация связи между способностью реализовать интересный игровой проект и наличием соответствующих знаний.	За отведённое время ученик пытается не только выполнить базовое задание, но и добавить собственные улучшения к проекту.
Развитие навыков сотрудничества со взрослыми и сверстниками, умение находить выход из спорных ситуаций.	Использование групповых проектов как способа показать преимущества командной работы.	Ученик не боится просить помощи и сам пытается помогать одноклассникам и преподавателю.
Овладение начальными навыками адаптации в динамично изменяющемся и развивающемся мире технологий.	Обсуждение примеров современных игр и технологий, демонстрирующих необходимость развития умения учиться.	При работе над проектом ученик сначала пытается найти решение самостоятельно или в документации Unity, и только затем обращается к преподавателю.
Метапредметные навыки		
Развитие и формирование учебных действий.	Создание условий для участия в обсуждении игровых и технических решений, продуктивное взаимодействие со сверстниками при разработке проекта.	Ученик легко общается, не боится просить помощь или оказывать её другим, способен кооперироваться для достижения общей цели.
Умение презентовать свою работу.	Презентация итоговых проектов модулей, обратная связь от преподавателя и других учеников.	Ученик во время презентации своего проекта уверенно удерживает внимание аудитории.
Развитие критического и алгоритмического мышления учеников.	Выполнение логических операций при программировании: анализ задачи, декомпозиция, составление алгоритма решения.	Ученик строит логическую цепочку рассуждений и самостоятельно планирует свою работу над задачей.
Развитие творческих и пространственных способностей учеников.	Планирование и реализация уровня или сцены по собственному замыслу с использованием инструментов Unity.	Ученик самостоятельно придумывает, обсуждает, планирует и реализует собственный игровой проект.

Планируемый результат	Способ достижения	Критерий достижения образовательного результата
Предметные навыки		
Усвоение и применение базовых навыков работы с ПК и специализированным ПО.	Повторение рутинных действий в редакторе Unity на каждом занятии.	Ученик самостоятельно ориентируется в интерфейсе Unity, открывает и сохраняет сцены и проекты.
Сформированность представления об архитектуре игрового проекта и работе игрового движка.	Выполнение учебных проектов, работа со сценами, объектами и компонентами.	Ученик понимает общие принципы устройства игрового проекта: сцена, объект, компонент, префаб.
Сформированность навыков левел-дизайна и работы с 3D-окружением.	Выполнение учебных проектов по созданию и оформлению 3D-уровней.	Ученик умеет самостоятельно проектировать и оформлять игровой уровень с использованием Terrain, освещения и эффектов.
Сформированность навыков программирования на языке C#.	Выполнение учебных проектов и итоговых модульных заданий по программированию.	Ученик умеет самостоятельно писать скрипты с переменными, условиями, циклами, методами и классами.
Умение создавать и настраивать анимацию игровых объектов и персонажей.	Выполнение учебных проектов по анимации.	Ученик самостоятельно создаёт анимационные клипы и настраивает переходы между ними через Animator Controller.
Умение разрабатывать и завершать игровой проект в 2D и 3D.	Разработка итоговых проектов модулей 5 и 6.	Ученик способен довести игровой проект от прототипа до готовой, презентуемой версии.

В течение курса ученики создадут не менее 6 индивидуальных проектов и не менее 2 групповых проектов.

№ п/п	Неделя	Форма занятия	Всего часов	Название темы	Место проведения	Форма контроля
1	1	интерактивная	2	Знакомство с Unity и создание первой сцены	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
2	2	интерактивная	2	Материалы, текстуры и префабы	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
3	3	интерактивная	2	Творческий проект: создание уровня	Компьютерный класс ТУ УГМК	Презентация проекта
4	4	интерактивная	2	Введение в левел-дизайн и система Terrain	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
5	5	интерактивная	2	Детализация ландшафта: деревья и трава	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
6	6	интерактивная	2	Световой дизайн и освещение	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
7	7	интерактивная	2	Визуальные и звуковые эффекты	Компьютерный класс ТУ УГМК	Презентация проекта
8	8	интерактивная	2	Основы программирования и работа с GitHub	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
9	9	интерактивная	2	Хранение данных. Переменные	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
10	10	интерактивная	2	Условные конструкции	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
11	11	интерактивная	2	Циклы	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
12	12	интерактивная	2	Методы и функции	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
13	13	интерактивная	2	Практика: методы и функции	Компьютерный класс ТУ УГМК	Практическая работа с

№ п/п	Неделя	Форма занятия	Всего часов	Название темы	Место проведения	Форма контроля
						проверкой преподавателем
14	14	интерактивная	2	Классы и взаимодействие объектов	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
15	15	интерактивная	2	Практика: классы и системы	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
16	16	интерактивная	2	Массивы, списки и UI	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
17	17	интерактивная	2	Итоговый проект модуля. Часть 1	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
18	18	интерактивная	2	Итоговый проект модуля. Часть 2	Компьютерный класс ТУ УГМК	Презентация проекта
19	19	интерактивная	2	Введение в анимацию	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
20	20	интерактивная	2	Анимация персонажа. Часть 1	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
21	21	интерактивная	2	Контроллер движения персонажа	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
22	22	интерактивная	2	Анимация персонажа. Часть 2: Blend Trees	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
23	23	интерактивная	2	Завершение проекта. Кат- сцены и Timeline	Компьютерный класс ТУ УГМК	Презентация проекта
24	24	интерактивная	2	Введение в 2D- разработку. Начало проекта	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
25	25	интерактивная	2	2D-физика и движение персонажа	Компьютерный класс ТУ УГМК	Практическая работа с

№ п/п	Неделя	Форма занятия	Всего часов	Название темы	Место проведения	Форма контроля
						проверкой преподавателем
26	26	интерактивная	2	Tilemap и создание уровней	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
27	27	интерактивная	2	Игровая механика RPG	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
28	28	интерактивная	2	Завершение 2D RPG-проекта	Компьютерный класс ТУ УГМК	Презентация проекта
29	29	интерактивная	2	Побег из лаборатории. Создание прототипа уровня	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
30	30	интерактивная	2	Препятствия и левел-дизайн. Физика и анимация	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
31	31	интерактивная	2	Противники и финальная механика побега	Компьютерный класс ТУ УГМК	Практическая работа с проверкой преподавателем
32	32	интерактивная	2	Создание GIF-анимации и запись геймплея	Компьютерный класс ТУ УГМК	Презентация проекта
			64	Всего часов:		

1.6. Календарный учебный график

Количество учебных недель по программе – 32 недели

Количество учебных дней – 32

Форма занятий интерактивная, включает в себя объяснение, практическую работу в редакторе Unity, разбор кода, демонстрацию результатов работы и рефлексии.

2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

2.1. Условия реализации программы

Материально-технические условия реализации

- учебный кабинет (компьютерный класс) с местами для 12 учеников;
- компьютеры для учеников (1 ученик — 1 компьютер, конфигурация с поддержкой Unity 2021.3 LTS) и компьютер для преподавателя;
- проектор, подключённый к компьютеру преподавателя, выводящий изображение на экран;
- установленное программное обеспечение: Unity Hub, Unity 2021.3 LTS и выше, редактор кода (Visual Studio / Visual Studio Code), Git;
- учётные записи учеников на GitHub для сохранения версий учебных проектов;
- наушники или колонки для работы со звуковыми эффектами;
- доска или флипчарт, мел или маркер.

Учебно-методическое и информационное обеспечение программы:

- поурочные методические сценарии к занятиям;
- тематические презентации (PowerPoint и веб-версии) к каждому занятию;
- комплект скриншотов и иллюстраций, иллюстрирующих пошаговое выполнение практических заданий;
- заготовки проектов и ассеты, необходимые для практической части занятий;
- итоговые проекты предыдущих потоков как примеры для демонстрации ученикам.

Кадровое обеспечение: Педагогическая деятельность осуществляется лицами, имеющими среднее профессиональное или высшее образование (в том числе по направлениям, соответствующим направлению программы) и отвечающим квалификационным требованиям, указанным в квалификационных справочниках, и (или) профессиональных стандартах.

2.2. Форма аттестации/контроля и оценочные материалы

Текущий контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов на нескольких уровнях:

- на каждом занятии: опрос по теории, выполнение практического задания в редакторе Unity с проверкой преподавателем, взаимоконтроль учеников в парах, самоконтроль ученика;
- в конце каждого модуля: презентация итогового проекта модуля и его обсуждение с преподавателем и группой.

Для контроля сформированности результатов освоения программы используются следующие механики:

- проверка работающего результата непосредственно в редакторе Unity (Play Mode);
- проверка кода скриптов на соответствие поставленной задаче;
- устный опрос по терминологии и принципам работы инструментов движка;
- презентация и защита индивидуальных и групповых проектов;
- итоговая презентация курса — демонстрация уровня «Побег из лаборатории» и подготовленного по нему GIF-ролика.

3. СПИСОК ЛИТЕРАТУРЫ

Список литературы

1. Официальная документация Unity. — Unity Technologies, <https://docs.unity3d.com>.
2. Джон Скит. С# для профессионалов. — Питер, 2021.
3. Троелсен Э., Джепикс Ф. Язык программирования С# и платформа .NET. — Диалектика, 2021.
4. Хокинг Д. Unity в действии. Мультиплатформенная разработка на С#. — Питер, 2020.
5. Джефффри Ричардсон. Изучаем программирование на С#. — БХВ-Петербург, 2019.
6. Кайл Стил. С# для детей. Самоучитель по программированию. — Манн, Иванов и Фербер (МИФ), 2021.
7. Флеминг Дж. Разработка 2D-игр на Unity. — ДМК Пресс, 2020.
8. Гриффитс Дэвид. Изучаем программирование на примере создания игр. — Питер, 2018.
9. Э. Танненбаум, Д. Уэзеролл. Компьютерные сети. Пятое издание. — Питер, 2012.
10. Л.Л. Босова, А.Ю. Босова. Информатика. Учебники для 6–8 классов. — БИНОМ. Лаборатория Знаний.
11. И.Г. Семакин, Л.А. Залогова, С.В. Русаков. Информатика. Учебник для 7 класса. — БИНОМ. Лаборатория Знаний.
12. Nystrom R. Game Programming Patterns. — Genever Benning, 2014.