


ИНДИВИДУАЛЬНЫЙ ПРЕДПРИНИМАТЕЛЬ

УШАКОВ ВИТАЛИЙ АНАТОЛЬЕВИЧ

“УТВЕРЖДАЮ”

Индивидуальный предприниматель

Ушаков Виталий Анатольевич

 / Ушаков В.А.

Приказ № 9 от 29.05.2026г.

б/п

Дополнительное образование.

Дополнительное образование детей и взрослых.

Дополнительная общеобразовательная общеразвивающая

программа технической направленности

“Теория разработки игр: первые шаги юного игродела!”

Направленность: техническая

Уровень: базовый

Срок реализации: 12 недель (44 часа)

Автор-составитель:

Ушаков В.А.

г. Благовещенск

2026г

СОДЕРЖАНИЕ

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

1.1.	Пояснительная записка (характеристика, нормативная база, актуальность)	2
1.2.	Цель и задачи программы	6
1.3.	Содержание программы	8
1.3.1.	Учебный план	8
1.3.2.	Учебно-тематический план	8
1.3.3.	Содержание учебно-тематического плана	10
1.4.	Планируемые результаты	25

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1.	Календарный учебный график	26
2.2.	Условия реализации программы	28
2.2.1.	Материально-техническое обеспечение	29
2.2.2.	Кадровое обеспечение	29
2.3.	Формы контроля и аттестации	30
2.4.	Оценочные материалы	31
2.5.	Методические материалы	36
2.6.	Организация учета и хранения результатов образовательного процесса	36
2.7.	Информационное обеспечение	37

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

1.1. Пояснительная записка (общая характеристика, нормативная база, актуальность)

Направленность программы – техническая.

Программа направлена на удовлетворение индивидуальных потребностей обучающихся в интеллектуальном развитии, формированию технологической грамотности и современных компетенций обучающихся в области технических и естественных наук.

Уровень сложности программы – базовый.

Нормативно-правовое обеспечение разработки и реализации программы:

1. Федеральные законы:

ФЗ №273 «Об образовании в Российской Федерации» (ст. 2, 12, 75):

– Определяет понятие «дополнительное образование», требования к программам

– Устанавливает порядок разработки и утверждения программ

– Регламентирует особенности дополнительного образования детей и взрослых.

ФЗ №152 «О персональных данных»:

— Обязательность защиты персональных данных обучающихся (при наличии их обработки в программе).

2. Подзаконные акты и приказы:

Приказ Минпросвещения № 629 от 27.07.2022 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»:

– Требования к структуре программы (целевой, содержательный, организационный разделы).

– Правила учета возрастных и индивидуальных особенностей обучающихся.

СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»:

– Требования к продолжительности занятий, оборудованию, освещению, работе за компьютером (для детей).

Концепция развития дополнительного образования детей (Распоряжение Правительства РФ №678-р от 31.03.2022):

– Акцент на развитие технической и IT-направленности программ.

– Постановление Правительства Российской Федерации от 11.10.2023 № 1678 "Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ".

3. Методические рекомендации:

Письмо Минпросвещения №06-1844 от 18.12.2020 «О направлении методических рекомендаций»:

– Рекомендации по проектированию дополнительных общеразвивающих программ.

Примерные программы технической направленности (опубликованы на портале Минпросвещения или региональных ресурсах).

4. Распорядительные документы организации:

Локальные документы по организации образовательной деятельности.

Приказ индивидуального предпринимателя об утверждении дополнительной общеобразовательной общеразвивающей программы технической направленности.

Актуальность программы

Современный период развития общества характеризуется сильным влиянием на него цифровых технологий. **Цифровые технологии и Интернет проникли и прочно закрепились во всех сферах деятельности современного человека и являются на сегодняшний день основным источником получения, обработки, хранения и защиты больших объемов разнообразной информации, средств коммуникации, самореализации, интеллектуального развития, социализации и получения образования**, а также стратегически важным инструментом для построения бизнеса.

Современные люди знакомы в основном с игровыми компьютерными программами и используют компьютерную технику преимущественно для развлечений, а познавательные, в частности образовательные, мотивы работы с компьютером находятся на низком уровне. В связи с этим компьютерные технологии должны стать отдельным направлением в рамках целостного образовательного процесса, способствующим повышению эффективности образования и самореализации.

Школьное образование в сфере цифровых технологий на сегодняшний день ограничивается в основном только уроками информатики, в рамках которых осуществляется обучение детей общей компьютерной грамотности, работе в офисных программах, навыкам поиска информации в сети Интернет и т.п. Более глубоких знаний и специфических навыков, которые могут быть необходимыми в дальнейшей жизни, школьная программа не предусматривает. Таким образом, можно говорить о том, что в России на сегодняшний день отсутствует комплексный, четко выстроенный образовательный процесс в сфере цифровых технологий.

В данных условиях дополнительная общеобразовательная общеразвивающая программа «Теория разработки игр: первые шаги юного игродела!»» (далее – Программа) выполняет функцию компенсации

недостатков школьного обучения, выступает источником получения учащимися современных и востребованных цифровых компетенций, является средством включения в техническое творчество, что способствует развитию логического мышления, навыка решения задач, реализации творческих способностей и ранней профориентации.

Отличительные особенности, новизна и педагогическая целесообразность программы:

1. Просто и наглядно: Изучение начинается с базовых понятий программирования на примерах создания несложных проектов в среде Scratch, что обеспечивает быстрое усвоение материала и получение первых результатов.
2. Практико-ориентированный подход и самостоятельность: Учащиеся выполняют индивидуальные практические работы проекты самостоятельно, закрепляя теоретические знания, полученные на онлайн-занятиях.
3. Творческий подход: Проектная деятельность стимулирует развитие фантазии и креативности. Обучающиеся в ходе выполнения домашних заданий самостоятельно экспериментируют с графикой, звуком и анимацией, создавая уникальные цифровые продукты.
4. Педагогическая эффективность: Благодаря современным подходам к обучению, развивается интерес к информационным технологиям и повышается уровень мотивации учащихся к дальнейшему саморазвитию в сфере ИТ.
5. Профессиональная ориентация: Формирование представлений о возможных направлениях профессионального роста в будущем, связанных с разработкой цифровых продуктов и информационных технологий.
6. Современные компетенции: Полученные навыки позволяют успешно адаптироваться к реалиям современного информационного общества и активно пользоваться цифровыми технологиями в повседневных ситуациях.

Новизна Программы обусловлена тем, что данная Программа комбинирует теорию и самостоятельное творчество детей. Хотя практическая часть выполняется самостоятельно, занятия содержат подробные рекомендации и руководства, которые способствуют осознанному выполнению заданий. Вместо традиционных упражнений, учащиеся погружаются в создание полноценных мини-игр и историй, формируя уникальный творческий опыт.

Педагогическая целесообразность Программы заключается в том, что Программа посредством позволяет обучающимся приобрести современные цифровые компетенции: освоить навыки работы с ПК и познакомиться с онлайн-платформой, которая дает возможность понять базовые принципы программирования для более успешного дальнейшего изучения языков программирования; освоить навыки творческой и проектной работы; развить профессиональные (hard skills) и надпрофессиональные компетенции (soft skills).

Адресат Программы

Возраст обучающихся – программа разработана для детей и подростков в возрасте от 10 до 14 лет.

К освоению программы допускаются любые лица без предъявления требований к уровню образования.

Программа будет интересна и полезна тем, кто проявляют интерес к созданию интерактивных игр и проектов. Данная образовательная программа направлена на введение учащихся в сферу информационных технологий посредством изучения основ программирования на примере визуальной событийно-ориентированной среды Scratch. Несмотря на то, что основной акцент делается на Scratch как наиболее доступной и наглядной среде для начинающих, целью курса является формирование общего представления о принципах и процессах программирования, развитии навыков критического и аналитического мышления, готовности применять полученные знания в создании цифровых продуктов различного назначения.

Уровень освоения дополнительной общеобразовательной программы – дополнительной общеразвивающей программы – базовый. Предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка, гарантированно обеспечивают трансляцию общей и целостной картины в рамках содержательно- тематического направления программы.

Особенности набора: Программа подходит для тех, кто уверенно владеет базовыми навыками работы с мышкой и клавиатурой.

Объем и срок освоения программы.

Объем программы составляет 44 академических (учебных) часа, из них:

– теоретическое занятие (видеоконференция) – 14 часа;

– групповое консультирование (видеоконференция) – 30 часа

Нормативный срок освоения программы – 44 часа

Особенности организации образовательного процесса: форма обучения – исключительно дистанционная с применением электронного обучения и дистанционных образовательных технологий.

Дистанционные образовательные технологии, используемые для организации учебного процесса:

Занятия проходят с использованием электронного курса «Теория разработки игр: первые шаги юного игродела!», размещенного на платформе Skillspase (ООО «СКИЛЛСПЕЙС», ИНН 9701194047) в формате видеоконференций (вебинары) с

использованием видео-конференц-связи «Яндекс Телемост» (ООО «ЯНДЕКС», ИНН 7736207543). Данный формат организации учебного процесса обеспечивает учет контингента обучающихся.

Язык реализации программы – русский.

Количество обучающихся (наполнение групп) – до 10 человек.

Организационные формы обучения: Занятия проводятся всем составом в группах одного возраста или разновозрастных группах.

Режим занятий:

Для всех видов образовательной (учебной) деятельности академический час устанавливается продолжительностью 45 минут.

Объем недельной учебной нагрузки обучающихся составляет не более 4 академических часов в неделю. Количество учебных дней по образовательной программе – 2 раза в неделю. Объем учебной нагрузки в учебный день – 2 часа. Перерывы между занятиями в рамках учебного дня составляют не менее 15 минут для проведения физкультминуток, гимнастики для глаз, обеспечивается контроль за осанкой, в том числе, во время использования электронных средств обучения (ЭСО).

Продолжительность непрерывного использования экрана (электронного способа обучения) составляет 7 минут, далее педагог дополнительного образования предлагает обучающимся самостоятельно выполнить задание на бумаге или сформулировать ответ на вопрос (на выполнение задания дается 7 минут. После выполнения заданий обучающиеся возвращаются к экрану для подведения итогов занятия. Практические занятия проводятся в форме выполнения игровых заданий, бесед.

1.2. Цель и задачи программы.

Цель – познакомить детей 10–14 лет с основами разработки игр через творчество, логику и самостоятельную работу, чтобы они смогли создать свою первую игру. Курс направлен на развитие навыков геймдизайна, программирования, графики и критического мышления, а также на вдохновение к дальнейшему изучению игровой индустрии.

Задачи Программы

Обучающие:

- сформировать у обучающихся навыки работы с ПК и применения полученных знаний в практической деятельности;
- сформировать представление об универсальных возможностях использования цифровых технологий и программного обеспечения как средств обучения и развития;

- изучить возможности использования цифровых продуктов как инструментов для практической деятельности.

Развивающие:

- сформировать навыки поиска, сбора, анализа, организации представления и передачи, а также хранения и защиты информации в открытом информационном обществе и окружающей реальности;

- способствовать формированию у обучающихся «гибких» навыков — надпрофессиональных компетенций и универсальных знаний, востребованных в любой сфере деятельности человека — и профессиональной, и социальной: цифровая грамотность, стратегическое мышление, системное мышление, критическое мышление, навыки коммуникации, самоорганизации и самообучения

- сформировать умение ориентироваться в информационном пространстве, собирать, анализировать и обобщать информацию, делать выводы;

- сформировать умение решать принципиально новые задачи, порожденные привнесенным цифровыми технологиями новым подходом к анализу окружающей деятельности.

Воспитательные:

- сформировать у обучающихся правильный методологический подход к познавательной и практической деятельности;

- развить коммуникативные компетенции — навыки межличностного взаимодействия, сотрудничества и работы в команде;

- воспитать в обучающихся такие качества, как усидчивость, целеустремленность, умение добиваться поставленных задач, трудолюбие, инициативность и настойчивость в преодолении трудностей;

- сформировать умения планировать свою деятельность, критически оценивать результаты своей работы, готовности исправлять свои ошибки.

1.3. Содержание программы

1.3.1. Учебный план

№	Наименование раздела	Количество часов			Форма промежуточной аттестации
		Всего	Теория	Консультация	
1	Раздел 1. Основы разработки игр на примере среды Scratch	6	2	4	-
2	Раздел 2. Первая игра «Пенальти»	6	2	4	-
3	Раздел 3. Работа с фонами и игрой «Лабиринт»	6	2	4	-
4	Раздел 4. Применение событий и игра «Битва магов»	6	2	4	-
5	Раздел 5. Использование переменных и клонов (игра «Динозавр»)	6	2	4	-
6	Раздел 6. Создание сложных игровых проектов	6	2	4	-
Итоговая аттестация — Открытая презентация проекта (игр Flappy Bird и Space Battle, созданных самостоятельно в течение учебного периода)		8	2	6	Открытая презентация проекта, собеседование
Итого		44	14	30	

1.3.2. Учебно-тематический план

№	Наименование раздела и темы	Количество часов			Форма промежуточной аттестации
		Всего	Теория	Консультация	
1	<i>Раздел 1. Основы разработки игр на примере среды Scratch</i>	6	2	4	-
	Тема 1.1. Знакомство с программой, изучение базовых понятий. Освоение инструментов программы. Знакомство с понятиями угол поворота, ось координат, случайное число из диапазона, команды блока «Движение».	3	1	2	-

	Тема 1.2. Знакомство с понятиями цикл, костюм, звук, событие, эффект, способ вращения.	3	1	2	-
2	Раздел 2. Первая игра «Пенальти»	6	2	4	-
	Тема 2.1. Знакомство с понятиями угол поворота, масштабирование. Закрепление пройденного материала.	6	2	4	-
3	Раздел 3. Работа с фонами и игрой «Лабиринт»	6	2	4	-
	Тема 3.1. Создание и редактирование фонов. Изучение понятий условие, сенсор, координаты, таймер.	6	2	4	-
4	Раздел 4. Применение событий и игра «Битва магов»	6	2	4	-
	Тема 4.1. Изучение понятий векторная и растровая графика, использование событий, цикла, эффектов.	6	2	4	-
5	Раздел 5. Использование переменных и клонов (игра «Динозавр»)	6	2	4	-
	Тема 5.1. Изучение понятий переменных и клонов.	6	2	4	-
6	Раздел 6. Создание сложных игровых проектов	6	2	4	-
	Тема 6.1. Разработка многопользовательских игр	6	2	4	-
Итоговая аттестация — Открытая презентация проекта (игра Flappy Bird и Space Battle, созданных самостоятельно в течение учебного периода)		8	2	6	Открытая презентация проекта, устное собеседование
Итого		44	14	30	

1.3.3. Содержание учебно-тематического плана

Онлайн-курс ориентирован на начинающих разработчиков игр, желающих освоить основы программирования на примере среды Scratch. Курс состоит из шести основных разделов, каждый из которых включает теоретические занятия, самостоятельную практическую работу и консультирование по результатам практики.

Раздел 1. Основы разработки игр на примере среды Scratch

Тема 1.1. Знакомство с программой, изучение базовых понятий. Освоение

инструментов программы. Знакомство с понятиями угол поворота, ось координат, случайное число из диапазона, команды блока «Движение».

Теория. Введение в программу на примере среды Scratch. Знакомство с интерфейсом и основными возможностями программы. Изучение основных понятий программирования, таких как переменные, операторы, условия, циклы и массивы, на примере Scratch. Освоение инструментов для создания анимации и игр на примере Scratch. Знакомство с понятием «угол поворота» и его использование при создании анимации и игр. Изучение понятия «ось координат» и его применение при создании проектов на примере Scratch. Знакомство с генерацией случайных чисел из заданного диапазона и использование их в проектах. Изучение команд блока «Движение» и их применение для создания анимаций и игр на примере Scratch.

Задание на внедрение (самостоятельная работа): Сбор скриптов из блоков, создание и управление спрайтами.

Вопросы для самоконтроля:

1. Что такое среда программирования на примере программы Scratch?
2. Какие инструменты есть в панели инструментов на примере программы Scratch?
3. Кто такой спрайт и для чего он нужен?
4. Что означают слова "угол поворота" на примере программы Scratch?
5. Как называются линии, вдоль которых движется объект на экране?
6. Что такое случайное число и зачем оно нужно в играх?
7. Какие команды относятся к блоку "Движение"?
8. Как заставить героя двигаться вперед?
9. Что означает фраза "изменить положение по X и Y"?
10. Чем полезны инструкции, помогающие повернуть объект?

Вопросы для контроля (устное собеседование педагога по дополнительному образованию или обсуждение проектной работы в чат-группе с использованием видеоконференцсвязи «Яндекс Телемост»):

1. Расскажи простыми словами, что такое языки программирования?
2. Что интересного можно нарисовать в графическом редакторе на примере программы Scratch?
3. Почему герои на примере программы Scratch умеют поворачиваться и куда смотреть?
4. Представь, что твой герой хочет прыгнуть вправо — какой командой воспользуешься?
5. Что получится, если поставить координаты (0, 0)? Это какая-то особая точка?
6. Зачем нужно случайно выбирать числа в игре?
7. Тебе легче управлять героем мышкой или стрелками клавиатуры?

8. Если твоя игра стала слишком простой, как усложнить её с помощью команд движения?
9. Придумай задание для своего героя, используя повороты и прыжки.
10. Скажи, какое самое любимое занятие твоего героя в игре?
11. Каковы основные элементы интерфейса на примере программы Scratch? Что такое спрайт и как его создать?
12. Как добавить скрипт для спрайта?
13. Какие блоки используются для управления движением спрайта? Как добавить звук в программу?
14. Что такое костюмы для спрайта и как их использовать? Как создать переменные и использовать их в скриптах? Какие виды событий доступны на примере программы Scratch?
15. В чем разница между безусловными и условными операторами? Как использовать циклы на примере программы Scratch?

Тема 1.2. Знакомство с понятиями цикл, костюм, звук, событие, эффект, способ вращения.

Теория. Создание простой анимации. Работа с основными блоками на примере программы Scratch. Использование костюма для изменения внешнего вида спрайта. Программирование звука на примере Scratch. Разработка простого проекта с использованием звука. Применение события для запуска анимации. Создание более сложных проектов, использующих события. Применение различных эффектов для улучшения качества проекта. Изучение способов вращения спрайтов на примере Scratch.

Задание на внедрение (самостоятельная работа): Воспроизведение звука, применение команд «способ вращения», «повторять всегда», «повторять 10 раз», «говорить» и «думать», использование костюмов и эффектов.

Вопросы для самоконтроля:

1. Что такое цикл в программировании и как он используется на примере программы Scratch?
2. Что такое костюм на примере программы Scratch и зачем он нужен?
3. Что такое звук на примере программы Scratch и как его добавлять?
4. Что такое событие на примере программы Scratch?
5. Что такое эффект на примере программы Scratch и какие эффекты бывают?
6. Какие существуют способы вращения на примере программы Scratch?
7. Как выглядит кнопка запуска сценария на примере программы Scratch?
8. Какие костюмы у твоего персонажа в проекте?
9. Какие звуки используешь в своем проекте?

10. Какие эффекты применял в проекте?

Вопросы для контроля (устное собеседование педагога по дополнительному образованию или обсуждение проектной работы в чат-группе с использованием видеоконференцсвязи «Яндекс Телемост»):

1. Что такое цикл? Чем отличается повторяющийся цикл от одиночного действия?

2. Почему костюмов должно быть много у одного персонажа? Как создать и использовать костюм для спрайта на примере программы Scratch? Какие блоки используются для работы с костюмами, звуковыми и визуальными эффектами на примере программы Scratch?

3. Каким образом можно добавить звук в проект на примере программы Scratch? Как часто в твоей игре звучит музыка?

4. Что такое событие на примере программы Scratch и как оно используется при программировании? Откуда берётся событие в игре, которое запускает выполнение сценария?

5. Можем ли мы изменять внешний вид нашего персонажа прямо во время игры?

6. Зачем нам вообще нужны визуальные эффекты?

7. Чем различаются виды вращений "не вращается", "обычное вращение" и "левый-правый поворот"?

8. Всегда ли обязательно включать музыку в игру?

9. Как быстрее поменять костюм персонажа?

10. Как отличить настоящий звук от эффекта звука?

11. Какие эффекты можно применить к спрайтам и блокам на примере программы Scratch?

12. Какие способы вращения спрайтов доступны на примере программы Scratch и как они используются?

13. Как можно проверить работу циклов, звуков и эффектов в проектах на примере программы Scratch?

14. Как использовать переменные для работы со звуковыми эффектами в Scratch?

15. Как создать спрайт с анимацией на основе событий на примере программы Scratch?

Раздел 2. Первая игра «Пенальти»

Тема 2.1. Знакомство с понятиями угол поворота, масштабирование.
Закрепление пройденного материала.

Теория. Работа с основными понятиями программы: спрайты, блоки, скрипты.

Изучение понятия «угол поворота», возможности его изменения и применения на примере программы Scratch. Работа с инструментами масштабирования на примере программы Scratch: изменение размеров спрайтов, добавление эффектов масштабирования. Создание простых анимаций с использованием поворота и масштабирования объектов. Разработка игр с использованием поворота и масштабирования спрайтов. Применение скриптов для управления углом поворота и масштабом спрайтов.

Задание на внедрение (самостоятельная работа): Создание игры «Пенальти» с возможностью ее модификации.

Задание для проектной работы

Создание игры «Пенальти», в которой нападающим будет кот, пинающий футбольный мяч, вратарем будет пингвин, ловящий мяч.

Цель задания:

Изучение основ программирования на примере программы Scratch, работа с графикой и анимацией, создание интерактивных игр.

Содержание задания:

Изучение основных принципов программирования на примере программы Scratch (переменные, операторы, циклы, условные конструкции).

Создание спрайтов (нападающего и вратаря) и фонов для игры. Разработка анимации для каждого спрайта.

Написание программы для управления каждым из спрайтов. Создание системы подсчета очков.

Настройка звукового сопровождения. Тестирование и отладка программы.

Вопросы для самоконтроля:

1. Какие основные принципы программирования на примере программы Scratch вы изучили в ходе выполнения задания?
2. Что представляет собой спрайт и как он используется в игре?
3. Объясните разницу между оператором если и циклом повторять.
4. Какие команды необходимы для перемещения спрайта по сцене?
5. Какие звуковые эффекты использованы в вашей игре и зачем?
6. Как устроена ваша система подсчета очков?
7. Сколько раз пришлось изменить сценарий игры, пока программа заработала правильно?
8. Какие приемы помогли сделать вашего кота реалистичным нападающим?
9. Укажите три этапа тестирования вашей программы и поясните каждый этап.
10. Как вы организовали управление игроком в игре?

Вопросы для контроля (устное собеседование педагога дополнительного образования или обсуждение проектной работы в чат-группе с использованием видеоконференцсвязи «Яндекс Телемост»):

1. Чем отличается поведение спрайта-кота от поведения спрайта-пингвина?
2. Какие события запускают начало нового раунда игры?
3. Какие изменения произошли в коде, когда была обнаружена ошибка?
4. Могли бы вы реализовать дополнительный режим игры с двумя игроками?
Какие шаги потребовались бы для этого?
5. Что делает игру интересной и увлекательной для игрока?
6. Назовите одну проблему, которую вы столкнулись при написании скриптов для игры, и расскажите, как вы её решили.
7. Какие сценарии проигрыша предусмотрены в вашей игре?
8. Как организован цикл движений ваших персонажей?
9. Каким образом проверяется попадание мяча в ворота?
10. Чем вы научились от этого проекта и какие навыки теперь пригодятся в будущих проектах?

Дополнительные вопросы для самоконтроля и контроля:

1. Что такое угол поворота на примере программы Scratch? Как изменить угол поворота спрайта?
2. Что такое масштабирование на примере программы Scratch?
3. Как можно изменить масштаб спрайта на примере программы Scratch?
4. Какие блоки используются для изменения угла поворота и масштаба спрайта?
5. В каких ситуациях может потребоваться изменение угла поворота или масштаба спрайтов?
6. Перечислите основные свойства спрайтов на примере программы Scratch и дайте определение каждому из них.
7. Назовите основные команды, используемые при работе со спрайтами на примере программы Scratch.
8. Приведите примеры использования масштабирования и поворота спрайтов при создании игр и анимации на примере программы Scratch.
9. Каким образом можно проверить правильность работы кода, изменяющего угол поворота или масштаб спрайтов?

Раздел 3. Работа с фонами и игрой «Лабиринт»

Тема 3.1. Создание и редактирование фонов. Изучение понятий условие, сенсор, координаты, таймер.

Теория. Работа с фоном на примере программы Scratch: создание и редактирование различных фонов для проектов. Изучение понятия «условие» и его применения в проектах на примере программы Scratch: создание условий для запуска и остановки анимации, изменение поведения спрайтов в зависимости от условий. Знакомство с понятием «сенсор» на примере программы Scratch, его возможности и

применение в проектах. Работа с сенсорами касания, звука, движения. Изучение координат на примере программы Scratch, применение их для определения местоположения спрайтов на экране и взаимодействия с ними. Работа с таймером на примере программы Scratch: использование таймера для создания временных задержек в скриптах и анимациях, а также для создания интерактивности в проектах. Разработка простых игр и анимаций с использованием условий, сенсоров, координат и таймера.

Задание на внедрение (самостоятельная работа): Создание игры «Лабиринт».

Задание для проектной работы

Создание игры «Лабиринт», в которой будет два персонажа – Гига и Нано. Цель: Изучение основ программирования на примере программы Scratch, разработка игр с использованием спрайтов и анимации, освоение работы с логическими операторами и циклами.

Задачи:

Изучение основных понятий программирования на примере программы Scratch (переменные, блоки, операторы, условия, циклы).

Создание спрайтов для персонажей и фона для игры. Разработка анимации для спрайтов персонажей.

Написание программы для управления персонажами с использованием логических операторов и циклов.

Добавление препятствий и ловушек в лабиринт.

Создание системы подсчета баллов и определения победителей. Настройка звукового сопровождения и музыкального фона.

Тестирование и исправление ошибок в программе.

Вопросы для самоконтроля:

1. Какие новые понятия программирования на примере программы Scratch вы изучили?

2. Как создавались персонажи (спрайты) для вашей игры?

3. Опишите принцип работы цикла и условий в вашем проекте.

4. Что такое переменная и зачем она используется в игре?

5. Объясните механизм добавления анимаций вашим персонажам.

6. Почему важно тестировать игру и находить ошибки? Приведите пример такой ситуации.

7. Расскажите, какую роль играет музыкальное сопровождение в игровом процессе.

8. Как реализована система подсчёта очков и победы игроков?

9. Какие трудности возникли при создании лабиринта и каким способом были решены?

10. Что бы вы сделали иначе, если бы снова начали этот проект?

Вопросы для контроля (устное собеседование педагога или обсуждение проектной работы в чат-группе с использованием видеоконференцсвязи «Яндекс Телемост»):

1. Чем отличаются алгоритмы управления персонажем Гигой и Наной? Есть ли различия в поведении героев?
2. Какой блок на примере программы Scratch использовался чаще всего в вашем коде и почему именно он?
3. Если один игрок сталкивается с препятствием, меняется ли счёт другого игрока? Как это отражено в коде?
4. Можно ли считать вашу программу оптимальной с точки зрения скорости выполнения действий? Предложите улучшения.
5. Как организован цикл движения персонажей по лабиринту? Можете объяснить алгоритм пошагово?
6. Какие звуки вы использовали в игре и почему выбрали именно их?
7. Вы могли создать бесконечный уровень в своем лабиринте? Поясните технически.
8. Насколько ваш лабиринт соответствует требованиям задания? Оцените свою работу сами.
9. Что вам больше всего понравилось в работе над проектом и почему?
10. Какие знания, полученные в данном проекте, вы можете применить в будущем?

Раздел 4. Применение событий и игра «Битва магов»

Тема 4.1. Изучение понятий векторная и растровая графика, использование событий, цикла, эффектов.

Теория. Описание инструментов и технологий работы с векторной графикой. Обзор программ для работы с растровой графикой и их основных функций. Изучение основ работы с событиями в графических программах. Рассмотрение применения циклов в графических программах и их примеры использования. Ознакомление с эффектами и фильтрами в графических программах, их видами, применением и настройкой. Разработка простых графических проектов с использованием векторных и растровых изображений.

Задание на внедрение (самостоятельная работа): Создание игры «Битва магов»

Задание для проектной работы

Создание игры «Битва магов», в которой на фоне магического леса будет два персонажа – Волшебник и Ведьма.

Цель работы:

Освоение основ программирования в игровой форме, изучение инструментов и возможностей на примере программы Scratch.

Задача:

Разработать простую игру “Битва магов”, в которой два персонажа: волшебник и ведьма, будут сражаться на фоне магического леса.

План работы:

Изучение основ программирования на Scratch: переменных, операторов, блоков, циклов, условий.

Работа со спрайтами: создание двух персонажей, их анимация, настройка поведения и взаимодействия.

Разработка фонов: создание магического леса, настройка смены сцен.

Написание программы управления персонажами, с учетом их особенностей и возможностей.

Создание системы подсчета очков и индикации результатов. Настройка звукового сопровождения игры.

Тестирование и оптимизация игры, исправление возможных ошибок.

Вопросы для самоконтроля:

1. Что такое растровые и векторные рисунки? Приведи примеры.
2. Почему в компьютерной графике важна разница между растровыми и векторными изображениями?
3. Что такое событие на примере программы Scratch и как оно используется в нашей игре?
4. Какая команда управляет началом игры?
5. Чем полезен цикл в игре «Битва магов»?
6. Какие эффекты применяются в игре и для чего они нужны?
7. Как создается ощущение магии в игре?
8. Какие кнопки управляют действиями волшебника?
9. Что произойдет, если убрать проверку столкновений?
10. Как достигается плавность движений в игре?

Вопросы для контроля (устное собеседование педагога по дополнительному образованию или обсуждение проектной работы в чат-группе с использованием видеоконференцсвязи «Яндекс Телемост»):

Устное собеседование (индивидуальное интервью с ребёнком):

1. Давай посмотрим на твою игру. Расскажи, как начинается битва магов?
2. В чём заключается особенность твоей версии игры? Чем твоя версия отличается от стандартных игр подобного жанра?
3. Пришлось ли тебе исправить какие-то ошибки в процессе создания игры? Возникали ли проблемы, и как ты их решил?

4. Опиши путь прохождения уровней в твоей игре. По каким правилам развивается бой?
5. Участвуют ли оба персонажа одновременно в бою? Или это по очереди?
6. Сделал ли ты систему подсчёта очков? Как очки начисляются игрокам?
7. Получилось ли добиться желаемого результата? Будешь ли совершенствовать игру дальше?
8. Какой элемент оказался самым сложным в разработке? С какими проблемами столкнулся и как их разрешил?
9. Довольны ли твои друзья твоей игрой? Испытали ли они твое творение и дали ли обратную связь?
10. Какую новую идею ты хотел бы воплотить в следующей игре? Что вдохновляет тебя на будущие проекты?

Обсуждение в чат-группе (коллективное обсуждение):

1. Кого заинтересовала идея игры «Битва магов»?
Можете описать, что особенно привлекло ваше внимание?
2. Проанализируем коды наших игр.
Проверьте друг у друга наличие всех необходимых компонентов (например, событий).
3. Ваш персонаж похож на настоящую ведьму или волшебника?
Какой дизайн кажется удачным, а что можно улучшить?
4. Представьте себе мир, в котором живут ваши маги.
Это обычный лес или сказочная страна?
5. Что мешает победить вашему сопернику в игре?
Поделитесь хитростью, как победить врага.
6. Вам понравился рабочий процесс?
Легко ли создавать игры на примере программы Scratch или возникают сложности?
7. Если продолжить разработку игры, что хотелось бы добавить?
Возможно, новых персонажей или интересных бонусов?
8. Любишь играть в свою собственную игру?
Часто ли побеждает твой герой, или всё-таки противник сильнее?

Раздел 5. Использование переменных и клонов (игра «Динозавр»)

Тема 5.1. Изучение понятий переменных и клонов.

Теория. Знакомство с понятием переменной, ее определение и использование. Освоение работы с клонами, создание копий спрайтов и управление ими. Обучение созданию проектов с использованием переменных и клонов.

Задание на внедрение (самостоятельная работа): Создание игры «Динозавр Google Chrome» и последующая ее модификация.

Задание для самостоятельной проектной работы

Создание игры «Динозавр Google Chrome», в которой динозавр будет взаимодействовать с кактусом

Цель:

Изучить основные понятия программирования на примере программы Scratch, получить практические навыки создания игр, познакомиться с основами работы со спрайтами и фонами.

Задачи:

Изучить основные понятия программирования: переменные, блоки, условия, циклы;

Освоить создание и редактирование спрайтов и фонов;

Разработать алгоритм движения и взаимодействия динозавра и кактуса;
Реализовать анимацию и звуковое сопровождение игры;

Написать программу управления динозавром с помощью клавиатуры или мышки;

Протестировать и оптимизировать игру, исправить найденные ошибки.

Вопросы для самоконтроля:

1. Что такое переменная на примере программы Scratch и зачем она нужна?

Приведи пример переменной из твоей игры.

2. Расскажи, как создаешь новый спрайт на примере программы Scratch?

Что ты делал первым делом, когда начинал проект?

3. Какие переменные ты использовал в своей игре «Динозавр»?

Объясни, как каждая переменная помогает в управлении игрой.

4. Что такое клон на примере программы Scratch и как он помогает в игре?

Пример из твоей игры, где использован клон.

5. Почему важно использовать цикл в движущихся объектах?

Как ты использовал цикл в своей игре?

6. В каком месте игры происходит основное действие?

Опиши фон и окружающий ландшафт твоей игры.

7. Какие действия совершает динозавр в твоей игре?

Подпрыгивает, бежит или атакует кактус?

8. Работает ли твоя игра на клавишах или мыши?

Подробно объясни своё решение выбора способа управления.

9. Есть ли у тебя звуковые эффекты в игре?

Расскажи, какие звуки сопровождают игру и как они влияют на восприятие.

10. Проверял ли ты свою игру на ошибки?

Нашёл ли ты ошибки и как исправлял их?

Вопросы для контроля (устное собеседование педагога по дополнительному

образованию или обсуждение проектной работы в чат-группе с использованием видеоконференцсвязи «Яндекс Телемост»):

Устное собеседование (индивидуальное интервью с ребёнком):

1. Представь, что твой динозавр потерял силы и перестал прыгать.
Какие действия нужно предпринять, чтобы вернуть его подвижность?
2. Создай ситуацию, когда динозавр теряет жизнь.
Когда и как это произойдет в твоей игре?
3. Опиши, как изменится скорость динозавра, если увеличить значение переменной? Почему увеличение значения ускоряет персонажа?
4. Какой ключевой элемент в твоей игре заставляет игрока держать внимание?
Например, сложность препятствия или динамика изменений.
5. Предположим, что кактусов стало намного больше.
Как это повлияет на прохождение игры?
6. Реализовал ли ты дополнительное условие окончания игры?
Если да, расскажи подробнее, как оно работает.
7. Рассчитывал ли ты количество жизней для динозавра?
Сколько попыток ты предоставил игроку и почему именно столько?
8. Интересуется ли твой одноклассник или брат/сестра сыграть в твою игру?
Узнал ли ты их мнение о твоём творении?
9. Появлялись ли баги в твоей игре, и как ты их искал?
Применял ли ты метод последовательного воспроизведения ошибок?
10. Продолжаешь ли улучшать свою игру?
Есть ли планы внести что-то новое или дополнить существующие элементы?

Обсуждение в чат-группе (коллективное обсуждение):

1. Поднимите руку, если кто-то ещё играл в версию «Динозаврика» в браузере Google Chrome?
Дополняют ли разработчики классики новыми возможностями?
2. Вероятно, кто-то заменил кактус другим объектом. Пусть поделится идеей замены и причинами выбора альтернативного элемента.
3. Встречал ли кто-то ошибку, когда динозавр застревал на препятствии?
Раскройте причины возникновения такой ошибки и методы её исправления.
4. Попросите товарища показать свой лучший вариант прыжка динозавра.
Оцените высоту и длину прыжков различных версий игры.
5. Выразите своё мнение о важности переменных в игре.
Сравните использование переменных и отсутствие таковых.
6. Если дать игроку неограниченное количество жизней, изменится ли игра кардинально? Стоит ли ограничивать попытки и почему?
7. Некоторые ребята создали оригинальные уровни для своей игры.
Рассказывают ли они о специфике собственного дизайна уровня?

8. Интересно услышать историю самой сложной ошибки в вашей игре.
Вот уж действительно интересная задача — найдите общий корень проблем!

Раздел 6. Создание сложных игровых проектов

Тема 6.1. Разработка многопользовательских игр

Теория. Знакомство с многопользовательскими играми. Изучение основных принципов разработки многопользовательских игр. Работа с движком на примере программы Scratch. Создание спрайтов и фонов для игры. Разработка сценария игры. Определение правил и условий победы. Программирование логики игры. Использование переменных, условий, циклов и других элементов языка на примере программы Scratch. Создание многопользовательского режима игры. Работа с сетевыми функциями на примере программы Scratch. Тестирование и отладка игры. Устранение возможных ошибок и проблем. Оформление игры. Добавление звуковых эффектов, музыки и графических элементов. Публикация игры на платформе на примере программы Scratch. Получение обратной связи от игроков.

Задание на внедрение (самостоятельная работа): Создать проект на примере программы Scratch, который будет сочетать в себе все изученные ранее понятия и инструменты, с целью получения сложного и увлекательного продукта, такого как игра с сюжетом, анимацией и аудио эффектами. В рамках проекта применить переменные, клонирование, события, изменение внешнего вида спрайтов (костюмы), звуки, эффекты и прочие доступные инструменты. Продукт должен обладать понятным и логически организованным интерфейсом, а также поддерживать сохранение и загрузку прогресса пользователя.

Задание для самостоятельной проектной работы

Создание простой многопользовательской онлайн-игры на примере программы Scratch, в которой два пользователя могут играть друг против друга в «камень-ножницы-бумага».

Цель задания:

Разработка простой, но эффективной многопользовательской online-игры на примере программы Scratch, где два участника смогут соревноваться друг с другом в популярной игре «камень-ножницы-бумага», используя логику и стратегию.

Шаги выполнения:

Создание нового проекта на примере программы Scratch и открытие окна «Костюмы». Разработка двух уникальных костюмов для каждого из участников игры.

Открытие окна «Сцены» и создание двух различных сцен, отражающих игровую ситуацию для каждого из двух игроков. Добавление соответствующих костюмов и спрайтов, представляющих участников на сцене.

Написание скрипта, определяющего победителя в каждом раунде. Этот скрипт будет привязан к обоим игрокам и будет отслеживать, кто первый нажал на экран. В

качестве примера скрипта можно использовать следующий:

Когда я получу сигнал "нажать": Если я = (Игрок 1):

Перейти на сцену 2 Иначе Если я = (Игрок 2):

Перейти на сцену 3

Конец Если```

Создание двух разных фонов для каждой из сцен в окне «Фоны». Фон должен соответствовать стилистике игры и быть легко различимым для участников.

Написание скриптов для каждой из сцен, которые изменят фон сцены, в зависимости от результата раунда. Например, если игрок 1 выиграл, фон меняется на фон 1, если же игрок 2 выиграл, фон изменяется на фон 2.

Создание многопользовательского режима путем открытия окна «Сетевые проекты» и выбора «Добавить сервер». Ввод имени сервера и его запуск. Привлечение других участников к игре через приглашение присоединиться к созданному серверу.

Тестирование игры, убеждение в ее работоспособности и готовности к использованию.

Вопросы для самоконтроля:

1. Что такое многопользовательская игра и чем она отличается от обычной игры? (Приведи пример разницы между ними.)

2. Какую роль играют костюмы в игре «Камень-Ножницы-Бумага»? (Почему важен выбор костюмов?)

3. Зачем нужны сцены на примере программы Scratch и как ты их использовал в своей игре? (Какие сцены созданы и как они работают?)

4. Какие сценарии ты прописал для определения победителя в каждом раунде? (Расскажи коротко, как определяется победитель.)

5. Чем полезен сетевой режим на примере программы Scratch? (Каким образом пользователи подключаются к игре и начинают играть?)

6. В чем состоит важность тестирования игры перед ее публикацией? (Что нужно проверять и как?)

7. Объясни, как организована смена фонов в зависимости от результата раунда? (При победе какого игрока меняется фон и почему?)

8. Почему важно иметь удобный интерфейс для управления в игре? (Чем удобнее пользоваться твоим управлением и почему?)

9. Если возникнет ситуация, когда два игрока выбирают одинаковые фигуры, что произойдет? (Будет ли ничья и как это отразится на игровом процессе?)

10. Сталкивался ли ты с ошибками при разработке игры и как ты их устранял? (Какие были самые распространённые ошибки и как решить подобные проблемы?)

Вопросы для контроля (устное собеседование педагога по дополнительному образованию или обсуждение проектной работы в чат-группе с использованием

видеоконференцсвязи «Яндекс Телемост»):

Устное собеседование (индивидуальное интервью с ребёнком):

1. Расскажи, как ты создавал костюмы для фигур «камень», «ножницы» и «бумагу»? (Поделись своим процессом создания изображений и дизайном фигур.)

2. Программируя скретч-коды, какие действия выполнялись первыми, когда запускалась игра? (Началось ли соревнование автоматически или нужно совершить определенное действие?)

3. Решив запустить игру на двоих пользователей, сталкивались ли вы с техническими трудностями? (Например, проблемы синхронизации хода игры или подключение нескольких устройств.)

4. Можешь привести конкретный пример кода, который определяет исход раунда в игре? (Покажи, как проверяется выигрыш или проигрыш игрока.)

5. У тебя есть специальная сцена, отображающая конец игры? (Если да, как она выглядит и что там происходит?)

6. Разработав игру, испытывал ли ты желание поиграть в неё с друзьями? (Был ли интересен собственный проект, играли ли вы вдвоём с кем-то?)

7. Что сложнее всего оказалось в разработке многопользовательской игры? (Какие конкретные моменты вызвали наибольшие трудности?)

8. Изменялся ли фон игры в зависимости от выигрышей и поражений? (Есть ли наглядные признаки победы или поражения?)

9. Как ты считаешь, подходит ли данная игра для соревнований между детьми школьного возраста? (Привлекательна ли она для участия в турнирах или соревнованиях?)

10. Готов ли ты выложить свою игру в сеть и пригласить друзей попробовать ее? (Как думаешь, понравится ли им твой проект?)

Обсуждение в чат-группе (коллективное обсуждение):

1. Все ли смогли придумать уникальный дизайн для костюмов камней, ножниц и бумаги? (Как выглядели ваши костюмы и кто показал оригинальность в дизайне?)

2. Проблемы синхронизации при игре на двух устройствах — встречались ли они у кого-то? (Опытные дети делятся, как преодолеть подобную проблему.)

3. Если один игрок выигрывает несколько раундов подряд, меняется ли что-то визуально? (Делился ли кто-то способами награждения или мотивации?)

4. Были ли задуманы специальные бонусы и награды для победителя матча? (Расскажи, как мотивировал победителей своей игрой.)

5. Проводили ли вы турнир с друзьями по сети, используя свою игру?

Итоговая аттестация — Открытая презентация проекта (игр Flappy Bird и Space Battle, созданных самостоятельно в течение учебного периода)

В целях обеспечения прозрачности и объективности процедуры итоговой аттестации обучающихся по дополнительной общеобразовательной общеразвивающей программе технической направленности «Теория разработки игр: первые шаги юного игродела!»», проводимой в формате публичной презентации проекта с устным собеседованием-рефлексией используется система прокторинга (предоставляется платформой Антитренинги, запись в реестре отечественного ПО №15883 от 09.12.2022).

Тема 7.1. Закрепление пройденного материала.

Теория. Повторение всего теоретического материала курса, устное собеседование-рефлексия ("Что было труднее всего?")

Задание на внедрение (самостоятельная работа): Подготовка презентации проекта - созданной самостоятельно игры Flappy Bird.

Тема 7.2. Закрепление пройденного материала и подведение итогов курса.

Теория. Повторение всего теоретического материала курса, устное собеседование-рефлексия ("Что было труднее всего?")

Задание на внедрение (самостоятельная работа): Подготовка презентации проекта - созданной самостоятельно игры Space Battle.

1.4. Планируемые результаты

В результате освоения программы обучающиеся должны обладать следующими знаниями, умениями и навыками:

Знания:

- знание компьютерной терминологии, правил работы с ПК, основных правил техники безопасности, информационной безопасности и цифровой гигиены;
- знание основных понятий и элементов информационной культуры;
- знание основных правил и инструментов работы на примере программы Scratch;
- знание базовых принципов разработки современных цифровых проектов.

Умения:

- умение использовать средства цифровых технологий для решения когнитивных, коммуникативных и организационных задач в образовательной деятельности и повседневной жизни;
- умение ориентироваться в современном информационном пространстве;

- умение планировать, контролировать и оценивать собственные действия в соответствии с поставленной задачей и условиями её реализации, определять наиболее эффективные способы достижения результата;
- умение применять приобретенные знания и навыки в практической деятельности;
- умение создавать конкурентоспособный IT-продукт;
- умение презентовать проекты.

Навыки:

- навыки логического и алгоритмического мышления, критического мышления;
- творческое мышление;
- навыки самоорганизации и самообучения;
- коммуникативные навыки;
- развитие внимания и памяти;
- способность к самостоятельной информационно-познавательной деятельности;
- навыки работы с информацией – ее поиска, анализа, структурирования, организации представления и передачи, хранения и защиты;
- навыки сотрудничества и работы в команде, способность конструктивно разрешать конфликты с учетом интересов сторон.

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1. Календарный учебный график

Пояснительная записка:

Календарный учебный график в полном объеме учитывает индивидуальные, возрастные, психофизические особенности обучающихся и отвечает требованиям охраны их жизни и здоровья.

Продолжительность учебного часа (учебных занятий) 45 минут.

При этом продолжительность непрерывного использования экрана с использованием электронного курса “Теория разработки игр: первые шаги юного игродела!”, размещенного на платформе Skillspace) составляет 7 минут.

Далее педагог дополнительного образования предлагает обучающимся самостоятельно выполнить задание на бумаге или сформулировать ответ на вопрос (на выполнение задания дается 7 минут).

После выполнения задания обучающиеся возвращаются к экрану для подведения итогов в формате устного собеседования или обсуждения в чат-группе с использованием видеоконференцсвязи «Яндекс Телемост» (обратная связь "Обсудим вместе"), где педагог дополнительного образования ставит конкретные вопросы о

пройденном материале.

Продолжительность перерыва для отдыха между занятиями 15 минут –предусматривается проведение физкультминуток, гимнастики для глаз, обеспечивается контроль за осанкой.

Объем образовательной нагрузки обучающихся составляет в учебный день – не более 2 академических часов, в неделю - не более 4 академических часов.

Комплектование групп, составление календарного учебного плана, организация (в случае необходимости) корректировки календарного учебного графика за счет объединения или уплотнения тем занятий, осуществляется педагогом дополнительного образования, реализующим дополнительную общеобразовательную общеразвивающую программу.

№	Раздел	Распределение часов по неделям											
		1-я нед.	2-я нед.	3-я нед.	4-я нед.	5-я нед.	6-я нед.	7-я нед.	8-я нед.	9-я нед.	10-я нед.	11-я нед.	12-я нед.
1	Основы разработки игр на примере среды Scratch	4	2										
2	Первая игра «Пенальти»		2	4									
3	Работа с фонами и игрой «Лабиринт»				4	2							
4	Применение событий и игра «Битва магов»					2	4						
5	Использование переменных и клонов (игра «Динозавр»)							4	2				
6	Создание сложных игровых проектов								2	4			

7	Итоговая аттестация — Открытая презентация проекта (игр Flappy Bird и Space Battle, созданных самостоятельно в течение учебного периода)									4	2	2
---	---	--	--	--	--	--	--	--	--	---	---	---

ИТОГО: 44 часа

2.2. Условия реализации программы

Форма обучения: с использованием электронного обучения и дистанционных образовательных технологий.

Форма организации образовательной деятельности обучающихся:

Занятия проходят с использованием электронного курса “Теория разработки игр: первые шаги юного игродела!”, размещенного на платформе Skillspase (ООО «СКИЛЛСПЕЙС», ИНН 9701194047) в формате видеоконференций (вебинары) с использованием видеоконференцсвязи «Яндекс Телемост» (ООО «ЯНДЕКС», ИНН 7736207543 и системы прокторинга (ООО "ИНТЕРНЕТ УНИВЕРСИТЕТ", ИНН: 7726407960).

Данный формат организации учебного процесса обеспечивает учет контингента обучающихся.

Авторизация обучающегося осуществляется в системе дистанционного обучения на платформе Skillspase, где также размещаются активные ссылки на лекционные и практические занятия.

Технология видеоконференцсвязи «Яндекс Телемост» обеспечивает возможность:

- авторизации пользователей,
- трансляции экрана,
- осуществления записи видеоконференции,
- взаимодействия с обучающимися посредством видео, аудио и чата.

По каждому обучающемуся образовательной программы можно ознакомиться с

информацией о нем (учет контингента обучающихся), с процессом его обучения, с тем, какие модули обучающийся завершил, какие оценки получил.

Занятия проходят согласно календарному учебному графику и учебно-тематическому плану.

2.2.1. Материально-техническое обеспечение

1. Технические требования

- Скоростной интернет: Обеспечение стабильного подключения к сети интернет, рекомендуемая пропускная способность сети для участника не менее 2 Мбит/сек исходящего и входящего канала.

2. Оборудование для преподавателей и обучающихся

- Компьютеры/ноутбуки: Все участники процесса должны иметь доступ к современным устройствам с достаточной производительностью для комфортной работы в интернете.

- Веб-камера с разрешением SD или HD: встроенная или USB.

- Динамики и микрофоны: встроенные, подключаемые по USB или беспроводные (Bluetooth).

- Наушники/системы звукоизоляции: важно исключить посторонние шумы и обеспечить четкость звука.

Высококачественное оборудование для видеоконференций позволяет проводить лекции и семинары в режиме реального времени.

3. Программное обеспечение

- Платформа для проведения вебинаров и лекций: «Яндекс Телемост» (ООО «ЯНДЕКС») для организации групповых занятий в формате видеоконференцсвязи.

- Система управления обучением: Платформа «Skillspace» (ООО «СКИЛЛСПЕЙС») для размещения учебных материалов- презентаций, ссылок на вебинары, заданий и отслеживания успеваемости обучающихся.

- Система прокторинга: Платформа Антитренинги ООО "ИНТЕРНЕТ УНИВЕРСИТЕТ для записи всех действий обучающихся, включая веб-камеру, микрофон и экран компьютера.

- Использование браузера Chrome, Chrome Mobile выше 83 версии, Firefox выше 85 версии, другие браузеры на базе Chromium (Opera, Vivaldi, Yandex) выше 83 версии.

Каждому обучающемуся и педагогу дополнительного образования предоставляется доступ к участию и проведению занятий посредством направления персонального логина и пароля до проведения занятий.

2.2.2. Кадровое обеспечение

Педагогическая деятельность по дополнительной общеразвивающей программе осуществляется лицами, имеющими среднее профессиональное или высшее образование, в том числе по направлению, соответствующему направлению дополнительной общеразвивающей программы, и отвечающими квалификационным требованиям, указанным в Едином квалификационном справочнике должностей руководителей, специалистов и служащих, раздел «Квалификационные характеристики должностей работников образования», утвержденным приказом министерства здравоохранения и социального развития Российской Федерации от 26 августа 2010 года № 761н (в отношении должности «педагог дополнительного образования»), а также в профессиональном стандарте «Педагог дополнительного образования детей и взрослых», утвержденном приказом министерства труда и социальной защиты Российской Федерации от 22 сентября 2021 года № 652н.

Организация вправе привлекать к занятию педагогической деятельностью по данной дополнительной общеобразовательной программе лиц, обучающихся по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности настоящей образовательной программы, и успешно прошедших промежуточную аттестацию не менее чем за два года обучения.

Все педагогические работники осваивают дополнительные профессиональные программы – программы повышения квалификации по профилю педагогической деятельности не реже одного раза в три года.

2.3. Формы контроля и аттестации

Отслеживание динамики освоения программы и анализ результатов ее освоения осуществляется в форме педагогического мониторинга, который осуществляется в течение всего учебного периода и включает:

- **первичную диагностику**, которая проводится при поступлении обучающегося для определения уровня его подготовки. Форма проведения – устное собеседование с использованием видеоконференцсвязи «Яндекс Телемост»;
- **итоговую аттестацию**, которая осуществляется по окончании освоения Программы для демонстрации освоения навыков. Форма- открытая презентация проектов и устное собеседование-рефлексия с использованием системы прокторинга, которая предоставляется платформой Антитренинги (запись в реестре отечественного ПО №15883 от 09.12.2022, ООО "ИНТЕРНЕТ УНИВЕРСИТЕТ", ИНН: 7726407960). Обучающиеся демонстрируют созданные ими самостоятельно игровые разработки (Flappy Bird и Space Battle) перед виртуальной аудиторией. Для этого они готовят презентацию, в которой показывают финальный результат своей практической работы, подробно рассказывают о сложностях при разработке и путях их преодоления, взаимодействии игровых элементов, концепции, целях, механизмах управления, особенностях сюжета, работе интерфейсов и особенностях проектов Flappy Bird и Space Battle. Презентация проходит в формате видеоконференции с последующим устным собеседованием от педагога дополнительного образования

("Что было труднее всего?") и возможностью вопросов/комментариев о проделанной работе автору презентации от других обучающихся в чате.

2.4. Оценочные материалы

2.4.1. Вопросы в рамках первичной диагностики

Вопросы для устного собеседования с использованием видеоконференцсвязи «Яндекс Телемост»):

1. Какие знания и опыт в разработке игр у вас есть?
2. Что такое среда на примере программы Scratch? Какие ее особенности и возможности в разработке игр?
3. Какие компоненты в игре можно создавать на примере программы Scratch? Назовите несколько примеров.
5. Какие блоки программирования на примере программы Scratch используются для управления персонажем или объектами в игре?
6. Какие преимущества и недостатки на примере программы Scratch по сравнению с другими средами или языками программирования для разработки игр?
7. Какими достижениями вы будете оценивать свой успех в обучении новичков разработке игр на примере программы Scratch?

2.4.3. Итоговая аттестация в форме открытой презентации проекта с устным собеседованием-рефлексией с использованием системы прокторинга, которая предоставляется платформой Антитренинги (запись в реестре отечественного ПО №15883 от 09.12.2022)

Задание 1 на внедрение (самостоятельная работа)

Тема:

Создание игры Flappy Bird.

Цель задания:

Научиться создавать игры с использованием различных элементов и инструментов на примере программы Scratch, углубить понимание основных понятий программирования, а также развить навыки работы с графикой и анимацией.

Описание задания:

Создать игру Flappy Bird, которая представляет собой аркаду с элементами головоломки. Игроку предстоит управлять птичкой, преодолевая различные препятствия, при этом избегая столкновений с ними.

В ходе выполнения задания необходимо:

Изучить основы программирования на примере языка Scratch: переменные, операторы, блоки команд, циклы и условия.

Освоить работу с графикой: создание и редактирование спрайтов, фонов, костюмов, а также изменение их свойств.

Разработать алгоритм поведения спрайта: определение условий, при которых спрайт может двигаться, останавливаться, изменять направление движения и т.д.

Реализовать анимацию спрайта, используя различные эффекты и команды программирования.

Написать программу для управления спрайтом с клавиатуры или мыши, учитывая особенности игрового процесса Flappy Bird (например, управление птичкой с помощью клавиш “стрелка вверх” и “стрелка вниз”).

Добавить в игру препятствия и ловушки, которые будут менять свое местоположение, форму и цвет в процессе игры. Это позволит сделать игру более разнообразной и интересной.

Реализовать систему подсчета очков и определения результатов игры.

Создать звуковое сопровождение для игры, которое будет меняться в зависимости от действий игрока и происходящих событий.

Протестировать созданную игру, исправить возможные ошибки и недочеты, а также оптимизировать программу для улучшения производительности и удобства использования.

Задание 2 на внедрение (самостоятельная работа)

Тема:

Создание игры Space Battle.

Цель:

Научиться разрабатывать игры в среде программирования на примере программы Scratch, освоить работу с различными элементами и инструментами программы, углубить знание основ программирования и развить навыки графического и анимационного оформления.

Задание:

Разработать игру под названием “Space Battle” – аркада с элементами стратегии, в которой игрок управляет космическим кораблем и сражается с врагами.

Ход выполнения:

Изучить основы программирования на примере программы Scratch: переменные, операции, блоки, циклы, условия.

Освоить работу со спрайтами, фонами и костюмами: создание, редактирование и настройка.

Разработать алгоритм поведения корабля, включая движение, маневры, стрельбу и взаимодействие с другими объектами.

Реализовать анимацию корабля, используя различные команды и эффекты. Написать программу для управления кораблем с помощью клавиатуры, учитывая особенности космического боя (управление клавишами “вверх”, “вниз”, “вправо”, “влево”, “огонь”).

Добавить в игру врагов, которые будут появляться, перемещаться и исчезать в

случайном порядке.

Реализовать систему подсчета баллов и отображения результатов игры.

Создать музыкальное сопровождение для игры и звуковые эффекты для различных действий.

Протестировать игру, исправить ошибки и оптимизировать код для улучшения эффективности и пользовательского опыта.

Структура итоговой аттестации:

Блок 1. Подготовительный этап

- Обучающиеся заранее выполняют задание для практической работы по созданию игр Flappy Bird и Space Battle.

- Обучающиеся заранее получают рекомендации и требования к оформлению презентации и выступлению.

- Подготовка презентационных материалов и репетиционные тренировки.

Блок 2. Презентация проекта с использованием системы прокторинга, которая предоставляется платформой Антитренинги (запись в реестре отечественного ПО №15883 от 09.12.2022).

Участники последовательно выступают перед виртуальной аудиторией и показывают свои созданные игровые проекты (Flappy Bird и Space Battle) с акцентом на следующем содержании:

1. Описание концепции игры:

- Идея и замысел игры.

- Цели и задачи проекта.

2. Особенности сюжета и игрового процесса:

- Основные правила и механика игры.

- Уникальность и привлекательность проекта.

3. Управление и взаимодействие игровых элементов:

- Механизмы управления персонажами и объектами.

- Логика взаимодействия элементов игры.

4. Интерфейс и визуализация:

- Внешний вид меню, окон, элементов интерфейса.

- Графическое оформление и анимации.

5. Технические характеристики:

- Сложности и проблемы, возникающие при разработке.

- Пути их преодоления и достигнутые результаты.

6. Результат работы:

- Показ финального продукта (демонстрация игры в действии).

Блок 3. Устное собеседование-рефлексия педагога дополнительного образования и обсуждение в чат-группе с использованием системы прокторинга, которая предоставляется платформой Антитренинги (запись в реестре отечественного ПО №15883 от 09.12.2022):

После защиты проекта обучающийся переходит к устному собеседованию, отвечая на вопросы педагога дополнительного образования и других обучающихся:

1. Рефлексия:

- Что было труднее всего?
- Что лично ученик извлек из опыта работы над проектом?

2. Обратная связь от слушателей:

- Другие участники группы задают вопросы, комментируют и предлагают дополнения.

3. Итоговый комментарий педагога дополнительного образования:

- Индивидуальная оценка проекта и выступления.
- Конструктивные замечания и рекомендации по совершенствованию проекта.

Блок 4. Завершение аттестации:

- Педагог дополнительного образования подводит итоги выступлений и благодарит участников.

Требования к выступлению:

- Продолжительность презентации: 5–7 минут.
- Четкая структура изложения материала.
- Грамотное владение терминами и понятиями программирования.

Критерии оценки:

- **личностные:** личное отношение к проекту; способность аргументированно и уверенно отвечать на вопросы; готовность к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов программирования; способность и готовность к общению и сотрудничеству со сверстниками и взрослыми в процессе образовательной и творческой деятельности; умение самостоятельно планировать, осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей; выбирать успешные стратегии в различных ситуациях; владение навыками познавательной и проектной работы; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; развитие навыков самоанализа и самооценки своей деятельности;

- **предметные:** качество подготовки презентации; глубина проработанности темы и демонстрация приобретенных навыков, а именно алгоритмизации задачи, освоения основных этапов решения задачи, навыков разработки несложных программ, навыков разработки

проекта, его структуры, дизайна.

Для определения степени освоения программы используется уровневая оценка:

- **Высокий уровень:** Обучающийся свободно выступил публично, представил качественный проект и презентацию, готов к профессиональному росту, глубоко освоил программу, свободно отвечает на вопросы, грамотно выполняет задания, активно ищет дополнительную информацию, делает выводы, проявляет инициативу и творческое мышление.
- **Средний уровень:** Выступление приемлемое, проект и презентация соответствует основным требованиям, обучающийся владеет цифровыми инструментами, но имеют слабые стороны, знает основы материала, иногда ошибается, отвечает неполно, нуждается в подсказках, интересуется отдельными аспектами темы, целенаправленно работает над достижением результата.
- **Низкий уровень:** Обучающийся испытывает большие трудности при выступлении, проект и презентация низкого качества, слабо владеют цифровыми технологиями, отсутствует стремление к профессиональному развитию, ограниченно владеет материалом, допускает существенные ошибки, плохо ориентируется в изученной информации, пассивен, действует строго по инструкции, не способен сделать собственные выводы.

Инструкция проктора при проведении итоговой аттестации в формате публичной презентации проекта с устным собеседованием-рефлексией

Синхронный прокторинг — это процесс мониторинга аттестационного мероприятия в режиме реального времени с использованием технологий видеонаблюдения И системы прокторинга.

1. Проктор обязан следить за соблюдением порядка проведения аттестационного онлайн-мероприятия и синхронного прокторинга, своевременно пресекать нарушения.

2. Проктор контролирует техническую готовность и проводит тестирование телекоммуникационного оборудования перед началом аттестации:

- Камеры высокого разрешения, обеспечивающие четкое изображение лица и рук.

- Микрофоны, позволяющие слышать речь и фиксировать любые посторонние звуки.

- Интернет-соединение: стабильная связь для передачи видеопотоков.
 - Платформы Антитренинги (запись в реестре отечественного ПО №15883 от 09.12.2022)
 - Обеспечение достаточного освещения и отсутствия посторонних шумов.
3. Проктор ведет мониторинг действий обучающихся в режиме реального времени и протокол проведения итоговой аттестации, регистрирует факт выявления возможных нарушений и составляет акт о нарушении (при наличии оснований).
 3. Проктор проводит инструктаж обучающихся перед началом аттестации: разъяснение правил поведения участников и ведения видеонаблюдения.
 4. По окончании заседания докладывает руководителю о результатах наблюдения.

Заключение:

Итоговая аттестация направлена на формирование навыков публичного выступления, систематизацию знаний и получение опыта разработки полноценных игровых продуктов в среде программирования на примере программы Scratch, важных для дальнейшего профессионального роста и развития цифровых компетенций.

2.4. Методические материалы

Теоретические знания подаются с помощью словесного изложения информации, наглядного материала — тематических презентаций, просмотр которых осуществляется с помощью мультимедийного проектора, а также использования компьютерных программных средств (среды визуального программирования на примере программы Scratch, программы просмотра подготовки и презентаций Powerpoint).

В качестве наглядного материала используются презентации со следующей тематикой:

1. Среда программирования на примере программы Scratch;
2. Спрайт, костюм, скрипт;
3. Игра «Пенальти»;
4. Игра «Лабиринт»;
5. Взаимодействие объектов. Игра «Битва магов»;
6. Браузерная игра;
7. Мобильная игра. Игра Flappy Bird;
8. Классическая игра. Игра Space Battle.

2.5. Организация учета и хранения результатов образовательного процесса

Учет и хранение результатов освоения дополнительной общеобразовательной общеразвивающей программы «Теоретические и практические основы создания игр» организуется в целях фиксации факта и результатов итоговой аттестации обучающихся.

Учет результатов осуществляется педагогом, реализующим программу, путем ведения электронного Журнала учета результатов итоговой аттестации в формате Microsoft Excel (или аналогичном редакторе электронных таблиц).

Журнал содержит следующие сведения по каждому обучающемуся, успешно прошедшему итоговую аттестацию в форме защиты выпускного проекта:

1. Фамилия, имя, отчество обучающегося.
2. Дата защиты выпускного проекта.
3. Наименование защищенного проекта (игры Flappy Bird и/или Space Battle).
4. Итоговая оценка по результатам защиты (высокий, средний, низкий уровень).

Внесение записей в Журнал осуществляется педагогом на основании проведенной защиты в течение 3 (трех) рабочих дней с даты ее проведения.

Хранение Журнала осуществляется в электронно-цифровой среде (на электронном носителе) у педагога, реализующего программу. Срок хранения Журнала составляет 3 (три) года после завершения реализации программы. Ответственность за достоверность и своевременность внесения сведений возлагается на педагога дополнительного образования.

2.6. Информационное обеспечение

Основные информационные источники:

1. Голиков Денис. Scratch для учителей и родителей. Знакомство с популярной детской средой программирования, 2017;
2. Рейна Бердитт. Программирование на Scratch с нуля. Создаем веселые игры, охотимся за багами и пишем первые программы!/ пер. с англ. Е. Желнаковой — Москва: Эксмо, 2023;
3. Мажед Маржи. Scratch для детей. Самоучитель по программированию / пер. с англ. М. Гескиной и С. Таскаевой — Москва: Манн, Иванов и Фербер, 2017.