

**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
ОНЛАЙН-ШКОЛА «ТОЧКА ЗНАНИЙ»**

УТВЕРЖДЕНА
Приказом Генерального
директора
ООО «Точка знаний»
№ 19 от «29» августа 2025 г.

**Дополнительная общеразвивающая программа
«Разработка игр в Roblox. Ступень 2»**

(трудоемкость 97 ак. ч.)

Разработчик:
Скрыпченко Егор Николаевич
Преподаватель дополнительного образования

Возраст: дети (от 9 лет)
Срок обучения: 9 месяцев

Краснодар, 2025 г.

СОДЕРЖАНИЕ

1. Общая характеристика.....	3
1.1. Пояснительная записка.....	3
1.2. Направленность.....	3
1.3. Актуальность программы.....	3
1.4. Цели и задачи программы.....	3
1.5. Категория обучающихся.....	3
1.6. Форма обучения и сроки освоения.....	4
1.7. Форма организации образовательной деятельности.....	4
1.8. Документ, выдаваемый после завершения обучения.....	4
2. Содержание программы.....	5
2.1. Календарный учебный график.....	5
2.2. Учебный план.....	5
2.3. Рабочая программа.....	7
2.4. Кадровое обеспечение.....	13
2.5. Материально-техническое обеспечение реализации программы.....	13
2.6. Форма аттестации и оценочные материалы.....	14
2.7. Список рекомендованной литературы.....	18
2.8. Список использованной литература.....	19
2.8.1. Электронные ресурсы.....	19

1. ОБЩАЯ ХАРАКТЕРИСТИКА

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Разработка игр в Roblox. Ступень 2» (далее – Программа) разработана в соответствии с Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации», Приказом Министерства просвещения РФ от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам», Письмом Министерства образования и науки РФ от 18.11.2015 № 09-3242 «О направлении Методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разно уровневые)».

Создание игр в Roblox Studio, интегрирующее углублённые знания геймдизайна, 3D-моделирования и программирования на языке Lua, позволяет учащимся развивать практические навыки через проектирование сложных игровых механик и интерактивных систем. Курс акцентирует внимание на использовании скриптов, работе с событиями, настройке взаимодействий между объектами и расширенных возможностях редактора, что углубляет понимание разработки игр и её ключевых концепций.

1.2. Направленность: техническая.

1.3. Актуальность программы:

Программа актуальна из-за возрастающей популярности платформы Roblox как среды для обучения и творчества в игровой индустрии. Владение навыками программирования на языке Lua помогает учащимся развивать логические и аналитические способности, что важно для их учёбы и будущей карьеры в IT. Углублённое знакомство с игровыми технологиями формирует интерес к программированию, геймдизайну и способствует дальнейшему профессиональному развитию в сфере цифровых технологий.

1.4. Цели и задачи Программы:

Углубить знания учащихся в области разработки игр на платформе Roblox, научить применять продвинутые возможности Lua и инструменты Roblox Studio для создания более сложных игровых механик, интерфейсов и многопользовательских режимов.

Задачи программы:

- Углубить понимание языка Lua, включая функции высшего порядка, модули и клиент-серверное взаимодействие;
- Научить использовать Remote Events и Remote Functions для построения многопользовательских сценариев;
- Развить навыки создания продвинутых интерфейсов (GUI 2.0) и взаимодействия с пользователем;
- Обучить работе с системой данных игрока и сохранения прогресса;
- Познакомить с основами оптимизации и структурирования игровых проектов;
- Научить создавать и настраивать многопользовательские мини-игры и симуляторы;
- Развить навыки презентации, отладки и публикации полноценных проектов.

1.5. Категория обучающихся: к освоению дополнительной общеобразовательной общеразвивающей программы допускаются учащиеся 3-9-х классов, в возрасте 9-15 лет.

В результате изучения курса обучающиеся должны

знать:

- Расширенные возможности языка Lua, включая модули, замыкания, функции высшего порядка;
- Принципы клиент-серверного взаимодействия и безопасность серверной логики;
- Механизмы работы с Remote Events и Remote Functions в многопользовательских играх;
- Структуру пользовательского интерфейса в Roblox Studio и способы его расширения (GUI 2.0);
- Основы организации данных игрока и сохранения прогресса;
- Принципы построения масштабируемых игровых систем и архитектур.

уметь:

- Создавать модули и организовывать структуру игрового кода;
- Реализовывать взаимодействие между клиентом и сервером с помощью Remote Events/Functions;
- Настраивать сложные GUI-экраны, включая меню, и интерфейсы взаимодействия;

- Разрабатывать многопользовательские механики: PvP, кооператив, симуляторы;
- Использовать визуальные эффекты и анимации для улучшения пользовательского опыта;
- Оптимизировать производительность проекта и выявлять ошибки;
- Разрабатывать, тестировать и публиковать полнофункциональные многопользовательские игры.

1.6. Форма обучения и сроки освоения:

Программа реализуется исключительно с применением дистанционных образовательных технологий и электронного обучения заочной форме.

Сроки освоения программы – **9 месяцев.**

Лекционные занятия онлайн (вебинар) – **48 ак.ч.**

Самостоятельная работа – **48 ак.ч.**

Итоговая аттестация (тестирование) - **1 ак.ч.**

Период обучения и режим занятий:

Занятия проводятся 1 раз в неделю по **1,33 ак.ч.**

1.7. Форма организации образовательной деятельности: групповая.

1.8. Документ, выдаваемый после завершения обучения: сертификат об обучении.

2. СОДЕРЖАНИЕ ПРОГРАММЫ

Программа включает: 36 уроков и итоговый тест.

Перечень, трудоемкость, последовательность и распределение уроков, формы промежуточной аттестации слушателей определяются учебным (тематическим) планом.

Содержание каждого урока включает лекционный и практический материал.

2.1 Календарный учебный график

Общая учебная нагрузка – 36 дней																	
1 мес яц		2 ме ся ц		3 мес яц		4 мес яц		5 мес яц		6 месяц		7 месяц		8 месяц		9 меся ц	
ЛЗ, ЕР	СР,	ЛЗ, СР, ЕТ		ЛЗ, ЕР	СР,	ЛЗ, ЕР	СР,	ЛЗ, ЕР	СР,	ЛЗ, ЕР	СР,	ЛЗ, СР, ЕТ		ЛЗ, ЕР	СР,	ЛЗ, СР, ЕР ИА	

ЛЗ – лекционные занятия

СР – самостоятельная работа

ЕТ – ежемесячное тестирование

ИА – итоговая аттестация

2.2. Учебный план

№ п / п	Название модулей	Количество часов			
		Всего ак.ч.	Лекционн ые занятия онлайн (видеоурок/ вебинар) ак.ч.	Самостоя- тельная работа ак.ч.	Форма проверки
1	Модуль №1: Углублённое программирование на Lua	20	10	10	ТК, ЕТ
2	Модуль №2: Расширенные механики и интерфейс (GUI 2.0)	22	11	11	ТК, ЕТ

3	Модуль №3: Разработка многопользовательских игр	22	11	11	ТК, ЕТ
4	Модуль №4: Финальные проекты и публикация	32	16	16	ТК, ЕТ
	Итоговая аттестация	1	0	1	Тестирование
	ИТОГО	97	48	49	

ТК – текущий контроль

ЕТ – ежемесячное тестирование

2.3. Рабочая программа

Модуль №1: Углублённое программирование на Lua

Урок №1: Переменные в Lua

Понятие переменной и зачем она нужна в программировании.
Как создавать переменные и присваивать им значения.
Типы данных в Lua: числа, строки, логические значения.
Именованные переменные — правила и лучшие практики.
Практика: создание переменных и вывод значений в Output.
По окончании урока слушатели выполняют домашнее задание.

Урок №2: Циклы

Что такое цикл и зачем он нужен в играх.
Виды циклов в Lua: `for`, `while`, `repeat until`.
Примеры повторяющихся действий с использованием цикла.
Как использовать переменные-счётчики и условия выхода.
Практика: создаём простой цикл с выводом чисел.
По окончании урока слушатели выполняют домашнее задание.

Урок №3: Условия (если...то)

Что такое условие в программировании.
Операторы `if`, `else`, `elseif` в Lua.
Как делать выбор между разными действиями в коде.
Примеры использования условий в играх: проверка жизней, очков и т.д.
Практика: написание скрипта с условием.
По окончании урока слушатели выполняют домашнее задание.

Урок №4: Операторы и выражения в Lua

Что такое выражения и операторы.
Математические операторы: `+`, `-`, `*`, `/`, `%`.
Операторы сравнения: `==`, `~=`, `<`, `>`, `<=`, `>=`.
Логические операторы: `and`, `or`, `not`.
Комбинирование условий и выражений для более сложной логики.
Практика: создание выражений и тестирование условий.
По окончании урока слушатели выполняют домашнее задание.

Урок №5: Функции

Что такое функция и зачем она нужна.
Как объявить и вызвать функцию в Lua.
Параметры и возвращаемые значения.
Примеры использования функций в игровых скриптах.
Как структурировать код с помощью функций.
Практика: написание функций с разными параметрами.
По окончании урока слушатели выполняют домашнее задание.

Урок №6: Массивы

Что такое массив (таблица) в Lua.
Как создать массив и обращаться к его элементам.
Добавление, изменение и удаление элементов массива.
Индексация: с чего начинаются индексы в Lua.
Практика: работа с массивами и вывод значений.
По окончании урока слушатели выполняют домашнее задание.

Урок №7: Цикл for с ipairs

Что такое ipairs и зачем он нужен.

Чем отличается ipairs от pairs.

Как пройти по массиву с помощью for и ipairs.

Примеры использования: подсчёт значений, отображение списка.

Практика: создание цикла с ipairs для вывода элементов массива.

По окончании урока слушатели выполняют домашнее задание.

Урок №8: События

Что такое события в Roblox и как они работают.

Различие между событиями клиента и сервера.

Создание и обработка пользовательских событий (Custom Events).

Примеры: нажатие кнопки, столкновение, сбор предмета.

Практика: подключение обработчика события и выполнение действий.

По окончании урока слушатели выполняют домашнее задание.

Модуль №2: Расширенные механики и интерфейс (GUI 2.0)

Урок №9: Основы интерфейсов в Roblox

Что такое интерфейс (UI) в Roblox и зачем он нужен.

Обзор элементов: ScreenGui, TextLabel, ImageLabel, TextButton.

Размещение и масштабирование UI-элементов.

Панель свойств для настройки внешнего вида.

Практика: создание простого интерфейса с текстом и кнопкой.

По окончании урока слушатели выполняют домашнее задание.

Урок №10: Создание главного меню игры

Планирование структуры меню: логотип, кнопки, фон.

Создание кнопок: «Играть», «Настройки», «Выход» и т.д.

Привязка кнопок к действиям с помощью событий.

Переход от главного меню к игровому процессу.

Практика: сборка главного меню в игре.

По окончании урока слушатели выполняют домашнее задание.

Урок №11: Индикаторы жизней и очков

Создание HUD — панелей состояния игрока.

Панель здоровья: визуализация количества жизней.

Счётчик очков и его обновление при сборе предметов или победах.

Обновление UI через скрипты Lua.

Практика: настройка интерфейса с жизнями и очками.

По окончании урока слушатели выполняют домашнее задание.

Урок №12: Меню выбора команды

Пример команд: красная команда, синяя команда.

Создание меню с кнопками выбора стороны.

Скрипт выбора команды и переноса игрока.

Отображение выбранной команды на экране.

Практика: сборка и тестирование команды выбора.

По окончании урока слушатели выполняют домашнее задание.

Урок №13: Добавление анимаций для UI-элементов

Что такое анимация UI и как она улучшает восприятие.

Инструменты Roblox для создания анимации: TweenService.

Примеры: появление кнопок, плавное перемещение, увеличение при наведении.

Скрипт анимации с использованием TweenInfo и событий.

Практика: добавление анимаций к кнопкам и панелям.

По окончании урока слушатели выполняют домашнее задание.

Урок №14: Создание внутриигрового магазина

Зачем нужен магазин в игре.

Размещение интерфейса магазина: панель, кнопки покупки, названия предметов.

Настройка цен и связка с внутриигровой валютой.

Обработка покупки с помощью событий и скриптов.

Практика: добавление кнопки покупки предмета.

По окончании урока слушатели выполняют домашнее задание.

Урок №15: Настройка валюты и тестирование внутриигровой экономики

Создание системы игровой валюты (монеты, кристаллы и др.).

Хранение и отображение количества валюты на экране.

Добавление валюты за действия (например, победу или сбор предметов).

Проверка работы покупок и вычитания средств.

Практика: начисление и трата валюты в магазине.

По окончании урока слушатели выполняют домашнее задание.

Урок №16: Система инвентаря и предметов

Что такое инвентарь и как он помогает игроку.

Создание интерфейса инвентаря с ячейками для предметов.

Добавление предметов в инвентарь при покупке или сборе.

Просмотр и выбор предметов из инвентаря.

Практика: создание простого интерфейса инвентаря с 3–5 слотами.

По окончании урока слушатели выполняют домашнее задание.

Модуль №3: Разработка многопользовательских игр

Урок №17: Создание квеста. Часть 1

Что такое квест и зачем он нужен в игре.

Планирование простого задания для игрока (например, найти предмет).

Создание NPC (персонажа), который выдаёт задание.

Добавление диалога и интерфейса с описанием квеста.

Практика: игрок получает задание и видит его цель.

По окончании урока слушатели выполняют домашнее задание.

Урок №18: Создание квеста. Часть 2

Проверка условий выполнения задания (например, найден ли предмет).

Настройка логики завершения квеста.

Обновление интерфейса при выполнении задания.

Вознаграждение игрока за выполнение (опыт, предмет, валюта).

Практика: завершение простого квеста и получение награды.

По окончании урока слушатели выполняют домашнее задание.

Урок №19: Создание квеста. Часть 3

Добавление нескольких шагов в задание (собрать, отнести, передать).

Настройка последовательности выполнения шагов.

Интерфейс слежения за прогрессом квеста.

Практика: создание цепочки из 2–3 этапов в одном квесте.

По окончании урока слушатели выполняют домашнее задание.

Урок №20: Викторина-игра. Часть 1

Планирование структуры викторины: вопросы, ответы, баллы.
Создание интерфейса: вопрос, варианты, кнопки.
Скрипт показа первого вопроса.
Практика: первый вопрос с обработкой ответа.
По окончании урока слушатели выполняют домашнее задание.

Урок №21: Викторина-игра. Часть 2

Переход к следующему вопросу.
Начисление баллов за правильные ответы.
Сообщения о правильности ответа.
Практика: 3 вопроса с переходом между ними.
По окончании урока слушатели выполняют домашнее задание.

Урок №22: Викторина-игра. Часть 3

Завершение игры: подсчёт очков и финальное сообщение.
Перезапуск викторины с начала.
Сохранение лучшего результата.
Практика: полноценная викторина из 5 вопросов.
По окончании урока слушатели выполняют домашнее задание.

Урок №23: Магазин с покупками и валютой. Часть 1

Подробная настройка валюты и отображения баланса.
Создание категории товаров: инструменты, предметы.
Создание UI-магазина с вкладками и описаниями.
Практика: добавление 2 товаров с разной ценой.
По окончании урока слушатели выполняют домашнее задание.

Урок №24: Магазин с покупками и валютой. Часть 2

Обработка покупки и проверка наличия средств.
Добавление предмета в инвентарь после покупки.
Обновление баланса игрока.
Практика: купить предмет и использовать его в игре.
По окончании урока слушатели выполняют домашнее задание.

Модуль №4: Финальные проекты и публикация

Урок №25: Тусооп. Часть 1

Что такое жанр Тусооп и его особенности.
Создание стартовой базы и первой кнопки покупки.
Скрипт нажатия и появления нового объекта.
Практика: строительство первого элемента фабрики.
По окончании урока слушатели выполняют домашнее задание.

Урок №26: Тусооп. Часть 2

Добавление новых улучшений: кнопки покупки зданий и объектов.
Настройка цен и логики покупки.
Создание визуального прогресса развития базы.
Практика: добавление второго и третьего уровня базы.
По окончании урока слушатели выполняют домашнее задание.

Урок №27: Тусооп. Часть 3

Начисление игровой валюты за время или действия.

Автоматический сбор дохода с помощью механизмов.
Создание визуального счётчика ресурсов.
Практика: реализация автоматического заработка.
По окончании урока слушатели выполняют домашнее задание.

Урок №28: Тусооп. Часть 4

Добавление декора и эффектов для улучшения внешнего вида.
Оптимизация скриптов и устранение ошибок.
Финальная сборка и тестирование Тусооп-игры.
Практика: презентация своей базы и её возможностей.
По окончании урока слушатели выполняют домашнее задание.

Урок №29: Tower Defense. Часть 1

Что такое жанр Tower Defense и его правила.
Создание карты и маршрута движения врагов.
Появление волны врагов по таймеру.
Практика: первая волна врагов движется по маршруту.
По окончании урока слушатели выполняют домашнее задание.

Урок №30: Tower Defense. Часть 2

Добавление башен: установка, внешний вид, область действия.
Скрипт наведения башни на ближайшего врага.
Практика: башня стреляет по врагам.
По окончании урока слушатели выполняют домашнее задание.

Урок №31: Tower Defense. Часть 3

Начисление валюты за уничтожение врагов.
Добавление улучшений для башен (урон, радиус, скорость).
Интерфейс прокачки башни.
Практика: башни становятся сильнее при апгрейде.
По окончании урока слушатели выполняют домашнее задание.

Урок №32: Tower Defense. Часть 4

Добавление разных типов врагов (быстрые, бронированные).
Настройка интерфейса: волна, жизни базы, результат.
Финальное тестирование мини-игры.
Практика: завершение полноценной Tower Defense-игры.
По окончании урока слушатели выполняют домашнее задание.

Урок №33: Работа с чатом. Часть 1

Что такое чат в Roblox и как его настроить.
Обработка сообщений от игроков.
Создание простых команд в чате (например, /help, /dance).
Практика: написание и проверка своих команд.
По окончании урока слушатели выполняют домашнее задание.

Урок №34: Работа с чатом. Часть 2

Создание кастомных визуальных сообщений в чате.
Цветной текст, подписи, отображение имени игрока.
Отправка системных сообщений от NPC или игры.
Практика: чат-уведомления о событиях (начало раунда, победа и т.д.).
По окончании урока слушатели выполняют домашнее задание.

Урок №35: Итоговый проект. Часть 1

Выбор идеи и жанра финальной игры.

Сборка основных механик: карта, интерфейс, логика игры.

Настройка внешнего вида и взаимодействия объектов.

Практика: начало разработки собственного проекта.

По окончании урока слушатели продолжают проект дома.

Урок №36: Итоговый проект. Часть 2

Тестирование и доработка проекта.

Исправление ошибок, улучшение внешнего вида и UX.

Подготовка презентации проекта: рассказ о функционале.

Финальная защита и демонстрация игры перед группой.

По окончании урока учащиеся получают обратную связь и оценки.

Итоговая аттестация

Тестирование

2.4. Кадровое обеспечение

Реализация программы обеспечивается педагогическими кадрами, имеющими высшее профессиональное образование и отвечающими квалификационным требованиям, указанным в квалификационных справочниках, и (или) профессиональным стандартам, в рамках изучаемого цикла.

2.5. Материально-техническое и программное обеспечение реализации программы

ООО «ТОЧКА ЗНАНИЙ» располагает материально-технической базой, обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, итоговой аттестации слушателей, предусмотренных учебным планом:

Место работы преподавателя	<ul style="list-style-type: none">✓ Стол компьютерный/письменный 120x60 - 1 шт;✓ Кресло ARDOR GAMING Gun Shield 10B - 1 шт;✓ ИБП Dexp IEC Plus LCD 3000VA - 1 шт;✓ Переходник от ИБП к сетевому фильтру - 1 шт;✓ Сетевой фильтр с USB разъемами - 1 шт;✓ Софтбокс 50x70 600BT - 1 шт;✓ Лампа студийная для софтбокса 135W E27 5500K, Холодный белый свет, 135 Вт, Люминесцентная (энергосберегающая) - 4шт (запасные);✓ Предохранители в софтбокс 3- 5А, 250V - 5 штук;✓ Монитор 21.45" DEXP DF22N2 черный - 1 шт;✓ Микрофон петличный VOYA BY-M1 - 1 шт;✓ Apple iPad 10.2" (2021) Wi-Fi 64Gb Silver - 1 шт;✓ Apple MacBook Pro 16", M1, 16ГБ, 2021г - 1 шт;✓ Стилус WiWU Pencil Pro - 1 шт;✓ Хаб Baseus (4K@60Hz 8-in-1 STARJOY 8-PORT TYPE-C) - 1 шт;✓ Патч-корд RJ 45 - 1 шт;✓ Футболки (мерч) "Точка знаний" - 2 шт.
----------------------------	---

Программное обеспечение: лицензионные системные программы, обеспечивающие взаимодействие всех других программ с оборудованием и взаимодействие пользователя персонального компьютера с программами. Универсальные офисные прикладные программы и средства ИКТ, например, программа подготовки презентаций; использование Интернета, электронной почты; использование автоматизированных поисковых систем Интернета.

Информационно-образовательная среда включает в себя образовательную LMS «Точка Знаний».

Образовательная LMS «Точка Знаний» обеспечивает через Интернет доступ к:

- электронным информационным и образовательным ресурсам ООО «ТОЧКА ЗНАНИЙ»;
- доступ к нормативным и организационно-методическим документам, регламентирующим образовательный процесс в ООО «ТОЧКА ЗНАНИЙ»;

- систему электронного учёта слушателей; (см. [Положение](#))
- взаимодействие слушателей с преподавателями, организаторами образовательного процесса и администрацией ООО «ТОЧКА ЗНАНИЙ». (см. [Положение](#))

2.6. Форма аттестации и оценочные материалы

Программой предусмотрен текущий контроль в виде домашнего задания и тестов, размещенных в уроках на образовательной LMS «Точка Знаний», промежуточный контроль в виде тестов, согласно учебному плану.

Итоговая аттестация проводится в виде теста. Итоговый тест включает в себя 20 вопросов. Для успешного прохождения теста необходимо правильно выполнить не менее 85 % заданий.

Обучающиеся допускаются к итоговой аттестации после изучения Модулей программы в объеме, предусмотренном учебным планом программы.

Оценка качества освоения учебной программы проводится в процессе итоговой аттестации в форме тестирования.

Оценка	Критерии
Зачтено	Оценка «Зачтено» выставляется обучающемуся, продемонстрировавшему твердые и всесторонние знания материалы, умение применять полученные в рамках занятий практические навыки и умения. Достижения за период обучения и результаты текущей аттестации демонстрировали отличный уровень знаний и умений обучающегося. Не менее 85% правильных ответов при решении теста.
Не зачтено	Оценка «Не зачтено» выставляется обучающемуся, который в недостаточной мере овладел теоретическим материалом по дисциплине, допустил ряд грубых ошибок при выполнении практических заданий, а также не выполнил требований, предъявляемых к промежуточной аттестации. Достижения за период обучения и результаты текущей аттестации демонстрировали неудовлетворительный уровень знаний и умений обучающегося. Менее 80% правильных ответов при решении теста.

Примеры оценочных материалов текущего контроля

ДЗ №9: основы интерфейсов в Roblox

Задание 1

Вам нужно создать игру, в которой при касании определённого объекта (Part) на экране появляется сообщение в виде интерфейса (UI).

Пошаговая инструкция

1. Открой Roblox Studio и создай новый пустой проект.
2. В **Workspace** добавь Part — это будет объект, который игрок должен коснуться.
3. В **StarterGui** добавь ScreenGui, а внутри него — Frame.
 - Назови ScreenGui как хочешь, например, TouchMessageGui.
 - В Frame добавь TextLabel с текстом, например: «Добро пожаловать!»
 - Включи TextScaled, чтобы текст автоматически подстраивался.
 - Установи Visible = false, чтобы UI изначально был скрыт.
4. Вставь следующий скрипт внутрь Part:

```
lua
Копировать Редактировать
local part = script.Parent

part.Touched:Connect(function(hit)
```

```

    local player = game.Players:GetPlayerFromCharacter(hit.Parent)
    if player then
        local gui =
player:WaitForChild("PlayerGui"):FindFirstChild("TouchMessageGui")
        if gui then
            gui.Enabled = true
            gui.Frame.Visible = true
        end
    end
end)

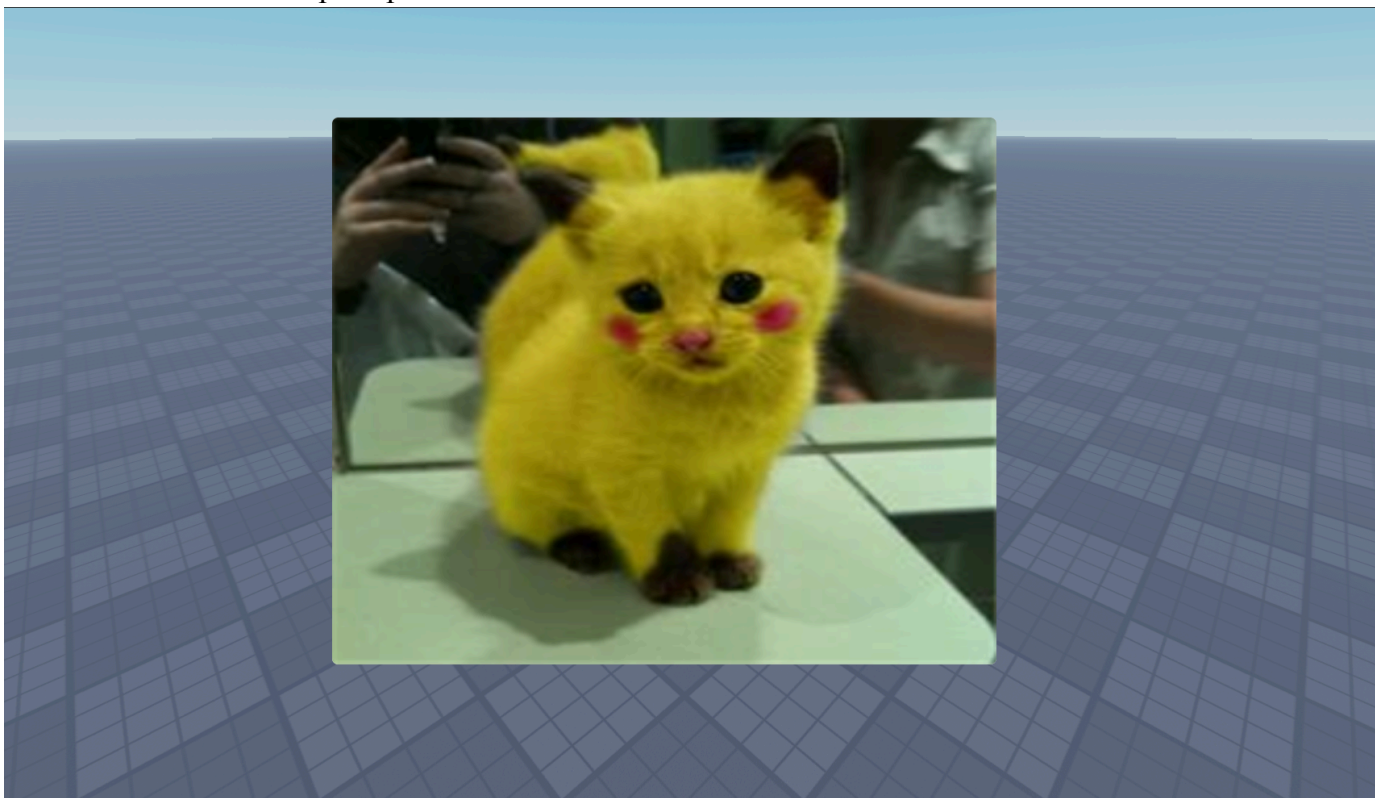
```

5. Запусти игру (кнопка **Play**) и коснись Part — интерфейс должен появиться.

Что должно получиться

- При касании объекта на экране появляется приветственное окно.
- Если у тебя всё работает правильно — UI отображается только при первом касании.

Возможный итоговой пример:



Примеры оценочных материалов промежуточного контроля

Вопрос 1. Какой тип переменной в Lua используется для хранения целых чисел?

1. string
2. integer (number)
boolean
3. table
4. nil

Вопрос 2. Какая функция в Lua запускается при касании объекта игроком?

1. Touched
2. ClickDetector
3. OnTouch
4. PlayerTouch

5. TouchEvent

Вопрос 3. Как в Roblox Studio привязать скрипт к конкретному объекту?

1. Перетащить скрипт в Workspace
2. Вставить скрипт внутрь объекта через Insert Object
3. Открыть Explorer и выбрать File → Attach Script
4. Использовать Toolbox
5. Настроить в Properties

Вопрос 4. Какой тип скрипта используется для работы с клиентской частью (интерфейс, эффекты)?

1. Script
2. LocalScript
3. ModuleScript
4. TouchScript
5. UIScript

Вопрос 5. Как вызвать сообщение в консоли во время выполнения скрипта?

1. alert("Hello")
2. show("Hello")
3. print("Hello")
4. debug("Hello")
5. message("Hello")

Вопрос 6. Как в Lua создать переменную с текстовым значением?

1. local name = "Player"
2. var name = Player
3. name := "Player"
4. string name = "Player"
5. set name = "Player"

Вопрос 7. Какой объект в Roblox используется для воспроизведения звука?

1. Sound
2. Audio
3. Music
4. SoundTrack
5. Speaker

Вопрос 8. Как сделать, чтобы объект не падал под действием физики?

1. Включить свойство Anchored
2. Выключить свойство Visible
3. Включить свойство Collision
4. Переместить объект в Lighting
5. Вставить скрипт с функцией AntiGravity

Вопрос 9. Какое событие в Lua используется для отслеживания нажатия кнопки игроком в интерфейсе?

1. ButtonClick
2. Click

3. MouseClick
4. Activated
5. Pressed

Вопрос 10. Как в Roblox Studio организовать многократное использование кода в разных скриптах?

1. Использовать LocalScript
2. Дублировать код вручную
3. Создать ModuleScript
4. Вставить код в Properties
5. Применить Toolbox

Примеры оценочных материалов итоговой аттестации

Вопрос 1. Какой тип скрипта в Roblox Studio используется для написания логики, выполняемой только на сервере?

1. LocalScript
2. Script
3. ModuleScript
4. UIScript
5. ServerLocal

Вопрос 2. Как в Lua правильно записать условие, проверяющее, что переменная `coins` больше 10?

1. `if coins > 10 then`
2. `if coins ==> 10 then`
3. `if coins >= 10 then`
4. `if coins more 10 then`
5. `if coins.gt(10) then`

Вопрос 3. Какой тип переменной в Lua может хранить сразу несколько значений в виде набора?

1. string
2. table
number
3. array
4. boolean

Вопрос 4. Какое событие используется для обнаружения нажатия клавиши на клавиатуре игроком?

1. KeyPress
2. KeyDown
3. InputBegan
4. KeyboardEvent
5. OnKey

Вопрос 5. Как в Roblox Studio воспроизвести анимацию у персонажа?

1. Вставить анимацию в Workspace
2. Использовать AnimationController и Animator
3. Включить свойство PlayAnimation в Properties
4. Перетащить анимацию в StarterGui
5. Настроить её в Lighting

Вопрос 6. Что делает метод WaitForChild() в Lua?

1. Ожидает появления объекта в иерархии перед использованием
2. Приостанавливает игру на 1 секунду
3. Запускает функцию после задержки
4. Ожидает загрузки скрипта
5. Ждёт завершения анимации

Вопрос 7. Какой объект используется для организации взаимодействия между сервером и клиентом?

1. ClickDetector
2. RemoteEvent
3. LocalScript
4. BindableFunction
5. PlayerHandler

Вопрос 8. Как в Lua создать цикл, который выполнит код 5 раз?

1. for i = 1, 5 do ... end
2. loop 5 times ... end
3. repeat 5 do ... end
4. while i < 5 do ... end
times(5) do ... end

Вопрос 9. Какой инструмент в Roblox Studio используется для настройки коллизий между объектами?

1. Move
2. Scale
3. Collision Groups
4. Anchored Tool
5. Physics Editor

Вопрос 10. Как в Roblox Studio можно оптимизировать игру при большом количестве объектов?

1. Удалить все скрипты
2. Объединять части в модели и использовать Anchor
3. Перемещать все объекты в StarterGui
4. Использовать только LocalScripts
5. Вставить все объекты в Lighting

Вопрос 11. Как в Lua объединить две строки в одну?

1. str1 + str2
2. str1 .. str2
3. concat(str1, str2)
4. merge(str1, str2)
5. str1 & str2

Вопрос 12. Какой объект используется для хранения и воспроизведения анимаций у персонажей в Roblox?

1. Animator
2. AnimationPlayer

3. MotionHandler
4. AnimationController
5. CharacterAnimator

Вопрос 13. Какое свойство объекта Part отвечает за его прозрачность?

1. Transparency
2. Visible
3. Alpha
4. Opacity
5. SeeThrough

Вопрос 14. Как в Lua создать функцию с именем `jumpBoost`?

1. `func jumpBoost()`
2. `function jumpBoost() end`
3. `def jumpBoost() end`
4. `jumpBoost = func() end`
5. `new jumpBoost()`

Вопрос 15. Какое событие используется для обнаружения клика по объекту в 3D-сцене?

1. Click
2. MouseClick
3. Activated
4. Pressed
5. OnClick

Вопрос 16. Как в Roblox Studio можно сохранить данные игрока между сессиями?

1. DataStoreService
2. SaveDataModule
3. PlayerStorage
4. GameSettings
5. Workspace

Вопрос 17. Какой тип переменной в Lua может принимать только значения `true` или `false`?

1. bool
2. boolean
3. bit
4. logic
5. binary

Вопрос 18. Какой метод используется для воспроизведения звука в Roblox?

1. `Sound:Start()`
2. `Sound:Play()`
3. `Sound:Run()`
4. `Sound:Begin()`
5. `Sound:Activate()`

Вопрос 19. Как в Roblox Studio можно сделать, чтобы объект двигался по определённому пути?

1. PathfindingService

2. MoveTool
3. AutoWalkModule
4. PlayerPath
5. MovementHandler

Вопрос 20. Как в Lua приостановить выполнение кода на 2 секунды?

1. delay(2)
2. wait(2)
3. sleep(2)
4. pause(2)
5. timer(2)

2.7. Список рекомендованной литературы

1. Документация по Roblox <https://create.roblox.com/docs>

2.8. Список использованной литературы

2.8.1. Электронные ресурсы

1. Электронная библиотека <https://elibrary.ru/>
2. Мадтест-онлайн конструктор тестов <https://onlinetestpad.com/>
3. Среда разработки для создания игр <https://create.roblox.com/>