

Общество с ограниченной ответственностью

«Проектная школа - Самара»

443015 г. Самара, ул. Главная, д.3, оф.20, тел. 8(927)777-09-67

E-mail: mironov@100lines.ru

УТВЕРЖДАЮ

Директор

ООО «Проектная школа — Самара»

Е.Н.Миронов

15 декабря 2021 г.



ДОПОЛНИТЕЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ

(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА

«СОЗДАНИЕ ИГР НА ROBLOX»

Возраст детей: от 9 лет

Срок обучения: 72 академических часа

Автор-составитель:

Мальцев С.А., педагог ДО

г.о. Самара, 2021

Введение

Программа «Создание игр на Roblox» разработана в соответствии с требованиями к дополнительным общеобразовательным общеразвивающим программам технической направленности и направлена на формирование у детей подросткового возраста навыков текстового программирования на языке Lua, основ 3D-моделирования и геймдизайна в среде Roblox Studio.

Платформа Roblox является не просто игрой, а мощным инструментом для создания пользовательского контента (UGC). Среда разработки Roblox Studio позволяет создавать полноценные 3D-миры, используя профессиональные инструменты левел-дизайна и скриптинга. Переход от визуального программирования (как в Scratch) к текстовому коду (Lua) является важным шагом в IT-образовании, развивающим внимательность к синтаксису и структуре данных.

Актуальность программы обусловлена высоким интересом подростков к платформе Roblox и необходимостью трансформации этого интереса из потребительского в созидательный. Программа отвечает социальному заказу на подготовку кадров, владеющих основами современной игровой индустрии, включая работу с 3D-пространством, сетевыми взаимодействиями и объектно-ориентированным подходом.

1. Пояснительная записка

Актуальность программы. В современной цифровой экономике навыки создания 3D-контента и понимания логики работы многопользовательских приложений становятся всё более востребованными. Roblox Studio предоставляет уникальную возможность изучить реальный язык программирования Lua в контексте создания визуально привлекательных продуктов. Программа актуальна, так как позволяет детям освоить переход от алгоритмического мышления к инженерному, работая с физическими движениями, событиями сервера и клиента, а также основами UI/UX дизайна в 3D.

Отличительные особенности программы. Программа сочетает в себе элементы классического программирования и современного геймдева. Особенностью является разделение логики на «Клиент» (LocalScript) и «Сервер» (Script), что формирует понимание архитектуры клиент-серверных приложений. Также уделяется внимание работе с Terrain Editor (ландшафтный дизайн) и Toolbox (библиотека ассетов), обучая детей легальному использованию чужих ресурсов и созданию своих собственных моделей.

Учебные дисциплины:

Основы алгоритмизации и синтаксиса Lua: переменные, типы данных, условия, циклы, функции, таблицы. 3D-моделирование и Левел-дизайн: работа с примитивами

(Parts), свойствами объектов, освещение, материалы, ландшафт. Основы геймдизайна: проектирование механик, баланс экономики игры, удержание игрока (Retention), монетизация (Game Passes).

Практические дисциплины:

Скриптинг в Roblox Studio: написание кода для управления персонажем, взаимодействия с объектами, создания интерфейсов. Работа с физическим движком: настройка коллизий, гравитации, сил (BodyMovers). Проектная деятельность: разработка, тестирование (Play Solo/Test Server) и публикация собственной игры.

Цель образовательной программы: формирование у обучающихся компетенций в области разработки 3D-игр, освоение основ текстового программирования на языке Lua и создание собственного функционального игрового проекта в среде Roblox Studio.

Задачи программы:

Обучающие:

- познакомить с интерфейсом Roblox Studio, структурой Explorer и Properties;
- научить основам синтаксиса языка Lua (переменные, функции, таблицы, модули);
- сформировать понимание различий между Client-side и Server-side скриптами;
- обучить созданию интерактивных объектов, систем подсчета очков, здоровья и инвентаря;
- научить работать с GUI (Graphical User Interface) для создания меню и HUD.

Воспитывающие:

- воспитывать культуру безопасного поведения в онлайн-среде и этику сетевого общения;
- формировать уважение к авторскому праву при использовании ассетов из Toolbox;
- воспитывать усидчивость при отладке сложного кода и работе с 3D-пространством.

Развивающие:

- развивать пространственное мышление и навыки 3D-композиции;
- развивать логическое мышление через решение задач на оптимизацию кода и производительности;
- развивать навыки проектного менеджмента: от идеи до релиза.

2. Принципы, формы и методы обучения

Основными **принципами обучения** являются:

1. Научность. Этот принцип предопределяет сообщение учащимся только достоверных, проверенных практикой сведений. Даже в игровой среде Roblox Studio преподаются

настоящие концепции программирования на языке Lua (переменные, функции, таблицы, события), которые являются базой для профессиональной разработки ПО и игр.

2. Доступность. Предусматривает соответствие объема и глубины учебного материала уровню общего развития учащихся. Сложные технические концепции (например, клиент-серверная архитектура) объясняются через жизненные аналогии и визуальные образы интерфейса Studio, что снижает порог входа в текстовое программирование.

3. Связь теории с практикой. Обязывает вести обучение так, чтобы учащиеся могли сознательно применять приобретенные ими знания на практике. Каждая изученная конструкция языка Lua немедленно применяется для создания конкретной игровой механики (движение, взаимодействие, подсчет очков).

4. Воспитательный характер обучения. Процесс обучения является воспитывающим: учащийся не только приобретает технические знания, но и развивает усидчивость, умение преодолевать трудности при поиске синтаксических ошибок (багов), а также морально-духовные качества, такие как ответственность за создаваемый цифровой контент.

5. Активность обучения. В процессе обучения все действия должны быть обоснованы. Активность предполагает самостоятельность: ребенок не просто копирует код за педагогом, а понимает логику работы скрипта и может модифицировать его под свои задачи.

6. Наглядность. Объяснение техники программирования на конкретных готовых изделиях (работающих играх). Интерфейс Roblox Studio, окно Properties и визуализация кода в редакторе служат мощными наглядными пособиями, позволяющими видеть изменения свойств объектов в реальном времени.

7. Систематичность и последовательность. Учебный материал дается по определенной системе: от простого манипулирования 3D-объектами к написанию линейных скриптов, затем к условиям, циклам и сложным взаимодействиям между объектами. От частного к общему.

8. Индивидуальный подход в обучении. Педагог исходит из индивидуальных особенностей учащегося (темп набора текста, уровень логического мышления, интерес к дизайну или коду) и, опираясь на сильные стороны, доводит его подготовленность до уровня общих требований с перспективой на творческое опережение.

Формы организации образовательного процесса: сочетание фронтально-групповой и индивидуальной форм организации учебной деятельности (групповое обсуждение идей геймдизайна, индивидуальное выполнение практических заданий по написанию скриптов и сборке уровней, коллективное тестирование и оценка полученных результатов в режиме Multiplayer).

Методы обучения:

1. Объяснительно-иллюстративный – предоставление информации (объяснения синтаксиса Lua, рассказ об архитектуре Roblox, инструктаж по инструментам Studio, работа с технологическими картами, демонстрация готовых игровых механик).
2. Эвристический – метод творческой деятельности (придумывание собственных уникальных игровых правил, создание нестандартных уровней, поиск оригинальных способов применения стандартных инструментов).
3. Проблемный – постановка проблемы (например, «почему персонаж проваливается сквозь пол?») или «как сделать так, чтобы очки сохранялись после выхода из игры?») и самостоятельный поиск ее решения через изучение документации и эксперименты с кодом.
4. Программированный – набор операций, который необходимо выполнить в ходе практических работ по пошаговым алгоритмическим картам (написание шаблонных скриптов для типовых задач).
5. Репродуктивный – воспроизведение знаний и способов деятельности (написание скриптов по образцу педагога, упражнения по аналогу для закрепления синтаксиса).
6. Поисковый – решение проблемных задач с помощью педагога и самостоятельный поиск оптимального алгоритма или способа оптимизации кода.

Главный метод, который используется при изучении программирования – это метод проектов. Основные этапы разработки проекта – игры в Roblox:

1. Освоение теоретического базиса (изучение новых функций API Roblox и конструкций Lua), необходимого для дальнейшей продуктивной работы;
2. Определение идеи проекта (жанр игры, ключевая механика, целевая аудитория);
3. Цель и задачи представляемого проекта (что должен уметь делать игрок, какова цель игры);
4. Создание 3D-графики (левел-дизайн), подбор звуков и написание кода (скриптинг);
5. Детальный разбор принципа работы полученного алгоритма и архитектуры игры;
6. Тестирование (в том числе в режиме сервера), устранение неисправностей (отладка/дебаггинг) и доработка проекта перед публикацией.

3. Механизм отслеживания результатов

Контроль освоения учебного материала учащимися проходит в три этапа:

1. Входной мониторинг сформированности базовых навыков работы с ПК, понимания логики алгоритмов и опыта взаимодействия с платформой Roblox (устный опрос, мини-

тест на знание базовых концепций программирования, анкетирование об опыте игры в Roblox).

2. Проведение промежуточных (текущих) контрольных тестов в вопросно-ответной форме и в виде защиты мини-проектов (например, защита игры «Obby» после изучения 3D-объектов и событий касания, или «Clicker» после изучения переменных и GUI).

3. Итоговый мониторинг сформированности компетенций учащихся; создание, публикация и публичная защита собственных итоговых проектов (полноценных 3D-игр) на основе пройденного материала с демонстрацией исходного кода на Lua.

В конце курса проводится анализ качества данной программы (содержания, методической подачи и организационных моментов) и, по необходимости, проводится коррекция программы для следующего набора.

4. Прогнозируемый результат

В конце обучения по данному курсу ученики должны знать:

- общенаучные и технические базовые термины информатики и геймдева (алгоритм, скрипт, баг, отладка, интерфейс, рендеринг, физический движок);
- элементную базу среды Roblox Studio (Part, Mesh, Script, LocalScript, ModuleScript, Explorer, Properties, Toolbox);
- категории объектов и их назначение (Workspace, Players, Lighting, SoundService, ReplicatedStorage);
- основы декартовой системы координат в 3D-пространстве (оси X, Y, Z) и принцип работы с позиционированием объектов;
- порядок взаимодействия клиента и сервера через события (Events) и RemoteEvents;
- назначение и принцип работы конструкций языка Lua: переменных, условий (if/then/else), циклов (for, while) и функций;
- назначение и принцип работы таблиц (Tables) для хранения данных и списков объектов;
- правила цифровой гигиены, этики поведения в онлайн-среде и техники безопасности при работе за компьютером.

В конце обучения по данному курсу ученики должны уметь:

- создавать интерактивные 3D-проекты и игры с использованием текстового программирования на языке Lua;
- управлять свойствами объектов (цвет, прозрачность, размер, материал) и их поведением с помощью алгоритмов;
- пользоваться логическими операторами и условиями для создания игровой логики (проверка столкновений, триггеры);

- создавать и использовать переменные для подсчета очков, управления здоровьем персонажа и таймерами;
- использовать инструменты Terrain Editor и Parts для создания ландшафта и уровней (левел-дизайн);
- применять GUI (Graphical User Interface) для создания меню, кнопок и интерфейса игрока (HUD);
- самостоятельно находить и исправлять синтаксические и логические ошибки в своем коде, используя окно Output (отладка);
- презентовать свою работу, аргументировать принятые проектные решения и публиковать готовый проект на платформе.

5. Учебно-тематический план обучения

№ п/п	Тема	часы		
		всего	теория	практ.
1	Вводное занятие. Техника безопасности. Обзор Roblox Studio. Интерфейс.	2	0,5	1,5
2	Тема 1. Мир Roblox. Части (Parts) и их свойства. Координаты X, Y, Z.	2	0,5	1,5
3	Тема 2. Инструменты трансформации. Move, Scale, Rotate. Группировка объектов.	2	0,5	1,5
4	Тема 3. Материалы и цвета. Создание простого уровня (карта).	2	0,5	1,5
5	Тема 4. Введение в Lua. Переменные и типы данных. Print и Output.	2	1,0	1,0
6	Тема 5. Скрипты vs LocalScripts. Где писать код? Иерархия Explorer.	2	1,0	1,0
7	Тема 6. Управление объектами через код. Изменение свойств (Color, Transparency).	2	0,5	1,5
8	Тема 7. События (Events). Событие Touched. Создание ловушек.	2	0,5	1,5
9	Тема 8. Условные операторы (if/then/else). Логика проверки условий.	2	1,0	1,0
10	Тема 9. Мини-проект «Obby» (Полоса препятствий).	2	0,2	1,8

	Сборка и скрипты чекпоинтов.			
11	Тема 10. Циклы (while, for). Спавн объектов.	2	0,5	1,5
12	Тема 11. Функции. Создание переиспользуемого кода.	2	0,5	1,5
13	Тема 12. ClickDetector. Взаимодействие с игроком по клику.	2	0,5	1,5
14	Тема 13. Введение в GUI. ScreenGui, Frame, TextLabel.	2	0,5	1,5
15	Тема 14. Скриптинг GUI. Кнопки и обработка нажатий.	2	0,5	1,5
16	Тема 15. Переменные в игре. Счетчик очков и жизней.	2	0,5	1,5
17	Тема 16. Мини-проект «Clicker» (Кликер). Базовая экономика.	2	0,2	1,8
18	Тема 17. Таблицы (Tables) в Lua. Хранение списков данных.	2	1,0	1,0
19	Тема 18. Работа со звуком и эффектами. ParticleEmitter.	2	0,5	1,5
20	Тема 19. Освещение и атмосфера. Skybox, Fog, Lighting.	2	0,5	1,5
21	Тема 20. Инструменты (Tools). Создание оружия или предметов в руки.	2	0	2,0
22	Тема 21. Анимация персонажа. Animation Editor.	2	0,5	1,5
23	Тема 22. NPC и искусственный интеллект. PathfindingService.	2	0,5	1,5
24	Тема 23. RemoteEvents. Общение Клиента и Сервера (безопасность).	2	0,5	1,5
25	Тема 24. DataStore Service. Сохранение прогресса игрока.	2	0,5	1,5
26	Тема 25. Магазин внутри игры. Game Passes и Developer Products.	2	0,5	1,5
27	Тема 26. Основы геймдизайна. Баланс и удержание игрока.	2	0,5	1,5
28	Тема 27. Оптимизация игры. Union, Negate, снижение лагов.	2	0,5	1,5
29	Тема 28. Итоговый проект. Выбор жанра и прототипирование.	2	0,2	1,8
30	Тема 29. Итоговый проект. Создание уровня и ассетов.	2	0	2,0
31	Тема 30. Итоговый проект. Написание основной логики (скрипты).	2	0	2,0

32	Тема 31. Итоговый проект. UI, звук, эффекты.	2	0	2,0
33	Тема 32. Итоговый проект. Тестирование и отладка (Bug fixing).	2	0	2,0
34	Тема 33. Подготовка к публикации. Настройка места (Place).	2	0,5	1,5
35	Тема 34. Защита итоговых проектов. Презентация игр.	2	0	2,0
36	Тема 35. Рефлексия. Подведение итогов курса.	2	0	2,0
	ИТОГО	72	15,6	56,4

6. Обеспечение программы

Для организации образовательного процесса по программе необходимы следующие ресурсы.

Кадровые: Специалист, имеющий опыт разработки в Roblox Studio, владеющий языком Lua, понимающий специфику платформы и возрастную психологию подростков.

Материально-технические:

Компьютерный класс с ПК или ноутбуками (минимальные требования: CPU i3/i5, 8GB RAM, дискретная видеокарта желательно). Стабильный высокоскоростной доступ в Интернет (для загрузки ассетов и публикации). Наушники с микрофоном (для тестирования звука и коммуникации). Аккаунты в Roblox для каждого учащегося (или учебные аккаунты).

Программно-методические:

1. Установленная среда Roblox Studio.
2. Аккаунт разработчика в Roblox.
3. Методическое пособие с примерами кода Lua.
4. Шаблоны проектов (Baseplate, Obby Template).
5. Доступ к официальной документации Roblox Creator Hub.

7. Литература

Нормативно-правовая база:

1. Федеральный закон РФ «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
2. СанПиН 2.4.3648-20.

Для педагога (учебная и методическая литература):

3. Официальная документация Roblox Creator Hub: <https://create.roblox.com/docs>
4. AlvinBlox. Roblox Lua Scripting Tutorial Series (YouTube/Website).
5. TheDevKing. Roblox Studio Tutorials.
6. Геймдизайн. Как создать игру, в которую будут играть все. Джесси Шелл.

Для обучающихся:

7. Справочник по языку Lua (Lua 5.1 Reference Manual).
8. Интерактивные курсы на платформах типа CodeCombat (для базы) и специализированные курсы по Roblox.

Электронные ресурсы:

9. Roblox Developer Forum: <https://devforum.roblox.com/>
10. Библиотека бесплатных моделей и скриптов: Roblox Toolbox.