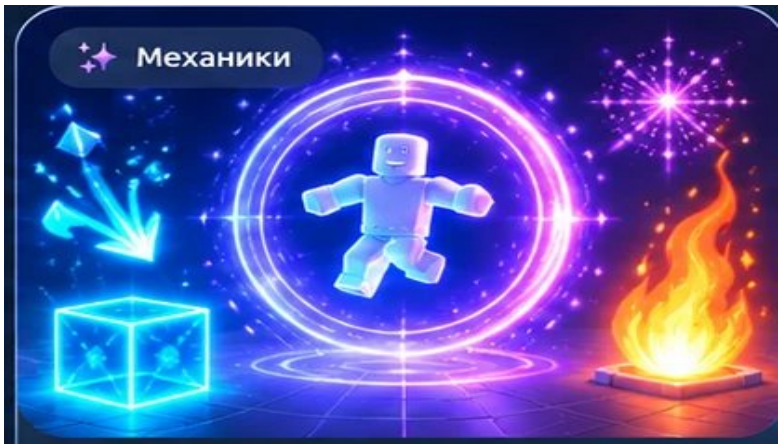
Эффекты и анимации: визуальная мастерская

Учебное руководство по шести эффектам и пяти анимациям объектов



ГЛАВА 01 / 10 — Зачем игре визуальная обратная связь

Теория и смысл

Эффекты помогают игроку понять, что произошло: награда собрана, дверь открыта, портал активен. В наборе шесть эффектов частиц и пять анимаций деталей. Они украшают игру, но не заменяют логику урона, телепортации или наград.

Практика

Составьте таблицу из пяти игровых событий и подберите каждому один визуальный сигнал. Отметьте, какие сигналы должны быть заметны сразу, а какие можно сделать ненавязчивыми.

Проверка

Попросите другого ученика угадать смысл эффекта без текста.

Самостоятельное задание

Придумайте единую палитру: голубой для воды, зелёный для лечения, золотой для победы.

Важно

Не используйте слишком яркие мигающие эффекты и плотные частицы, мешающие управлению.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 02 / 10 — Установка модулей в Roblox Studio

Теория и смысл

Архив содержит Modules/Effects.lua, Modules/Animations.lua и Demo/Demo.server.lua. Модули нужны для повторного использования функций, а демонстрационный скрипт строит площадку. Сами файлы не являются готовой моделью .rbxm.

Практика

Создайте ServerScriptService/IgrovenokResources/Modules. Внутри Modules создайте ModuleScript Effects и Animations. Рядом с папкой Modules создайте Script Demo. Скопируйте содержимое соответствующих файлов.

Проверка

После Play в Workspace должна появиться папка IgrovenokDemo.

Самостоятельное задание

Нарисуйте дерево Explorer и объясните, почему require получает модуль по пути.

Важно

Ошибки написания имени папки или модуля приводят к невозможности найти объект.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 03 / 10 — Искры и конфетти

Теория и смысл

Effects.SparkBurst(part) подходит для короткого сигнала при сборе награды. Effects.Confetti(part) — для победы или открытия подарка. Оба эффекта должны появляться около конкретной детали, а не в случайной точке мира.

Практика

Запустите демоплощадку, подойдите к плиткам и нажмите E. Найдите в Effects.lua функции SparkBurst и Confetti. Измените один параметр оформления и сравните результат.

Проверка

Эффект появляется на выбранной детали, а после завершения не оставляет бесконечное количество объектов.

Самостоятельное задание

Создайте мини-награду: при касании блока появляется конфетти и проигрывается подходящий звук.

Важно

Сами визуальные функции не должны выдавать игровую валюту.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 04 / 10 — Вода и огонь

Теория и смысл

Effects.WaterSplash(part) имитирует всплеск, а Effects.FireBurst(part) — короткую огненную вспышку. Эти эффекты не добавляют физическую воду, температуру или урон. Их задача — визуально обозначить игровое событие.

Практика

Сделайте две площадки: синюю «вода» и красную «огонь». Запустите соответствующие эффекты. Сравните длительность и плотность частиц, затем проверьте сцену на мобильном профиле.

Проверка

Убедитесь, что эффекты не заслоняют обзор и не ухудшают заметно отзывчивость игры.

Самостоятельное задание

Используйте всплеск при приземлении в водную зону, а огонь — как предупреждение перед опасностью.

Важно

Урон от огня нужно реализовывать отдельной серверной логикой.

Effects.lua

ФРАГМЕНТ ИСХОДНОГО КОДА LUAU

```lua
-- Игровёнок | Библиотека эффектов (Luau)
-- Поместите ModuleScript в ServerScriptService/IgrovenokResources/Modules
local Effects = {}
local Debris = game:GetService("Debris")
local TweenService = game:GetService("TweenService")

local function emitter(parent, color, amount, speed, lifetime, size, spread)
    local p = Instance.new("ParticleEmitter")
    p.Name = "IgrovenokParticles"
    p.Color = ColorSequence.new(color)
    p.Rate = 0
    p.Speed = NumberRange.new(speed[1], speed[2])
    p.Lifetime = NumberRange.new(lifetime[1], lifetime[2])
    p.Size = NumberSequence.new({NumberSequenceKeypoint.new(0, size), NumberSequenceK
    p.SpreadAngle = Vector2.new(spread, spread)
    p.LightEmission = 0.65
    p.Parent = parent
    p:Emit(amount)
    Debris:AddItem(p, lifetime[2] + 1)
    return p
end
```

Иллюстрация: исходный код / схема установки, не снимок игрового процесса.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 05 / 10 — Магическое сияние и портал

Теория и смысл

Effects.MagicAura(part) выделяет важный объект, а Effects.Portal(part) создаёт визуальный образ портала. В этой библиотеке портал не переносит персонажа. Это удобный повод обсудить разделение внешнего вида и игровой механики.

Практика

Создайте две детали, примените к ним MagicAura и Portal. Затем добавьте рядом отдельную рабочую механику телепорта из набора полезных скриптов. Сравните роли двух файлов.

Проверка

Персонаж переносится только при наличии отдельного скрипта телепорта; визуальный эффект сам по себе ничего не перемещает.

Самостоятельное задание

Сделайте портал с изменяющимся цветом, который сигнализирует о доступности перехода.

Важно

Не называйте визуальный модуль «готовым телепортом» в описании игры.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 06 / 10 — Вращение и левитация

Теория и смысл

Animations.Spin(part, seconds) вращает деталь, а Animations.Bob(part, height, seconds) создаёт плавное движение вверх-вниз. Это анимации объектов, не скелетные анимации персонажей.

Практика

Выберите два закреплённых объекта. Для первого включите Spin, для второго Bob. Сравните движение при разных периодах и высоте. Изменяйте параметры постепенно и записывайте наблюдения.

Проверка

Предмет должен двигаться предсказуемо, не исчезать из поля зрения и не создавать непроходимое препятствие.

Самостоятельное задание

Создайте коллекционный кристалл, который одновременно светится и плавно парит.

Важно

Движение деталей, с которыми взаимодействует физика, требует отдельной проверки коллизий.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 07 / 10 — Платформа и дверь

Теория и смысл

Animations.MovingPlatform(part, offset, seconds) перемещает объект между точками. Animations.DoorSwing(part, degrees, seconds) поворачивает деталь вокруг её центра. Это простой учебный пример, но настоящей двери с петлями нужна корректная точка вращения.

Практика

Запустите оба примера на демоплощадке. Отметьте начальную и конечную позицию платформы. Для двери сравните поворот вокруг центра с ожидаемым вращением вокруг петли.

Проверка

Проверьте, что платформа возвращается, а дверь не пересекает критические элементы уровня.

Самостоятельное задание

Соберите мини-комнату с дверью и движущимся мостом.

Важно

Для точной механики дверей используйте отдельную модель с Pivot или петлями.



Иллюстрация: исходный код / схема установки, не снимок игрового процесса.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 08 / 10 — Пульсация и сочетание эффектов

Теория и смысл

Animations.Pulse(part, factor, seconds) меняет размер детали. Если одновременно применять пульсацию, вращение и частицы, важно соблюдать читаемость сцены. Не каждое сочетание выглядит хорошо: слишком много движущихся элементов отвлекает игрока.

Практика

Создайте объект «награда». Добавьте к нему пульсацию и один эффект частиц. Сравните вариант с двумя эффектами и вариант с четырьмя. Попросите другого ученика выбрать более понятный.

Проверка

Главный объект должен оставаться различимым на фоне и не мешать взаимодействию.

Самостоятельное задание

Сделайте три варианта подсказки для редкого предмета: лёгкая, средняя, яркая.

Важно

Изменение размера влияет на геометрию; при касаниях и коллизиях нужны дополнительные тесты.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 09 / 10 — Повторное использование ModuleScript

Теория и смысл

Модули позволяют не копировать одинаковый код во множество объектов. Скрипт вызывает функцию по имени и передаёт ей деталь и параметры. Такой подход облегчает поддержку: изменение одной функции отражается во всех местах её применения.

Практика

Создайте Script в ServerScriptService и подключите оба модуля через require. Вызовите Effects.MagicAura(part) и Animations.Bob(part, 2, 1.5) для тестовой детали.

Проверка

Проверьте, что функция вызывается без ошибок и получает именно BasePart.

Самостоятельное задание

Создайте новую обёртку Celebrate(part), объединяющую сияние и конфетти, не копируя исходники эффектов.

Важно

Не выполняйте require в цикле без необходимости; заранее продумайте жизненный цикл создаваемых объектов.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.

ГЛАВА 10 / 10 — Собственная демонстрационная выставка

Теория и смысл

Итоговый проект — выставка из шести эффектов и пяти анимаций, где ученик может объяснить назначение каждого примера. Важны удобство просмотра, читаемые названия, управление и разумная производительность.

Практика

Постройте маршрут из пяти остановок. На каждой разместите одну или две демонстрации и табличку с названием функции. Проверьте с другим игроком, что он понимает, куда идти и что нажимать.

Проверка

В Output нет ошибок; эффекты запускаются, а игрок не застревает в геометрии.

Самостоятельное задание

Добавьте переключатель «экономичный режим», уменьшающий интенсивность декоративных частиц.

Важно

Перед публикацией протестируйте на слабом устройстве; функции являются учебными и не проходили нагрузочные испытания.

Файл проекта: igrovenok-effects-animations-v1.zip • Учебные исходники Luau; проверка в Roblox Studio требуется.