

2 ИГРА: СТРУКТУРА И ТЕХНОЛОГИИ

В этой главе мы изучим разработку базовых игровых механик: сбор предметов, нанесение урона, лечение, разрушение, анимация, обработка событий, диалоги, эффекты, хранение данных игрока, hud и создание магазина, а также основы гейм-дизайна: работа с окружением, подсказки, разработка локаций, импорт 3d-моделей.

ЧТО ТАКОЕ ИГРА

Чтобы начать создавать игры, нужно разобраться в самом понятии игры. Однозначного определения нет, но вот несколько вариантов¹:

- **Игра** — это один из видов деятельности, характерных для животных и человека, она имитирует обстановку предметной и социальной действительности.
- **Игра** — это соревнование или состязание между участниками (детьми или взрослыми) по заранее согласованным, строго определенным правилам (условиям), направленным на достижение определенных общепринятых целей.
- **Игра** — форма деятельности в условных ситуациях, направленная на воссоздание и усвоение общественного опыта, фиксированного в социально закрепленных способах осуществления предметных действий, в предметах науки и культуры.
- **Игра** — это занятия, действия, формы общения детей, не носящие обязательного характера, приносящие чувство радости, удовольствия от достижения игрового результата.

Определений много, и все они разные, но из них можно взять ключевые фразы:

- обучение, социализация;
- досуг, развлечение, эмоции;
- правила, получение результата;
- получение опыта и навыков.

Определившись с ключевыми фразами, перейдем к определению компьютерной игры.

Компьютерная игра — это компьютерная система, способствующая такого рода досугам или социальным взаимодействиям, каковыми характеризуется понятие «игра» и явление спортивных игр в частности².

Как видно, понятие «компьютерная игра» опирается на фундаментальное понятие игры, но ограниченной рамками вычислительной техники (компьютеры, приставки, планшеты, телефоны и т. д.).

Таким образом, хорошая компьютерная игра должна:

- вызывать эмоции (желательно положительные);
- содержать органичную игровую механику, определяемую правилами игры;
- давать игровой опыт для роста персонажа в его мире;
- содержать небольшие и крупные цели, которые хочется достигать для получения игрового опыта и достижения результата.

¹ Источник: <https://vocabulary.ru/termin/igra.html>

² Определение взято из <https://cyclowiki.org>

Если стараться следовать этим правилам, то твоя игра, скорее всего, будет интересна многим пользователям. Не буду углубляться в теорию разработки игр, так как об этом написано немало книг, а сразу перейду к ее практической реализации. На примерах мы рассмотрим правдивость теории, описанной выше.

КАМЕРА, СВЕТ, АТМОСФЕРА

При выборе жанра игры нужно определиться с концепцией проекта: о чем будет игра, какие у нее правила, персонажи и механики. От этого зависит выбор работы камер, а также настройка света и окружения для создания атмосферы погружения в историю.

КАМЕРА

Камеру, свет и атмосферу можно настраивать как вручную, так и с помощью скриптов.

Начнем с настройки камеры. Когда игрок попадает в игру, то по умолчанию он уже имеет камеру, благодаря которой и видит окружение. Стандартные настройки камеры находятся в папке StarterPlayer в окне Explorer. Щелкнув по ней, откроем ее свойства, которые сможем настроить в окне Properties (рис. 2.1). Эти свойства определяют настройки персонажа в игре: место загрузки персонажа, характеристики прыжка, скорости перемещения, способы управления и настройка камеры.

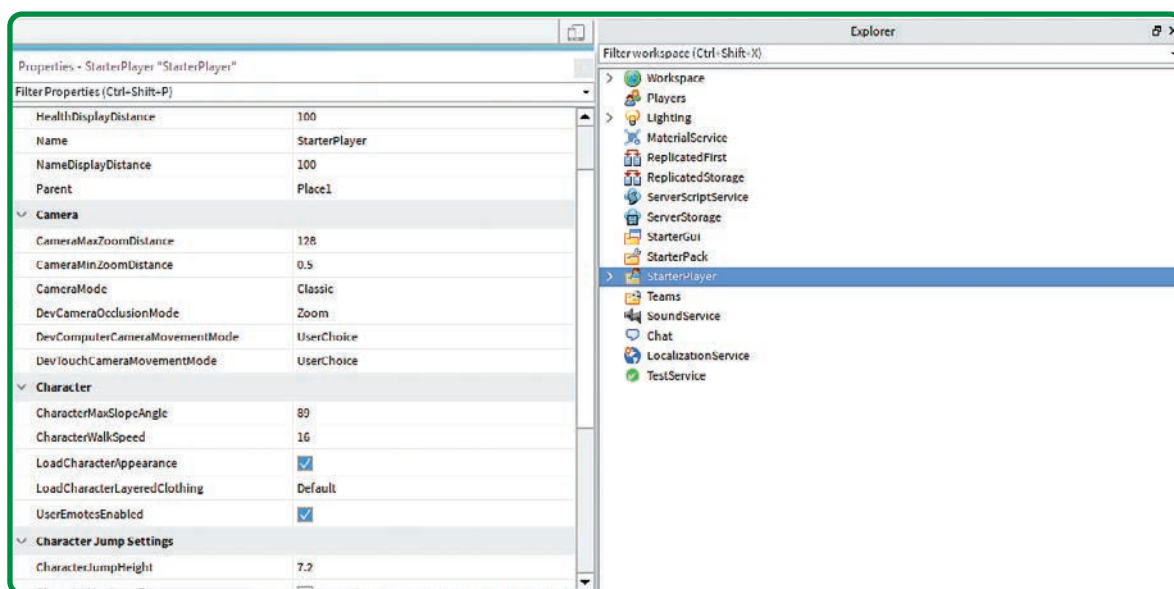


РИС. 2.1. НАСТРОЙКА ПАРАМЕТРОВ ИГРОКА

Рассмотрим подробнее параметры свойства Camera.

Для настройки максимального приближения и удаления камеры используют два параметра: CameraMaxZoomDistance и CameraMinZoomDistance.

Параметр CameraMode имеет два варианта: Classic и LockFirstPerson. В первом разрешается изменение масштаба камеры и вращение вокруг игрока, а во втором вид только от первого лица. Дополнительно можно настроить ограничения камеры для игр на компьютере и мобильных устройствах: DevComputerCameraMovementMode и DevTouchCameraMovementMode. Можно настроить и поведение камеры, если между игроком и камерой возникает препятствие: камера приближается к игроку, камера видит сквозь препятствие. За это отвечает параметр DevCameraOcclusionMode. Подробно об этих настройках можно почитать здесь: <https://developer.roblox.com/en-us/articles/customizing-the-camera>.

Допустим, нужно, чтобы вид для всех игроков был от третьего лица, без возможности изменять, приближать и удалять камеру. Для этого укажем одинаковые значения для минимальной и максимальной дистанции камеры (рис. 2.2).

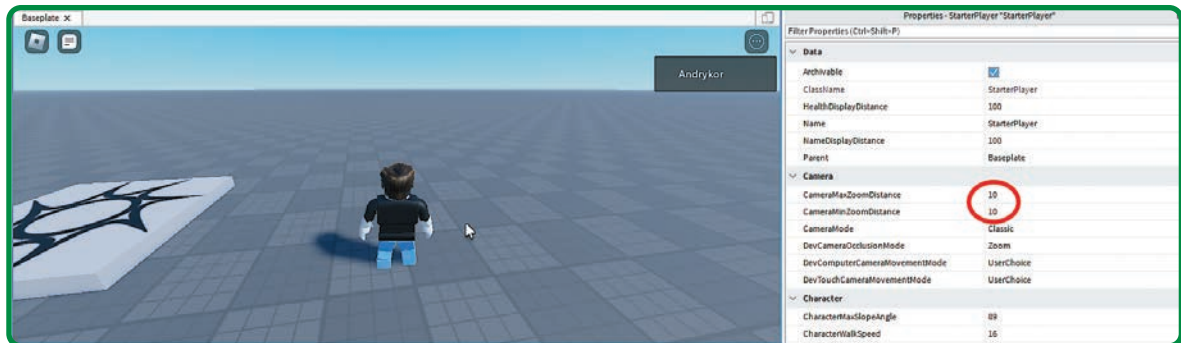


РИС. 2.2. ВИД ОТ ТРЕТЬЕГО ЛИЦА

Более специфичные варианты настройки камеры требуют написания скриптов. Код для камеры должен быть записан в файле localScript, который размещают либо в папку StarterPlayerScripts, либо в StarterPack.

Рассмотрим положение камеры, которая отслеживает положение игрока, сверху. Вернем начальные характеристики камеры в исходное положение. Создадим localScript и разместим в папку StarterPlayerScripts. Ниже представлен код, который будет содержать этот файл.

```
wait(2)
-- сервис для управления временем
local RunService = game:GetService("RunService")
-- подключаемся к камере
local Camera = workspace.CurrentCamera
-- определяем локального игрока
local player = game:GetService("Players").LocalPlayer
local character = player.Character or player.CharacterAdded:Wait()

-- отслеживание положения с ориентацией
```

```

Camera.CameraType = Enum.CameraType.Scriptable

--RunService.Heartbeat -- запуск кадра после симуляции физики
--RunService.Stepped -- запуск кадра перед симуляцией физики

RunService.Stepped :Connect(function()
    -- выстраиваем положение модели относительно игрока на каждом кадре
    Camera.CFrame = CFrame.new(character:WaitForChild("HumanoidRootPart").
Position) +
                        Vector3.new(0,25,0)
    Camera.CFrame *= CFrame.Angles(math.rad(-90),0,0)
end)

```

Если запустить проект, то мы увидим, как после загрузки камера перейдет в режим просмотра сверху и будет следовать за персонажем (рис. 2.3).

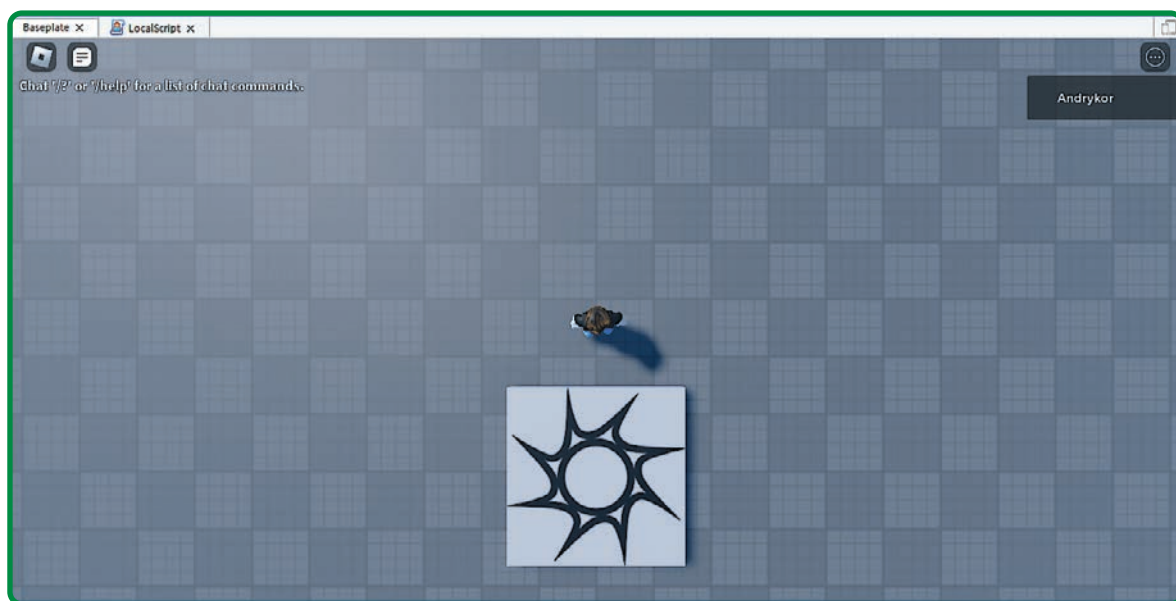


РИС. 2.3. КАМЕРА: ВИД СВЕРХУ

СВЕТ И ЦВЕТ

В разработке игр свет и сочетание цветов играют одну из важных ролей. С помощью цвета можно передать настроение в игре и вызвать у игрока соответствующие эмоции. Цвет влияет на реакцию, концентрацию внимания, скорость утомляемости. С помощью цвета можно также противопоставить игровых персонажей. Если мир большой или в нем используется много игровых объектов, то для помощи игроку в ориентации и порядка действий используют определенные контрастные цвета и подсветку.

Приведу пример одной игровой сцены, созданной в Roblox Studio с разным выбором цветовой палитры и освещения (рис. 2.4–2.6).

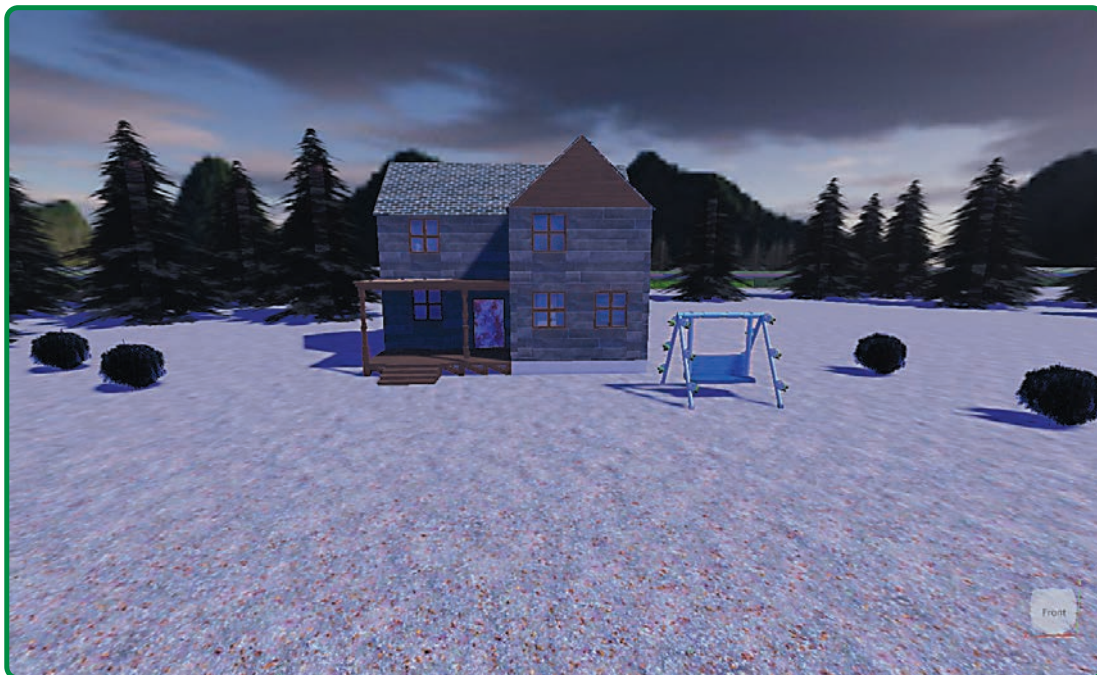


РИС. 2.4. ТЕМНЫЕ ТОНА С ПРЕОБЛАДАНИЕМ СИНЕГО

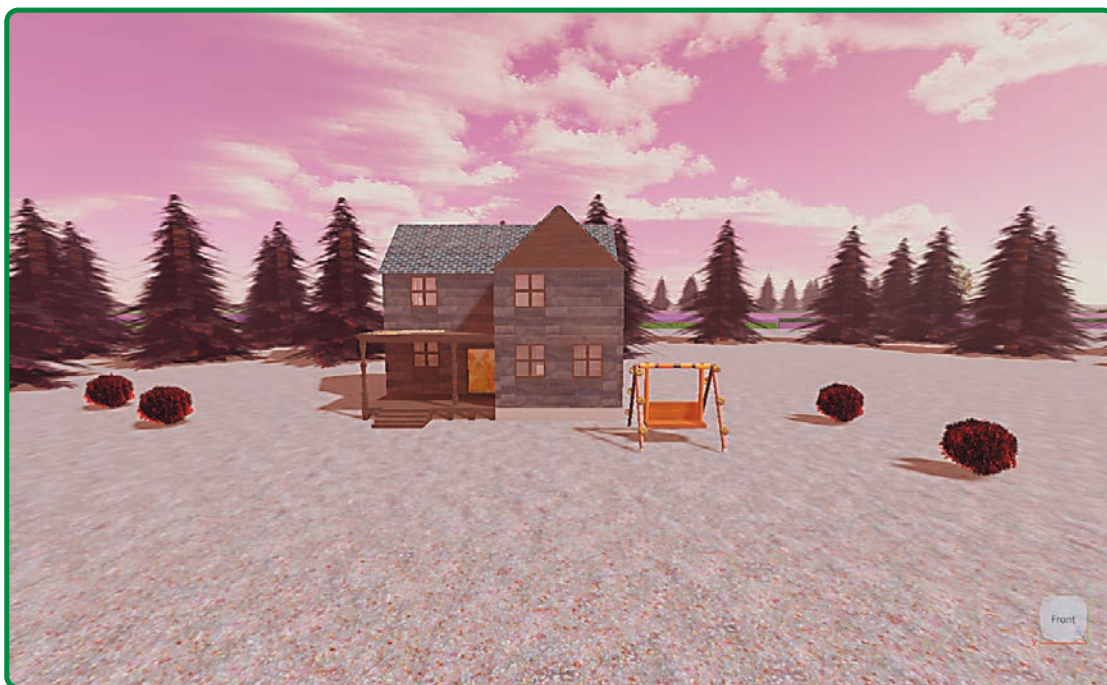


РИС. 2.5. ЯРКИЕ ТОНА: КРАСНЫЙ И ОРАНЖЕВЫЙ ЦВЕТА



РИС. 2.6. ЯРКИЕ ТОНА С ПРЕОБЛАДАНИЕМ ЗЕЛЕННОГО И ГОЛУБОГО ЦВЕТА

Рассматривая такие локации, можно испытать определенные эмоции и сделать предположения о жанре этих игр: хоррор, фэнтези, творческий жанр, RPG (role-play game, ролевая игра). На скриншотах видны игровые объекты, которые выделяются на общем фоне, тем самым вызывая интерес игрока и желание подойти к ним.

Мы рассмотрели дневное время, где можно изменять цвет освещения и текстуру неба. Аналогично можно работать и с ночным временем суток: сделать его сказочным, добавив элементы люминесценции, или мрачным и зловещим.

Цвет может использоваться для разграничения персонажей по их назначению или по отношению к игроку и другим персонажам. На рисунке ниже представлены пять роботов разного цвета (рис. 2.7). Посмотри на них и определи для себя, какие из них вызывают симпатию, а какие настороженность. Какие роботы подошли бы на роли добрых персонажей, а какие — на злых?

Роботы одинаковы по своей конструкции, но разные по цвету. Скорее всего, большинство читателей выберут трех роботов в качестве положительных персонажей (зеленый, синий и желтый), а оставшихся двух (красный и серый) — в качестве отрицательных. Наш мозг сразу разделил роботов, хотя мы и не знаем, что они на самом деле делают в игре. Можно использовать это в сюжете: либо подыграть мозгу, либо сыграть на контрасте (серый и красный — добрые и полезные, а остальные вредные и опасные).

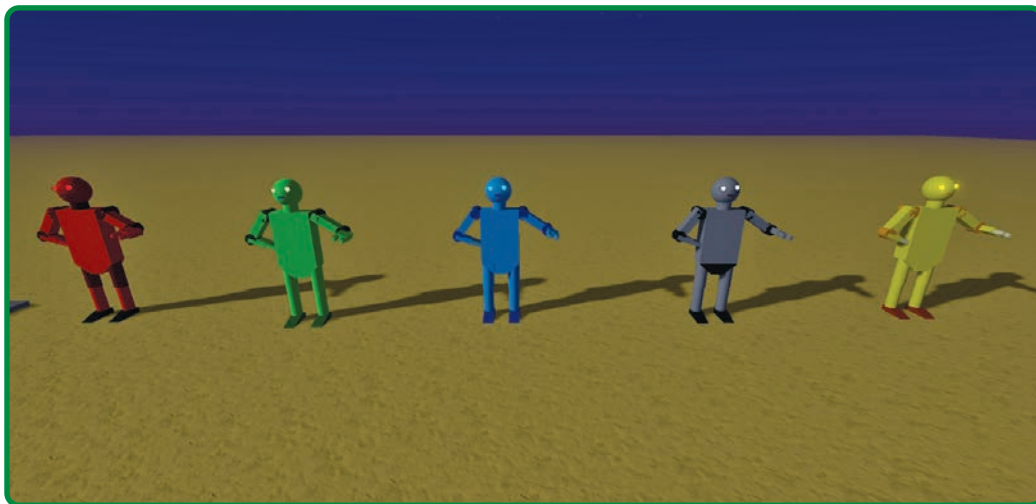


РИС. 2.7. РАЗНОЦВЕТНЫЕ РОБОТЫ

Если твоя игра командная, где одна группа игроков играет против другой, то для идентификации «свой-чужой» также можно использовать цвет (рис. 2.8).

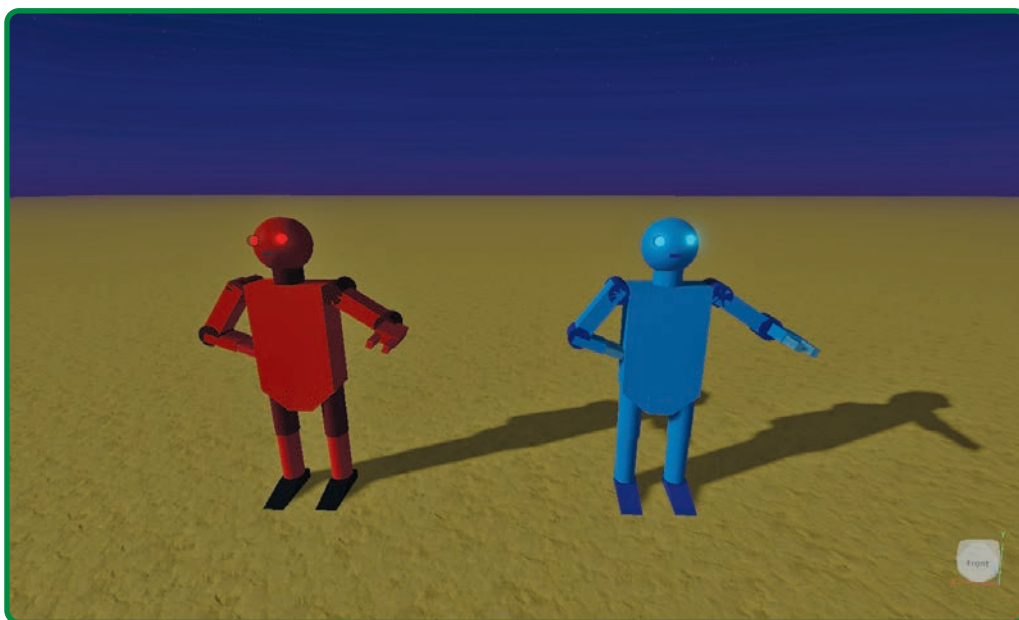


РИС. 2.8. РАЗБИВАЕМСЯ НА КОМАНДЫ

Экспериментируй в своей игре с цветами. Есть очень много информации в интернете и масса книг по использованию цвета в дизайне продукта, разработке игр и т. д. Поэтому так важно вначале придумать сюжет игры, персонажей и механику. Понимая, как у тебя будет всё это работать, ты делаешь акценты

цветовым решением, освещением, звуком, всплывающими подсказками и т. д., чтобы помочь игроку погрузиться в игру, заинтересовать его и показать, какие любовь и труд были в нее вложены.

АТМОСФЕРА

Под атмосферой в игре подразумеваются сразу два понятия:

- формирование игрового окружения для создания эмоциональной картины (веселое фэнтези, выживание, творческий режим, хоррор, масштабность событий);
- состав воздуха, поведение лучей света в воздухе (вблизи и на горизонте), атмосферные явления.

Первое понятие я рассмотрел выше, и, думаю, эта информация уже дала представление об атмосфере игры. Поэтому сейчас рассмотрим воздух и атмосферные явления.

Например, туман часто используют, чтобы оградить пространство или скрыть его либо временно, либо навсегда. Это делается ради экономии ресурсов компьютера и экономии времени на разработку фоновых изображений или объектов. Туман используют и для визуализации реалистичных атмосферных явлений: в горах, над рекой или над болотом.

Воздух тоже может быть разным. В лесу или высоко в горах он может быть чистым, а в горах еще и разреженным. В городе, селах или в пустынях в воздухе может быть пыль, выхлопы, летающие песчинки, клубы дыма и т. д. В зависимости от качества воздуха по-разному будет себя вести и освещение: где-то оно яркое, а где-то тусклое. В каком-то случае свет будет однотонный, а в каком-то распадаться на спектр или изменять цвет в зависимости от положения источника света и распространения лучей в игровом мире.

Настройки атмосферы воздуха находятся во вкладке **Lighting** в окне **Explorer**. По умолчанию при создании нового проекта в этой вкладке есть уже элемент **Atmosphere**: это продвинутый вариант настройки.

Пройдемся по свойствам этого элемента:

- **Density** — количество частиц в воздухе. От их концентрации зависит видимость и прохождение света.
- **Offset** — уровень детализации. Этот параметр нужно правильно сочетать с предыдущим.
- **Haze** — настройка тумана. В сочетании с параметром **Color** можно настроить его цвет.
- **Glare** — блики от солнца.
- **Decay** — настройка цвета с градацией от выбранного цвета к цвету основного освещения во вкладке **Lighting** и цвету тумана.

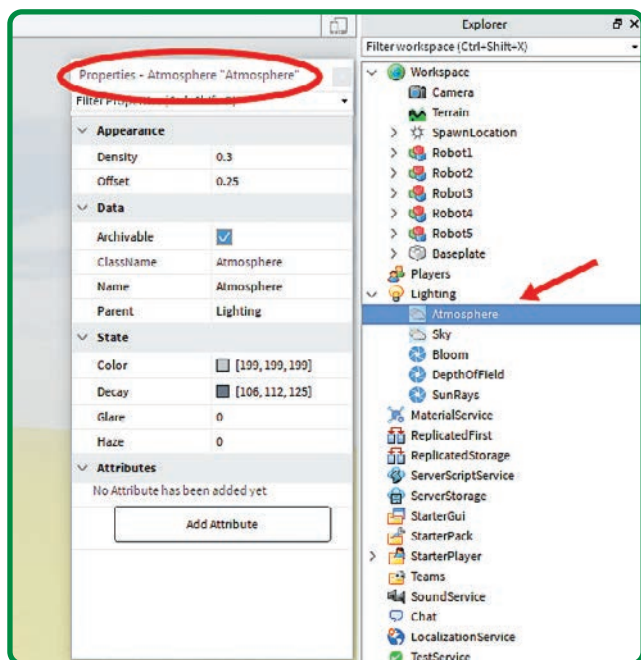


РИС. 2.9. НАСТРОЙКА АТМОСФЕРЫ В ИГРЕ

В старых версиях Roblox некоторые настройки были во вкладке Lighting. Их можно включить и сейчас (настройка тумана, частиц), если удалить элемент Atmosphere (рис. 2.10).

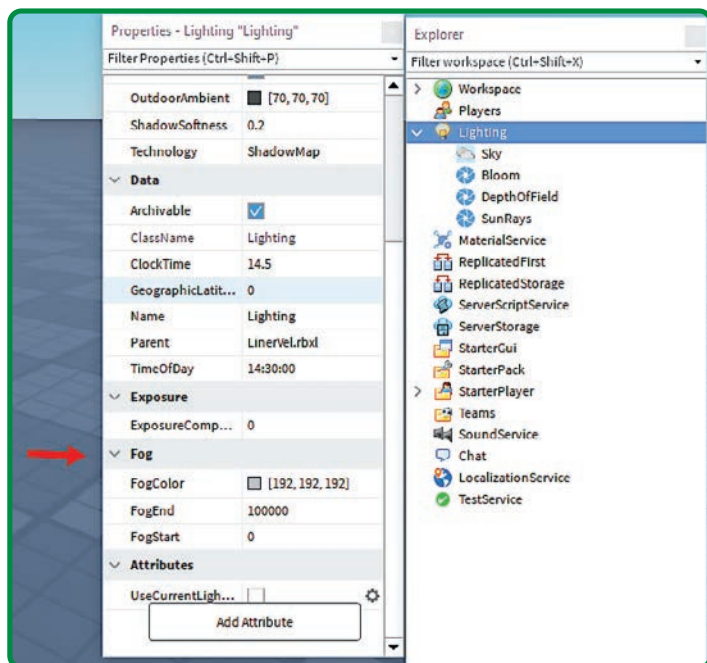


РИС. 2.10. НАСТРОЙКА ТУМАНА И ЧАСТИЦ АЛЬТЕРНАТИВНЫМ СПОСОБОМ