ЛЕГКОЕ ПРОГРАММИРОВАНИЕ

SCRATCH ДЛЯ ДЕТЕЙ

САМОУЧИТЕЛЬ ПО ПРОГРАММИРОВАНИЮ

МАЖЕД МАРЖИ

МИФ ДЕТСТВО

no starch press

ОГЛАВЛЕНИЕ

Установка прозрачных цветов

Когда два изображения накладываются друг на друга, то, которое оказывается сверху, перекрывает часть нижнего. Точно так же спрайты закрывают часть **Сцены**. Если вы хотите видеть, как выглядит **Сцена** за каким-то изображением, нужно воспользоваться графическим редактором, чтобы сделать хотя бы часть изображения *прозрачным*, как пингвин справа на рис. 1.17.

Кликните на цветовой палитре по квадратику с красной диагональной линией и рисуйте «прозрачным» цветом, чтобы сделать что-то невидимым. Представьте себе, что это знак «Цвета нет», что-то вроде знака «Не курить» с красной линией, перечеркивающей сигарету.

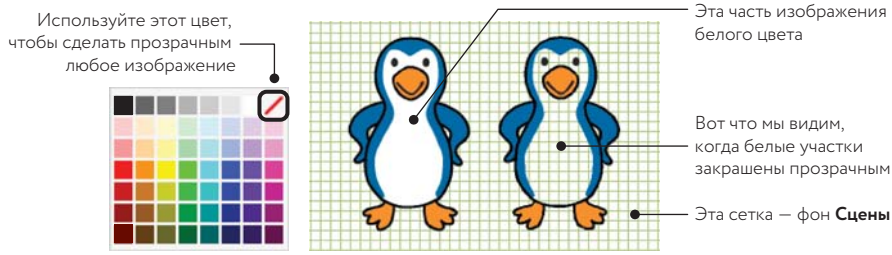

Рис. 1.17. Вы можете сделать прозрачной любую часть изображения, залив ее «прозрачным» цветом

Теперь, когда вы разобрались с интерфейсом Scratch, мы с толком используем эти знания и сделаем кое-что любопытное. Закатайте рукава и приготовьтесь: мы создаем игру!

Ваша первая игра в Scratch

Pong.sb2
Pong_NoCode.sb2

В этом разделе вы создадите свою компьютерную игру, в которой пользователю нужно будет передвигать ракетку, чтобы не дать прыгающему мячику удариться об пол. Она основана на классической аркадной игре Pong. Пользовательский интерфейс показан на рис. 1.18.

Как показано на рисунке, мяч начинает движение в верхней части **Сцены** и перемещается вниз под случайным углом, отскакивая от краев **Сцены**. Игрок перемещает ракетку по горизонтали (при помощи мыши), чтобы отбить мяч обратно наверх. Если мяч коснется нижней границы **Сцены**, игра окончена. Мы будем создавать игру пошагово, и первым делом нужно открыть новый проект. Выберите **Файл — Новый**, чтобы начать новый проект Scratch. Затем удалите спрайт-кота, кликнув по нему правой кнопкой мыши и выбрав **Удалить** из выпадающего меню.

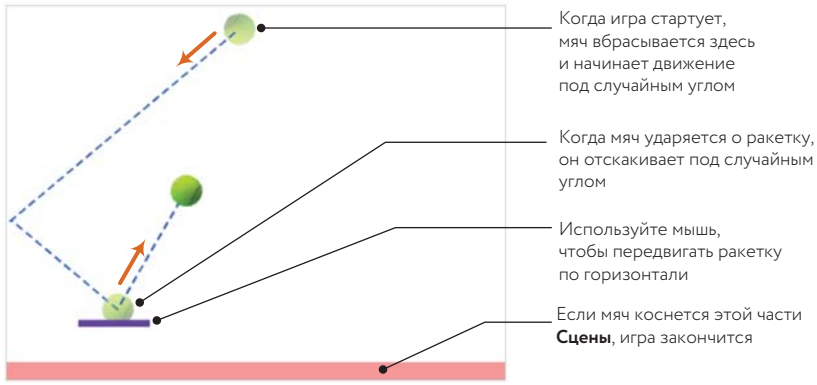

Рис. 1.18. Наша игра на экране

Шаг 1: подготовка фона

Чтобы определить, когда мяч попадает мимо ракетки, мы обозначим нижнюю границу **Сцены** определенным цветом и используем блок **дотронуться до цвета ?** (из раздела **Сенсоры**), чтобы определять, когда мяч будет касаться этого цвета. Сейчас у нас белый фон, и мы можем нарисовать у нижней границы **Сцены** тонкий цветной прямоугольник, как показано на рис. 1.19.

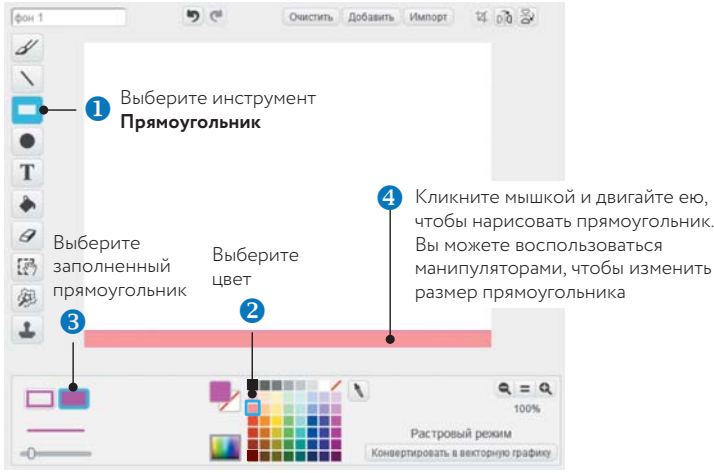

Рис. 1.19. Пошаговый процесс рисования прямоугольника внизу фона **Сцены**

Кликните по иконке **Сцены**, чтобы выбрать ее, а затем идите в закладку **Фон**. Повторите шаги, показанные на рис. 1.19, чтобы нарисовать тонкий прямоугольник внизу фона **Сцены**.

Шаг 2: добавляем ракетку и мяч

Нажмите кнопку **Нарисовать новый спрайт** наверху списка спрайтов, чтобы добавить в свой проект спрайт Ракетка (Paddle). Поскольку ракетка — просто узкий короткий прямоугольник, повторите шаг 1, чтобы нарисовать нужную фигуру, как на рис. 1.18. Раскрасьте ее в любой цвет, какой вам нравится, и установите центр приблизительно посередине прямоугольника. Затем дайте спрайту имя, которое объясняло бы, что он собой представляет. Я назвал его Ракеткой. А также кликните по изображению ракетки на **Сцене** и передвиньте его так, чтобы координата *y* была около 120.

Теперь у нас есть ракетка, но пока не хватает мяча, который скакал бы вокруг. Нажмите **Выбрать спрайт из библиотеки** наверху в списке спрайтов, чтобы импортировать спрайт.

В появившемся окне выберите категорию **Предметы**, затем изображение теннисного мяча и добавьте его в ваш проект. Переименуйте спрайт в Мяч.

Прежде чем вы начнете работать над скриптами для игры, выберите **Файл — Скачать на свой компьютер**, чтобы сохранить то, что вы уже сделали. В появившемся диалоговом окне выберите папку, куда вы хотите сохранить свою работу, назовите файл *Pong.sb2* и нажмите **Сохранить**. Если вы вошли на сайт Scratch под своим логином, вы можете сохранить свою работу в *облаке* (на сервере Scratch). В любом случае не забывайте сохранять файлы достаточно часто.

Теперь, когда у нас есть спрайты Ракетка и Мяч, **Сцена** должна выглядеть приблизительно как на рис. 1.18. Если на этом этапе у вас возникнут трудности, вы можете открыть файл *Pong_NoCode.sb2*, в котором есть всё, что мы только что сделали. Следующим шагом будет добавление скриптов, необходимых, чтобы игра заработала. Не переживайте из-за отдельных непонятных блоков, мы их обсудим позже. А пока сосредоточимся на том, чтобы научиться складывать из отдельных частей готовый проект.

Шаг 3: начать игру и заставить спрайты двигаться

Вам решать, как игроки могут начать новый раунд. Например, нажав кнопку, кликнув по спрайту на **Сцене** или даже хлопнув в ладоши или помахав рукой (если у вас есть веб-камера).

Зеленый флажок (расположенный над **Сценой**) — еще одна популярная функция, ею-то мы и воспользуемся.

Идея проста. Любой скрипт, начинающийся с блока-триггера **когда щелкнут по**  , начинает работать, как только вы нажмете эту кнопку. Флажок становится ярко-зеленым и остается таким до тех пор, пока скрипт не завершит работу. Чтобы увидеть это в действии, создайте скрипт для спрайта Ракетка, как показано на рис. 1.20.

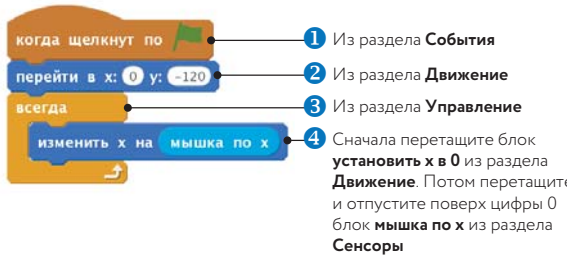

Рис. 1.20. Скрипт для спрайта Ракетка

После того как игрок кликает по зеленому флажку ❶, блок **перейти в x: y:** ❷ устанавливает вертикальную позицию ракетки на –120, просто на тот случай, если вы до этого мышкой убрали ее оттуда. Ракетка должна парить прямо над розовым прямоугольником внизу **Сцены**, так что если ваш прямоугольник толще, измените координаты ракетки на те, которые вам подходят лучше.

Затем в скрипте использован блок **всегда** ❸, чтобы постоянно отслеживать позицию мышки. Мы будем двигать ракетку туда-сюда за счет того, что укажем в качестве ее позиции по оси *x* позицию мыши ❹. Запустите скрипт (кликнув по зеленому флажку) и попробуйте двигать мышь горизонтально. Ракетка должна следовать за ней.

Кликните по значку **Стоп** рядом с зеленым флажком, чтобы остановить скрипт.

Скрипт для спрайта-мяча немного длиннее, чем предыдущий, так что я разобью его на простые части. Мяч должен начинать двигаться, как только мы кликаем по зеленому флажку, поэтому первым делом добавьте спрайту-мячу скрипт с рис. 1.21.

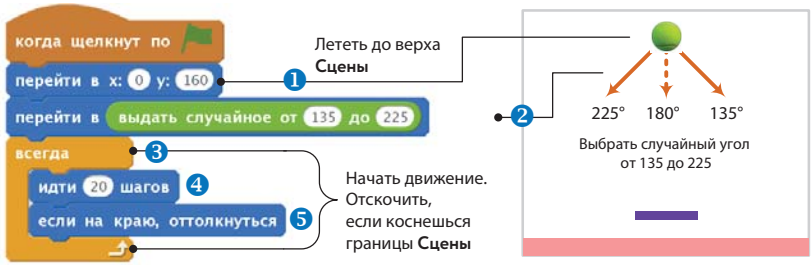

Рис. 1.21. Первая часть скрипта для мяча

Сначала мы перемещаем мяч наверх **Сцены** ❶ и заставляем его лететь вниз под случайным углом, используя блок **выбрать случайный угол** ❷ (из раздела **Операторы**). Затем скрипт использует блок **всегда** ❸, чтобы мяч перемещался по **Сцене** и отпрыгивал ❹ от ее краев. Используйте

зеленый флажок, чтобы проверить, что у вас получилось. Мяч должен передвигаться зигзагами, а ракетка — следовать за мышкой.

УПРАЖНЕНИЕ 1.11

Замените цифру 12 внутри блока **идти 12 шагов** на другие значения, запустите скрипт и посмотрите, что получится. Это покажет вам, как сделать игру для пользователя сложнее или легче. Когда закончите — нажмите кнопку **Стоп**.

Теперь пора добавить самое веселое — блоки, которые заставят мяч отскакивать от ракетки. Мы можем изменить блок **всегда**, который только что создали, так, что мяч будет лететь вверх, отскочив от ракетки, как показано на рис. 1.22.

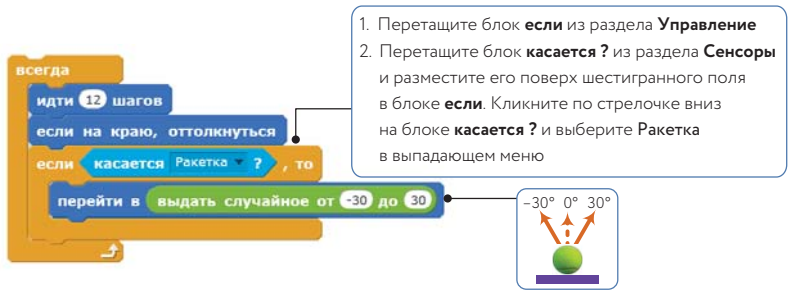

Рис. 1.22. Добавление блока, чтобы мяч отскакивал вверх

Когда мяч и ракетка соприкасаются, мы даем мячу команду повернуть в случайном направлении от –30 до 30. Когда блок **всегда** зайдет на второй круг, он начнет выполнять блок **идти**, который теперь будет заставлять мяч лететь вверх.

Кликните снова по зеленому флажку, чтобы протестировать эту часть игры. Когда вы удостоверитесь, что мяч отскакивает от ракетки, как и предполагалось, нажмите **Стоп**.

Теперь единственный недостающий фрагмент — код, который останавливает игру, если мяч касается нижней границы **Сцены**. Добавьте спрайту-мячу скрипт, как показано на рис. 1.23, либо непосредственно перед, либо сразу после блока **если** (рис. 1.22). Блок **касается цвета ?** вы найдете в разделе **Сенсоры**, блок **стоп** — в разделе **Управление**.

Когда вы кликнете мышью по цветному квадрату внутри блока **касается цвета ?**, стрелка курсора изменится на руку. Когда вы переместите курсор и кликнете по розовому прямоугольнику внизу **Сцены**, этот квадрат внутри блока станет такого же цвета. Блок **стоп все** делает именно

то, о чем говорит его название: останавливает все работающие скрипты. Ни Мяч, ни Ракетка не будут исключением.

Рис. 1.23. Блоки для окончания игры

Теперь наша простейшая игра Pong полностью готова к работе. Кликните по зеленому флажку и сыграйте несколько раз, чтобы протестировать ее. Надеюсь, после того как вы увидели, что можете самостоятельно создать целую игру, сделав так мало, вы согласитесь, что Scratch — что-то невероятное!

Шаг 4: приправим блюдо звуком

Играть, конечно, гораздо веселее, когда есть звук. Добавим один завершающий штрих: звук, который будет раздаваться каждый раз, когда мы отбиваем мяч. Дважды кликните по мячу на **Сцене**, чтобы выбрать его, а затем зайдите в закладку **Звуки**. Кликните по кнопке **Выбрать звук из библиотеки**, чтобы добавить звук спрайту-мячу. В появившемся диалоговом окне выберите категорию **Эффекты**, затем **pop** и нажмите **OK**, чтобы добавить звук в закладку **Звуки**. Потом вернитесь в закладку **Скрипты** и вставьте блок **играть звук** (из раздела **Звуки**), как показано на рис. 1.24.

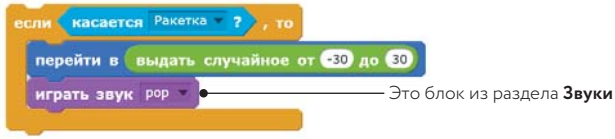

Рис. 1.24. Проигрывать звук, когда мяч касается ракетки

Запустите игру еще раз. На этот раз вы будете слышать короткий звук каждый раз, когда мяч касается ракетки.

Поздравляю! Ваша игра готова (если, конечно, вы не хотите добавить еще каких-нибудь функций). Вы только что написали свою первую программу в Scratch.

Если вы хотите еще поэкспериментировать, попробуйте продублировать спрайт-мяч, чтобы у вас было два (или более) мяча в игре, и посмотрите, что из этого получится!

В следующем разделе я познакомлю вас с разными типами блоков в Scratch. Вы узнаете многое о том, как работают эти блоки, а пока мы вкратце пройдемся по ним.

Блоки Scratch: обзор

Из этого раздела вы узнаете о разных блоках, с которыми работает Scratch, выучите их названия и области применения. Наша цель — разобраться в терминологии, которая встретится вам в следующих главах. Вы можете возвращаться к этому разделу позже, чтобы освежить что-то в памяти.

Как показано на рис. 1.25, Scratch использует четыре типа блоков: командные, функции, триггеры и контрольные. У *командных* и *контрольных блоков* (их еще называют *стеками*) снизу есть небольшой выступ и/или небольшая выемка наверху. Такие блоки можно соединять между собой в стопки. У *блоков-триггеров,* которые также называют *шляпами*, закругленный верх, потому что они помещаются наверх стопки блоков. Триггеры соединяют события в скрипты. Они ждут события — например нажатия кнопки или клика мышкой — и запускают расположенные под ними блоки, когда оно происходит. Например, все скрипты, начинающиеся с блока **когда щелкнут по**  , запускаются, если пользователь кликает по зеленому флажку.

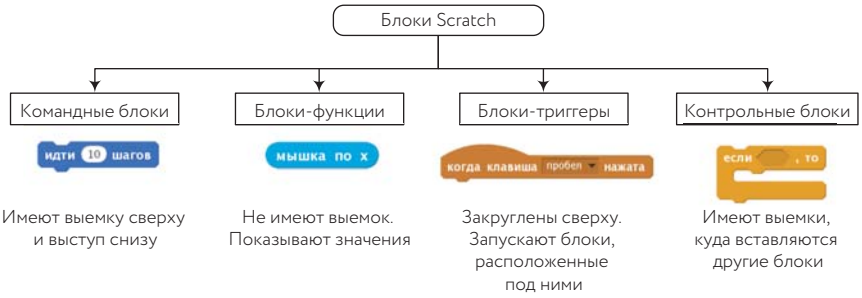

Рис. 1.25. Четыре типа блоков, используемых в Scratch

У *блоков-функций* (их мы называем *репортерами*) нет выемок или выступов. Они не могут самостоятельно сформировать слой скрипта, а используются как вставки в другие блоки. Форма этих блоков обозначает тип данных, которые в них содержатся. Так, например, блоки с закругленными краями содержат цифры или дополнительные условия, а блоки с заостренными краями показывают, истинно что-то или ложно. Это проиллюстрировано на рис. 1.26.

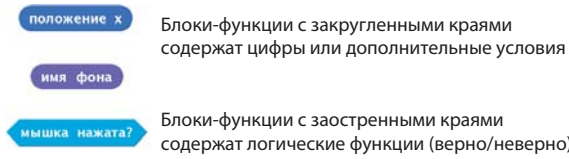

Рис. 1.26. Форма блока-функции говорит о типе данных, которые он содержит

Рядом с некоторыми блоками-функциями находятся чекбоксы. Если вы поставите там галочку, на **Сцене** появится *монитор*, который будет показывать актуальное значение репортера. Выберите спрайт и поставьте галочку в чекбоксе блока **положение x** (из раздела **Движение**). Затем передвигайте спрайт по всей **Сцене** и следите за этим монитором. Его показания должны меняться, когда вы двигаете спрайт туда-сюда.

Арифметические операторы и функции

Посмотрим на арифметические операторы и функции, поддерживаемые средой Scratch. Теперь, если вы потеряете свой калькулятор, не стоит волноваться! Вы сможете смастерить себе калькулятор в Scratch при помощи блоков из раздела **Операторы**, с которыми вы познакомитесь в этом разделе.

Арифметические операторы

Scratch поддерживает четыре основных арифметических действия: сложение (+), вычитание (–), умножение (*) и деление (/). Блоки, которые используются для осуществления этих операций, называются *операторами*. Они показаны на рис. 1.27. Поскольку они производят числа, их можно использовать как вставки в любой блок, принимающий цифры.

Рис. 1.27. Арифметические операторы в Scratch

Scratch также поддерживает оператор модуля (**модуль**), который показывает остаток при делении одного числа на другое. Например, **10 модуль 3** дает 1, потому что остаток при делении 10 на 3 — это 1.

издательство
МАНН, ИВАНОВ И ФЕРБЕР
Максимально полезные книги