

Винницкий Ю. А.
Григорьев А. Т.

SCRATCH И ARDUINO

ДЛЯ ЮНЫХ ПРОГРАММИСТОВ
И КОНСТРУКТОРОВ

Санкт-Петербург
«БХВ-Петербург»
2018

УДК 004.43-053.2
ББК 32.973.26-018.1
В48

Винницкий, Ю. А.

В48 Scratch и Arduino для юных программистов и конструкторов / Ю. А. Винницкий, А. Т. Григорьев. — СПб.: БХВ-Петербург, 2018. — 176 с.: ил.

ISBN 978-5-9775-3937-1

Книга написана на основе опыта работы с детьми 3–7 классов и посвящена созданию творческих проектов для юных программистов и конструкторов. На примерах разработки простых компьютерных игр продемонстрированы базовые приемы визуального программирования в среде Scratch. Рассмотрено применение плат Arduino, плат расширения и различных внешних датчиков в детских конструкторских проектах. Даны основы визуального программирования этих устройств в среде mBlock и креативного программирования путем написания интерактивных компьютерных игр, в которых управлением персонажами осуществляется посредством Arduino. Также дети освоят методы создания автономных «умных вещей», работающих под управлением Arduino без подключения к компьютеру.

Электронный архив на сайте издательства содержит дополнительные материалы и листинги всех программ.

Для детей младшего и среднего школьного возраста

УДК 004.43-053.2
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Людмилы Гауль</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Марины Дамбиевой</i>

«БХВ-Петербург», 191036, Санкт-Петербург, Гончарная ул., 20.

ISBN 978-5-9775-3937-1

© ООО «БХВ», 2018
© Оформление. ООО «БХВ-Петербург», 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
-----------------------	----------

ЮНЫМ КОНСТРУКТОРАМ-ИЗОБРЕТАТЕЛЯМ: ВРЕМЯ СОЗДАВАТЬ БУДУЩЕЕ!	7
---	----------

ГЛАВА 1. СПРАЙТЫ, СКРИПТЫ, ЧЕРЕПАХИ И ДРУГИЕ ЖИТЕЛИ ВИРТУАЛЬНОГО МИРА В SCRATCH	10
--	-----------



Знакомьтесь — это Scratch!	10
«Черепашья графика».....	15
Собственные блоки для очень сложных узоров	23
Проект «Спираль из квадратов».....	27
Справочная система Scratch и проекты «Шаг за шагом»	35
Игра «Пинг-понг»	37

ГЛАВА 2. ЗНАКОМИМСЯ С КОНТРОЛЛЕРАМИ ARDUINO И УЧИМСЯ ПРОГРАММИРОВАТЬ ИХ В СРЕДЕ MBLOCK	42
---	-----------



Arduino — это просто!.....	42
Scratch для Arduino? Знакомьтесь — mBlock!	46
Основы работы с Arduino в mBlock	48
Подключение внешних светодиодов к плате Arduino Uno.....	58
Расширенные возможности цифровых портов Arduino	68
Макетная плата — инструмент для творческого конструирования	73

Обычная кнопка — пример цифрового датчика	83
Аналоговые датчики	90
Возможности сложных цифровых датчиков	99

ГЛАВА 3. ПРИДУМЫВАЕМ, КОНСТРУИРУЕМ, ИГРАЕМ! ARDUINO + MBLOCK = КРЕАТИВНЫЕ ИГРЫ!.....108



Компьютерные игры — простор для творчества! ...	108
Проект «Голодная рыбка» версия 1: управление клавишей <Пробел>	109
Проект «Голодная рыбка» версия 2: управление с Arduino с помощью кнопки.....	127
Проект «Голодная рыбка» версия 3: оптимизация программы.....	128
Проект «Голодная рыбка» версия 4: модификация программы, создание новых вариантов игры	134

ГЛАВА 4. «УМНЫЙ ДОМИК». АВТОНОМНЫЙ ПРОЕКТ С ПЛАТОЙ ARDUINO.....140



Собираем электрическую схему «умного домика»	141
Проект «Умный домик» версия 1	146
Проект «Умный домик» версия 2: автономная работа	157
Как вернуть возможность управлять Arduino интерактивно с использованием среды mBlock?....	167

ЗАКЛЮЧЕНИЕ..... 169

ПОЛЕЗНЫЕ КНИГИ..... 170

ПРИЛОЖЕНИЕ. СОДЕРЖАНИЕ ЭЛЕКТРОННОГО АРХИВА..... 171

Фрагмент проекта «Собачка с мячом». Используем потенциометр. Программируем вращение баскетбольного мяча.	171
---	-----

ВВЕДЕНИЕ

Scratch — популярнейший визуальный язык программирования, который идеально подходит для обучения детей от 8 лет, позволяет создавать творческие проекты, в непринуждённой игровой форме осваивать алгоритмизацию и программирование, развивать техническое мышление и инженерные навыки.

Arduino — самый распространённый микроконтроллер для создания электронных самоделок. Маленькая печатная плата совершила настоящую революцию в мире технического творчества, придав новый импульс развитию робототехники и конструирования. Программирование и использование микроконтроллеров перестало быть уделом избранных. А с появлением графических средств разработки на основе Scratch обучить Arduino выполнению нужной работы стало ещё проще. Грядёт новая революция — программирование микроконтроллеров становится доступным не только взрослым, но и детям!

Хотите попробовать? В ваших руках книга, в которой Scratch и Arduino объединяются в общих творческих проектах. Книга написана опытными преподавателями, в её основе — многолетний опыт практических занятий с детьми. Материалы предложены в игровой форме и занимательных проектах, с которыми интересно разбираться и детям, и взрослым, так что совместное техническое творчество может стать занимательным и полезным семейным досугом. Всё, что для этого нужно — книга, домашний компьютер и небольшой набор распространённых деталей.

Шаг за шагом авторы проведут вас в мир программирования и конструирования:

- **в главе 1** предложены проекты по рисованию в Scratch, что позволит получить основные навыки работы в этой визуальной среде;
- **в главе 2** читатели познакомятся с платами Arduino и смогут реализовать первые проекты с подключаемыми электронными компонентами;

- **в главе 3** юные изобретатели смогут почувствовать себя в роли программистов-разработчиков компьютерных игр, получат навыки отладки и модификации компьютерных программ, а также соберут на основе Arduino оригинальные пульта для управления игровым процессом;
- **в главе 4**, заключительной части книги, можно познакомиться с технологиями «умного дома» и сборкой первого автономного устройства управления.

А дальше... дальше открывается огромный мир возможностей. Роботы, квадрокоптеры, 3D-принтеры, манипуляторы, «умные вещи» и игровые проекты – лишь небольшой перечень путей дальнейшего развития. Главное – увлечь, показать, научить сделать первый шаг. И мы очень надеемся, что эта книга станет хорошим проводником в этот замечательный мир творчества.

Не зря в ближайшие годы инженерное образование и техническое творчество подрастающего поколения будут являться особо важной сферой для дальнейшего технологического развития нашей страны.

Нужно растить инженеров, конструкторов, изобретателей, создавать своё будущее!

ЮНЫМ КОНСТРУКТОРАМ- ИЗОБРЕТАТЕЛЯМ: ВРЕМЯ СОЗДАВАТЬ БУДУЩЕЕ!

Привет, наши юные читатели! Возможно, вам захочется пропустить это введение, чтобы побыстрее начать экспериментировать, сочинять компьютерные игры и собирать электронные самоделки. Мы вполне разделяем это стремление, ведь именно для тех, кто любит создавать что-то новое, книга и написана. Но тем не менее, считаем важным рассказать, что побудило нас взяться за неё.

Мы живём в интересное время. Нас окружают технические чудеса, о которых писатели-фантасты совсем недавно могли лишь мечтать. В руках почти у любого человека можно увидеть смартфон. Задумайтесь: это же миниатюрный компьютер, несопоставимо более мощный, чем тот, что использовался для расчёта траектории полёта первых космических кораблей и занимал целый этаж. Никого уже не удивляют домашние роботы-пылесосы или системы автоматической парковки в автомобилях. Мы привыкаем к чудесам и порой даже не задумываемся над тем, как же создаются разные технические диковинки, какой путь проходит изобретение от возникновения идеи до воплощения её в привычный всем бытовой прибор.

Но, раз вы читаете эту книгу, вряд ли подобные вопросы вам безразличны. И это здорово! Перед вами открывается огромная сказочная страна! Здесь будет своё волшебство — техническое, будут и специальные магические слова — программные команды, будут виртуальные существа и забавные устройства, выполняющие наши приказы. И происходить всё будет не только в воображении и на экране компьютера. Добро пожаловать в мир конструирования и творчества, инженеры будущего!

Вам потребуются знания, упорство и смелость мысли. Да-да, смелость! Ведь для того чтобы отказаться от обыденного и сделать нечто по-настоящему новое, нужна именно она. Подумайте, как непросто было создателям самых привычных для нас с вами вещей. В 1975 году два друга — никому неизвестные студенты Билл Гейтс и Пол Аллен — прочитали опубликованную в журнале «Popular

Electronics» («Популярная электроника») статью о появившейся новинке — персональном компьютере Altair 8800, продающемся в виде набора для сборки. Воодушевившись, они попытались написать для него программу — интерпретатор языка Basic, после чего и основали компанию **Microsoft**, в штате которой в первый год работы было всего 3 человека. А двое других друзей — Стив Джобс и Стив Возняк — примерно в то же время решили наладить собственное производство компьютеров. И 1 апреля 1976 года официально была основана компания **Apple**, а на рынке появился собранный вручную Apple Computer I.

Может, вам кажется, что такие примеры перестали быть актуальными? Всё уже изобретено? Но посмотрите, сколько нового появилось только за последние годы! Взять хотя бы 3D-принтеры — устройства, создающие из пластиковой нити готовые предметы. Совсем недавно технология трёхмерной печати показалась бы фантастикой, теперь же 3D-принтером никого не удивишь. Его можно собрать даже в домашних условиях, и существуют специальные наборы, включающие все необходимые детали и инструкции. Всё, как в далёких 1970-х годах с Altair 8800. История повторяется!



Apple Computer I

Роботы, квадрокоптеры, 3D-принтеры и многие другие технические новинки стали доступны благодаря появлению **микроконтроллеров**, по сути — маленьких и дешёвых компьютеров, запрограммированных на порученную им работу. Современный размах использования микроконтроллеров впечатляет. Они способны управлять самыми разными электронными и механическими устройствами, повсеместно используются в средствах связи, современных автомобилях, производственных станках, в массовой домашней технике и даже в игрушках.

Благодаря микроконтроллерам окружающие нас вещи становятся всё «умнее», а благодаря творческим идеям «умных» вещей становится всё больше.

Сравнительно недавно в мире появилась замечательная программная среда **Scratch**, навсегда изменив представление о том, с какого возраста можно серьёзно заниматься программированием. А новые модификации программ

на основе Scratch позволили даже детям программировать микроконтроллеры, что раньше вообще невозможно было представить! И наша книга именно о том, как работать в этих доступных средах, как создавать новые электронные устройства на основе микроконтроллеров и программировать их на выполнение необходимой работы.

Для того чтобы процесс был интереснее, мы будем создавать компьютерные игры и играть в них, используя необычные электронные устройства, от которых всего один шаг к самым настоящим «умным» вещам, например пульту управления бытовыми приборами, контрольной системе для освещения.

Читая книгу, вы можете сразу обращаться к страницам, на которых приведены программы в готовом для реализации виде, а можете пройти вместе с нами все шаги разработки, выступить в роли программиста и разобраться с каждым элементом проекта. Выбор за вами. Но главное — дерзайте, придумывайте, пробуйте, ошибайтесь и пробуйте вновь, создавая новое будущее. Ведь вполне вероятно, что именно ваше изобретение завоюет мир через несколько лет! Удачи!

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

Жирным шрифтом выделены элементы интерфейса программ — меню, команды, кнопки.

Названия блоков программирования выделены так: **идти 10 шагов**.

Названия переменных и параметров выделены **узким жирным шрифтом**.

Названия клавиш клавиатуры заключены в угловые скобки, например <Пробел>.

ЭЛЕКТРОННЫЙ АРХИВ

В **электронном архиве**, который можно скачать по ссылке <ftp://ftp.bhv.ru/9785977539371.zip> или найти на странице книги на сайте издательства по адресу www.bhv.ru, сохранены файлы всех проектов, описанных в книге. Все проекты распределены по главам. Обратите внимание на то, что помимо финальных программ там расположены и промежуточные версии, по которым шаг за шагом можно проследить путь создания проекта. Таким образом, если что-то и пойдёт не так, как рассказывается в книге, можно заглянуть в архив и посмотреть, как должен был выглядеть результат на том или ином этапе разработки.

СПРАЙТЫ, СКРИПТЫ, ЧЕРЕПАХИ И ДРУГИЕ ЖИТЕЛИ ВИРТУАЛЬНОГО МИРА В SCRATCH

ЗНАКОМЬТЕСЬ – ЭТО SCRATCH!

Дорогие читатели! В следующих главах вы научитесь собирать разнообразные электронные устройства на базе контроллера Arduino и «оживлять» их с помощью интересных программ. Но прежде всего следует выбрать язык программирования, который мы будем использовать для этого.

Юным исследователям более всего подходит программирование в среде Scratch, где программы (называемые **сценариями**) собираются из блоков, как из кубиков конструктора. Предлагаем вам познакомиться с этой средой и её основными возможностями.

ОТКУДА ПОЯВИЛОСЬ НАЗВАНИЕ «SCRATCH»?

Одни авторы утверждают, что своим названием программа обязана главному действующему персонажу, коту-царапке (рис. 1.1), так как в английском языке есть глагол to scratch, означающий «царапать». Кстати, персонажи в Scratch называются **спрайтами**.



Рис. 1.1. Спрайт кота

Но есть и другие версии. Так, в американском варианте английского языка есть выражение *to start from scratch* – это значит «начать с самого начала (с основ)». И правда, Scratch и в нашем случае – начало, основа, с изучения которой вы войдёте в мир профессиональной разработки как будущие программисты, схемотехники, инженеры-конструкторы или технические дизайнеры. А по ещё одной версии название произошло от слова *scratching* – техники, используемой хип-хоп-диджеями, которые крутят виниловые пластинки взад-вперёд руками для того, чтобы смешивать музыкальные темы. Выбирайте, какой вариант вам больше нравится!

ГДЕ ВЗЯТЬ SCRATCH?

Тут всё просто. Эта среда является представителем **свободного программного обеспечения**, то есть любой желающий может её использовать и даже модифицировать.

Официальная страница проекта Scratch – <https://scratch.mit.edu/>.

Зайдя на данный ресурс, вы увидите много (всего представлено более 20 миллионов) уже готовых программ-сценариев, созданных в сообществе Scratch, в состав которого и вы можете войти, воспользовавшись ссылкой **Присоединяйся**. Регистрация даст возможность хранить ваши новые сценарии в облаке Scratch, общаться с единомышленниками – изобретателями со всего мира.

Для того чтобы попробовать программирование в среде Scratch, даже не обязательно загружать и устанавливать программу на компьютер, можно перейти на вкладку **Создавай** (рис. 1.2) и воспользоваться онлайн-версией программы. Этот вариант особенно удобен в случае, когда работать со Scratch приходится вдали от основного рабочего места, например в дороге, с планшета.

Для постоянной работы с программой удобнее стационарная версия, которую можно загрузить со страницы <https://scratch.mit.edu/scratch2download/>. Потребуется указать операционную систему своего

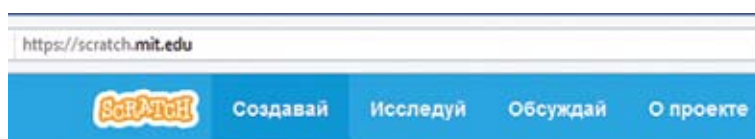



Рис. 1.2. «Создавай» – переход к онлайн-версии Scratch

компьютера (доступны варианты для Windows, Linux, Mac OS X), а затем последовательно загрузить и установить на компьютер две программы: **Adobe AIR** и **Scratch 2.0 Offline Editor**.

ЗАПУСК И НАЧАЛО РАБОТЫ

После установки запустите среду Scratch, дважды щёлкнув по её значку на рабочем столе компьютера либо воспользовавшись главным меню операционной системы.

Выберите язык, на котором вы будете общаться со Scratch, – щёлкните по значку глобуса  в меню программы и, прокрутив длинный список, найдите русский язык (рис. 1.3).

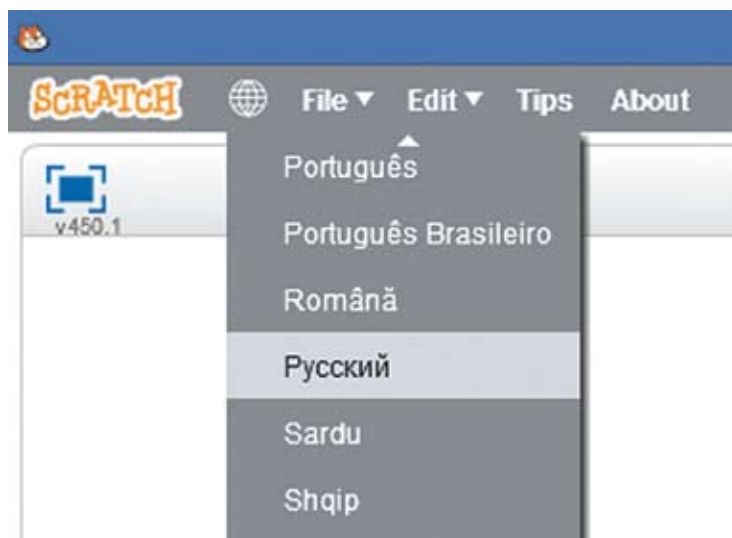


Рис. 1.3. Выбираем язык общения с программой

СЦЕНА И ПЕРСОНАЖИ НА НЕЙ

Взгляните на экран (рис. 1.4). Вот уже знакомый царапающийся персонаж – **спрайт** ①. Спрайт перемещается по специальной области экрана – **сцене** ②.

В середине экрана программы находится **окно блоков** ③, которые мы будем перетаскивать в правую область ④, создавая программы (в терминологии Scratch – **скрипты** или **сценарии**).

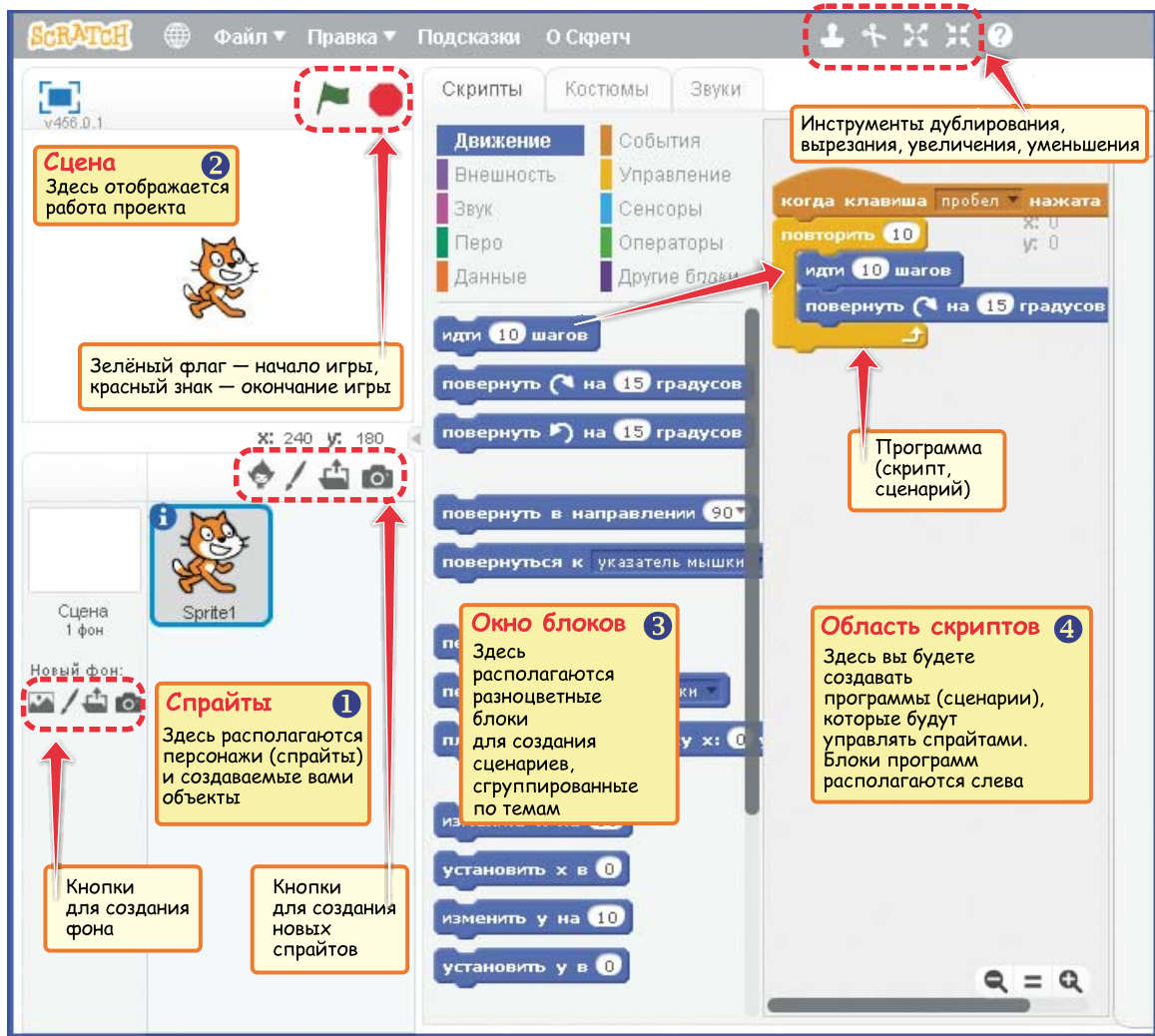


Рис. 1.4. Окно программы Scratch 2.0

ВЫБОР БЛОКОВ ПО ЦВЕТУ

Все блоки распределены по группам-категориям и выделены разными цветами, поэтому очень просто собирать скрипты, которые представлены в нашей книге: посмотрите на цвет блоков, откройте соответствующую по цвету группу и выберите нужные блоки-команды.

КООРДИНАТЫ ПЕРСОНАЖА-СПРАЙТА

Во время выполнения скриптов ваш спрайт будет перемещаться по сцене. Его текущие координаты (положение по осям x и y) отображаются в правом верхнем углу области скриптов (см. рис. 1.4). Команды вроде `идти 10 шагов` иногда вводят в заблуждение, ведь при их исполнении спрайт не собирается бодро шагать 10 человеческих шагов. Он их проходит, конечно, но под **шагами** подразумевается перемещение на указанное количество экранных точек, которых всего в окне сцены 480 по горизонтали (слева направо) и 360 по вертикали (снизу вверх). Точка с координатами $(0, 0)$ находится в центре сцены, координаты X и Y могут быть **положительными** (когда персонаж перемещается вправо или вверх) и **отрицательными** (когда персонаж перемещается влево или вниз) – рис. 1.5.

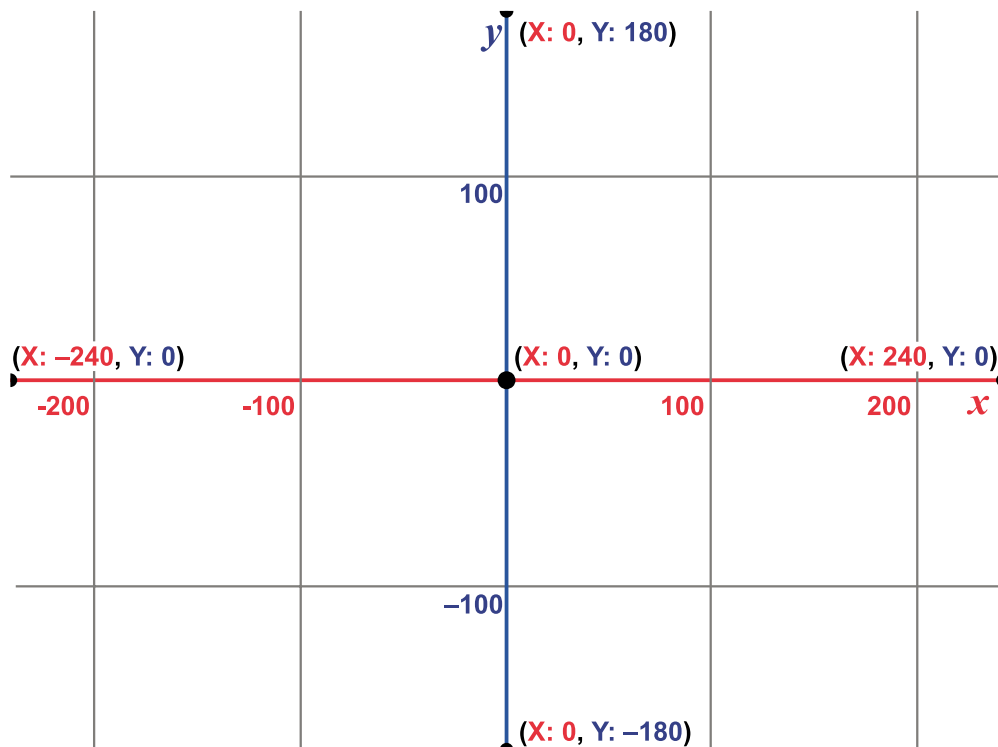


Рис. 1.5. Координаты на сцене Scratch

Ну вот, мы немного освоились, самое время приступить к программированию в этой замечательной среде.

«ЧЕРЕПАШЬЯ ГРАФИКА»

Если вы успели закрыть Scratch, запустите программу вновь, разверните её окно на весь экран и вспомните, какие основные элементы среды Scratch мы уже рассмотрели. Сцена, спрайт, блоки, скрипты, координаты... Всё помните? Готовы продолжить? Тогда начнём... играть.

Учиться программировать мы будем играя. А точнее — рисуя! Вам предстоит освоить «черепашью графику».

ПРИ ЧЁМ ТУТ ЧЕРЕПАХИ?

Это забавное название «**черепашья графика**» (turtle graphics) появилось благодаря математику, программисту и психологу Сэймуру Пейперту и инженеру Идйт Харэль. Они изобрели язык программирования Logo и придумали для него особого исполнителя, имитирующего робота, вооружённого пером. Этот робот, и правда, двигается очень медленно, как черепашка. Зато старательно и аккуратно вырисовывает каждую чёрточку, жужжа моторами, поворачиваясь и переезжая с места на место (рис. 1.6).



Рис. 1.6. Робот Sparki пишет своё имя

В наших проектах вместо черепашки будут спрайты Scratch, а придумывать и программировать рисунки будете вы сами.

ГОТОВИМ ЧЕРЕПАШКУ

После запуска Scratch у нас на сцене всё тот же спрайт-кот. Но рисовать с его помощью будет неудобно, кот большой и будет закрывать от нас часть рисунка. Удалим его и добавим из библиотеки спрайтов маленькую скромную стрелочку с именем **Arrow1**.

Итак, щёлкните правой кнопкой мыши по спрайту кота и выберите команду **удалить** (рис. 1.7).

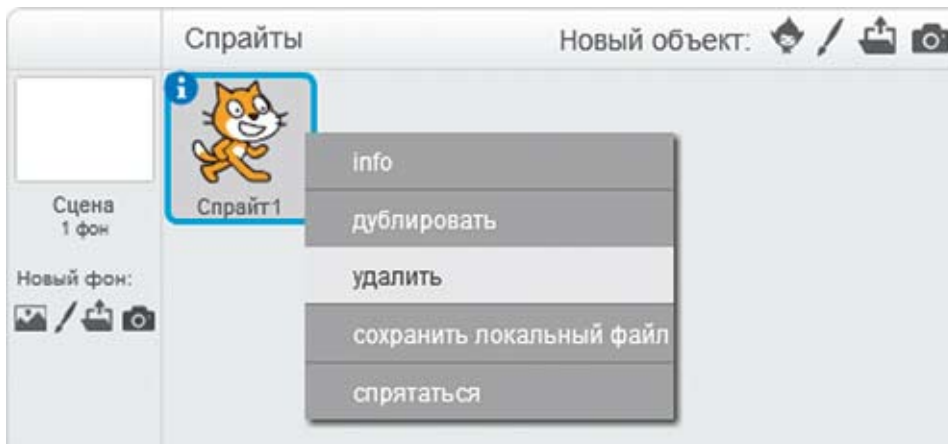



Рис. 1.7. Удаление спрайта с помощью контекстного меню

Щёлкните левой кнопкой мыши (далее – просто «щёлкните») по значку  **Выбрать спрайт из библиотеки** (рис. 1.8).

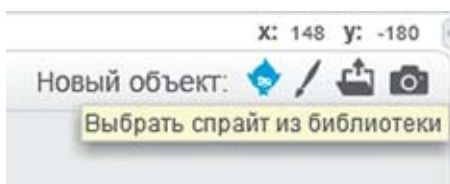


Рис. 1.8. Значок открытия окна библиотеки спрайтов

Откроется окно с **библиотекой спрайтов**. Щёлкните по значку **Arrow1** (рис. 1.9), а затем нажмите кнопку **ОК**.



Рис. 1.9. Спрайт Arrow1 больше подойдёт для рисования

На сцене появилась стрелочка, но даже она слишком велика. Предусмотрим её уменьшение, для этого служит блок **установить размер 100%** из категории скриптов **Внешность**. Перетащите этот блок в область скриптов и вместо 100% введите, например, 20% (то есть у нас будет стрелка в 5 раз меньше). Нам понадобится ещё несколько блоков, в первую очередь **опустить перо** (категория **Перо**), **идти 10 шагов**, **повернуть ↻ на 15 градусов**, **повернуть ↺ на 15 градусов** (категория **Движение**). Вытащите эти блоки в область пользовательских скриптов спрайта Arrow1 (рис. 1.10).

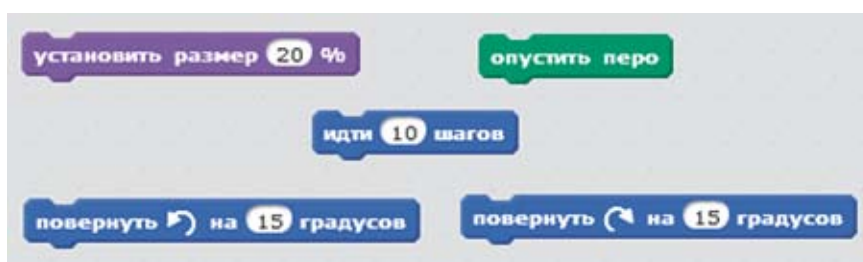


Рис. 1.10. Набор блоков для первой программы

Щёлкните на блоке **установить размер 20%**. Посмотрите: размер спрайта на сцене уменьшился. Щёлкните на блоке **опустить перо**, чтобы начать рисовать. Теперь можно пощёлкать мышью по блокам движения и поворотов и понаблюдать, что получается. Стрелка-черепашка исправно поворачивается, движется, оставляя за собой след. Хотите стереть линии, чтобы попробовать нарисовать новый рисунок? Вытащите блок **очистить** из категории **Перо** и щёлкните по нему.



Внимание!

Не путайте блоки **установить размер 20%** и **изменить размер на 10**. Это распространённая ошибка!

ЗАДАЁМ НАЧАЛЬНОЕ ПОЛОЖЕНИЕ И НАПРАВЛЕНИЕ ДВИЖЕНИЯ СПРАЙТА

Спрайт во время движения может «спрятаться» за край сцены. Для того чтобы не вытаскивать его каждый раз мышью, воспользуемся блоком **перейти в x: 0 y: 0** из категории скриптов **Движение**. Вытащите этот блок на сцену, отредактируйте координаты X и Y , щёлкните по нему мышью. Спрайт окажется в центре экрана. (Помните схему координат на рис. 1.5?) Этот блок поможет задавать положение спрайта при выполнении программы.

Нам нужен ещё один блок из категории **Движение**. Это блок **повернуть в направлении 90**. Он задаёт направление, в котором будет двигаться спрайт. Перетащите его в область сценариев.

ГРУППИРУЕМ БЛОКИ В ПРОГРАММНЫЕ СЦЕНАРИИ

Вы уже обратили внимание, что блоки можно склеивать друг с другом в группы и разрывать группы в произвольном месте? Достаточно мышью ухватить один блок и подвести его к другому — и блоки сцепятся. Сгруппируйте блоки (рис. 1.11).

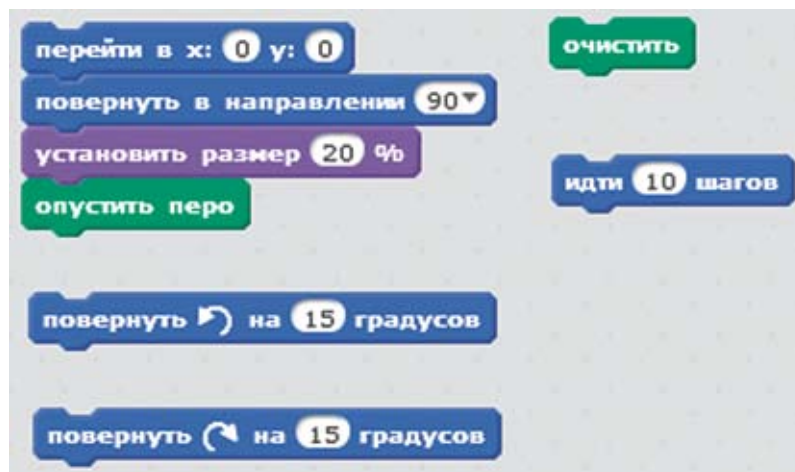


Рис. 1.11. Группируем блоки в скрипт

Теперь одним щелчком на группе можно выполнить сразу несколько действий. Это и называется **скрипт** — программный сценарий некоторых действий.

ДОБАВЛЯЕМ РЕАКЦИЮ НА СОБЫТИЯ

Вы уже заметили, что рисовать, щёлкая мышью по отдельным блокам или даже целым скриптам, неудобно? Приходится всё время переводить взгляд то на область скриптов, то на сцену. Намного проще будет запускать нужные команды нажатием на кнопки компьютерной клавиатуры. В этом помогут специальные вводные блоки из категории **События** (они похожи на «шапочки» скриптов). Нам понадобится три «шапочки» для клавиш <↑> (стрелка вверх), <←> (стрелка влево) и <→> (стрелка направо). Используйте блок **когда клавиша пробел нажата** (перетащите его в область сценариев, выберите нужный вариант из списка).

Заодно заставим нашу стрелку перемещаться чуть шустрее, продвигаясь на 10 шагов и разворачиваясь сразу на 45 градусов (рис. 1.12).

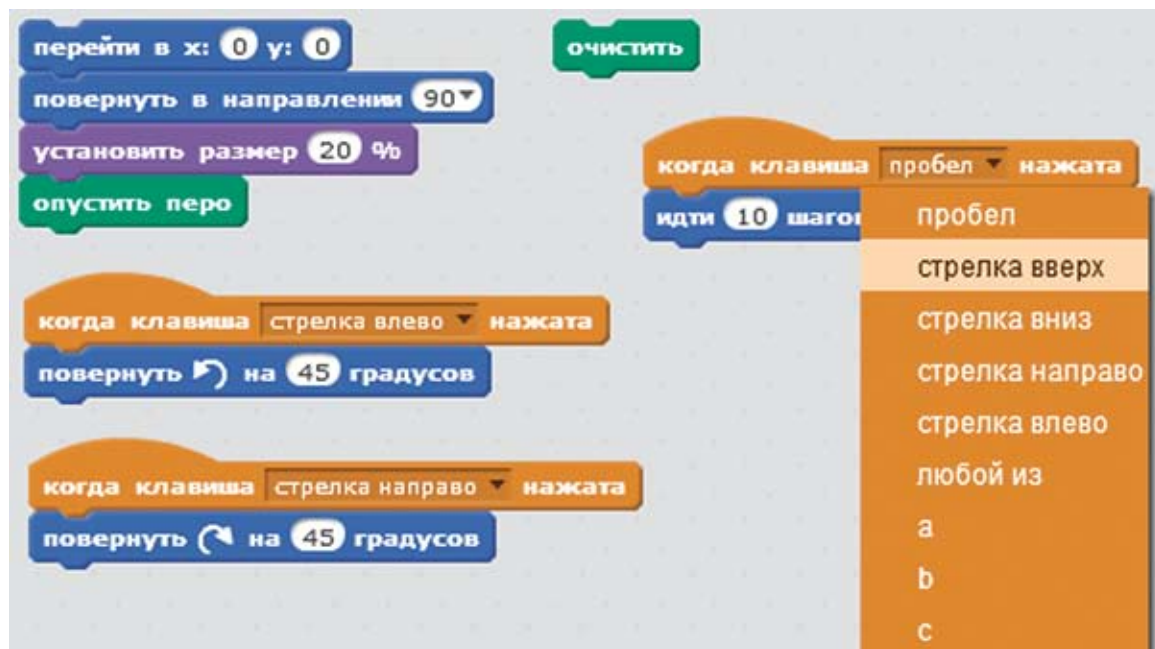


Рис. 1.12. Скрипты для управления с помощью клавиш

Попробуйте с помощью клавиш порисовать на экране. Правда, удобно?! Поздравляем! Вы написали первую программу в среде Scratch! И использовали для её создания событийно-ориентированный подход в программировании.

УЧИМСЯ СОХРАНЯТЬ ПРОЕКТЫ

Для сохранения проекта в меню **Файл** выберите команду **Сохранить как** (рис. 1.13).

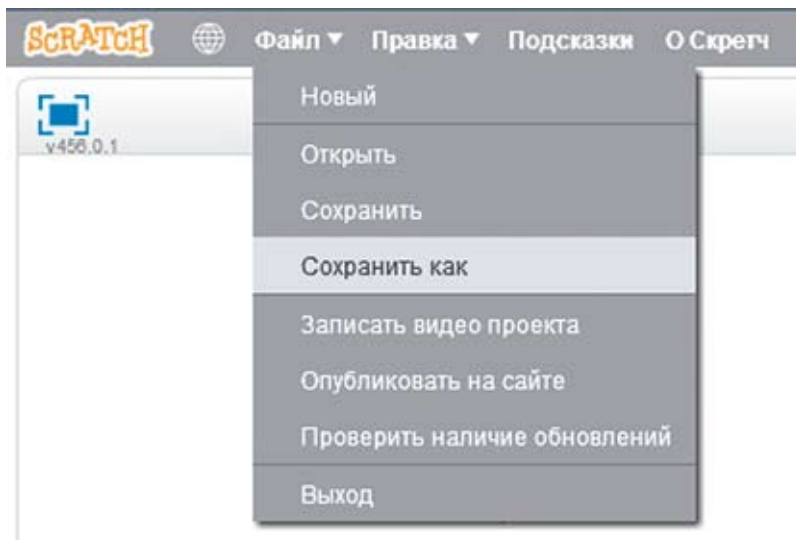


Рис. 1.13. Сохраняем проект

Укажите место сохранения и имя проекта. В дальнейшем можно будет пользоваться командой **Сохранить** из меню **Файл**.

Открыть сохранённый проект можно командой **Открыть** из меню **Файл**.



Задания

1. Добавьте заголовочные блоки к скриптам подготовки к началу рисования (нажатие клавиши <0>, «ноль») и очистки экрана (нажатие клавиши <Пробел>).
2. Измените набор скриптов таким образом, чтобы можно было поднимать и опускать перо в произвольное время, то есть перемещать спрайт без рисования линии. Самостоятельно выберите клавиши для поднятия и опускания пера.

УЧИМ СПРАЙТ ВЫПОЛНЯТЬ СЛОЖНЫЕ СКРИПТЫ

Когда вы порисуете и поэкспериментируете вдоволь, то скорее всего заметите, что для получения симметричного узора (рис. 1.14) приходится повторять однотипные действия, воспроизводя простые элементы рисунка.

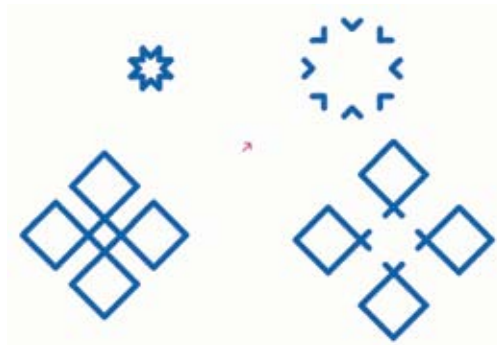


Рис. 1.14. Пример простого симметричного узора

Создадим для таких повторяющихся действий отдельные скрипты. Они будут выполняться при нажатии клавиш <1> и <2>. Составлять узоры станет проще.



Совет

Кстати, в Scratch есть маленькая хитрость: блоки, целые скрипты или их части можно дублировать, ускоряя процесс программирования. Для этого нужно щёлкнуть правой кнопкой мыши по нужному блоку и выбрать команду **дублировать**.

В наш стартовый скрипт добавим пару блоков: **установить размер пера 1**, **установить цвет [] для пера** из категории **Перо**. В последнем случае для того, чтобы выбрать цвет, надо щёлкнуть по цветному квадратику, а потом щёлкнуть по любому месту на экране с нужным цветом.

Теперь отредактируем исходное направление стрелки, пусть при запуске скрипта она будет повернута в направлении 45 градусов (то есть в правый верхний угол экрана). Скрипты становятся всё сложнее (рис. 1.15).