

ДЕНИС ГОЛИКОВ

bhv®

# SCRATCH для юных программистов

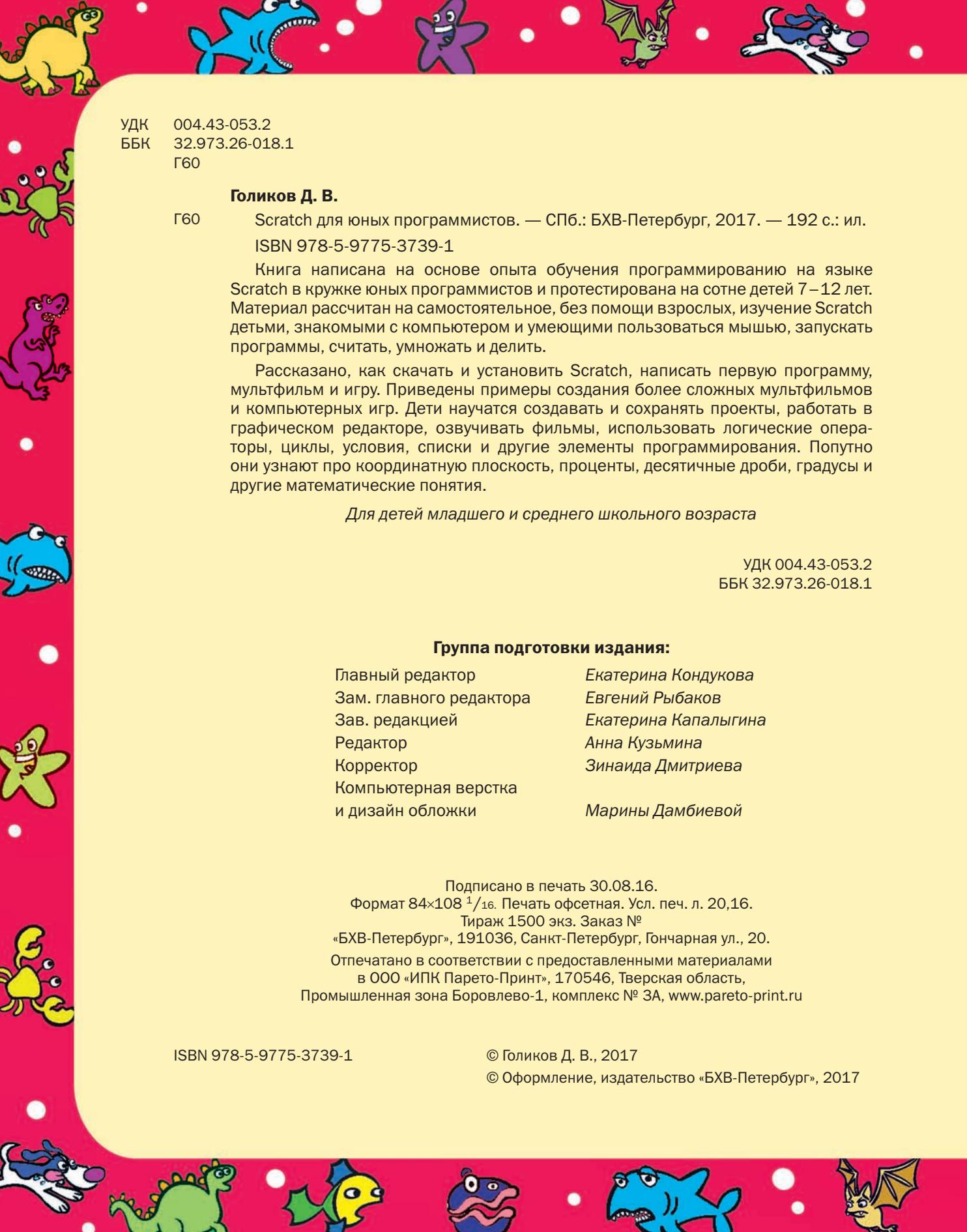


ДЕНИС ГОЛИКОВ

# SCRATCH для юных программистов



Санкт-Петербург  
«БХВ-Петербург»  
2017



УДК 004.43-053.2

ББК 32.973.26-018.1

Г60

**Голиков Д. В.**

Г60      Scratch для юных программистов. — СПб.: БХВ-Петербург, 2017. — 192 с.: ил.

ISBN 978-5-9775-3739-1

Книга написана на основе опыта обучения программированию на языке Scratch в кружке юных программистов и протестирована на сотне детей 7–12 лет. Материал рассчитан на самостоятельное, без помощи взрослых, изучение Scratch детьми, знакомыми с компьютером и умеющими пользоваться мышью, запускать программы, считать, умножать и делить.

Рассказано, как скачать и установить Scratch, написать первую программу, мультфильм и игру. Приведены примеры создания более сложных мультфильмов и компьютерных игр. Дети научатся создавать и сохранять проекты, работать в графическом редакторе, озвучивать фильмы, использовать логические операторы, циклы, условия, списки и другие элементы программирования. Попутно они узнают про координатную плоскость, проценты, десятичные дроби, градусы и другие математические понятия.

*Для детей младшего и среднего школьного возраста*

УДК 004.43-053.2

ББК 32.973.26-018.1

**Группа подготовки издания:**

Главный редактор	Екатерина Кондукова
Зам. главного редактора	Евгений Рыбаков
Зав. редакцией	Екатерина Капалыгина
Редактор	Анна Кузьмина
Корректор	Зинаида Дмитриева
Компьютерная верстка	
и дизайн обложки	Марины Дамбировой

Подписано в печать 30.08.16.

Формат 84×108  $\frac{1}{16}$ . Печать офсетная. Усл. печ. л. 20,16.

Тираж 1500 экз. Заказ №

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

Отпечатано в соответствии с предоставленными материалами  
в ООО «ИПК Парето-Принт», 170546, Тверская область,  
Промышленная зона Боровлево-1, комплекс № 3А, [www.pareto-print.ru](http://www.pareto-print.ru)

ISBN 978-5-9775-3739-1

© Голиков Д. В., 2017

© Оформление, издательство «БХВ-Петербург», 2017



# ВВЕДЕНИЕ ДЛЯ ВЗРОСЛЫХ

Дорогие друзья! (Сейчас я обращаюсь к детям.) Введение можете не читать, переходите сразу к главе 1, начинайте делать весёлые мультики. А родителям будет полезно узнать некоторые сведения о Scratch.

Целью книги является наглядное обучение программированию школьников младших классов. Книга написана на основе опыта обучения программированию на Scratch в кружке юных программистов и протестирована на сотне детей 7–12 лет.

Материал рассчитан на самостоятельное, без помощи взрослых, изучение Scratch школьниками 2–5 классов, имеющими базовые навыки управления компьютером. Дети должны уметь пользоваться мышью, запускать программы, щелкая по их ярлыкам, и т. п., а также считать, умножать и делить. Более сложные математические понятия (отрицательные числа, десятичные дроби, проценты, оси координат, градусы) объяснены на страницах книги.

Особенность книги — очень подробное пошаговое описание процесса создания программ. Все остальные книги опускают многие стороны процесса, подразумевая, что читатель сам догадается о мелких деталях. Здесь же скриншотами (экранными снимками) представлен процесс создания проектов целиком. Все вопросы, возникающие у детей, были сняты в ходе тестирования книги в кружке. Принцип обучения такой: сначала конструируем сложную и непонятную программу (именно конструируем, так как процесс программирования в Scratch подобен созданию моделей из деталей конструктора), потом запускаем её и пытаемся немного изменить. Наблюдая за сделанными изменениями, начинаем понимать, как она работает. В книге нет никакого введения, дети сразу начинают делать весёлые мультики, а потом даже игры. Одна из главных целей — постараться, чтобы ребёнку не было скучно, поэтому шутки и юмор для игр добавлял мой сын пятиклассник.

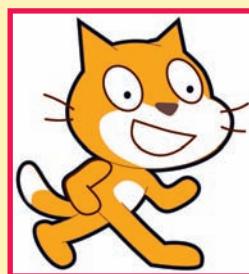




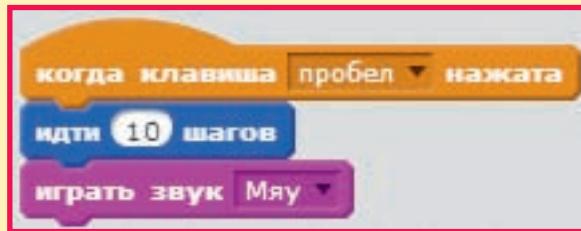
## ЧТО ТАКОЕ SCRATCH?

Scratch — это визуальный язык программирования, в котором программа складывается из разноцветных блоков. Детям ничего не нужно писать, как в других языках программирования. Блоки имеют защёлки, которые не позволяют соединить несовместимые блоки.

Талисманом Scratch является симпатичный рыжий Кот. Он встречает всех, открывших редактор.



Знакомство с программированием начинается с создания простейших программ, например таких, как вот эта.



При нажатии на клавишу <Пробел> эта программа передвинет Кота на 10 шагов и проиграет звук «Мяу».

## КТО СОЗДАЛ SCRATCH?

Проект по созданию Scratch инициирован в 2003 г. при финансовой поддержке компаний Science Foundation, Intel Foundation, Microsoft, MacArthur Foundation, LEGO Foundation, Code-to-Learn Foundation, Google, Dell, Fastly, Inversoft и MIT Media Lab research consortia.





Scratch создан в лаборатории Lifelong Kindergarten Массачусетского технологического института под руководством профессора Митчела Резника (Mitchel Resnick) в 2007 г.



Познакомиться с командой разработчиков Scratch вы можете на странице

<https://scratch.mit.edu/info/credits/>

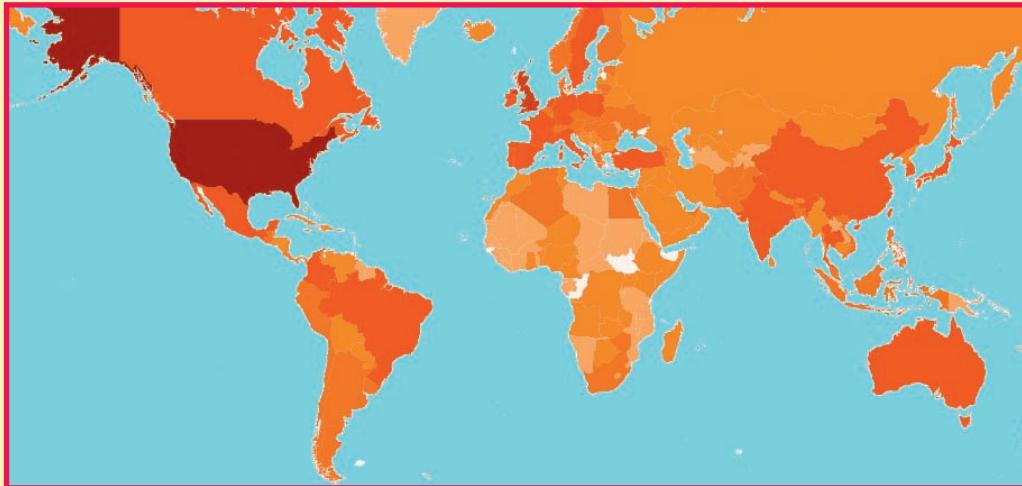
## НА КАКОЙ ВОЗРАСТ РАССЧИТАН SCRATCH?

Создатели Scratch разрабатывали его специально для детей 8–16 лет. Однако 6–7-летние дети, которые умеют читать, считать, а также пользоваться мышью, тоже могут создать простые проекты.

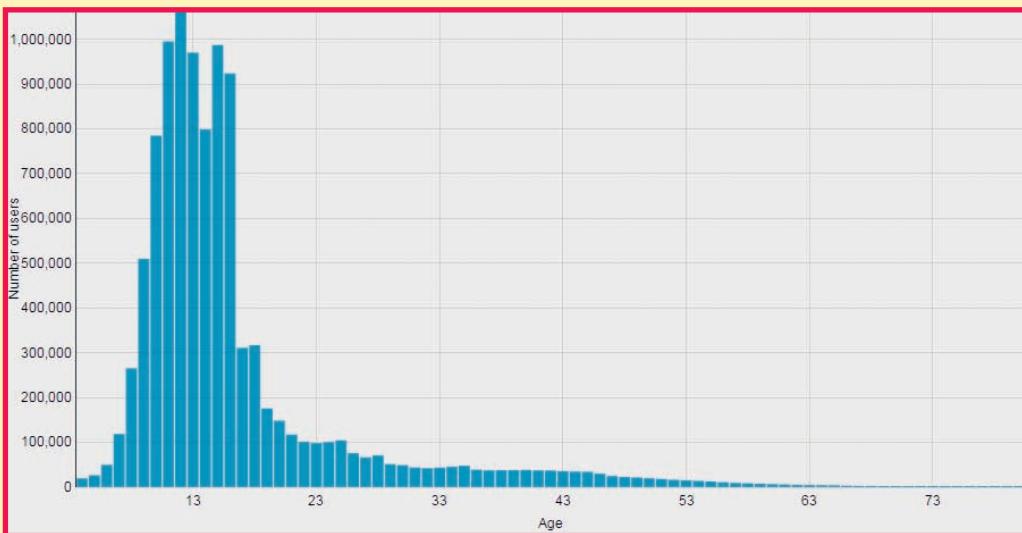
## НАСКОЛЬКО ПОПУЛЯРЕН SCRATCH?

На сайте <http://scratch.mit.edu> зарегистрировано более 13 млн пользователей со всего мира. Из них 5,7 млн из США, 1,4 млн из Великобритании и всего 0,008 млн из России.





Основной возраст участников сообщества 9–16 лет.



Подробную статистику о Scratch можно посмотреть на странице <https://scratch.mit.edu/statistics/>

## ГДЕ НАЙТИ SCRATCH?

Существует два способа работы в среде Scratch. Самый простой способ — работа в онлайновом редакторе Scratch, который можно запустить по адресу:

<https://scratch.mit.edu/projects/editor/>



Для того чтобы иметь возможность сохранять созданные проекты, необходимо зарегистрироваться.

Второй способ работы — работа в оффлайновом редакторе, который можно скачать со страницы

<https://scratch.mit.edu/scratch2download/>

Существуют версии под Windows, Linux и Mac OS X.

Оба редактора Scratch полностью идентичны и имеют одинаковые функции. Проекты, созданные в оффлайновом редакторе, можно загрузить на сайт, и наоборот, проекты, созданные в онлайновом редакторе, можно скачать на свой компьютер.

В книге процесс программирования рассматривается на базе оффлайнового редактора Scratch.

## ГДЕ МОЖНО ИСПОЛЬЗОВАТЬ SCRATCH?

Программирование на Scratch — очень весёлое занятие, поэтому лучше всего заниматься им в группах, тогда дети смогут сразу делиться своими проектами, обсуждать их, совместно придумывать сюжеты.

Scratch идеально подходит для использования на дополнительных уроках в начальных классах (в группах продлённого дня). Дети очень увлекаются созданием проектов, благодаря чему их поведение улучшается.

Scratch можно использовать в библиотеках, оборудованных компьютерами. Там дети могут создавать проекты о героях прочитанных книг, работать совместно.

Scratch хорошо подходит для организации кружков юных программистов на базе учреждений дополнительного образования.

При использовании Scratch дома желательно зарегистрироваться на сайте и размещать все проекты там. На сайте есть большое русскоязычное сообщество, в котором дети смогут найти единомышленников, задавать вопросы и обсуждать проекты.



# ГДЕ НАЙТИ ДОПОЛНИТЕЛЬНУЮ ИНФОРМАЦИЮ О SCRATCH?

Дополнительная информация о Scratch на русском языке:

на странице официального форума по адресу:

- <https://scratch.mit.edu/discuss/27/>;
- в ScratchWiki по адресу <http://scratch-wiki.info/>;
- в Википедии;
- на сайте <http://scratch4russia.com/>.

## О КНИГЕ

### О ЧЁМ УЗНАЮТ ДЕТИ, ПРОЧИТАВШИЕ ЭТУ КНИГУ?

Дети узнают о том, что такое цикл, условный блок, цикл с условием, логическое выражение, координатная плоскость, процент, десятичная дробь, градус, переменная, список.

### ЧЕМУ НАУЧАТСЯ ДЕТИ, ПРОЧИТАВШИЕ КНИГУ?

Дети научатся создавать мультфильмы, игры, сложные скрипты (то есть программы), рисовать в векторном и растровом графических редакторах, изменять звук, вводить, выводить и обрабатывать информацию.

## ПРАВИЛА РАБОТЫ С КНИГОЙ

Книга состоит из 19 глав. Создание проектов разбирается подробно, по шагам, с объяснением новых понятий и блоков. В конце каждой главы приведены задания для самостоятельного выполнения. Главы нужно изучать последовательно, одну за другой, иначе можно пропустить объяснение важных понятий. Будет лучше, если все созданные проекты ваш ребёнок станет выкладывать на сайте <http://scratch.mit.edu>. В этом случае я смогу ответить на



их вопросы и проверить выполнение заданий. Обязательно добавьте меня в друзья на этом сайте. Мой профиль:  
[https://scratch.mit.edu/users/scratch\\_book/](https://scratch.mit.edu/users/scratch_book/)

## УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

Жирным шрифтом выделены элементы интерфейса программы Scratch.

Названия блоков выделены узким шрифтом.

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

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

## УСТАНОВКА SCRATCH

Если вы решили использовать оффлайн-версию программы, прежде всего помогите своим детям установить её на компьютер. Для этого перейдите по ссылке <https://scratch.mit.edu/scratch2download/> и последовательно установите Adobe Air и Scratch Offline Editor.

## ОБ АВТОРЕ

Голиков Денис Владимирович — Scratch-евангелист. Окончил МЭИ по специальности «Промышленная электроника».

В 2013–2015 гг. педагог дополнительного образования по Scratch. В 2014 г. кружок Scratch награждён премией губернатора Московской области.

В 2015 г. финалист Конкурса инноваций в образовании организованного Институтом образования НИУ ВШЭ при поддержке Агентства стратегических инициатив.

## КОНТАКТЫ

- электронная почта автора [scratch.book@ya.ru](mailto:scratch.book@ya.ru);
- страница автора в Facebook <https://www.facebook.com/ScratchBook4u>;

- страница автора в социальной сети «ВКонтакте» VK <http://vk.com/scratch.book>;
- работы автора на сайте Scratch [https://scratch.mit.edu/users/scratch\\_book/](https://scratch.mit.edu/users/scratch_book/);
- сайт автора в Интернете <http://scratch4russia.com/>.

## БЛАГОДАРНОСТИ

Огромное спасибо моему сыну Артёму, который помогал придумывать игры и шутки для книги.

Выражаю благодарность всем детям, посещавшим в 2013–2015 гг. «Кружок юных программистов» в г. Химки. Без вас написание этой книги было бы невозможным.

Огромное спасибо коллективу издательства «БХВ-Петербург» и лично Евгению Рыбакову и Анне Кузьминой.



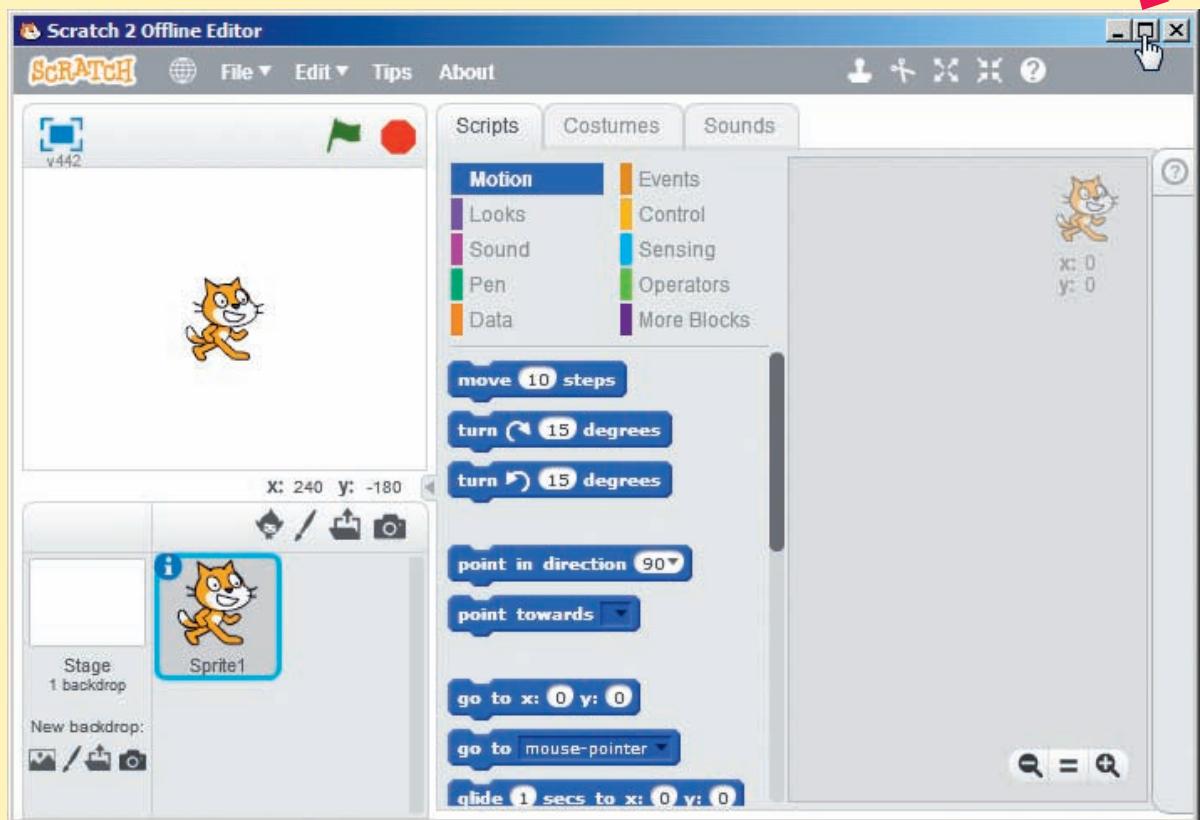
# ГЛАВА 1. ЗНАКОМСТВО СО SCRATCH

## 1.1. ЗНАКОМСТВО С ИНТЕРФЕЙСОМ

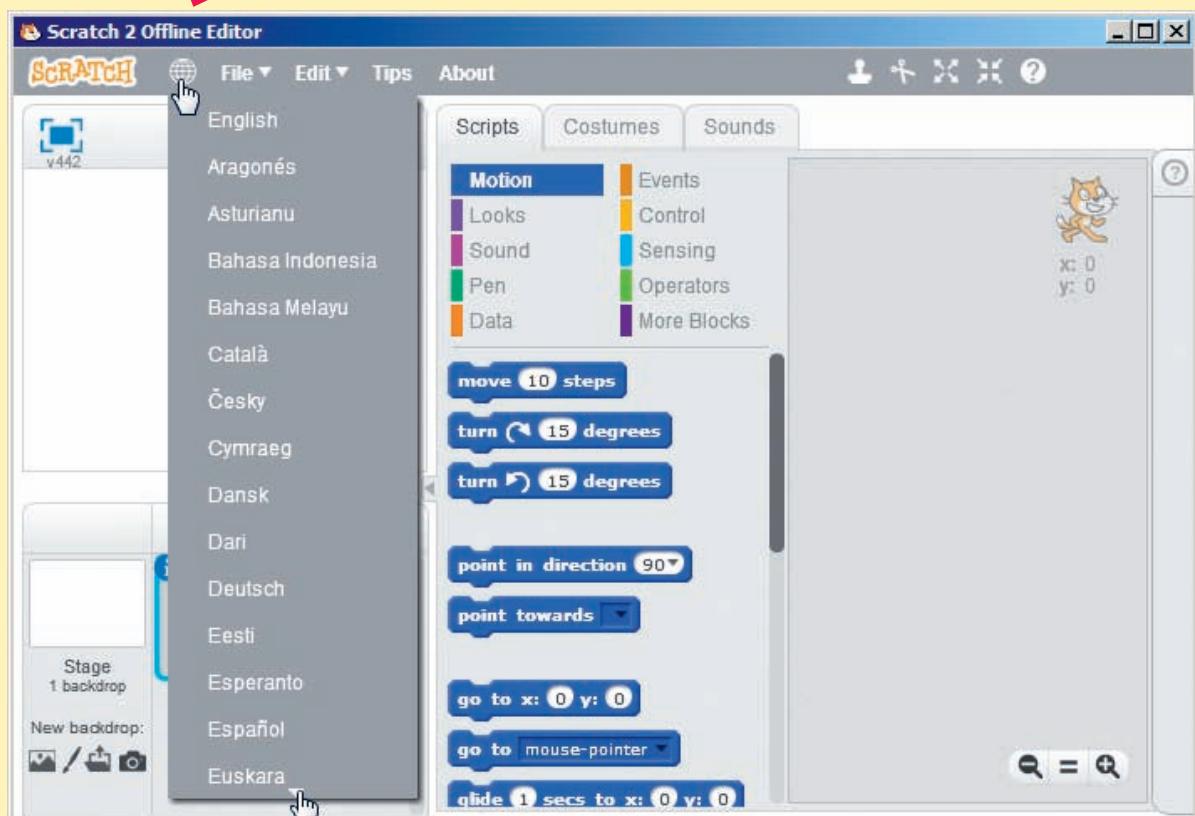


Запустите Scratch с помощью ярлыка на рабочем столе — дважды щёлкните мышью по значку.

Откроется окно программы. Разверните его на весь экран, нажав, как обычно, на среднюю кнопку — **квадратик** — в правом верхнем углу окна.



Если в запущившейся программе вы увидите надписи на английском языке, то первым делом надо переключить её на русский интерфейс. Для этого щёлкните на значке **глобуса** в строке меню, а затем аккуратно двигайте мышь на маленький треугольничек в самом конце списка языков.



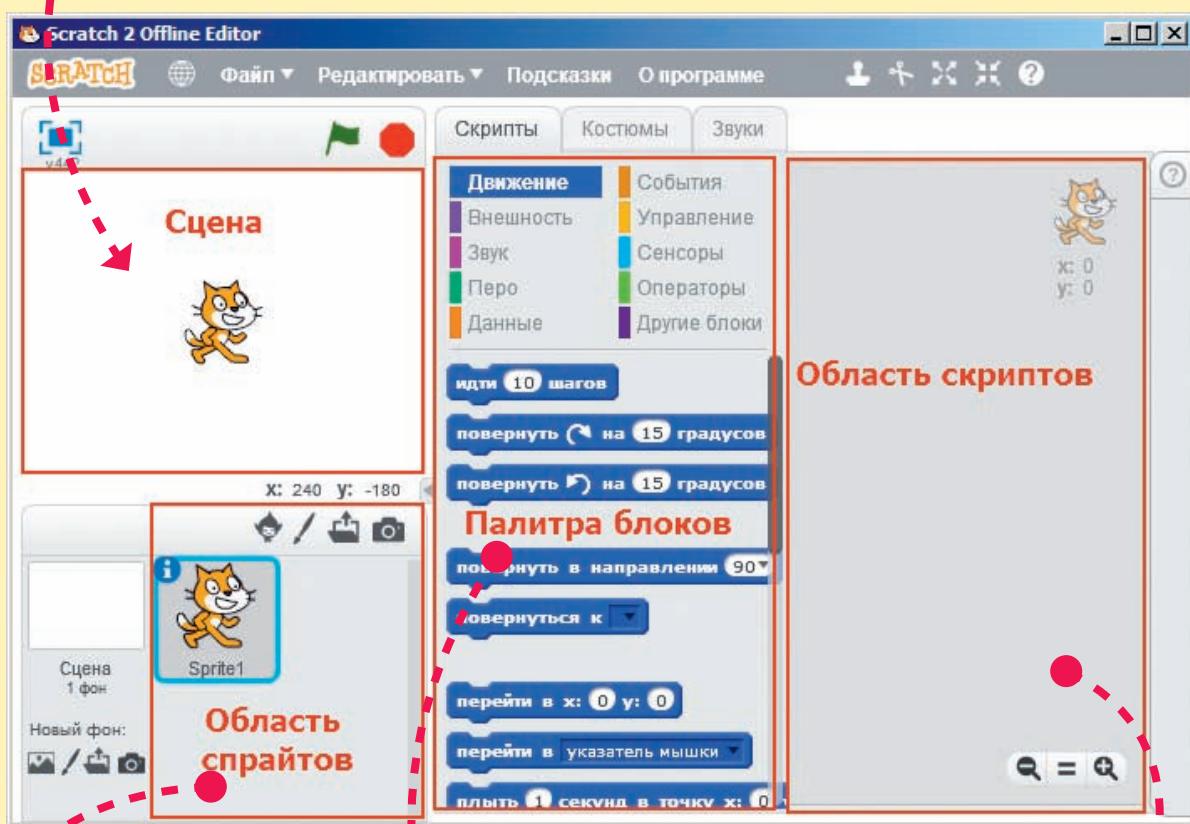


Прокрутите список вниз до самого конца и выберите русский язык.

Теперь можно осмотреться.



Белое поле слева — это **сцена**, на ней будет видно, как работает проект. По сцене будут перемещаться **спрайты** (персонажи), на ней вы будете рисовать и изменять её фон. Сейчас на сцене всего один спрайт — Кот.



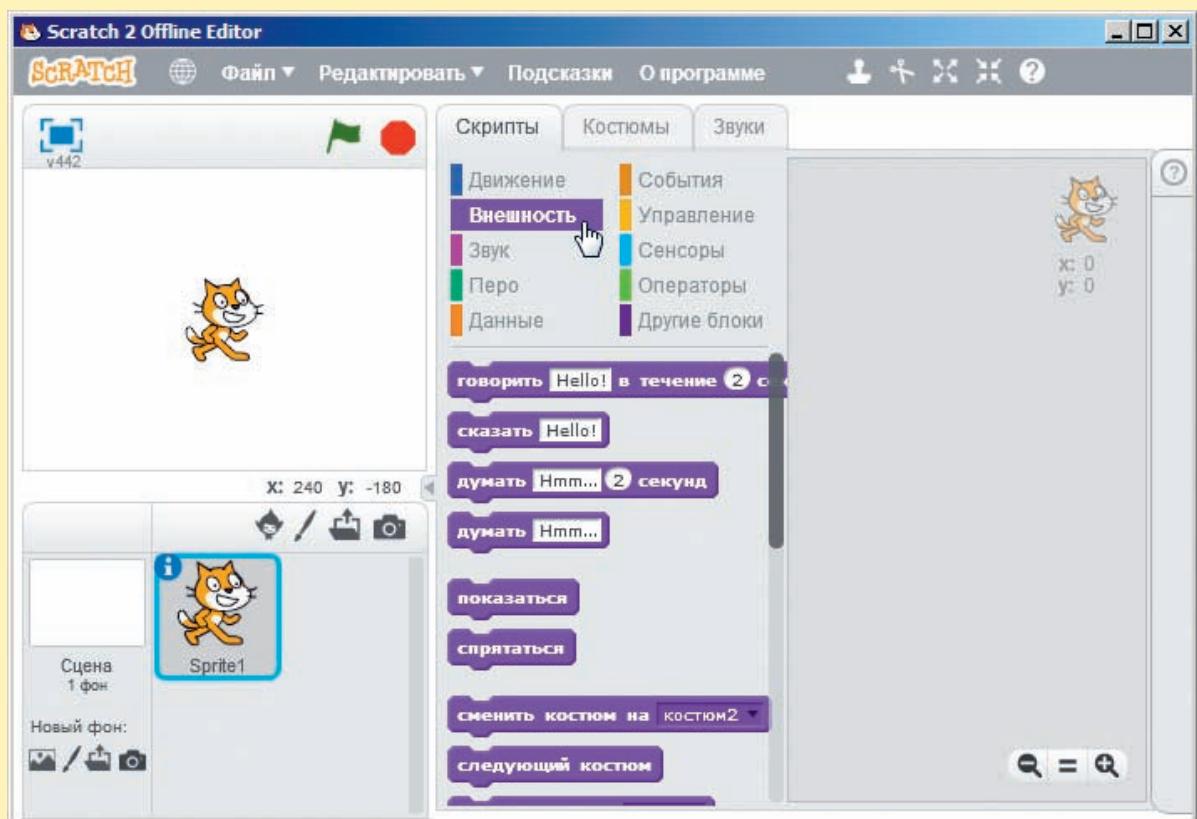
Все спрайты проекта находятся в **области спрайтов**, которая расположена под сценой.

Справа — огромная **область скриптов**, там мы будем собирать скрипты проекта из разноцветных **блоков**, которые хранятся в **палитре блоков**, расположенной в самом центре экрана.



Скрипты — это кусочки, наборы цветных блоков, из которых состоит программа. У каждого спрайта скрипты собственные, описывающие поведение только этого спрайта.

Сейчас выбраны синие блоки **Движение**. Выберите фиолетовые блоки **Внешность**.

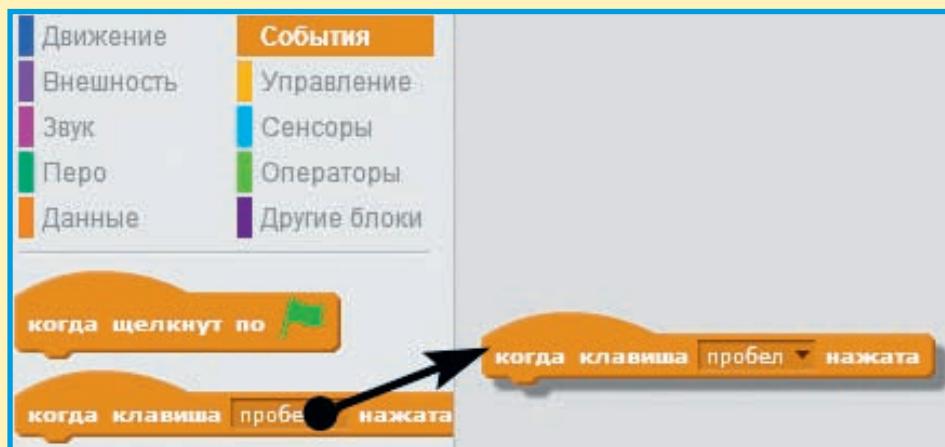


Пощёлкайте по блокам других цветов. С остальными вкладочками и кнопочками мы познакомимся позднее, а теперь пришло время сделать первый проект!

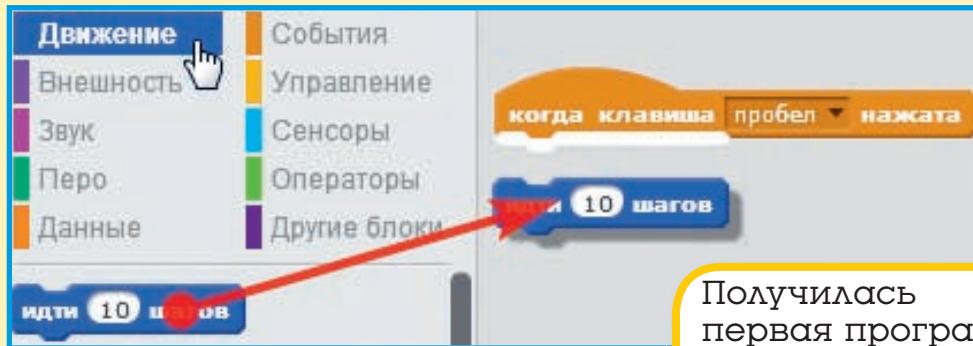


## 1.2. ПЕРВЫЙ ПРОЕКТ

Выберите блоки **События**. Щёлкните мышью на блоке когда клавиша пробел нажата и, не отпуская левую кнопку мыши, тяните его в область скриптов.



Расположите блок в верхней части области скриптов и отпустите левую кнопку мыши. Затем выберите синие блоки **Движение** и вытащите в область скриптов блок иди 10 шагов. Тащите его прямо к первому блоку. Когда он захочет к нему прицепиться, то появится белая полоса, в этот момент отпускайте левую кнопку мыши — блок встанет на место.



Получилась первая программа, состоящая из одного скрипта.

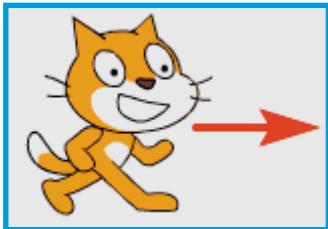


### Важно!

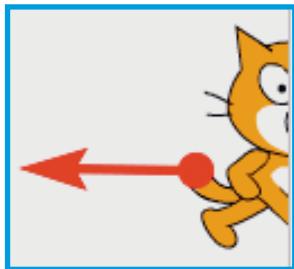
Каждый скрипт начинается с блока **События** с круглой «шапочкой». Скрипт выполняется сверху вниз. Каждый блок по очереди выполняет своё действие.



Нажмайте клавишу <Пробел> и посмотрите, что будет происходить на сцене. Котик пойдёт направо!



Если нажимать клавишу <Пробел> много раз, то Котик скроется за краем сцены. Вытащите его обратно за хвост.

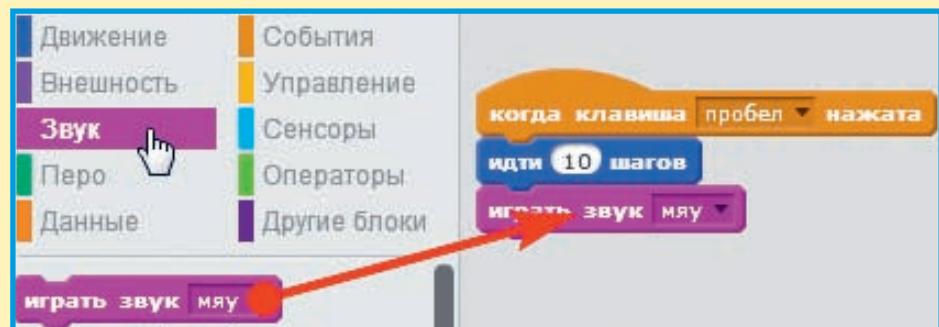


Теперь нажмите клавишу <Пробел> и не отпускайте её — Кот побежит! Снова вытащите его на середину сцены. Отлично! Бегать Кота научили, теперь научим его мяукать!

## 1.3. БЛОКИ ЗВУКА

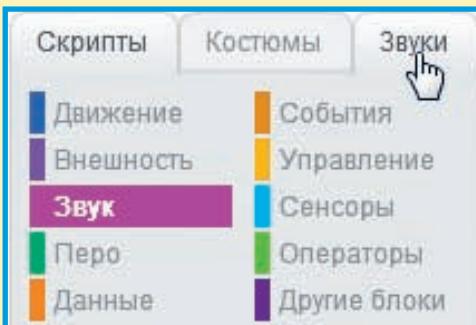
Выберите сиреневые блоки **Звук**. Прицепите к скрипту блок играть звук мяу (meow).

Снова нажмите клавишу <Пробел> — Кот идёт и мяукает!

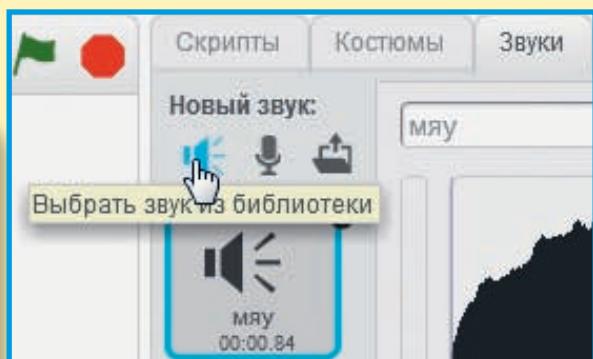


### Совет

Если Кот не мяукает, то проверьте громкость звука и включите колонки.



Мяукать Кота научили, а теперь попробуем научить его лаять! Щёлкните на вкладке **Звуки**.



Затем нажмите кнопку **Выбрать звук из библиотеки**.

Откроется библиотека звуков. Она содержит более 120 различных звуков, мелодий и эффектов.



## Совет

Для того чтобы прослушать звуки, нажмите кнопку Play.

alien creak2      beat box1  
bird      birthday

Выберите слева категорию **Животное**, в центре окна найдите звук dog1 и нажмите кнопку **OK**.

Библиотека звуков

Категория

Все

Животное **dog1**

Эффекты

Электроника

Человек

Инструменты

Музыкальная петля

Скрипты Костюмы

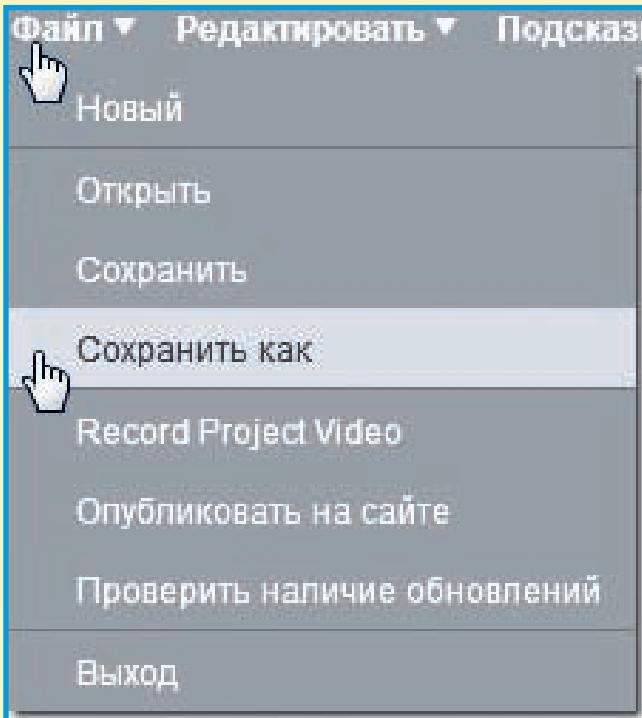
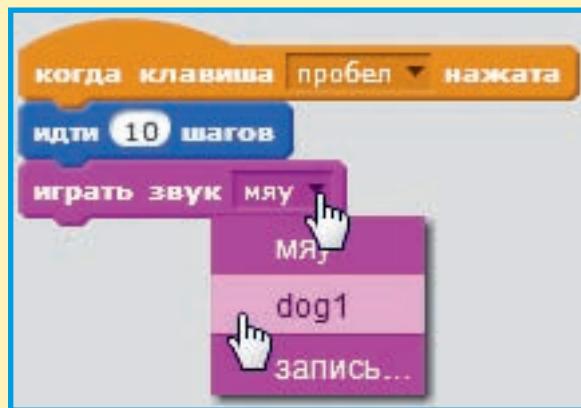
Новый звук:

1 Мяу 00:00:84

2 dog1 00:00:16

Теперь Кот может использовать два звука — либо «мяу», либо «dog1».

Перейдите на вкладку **Скрипты**. Нажав на чёрный треугольничек, раскройте список звуков в сиреневом блоке и выберите звук «dog1».

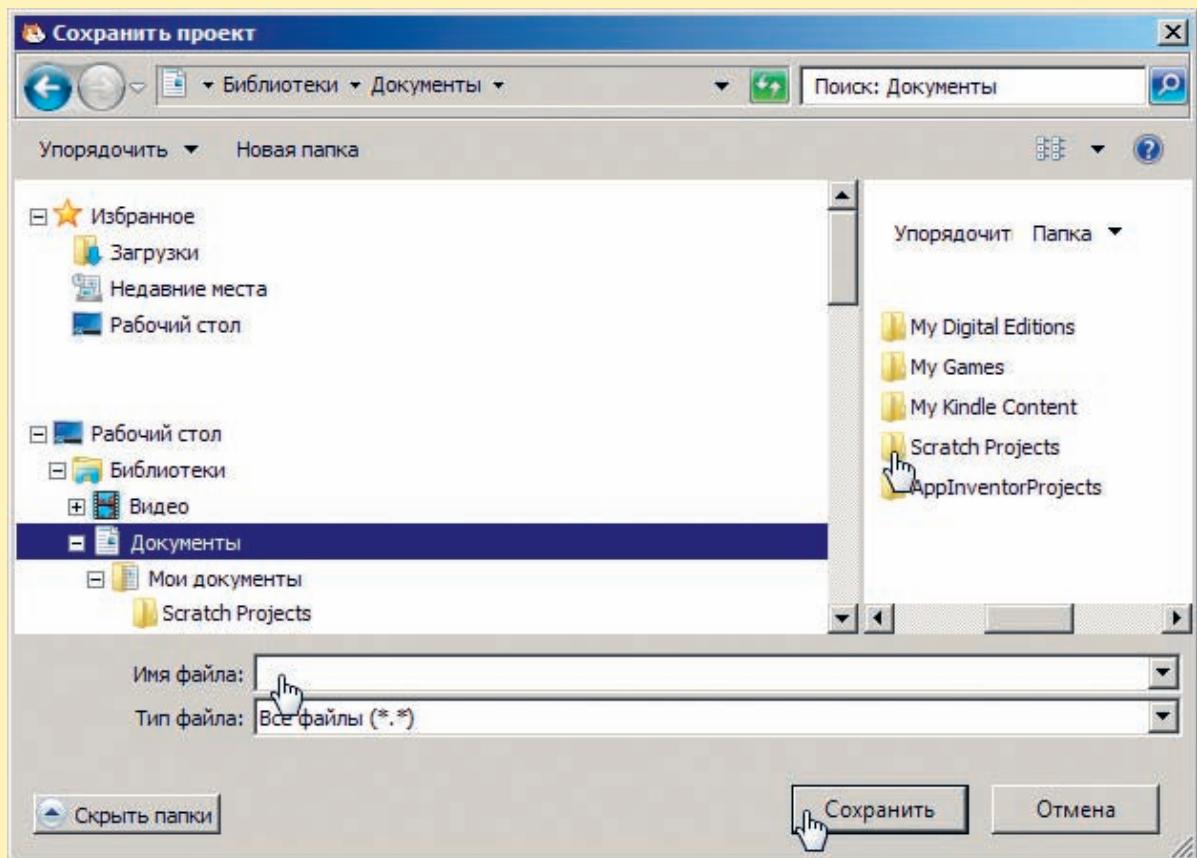


Нажимайте клавишу <Пробел>, теперь Кот бегает и лает! Так гораздо веселее!  
Ну, вот вы и сделали свой первый проект на Scratch, осталось только сохранить его на память. Для этого щёлкните в строке меню на команде **Файл**, а затем подведите указатель мыши к строке **Сохранить как**. Далее в книге я буду называть это действие так: в меню **Файл** выберите команду **Сохранить как**.



Лучше всего для сохранения проектов подходит папка Scratch Projects, которая автоматически появилась после установки Scratch.

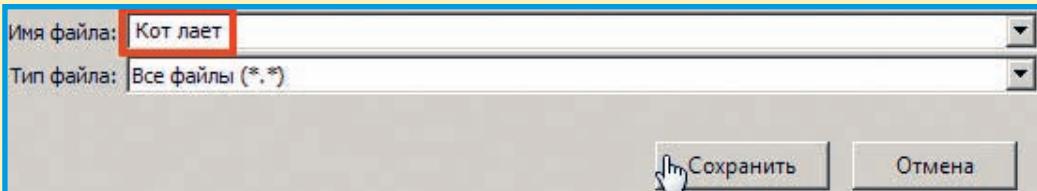
Введите понятное имя файла, например «Кот лает», и нажмите кнопку **Сохранить**.



### *Совет*

Всегда давайте файлам осмысленные имена. Если вы напишете что-то вроде fkjvkdb345, то через некоторое время не сможете вспомнить, о чём же этот проект, а когда таких проектов станет много, то вы совсем запутаетесь.



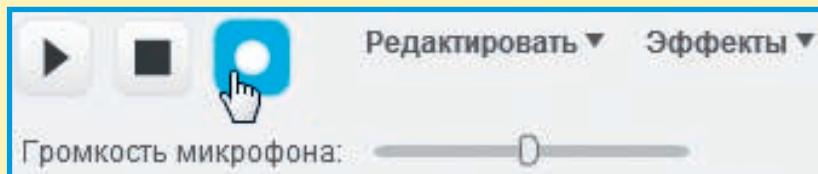


Первый проект сохранён.

## 1.4. ЗАДАНИЯ

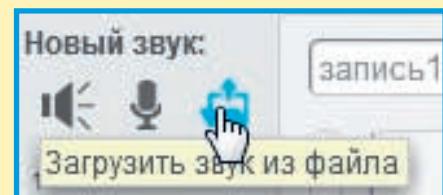
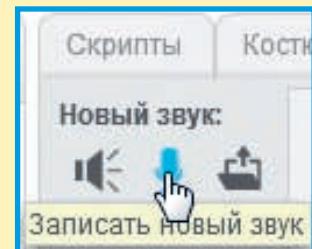
1. С помощью кнопки **Записать новый звук** (на вкладке **Звуки**) попробуйте записать звук с микрофона.

Для начала записи нажмите на кнопку с **кружочком**.



Для остановки записи нажмите кнопку с **квадратом**, а для прослушивания — кнопку с **треугольником**. Всё как на настоящем проигрывателе!

2. С помощью кнопки **Загрузить звук из файла** загрузите в проект свою любимую композицию из файла с расширением mp3. (Надеюсь, вы знаете, что такое расширение файла?)





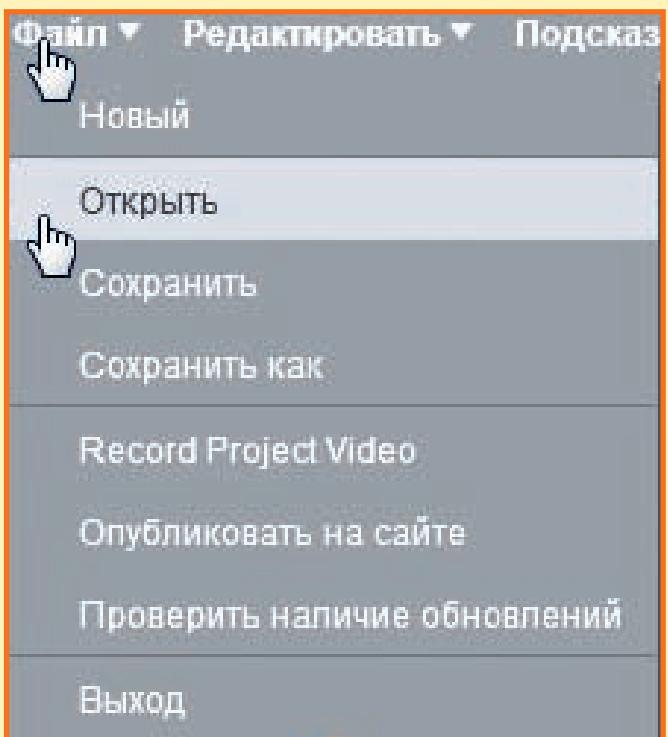
## ГЛАВА 2. УСЛОЖНЕНИЕ ПЕРВОГО ПРОЕКТА

### 2.1. ЗАГРУЗКА ПЕРВОГО ПРОЕКТА

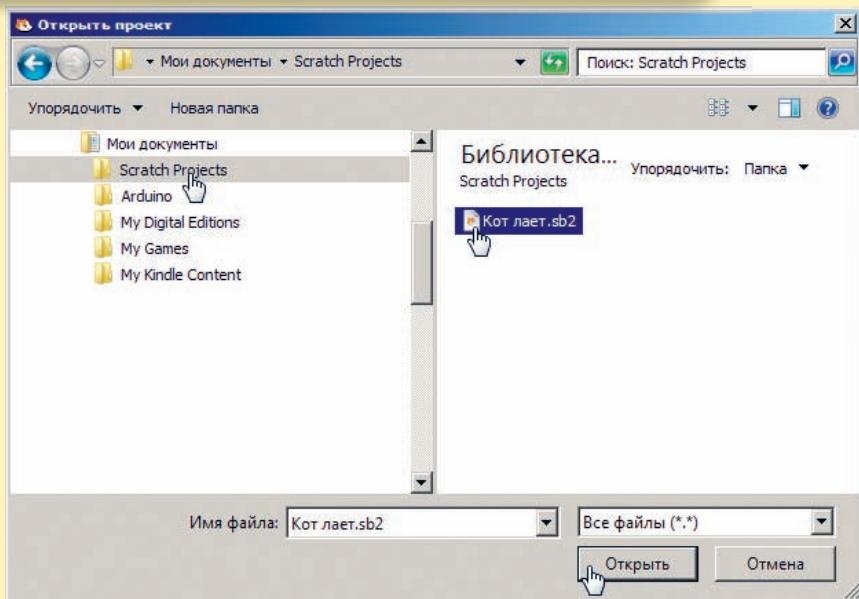


Запустите Scratch с помощью ярлыка на рабочем столе.

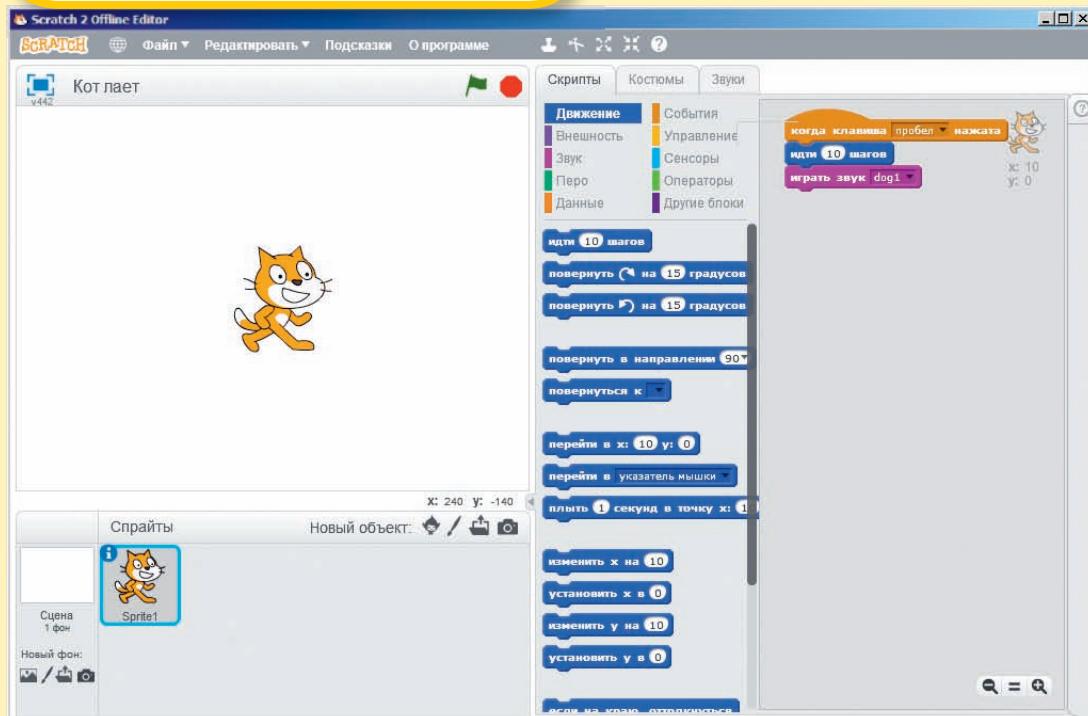
В меню **Файл** выберите команду **Открыть**.



Перейдите в папку, в которой был сохранён первый проект, выберите его и нажмите кнопку **Открыть**.



Откроется первый проект.





## 2.2. ИЗМЕНЕНИЕ СКОРОСТИ ДВИЖЕНИЯ

Первым делом удалите блок звука, чтобы лай не отвлекал вас от знакомства с новыми блоками. Для этого нажмите на сиреневый блок звука и тащите его обратно в палитру блоков. Отпустите левую кнопку мыши. Когда блок будет находиться над палитрой, он исчезнет.

**Движение**

- Внешность
- Звук
- Перо
- Данные

**События**

- Управление
- Сенсоры
- Операторы
- Другие блоки

```

когда клавиша пробел нажата
идти 10 шагов
играть звук dog
и если на краю, оттолкнуться

```

Теперь добавьте к скрипту новый блок из набора блоков **Движение** — если на краю, оттолкнуться.

```

когда клавиша пробел нажата
идти 10 шагов
если на краю, оттолкнуться

```

Нажмайте клавишу <Пробел> и посмотрите, что произойдёт, когда Кот дойдёт до края экрана. Он оттолкнётся и пойдёт в обратном направлении! Теперь вам не придётся вытаскивать его за хвост.

А теперь попробуйте изменить скорость движения Кота. Для этого надо изменить значение в блоке **идти**. Щёлкните на числе **10**. Оно посинеет.

```

когда клавиша пробел нажата
идти 10 шагов
если на краю, оттолкнуться

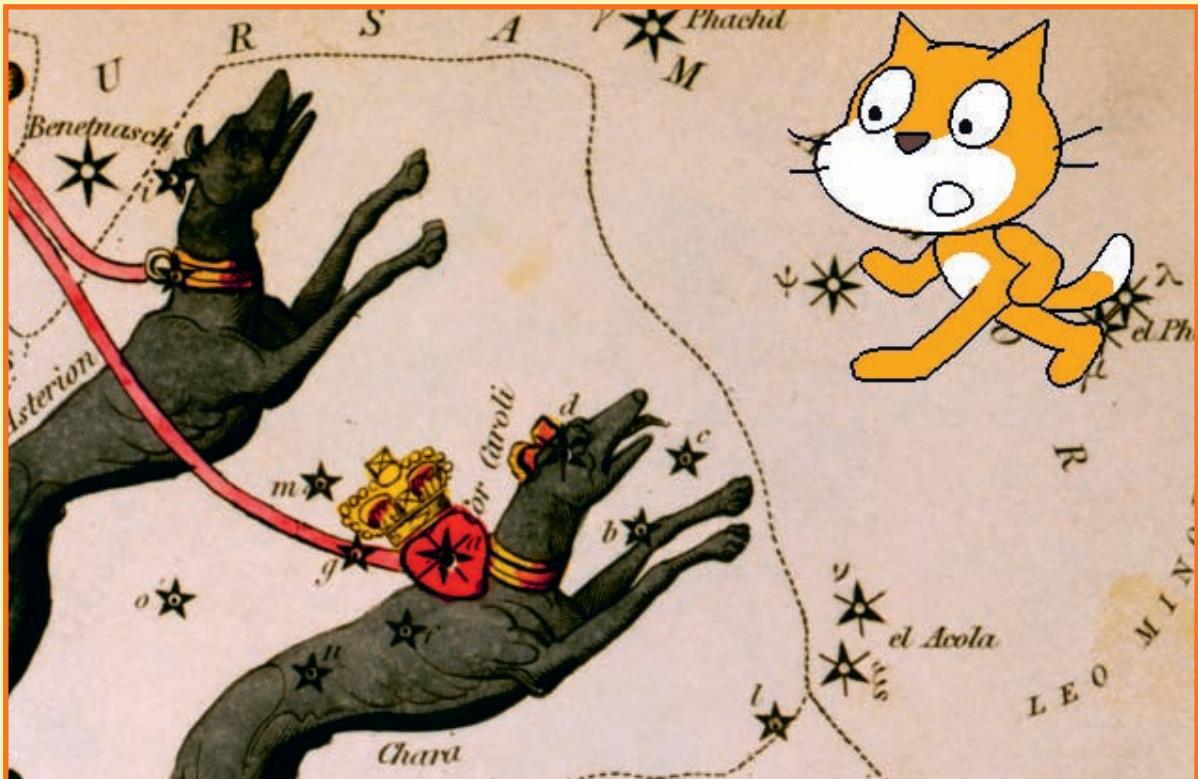
```





### Справка

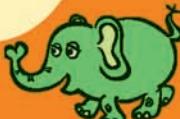
Гончие псы — созвездие, расположенное прямо под ковшом Большой Медведицы.



Ох, и не весело пришлось Котику в этой компании!

Если вы ввели очень большое число и отправили Кота в космос, то существует только один способ вернуть его обратно — лететь в космос за ним. Вы готовы?

Шучу! На самом деле лететь  
никуда не нужно. Надо  
просто щёлкнуть на красном  
восьмиугольнике.





Не хочу к Гончим псым!

Х: 240 У: -180

Мой объект: /

А затем дважды щёлкнуть на блоке перейти в указатель мышки. Кот появится в правом нижнем углу сцены.

Как вы уже заметили, при движении влево Кот всегда переворачивается вверх тормашками. Это не похоже на движение настоящих котов, обычно они всегда передвигаются тормашками вниз.

Существует блок, который запрещает спрайту переворачиваться. Это стиль вращения влево-вправо. Прицепите этот блок к скрипту.

когда клавиша пробел нажата

идти 10 шагов

если на краю, оттолкнуться

стиль вращения влево-вправо

Теперь Котик будет бегать ногами вниз. Проверьте, так ли это.

В Scratch существует всего три стиля вращения: «влево-вправо», «не вращать» и «кругом».

стиль вращения влево-вправо

влево-вправо

не вращать

кругом

У каждого нового спрайта, добавленного в проект, установлен стиль вращения «кругом», поэтому блок стиля вращения мы будем использовать очень часто.

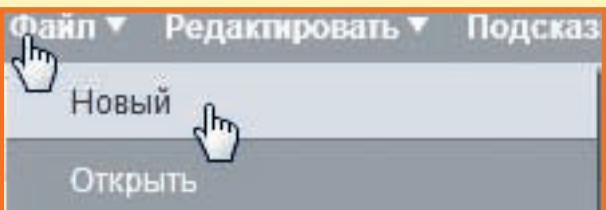
Не забудьте сохранить проект: в меню **Файл** выберите команду **Сохранить**.



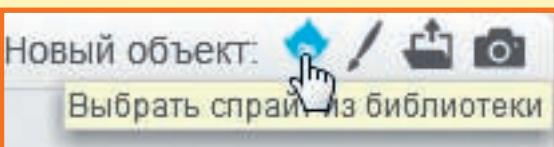
## 2.3. АВТОМОБИЛЬ С ПЯТЬЮ СКОРОСТЯМИ

Вы научились перемещать спрайты с различной скоростью, а теперь давайте создадим новый проект, в котором автомобиль, как в жизни, будет иметь пять скоростей.

В меню **Файл** выберите команду **Новый**.



Появится новый проект, в котором, как обычно, есть только Кот. Но нам же нужен автомобиль! Давайте загрузим его из библиотеки спрайтов. Для этого в области спрайтов щёлкните на значке маленького человечка.



### Библиотека спрайтов

#### Категория

Все

Животные

Фантастика

Буквы

Люди

Предметы

Транспорт

Библиотека спрайтов очень похожа на библиотеку звуков. Выберите слева категорию **Транспорт**.





Выберите понравившийся  
вам автомобиль и нажмите  
кнопку **OK**.



Convertible1

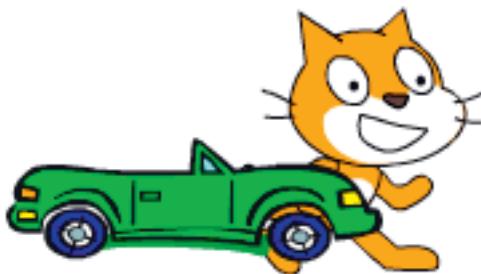


Convertible2



Convertible3

Теперь на сцене два спрайта.



Они оба показаны  
в области спрайтов.

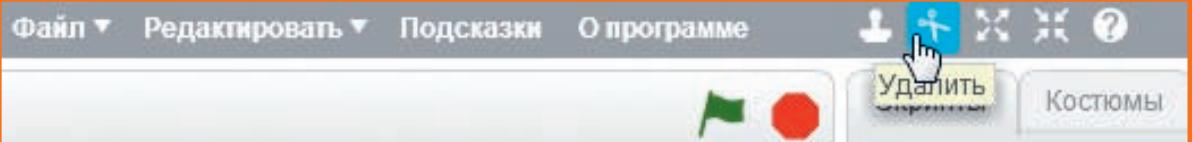


Sprite1



Convertible3

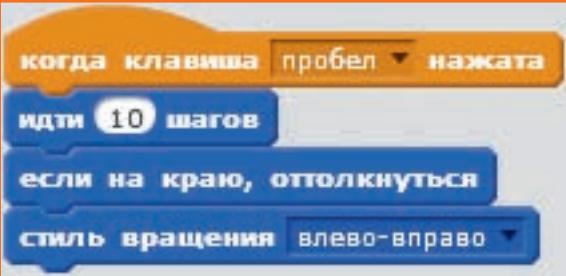
Этот проект будет про автомобиль, поэтому  
Кот не нужен. Удалите его, щёлкнув на значке  
**ножниц** в строке меню, а затем на Котике.



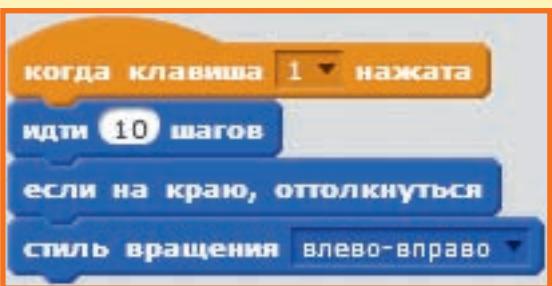
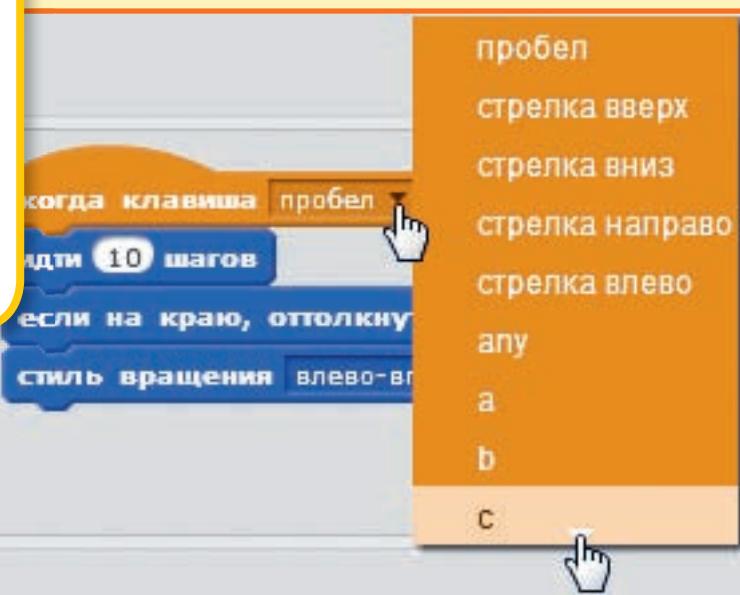


Остался только один спрайт — автомобиль. Теперь его надо запрограммировать. У автомобиля будет 5 скоростей, поэтому нужно создать 5 разных скриптов.

Сначала запрограммируем  **первую скорость**. Постройте вот такой скрипт.



Теперь переключим управление с клавиши <Пробел> на клавишу <1>. Для этого раскройте выпадающий список, нажав на чёрный треугольничек в блоке когда клавиша пробел нажата, и двигайте курсор вниз.

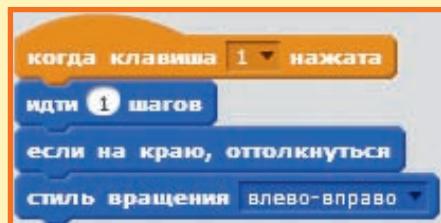


Выберите единичку.





Теперь измените значение в блоке идти на 1. На первой скорости машина будет ехать очень медленно.

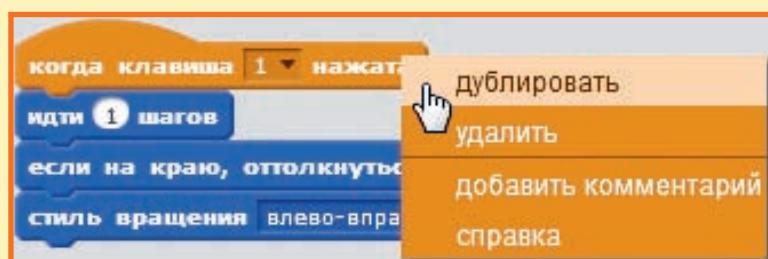


Первый скрипт автомобиля готов! Протестируйте его. Нам надо сделать ещё четыре похожих скрипта.

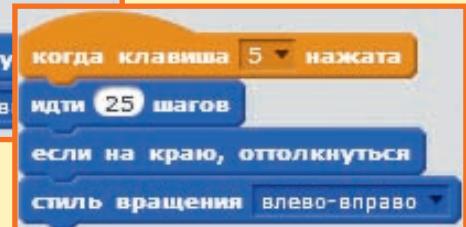
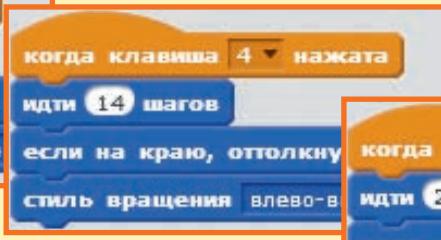
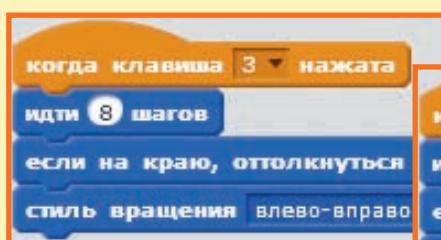
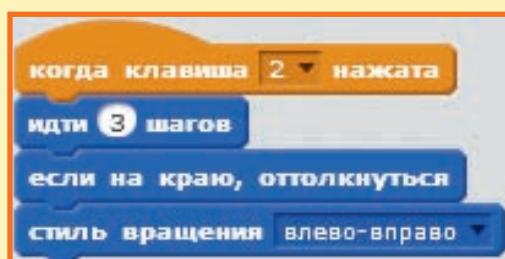


### Совет

Есть простой способ скопировать скрипт. Щёлкните на верхнем блоке скрипта правой кнопкой мыши и выберите команду дублировать.



Дублируйте первый скрипт и измените клавишу управления на 2, а скорость перемещения на 3. Сделайте ещё три скрипта.





Проект почти готов, давайте украсим сцену — добавим красивый фон. Нажмите кнопку **Выбрать фон из библиотеки**.

Выберите тему фона — **Город**.

### Тема

Замок

Город

Полёт

Выходные

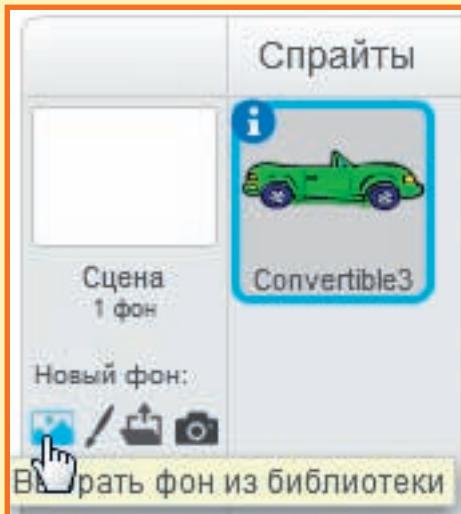
Музыка и танцы

Природа

Космос

Спорт

Подводный мир



Выберите фон, например вот этот, и нажмите кнопку **OK**.



urban1



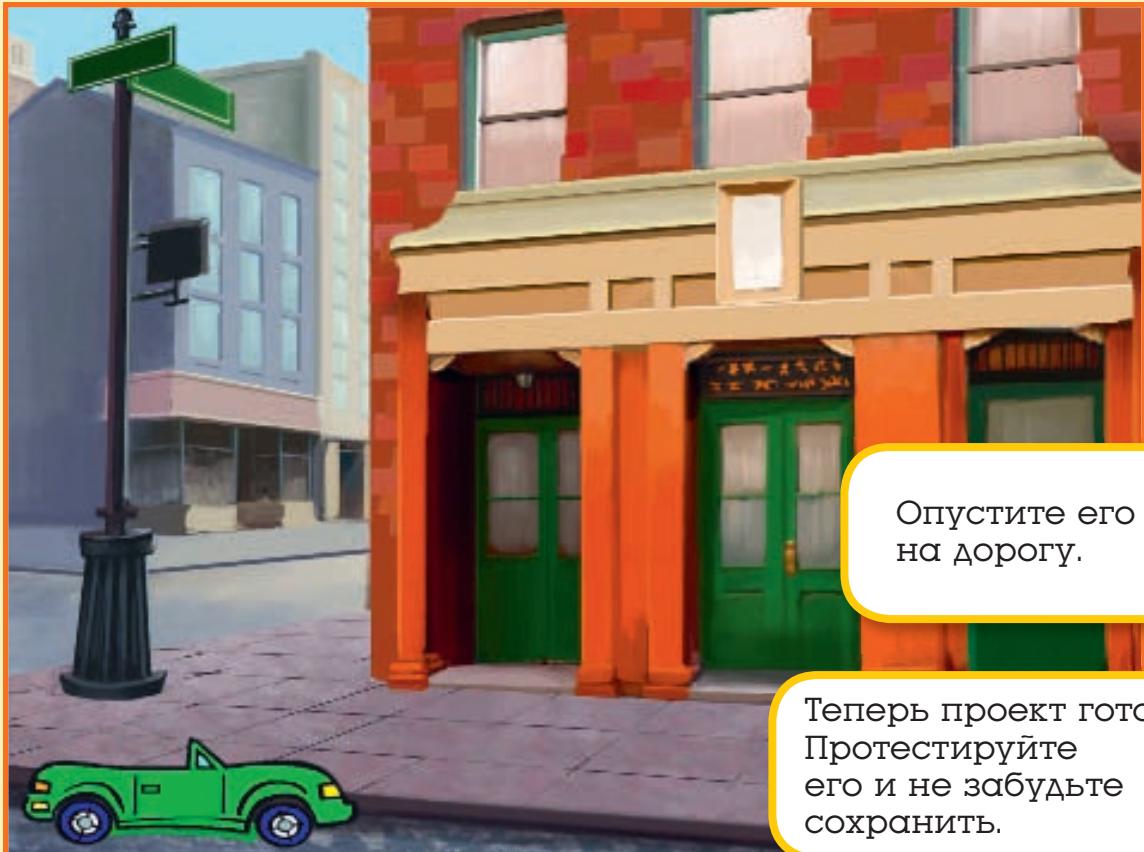
urban2





Автомобиль расположен не очень  
удачно — он стоит на фонаре.





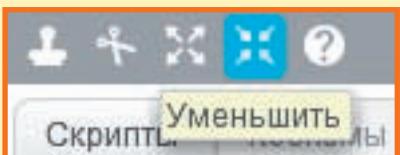
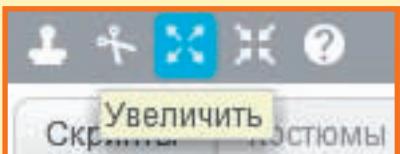
## 2.4. ЗАДАНИЯ

1. Добавьте автомобилю шестую скорость.

2. Немного увеличьте размер автомобиля с помощью инструмента **Увеличить**.

3. Добавьте в проект пешеходов, которые будут стоять на тротуаре. Измените их размер с помощью инструмента **Уменьшить**.

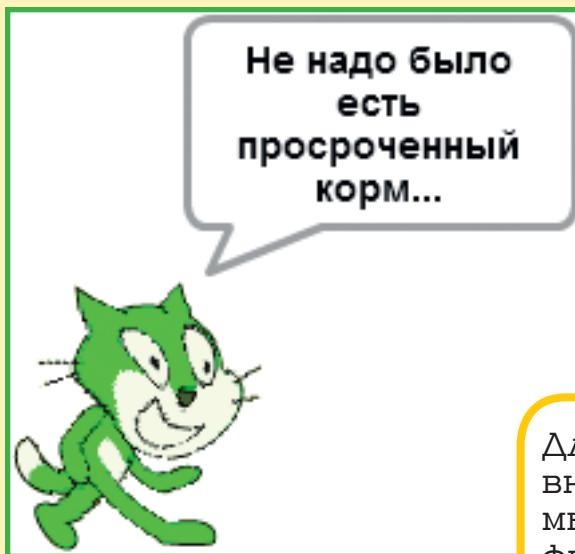
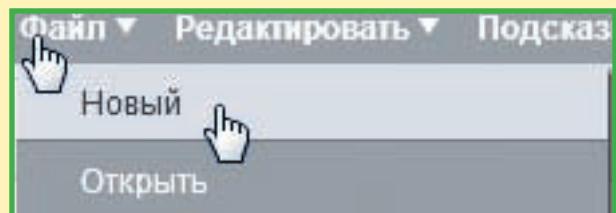
4. Добавьте ещё один автомобиль, у которого будет только четыре скорости.



# ГЛАВА 3. ЗНАКОМСТВО С ЭФФЕКТАМИ

## 3.1. СОЗДАНИЕ НОВОГО ПРОЕКТА

Создайте новый проект: в меню **Файл** выберите команду **Новый**.

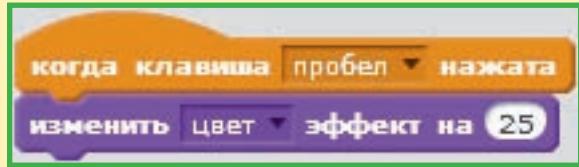


Появится новый проект, в котором, как обычно, есть только Кот. Он-то нам и нужен. Сегодня ему предстоит стать очень эффектным, например таким.

Для изменения внешнего вида Кота мы будем использовать фиолетовые блоки **Внешность**.

## 3.2. ЦВЕТОВОЙ ЭФФЕКТ

Цветовой эффект изменяет цвет спрайта. Создайте вот такой скрипт.





Протестируйте его.

При нажатии клавиши <Пробел> Кот меняет цвет. Выглядит весело, но почему так происходит? Дело в том, что все цвета в Scratch имеют свои числовые значения от 0 до 200. Например, красный цвет имеет значение 0, жёлтый — 25, и так далее (как на картинке).

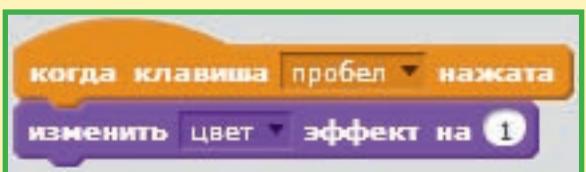


Таким образом, изменяя эффект цвета на 25, Котик пройдёт через испытание палитрой и снова станет рыжим. Это произойдёт ровно через восемь нажатий клавиши <Пробел> ( $25 \cdot 8 = 200$ ). Проверьте это. Измените значение 25 на 1. Нажмайте клавишу <Пробел> — теперь цвет изменяется очень плавно.



### Вопрос

Сколько раз надо нажать клавишу <Пробел>, чтобы окрас Котика вернулся к рыжему цвету?



Вернуть Коту первоначальный вид можно, дважды щёлкнув на фиолетовом блоке убрать графические эффекты.

убрать графические эффекты



### 3.3. ЭФФЕКТ РЫБЬЕГО ГЛАЗА

Рыбий глаз — очень интересный эффект, раздувающий спрайт. Выберите его из выпадающего списка эффектов. Введите значение 25.

Нажмайте клавишу <Пробел> и смотрите, что получится.



Раздуло Котика не слабо, наверное, вчерашнее молоко было позавчерашим...

В отличие от эффекта цвета, этот эффект бесконечный. Если не отпускать клавишу <Пробел>, то Кот превратится в лепёшку. Вот таким он станет через сто нажатий на <Пробел>.



Измените значение 25 на 1. Нажимайте клавишу <Пробел> — теперь Кот раздувается не спеша.

Не забывайте, что вернуть Коту первоначальный вид можно с помощью блока убрать графические эффекты.

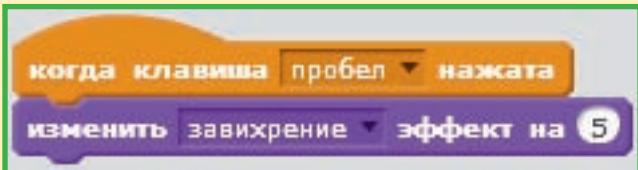
убрать графические эффекты



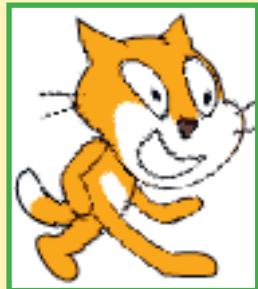


## 3.4. ЭФФЕКТ ЗАВИХРЕНИЯ

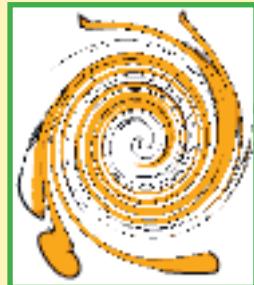
Завихрение — тоже очень зрелищный эффект. Выберите его из выпадающего списка и введите значение 5.



Нажимайте клавишу <Пробел> и посмотрите, как закрутит Котика. Вот таким он станет через 15 нажатий на <Пробел>, то есть значение эффекта равно  $15 \cdot 5 = 75$ .



Эффект завихрения бесконечный, как и рыбий глаз. Вот каким будет Котик при значении эффекта завихрения в 1000.



Теперь вам точно нужен блок убрать графические эффекты.

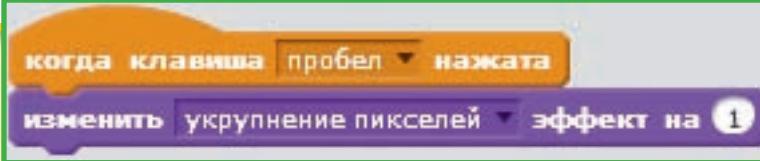
**убрать графические эффекты**

## 3.5. ЭФФЕКТ УКРУПНЕНИЯ ПИКСЕЛОВ

Этот эффект укрупняет точки, из которых состоит изображение — пиксели. Выглядит это вот так.



Соберите вот такой скрипт  
и протестируйте его работу.



Посмотрите, что будет при  
значении эффекта 100 и больше.



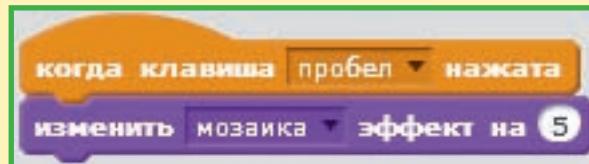
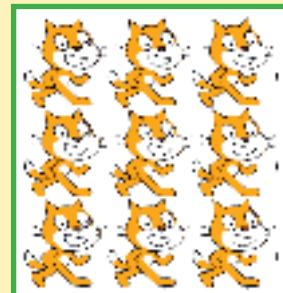
При значении эффекта 350 Кот  
превратится в оранжевый  
кирпич, а при значении  
больше 460 совсем исчезнет.  
Не забывайте о блоке убрать  
графические эффекты!

убрать графические эффекты

## 3.6. ЭФФЕКТ МОЗАИКИ

Этот эффект работает вот так.

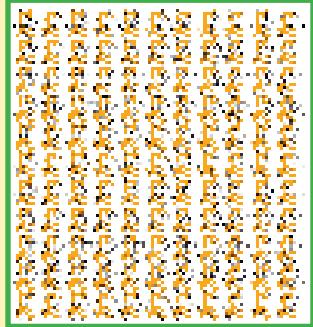
Спрайт превращается в мозаику.  
Соберите вот такой скрипт  
и протестируйте его.





При значении 100 и более спрайт выглядит как армия малюсеньких клонов, но это по-прежнему один спрайт.

Верните Котику первоначальный вид.



**убрать графические эффекты**

## 3.7. ЭФФЕКТ ЯРКОСТИ

Этот эффект увеличивает яркость спрайта. Сделайте вот такой скриптов и посмотрите, как он работает.

Вот как будет выглядеть Котик после десяти нажатий клавиши <Пробел>.



когда клавиша пробел нажата  
изменить яркость на 5

При значении эффекта 100 Кот исчезнет. Таким образом, значение эффекта яркости может изменяться от 0 до 100.

Не забывайте о блоке убрать графические эффекты!

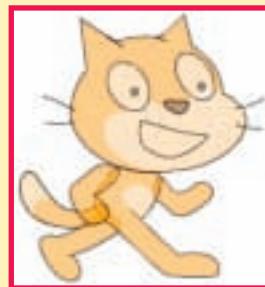
**убрать графические эффекты**

## 3.8. ЭФФЕКТ ПРИЗРАКА

Этот эффект позволяет спрайту исчезать и появляться. Он изменяет прозрачность спрайта. Протестируйте его работу.

когда клавиша пробел нажата  
изменить призрак эффект на 5

Вот так будет выглядеть Котик при значении эффекта призрака равном 50.



При значении эффекта призрака равном 100 спрайт станет невидимым. Верните Котику первоначальный вид.

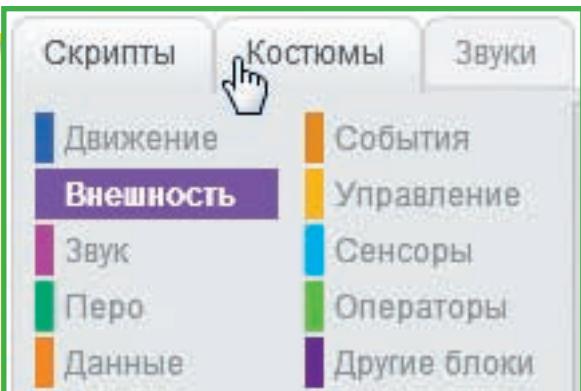
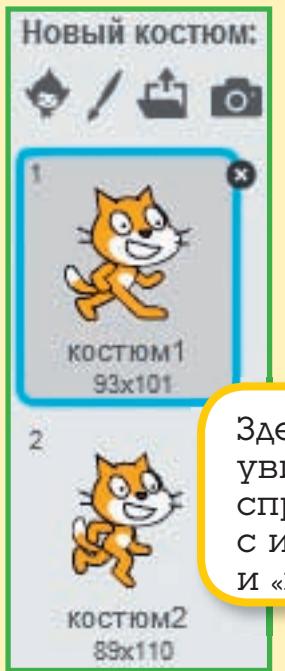
убрать графические эффекты

## 3.9. АНИМАЦИЯ

Для того чтобы при движении Кот не плыл, а шевелил лапами, надо применить блок следующий костюм.

когда клавиша пробел нажата  
идти 10 шагов  
если на краю, оттолкнуться  
стиль вращения влево-вправо  
следующий костюм

Этот блок переключает по очереди все костюмы спрайта.  
Нажимайте клавишу <Пробел>, и Кот зашагает! Это  
происходит из-за того, что у Кота два костюма.  
Перейдите на вкладку **Костюмы**.

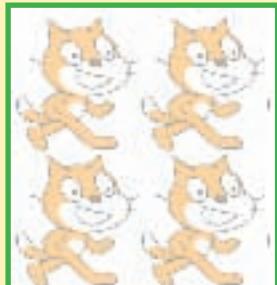


Здесь вы можете  
увидеть, что у  
спрайта два костюма  
с именами «костюм1»  
и «костюм2».

Нажимайте  
клавишу <Пробел>,  
и вы увидите, как  
костюмы по очереди  
переключаются.

## 3.10. ВОПРОСЫ

1. Какие эффекты применены к Коту?



2. А теперь?

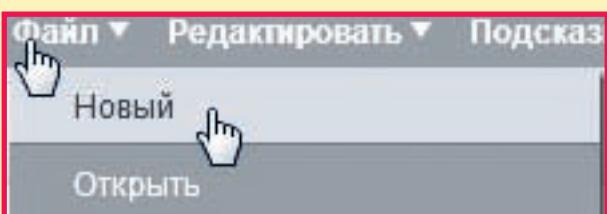




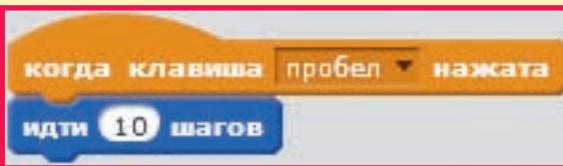
# ГЛАВА 4. ЗНАКОМСТВО С ОТРИЦАТЕЛЬНЫМИ ЧИСЛАМИ

## 4.1. ХОДИМ ЗАДОМ НАПЕРЁД

Создайте новый проект: в меню **Файл** выберите команду **Новый**.



Появится новый проект, в котором, как обычно, есть только Кот. Сделайте простой скрипт гуляющего Кота.

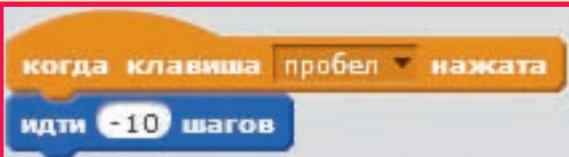


Нажимайте клавишу <Пробел>. Кот, как обычно, гуляет по сцене.

Для того чтобы научить его ходить задом наперёд и делать всё наоборот, нам понадобится отрицательное число  $-10$ . Это то же самое, что и  $10$ , только с минусом. Отрицательные числа очень похожи на обычновенные числа, только они делают всё наоборот, право меняют на лево, верх на низ, перед на зад, разгон на торможение, рост на снижение, взлёт на падение и так далее.

Замените в скрипте  $10$  на  $-10$ .

Протестируйте скрипт, Кот идёт задом наперёд!

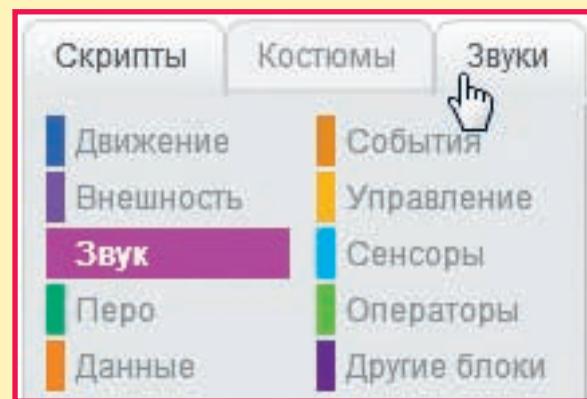


## 4.2. ПЕРЕВОРАЧИВАЕМ ЗВУКИ

А теперь давайте научим его мяукать задом наперёд! Это можно сделать и без отрицательных чисел. Добавьте к скрипту блок мяуканья.

```
когда клавиша пробел нажата
стиль вращения влево-вправо
идти -10 шагов
если на краю, оттолкнуться
играть звук мяу
```

Перейдите на вкладку **Звуки**.



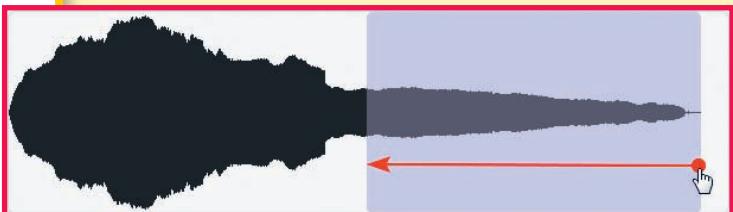
Здесь вы увидите графическое изображение звука. Чем шире полоса, тем звук громче.



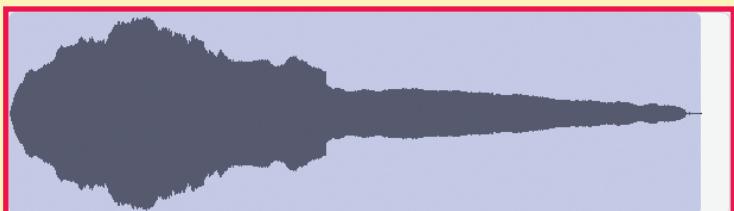
Нажмите на кнопку с треугольником.



Вы услышите звук.  
Для того чтобы проиграть его задом наперёд, надо сначала выделить весь звук. Щёлкните мышью справа от звука и, не отпуская мышь, двигайте её к началу влево...



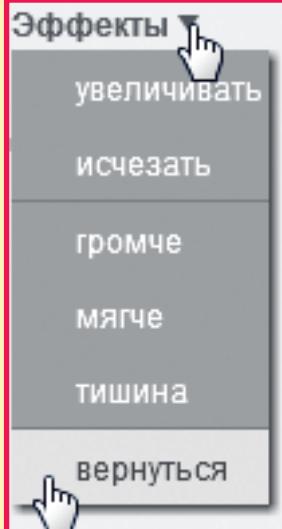
...до тех пор, пока не будет выделен весь звук.

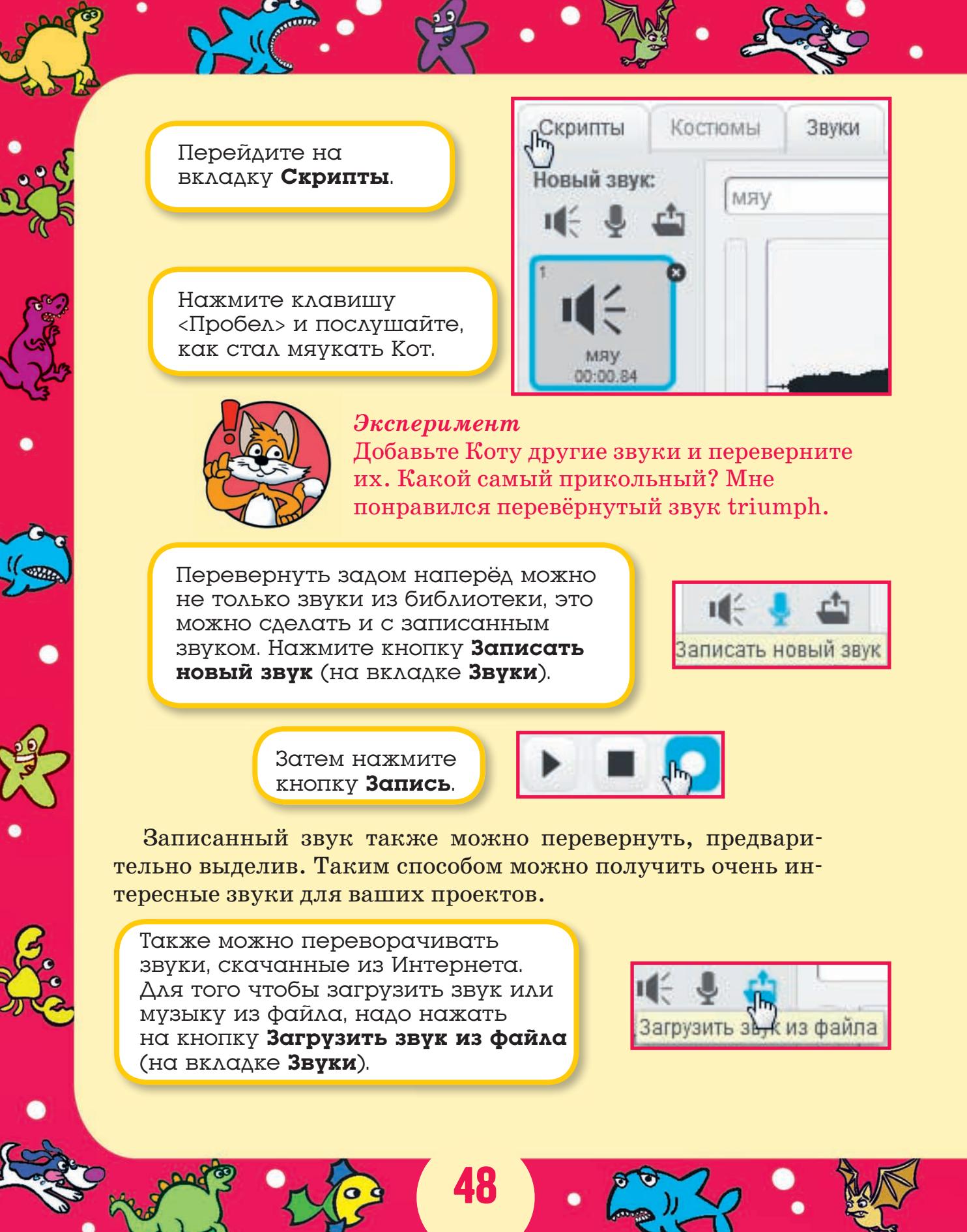


### Совет

Для того чтобы выделить весь звук, можно использовать сочетание клавиш <Ctrl>+<A>.

А теперь перевернём звук задом наперёд! Раскройте выпадающий список **Эффекты** (он находится под изображением звука) и выберите команду **вернуться**.





Перейдите на вкладку **Скрипты**.

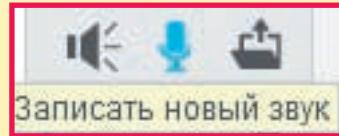
Нажмите клавишу <Пробел> и послушайте, как стал мяукать Кот.



### Эксперимент

Добавьте Коту другие звуки и переверните их. Какой самый прикольный? Мне понравился перевёрнутый звук *triumph*.

Перевернуть задом наперёд можно не только звуки из библиотеки, это можно сделать и с записанным звуком. Нажмите кнопку **Записать новый звук** (на вкладке **Звуки**).



Затем нажмите кнопку **Запись**.



Записанный звук также можно перевернуть, предварительно выделив. Таким способом можно получить очень интересные звуки для ваших проектов.

Также можно переворачивать звуки, скачанные из Интернета. Для того чтобы загрузить звук или музыку из файла, надо нажать на кнопку **Загрузить звук из файла** (на вкладке **Звуки**).





В открывшемся окне необходимо выбрать нужный файл и нажать кнопку **Открыть**. Теперь можно проиграть файл задом наперёд или применить к нему другие звуковые эффекты.



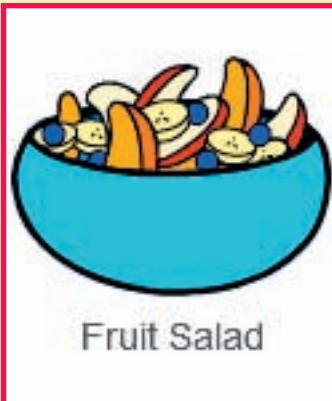
### *Совет*

Загружайте только файлы формата MP3 или WAV! Страйтесь не загружать файлы большого размера. Если в вашем проекте используется только часть музыкального файла, то удалите его неиспользуемую часть, выделив её и нажав клавишу <Delete>.

## 4.3. ПРИВИДЕНИЕ

Используя отрицательные числа, можно сделать много интересных проектов. Например, Привидение, появляющееся из ниоткуда.

Откройте библиотеку спрайтов.



Выберите Привидение и нажмите кнопку **OK**.

Теперь у нас два спрайта.

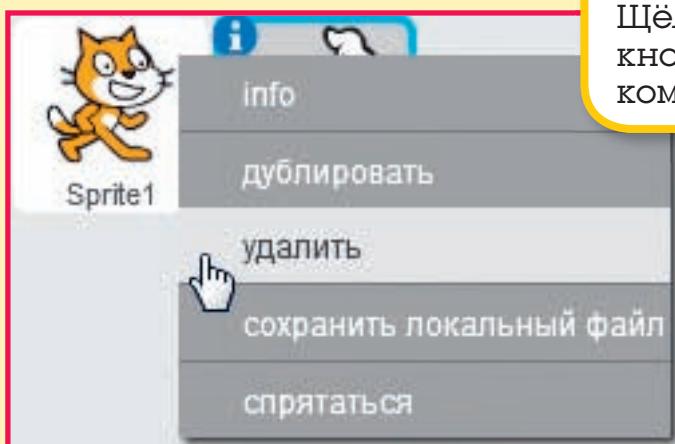


Sprite1

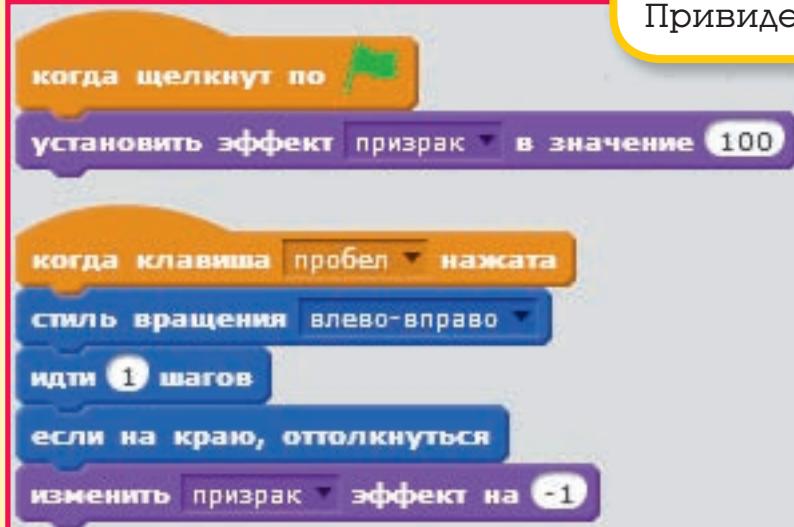


Ghost1

Кот нам не нужен.  
Щёлкните на нём правой  
кнопкой мыши и выберите  
команду **УДАЛИТЬ**.



Сделайте вот такие  
два скрипта для  
Привидения.

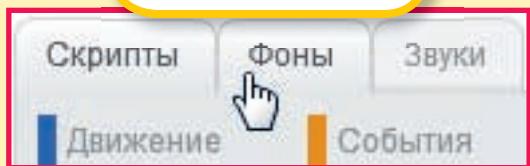




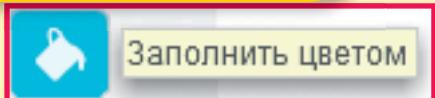
Протестируйте работу скриптов. Щёлкните на зелёном флагжке (он находится в правом верхнем углу сцены) — Призидение исчезнет. Нажимайте клавишу <Пробел>, и оно появится из ниоткуда! Но почему это происходит? Всё очень просто. Когда вы нажимаете на зелёный флагок, эффект призрака становится равным 100, и Призидение исчезает. При нажатии клавиши <Пробел> значение эффекта призрака каждый раз уменьшается на 1 и после ста нажатий становится равным нулю.

Давайте добавим жути и нарисуем на сцене тёмную ночь. Выберите сцену.

Перейдите на вкладку **Фоны**.



Выберите инструмент **Заполнить цветом**.



Щёлкните на фоне — он покернеет.



По умолчанию всегда выбран чёрный цвет.



Выберите инструмент **Кисть**.



Нарисуйте звёзды.



Снова нажмите на **зелёный флагок**, а затем на <Пробел>. Так гораздо страшнее!  
Сохраните проект.

## 4.4. ВОПРОСЫ

1. Что получится, если мы изменим 10 на  $-2$ ?
2. На улице было  $+25$  градусов, а затем температура изменилась на  $-5$  градусов. Какая температура стала на улице?

## 4.5. ЗАДАНИЯ

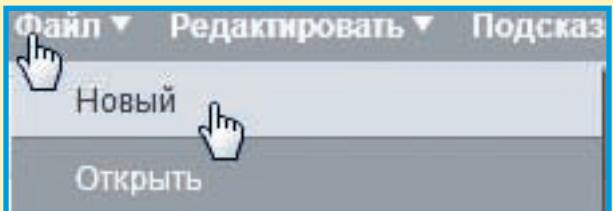
1. С помощью инструмента записи голоса запишите какую-нибудь фразу и переверните её задом наперёд.
2. Добавьте в проект с Привидением жуткую музыку из библиотеки звуков.
3. Добавьте в проект с Привидением ещё одно привидение, которое будет появляться при нажатии на клавишу <1>.



# ГЛАВА 5. ЗНАКОМСТВО С ПЕРОМ

## 5.1. РИСУЕМ КАРАКУЛИ

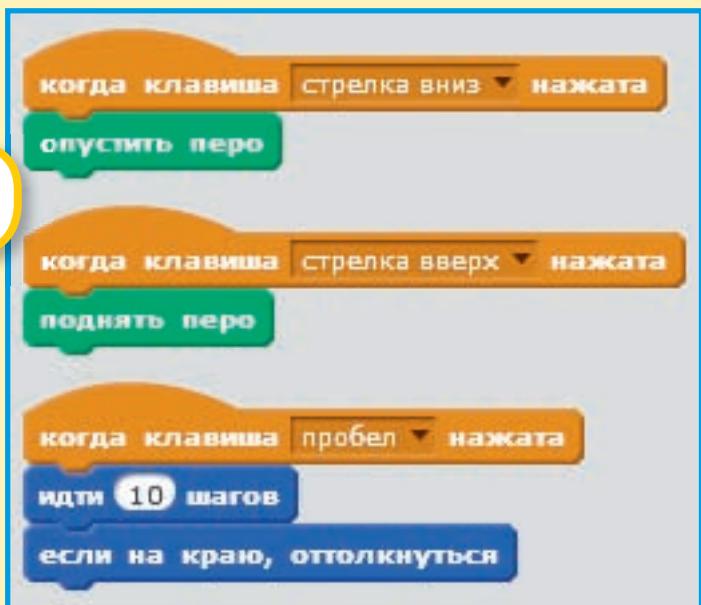
Создайте новый проект: в меню **Файл** выберите команду **Новый**.



Появится новый проект, в котором, как обычно, есть только Кот, и сейчас мы научим его рисовать на сцене. Для этого мы будем использовать зелёные блоки Перо.

Перо — это такой цветной фломастер, которым спрайт может рисовать на сцене. Перо может быть поднято — тогда спрайт при движении ничего не рисует, или опущено — тогда при движении спрайта за ним тянется линия.

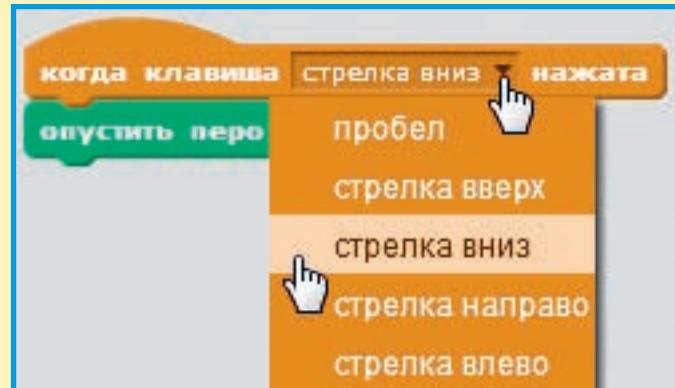
Сделайте вот такую программу для Кота.





## Подсказка

Для выбора нужной клавиши раскройте выпадающий список.



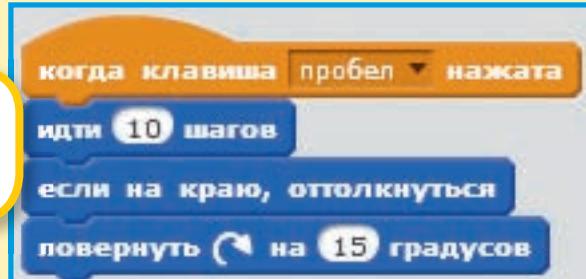
Просто нажмите клавишу <Пробел> — Котик, как обычно, гуляет по сцене. А теперь нажмите клавишу <↓>, а затем снова нажмите <Пробел>. Перо будет опущено, и за Котом потянется тонкая синяя линия.



Если потом нажать клавишу <↑>, то перо будет поднято, и линия появляться не станет.

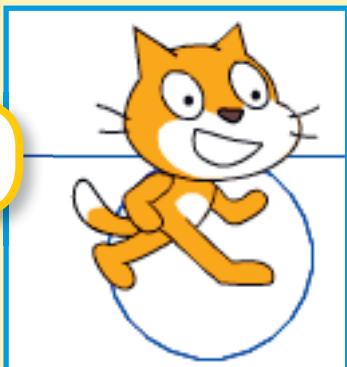


Немного повеселимся. Вам понадобится блок поворота. Добавьте его.





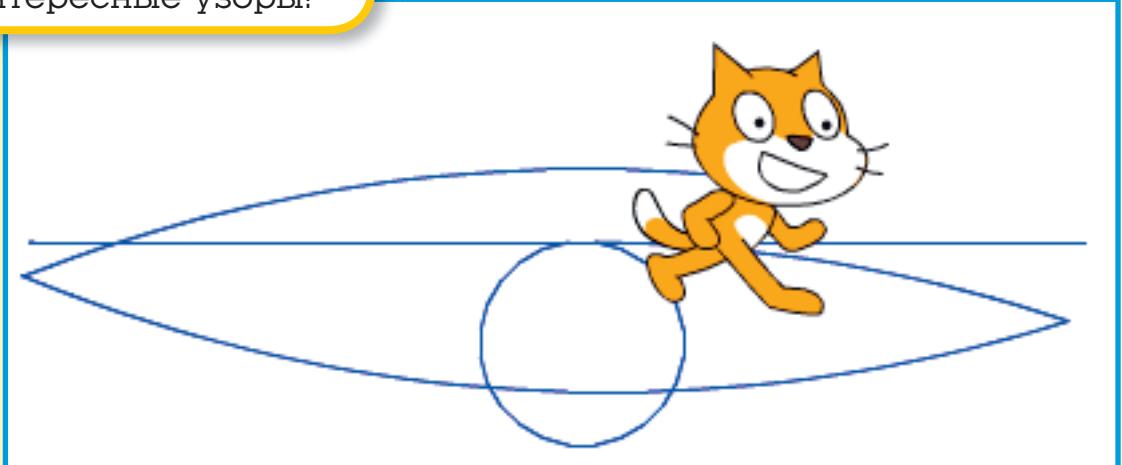
Нажимайте клавишу <Пробел> — Кот нарисует окружность!



Это происходит потому, что каждый раз Кот делает 10 шагов вперёд и немного поворачивается по часовой стрелке. Задайте угол поворота 1 градус.

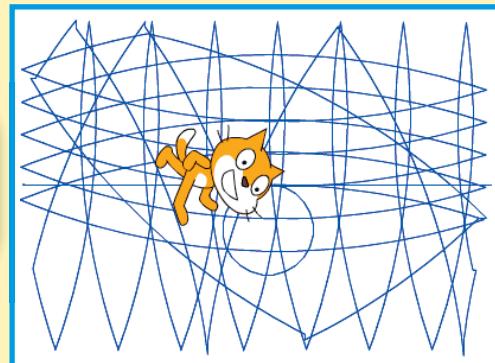
```
когда клавиша пробел нажата  
идти 10 шагов  
если на краю, оттолкнуться  
поворнуть ° на 1 градусов
```

Теперь Кот поворачивает по чуть-чуть, и на сцене появляются очень интересные узоры!





Если нажимать клавишу <Пробел> много раз, то вся сцена будет размалёвана.



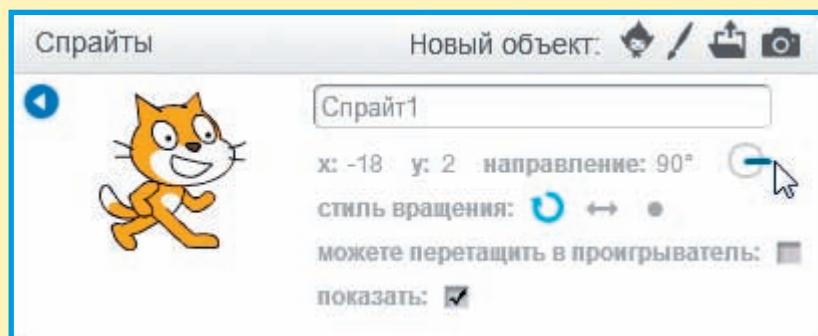
### Совет

Для того чтобы очистить сцену, дважды щёлкните мышью на блоке очистить.



Чтобы повернуть Кота в исходное положение, так сказать поставить его на лапы, можно воспользоваться одним секретным действием. Щёлкните на значке с буквой **i** в левом верхнем углу спрайта.

Откроется область, в которой можно увидеть некоторые свойства спрайта и даже изменить их. Нам нужно Котика повернуть. Перемещайте синюю линию на окружности и посмотрите, как наш персонаж начнёт вертеться-крутиться.



### Эксперимент

Изменяйте значения в блоках **идти** и **поворнуть**.

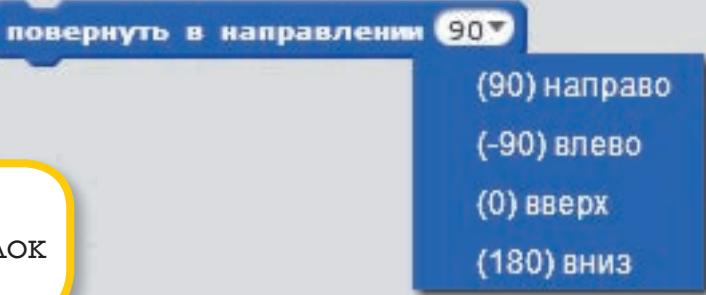
Да уж... Художник из Котика получился не очень...



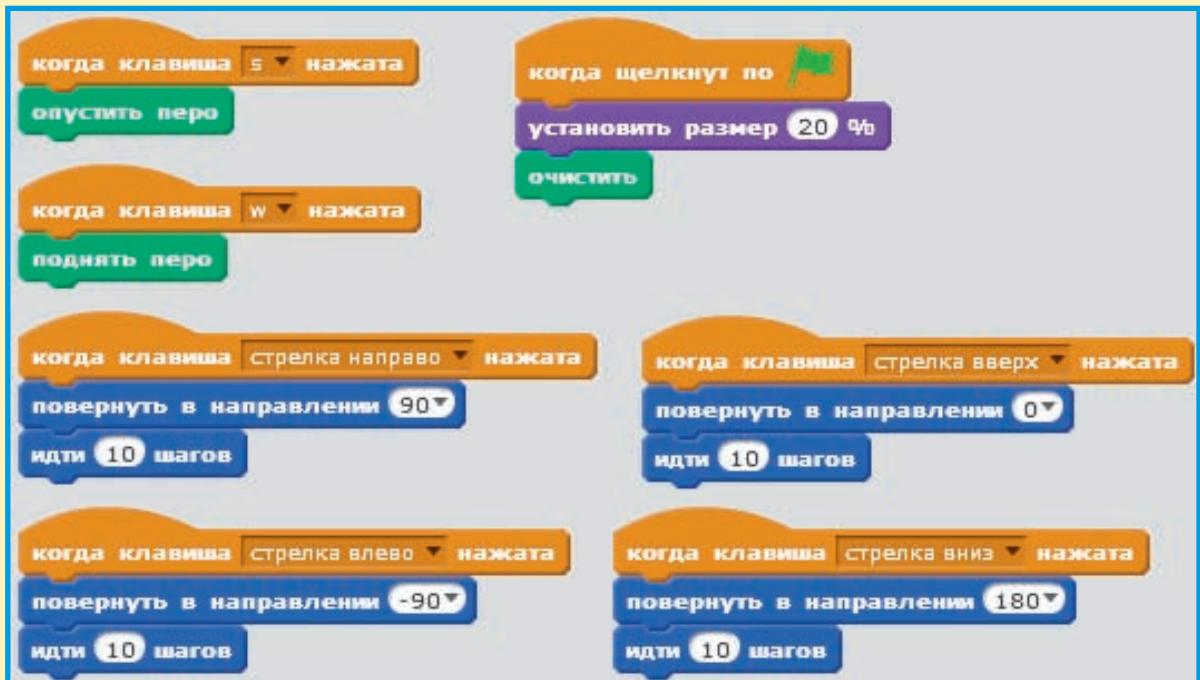
## 5.2. РИСУЕМ КРАСИВО

Давайте будем управлять движением Кота с помощью стрелок на клавиатуре. Может, тогда получится нарисовать что-нибудь красивое.

Создайте новый проект.  
Здесь вам пригодится блок  
поворнуть в направлении.



Этот блок поворачивает спрайт в указанном направлении. Сконструируйте для Кота вот такую программу. Она состоит из семи скриптов.





При нажатии на зелёный флагок Котик уменьшится в размере и очистит сцену.



### Подсказка

Нормальный размер спрайта равен 100%. Размер 20% ровно в 5 раз меньше нормального, а размер 50% ровно в 2 раза меньше нормального размера.

При нажатии клавиши <s> перо опустится, а при нажатии <w> перо поднимется. При нажатии клавиш-стрелок Котик будет поворачиваться в соответствующем направлении и делать по 10 шагов.



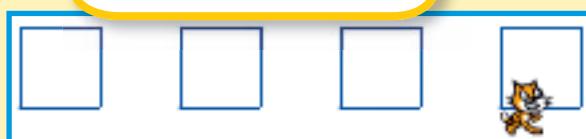
### Совет

Если перо не опускается и не поднимается, проверьте, что вы переключились на английскую раскладку клавиатуры.

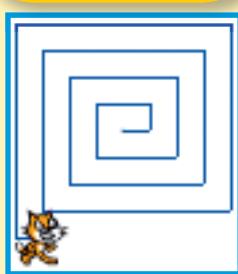
Теперь можно нарисовать всё, что состоит из прямоугольников. Попробуйте нарисовать три квадрата.



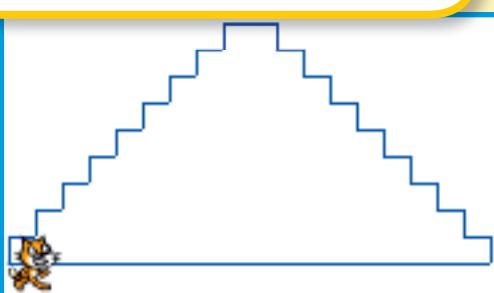
Ряд из квадратов.



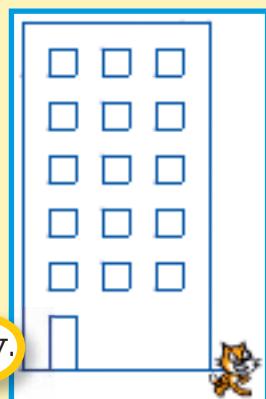
Сpirаль.



Пирамиду индейцев майя.



Многоэтажку.





Многоэтажку нарисовать не просто. У меня не получилось с первого раза. Любая ошибка, и приходится начинать сначала.

Давайте добавим в проект возможность стереть нарисованное. А что значит стереть нарисованное на белом фоне? Это значит закрасить белым! Нужно добавить в проект возможность рисовать белым.

Добавьте в проект два скрипта.



### Подсказка

Для выбора цвета в блоке установки цвета необходимо щёлкнуть в цветной квадратик (при этом курсор мыши изменит свою форму), а затем в любое место на экране, окрашенное в нужный цвет. Например, для выбора синего цвета можно щёлкнуть по синему блоку, а для выбора белого цвета — по сцене.



Теперь при нажатии на клавишу с цифрой 0 Котик будет стирать нарисованное, а при нажатии на единицу снова сможет рисовать.

Сохраните проект.



## 5.3. ВОПРОСЫ

1. Кот рисует короткими отрезками. Как увеличить длину отрезков?
2. Как изменить программу, чтобы она стирала нарисованное на сером фоне?
3. Что будет с Котом, если установить его размер не 20%, а 200%?

## 5.4. ЗАДАНИЯ

1. Добавьте в проект возможность рисовать красным цветом.
2. Установите размер кота в 10 раз меньше нормального.
3. Сделайте так, чтобы можно было очищать сцену, используя клавиатуру, не нажимая на зелёный флагок.
4. Для того чтобы было удобнее рисовать, сделайте так, чтобы цвет Кота был разным при опущенном пере и при поднятом пере.



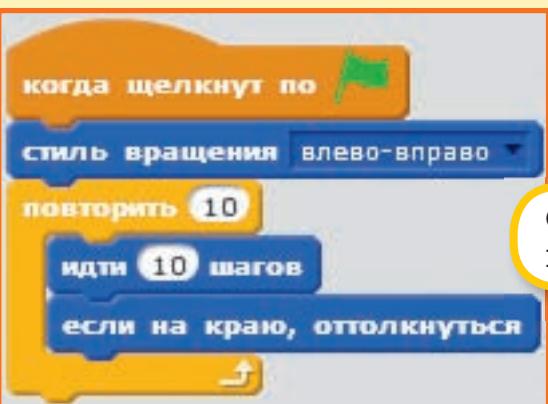
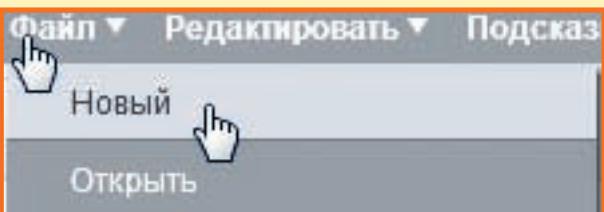


# ГЛАВА 6. ЦИКЛЫ

## 6.1. ЗНАКОМСТВО С ЦИКЛАМИ

В предыдущих главах для управления спрайтами вам приходилось постоянно нажимать клавиши. Это не всегда удобно. Для автоматического выполнения действий в Scratch есть очень важный светло-коричневый блок Повторить. Он позволяет повторить одно действие несколько раз подряд. Давайте попробуем.

Создайте новый проект: в меню **Файл** выберите команду **Новый**.



Сконструируйте для Кота вот такой скрипт.

Блок Повторить выполнит своё содержимое 10 раз.

Щёлкните по зелёному флагжку. Кот смело пошагает вперёд и сделает 10 раз по 10 шагов — всего 100 шагов. Снова щёлкните по флагжку — Кот опять сделает 100 шагов.



### Эксперимент

Вводите разные значения в блок Иди и в блок Повторить и посмотрите, что получится. Не вводите в блок Повторить значения больше 1000.



## 6.2. ЦИКЛЫ И ЭФФЕКТ ЦВЕТА

Вы уже знаете, что в Scratch существует 200 цветов. Давайте сделаем Кота, который будет автоматически изменять свой цвет 200 раз и снова становиться рыжим. Удалите созданный скрипт: для этого перетащите его целиком в палитру блоков и отпустите кнопку мыши. Скрипт исчезнет.

The screenshot shows the Scratch script palette with a script for a cat. The script starts with a green flag button event, followed by a 'repeat' loop. Inside the loop, the cat moves 10 steps, turns clockwise 10 degrees, and checks if it's at the edge. If yes, it bounces back. A red arrow points from the text 'когда щелкнут по флагу' to the green flag button icon. A yellow callout box contains the text: 'Теперь создайте для Кота вот такой скрипт.'

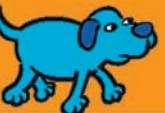
```
когда щелкнут по флагу
  повторить (200)
    иди 10 шагов
    повернуть (10) градусов
    если на краю, оттолкнуться
  конец
показать [кошка v]
поменять цвет на 1
```

Нажмите на **зелёный флагок**. Кот будет плавно изменять свой цвет, пока снова не станет рыжим.

The screenshot shows the Scratch script palette with a script for a cat. It starts with a green flag button event, followed by a 'repeat' loop of 200 iterations. Inside the loop, the cat changes its color and effect. A yellow callout box contains the text: 'Нажмите на зелёный флагок. Кот будет плавно изменять свой цвет, пока снова не станет рыжим.'

```
когда щелкнут по флагу
  повторить (200)
    изменить цвет на 1
    изменить эффект на 1
  конец
```



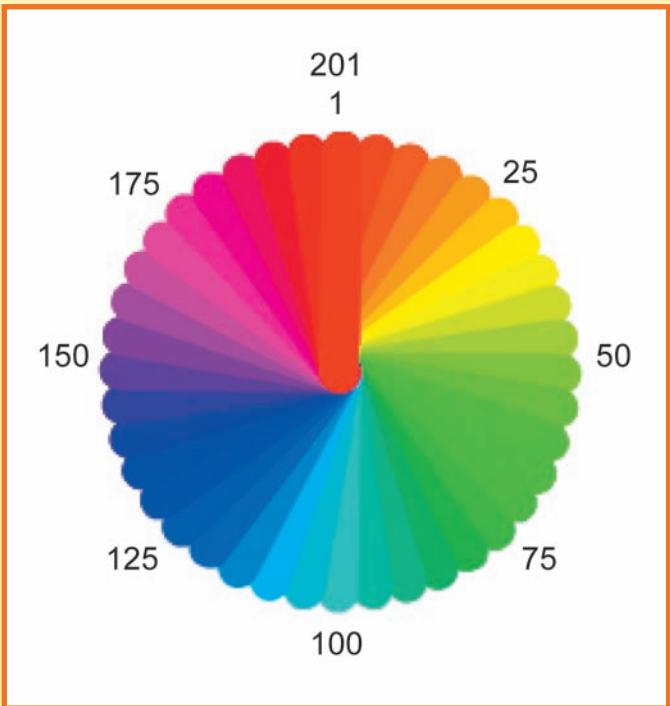


## Эксперимент

Измените значение в блоке повторить на 400.

А сколько раз теперь Кот пройдёт по цветовому кругу? Измените значение в блоке изменить цвет эффект на 2. Что изменится в работе скрипта?

Эффект цвета — это циклический эффект, цвета повторяются через 200. Цвет 1 — это то же самое, что и 201, что и 401 — это красный цвет, а цвет 125 — это то же самое, что и 325 и 525 — это синий.



## Вопрос

Какой цвет имеет номер 314159260150?

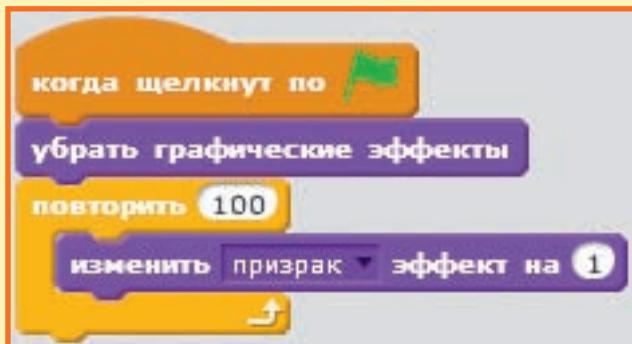
А 27182818280100?

Проверьте свой ответ.



## 6.3. ЦИКЛЫ И ЭФФЕКТ ПРИЗРАКА

А теперь давайте поиграем с эффектом призрака. Этот эффект изменяется от 0 до 100.



Создайте для Кота вот такой скрипт.

При нажатии на **зелёный флагок** сначала пропадут все графические эффекты, а затем Кот плавно исчезнет.

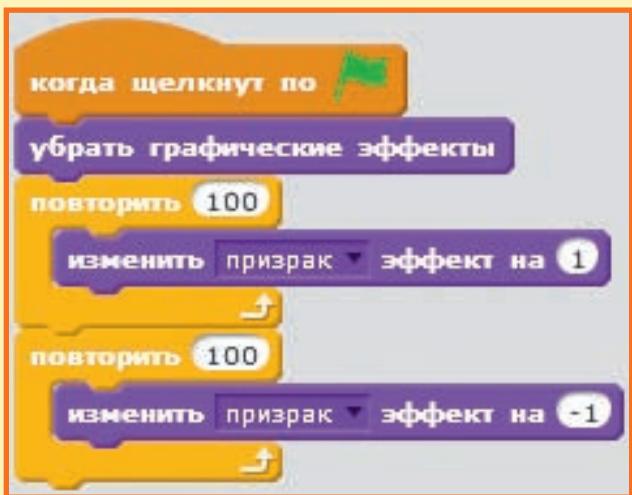


### Эксперимент

Вводить значение больше 100 в блок повторить не имеет смысла, ведь Кот не может стать дважды невидим!

А вот значение в блоке изменить призрак эффект поменять можно. На что это повлияет?

Очень жаль, что Котик постоянно пропадает. Давайте сделаем так, чтобы он плавно исчезал, а потом так же плавно появлялся.



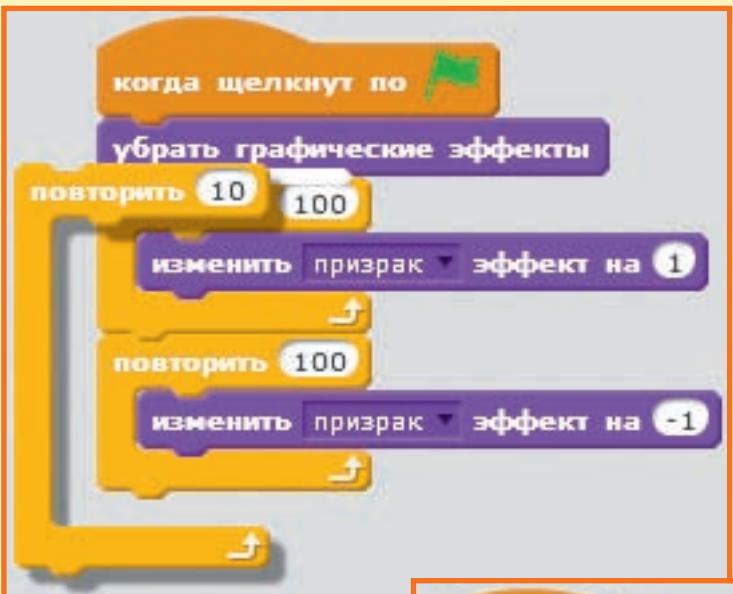
Для этого нам понадобится отрицательное число  $-1$ . Это число имеет перед собой знак «минус», поэтому, изменения эффект призрака на  $-1$ , мы будем уменьшать его значение от 100 до 0, пока Котик не станет таким как был. Доработайте скрипт.





Проверьте, как работает этот скрипт. Нажмите на зелёный флагожок. Кот уйдёт в страну призраков, а потом вернётся из неё!

А что, если нам надо запустить Кота в страну призраков 10 раз подряд? Для этого может пригодиться ещё один блок повтора! Выберите его в палитре блоков и подтащите к скрипту так, чтобы он обнял сразу два блока повтора.

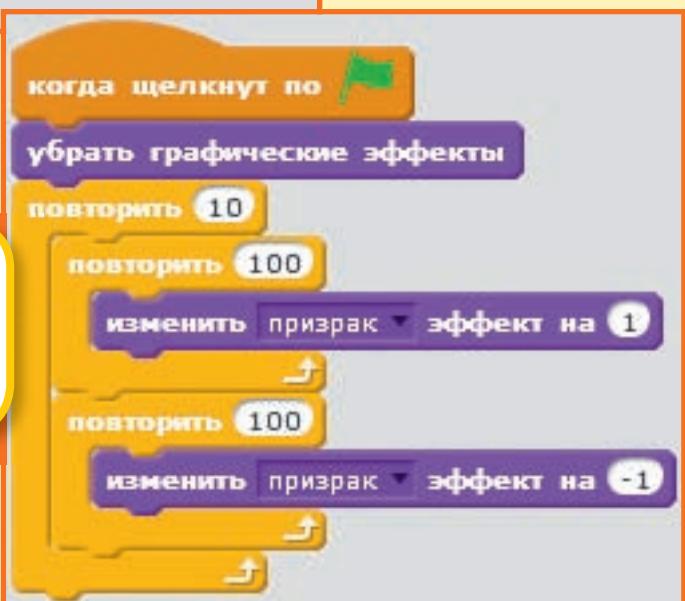


```
когда щелкнут по зелёному флагу
убрать графические эффекты
повторить (10)
  100
    изменить [призрак] [эффект на 1]
    изменить [призрак] [эффект на -1]
повторить (100)
  изменить [призрак] [эффект на 1]
  изменить [призрак] [эффект на -1]
```

Получится вот такой скрипт.

Запустите его. Кот станет невидимым 10 раз подряд.

Сохраните проект.



```
когда щелкнут по зелёному флагу
убрать графические эффекты
повторить (10)
  повторить (100)
    изменить [призрак] [эффект на 1]
    изменить [призрак] [эффект на -1]
повторить (100)
  изменить [призрак] [эффект на 1]
  изменить [призрак] [эффект на -1]
```



## 6.4. ВРАЩЕНИЕ

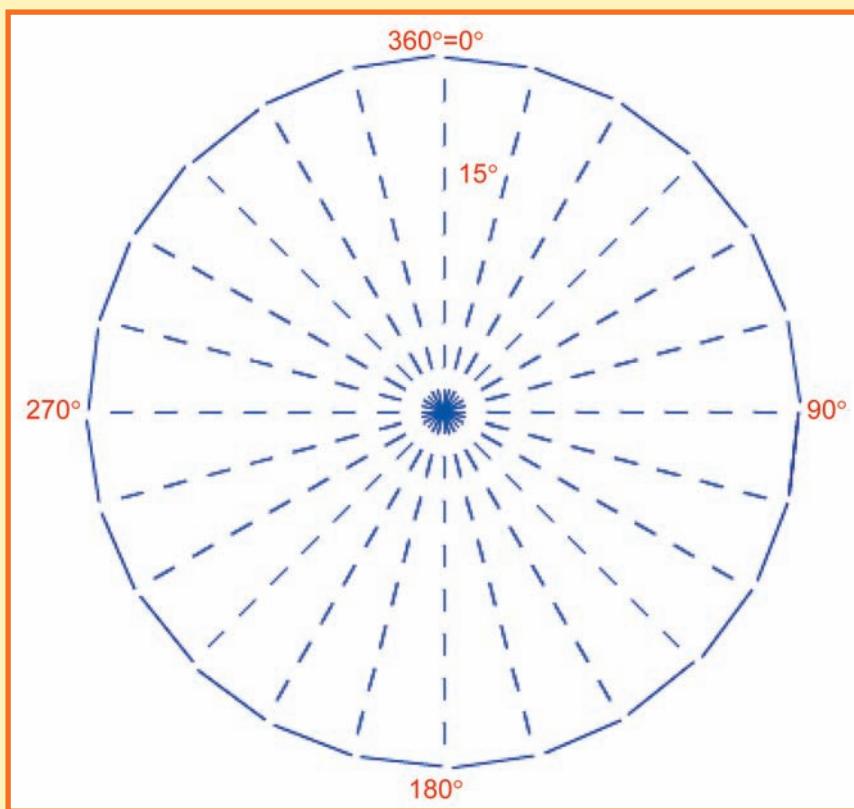
А теперь давайте познакомимся с блоками поворота. Их всего два: поворот по часовой стрелке и поворот против часовой стрелки.

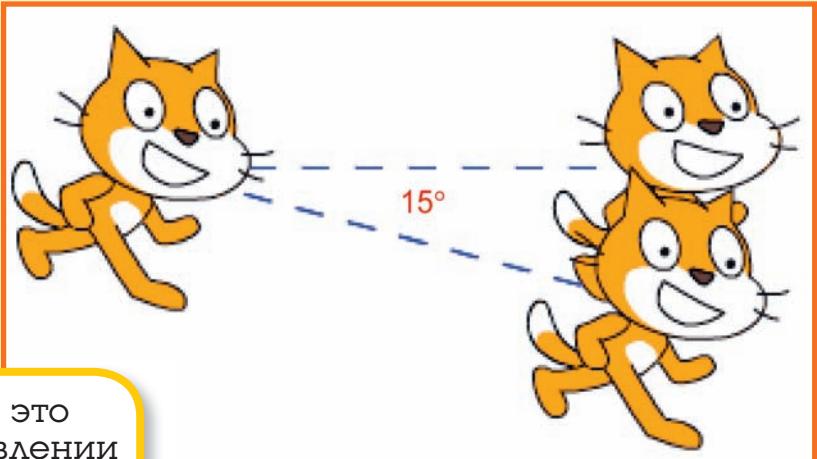
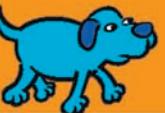
поворнуть на 15 градусов

поворнуть на 15 градусов

Как вы, наверное, уже знаете, окружность делится на 360 градусов, поэтому спрайт в Scratch может двигаться в одном из 360 направлений.

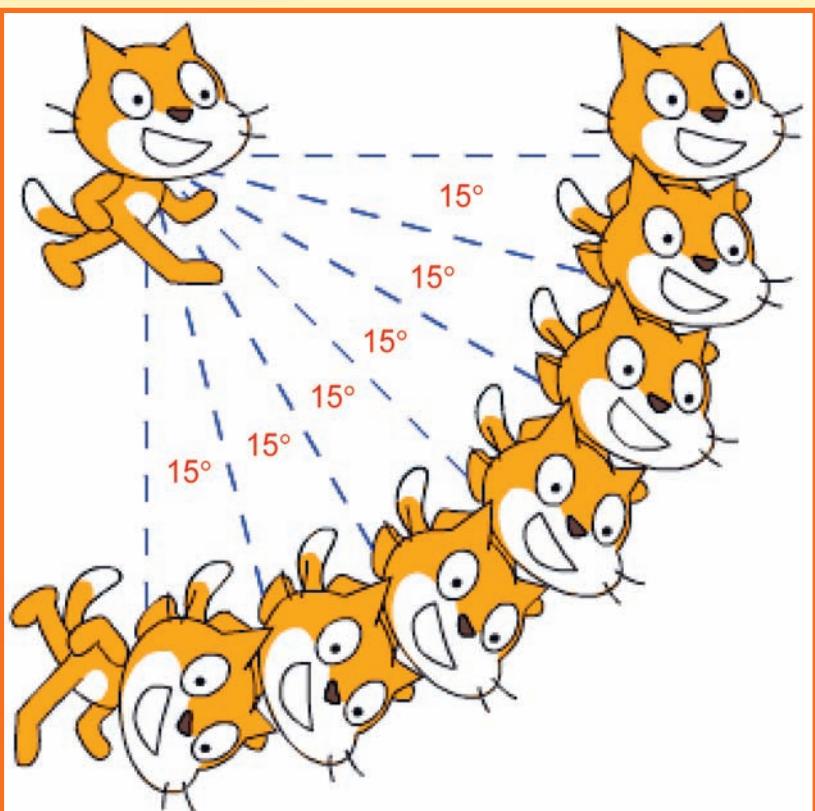
Градусы обозначаются маленьким кружочком, например,  $360^\circ$ .

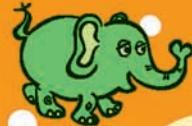




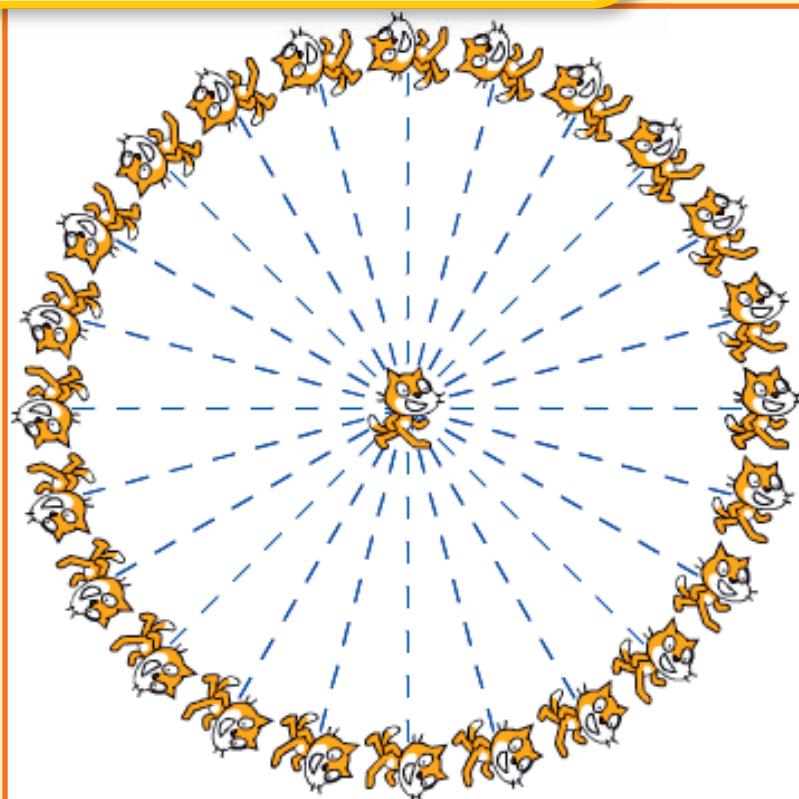
Движение вверх — это движение в направлении  $0^\circ$ , движение вправо — движение в направлении  $90^\circ$  и так далее. Поворот на  $15^\circ$  — это совсем немногого.

Для того чтобы повернуть персонаж под углом в  $90^\circ$ , надо 6 раз повернуть на  $15^\circ$ .





А для того чтобы совершить полный оборот, надо повернуться 24 раза по  $15^\circ$ .



### Совет

В блоке повернуть в направлении 90 спрятана подсказка по направлениям. Если вы забудете, где право и где лево, то всегда сможете подсмотреть правильный ответ.

Обратите внимание, что направления  $-90^\circ$  и  $270^\circ$  — это одно и то же.

поворнуть в направлении 90▼

- (90) направо
- (-90) влево
- (0) вверх
- (180) вниз





Для того чтобы лучше понять, что такое градусы, создайте новый проект и сделайте такой скрипт.

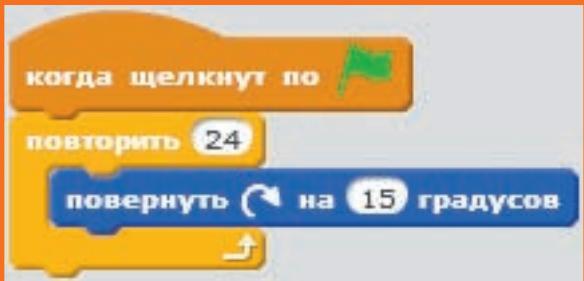
Вы уже знаете, что 24 раза по  $15^\circ$  — это полный оборот.

Нажмите на **зелёный флагок** и убедитесь в этом. Кот сделает ровно один оборот!



### Эксперимент

Изменяйте значения в блоках **повторить** и **поворнуть** так, чтобы произведение этих чисел было равно 360. Например, 36 и 10, 72 и 5, 10 и 36, 5 и 72, 2 и 180 и так далее. Обратите внимание на скорость вращения Кота.

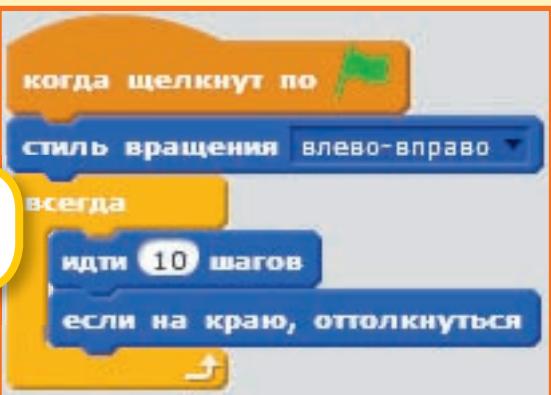


## 6.5. БЕСКОНЕЧНЫЙ ЦИКЛ

Интересно, а что будет, если ввести в блок **повторить** огромное значение, например, один миллион — 1 000 000? В этом случае прогулка Кота затянется на 8 часов, а если ввести один миллиард — 1 000 000 000, то Котик будет гулять целый год!

Однако для бесконечного выполнения скрипта лучше использовать блок **всегда**. Этот блок всегда повторяет своё содержимое до бесконечности.

Создайте для Кота вот такой скрипт.





Щёлкните по зелёному флагжку. Теперь Кота ничто не остановит! Можете съездить на лето к бабушке, а вернувшись, убедиться, что Кот всё ещё ходит по сцене кругом. Идёт направо — песнь заводит, налево — сказку говорит.

Шутка. Кота-сказочника вы сможете запрограммировать после того, как познакомитесь с условным блоком.

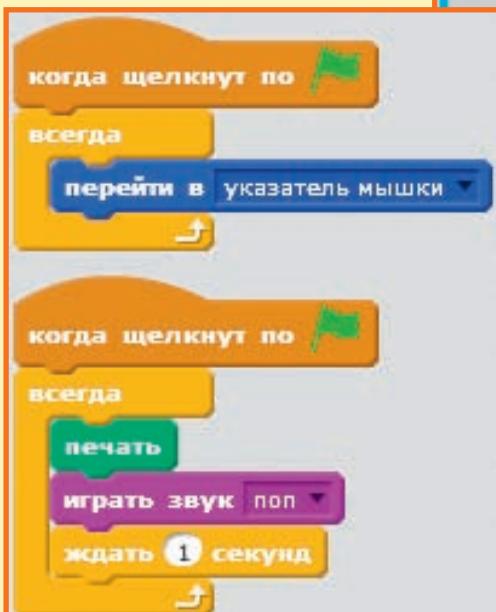
## 6.6. АВТОМАТИЧЕСКАЯ ПЕЧАТЬ

Давайте сделаем небольшой проект с применением блока всегда. Украсим двор яблоками!

Создайте новый проект. Добавьте из библиотеки спрайтов Яблоко.



Выберите Яблоко и нажмите кнопку **OK**.



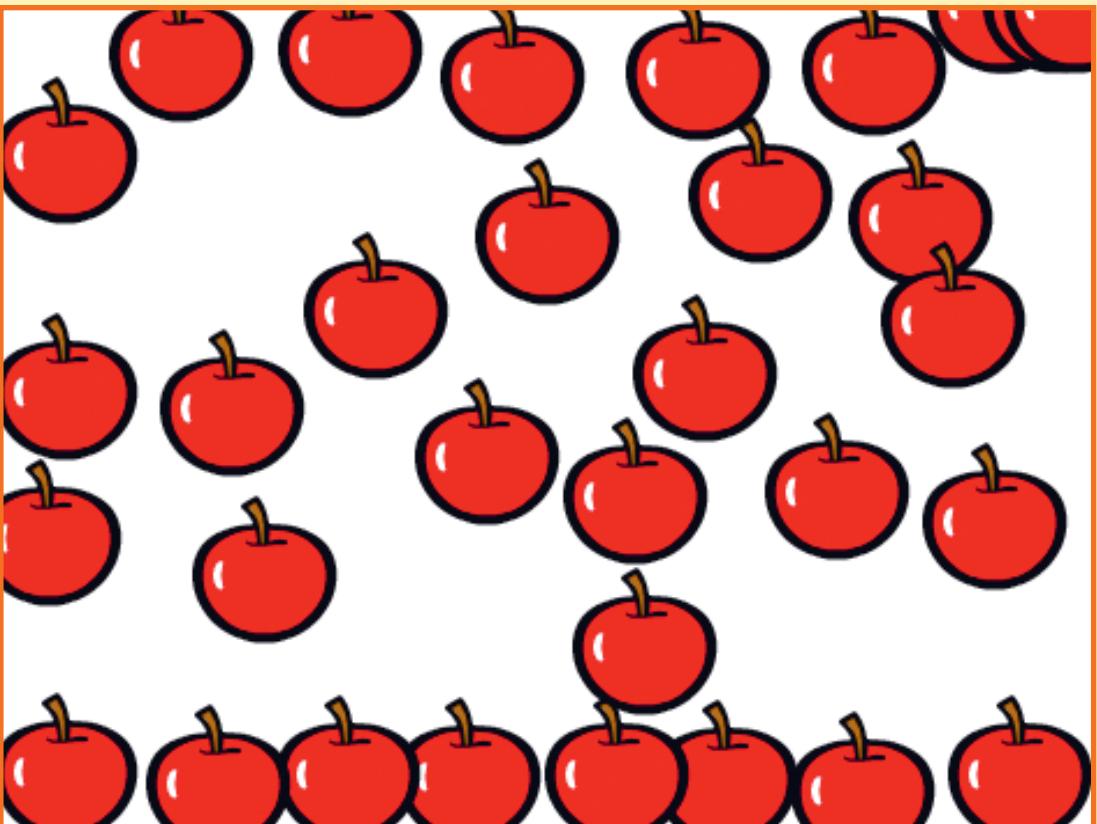
Кота удалите одним из знакомых вам способов.  
Сделайте вот такую программу из двух скриптов для яблока.





В этой программе используются сразу три новых блока: перейти в указатель мышки, печать и ждать. Первый скрипт делает так, чтобы Яблоко всегда перемещалось вслед за указателем мыши, а второй всегда отпечатывает на сцене изображение яблока, проигрывает звук и ждёт 1 секунду, чтобы яблоки не появлялись слишком быстро.

Протестируйте  
работу программы.



*Совет*

Для того чтобы убрать яблоки  
со сцены, дважды щёлкните  
по зелёному блоку очистить.

очистить





Разбрасывать яблоки на белом фоне не очень интересно. Добавьте на сцену какой-нибудь фон. Например вот этот.

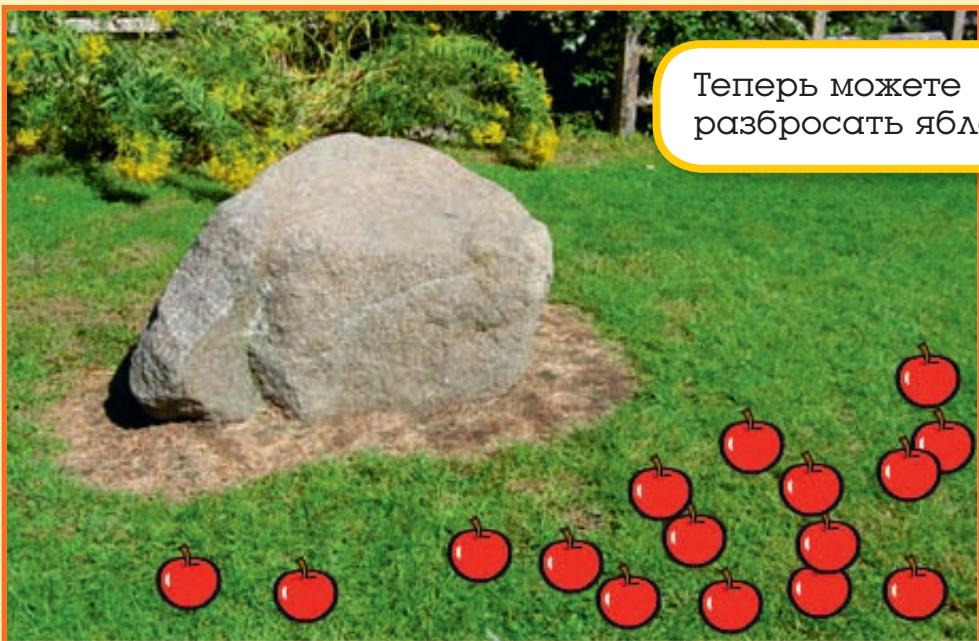
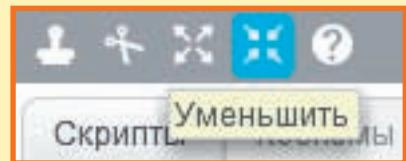


garden rock



gingerbread

Уменьшите размер яблока с помощью инструмента **Уменьшить**: щёлкните сначала по инструменту, а потом по Яблоку на сцене.



Теперь можете разбросать яблоки.

Если вы будете аккуратны, то сможете нарисовать красивую рамку или узор.

Проект готов. Сохраните его.





## 6.7. ЗАДАНИЯ

1. Измените фон сцены в проекте про яблочки.
2. Сделайте так, чтобы отпечатываемое яблоко изменяло цвет.
3. Сделайте так, чтобы отпечатываемое яблоко изменяло прозрачность.
4. Поэкспериментируйте с временем задержки. Вводите числа с точкой: 0.1, 0.2, 0.5, 0.01, 0.02, 0.05 и проанализируйте результат. Введите ноль. Введите отрицательное число, но будьте осторожны, не улетите в прошлое!
5. Измените звук.
6. Сделайте так, чтобы отпечаталось только 15 яблок.
7. Измените костюм Яблока на любой другой из библиотеки спрайтов.
8. Добавьте Яблоку несколько костюмов, которые будут отпечатываться по очереди.
9. Нарисуйте толпу людей.

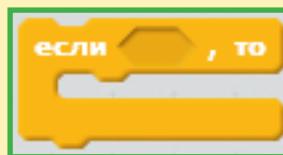




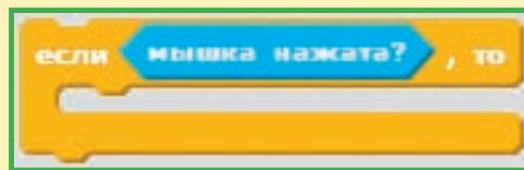
# ГЛАВА 7. УСЛОВНЫЙ БЛОК

## 7.1. ЗНАКОМСТВО С УСЛОВНЫМ БЛОКОМ

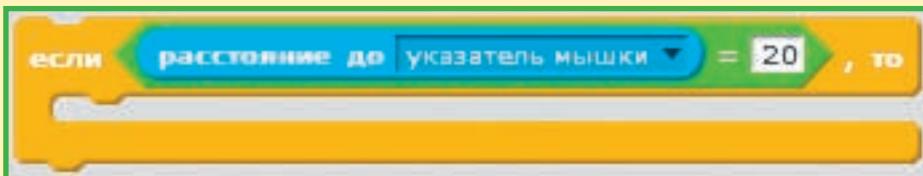
В играх и сложных проектах постоянно происходит множество событий. Например, герой перешёл на следующий уровень, герой нашёл аптечку, враг коснулся героя. При наступлении любого события программа должна выполнить определённые действия, и в этом ей помогает блок проверки условия если.



У этого блока внутри есть специальное шестиугольное углубление, в которое можно вставить шестиугольные зелёные операторы или голубые сенсорные блоки. Тогда блок если будет выглядеть вот так.



Или так.



Давайте же скорее применим этот блок!





## 7.2. ИГРА «ПОГОНЯ»

Жил да был Кот. Как-то раз, гуляя по городу, он повстречал Синего Пса. Пёс был очень зол и погнался за Котом, желая задать ему хорошую трёпку. Спасётся ли Кот, будет зависеть только от вас!

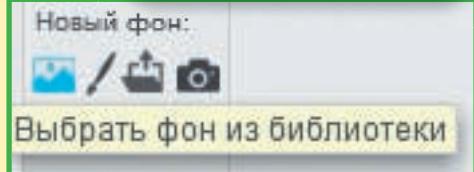


Создайте новый проект.  
Откройте библиотеку спрайтов.



Выберите Синего Пса  
и нажмите кнопку **OK**.

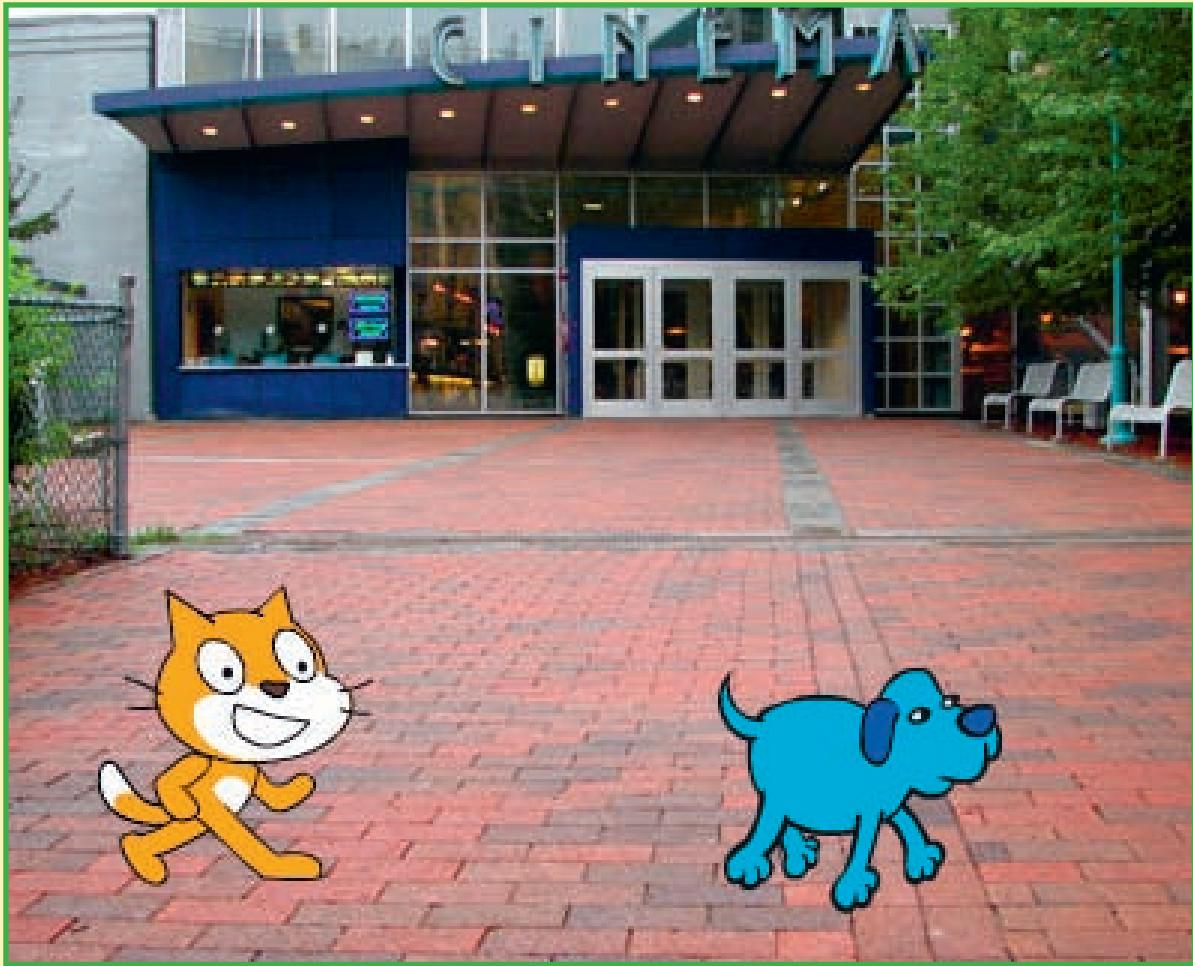
Теперь добавьте  
фон на сцену.



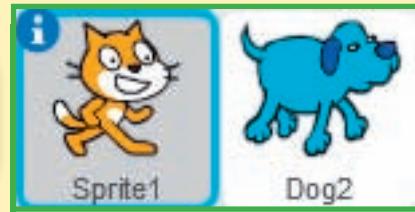
Выберите город и  
нажмите кнопку **OK**.



Теперь игра выглядит вот так.



В проекте всего два спрайта с именами Sprite1 и Dog2.





Начинаем программировать! Сначала создадим программу Кота. Она очень простая — состоит из двух скриптов.

Первый скрипт такой же, как и у Яблока из прошлого проекта. При нажатии на **зелёный флаг** Кот всегда будет следовать за указателем мыши. Второй скрипт использует новый блок **если**. Всегда, если Кот коснётся Dog2 (Пса), то он жалобно мяукнет, потому что касание вряд ли было приятным.

При нажатии на **зелёный флаг** Синий Пёс всегда будет поворачиваться к Коту и идти по 5 шагов. Если Кот не убежит, то очень скоро попадётся в лапы Пса. Второй скрипт очень похож на скрипт Кота. Всегда, если Синий Пёс коснётся Sprite1 (Кота), то он залаёт. Протестируйте игру. Убежать от Пса почти невозможно.

```

when green flag clicked
repeat
    point towards mouse
end

when green flag clicked
repeat
    if [touching [Dog2 v] then
        play sound [meow v] for [1] sec
    end
end

```

Программа Синего Пса вот такая.

```

when green flag clicked
repeat
    turn towards [Sprite1 v]
    move [5] steps
end

when green flag clicked
repeat
    if [touching [Sprite1 v] then
        play sound [dog1 v] for [1] sec
    end
end

```



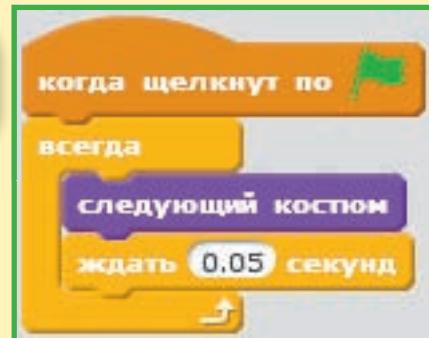
## 7.3. ДОРАБОТКА ИГРЫ

Кажется, Кот и Синий Пёс не бегают по сцене, а плавают. Давайте добавим анимацию!

Добавьте Коту вот такой скрипт.

Теперь Кот всегда будет изменять свой костюм через каждые пять сотых секунды.

Для Пса сделайте такой же скрипт. Для упрощения задачи можно скопировать скрипт Кота Синему Псу. Захватите мышью готовый скрипт, тащите прямо на спрайт Dog2 и там отпустите.



Отпускать скрипт надо тогда, когда курсор будет точно находиться над Псом. Готово! Это гораздо проще, чем собирать скрипт из блоков. Протестируйте работу программы.



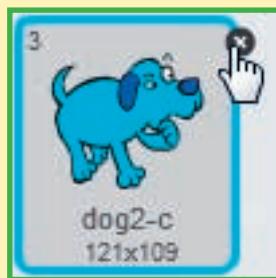


Что происходит?! Пёс делает один шаг и на секунду застывает в задумчивой позе! Но он же не философ! Эту ошибку (баг) надо исправить.

Перейдите на вкладку **Костюмы**. У Пса, в отличие от Кота, не два, а три костюма!



Последний костюм для погони не предназначен. Удалите его, нажав на крестик.



Теперь у Синего Пса только два беговых костюма. Программа будет работать правильно.

Сохраните её.

## 7.4. ЗАДАНИЯ

1. Усложните игру. Ускорьте бег Синего Пса.
2. Верните Псу задумчивый костюм и сделайте так, чтобы в начале игры он стоял в задумчивости 3 секунды и только потом отправлялся в погоню за Котом.

3. Сделайте так, чтобы после касания Кота Пёс, задумавшись, ждал 1 секунду.

4. Добавьте в проект Льва и сделайте так, чтобы при касании Льва Кот мяукал, а Пёс лаял. Лев при касании должен рычать. Запишите рычание с микрофона.

5. Сделайте проект про Кота, который будет управляться клавишами-стрелками вправо и влево, и при движении направо будет заводить песнь с помощью блока играть звук, а при движении налево будет говорить сказку с помощью блока говорить.

Жил старик со  
свою старухой у  
самого синего  
моря, и она  
испекла колобок.





# ГЛАВА 8. МУЛЬТИФИЛЬМ «АКУЛА И РЫБКА»

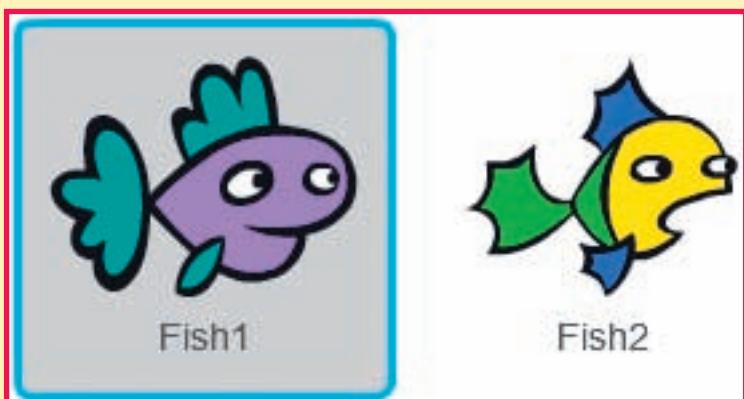
В синем море жили Рыбка и Акула, и как-то раз они по-встречались. А что произойдёт дальше, будет зависеть от того, как вы их запрограммируете.

## 8.1. СОЗДАЁМ ПЕРСОНАЖЕЙ

Запустите Scratch.  
Добавьте два новых спрайта, нажав кнопку  
**Выбрать спрайт из библиотеки**.



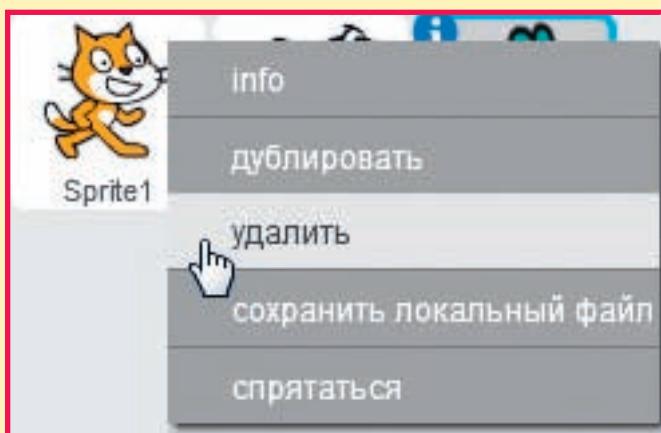
Выберите Акулу (Shark) и нажмите кнопку **OK**.



Затем выберите Рыбку (Fish1) и нажмите кнопку **OK**.



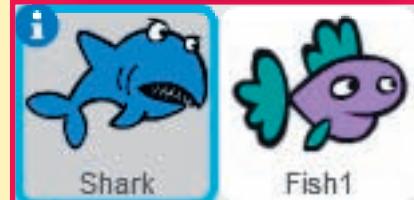
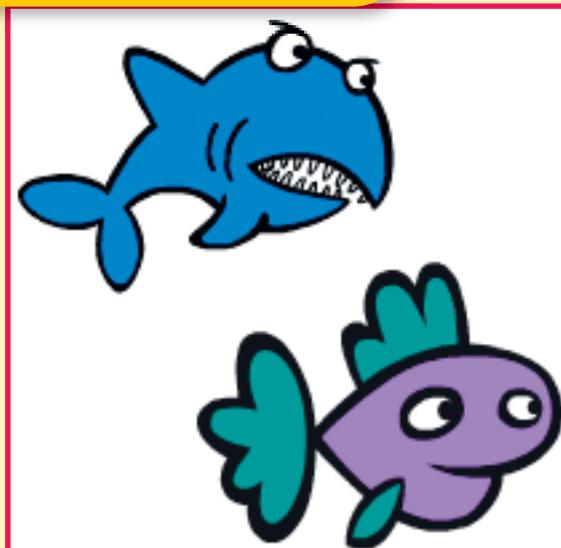
Теперь у нас есть три спрайта.



Кота в нашем мультильме не будет, удалите его. Для этого щёлкните на спрайте Кота правой кнопкой мыши и выберите команду **удалить**.

Остались только рыбы.

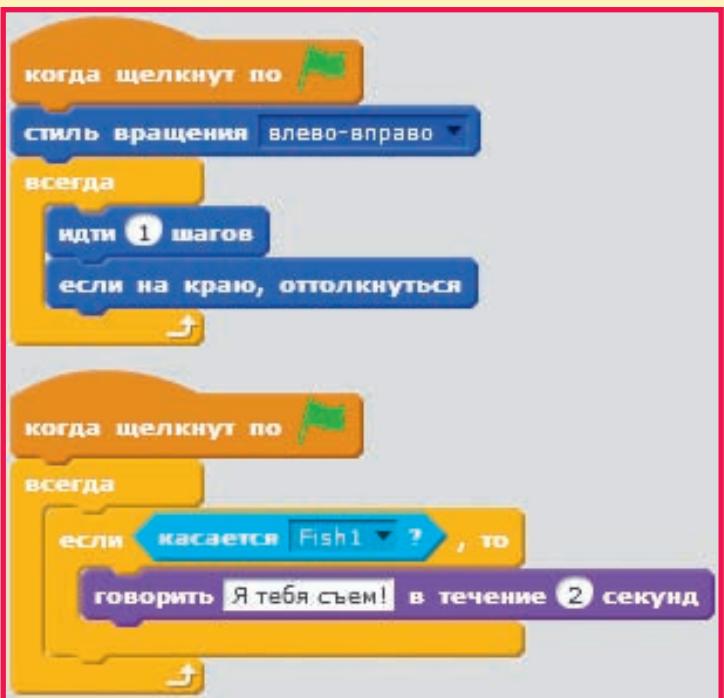
Расположите Акулу повыше, а Рыбку пониже.





## 8.2. ПРОГРАММИРУЕМ АКУЛУ

Рыбка будет находиться на месте, а Акула плавать и приговаривать «Я тебя съем!» в момент касания Рыбки. Сначала сделаем программу для Акулы.

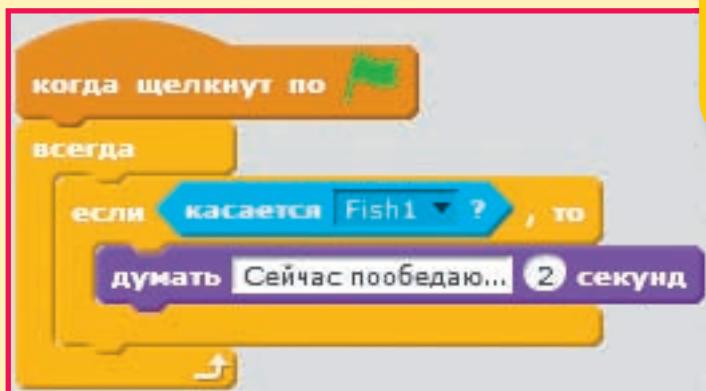


Программа Акулы состоит из двух скриптов. Скрипты запускаются при нажатии на зелёный флагок и работают одновременно. Верхний скрипт отвечает за движение Акулы, а нижний — за касание Рыбки.





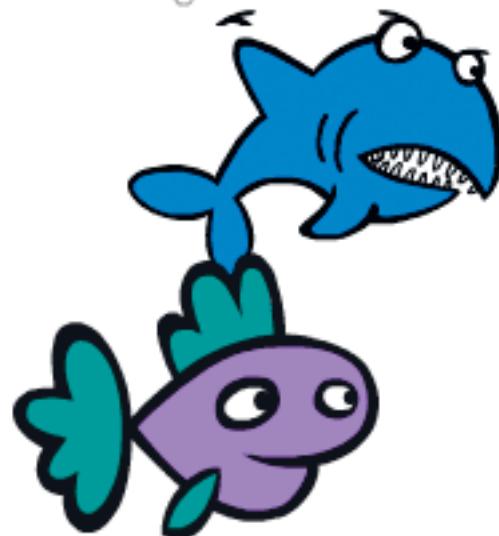
Ой! Кажется, рыбы не разговаривают...



Но думать-то они умеют! Давайте заменим блок говорить блоком думать. Второй скрипт примет такой вид.

Сейчас  
пообедаю...

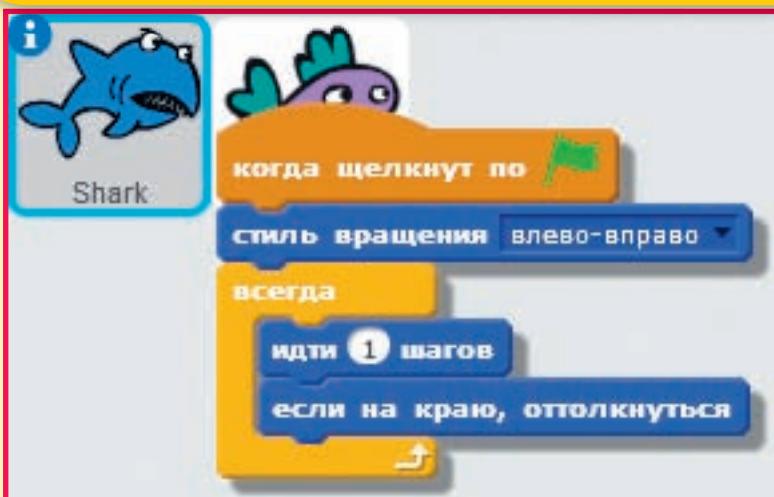
Нажмите на зелёный флашок — Акула поплыёт. Делает она это очень медленно, по одному шагу за раз. Если Акула плавает и ничего не думает, значит, она не касается Рыбки. Немного подвиньте персонажей, чтобы они соприкасались.



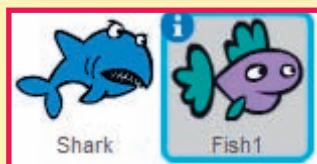


## 8.3. ПРОГРАММИРУЕМ РЫБКУ

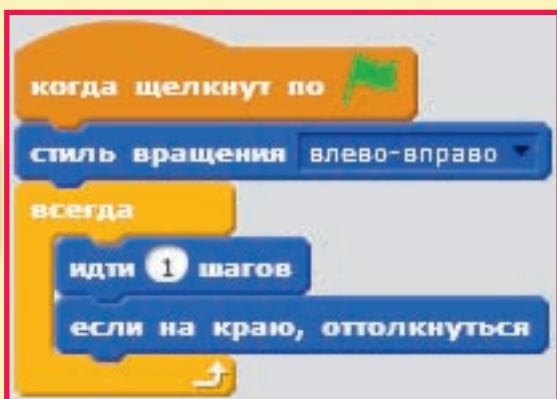
Надо дать Рыбке шанс на спасение. Пусть она тоже плавает. Скопируйте ей скрипт с движением Акулы. Нажмите на верхний блок скрипта движения, тащите его на Рыбку и отпустите в тот момент, когда курсор мыши будет точно над спрайтом Рыбки.



Теперь выберите спрайт Рыбки.



У неё появился такой же скрипт движения, как и у Акулы.



Запускаем программу. Красота! Все плавают! Но Рыбка никак не реагирует, когда её кусает Акула. Давайте скопируем ей и разговорный скрипт от Акулы.



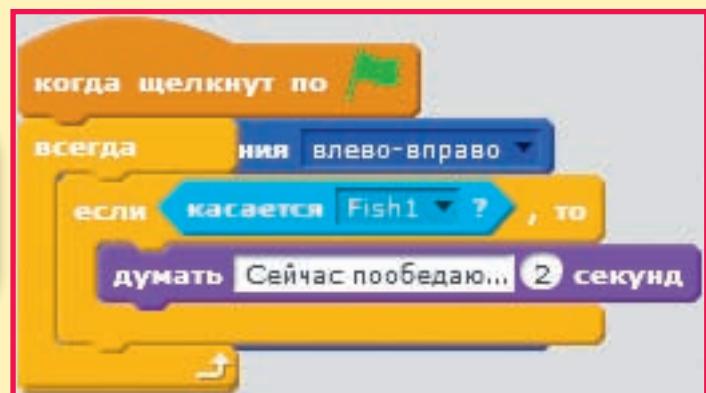


Выберите спрайт Акулы, тащите второй скрипт на Рыбку и отпустите в тот момент, когда курсор мыши будет точно над спрайтом Рыбки.

Теперь выберите спрайт Рыбки.

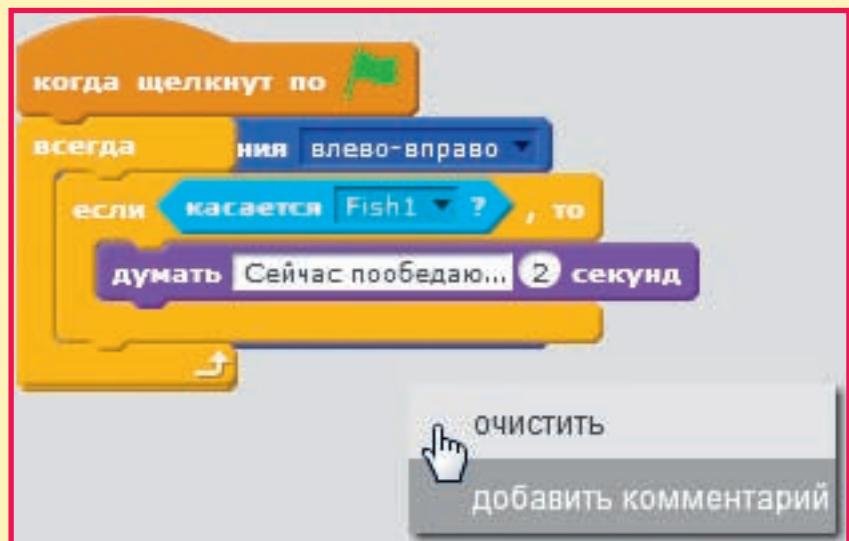


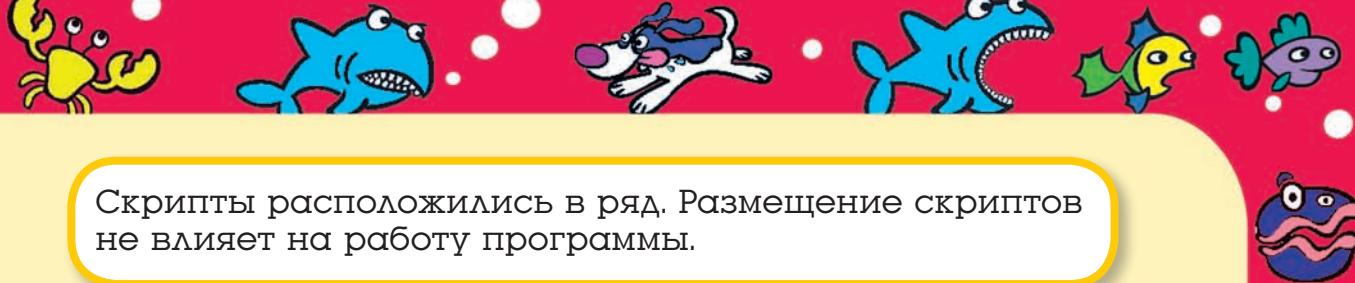
Кажется, скрипты смешались в кучу, второй скрипт лежит поверх первого!



### Совет

Для того чтобы привести расположение скриптов в порядок, есть хитрый способ. Щёлкните правой кнопкой мыши в пустом пространстве и выберите команду очистить.





Скрипты расположились в ряд. Размещение скриптов не влияет на работу программы.

```
когда щелкнут по зеленому флагу
стиль вращения влево-вправо
всегда
идти 1 шагов
если на краю, оттолкнуться
```

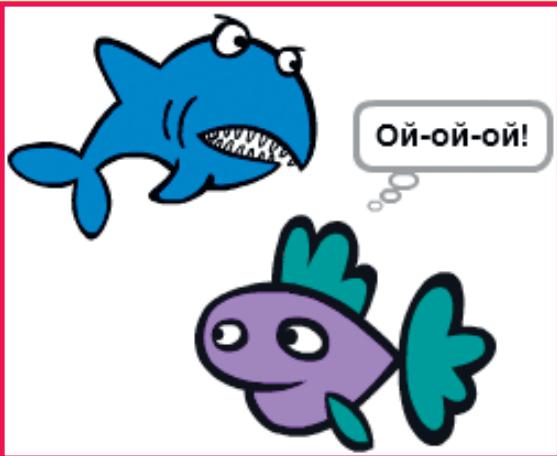
```
когда щелкнут по зеленому флагу
всегда
если касается Fish1 ?, то
думать Сейчас пообедаю... 2 секунд
```

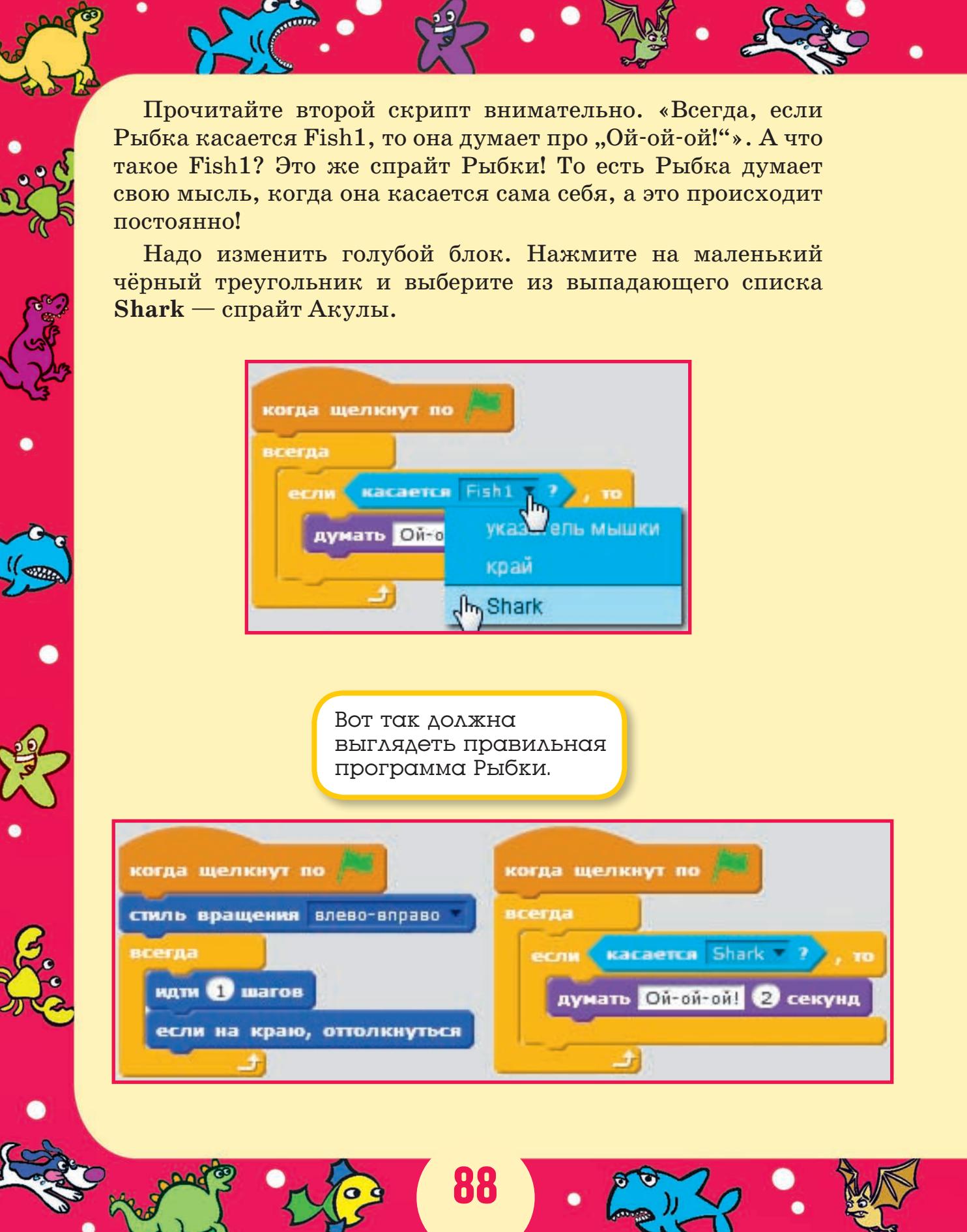
Теперь надо изменить мысль Рыбки. Напишите «Ой-ой-ой!».

```
когда щелкнут по зеленому флагу
всегда
если касается Fish1 ?, то
думать Ой-ой-ой! 2 секунд
```

## 8.4. ТЕСТИРУЕМ ПРОГРАММУ

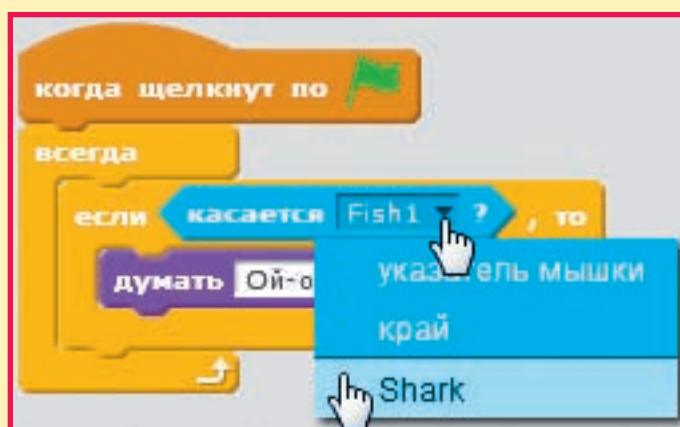
Запустите программу. Оба персонажа плавают вправо и влево, только Рыбка получилась очень уж трусливая. Она постоянно думает: «Ой-ой-ой!», даже тогда, когда не касается Акулы. Наверное, в разговорном скрипте Рыбки есть какая-то ошибка.





Прочитайте второй скрипт внимательно. «Всегда, если Рыбка касается Fish1, то она думает про „Ой-ой-ой!“». А что такое Fish1? Это же спрайт Рыбки! То есть Рыбка думает свою мысль, когда она касается сама себя, а это происходит постоянно!

Надо изменить голубой блок. Нажмите на маленький чёрный треугольник и выберите из выпадающего списка Shark — спрайт Акулы.



Вот так должна выглядеть правильная программа Рыбки.



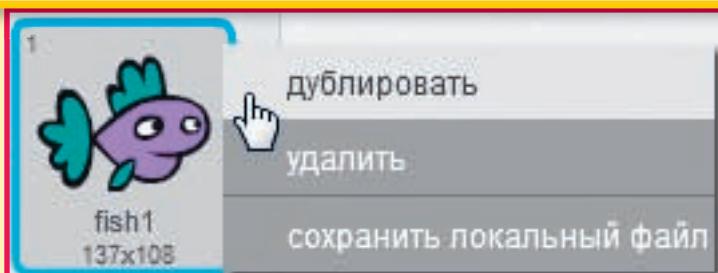


Сейчас  
пообедаю...

Ой-ой-ой!

Запустите программу. Теперь всё работает так, как было задумано. Однако Рыбка продолжает улыбаться в момент, когда её касается Акула. Это выглядит немного не естественно.

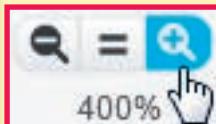
Давайте сделаем так, чтобы в момент касания Рыбка огорчалась. Для этого нам придётся создать новый костюм. Выберите Рыбку и перейдите на вкладку **Костюмы**. Щёлкните правой кнопкой мыши по костюму и выберите команду **дублировать**.

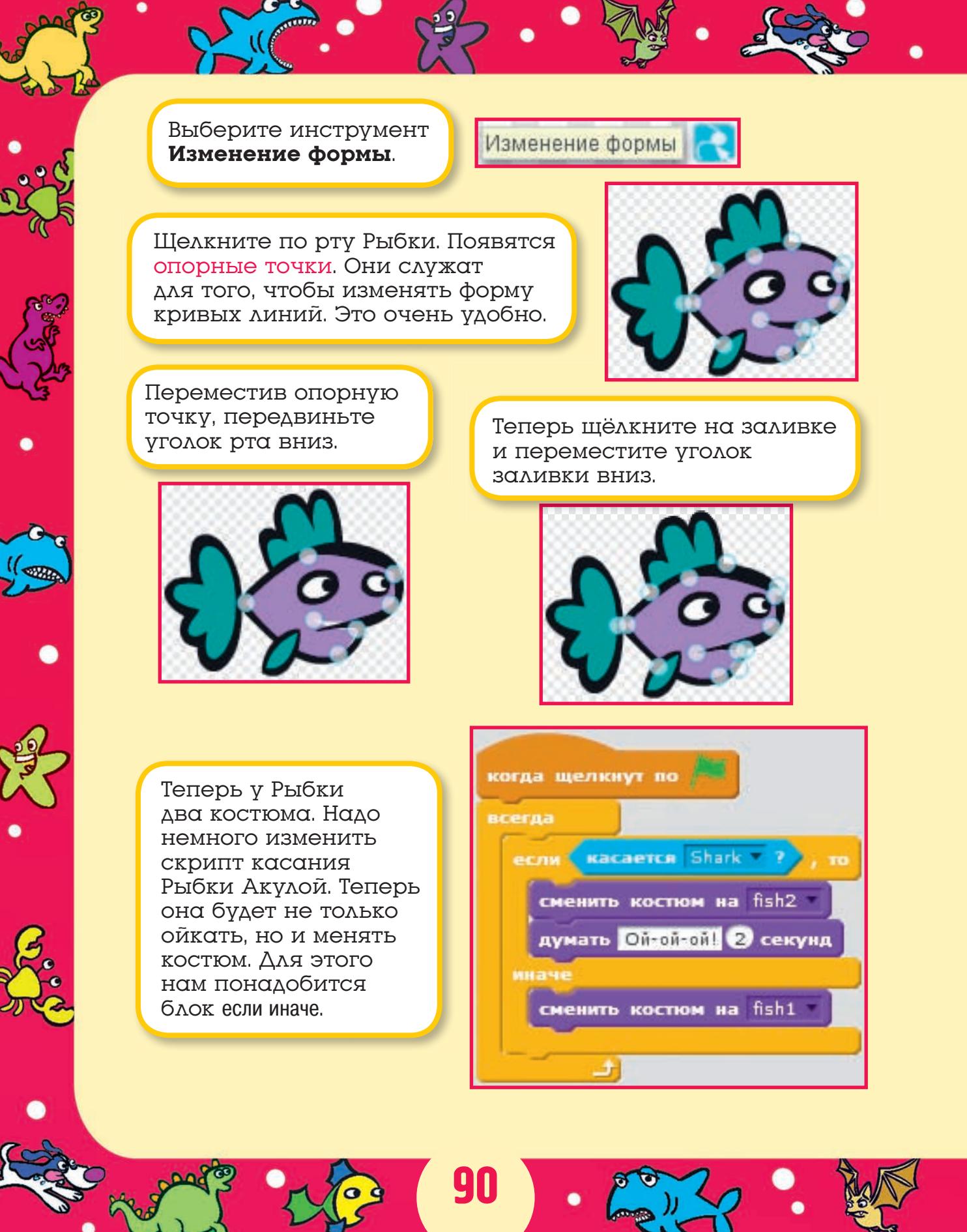


Появится новый костюм Рыбки — точная копия первого.

Увеличьте масштаб до 400% (это в четыре раза больше нормального размера).

Рыбка станет огромной!





Выберите инструмент  
**Изменение формы**.

Изменение формы

Щелкните по рту Рыбки. Появятся **опорные точки**. Они служат для того, чтобы изменять форму кривых линий. Это очень удобно.



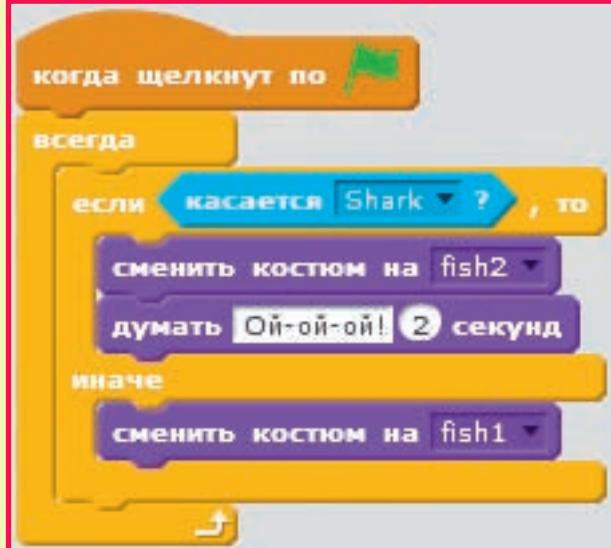
Переместив опорную точку, передвиньте уголок рта вниз.



Теперь щёлкните на заливке и переместите уголок заливки вниз.



Теперь у Рыбки два костюма. Надо немного изменить скрипт касания Рыбки Акулой. Теперь она будет не только ойкать, но и менять костюм. Для этого нам понадобится блок если иначе.





При выполнении условия блок если иначе выполнит верхние блоки — Рыбка сменит костюм на грустный и на 2 секунды задумается о своей судьбе. В случае если условие не выполняется, будет выполнен нижний блок, переодевающий Рыбку в весёлый костюм.

Мультик готов. Не забудьте сохранить его.

## 8.5. ЗАДАНИЯ

1. Добавьте морской фон.
2. Измените скрипт Акулы, чтобы при касании Рыбки она изменяла костюм и открывала пасть.
3. Сделайте так, чтобы Акула плавала в 2 раза быстрее рыбки.
4. Добавьте в мультик ещё одну рыбку, которая будет плавать и вежливо здороваться со всеми при касании.
5. Добавьте в скрипты движения блоки поворота, чтобы движение стало беспорядочным.
6. Добавьте аквалангиста, который будет молча плавать внизу и останавливаться через каждые 100 шагов на 3 секунды для того, чтобы полюбоваться красотой подводного мира.





# ГЛАВА 9. ЧТО ТАКОЕ КООРДИНАТЫ X И Y

## 9.1. ПЕРЕМЕЩЕНИЕ ПО ГОРИЗОНТАЛИ

Вы уже знаете, что для перемещения спрайтов по экрану существуют синие блоки идти, повернуть и повернуть в направлении.

идти 10 шагов

поворнуть ⚡ на 15 градусов

поворнуть в направлении 90°

Эти блоки позволяют переместить спрайт в любую точку экрана. Однако для перемещения спрайта в горизонтальном направлении (справа налево или слева направо) лучше использовать специальный блок изменить x.

изменить x на 10

Соберите вот такие скрипты и посмотрите, как они работают.

когда клавиша стрелка направо нажата  
изменить x на 10

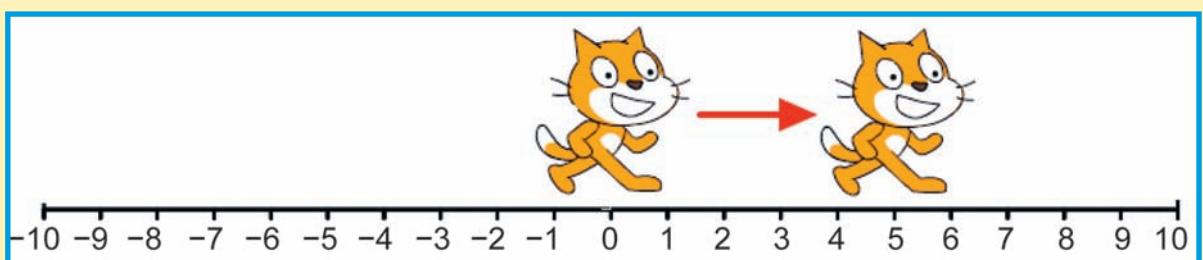
когда клавиша стрелка влево нажата  
изменить x на -10

Спрайт управляетя стрелками и перемещается в горизонтальном направлении, но почему это происходит? Для того чтобы понять, как работают эти скрипты, нужно вспомнить про координатную ось x (произносится как «ось икс»).

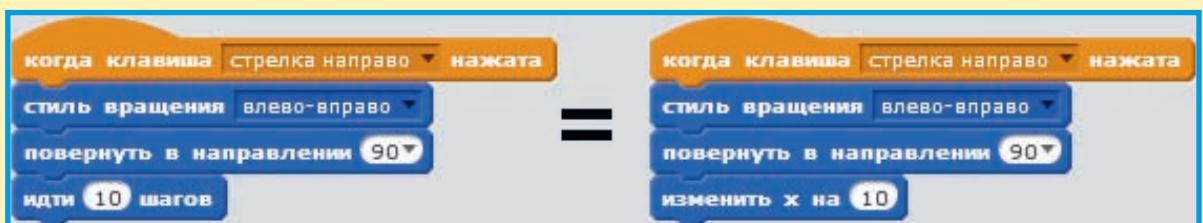




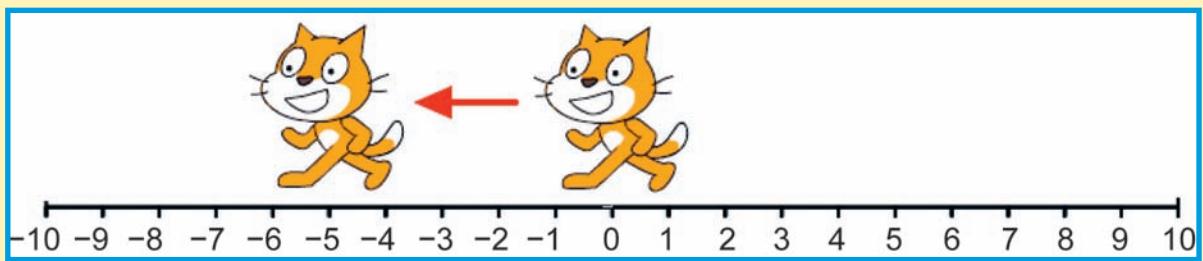
Ось  $x$  — это горизонтальная линия, состоящая из множества точек, каждая из которых имеет свой номер. Центр оси  $x$  находится в середине сцены, в этой точке координата  $X$  равна нулю. Значения  $X$  справа от центральной точки — положительные, а слева — отрицательные. Если спрайт расположен в центре экрана и его координата  $X$  равна нулю, а затем он получит команду изменить  $x$  на 5, то спрайт переместится на 5 шагов вправо.



Таким образом, получается, что следующие скрипты выполняют одно и то же действие.



В случае же если спрайт расположен в центре экрана, а затем он получит команду изменить  $x$  на  $-5$ , то спрайт переместится на 5 шагов влево.



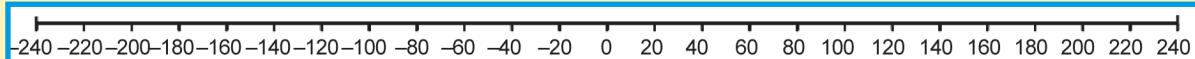


То есть эти скрипты эквивалентны.

```

когда клавиша стрелка влево нажата
стиль вращения влево-вправо
повернуть в направлении 90
идти -10 шагов
    =
когда клавиша стрелка влево нажата
стиль вращения влево-вправо
повернуть в направлении 90
изменить x на -10
  
```

Координата  $X$  в Scratch может изменяться от  $-240$  до  $240$ .



Если дать спрайту задание перейти в точку с координатой  $X$  больше  $240$ , то спрайт скроется за краем экрана, а если координата  $X$  станет больше миллиарда, то спрайт исчезнет в неизвестном направлении.



### Вопрос

Как вернуть на сцену исчезнувший спрайт?

Движение с помощью координаты  $X$  позволяет более гибко управлять перемещением спрайтов. Например, можно сделать так, чтобы Кот, который дошёл до правой границы сцены, появлялся слева и продолжал движение направо. И наоборот, Кот, который дошёл до левой границы, появлялся справа и продолжал движение налево.

```

когда щелкнут по флагу
стиль вращения влево-вправо
всегда
если положение x > 240, то
  установить x в -240
если положение x < -240, то
  установить x в 240
когда клавиша стрелка направо нажата
повернуть в направлении 90
изменить x на 10
когда клавиша стрелка влево нажата
повернуть в направлении -90
изменить x на -10
  
```

Создайте для Кота вот такую программу.



Обратите внимание, здесь использован новый блок установить  $x$ , он позволяет «телеportировать» спрайт в нужную точку. В нашем случае спрайт «телеportируется» на противоположную сторону сцены. Овальный блок положение  $x$  — это *защищённая переменная*, которая всегда хранит значение координаты  $X$  спрайта. Подробнее о переменных вы узнаете в следующих главах.

## 9.2. ПЕРЕМЕЩЕНИЕ ПО ВЕРТИКАЛИ

Для перемещения спрайтов по вертикали существует блок изменить  $y$ .

изменить  $y$  на 10

Сделайте вот такие скрипты и посмотрите, как они работают.

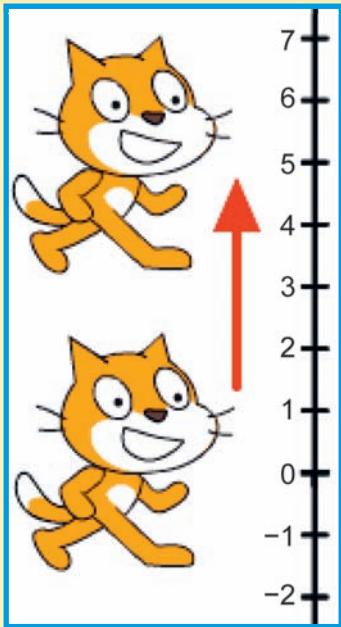
```
когда клавиша стрелка вверх нажата
изменить y на 10

когда клавиша стрелка вниз нажата
изменить y на -10
```

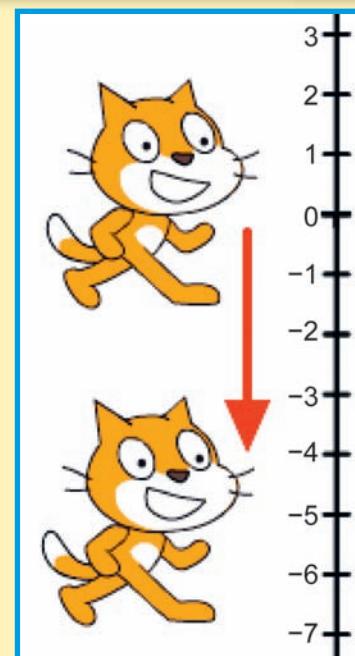
Спрайт управляетя стрелками и перемещается в вертикальном направлении. Для того чтобы понять, как работают эти скрипты, нужно вспомнить про координатную ось  $y$  (произносится как «ось игрек»). Ось  $y$  очень похожа на ось  $x$ , она тоже состоит из множества пронумерованных точек. Её центр находится в середине сцены. Есть только одно отличие — ось  $y$  расположена вертикально.



Значения  $Y$  сверху от центральной точки — положительные, а снизу — отрицательные. Если спрайт расположен в центре экрана и его координата  $Y$  равна нулю, а затем он получит команду изменить  $Y$  на 5, то спрайт переместится на 5 шагов вверх.



В случае же если спрайт расположен в центре экрана и затем он получит команду изменить  $Y$  на -5, то спрайт переместится на 5 шагов вниз.



Таким образом, получается, что следующие скрипты выполняют одно и то же действие.

The image shows two identical Scratch scripts side-by-side, separated by an equals sign (=). Both scripts perform the same actions: when the up arrow key is pressed, the cat sprite moves forward 10 steps, turns around in a circle, and changes its  $Y$  coordinate by 10 units (either up 5 or down 5).

Left script blocks:

- когда клавиша стрелка вверх нажата
- стиль вращения кругом
- поворнуть в направлении 0°
- идти 10 шагов

Right script blocks:

- когда клавиша стрелка вверх нажата
- стиль вращения кругом
- поворнуть в направлении 0°
- изменить  $Y$  на 10

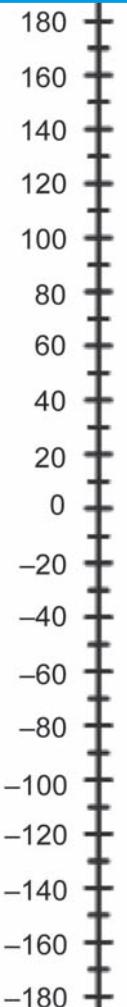
То есть эти скрипты эквивалентны.

когда клавиша стрелка вниз нажата  
повернуть в направлении 180  
идти 5 шагов



когда клавиша стрелка вниз нажата  
изменить у на -5

Координата У в Scratch может изменяться от  $-180$  до  $180$ .



Если дать спрайту задание перейти в точку с координатой У больше  $180$  или меньше  $-180$ , то спрайт, как вы, наверное, догадались, скроется за краем сцены. Добавьте в программу Кота вот такие скрипты.

когда клавиша стрелка вверх нажата  
изменить у на 5

когда клавиша стрелка вниз нажата  
изменить у на -5

когда щелкнут по флагу  
всегда

если положение у > 180, то  
установить у в -180

если положение у < -180, то  
установить у в 180



Теперь, если Кот поднимется до верхней границы сцены, то он появится снизу и продолжит движение вверх. А если Кот опустится до нижней границы, то он появится сверху и продолжит движение вниз.

Обратите внимание, здесь использован новый блок установить *y*. Он работает так же, как и блок установить *x*, и позволяет «телеportировать» спрайт в нужную точку. В нашем случае спрайт «телеportируется» на противоположную сторону сцены. Овальный блок положение *y* — это *защищённая переменная*, которая всегда хранит значение координаты *Y* спрайта. Подробнее о защищённых переменных вы узнаете в следующих главах.

## 9.3. РИСОВАНИЕ ПО КООРДИНАТАМ

Теперь, когда вы знакомы с координатами *X* и *Y*, можно попробовать нарисовать что-нибудь на сцене с применением новых блоков.

Создайте новый проект и соберите для Кота вот такой скрипт.



При запуске программы Котик изменит размер и станет в 5 раз меньше обычного (ведь 20% в 5 раз меньше, чем 100%). Сцена очистится, перо опустится, и Кот четыре раза изменит свои координаты так, что получится квадрат.



С применением блока повторить можно рисовать очень красивые узоры. Создайте новый проект и сделайте для Кота вот такой скрипт.

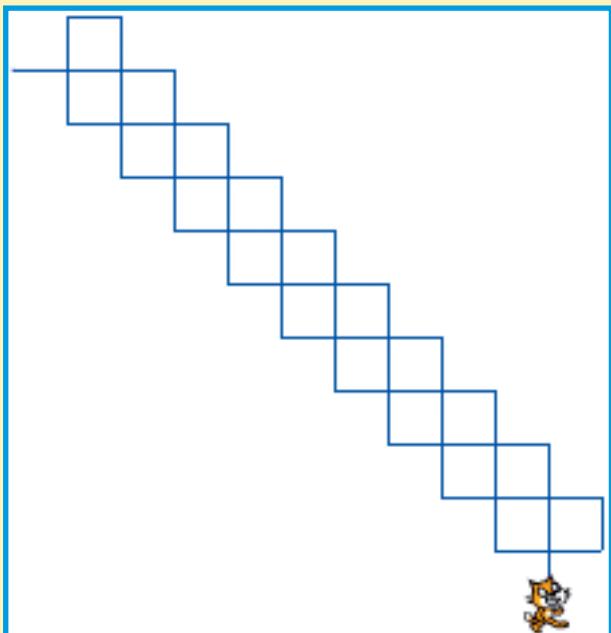
Перетащите Кота в центр сцены и запустите скрипт, он нарисует ряд квадратиков.

когда щелкнут по флагу  
установить размер 20 %  
очистить  
опустить перо  
повторить 10  
изменить x на 40  
изменить y на 20  
изменить x на -20  
изменить y на -20



Измените значение в самом нижнем блоке. Вместо  $-20$  введите  $-40$ . Теперь Котик рисует вот такой узор.

Изменяйте значения в блоках движения и нарисуйте свои узоры. Не забывайте вручную перемещать Кота в центр экрана.



## 9.4. ЗАДАНИЯ

1. Доработайте программу, в которой вы научили Котика рисовать: добавьте скрипт, который по нажатию клавиши <Пробел> очищает сцену, перемещает Кота в центр и восстанавливает его размер.

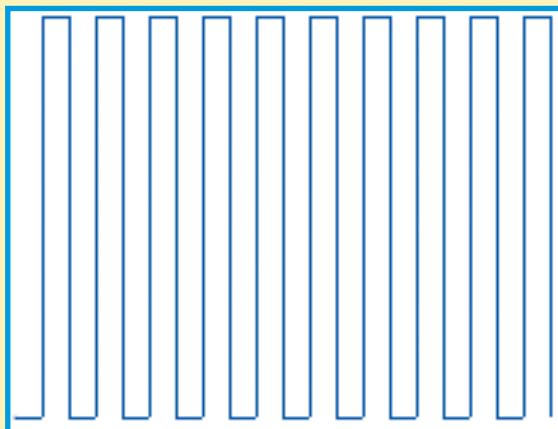
2. Сделайте так, чтобы при запуске программы Кот автоматически попадал в центр экрана, и вам не приходилось бы перетаскивать его вручную.

3. Измените цвет и толщину пера.

4. На листе тетради «в клеточку» нарисуйте змейку, а затем научите Кота её рисовать.



5. На листе тетради «в клеточку» нарисуйте забор, а затем научите Кота его рисовать.

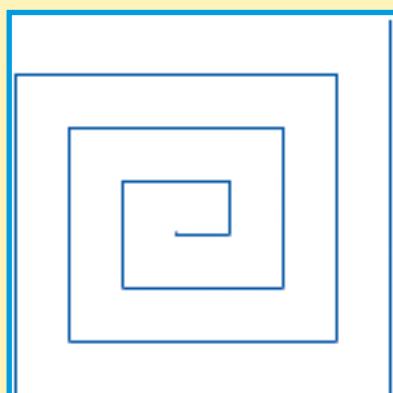


6. На листе тетради «в клеточку» нарисуйте спираль, а затем научите Кота её рисовать.



### Подсказка

Блок повтора здесь не нужен.



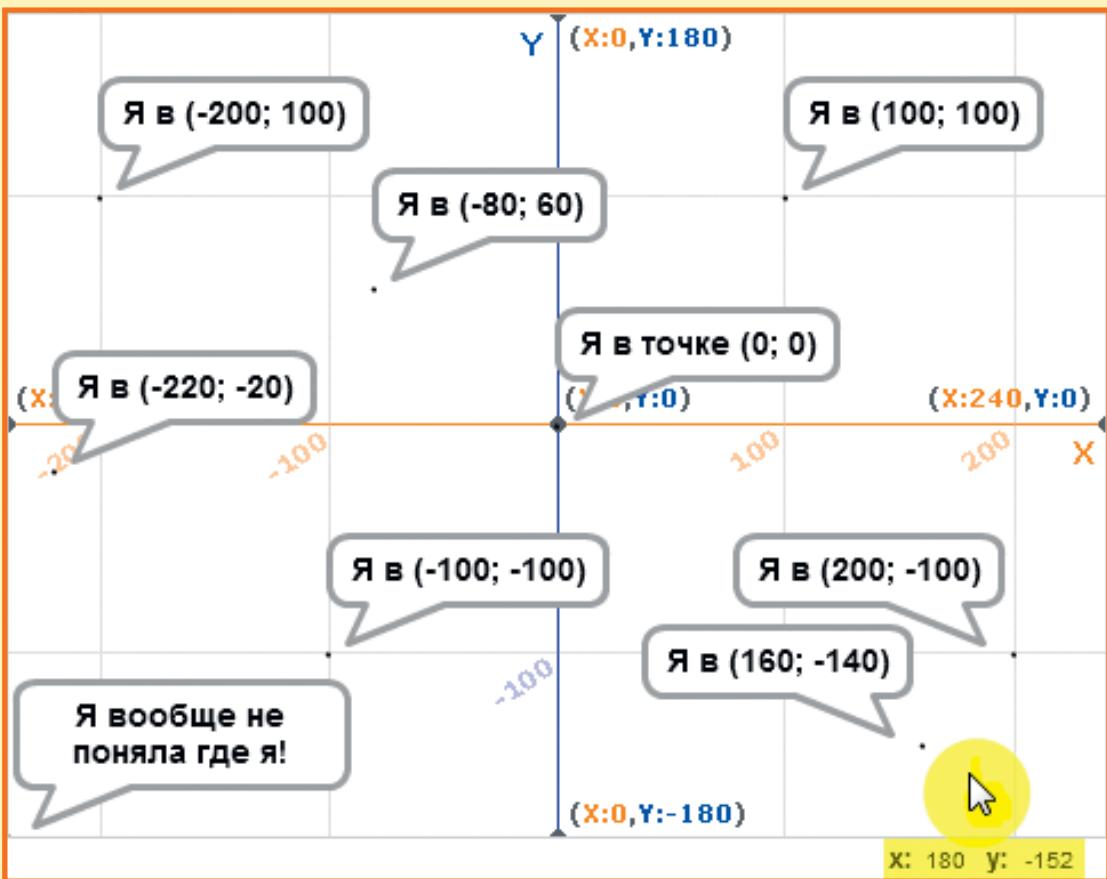


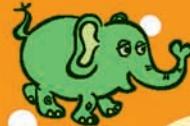
# ГЛАВА 10.

## МУЛЬТИФИЛЬМ «ПИКО И ПРИВИДЕНИЕ»

### 10.1. Координатная плоскость

Как вы знаете, монитор компьютера состоит из множества отдельных точек — пикселов. Сцена в Scratch также состоит из отдельных точек. Размер сцены 480 точек слева направо и 360 точек снизу вверх. Для того чтобы отличать точки одну от другой, им даны две координаты:  $X$  и  $Y$ . Каждая точка сцены имеет свои координаты. Координаты записывают в скобках: сначала  $X$ , потом  $Y$  —  $(X; Y)$ .



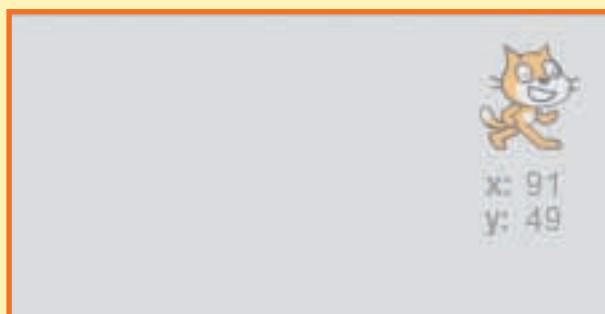


Обратите внимание на правый нижний угол сцены. Под ним указаны координаты кончика курсора мыши. Перемещайте мышь по сцене и посмотрите, как изменяются координаты. Попробуйте установить курсор точно в точку с координатами (0; 0).



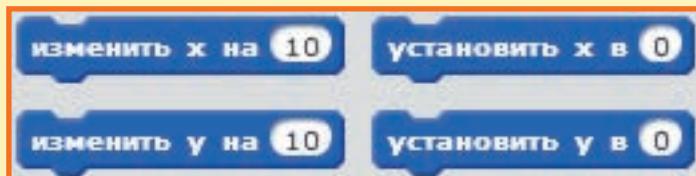
### Совет

В правом верхнем углу области скриптов вы всегда можете увидеть координаты выбранного спрайта. Сейчас Котик расположен в точке с координатами (91; 49).



## 10.2. НОВЫЕ БЛОКИ ПЕРЕМЕЩЕНИЯ ПО КООРДИНАТНОЙ ПЛОСКОСТИ

Вы уже научились пользоваться некоторыми блоками перемещения по сцене с помощью координат.





Осталось познакомиться ещё с двумя. Блок **перейти в x** «телеported» спрайт в точку сцены с указанными координатами. Он делает то же самое, что и пара блоков установить x и установить y. Следующие блоки переместят спрайт в центр сцены.



Ещё один блок **плыть 1 секунд в точку x: 0 y: 0** плавно перемещает спрайт в требуемую точку экрана.



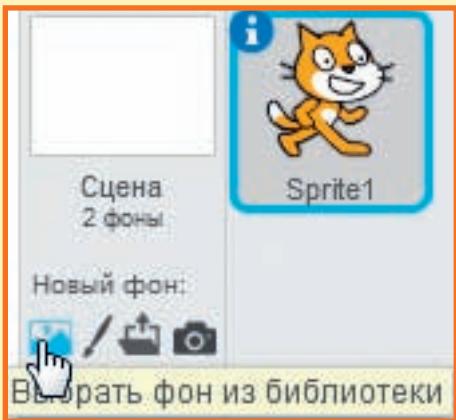
Теперь, когда вы знакомы со всеми блоками движения, можно создать мультфильм.

## 10.3. ДЕЛАЕМ МУЛЬТИФИЛЬМ

Как-то раз Пико гулял в старом замке. Он, как обычно, слушал свой плеер и совсем не заметил, как появилось Привидение! «Мама!» — закричал Пико. Привидение улыбнулось, сказав: «Шутка!» — и растаяло.

Давайте сделаем маленький мультфильм о приключениях Пико в старом замке.

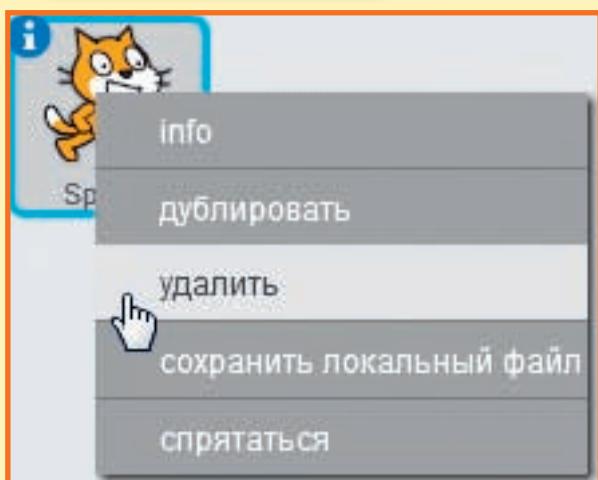
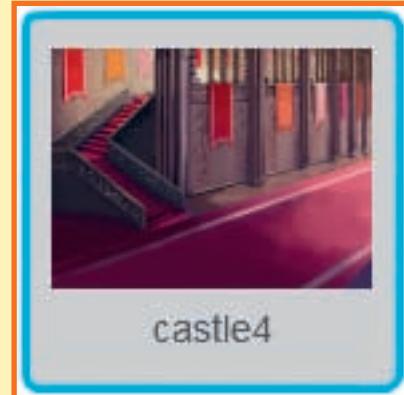
Создайте новый проект.  
Сначала измените фон.  
Нажмите кнопку **Выбрать фон из библиотеки**.



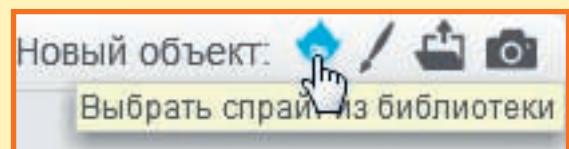


Выберите старый замок и нажмите кнопку **OK**.

Кота в нашей истории не будет. Удалите его.



Добавьте спрайты Пико и Привидения.



Теперь у нас есть старый замок, Пико и Привидение.



Начинаем программировать!



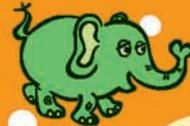
## 10.4. ПРОГРАММИРУЕМ ПИКО И ПРИВИДЕНИЕ

Сначала запрограммируем Пико.

```
когда щелкнут по флагу
  установить размер 75 %
  перейти в x: 200 у: -100
  стиль вращения влево-вправо
  повернуть в направлении -90°
  повторить (30)
    изменить x на (-5)
    следующий костюм
    ждать 0.3 секунд
  говорить [Мама!] в течение 2 секунд
```

Цикл анимации ходьбы Пико.

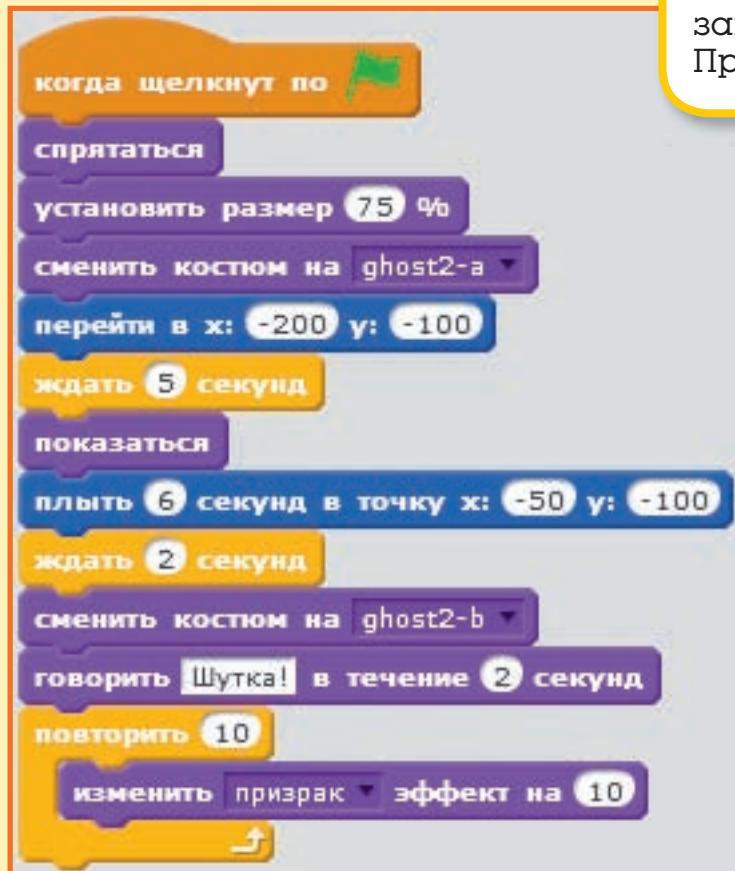
Давайте посмотрим, как работает этот скрипт. Сначала Пико немного уменьшит свой размер и перейдёт в правый нижний угол сцены, в точку с координатами  $(200; -100)$ . Затем мы задаём стиль вращения влево-вправо и поворачиваем Пико лицом налево, ведь если такой стиль вращения не задать, то Пико перевернётся вверх ногами! Затем Пико 30 раз циклически повторяет несколько действий — смещается влево на 5 шагов, изменяет костюм для анимации ходьбы и немного ждёт (примерно треть секунды). После того как работа цикла закончится, Пико окажется на 150 шагов левее, чем в начале — он будет почти в центре сцены.



## Обратите внимание!

К светло-коричневому блоку повторить через меню, открываемому правой кнопкой мыши, добавлено примечание. *Примечание* — это пометка, помогающая другим понять, как работает ваш проект. Не забывайте оставлять примечания в своих проектах!

Теперь  
запрограммируем  
Привидение.



```
когда щелкнут по флагу
спрятаться
установить размер на 75 %
сменить костюм на ghost2-a
перейти в x: -200 у: -100
ждать 5 секунд
показаться
плыть 6 секунд в точку x: -50 у: -100
ждать 2 секунды
сменить костюм на ghost2-b
говорить Шутка! в течение 2 секунд
повторить (10)
изменить [призрак] [эффект] на 10
```

После запуска программы Привидение спрячется, немножко уменьшит размер, наденет злой костюм, перейдёт в левый нижний угол экрана и притаится на 5 секунд. Когда терпение



Привидения лопнет, то оно появится и не спеша поплывёт на встречу Пико. Пико в этот момент всё ещё гуляет с закрытыми глазами и наслаждается музыкой. Но вот он заметил этот ужас и закричал! Привидение подождёт 2 секунды, пока Пико наорётся, переоденется в милый костюм и скажет: «Шутка!», после чего растает, циклически изменив эффект призрака 10 раз на 10.

Проект готов, сохраните его.

## 10.5. ЗАДАНИЯ

1. Ускорьте передвижение Пико.
2. Ускорьте передвижение Привидения.
3. А теперь замедлите исчезновение Привидения.
4. Сделайте так, чтобы после исчезновения Привидения Пико продолжил движение до левой границы сцены.
5. Добавьте ещё одно привидение, которое будет летать в верхней части сцены в милом костюме и раз в 5 секунд будет на 1 секунду становиться злым.
6. Добавьте смех Привидению, записав его.





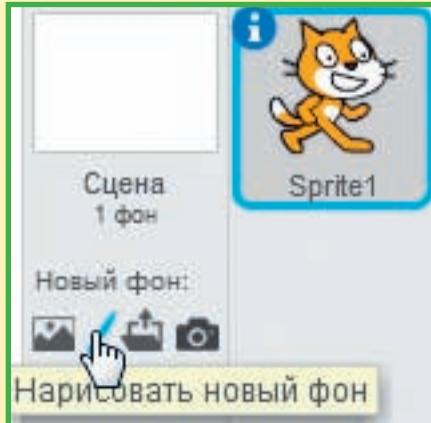
# ГЛАВА 11. ИГРА «ЛАБИРИНТ»

Жила-была девочка, и звали её Гига. Она была очень ответственная. У неё был младший брат. Его звали Нано, а он был очень безответственный. Как-то раз Нано пошёл гулять один и увидел тайный вход в лабиринт. Он не позвал сестру, а отправился бродить по тайному подземелью. Прошло три часа, и Гига развелась. «Где можно так долго гулять?» — подумала она и отправилась на поиски брата. Увидев вход в лабиринт, Гига всё поняла. Она хорошо знала Нано, из всех возможных проблем он всегда выбирал самые сложные. «Пойду на помощь», — решила бесстрашная Гига и вошла в лабиринт.

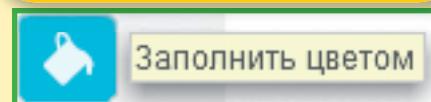
Давайте сделаем игру, в которой Гиге надо пройти через лабиринт и найти Нано. Стен касаться нельзя, ведь они из лавы!

## 11.1. РИСУЕМ ЛАБИРИНТ

Создайте новый проект.  
Начнём с лабиринта. Нажмите  
кнопку **Нарисовать новый фон**.



Выберите инструмент  
**Заполнить цветом**.



Выберите чёрный цвет.

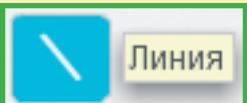




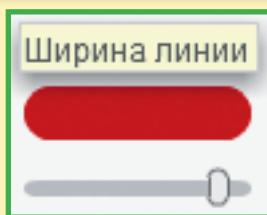
Залейте сцену чёрным.  
Теперь нарисуем стены из лавы.  
Выберите тёмно-малиновый цвет.



Выберите инструмент  
**Линия**.



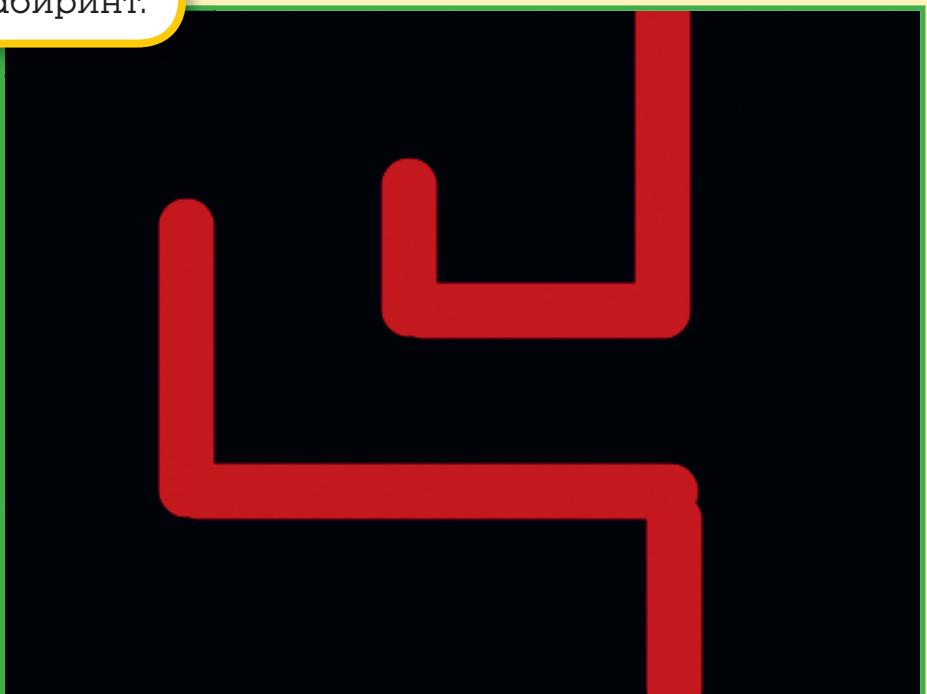
Увеличьте ширину линии.



### Совет

Для того чтобы линии  
получались ровными,  
рисуйте при нажатой  
клавише <Shift>.

Нарисуйте лабиринт.



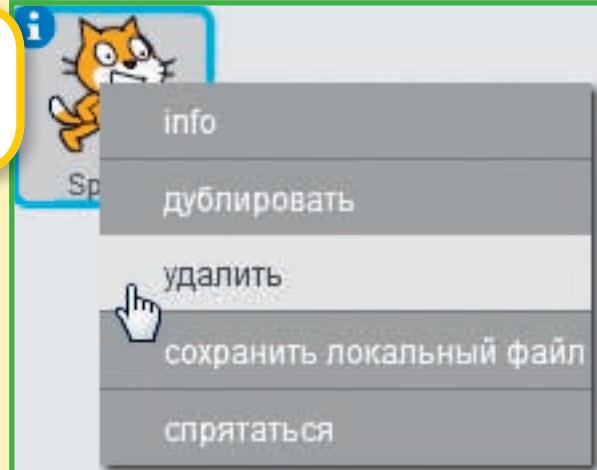
Кота в нашей истории не будет. Удалите его.

Добавьте спрайты Гиги и Нано.

Новый объект:  
Выбрать спрайт из библиотеки

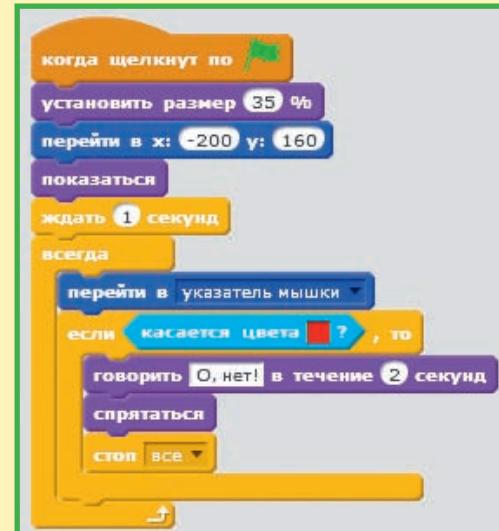
Теперь у нас есть лабиринт, Гига и Нано.

Можно начинать программировать!



## 11.2. ПРОГРАММИРУЕМ ГИГУ И НАНО

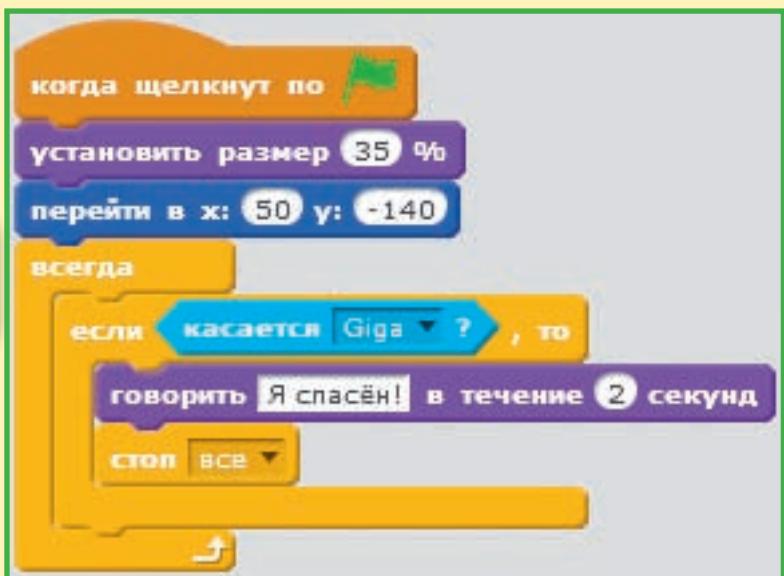
Сначала запрограммируем Гигу.





Рассмотрим, как работает скрипт. Сначала уменьшается размер Гиги, чтобы она смогла притиснуться через лабиринт, она перемещается в левый верхний угол, делается видимой и 1 секунду ждёт, чтобы вы также переместили курсор мыши в левый верхний угол. Далее Гига в вечном цикле постоянно цепляется за курсор мыши и проверяет, не касается ли она лавы малинового цвета. Если касается, то конец игры. Гига говорит печальную фразу, исчезает, а выполнение программы останавливается блоком СТОП все.

Теперь запрограммируем Нано.



Задача Нано очень простая — стой и жди, когда тебя спасут. В начале работы программы Нано уменьшается, так же как и сестра, и переходит в точку (50; -140). В вечном цикле он проверяет, не спасла ли его Гига? Если она коснётся брата, то он радостно закричит, а программа будет остановлена, так как вы выиграли.

Сохраните проект.

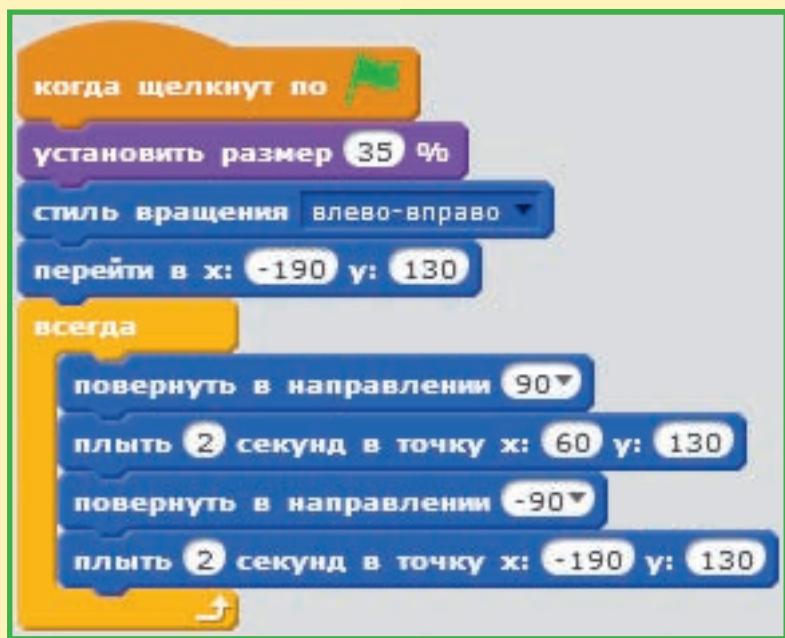
## 11.3. УСЛОЖНЯЕМ ИГРУ

Если вы немного поиграете в эту игру, то поймёте, что она не очень сложная. Для того чтобы сделать проект интереснее, можно добавить в лабиринт Охранников, с которыми нельзя встречаться Гиге.

Сегодня охранником поработает летучая мышь. Добавьте её из библиотеки спрайтов.



Соберите для Охранника такой скрипт.



Размер Охранника будет небольшой, стиль вращения влево-вправо. Полёт он начинает из левого верхнего угла из точки с координатами  $(-190; 130)$ . Затем в вечном цикле Охранник плывёт в точку  $(60; 130)$ , потом разворачивается и плывёт назад. Теперь Гиге будет непросто спасти Нано!



Скрипт Гиги  
надо немного  
модернизировать,  
чтобы она  
реагировала на  
Охранника.

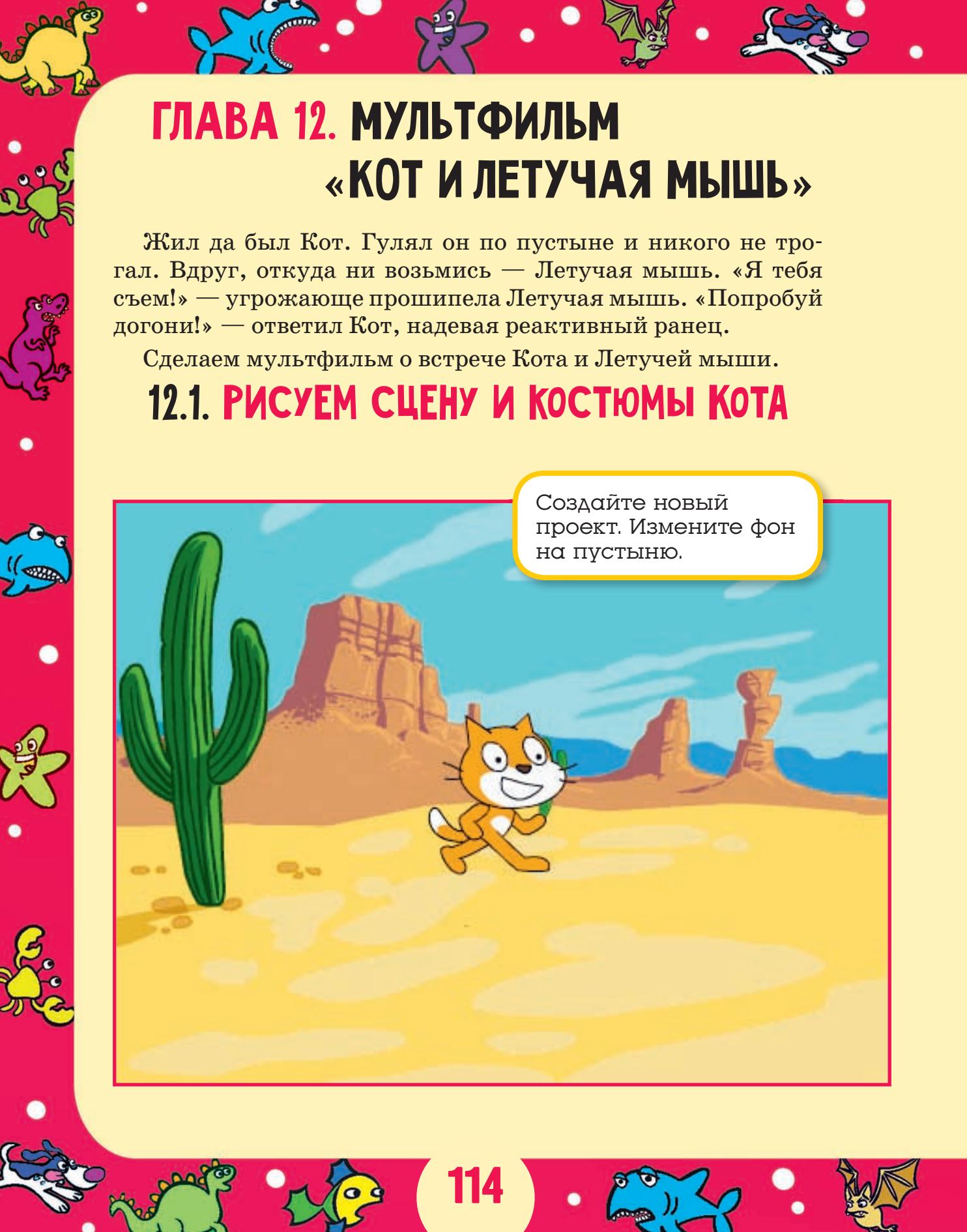
В скрипт добавлено  
ещё одно условие,  
которое срабатывает  
при касании  
Охранника (Bat1).  
Проект готов,  
сохраните его.

```
когда щелкнут по флагу
установить размер на 35 %
перейти в x: -200 y: 160
показаться
ждать [1] секунд
всегда
[перейти в [указатель мышки]
если [касается цвета [красный] ?] , то
[говорить [Oh, нет!] в течение [2] секунд
спрятаться
стоп [все]
если [касается [Bat1] ?] , то
[говорить [Прости, брат!] в течение [2] секунд
спрятаться
стоп [все]]]
```

## 11.4. ЗАДАНИЯ

1. Сделайте так, чтобы Гига управлялась клавишами-стрелками, а не мышью.
2. Ускорьте движение Охранника.
3. Добавьте ещё одного Охранника.
4. Уменьшите размер героев до 20% и нарисуйте новый, более сложный лабиринт.
5. У Летучей мыши два костюма. Сделайте так, чтобы при движении она махала крыльями.



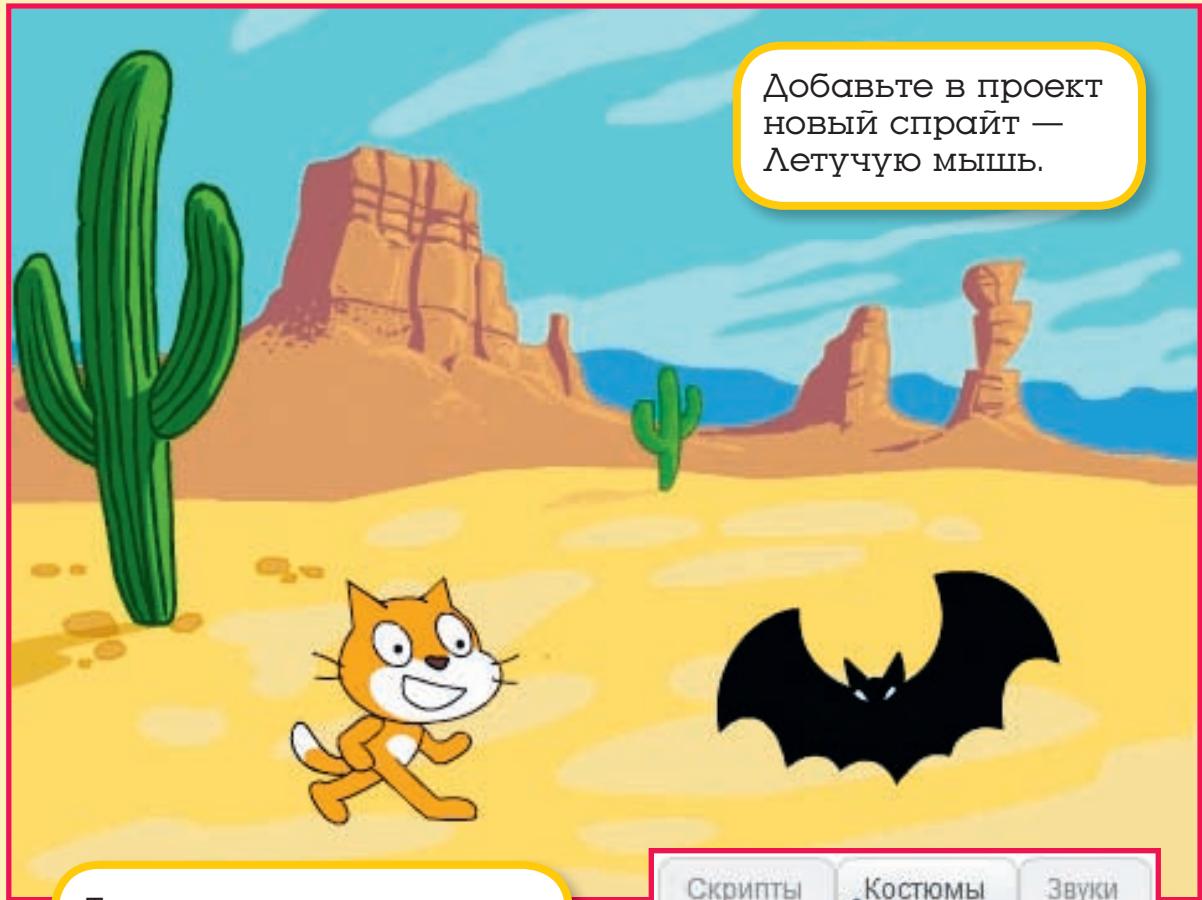


## ГЛАВА 12. МУЛЬТИФИЛЬМ «КОТ И ЛЕТУЧАЯ МЫШЬ»

Жил да был Кот. Гулял он по пустыне и никого не трогал. Вдруг, откуда ни возьмись — Летучая мышь. «Я тебя съем!» — угрожающе прошипела Летучая мышь. «Попробуй догони!» — ответил Кот, надевая реактивный ранец.

Сделаем мультфильм о встрече Кота и Летучей мыши.

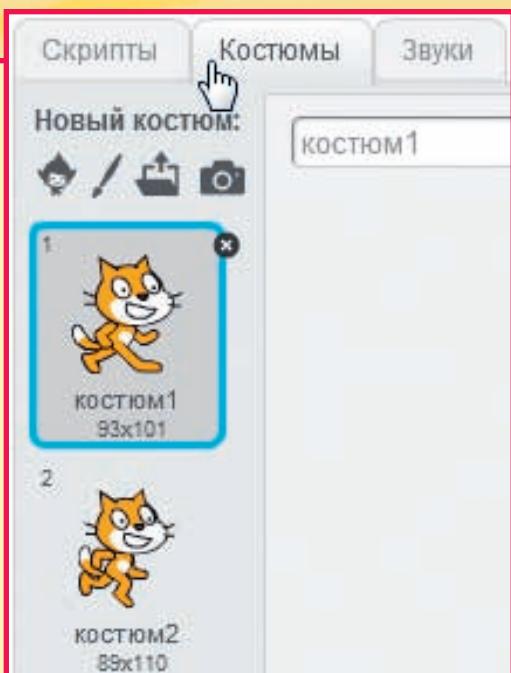
### 12.1. РИСУЕМ СЦЕНУ И КОСТЮМЫ КОТА

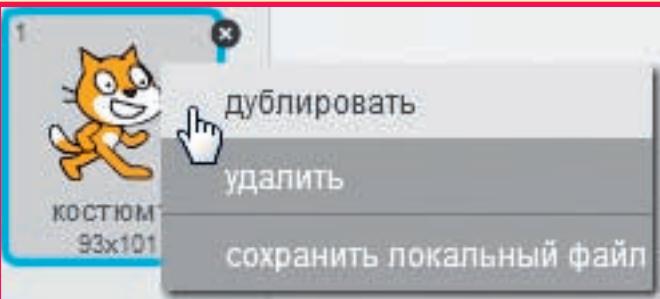


Добавьте в проект  
новый спрайт —  
Летучую мышь.

Теперь нужно нарисовать новые костюмы Коту и Летучей мыши. У Кота будет всего четыре костюма. Два обычных, еще один костюм с реактивным ранцем и один с включённым ранцем.

Выберите Кота и перейдите на вкладку **Костюмы**.





Щёлкните на первом костюме правой кнопкой мыши и выберите команду **дублировать**.

Появится новый костюм. На нём мы и нарисуем реактивный ранец. Увеличьте масштаб до 400% (это в четыре раза больше нормального размера).

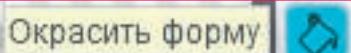


Кот стал огромным, так рисовать гораздо удобнее!

Выберите инструмент **Прямоугольник**.

Нарисуйте прямоугольник.

Теперь выберите инструмент **Окрасить форму**.



Выберите серый цвет.



Переключите цвета, щёлкнув на белом квадрате.



Выберите серый цвет другого оттенка.



Выберите способ заливки.



Щёлкните на прямоугольнике, он окрасится.



Теперь надо нарисовать сопло, из которого будет вырываться пламя. Снова выберите инструмент **Прямоугольник**, выберите закрашенный тип прямоугольника.





Выберите чёрный цвет и нарисуйте сопло.

Третий костюм Кота готов! Щёлкните на нём правой кнопкой и выберите команду **дублировать** — появится четвёртый костюм. На нём мы нарисуем пламя от реактивного ранца.

Выберите инструмент **Эллипс** (эллипс означает овал).

Эллипс (Shift: Круг)



Выберите жёлтый цвет и закрашенный тип оvals.



Нарисуйте тонкий овал пламени.



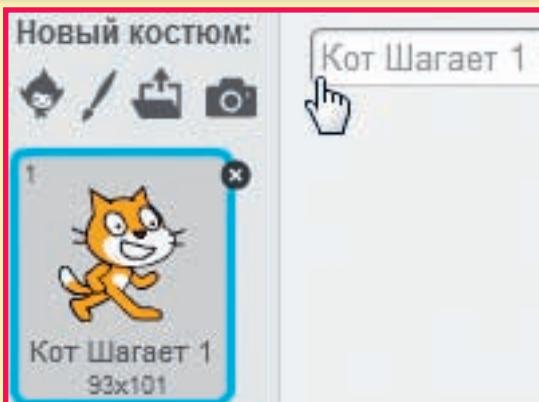


Выберите красный цвет и нарисуйте ещё один маленький овальчик около сопла.

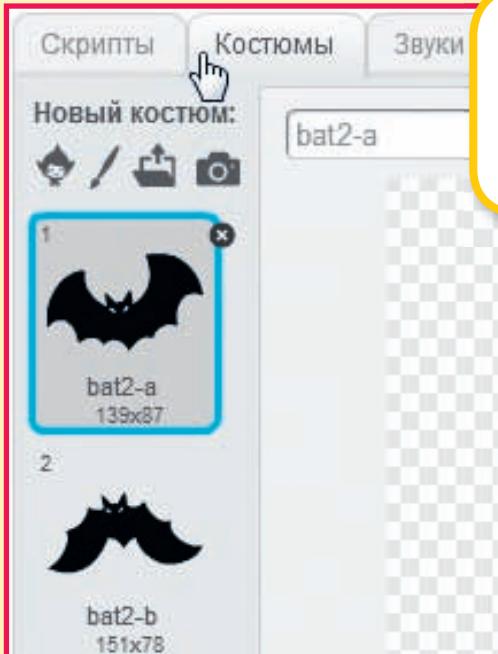
Переименуйте все костюмы.

Костюмы Кота готовы, но они имеют бессмысленные имена. Для того чтобы скрипты программы были понятней и легко читались, костюмам нужно дать осмысленные имена.

Выберите первый костюм Кота и переименуйте его, написав новое имя в поле ввода имени и нажав клавишу <Enter>. Пробелы в именах использовать можно.



## 12.2. РИСУЕМ КОСТЮМЫ ЛЕТУЧЕЙ МЫШИ



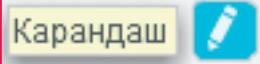
Теперь надо сделать ещё два костюма для Летучей мыши и можно начинать программировать.

Выберите Летучую мышь и перейдите на вкладку **Костюмы**.

Сейчас у нас есть два костюма нашего персонажа, которые мы используем для анимации движения. Нужно добавить ещё два костюма со слезами, ведь Летучая мышь очень расстроится, когда не сможет поймать Кота.

Дублируйте первый костюм. Выберите инструмент **Карандаш**.

Выберите голубой цвет и нарисуйте слёзы.



Дублируйте второй костюм и нарисуйте такие же слёзы.



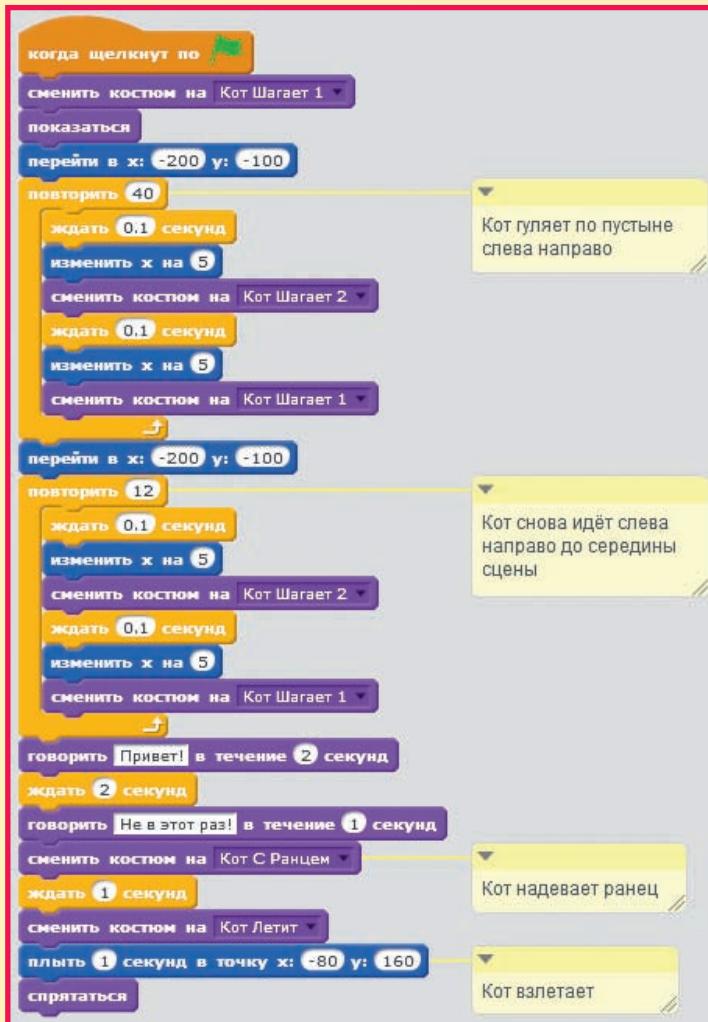
Все костюмы готовы. Осталось только переименовать их.

Первый костюм назовите «Крылья вверх», второй — «Крылья вниз», третий — «Крылья вверх плачет», четвёртый — «Крылья вниз плачет». Теперь можно начинать программировать.

# 12.3. ПРОГРАММИРУЕМ КОТА И ЛЕТУЧУЮ МЫШЬ

Сделайте вот такой скрипт для Кота.

Давайте посмотрим, как работает этот скрипт. Сначала Кот появляется и переходит в левый нижний угол сцены в точку  $(-200; -100)$ . Затем Кот 40 раз дважды изменяет X на 5 (идёт направо) и меняет костюм для анимации шагов.



После того как он дойдёт до правого края сцены, «телеопортируется» в точку  $(-200; -100)$  и продолжит анимированное движение до середины сцены. В этот момент Кот повстречает Летучую мышь и поздоровается с ней. Потом подождёт, пока она скажет свою фразу, и вежливо попрощается с Летучей мышью. После этого Кот изменяет свой костюм на костюм с реактивным ранцем, ждёт 1 секунду, запускает ранец, меняя костюм ещё раз, и уплывает в вышину, в точку  $(-80; 160)$ . После того как Кот уплывёт наверх, он скроется.

Теперь запрограммируйте Летучую мышь.

Рассмотрим, как работает этот скрипт. В начале, пока Кот гуляет по пустыне, Летучая мышь спрячется и ждёт 11 секунд. Затем она переходит в правую нижнюю часть сцены в точку с координатами (200; –100) и показывается. После этого она приближается к Коту, изменяя X на –5 и размахивая крыльями. Как только персонажи встретятся, Летучая мышь без промедления угрожает Коту расправой. После асимметричного ответа Кота ей остаётся лишь плакать и лететь за ним. Летучая мышь поднимется вверх и исчезнет.



### Совет

При создании скриптов будьте очень внимательны, не перепутайте блоки установить X и изменить X, старайтесь не потерять минусы у отрицательных чисел, не перепутайте костюмы.

Не забудьте сохранить проект.



## 12.4. ЗАДАНИЯ

1. Добавьте в мультфильм музыкальное сопровождение.
2. Запишите с микрофона фразы Кота и Летучей мыши и добавьте их в мультфильм вместо блоков говорить.
3. Найдите в Интернете звук реактивного двигателя и добавьте его в мультфильм.
4. Создайте новый спрайт с надписью «THE END» и запрограммируйте его на появление в конце мультфильма.
5. Добавьте персонажам скрипты, которые после нажатия клавиши <Пробел> отображают и Кота, и Летучую мышь и помещают каждого из них в свой угол сцены. Другими словами, запрограммируйте сброс настроек персонажей.
6. Чтобы Летучая мышь не расстраивалась и не осталась голодной, сбросьте ей тортик. Можете добавить озвучивание.
7. Сделайте свой коротенький мультфильм с двумя персонажами.

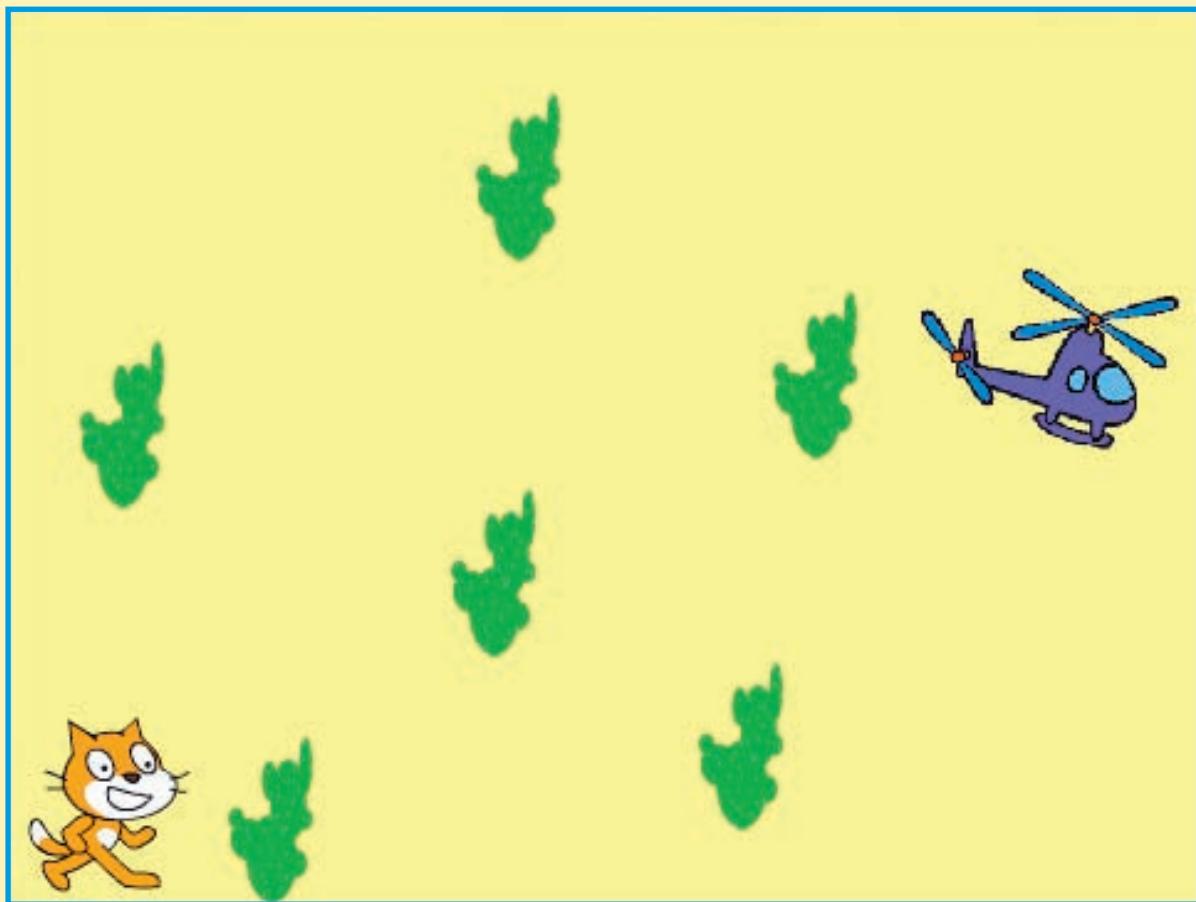


# ГЛАВА 13.

## ИГРА «ПРОЙДИ СКВОЗЬ КАКТУСЫ»

Кот включил реактивный ранец и удрал от Летучей мыши. Он летел в вышине и любовался красотой природы. Вдруг топливо закончилось, и Кот приземлился прямо в заросли кактусов. Как же ему выбраться?

Сделаем игру, в которой игроку предстоит провести Кота между кактусами, ни разу не коснувшись колючих иголок. Кот должен дойти до вертолёта, тогда он будет спасён.

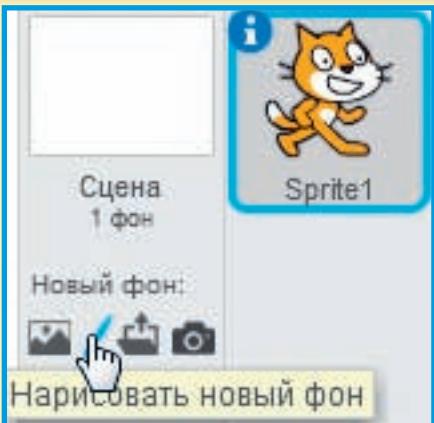




## 13.1. СОЗДАЁМ СПРАЙТЫ

Так как действие игры происходит в пустыне, то сначала закрасим сцену жёлтым цветом.

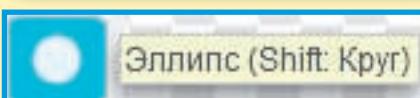
Нажмите кнопку **Нарисовать новый фон.**



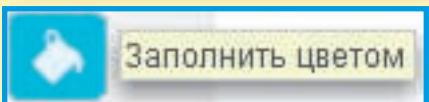
Закрасьте фон, просто щёлкнув мышью по нему.

Теперь нарисуем кактусы.  
Нажмите кнопку **Нарисовать новый спрайт.**

Выберите инструмент **Эллипс.**



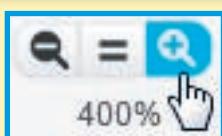
Выберите инструмент **Заполнить цветом.**



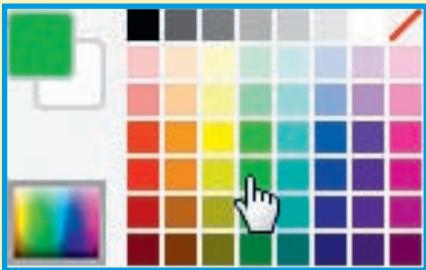
Выберите жёлтый цвет.



Увеличьте масштаб до 400%.



Выберите зелёный цвет.



Выберите закрашенный овал.



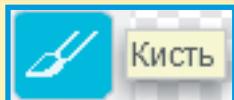
Нарисуйте кактус, состоящий из нескольких овалов разного размера.



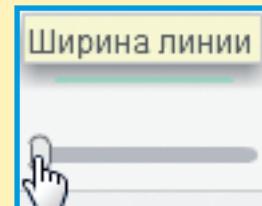
Выберите светло-зелёный цвет.



Выберите инструмент **Кисть**.

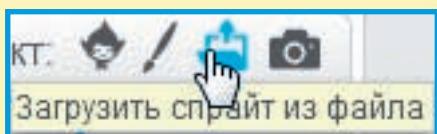


Установите минимальную ширину линии.



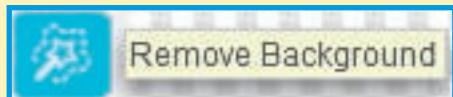
Нарисуйте точки — это будут иголочки. Кактус готов.





Если у вас получился некрасивый кактус, то вы всегда можете загрузить его изображение из Интернета, а затем добавить его в Scratch с помощью кнопки **Загрузить спрайт из файла**.

Если же загруженное изображение имеет не белый фон, то его можно удалить с помощью кнопки **Remove Background**.



Теперь добавьте из библиотеки спрайтов вертолёт.



Старайтесь искать в Интернете изображения на белом фоне, его очень просто удалить — стоит только залить его прозрачным цветом.  
Надеюсь, у вас получился красивый кактус.



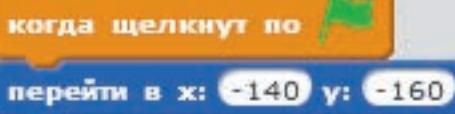
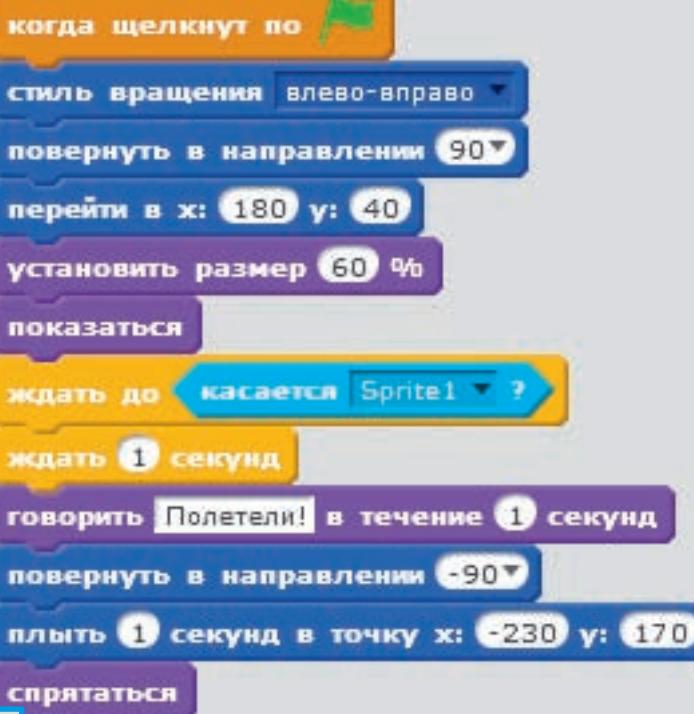
Теперь у нас есть закрашенный фон, Кот, Кактус и Вертолёт.  
Начинаем программировать!



## 13.2. ПРОГРАММИРУЕМ ПОВЕДЕНИЕ СПРАЙТОВ

Сначала запрограммируем Вертолёт.

Теперь расставим по сцене кактусы. Соберите для первого кактуса вот такой скрипт.



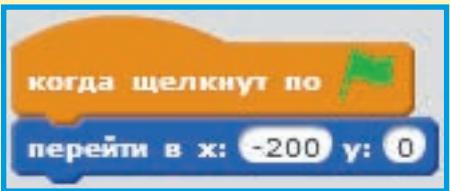
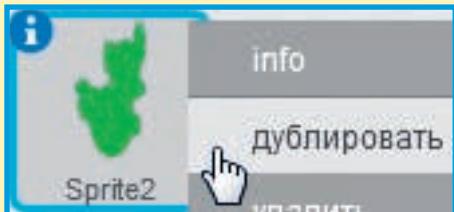
Принцип работы скрипта очень простой. Вертолёт перемещается в правую часть сцены, немного уменьшается в размере, появляется и ждёт, пока его не коснётся Котик (Sprite1). После этого он говорит фразу, поворачивается носом влево, быстро улетает в точку (-230; 170) и там скрывается.

Здесь использован новый блок ждать до. Этот блок останавливает выполнение скрипта до тех пор, пока не выполнится его условие.



Теперь продублируйте кактус.

Сделайте вот такой скрипт для второго кактуса.

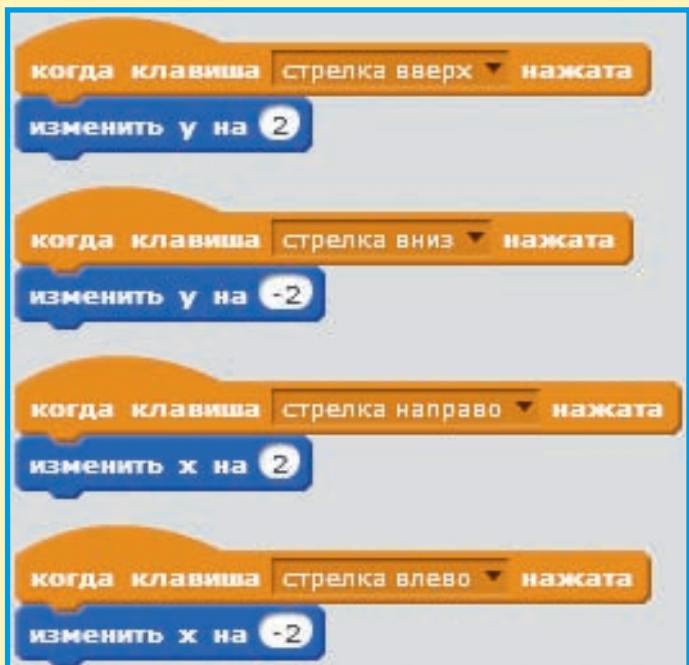


Продублируйте первый кактус ещё 4 раза и измените скрипты новых кактусов. Третий кактус должен находиться в точке  $(-50; -60)$ , четвёртый в точке  $(-40; 100)$ , пятый в точке  $(80; 20)$ , а шестой в точке  $(50; -130)$ .

Нажмите на зелёный флагок, и вы увидите, как будут расставлены кактусы.

Теперь надо запрограммировать поведение Кота. Его скрипты самые сложные, ведь ему надо взаимодействовать со всеми спрайтами!

Сначала сделайте четыре скрипта для управления Котом.





```
когда щелкнут по флагу
перейти в x: -200 y: -140
показаться
всегда
если касается Sprite2 ?, то
    говорить Ай! в течение 1 секунд
    перейти в x: -200 y: -140
если касается Sprite3 ?, то
    говорить Ай! в течение 1 секунд
    перейти в x: -200 y: -140
если касается Sprite4 ?, то
    говорить Ай! в течение 1 секунд
    перейти в x: -200 y: -140
если касается Sprite5 ?, то
    говорить Ай! в течение 1 секунд
    перейти в x: -200 y: -140
если касается Sprite6 ?, то
    говорить Ай! в течение 1 секунд
    перейти в x: -200 y: -140
если касается Sprite7 ?, то
    говорить Ай! в течение 1 секунд
    перейти в x: -200 y: -140
если касается Helicopter ?, то
    говорить Ура! в течение 1 секунд
    спрятаться
```

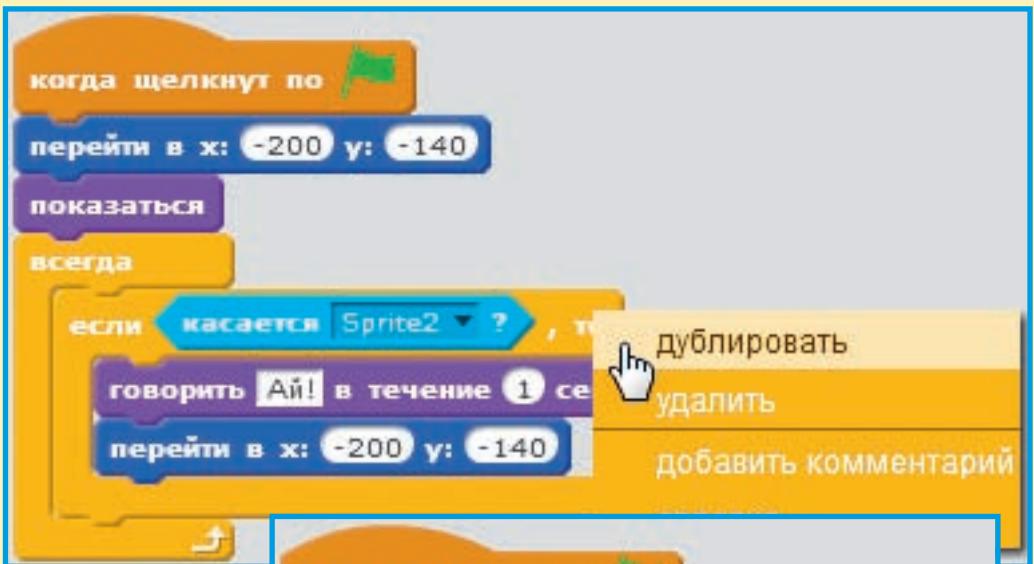
А теперь добавьте скрипт, описывающий взаимодействия с кактусами и Вертолётом.

Скрипт очень большой, но состоит из шести похожих блоков если, описывающих касание кактусов. При касании любого кактуса Кот скажет «Ай!» и перейдёт в начальную точку.



### Совет

Используйте команду **дублировать** из меню, которое появляется после щелчка правой кнопкой мыши, чтобы копировать блоки если.



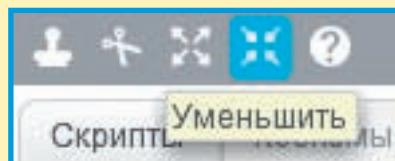
Вставляйте дублированный блок в блок всегда и изменяйте имя спрайта в голубом сенсорном блоке.





При касании вертолёта Кот скажет «Ура!» и спрячется (как будто сядет в Вертолёт).

Игра готова. Запустите её и попробуйте пройти к Вертолёту. Если кактусы получились очень большими, то уменьшите их с помощью кнопки Уменьшить.



Сохраните игру.

### 13.3. ЗАДАНИЯ

1. Увеличьте скорость движения Кота.
2. Добавьте ещё два кактуса и доработайте скрипт Кота.
3. Сделайте так, чтобы Кот управлялся мышью.
4. Замедлите полёт Вертолёта в 3 раза.
5. Найдите в Интернете звук мотора и добавьте этот звук летящему Вертолёту.
6. Найдите в Интернете изображение вертолёта и замените костюм Вертолёта.
7. Обратите внимание, что после остановки игры Кот и Вертолёт скрыты. Добавьте для этих персонажей скрипты, которые по нажатию клавиши <Пробел> возвращают и того, и другого на сцену и отображают их в левом нижнем углу (Кот) и правом верхнем углу (Вертолёт).





# ГЛАВА 14. ИГРА «ВЕДЬМА И ВОЛШЕБНИК»

Давным-давно, когда Волшебник только завершил эксперименты и создал первую в мире летающую метлу, в дверь его дома постучали. Погода была ужасная, и Волшебник не смог отказать одинокой путнице. Он пустил её в дом, накормил ужином, позволил обсохнуть и отдохнуть. Наутро женщина пропала, летающей метлы тоже не оказалось на месте. Волшебник понял, что это была Ведьма, которая давно мечтала о новом средстве передвижения. Он поклялся отомстить.

Давайте сделаем игру, в которой Волшебник повстречает Ведьму и отомстит ей. Всего у Волшебника есть 5 склянок с волшебными зельями. Ведьма будет летать, но это ей не поможет.

## 14.1. СОЗДАЁМ СПРАЙТЫ

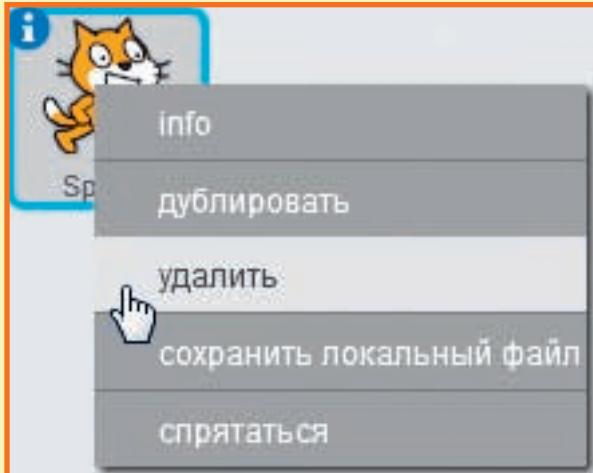
Встреча Волшебника и Ведьмы будет происходить в магическом лесу. Выберите его из библиотеки фонов.





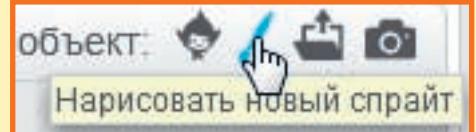
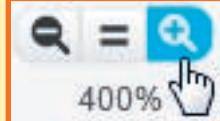
Кота в нашей истории не будет. Удалите его.

Добавьте Волшебника и Ведьму из библиотеки спрайтов.



Теперь у нас есть лес и главные герои. Нужно нарисовать склянки с зельями.  
Нажмите кнопку **Нарисовать новый спрайт**.

Увеличьте масштаб до 400%.

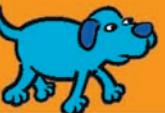


Нажмите кнопку **Конвертировать в векторную графику** (эта кнопка находится в правом нижнем углу экрана). В векторном режиме проще рисовать сложные объекты. (Надеюсь, вы знаете, что такое векторная и раcтровая графика.)

Растровый режим  
**Конвертировать в векторную графику**

Выберите инструмент **Эллипс**.

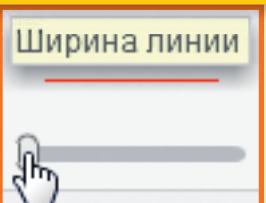




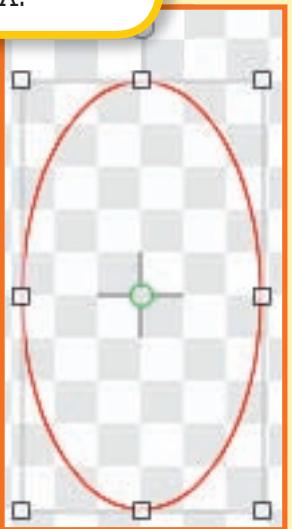
Выберите красный цвет.



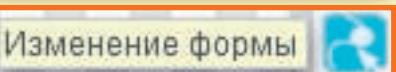
Установите минимальную ширину линии.



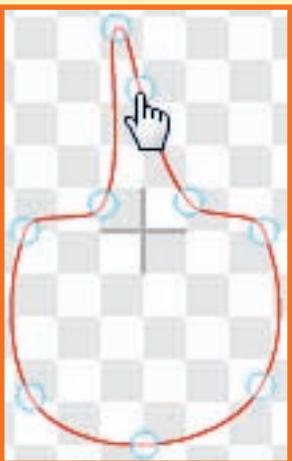
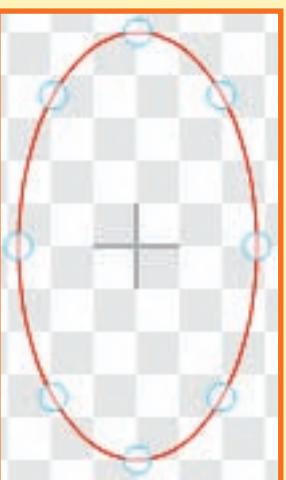
Нарисуйте овал.



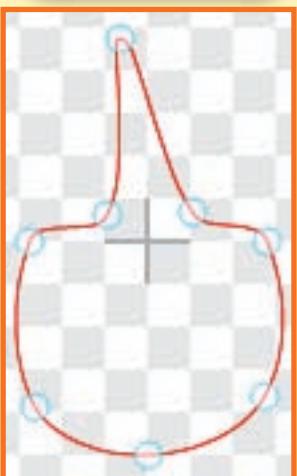
Выберите инструмент **Изменение формы**.

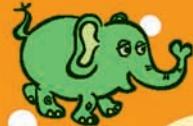


Теперь овал покрыт опорными точками. Перемещая эти точки, можно изменять форму объекта.

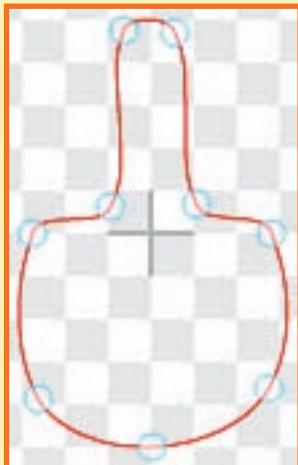


Если щёлкнуть на линии между опорными точками, то в этом месте появится новая опорная точка.





Нарисуйте горлышко склянки.



Теперь выберите инструмент **Окрасить форму**.

Окрасить форму



Переключите цвета.



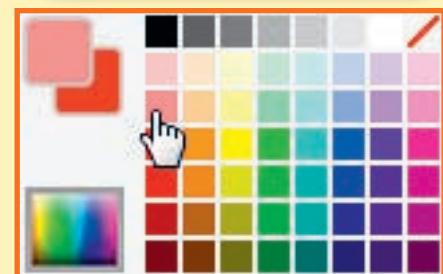
Задайте вот такой способ заливки.



Теперь надо нарисовать пробку. Выберите инструмент **Прямоугольник**.

Прямоугольник (Shift: Квадрат)

Вместо белого цвета выберите розовый.



Окрасьте склянку.





Щёлкните в палитре по чёрному цвету и нарисуйте прямоугольничек.



Выберите инструмент  
**Изменение формы**.



Первая склянка готова. Продублируйте её 4 раза и перекрасьте новые склянки с помощью инструмента **Окрасить форму**. Должно получиться 5 магических склянок.





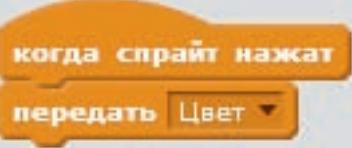
Разместите арсенал Волшебника в левой части экрана.

Все спрайты готовы, можно начинать программировать!



## 14.2. ПРОГРАММИРУЕМ СПРАЙТЫ

Сделайте вот такую программу для красной склянки.



Здесь использованы сразу два новых блока: когда спрайт нажат и передать. Как вы, наверное, догадались, блок когда спрайт



нажат позволяет выполнить то, что прицеплено к нему в том случае, когда щёлкнули по спрайту.

Блок передать осуществляет передачу сообщения с указанным именем.

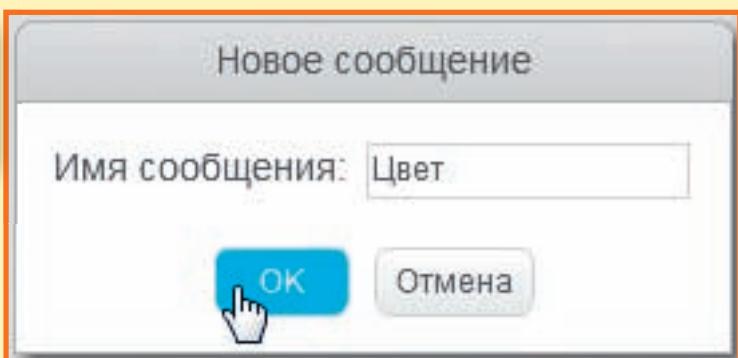
В нашем случае при щелчке на красной склянке будет передано сообщение с именем «Цвет». Сообщения нужны для того, чтобы остальные спрайты проекта понимали, что происходит, и реагировали, как задумано. В нашем случае на сообщения будут реагировать Ведьма и Волшебник. Волшебник будет произносить фразу, а Ведьма станет магически видоизменяться.



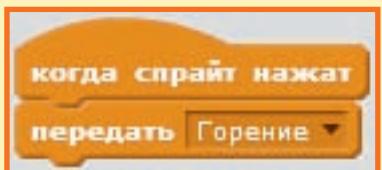
### Совет

Для того чтобы ввести новое сообщение, необходимо раскрыть выпадающий список и выбрать команду **новое сообщение**.

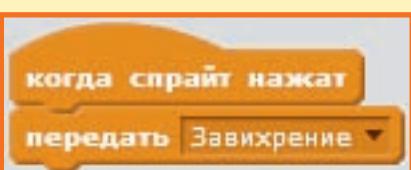
В окне ввода надо ввести имя сообщения и нажать кнопку **OK**.

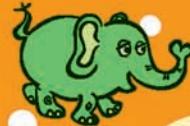


Для синей склянки сделайте вот такую программу.



Для зелёной склянки вот такую.





Для фиолетовой вот такую.

когда спрайт нажат  
передать Вздуйся

Для жёлтой вот такую.

когда спрайт нажат  
передать Окаменей

когда щелкнут по   
перейти в x: -140 у: -120

когда я получу Цвет  
говорить Глотни компотика! в течение 2 секунд

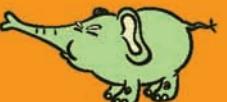
когда я получу Горение  
говорить Сгори, Ведьма! в течение 2 секунд

когда я получу Завихрение  
говорить Кружись, пока нос не отвалится! в течение 2 секунд

когда я получу Вздуйся  
говорить Чтоб ты лопнула! в течение 2 секунд

когда я получу Окаменей  
говорить Окаменей! в течение 2 секунд

Теперь  
запрограммируйте  
Волшебника. У него  
будет 6 скриптов.

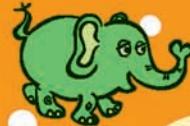


Соберите программу для Ведьмы. Она состоит из шести скриптов.

The image shows a Scratch script for the Witch character. It consists of six separate scripts, each triggered by a different sensor:

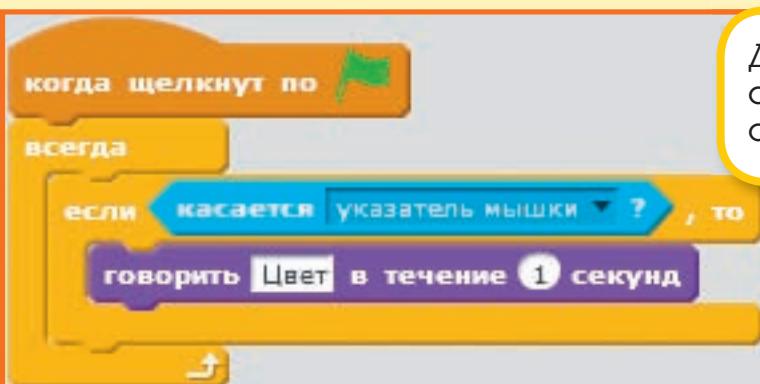
- Top Script:** Triggered by clicking a green flag. It includes:
  - Style: Turn Left-Right
  - Move to x: 150 y: -100
  - Turn 90 degrees
  - Say "Привет, Дедуля!" for 2 seconds
  - Repeat 100 [Fly for 1 second at point x: (random -80 to 220) y: (random -160 to 160) + Stop all scripts for [all sprites]]
- Color Sensor Script:** Triggered by receiving a "Color" message. It includes:
  - Wait 1 second
  - Repeat 500 [Change color effect by 2]
- Turbulence Sensor Script:** Triggered by receiving a "Turbulence" message. It includes:
  - Wait 1 second
  - Repeat 500 [Change turbulence effect by 2]
- Fire Sensor Script:** Triggered by receiving a "Fire" message. It includes:
  - Wait 1 second
  - Repeat 50 [Change fire effect by 2]
- Stone Sensor Script:** Triggered by receiving a "Stone" message. It includes:
  - Wait 1 second
  - Repeat 200 [Change stone effect by 2]
- Wind Sensor Script:** Triggered by receiving a "Wind" message. It includes:
  - Wait 1 second
  - Repeat 500 [Change wind effect by 2]

Для перемещения Ведьмы в случайном направлении мы использовали новый блок выдать случайное. Он каждый раз выдаёт новое значение из указанного диапазона. В нашем случае этот блок выдаёт значение для  $X$  из диапазона от  $-80$  до  $220$  и для  $Y$  от  $-160$  до  $160$ . Это сделано для того, чтобы Ведьма летела каждый раз в новую точку.

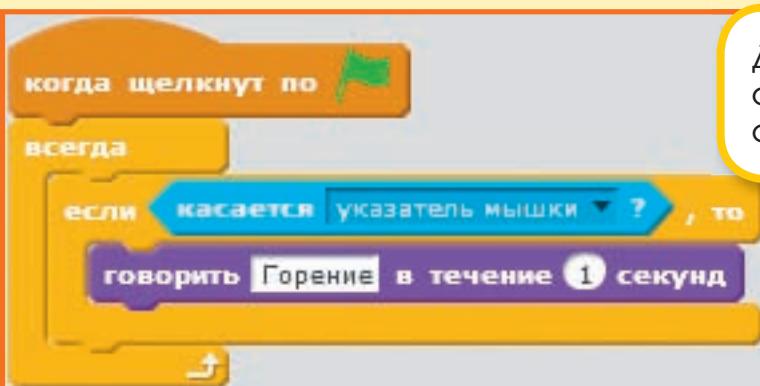


## 14.3. ВСПЛЫВАЮЩИЕ ПОДСКАЗКИ

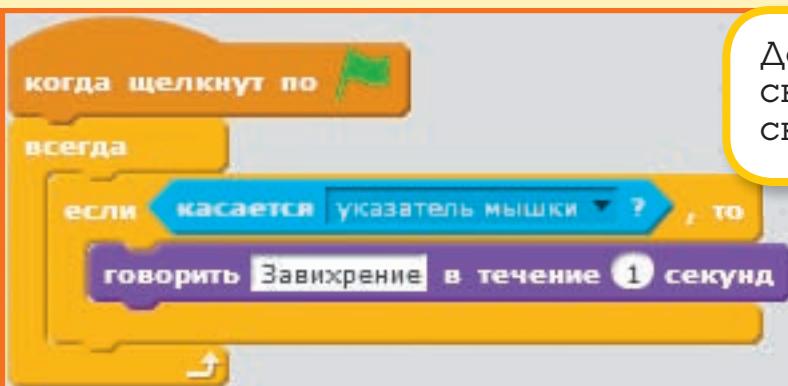
Я думаю, будет хорошей идеей добавить каждой склянке всплывающую подсказку, чтобы при наведении на неё курсора мыши склянка сообщала нам о своём содержимом. Так Волшебнику будет проще справиться с противником.



Добавьте красной склянке вот такой скрипт.

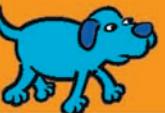


Добавьте синей склянке вот такой скрипт.



Добавьте зелёной склянке вот такой скрипт.





```

когда щелкнут по флагу
всегда
если касается указатель мыши ? , то
    говорить Ведущие в течение 1 секунд

```

Добавьте фиолетовой склянке вот такой скрипт.

```

когда щелкнут по флагу
всегда
если касается указатель мыши ? , то
    говорить Окаменеть в течение 1 секунд

```

Добавьте жёлтой склянке вот такой скрипт.

Проект готов, протестируйте его и сохраните.

## 14.4. ЗАДАНИЯ

1. Замедлите перемещение Ведьмы.
2. Сделайте новое зелье, которое будет накладывать на Ведьму сразу несколько эффектов.
3. Сделайте несколько зелий для Ведьмы, чтобы она могла отбиваться от Волшебника.
4. Сделайте так, чтобы при касании Волшебника Ведьма произносила какую-нибудь фразу. Можете попросить друга или родителей записать реплику Волшебника. Тогда у вас в игре персонажи смогут говорить разными голосами.
5. Сделайте так, чтобы при нажатии на склянку она на некоторое время становилась ярче, а потом снова возвращалась в обычное состояние.



# ГЛАВА 15. КОТ-МАТЕМАТИК

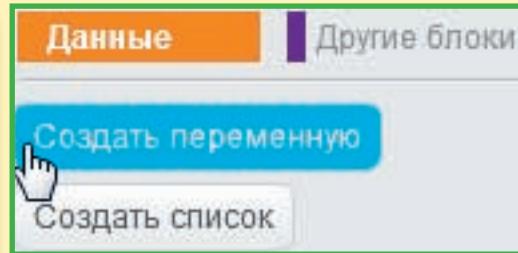
## 15.1. ПЕРЕМЕННЫЕ

Жил да был очень умный Котик. Он жил на рынке рядом с прилавком, на котором продавали рыбу. Весь день он наблюдал за торговлей и научился считать.

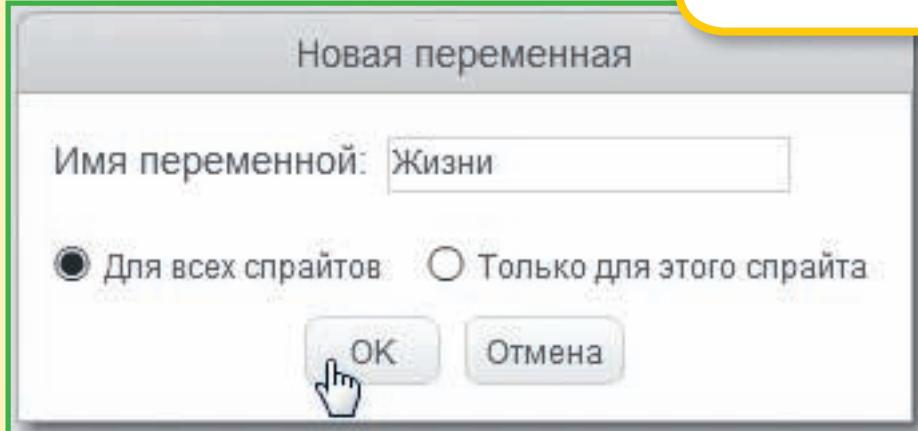
В этой главе мы сделаем проект про Кота-математика. Кот будет очень грамотным и сможет складывать числа.

При создании этого проекта нам понадобятся переменные. Переменные в Scratch — это такие овальчики, которые хранят в себе некоторые значения.

Для того чтобы создать переменную, нужно выбрать оранжевые блоки **Данные** и нажать кнопку **Создать переменную**.

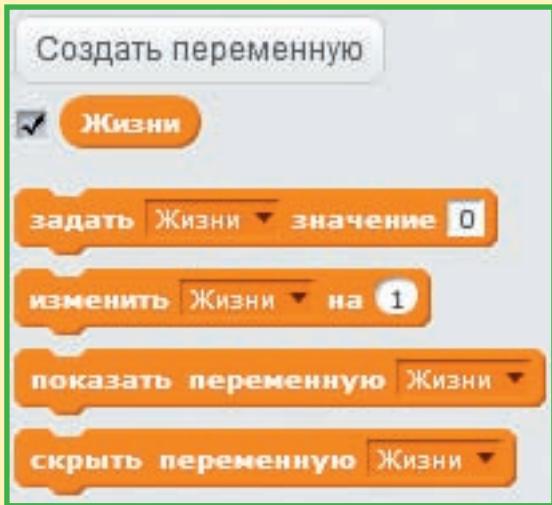


Необходимо ввести имя переменной, например **Жизни**, и нажать кнопку **OK**.





Появится овальная  
переменная и  
оранжевые блоки.

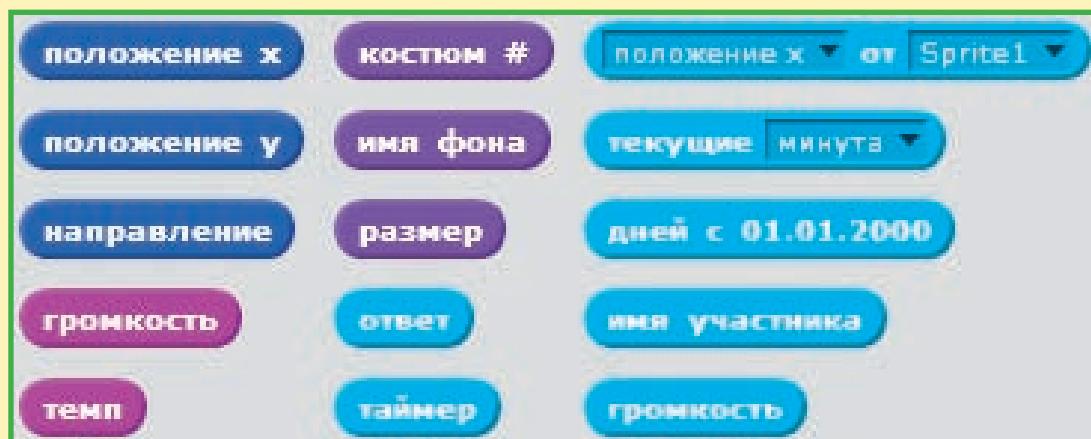


Блок задать значение позволяет записать в переменную значение. Оно может быть числом или словом. Если переменная хранит числовое значение, то блок изменить может изменить значение переменной. Блоки показать переменную и скрыть переменную делают то же самое, что и маленькая галочка рядом с именем переменной — отображают или скрывают значение переменной на сцене. Это выглядит вот так.





В Scratch есть два вида переменных: пользовательские и защищённые. Пользовательские переменные оранжевого цвета, вы можете создавать их, удалять, можете присваивать им любые значения. Защищённые переменные — это переменные разных цветов, изменить значение которых вы не можете. Эти переменные хранят информацию, которую получают автоматически. Вот все защищённые переменные Scratch.



Синие защищённые переменные хранят информацию о координатах спрайта и его направлении; фиолетовые — порядковый номер костюма, имя фона и размер спрайта; сиреневые переменные хранят громкость звука и темп мелодии; голубые — ваши ответы, которые вы будете вводить, время и положение курсора.

## 15.2. КОНСТРУИРУЕМ ИГРУ

Теперь, когда вы познакомились с переменными, можно сделать игру про Кота-математика. Сначала создайте две переменные: **Первое слагаемое** и **Второе слагаемое**.



Соберите для Кота вот такой скрипт.



Первое слагаемое 0  
Второе слагаемое 0

Здесь использован новый блок спросить и ждать. Этот блок отображает на сцене поле для ввода информации. Вы должны ввести в это поле первое слагаемое и нажать клавишу <Enter>.

Введите первое слагаемое.



|



Значение, введённое в поле ввода, попадает прямо в голубую защищённую переменную **ответ**. С помощью блока задать значение значение переменной **ответ** будет записано в переменную **Первое слагаемое**.

Первое слагаемое 0

Второе слагаемое 0

Ведите первое слагаемое.



|





Следующее значение, введённое в поле ввода, попадёт в переменную **ответ**, и сразу после этого будет записано в переменную **Второе слагаемое**.

Первое слагаемое

10

Второе слагаемое

0

Ведите второе  
слагаемое.



10



Первое слагаемое

10

Второе слагаемое

5

15



Котик сообщает сумму слагаемых. Какой молодец! Для получения ответа он использовал блок сложения чисел.



В каждое из полей этого блока вставляется какая-либо переменная. В нашем случае в поля вставлены слагаемые. С помощью блока говорить Кот произносит сумму слагаемых.

Сохраните проект.

## 15.3. ЗАДАНИЯ

1. Измените проект так, чтобы Кот умножал введённые значения.
2. Измените проект так, чтобы Кот находил сумму трёх чисел.
3. Добавьте в проект банкира, который будет переводить введённую сумму в рублях в пиастры по курсу 75 рублей за пиастр. (Надеюсь, вы читали роман Роберта Льюиса Стивенсона «Остров сокровищ»? У пиратов там были пиастры.)



4. Добавьте в проект заправщика, который будет спрашивать, сколько литров бензина залить, и говорить стоимость при цене 35 рублей за литр бензина.

5. Добавьте в проект автомобиль, который потребляет 10 литров бензина на 100 километров пути. Пускай этот автомобиль спрашивает расстояние, которое вы собираетесь проехать, и сообщает о затратах на бензин при цене 35 рублей за литр.

## 15.4. ОТГАДАЙ ЧИСЛО

Давайте сделаем ещё один математический проект — игру про отгадывание чисел. Кот будет загадывать число от 1 до 10, а задача игрока — угадать, какое число загадано. Всего 10 попыток.

**Создайте**  
новый проект и  
добавьте в него  
две переменные:  
**Очки** и **Загадал**.

Снимите галочку с переменной  
**Загадал**, и она  
пропадёт со  
сцены.

Соберите вот такую  
программу для Кота.

**Создать переменную**

- Загадал**
- Очки**

```

когда щелкнут по [запустить]
    задать [Очки] значение 0
    говорить [Отгадайте 10 чисел от 1 до 10, в течение 5 секунд]
    повторить [10]
        задать [Загадал] значение [выдать случайное от 1 до 10]
        спросить [Какое число я загадал?] и ждать
        если [ответ = Загадал] то
            изменить [Очки] на 1
        говорить [А теперь?] в течение 1 секунд
        говорить [Вы набрали] в течение 2 секунд
        говорить [Очки] в течение 2 секунд
        говорить [очков,] в течение 2 секунд
        если [Очки > 3] то
            говорить [Вы экстрасенс!] в течение 2 секунд
    конец
конец

```



В начале работы программы переменная **Очки** будет обнулена. Затем Кот скажет: «Отгадайте 10 чисел от 1 до 10». Потом 10 раз повторится цикл загадки. Сначала переменной **Загадал** будет задано случайное значение с помощью блока выдать случайное. Случайное значение будет выбрано из диапазона от 1 до 10. Затем Кот спросит: «Какое число я загадал?» Ваш ответ будет помещён в защищённую переменную **ответ**. Блок если сравнивает переменные **ответ** и **Загадал**. Если они равны, то переменная **Очки** увеличится на единицу, и Кот предложит отгадывать следующее число. После 10 проходов цикла Кот скажет, сколько очков вы набрали, и если вы набрали больше 3 очков, то вы экстрасенс! У меня не получается угадать больше одного числа.

Сохраните проект.

## 15.5. ВИДЫ ОТОБРАЖЕНИЯ ПЕРЕМЕННЫХ

Если напротив переменной установить галочку, то она будет отображаться на сцене вот в таком виде.



Если дважды щёлкнуть на этой переменной, то она изменит внешний вид и станет крупнее.

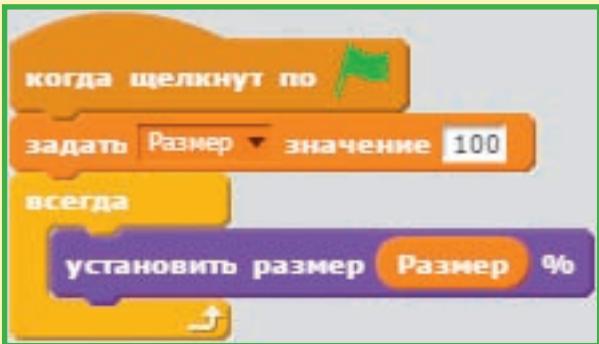


Если ещё раз дважды щёлкнуть по ней, то под значением переменной появится **рычажок**, с помощью которого можно изменять её значение.





Сделайте вот такой скрипт и посмотрите, как работает рычажок.



Перемещайте рычажок, и вы увидите, как будет изменяться размер Котика.



## 15.6. ЗАДАНИЯ

- Добавьте ещё одну переменную — **Номер попытки**. Пусть в ней отображается значение текущей попытки.
- Сыграйте в игру 10 раз и запишите набранные очки на бумаге. Измените диапазон случайных чисел от 1 до 5. Снова запишите ответы на бумаге. Почему теперь вы можете набрать больше очков?
- Сделайте так, чтобы Кот спрашивал не случайное число, а примеры из таблицы умножения. Для этого вам понадобятся ещё две переменные: **Слагаемое1** и **Слагаемое2**.





## ГЛАВА 16. ИГРА «ВЕРТОЛЁТ»

Смелый Кот прошёл сквозь кактусовую поляну и полетел на вертолёте. Однако Летучая мышь заметила его и позвала своих друзей на охоту. На Вертолёт нападают 35 Летучих мышей. У него есть 3 жизни — он может всего лишь 3 раза столкнуться с врагом. Цель игры — набрать 30 очков.



### 16.1. СОЗДАЁМ СПРАЙТЫ И ФОН

Так как дело происходит высоко в небе, то закрасьте фон голубым цветом. Можете нарисовать облака и солнце.

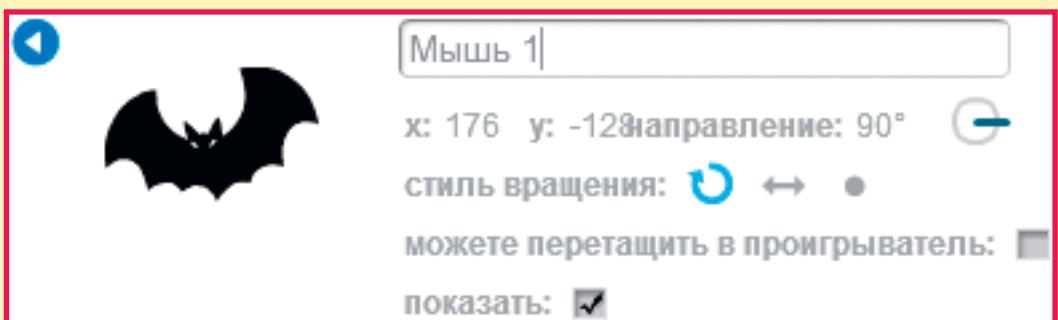




Спрайт Кота удалите.  
Добавьте спрайт  
Вертолёта и спрайт  
**Летучей мыши.**



Переименуйте спрайт Летучей мыши. Щёлкните на синем кружке с буквой «i», откроется окно свойств спрайта. Введите новое имя спрайта.



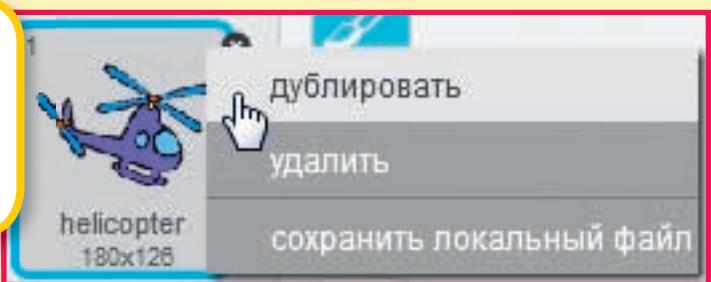
### Совет

Окно свойств спрайта можно использовать не только для переименования спрайта, но и для изменения его видимости, стиля вращения и направления.

Сначала мы запрограммируем одну Летучую мышь, а потом просто продублируем её. Новые летучие мыши будут содержать такие же спрайты, как и у первой мыши, ведь их поведение должно быть одинаковым.

У Вертолёта должно быть 4 костюма с разной степенью повреждений.

Выберите спрайт  
Вертолёта, перейдите  
на вкладку **Костюмы** и  
продублируйте первый  
костюм Вертолёта.





Инструментом **Кисть** нарисуйте небольшие повреждения Вертолёта.



Дублируйте второй костюм и добавьте ещё больше повреждений. Можете нарисовать белый или серый дым.



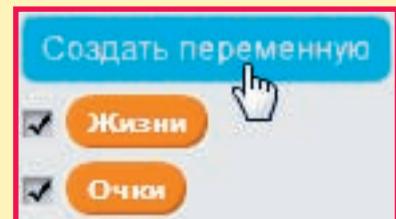
Дублируйте третий костюм Вертолёта и добавьте суперповреждения.



Спрайты готовы, можно программировать!

## 16.2. ПРОГРАММИРУЕМ СПРАЙТЫ

Сначала создайте две переменные: **Очки** и **Жизни**. Переменная **Очки** будет вести подсчёт очков до достижения 30, а переменная **Жизни** будет показывать оставшиеся жизни Вертолёта.





Сделайте 3 скрипта для Вертолёта.  
Первый скрипт такой.

Первый скрипт устанавливает размер Вертолёта в 2 раза меньше обычного, «одевает» на него первый костюм и делает Вертолёт видимым. Далее задаёт начальные значения переменных, перемещает Вертолёт в левый нижний угол сцены, в точку (-190; -150), а затем постоянно устанавливает координату Y Вертолёта в координату Y мыши. В результате вертикальное положение Вертолёта задаётся движением мыши. В горизонтальном направлении Вертолёт не двигается.

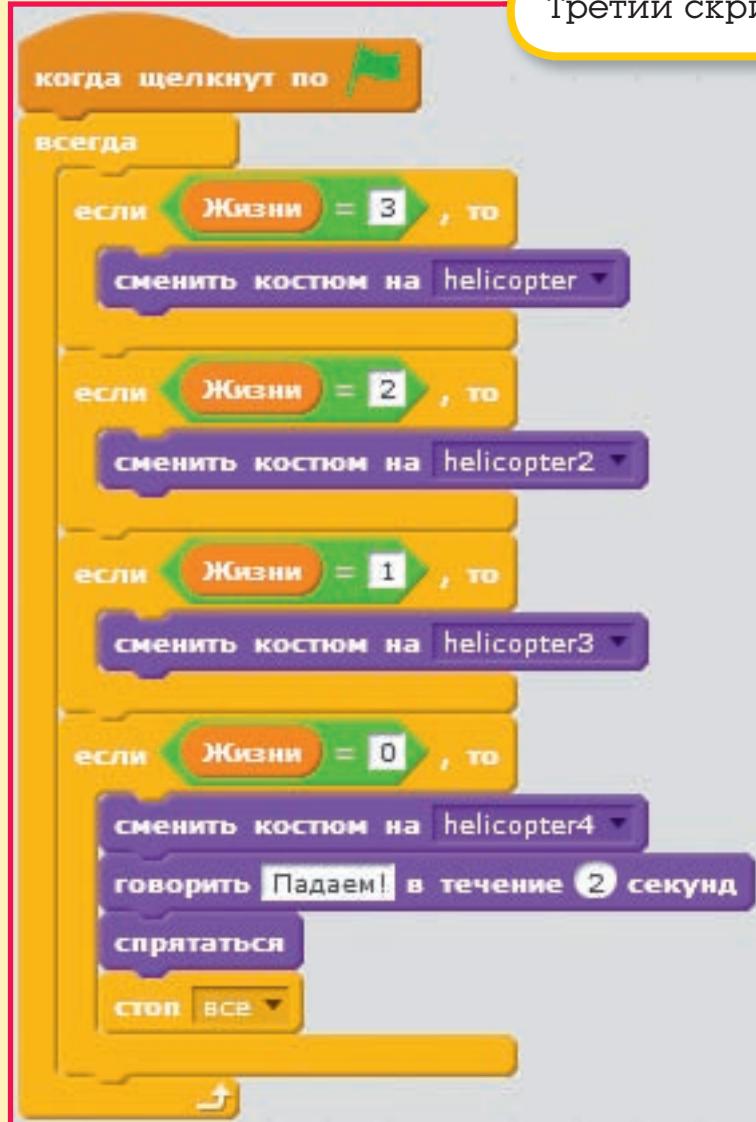
```
когда щелкнут по флагу
установить размер на 50%
сменить костюм на helicopter
показаться
задать [Очки] значение 0
задать [Жизни] значение 3
перейти в x: -190 y: -150
всегда
установить y в [мышка по y]
```

Теперь второй скрипт.

```
когда щелкнут по флагу
всегда
если [Очки] = 30, то
говорить [Победа!] в течение 2 секунд
```

Это скрипт победы. Всё время (блок всегда) проверяется, равна ли переменная **Очки** значению 30. Когда это случится, Вертолёт закричит: «Победа!»

Третий скрипт такой.



Третий скрипт Вертолёта изменяет его костюм в зависимости от количества оставшихся жизней. Если у Вертолёта 3 жизни, то он без повреждений; если 2 жизни, то он немного «помят»; если 1, то он еле летит; а если 0, то он падает и останавливает выполнение программы.



```
когда щелкнут по флагу
установить размер на 50 %
перейти в x: 200 у: выдать случайное от -180 до 180
ждать 1 секунд
всегда
изменить x на -3
```

Теперь запрограммируйте Летучую мышь, у неё 4 скрипта. Первый скрипт такой.

```
когда щелкнут по флагу
всегда
если касается край ? , то
изменить очки на 1
установить x в 200
установить у в выдать случайное от -180 до 180
```

Второй скрипт такой.

Второй скрипт всегда ждёт, пока Летучая мышь не коснётся левого края сцены. В этом случае он увеличивает переменную **Очки** на 1 и снова «телеопортирует» Летучую мышь в случайную точку к правой границе сцены.





когда щелкнут по

всегда

если касается Helicopter ? , то

изменить Жизни на -1

ждать 2 секунд

Теперь соберите третий скрипт.

Этот скрипт всегда ждёт, пока Летучая мышь не коснётся Вертолёта. Когда это произойдёт, переменная **Жизнь** уменьшается на 1. Двухсекундная задержка нужна для того, чтобы за одно касание отнималась только одна жизнь.



### Эксперимент

Удалите блок ждать и посмотрите, как будет изменяться переменная **Жизни** при соприкосновении Вертолёта и Летучей мыши.

Это скрипт с условием победы, он следит за значением переменной **Очки**. Если её значение сравняется с 30, то Летучие мыши заносят: «Опять не поймали!»

Протестируйте работу программы.

Играть с одним противником очень легко и неинтересно. Давайте добавим ещё немного Летучих мышей, чтобы их стало 5.



Дублируйте Летучую мышь 4 раза.



Измените в первом скрипте каждой Летучей мыши время задержки в блоке ждать. У первой мыши должна быть 1 секунда, у второй — 2 секунды, и так далее. Теперь они будут стартовать по очереди, а не всей гурьбой.

Протестируйте игру и сохраните её.

## 16.3. ЗАДАНИЯ

1. Сделайте так, чтобы Летучие мыши махали крыльями в полёте.
2. Сделайте так, чтобы при касании Вертолёта Летучая мышь на 0.1 секунды изменяла свою яркость.
3. Добавьте Вертолёту ещё две жизни и два костюма.
4. Ускорьте полёт Летучих мышей.
5. Добавьте ещё две Летучие мыши.
6. Сделайте так, чтобы после гибели Вертолёта появлялся Кот и спускался вниз на парашюте.





## ГЛАВА 17.

# ПОЛЁТ С УСКОРЕНИЕМ «ФЛЭППИ БЁРД»

## 17.1. СОЗДАЁМ СПРАЙТЫ И ФОН

Давайте сделаем свою версию популярной игры «Флэппи Бёрд». Как вы помните, задача птички — пролететь между труб и набрать как можно больше очков.



Сначала нарисуйте фон. Он может быть любым. Я нарисовал лето, поле с цветами и облака. Вы можете поместить





Птичку в более экстремальные условия. Включите фантазию!

Для запуска игры мы будем использовать спрайт Play (произносится как «плэй») в виде кнопки. Нажмите кнопку **Нарисовать новый спрайт**. Затем нажмите кнопку **Конвертировать в векторную графику**, которая находится в правом нижнем углу экрана. После этого будет удобнее рисовать.

Нарисуйте кнопку в виде оранжевого прямоугольника. Затем выберите инструмент **Текст**.



Напишите на кнопке слово «Play».

Play

Разместите кнопку в правой части сцены.

Теперь нарисуйте Птичку. Это персонаж игры, и поэтому надо нажать кнопку **Нарисовать новый спрайт**. Затем переключитесь в режим растровой графики, нажав кнопку **Конвертировать в растровую графику**. У Птички должно быть 2 костюма для анимации крыльев.

Вот первый костюм.



Вот второй костюм.

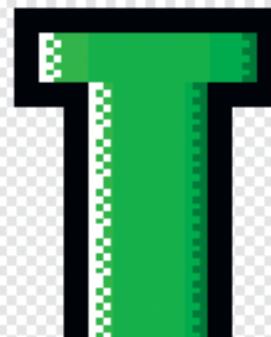
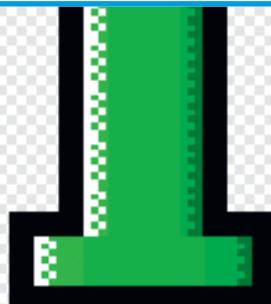


Как видите, они отличаются только формой крыльев.



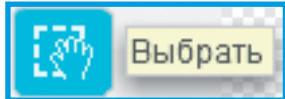
Теперь создайте Трубы. У них будет 5 разных костюмов.

Вот так должен выглядеть первый костюм.

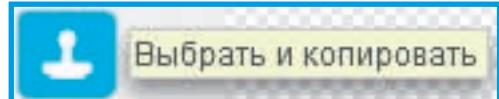


Для создания следующих костюмов дублируйте первый костюм и просто переместите трубы выше или ниже.

Для перемещения используйте инструмент **Выбрать**.

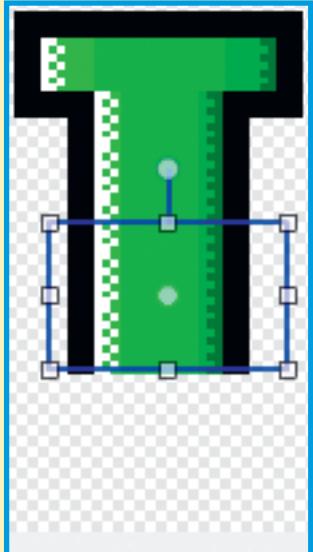


Для того чтобы удлинить часть трубы, выберите инструмент **Выбрать и копировать**.

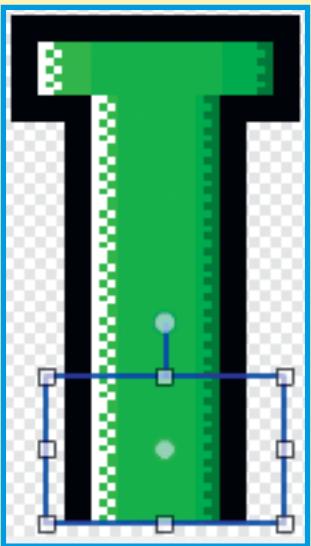




Выделите  
часть  
трубы.



Выделенная  
область  
автоматически  
скопируется.  
Останется только  
переместить её  
на новое место.



Создайте пять вот  
таких костюмов.

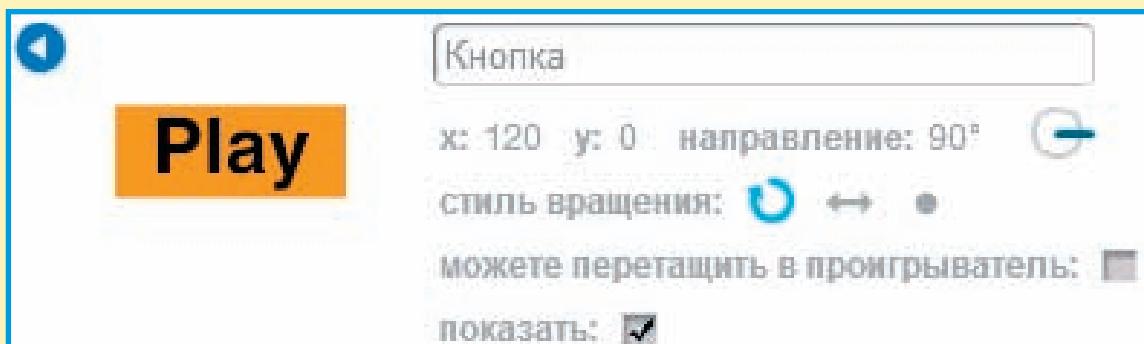
1		Костюм1 110x360
2		Костюм2 108x360
3		Костюм4 108x360
4		Костюм3 108x360
5		Костюм5 108x360

Все спрайты  
готовы, но они  
имеют странные  
имена: Sprite1,  
Sprite2 и Sprite3.



Давайте дадим им нормальные имена. Щёлкните на синем кружке с буквой «i», откроется окно свойств спрайта.

Введите новое имя спрайта.



Переименуйте все спрайты.



Теперь можно начинать программировать!

## 17.2. ПРОГРАММИРУЕМ ПОВЕДЕНИЕ СПРАЙТОВ

Сначала запрограммируйте кнопку Play.

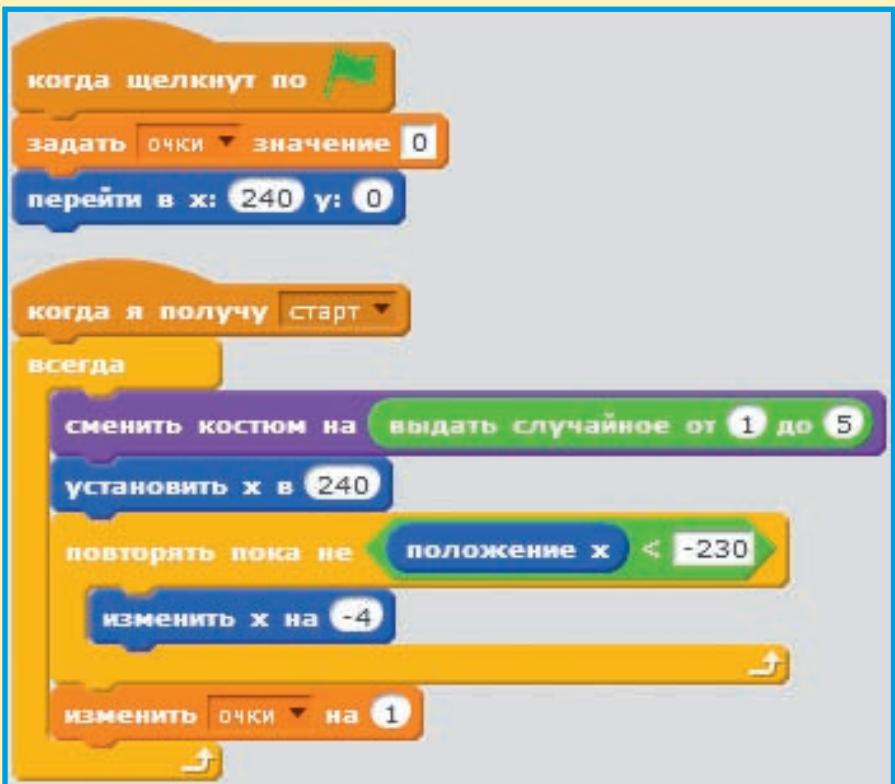
У неё очень простая программа. При запуске игры она появится в правой части экрана в точке (120; 0), а затем при



нажатии на неё спрячется и передаст сообщение «старт».

Теперь сделайте вот такую программу для Труб.

Сначала создайте переменную **очки**, она будет хранить количество очков, набранное игроком.





Основной скрипт Труб запускается после получения сообщения «старт». Сначала Трубы изменяют свой костюм на один из пяти, случайно выбранный с помощью блока выдать случайное. Потом Трубы перемещаются на правую границу сцены, где  $X = 240$ . Затем Трубы медленно двигаются налево, в цикле изменяя свою координату  $X$  на  $-4$ . Здесь использован новый блок цикла с условием повторять пока не. Этот блок повторяет своё содержимое до тех пор, пока не будет выполнено условие. В нашем случае он будет перемещать Трубы влево до тех пор, пока они не окажутся на левом краю сцены в точке, где  $X$  меньше  $-230$ . После того как Трубы «доплынут» до левой границы сцены, выполнение цикла будет завершено, и переменная **очки** увеличит своё значение на 1.

Теперь соберите скрипты для Птички. Сначала создайте ещё одну переменную — скорость. Она будет отвечать за скорость вертикального перемещения Птички (то есть вверх и вниз). Вправо и влево Птичка перемещаться не будет.

В начале работы программы Птичка перемещается в точку  $(-100; 0)$  и ждёт, когда игрок нажмёт кнопку **Play**. После нажатия **Play** и получения сообщения «старт» Птичка всегда будет изменять координату  $Y$  на величину, которая хранится в переменной **скорость**. В начале работы программы **скорость = 5**. Это значит, что Птичка будет лететь вверх, изменяя  $Y$  на 5. В этом же цикле постоянно проверяется условие, не коснулась ли Птичка Труб? Если это произойдёт, то работа программы будет тут же прекращена.

Третий скрипт Птички отвечает за изменение её вертикальной скорости — за «физику» игры. Вы, конечно, знаете, что любой предмет, брошенный вверх, летит с замедлением. Вертикальная скорость брошенного предмета постоянно уменьшается до тех пор, пока в верхней точке полёта не станет равной нулю — предмет перестанет подниматься и начнёт падать. Это означает, что скорость подъёма стала отрицательной, и направление движения изменилось с подъёма на падение. Для того чтобы смоделировать этот закон физики в игре, мы будем постоянно уменьшать значение переменной **скорость** на 0.5. При нажатии клавиши <Пробел> Птичка будет взлетать, махая крыльями, и устанавливать значение





переменной **скорость** равным 5. Как только клавиша <Пробел> будет отпущена, Птичка перестанет махать крыльями и продолжит уменьшать скорость на 0.5.

когда щелкнут по   
перейти в x: -100 у: 0

когда я получу **старт**  
задать **скорость** **значение** 5  
всегда  
изменить **у** на **скорость**  
если **касается** [Труба v] , то  
стоп все v

когда я получу **старт**  
всегда  
если **клавиша** [пробел v] **нажата?** , то  
задать **скорость** **значение** 5  
повторять пока не   не   [клавиша [пробел v] **нажата?**]  
следующий костюм  
иначе  
изменить **скорость** **на** -0.5

Игра готова, попробуйте набрать как можно больше очков.

Это скрипты Птички.





### Эксперимент

Изменяйте начальное значение переменной **скорость** и значение её отрицательного изменения. Посмотрите, как будет меняться полёт Птички.

Сохраните проект.

## 17.3. ЗАДАНИЯ

1. Доработайте проект, чтобы Птичка управлялась не только клавишей <Пробел>, но и щелчком мыши.
2. Измените скорость движения Труб.
3. Добавьте Монетку, которая при касании Птички будет давать сразу 5 очков.
4. Сделайте так, чтобы при касании Труб игра не прекращалась, а просто очки уменьшались на 1. Условие поражения — количество очков меньше нуля.
5. Добавьте условие победы. Если игрок наберёт 50 очков, то игра заканчивается и появляется надпись «Вы победили!».



## ГЛАВА 18. ИГРА «ЗАЩИТА БАЗЫ»

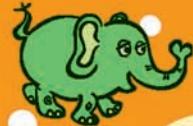
На севере построили Базу для хранения мороженого. Песцы не любили мороженое и были очень недовольны. Они решили закрыть Базу. Для защиты Базы соорудили пушку, которая может стрелять мороженым. Управляй пушкой и помоги защитить Базу! Победа не за горами, Песцов не больше двадцати!



### 18.1. СОЗДАЁМ СПРАЙТЫ И ФОН

Создайте новый проект. Удалите Кота.

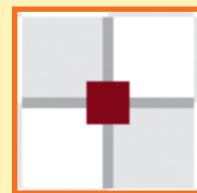
Дело происходит на севере, поэтому фон должен быть похож на снег. Закрасьте фон серым цветом.



Создайте спрайт Пушки.  
Установите масштаб  
800% и нарисуйте два  
прямоугольничка.



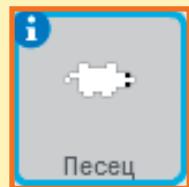
Создайте спрайт  
Снаряда. Установите  
масштаб 800% и  
нарисуйте точку.



Создайте спрайт  
Базы, это просто  
квадрат.



Создайте  
спрайт  
Песца.



### Совет

Песец белого цвета, поэтому нарисовать его  
на белом фоне очень трудно.

Чтобы упростить  
задачу, залейте фон  
тёмно-серым цветом.



После того как нарисуете  
Песца, залейте серый фон  
прозрачным цветом.



Посмотрите на странице 173, как выглядят костюмы на  
сером фоне. Вам может показаться, что это вовсе не Песец,  
а белый медведь, но я точно знаю, что это именно Песец!



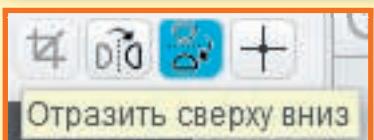
Создайте два костюма Песца, они нужны для анимации шагов.  
Первый костюм.



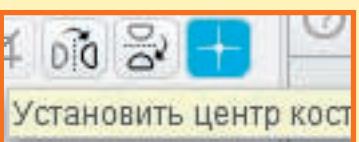
Выберите Песца и отразите его вертикально.



Второй костюм нарисуйте с применением зеркального отражения. Нажмите кнопку **Отразить сверху вниз**.



После отражения обязательно установите центр костюма на носу Песца. Выберите инструмент **Установить центр костюма**.



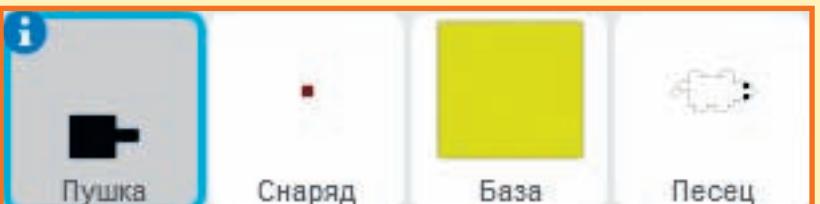
Установите центр первого костюма.



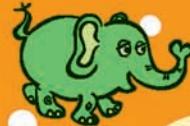
Установите центр второго костюма.



Все спрайты готовы, переименуйте их.

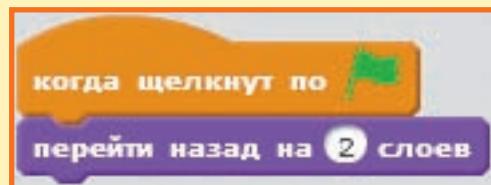


Теперь можно программировать!



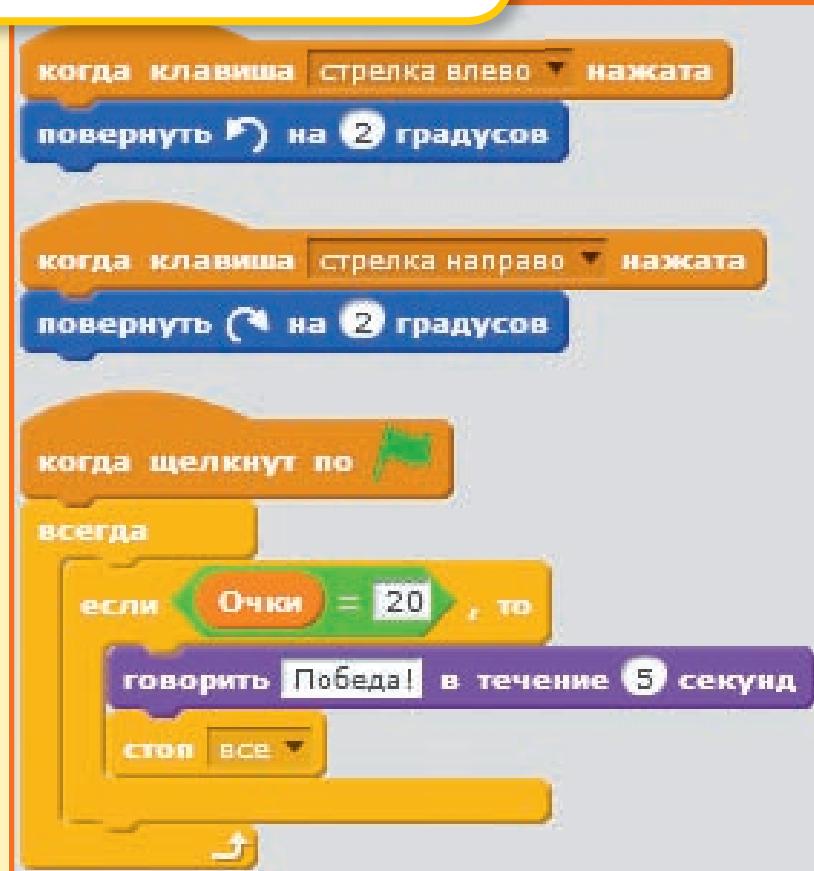
## 18.2. ПРОГРАММИРУЕМ ПОВЕДЕНИЕ СПРАЙТОВ

Программа Базы  
очень простая.



Так как База и Пушка находятся на одном и том же месте, то может получиться, что Пушка будет не видна из-за Базы. Для того чтобы Пушка была всегда видна, Базу надо переместить на 2 слоя назад.

Теперь запрограммируйте Пушку.  
Сначала создайте переменную **Очки**,  
она будет считать заработанные очки.



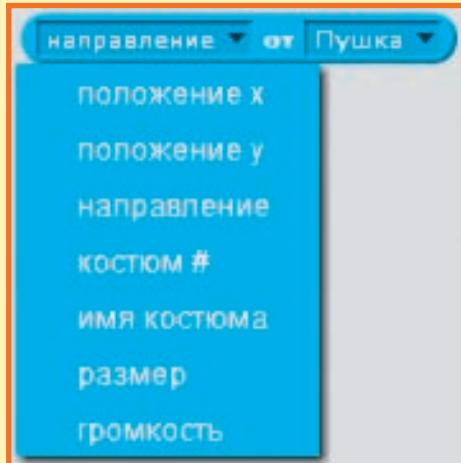


Первые два скрипта позволяют игроку поворачивать Пушку клавишами-стрелками  $\leftarrow$  и  $\rightarrow$ . Третий скрипт постоянно проверяет значение переменной **Очки**. Если значение равно 20, то вы победили!

Теперь  
запрограммируем  
Снаряд.

```
когда щелкнут по [зеленый флаг]
всегда
    повторять пока не [клавиша пробел нажата?]
        повернуть в направлении [направление от Пушка]
    конец
когда щелкнут по [зеленый флаг]
всегда
    перейти в [Пушка]
    ждать до [клавиша пробел нажата?]
    повторять пока не [касается край?]
        идти [10 шагов]
    конец
```





Первый скрипт постоянно поворачивает Снаряд в направлении Пушки, это позволяет Снаряду всегда вылетать точно из ствола Пушки, а не сбоку. Второй скрипт постоянно перемещает Снаряд в Пушку (перезаряжает её) и ждёт до тех пор, пока игрок не нажмёт клавишу <Пробел>. После того как <Пробел> будет нажат, снаряд полетит в нужном направлении, пока не коснётся края.

В первом скрипте Снаряда использована очень мощная встроенная переменная. Она позволяет получить многие параметры спрайтов и сцены.

Теперь запрограммируйте Песца. Спрайт этого персонажа всего один, поэтому для создания армии Песцов мы используем клоны. Создадим армию клонов!

Воспользуемся блоком создать клон.

создать клон себя самого

Для управления клонами служит блок когда я начинаю как клон.

когда я начинаю как клон

Для удаления клонов предназначен блок удалить клон.

удалить клон



когда щелкнут по

задать очки значение 0

повторить 20

перейти в x: выдать случайное от -240 до 240 у: 190

ждать 2 секунд

создать клон себя самого

когда я начинаю как клон

всегда

повернуться к База

повторять пока не касается База ?

идти 0.3 шагов

следующий костюм

говорить База закрыта! в течение 2 секунд

стоп все

когда я начинаю как клон

всегда

если касается Снаряд ?, то

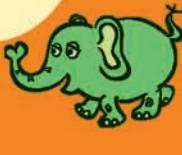
изменить очки на 1

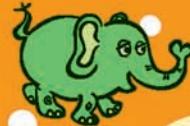
удалить клон

Сделайте вот такую программу для Песца.

Скрипт движения.

Скрипт встречи со снарядом.





Первый скрипт создаёт армию из 20 Песцов. Они появляются в случайном месте в верхней части сцены с интервалом в 2 секунды.

Второй скрипт отвечает за перемещение Песцов. Песцы всегда поворачиваются к Базе и медленно идут, пока не коснутся её. После каждого шага они меняют костюм для создания анимационного эффекта. После того как Песец коснётся базы, выполнение цикла с условием прекратится, Песец скажет, что База закрыта, и выполнение программы будет остановлено.

Третий скрипт отвечает за попадание снаряда в Песца. Если снаряд коснётся Песца, то переменная **Очки** будет увеличена на 1, а клон Песца будет удалён.



### Совет

Никогда не создавайте больше 100 клонов, это очень замедлит выполнение программы.

Протестируйте работу проекта и не забудьте сохранить его.

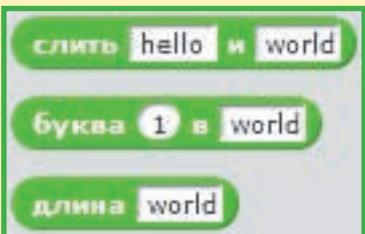
## 18.3. ЗАДАНИЯ

1. Ускорьте движение Песцов.
2. Сделайте так, чтобы Песцы могли появляться в любом месте сцены.
3. Сделайте так, чтобы Пушка стреляла лазером.
4. Добавьте Базе несколько жизней.
5. Добавьте на поле мины, которые будут помогать Базе.
6. Придумайте своих персонажей и замените костюмы Базы, Пушки, Снаряда и Песцов в соответствии с вашим замыслом (скрипты можно не изменять) — получится совсем другая игра!



# ГЛАВА 19. ВИКТОРИНА

## 19.1. РАБОТА С ТЕКСТОМ



В Scratch можно программировать работу со строками текста, так же как и с числовыми переменными. Стока может быть одной буквой, словом или предложением. Строки можно сравнивать, соединять, произносить

и выполнять с ними другие действия. Для работы с текстом существуют три овальных блока.

Они очень похожи на переменные, их можно вставлять в прямоугольные поля других блоков, например в блок говорить.

Спрайт скажет:

Привет, мир!



Блок слить позволяет соединить две строки. Например, если соединить две строки «Привет, » и «мир!», то получится строка «Привет, мир!». Блок буква создаёт строку, состоящую из одной буквы, вынутой из слова. Например, если вытащить первую букву из слова «мир», то получится строка «м». Блок длина позволяет определить длину слова. Например, длина слова «Колбаса» равна 7.

Спрайт скажет:



7



Блоки работы с текстом можно вставлять один в другой, чтобы получить предложения из нескольких слов.

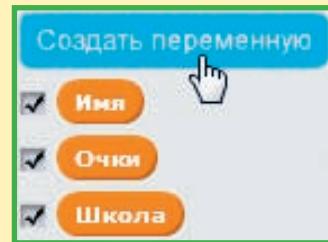


Выполняя этот блок, спрайт в течение 7 секунд будет говорить фразу «В слове „Колбаса“ всего 7 букв».

## 19.2. ПРОСТАЯ ВИКТОРИНА

Давайте потренируемся использовать блоки работы с текстом и сделаем игру «Викторина».

Создайте новый проект. Создайте три переменные.



Добавьте на сцену фон.



когда щелкнут по

задать Школа ▾ значение 0

задать Имя ▾ значение 0

задать Очки ▾ значение 0

спросить Ваше имя? и ждать

задать Имя ▾ значение ответ

спросить В какой школе Вы учитесь? и ждать

задать Школа ▾ значение ответ

говорить слить Здравствуйте, и Имя в течение 3 секунд

говорить Это викторина о столицах государств. в течение 3 секунд

говорить Всего 3 вопроса, в течение 3 секунд

говорить Названия столиц вводите с большой буквы, в течение 3 секунд

спросить Столица Франции? и ждать

если ответ = Париж , то

изменить Очки ▾ на 1

спросить Столица Португалии? и ждать

если ответ = Лиссабон , то

изменить Очки ▾ на 1

спросить Столица Словакии? и ждать

если ответ = Братислава , то

изменить Очки ▾ на 1

говорить слить слить Ученик и Школа и школы в течение 3 секунд

говорить слить По имени и Имя в течение 3 секунд

говорить слить слить Набрал и Очки и очков в викторине! в течение 3 секунд

Соберите для Кота вот такой скрипт.



Рассмотрим работу скрипта. Сначала, как обычно, обнуляются переменные. Далее Кот спросит ваше имя, которое будет помещено на хранение в переменную **Имя**. Затем он спросит, в какой школе вы учитесь, и поместит ответ в переменную **Школа**. Потом Кот произнесёт небольшой монолог и спросит название столицы Франции. Если содержимое встроенной переменной **ответ** совпадёт со словом «Париж», то значение переменной **Очки** увеличится на единицу. Кот задаст вопрос ещё два раза, а затем объявит о результатах викторины. Например, вот так: «Ученик 5 школы По имени Егор Набрал 3 очков в викторине!»

Сохраните проект.

## 19.3. ЗАДАНИЯ

1. Доработайте проект, сделайте 10 вопросов.
2. Помимо вопроса о школе, добавьте вопрос о классе, в котором учится игрок.
3. Измените проект, задайте вопросы не о столицах, а о чём-нибудь другом — о том, что вы очень хорошо знаете.
4. Добавьте в проект ещё один спрайт, который по результатам викторины поставит оценку. Если все 10 ответов правильные, то «5»; если одна ошибка, то «4»; если две ошибки, то «3»; а если больше двух ошибок, то «2».

## 19.4. ВИКТОРИНА СО СПИСКАМИ

Вы могли заметить, что викторина, состоящая даже из десяти вопросов, представляет собой очень громоздкую программу, а что будет, если вам понадобится задать 50 вопросов? Проект будет очень неповоротливым. Для того чтобы сделать проект короче, можно использовать списки. Списки — это наборы пронумерованных переменных. Например, список с именем **Столицы** хранит 3 значения.





**Столицы**

1	Москва
2	Берлин
3	Пекин
+	длина: 3

Эти 3 значения, как три переменные с одинаковым именем, но с разными номерами.

элемент 1 из Столицы

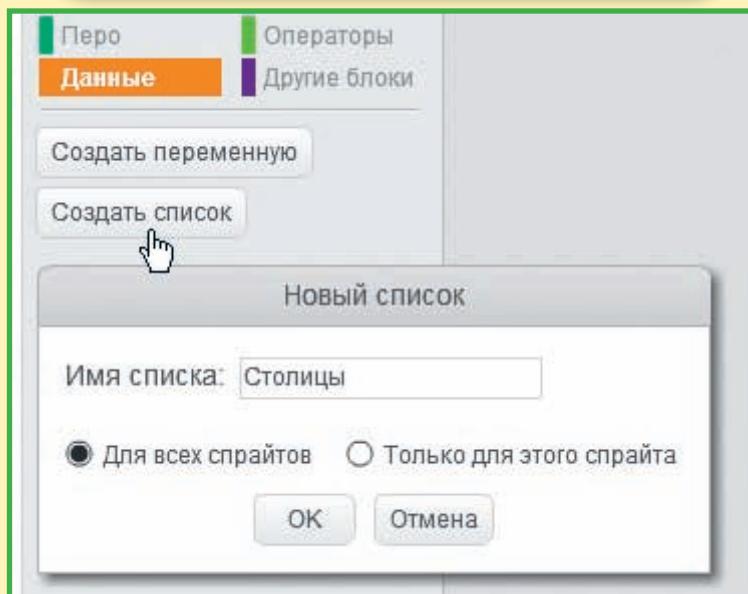
элемент 2 из Столицы

элемент 3 из Столицы

Элемент 1 из списка **Столицы** — это «Москва», элемент 2 из списка **Столицы** — это «Берлин», а элемент 3 — это «Пекин».

Для того чтобы лучше понять, как работать со списками, создайте знакомую вам викторину о столицах новым способом. Начните с нового проекта.

Сначала создайте список **Столицы**:  
в блоке **Данные** нажмите кнопку  
**Создать список** и введите имя  
списка.



Затем создайте список **Страны**.  
Создайте переменную **НомерВопроса**.  
А теперь соберите программу.

The Scratch script consists of the following blocks:

- When green flag clicked:
  - Delete all from Столицы
  - Delete all from Страны
  - Add Москва to Столицы
  - Add Берлин to Столицы
  - Add Пекин to Столицы
  - Add России to Страны
  - Add Германии to Страны
  - Add Китая to Страны
  - Set НомерВопроса to 1
  - Set очки to 0
- Repeat [3 times]
  - Ask [Какая столица у {Страны}] and wait
  - If answer = element НомерВопроса of Столицы then
    - Change очки by 1
    - Change НомерВопроса by 1
- Say [Вы набрали {очки} очков.] for [3 seconds]

В начале работы программы мы создаём два пустых списка **Столицы** и **Страны** и заполняем их названиями столиц и стран.



### Важно!

Порядковый номер страны должен  
обязательно соответствовать номеру столицы!



Переменная **НомерВопроса** будет изменяться в цикле от 1 до последнего вопроса. В цикле будет задаваться вопрос, содержащий название очередной страны из списка, и если ответ совпадёт с названием столицы из соответствующего списка, то переменная **Очки** будет увеличена на 1. После этого **НомерВопроса** увеличится на 1, и при очередном проходе цикла будет задан вопрос о следующей стране. Так будет продолжаться до тех пор, пока **НомерВопроса** не станет больше количества стран в списке. Выполнение цикла завершится, и Кот скажет, сколько очков вы набрали.



### *Обратите внимание!*

Цикл не изменится при увеличении количества стран в списке, изменится лишь число в условии цикла, с которым сравнивается переменная **НомерВопроса**! Только представьте, какого размера был бы скрипт, если бы мы не использовали списки и добавили 100 вопросов!

Сохраните проект.

## 19.5. ЗАДАНИЯ

1. Усовершенствуйте программу: списки не должны показываться на экране.
2. Добавьте ваши вопросы в викторину со списками.
3. Сделайте по два вопроса для каждой страны: про столицу и про континент, на котором находится страна.
4. Составьте по три вопроса на каждый объект вашей викторины.



# СЕКРЕТЫ И СОВЕТЫ

Всё ли вы запомнили? Всеми ли секретами готовы воспользоваться в будущих проектах? Если вдруг что-то забыли — не беда. Вы можете открыть книгу на нужной странице и вспомнить...

Как вернуть исчезнувший спрайт на сцену?	Стр. 27–28
Как вернуть спрайт в центр сцены?	Стр. 103
Как удалить персонаж со сцены?	Стр. 30
Как изменить размер персонажа?	Стр. 35
Как сменить персонажу костюм?	Стр. 43
Как изменить имя персонажа?	Стр. 155
Как узнать текущие координаты персонажа на сцене?	Стр. 102
Как добавить всплывающую подсказку?	Стр. 142
Как прослушать звук?	Стр. 19
Как укоротить звук?	Стр. 49
Как перевернуть звук?	Стр. 47
Как записать звук?	Стр. 48
Как дублировать скрипт?	Стр. 32
Как выровнять скрипты?	Стр. 86
Как сбросить эффекты?	Стр. 37
Как узнать координаты кончика курсора мыши?	Стр. 102
Как очистить сцену?	Стр. 56
Как рисовать ровные линии?	Стр. 109
Как нарисовать белый спрайт?	Стр. 172
Как стереть нарисованное?	Стр. 59
Как передать сообщение?	Стр. 139
Что делать, если программа «тормозит»?	Стр. 178



## ЗАКЛЮЧЕНИЕ

Итак, дорогие друзья, вы познакомились со Scratch и научились программировать простые игры и мультфильмы. Это большое достижение! Однако в этой книге описаны не все возможности программы. Не рассказано о создании собственных блоков, мелодий из нот, о работе с видео и о многих математических операторах. Но, как говорится, дорогу осилит идущий. Постарайтесь разобраться с этими темами самостоятельно, используя библиотеку знаний о Scratch (Scratch Wiki), расположенную по адресу <https://wiki.scratch.mit.edu>. Вопросы можете задавать на русскоязычном форуме о программировании на Scratch, расположенном по адресу <https://scratch.mit.edu/discuss/27/>. Также можете задавать вопросы автору по электронной почте [scratch.book@ya.ru](mailto:scratch.book@ya.ru).

**УДАЧИ! И ДО НОВЫХ ВСТРЕЧ!**





# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ ДЛЯ ВЗРОСЛЫХ</b>	<b>3</b>
<b>ГЛАВА 1.</b> Знакомство со Scratch	<b>11</b>
<b>ГЛАВА 2.</b> Усложнение первого проекта	<b>23</b>
<b>ГЛАВА 3.</b> Знакомство с эффектами	<b>36</b>
<b>ГЛАВА 4.</b> Знакомство с отрицательными числами	<b>45</b>
<b>ГЛАВА 5.</b> Знакомство с пером	<b>53</b>
<b>ГЛАВА 6.</b> Циклы	<b>61</b>
<b>ГЛАВА 7.</b> Условный блок	<b>74</b>
<b>ГЛАВА 8.</b> Мультфильм «Акула и Рыбка»	<b>81</b>
<b>ГЛАВА 9.</b> Что такое координаты X и Y	<b>92</b>
<b>ГЛАВА 10.</b> Мультфильм «Пико и Привидение»	<b>101</b>

## **ГЛАВА 11.**

Игра «Лабиринт»

**108**

## **ГЛАВА 12.**

Мультфильм «Кот и Летучая мышь»

**114**

## **ГЛАВА 13.**

Игра «Пройди сквозь кактусы»

**124**

## **ГЛАВА 14.**

Игра «Ведьма и Волшебник»

**133**

## **ГЛАВА 15.**

Кот-математик

**144**

## **ГЛАВА 16.**

Игра «Вертолёт»

**154**

## **ГЛАВА 17.**

Полёт с ускорением «Флэппи Бёрд»

**162**

## **ГЛАВА 18.**

Игра «Защита базы»

**171**

## **ГЛАВА 19.**

Викторина

**179**

## **СЕКРЕТЫ И СОВЕТЫ**

**186**

## **ЗАКЛЮЧЕНИЕ**

**187**