Порохова Ирина Алексеевна

РОБОТЫ LEGO WEDO. ЗАНЯТИЕ 3. СРЕДА ПРОГРАММИРОВАНИЯ СКРЕТЧ И ДВЕ ИГРЫ С ДЖОЙСТИКОМ

В двух предыдущих статьях по роботам Lego WeDo мы познакомились с базовыми и дополнительными моделями, научились программировать в среде Lego Education Software и подготовились к решению более сложных задач, которые относятся к робототехнике. Усложнение робота – это движение в двух направлениях: во-первых, это усложнение конструкции робота, использование нескольких моторов и нескольких датчиков одновременно, а во-вторых, это создание более сложных программ, допускающих гибкое, «умное», реагирование робота на внешние воздействия. К великому сожалению, добиться и того и другого (например, научить машинку, собранную из Lego WeDo, двигаться вдоль линии или проходить лабиринт) пока не представляется возможным: современные программные средства для роботов Lego WeDo пока не достаточно развиты. Тем не менее, даже на конструкторе Lego WeDo можно построить великолепную модель, относящуюся к настоящей робототехнике, – балансирующего на двух колесиках робота. Возможно, в скором будущем появятся программы, которые, как и Lego Education Software, будут поддерживать одновременную работу нескольких датчиков и моторов и которые, как программная среда Скретч, позволят создавать более сложные программы для тонкого управления моторами и датчиками.

В этой статье мы сделаем по шагу в двух направлениях – усложним самого робота и усложним программные средства. Сначала построим джойстик с датчиком наклона и будем с помощью джойстика управлять движением машинки с двумя моторами. Машинка, слушаясь джойстика, будет ездить вперед и назад и поворачивать вправо-влево. Эта задача может быть решена в базовой программе Lego Education Software. Затем мы познакомимся со средой программирования Скретч. Эта среда интересна как сама по себе, безотносительно к роботу Lego WeDo, так и необходима нам на последующих этапах, где робот будет обнаруживать препятствия, будет пытаться двигаться по линии или балансировать на двух колесах. Мы применим тот же джойстик для управления перемещением по экрану персонажа Скретча, рассмотрим несколько ярких проектов, выполненных на Скретче, и поучимся переводить программы с языка Lego Education WeDo Software на язык Скретч.

ДЖОЙСТИК И МАШИНКА

Персонажем на экране или реальным роботом интереснее управлять не с клавиатуры, а джойстиком. Джойстик с датчиком наклона можно собрать, например, по следующей схеме (рис. 1). Ручка джойстика состоит из двух половинок, каждая из которых крепится на оси, в результате чего может поворачиваться вокруг оси. Сверху установлен датчик наклона.

Теперь соберем машинку с двумя моторами (рис. 2 *а–е*) и заставим ее слушаться джойстика. В предыдущей статье мы уже со-

а) *б)* *в)*

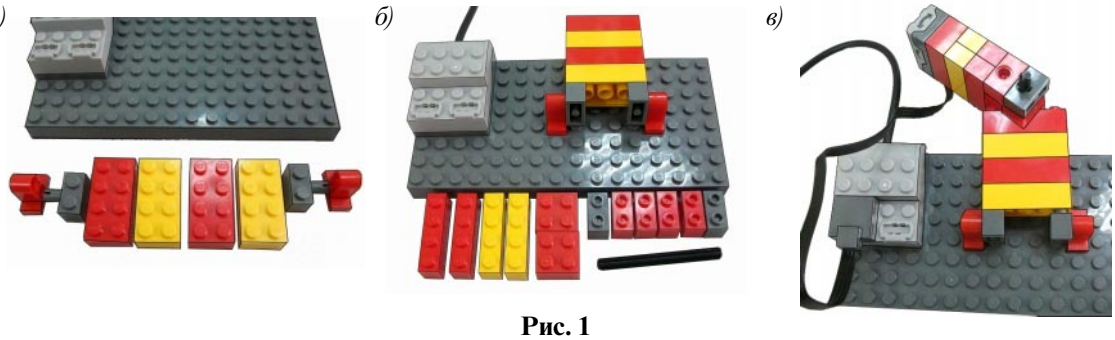
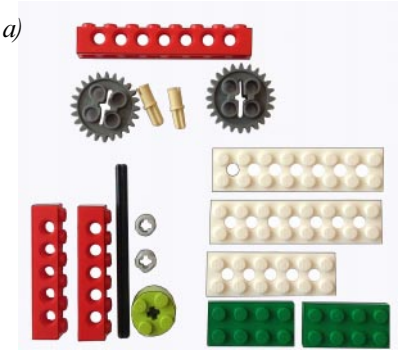
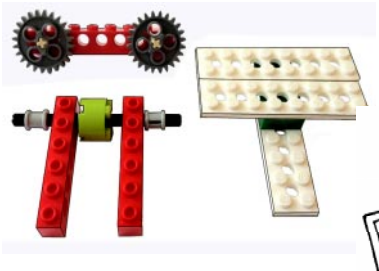
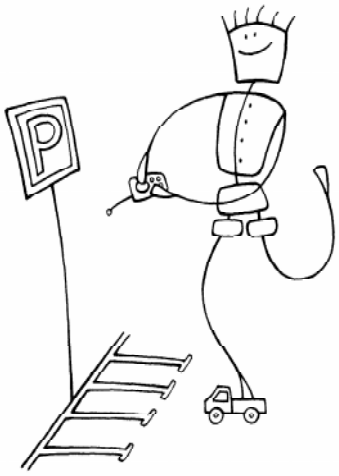


Рис. 1

а) *б)*



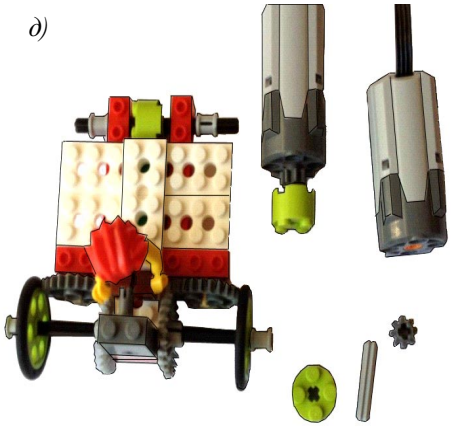




в) *г)*

...имеет смысл ... решать простые робототехнические задачи, такие как ... парковка.

д) *е)*



...великолепную модель... балансирующего на двух колёсах робота...

Рис. 2

бирали машинку с двумя моторами. Новая версия машинки, как нам кажется, обладает большей маневренностью.

После установки моторов машинка готова (рис. 2*е*).

Оба мотора следует подключить к одному Лего-коммутатору, а датчик наклона – к другому. Программа для управления моторами может иметь следующий вид (рис. 3).

Первый и второй мотор различаются маркировкой, которая устанавливается при нажатой клавише Shift. Все четыре программы запускаются одновременно при нажатии кнопки A (в английской раскладке) и далее каждая из программ ждет наклона датчика в определенную сторону. Для движения вперед или назад надо, чтобы моторы вращались в разные стороны, тогда колеса машинки будут вращаться в одинаковую сторону. Чтобы совершить поворот, надо чтобы одно колесо вращалось вперед, а другое вращалось назад или стояло на месте.

В принципе, такой машинкой можно управлять без джойстика просто нажатием кнопок со стрелками на клавиатуре. Такому способу управления соответствует более простая программа, не использующая циклы.

Напомним, что робот может быть оснащен максимум шестью моторами одновременно. Также можно подключить пару датчиков, например расстояния (только к разным Лего-коммутаторам), и они также будут различаться маркировкой. Поскольку машинка с двумя моторами может выполнять повороты, имеет смысл развивать это направление и решать простые робототехнические задачи, такие как программирование движения машинки по квадрату, по заданной траектории, парковка и т. д.

Рис. 3

Второе направление по усложнению робота связано с использованием более мощной среды программирования, которая приблизит нас к настоящей робототехнике.

ЗНАКОМЬТЕСЬ. СКРЕТЧ!

Скретч (Scratch) – это замечательная, интуитивно понятная среда программирования, в которой можно создавать интерактивные диалоги, презентации, мультфильмы и игры, писать программы и управлять Лего-моделями. Скретч является свободно распространяемой программой, которую можно загрузить с сайта http://scratch.mit.edu. Программирование осуществляется на 50 языках, в том числе, на русском языке. На сайте разработчиков существует огромный ресурс для обмена проектами, где каждый зарегистрированный пользователь может скачать чужие проекты, закачать свои и участвовать в обсуждении.

Скретч – это объектно-ориентированная среда, в которой блоки программ собираются из разноцветных кирпичиков-команд. Дети могут собирать свои программы-процедуры из блоков так же, как они собирали конструкции из разноцветных кирпичиков Лего. Можно нарисовать или загрузить картинки «героев» программы, так называемых спрайтов, приписать им простые инструкции и выстроить управляющие структуры. Спрайты могут взаимодействовать между собой, формируя сообщество в среде Скретч. В результате выполнения простых команд может складываться сложная модель, в которой будет взаимодействовать множество объектов, наделенных различными свойствами.

Начальный уровень программирования столь прост и доступен, что Скретч рассматривается в качестве средства обучения даже

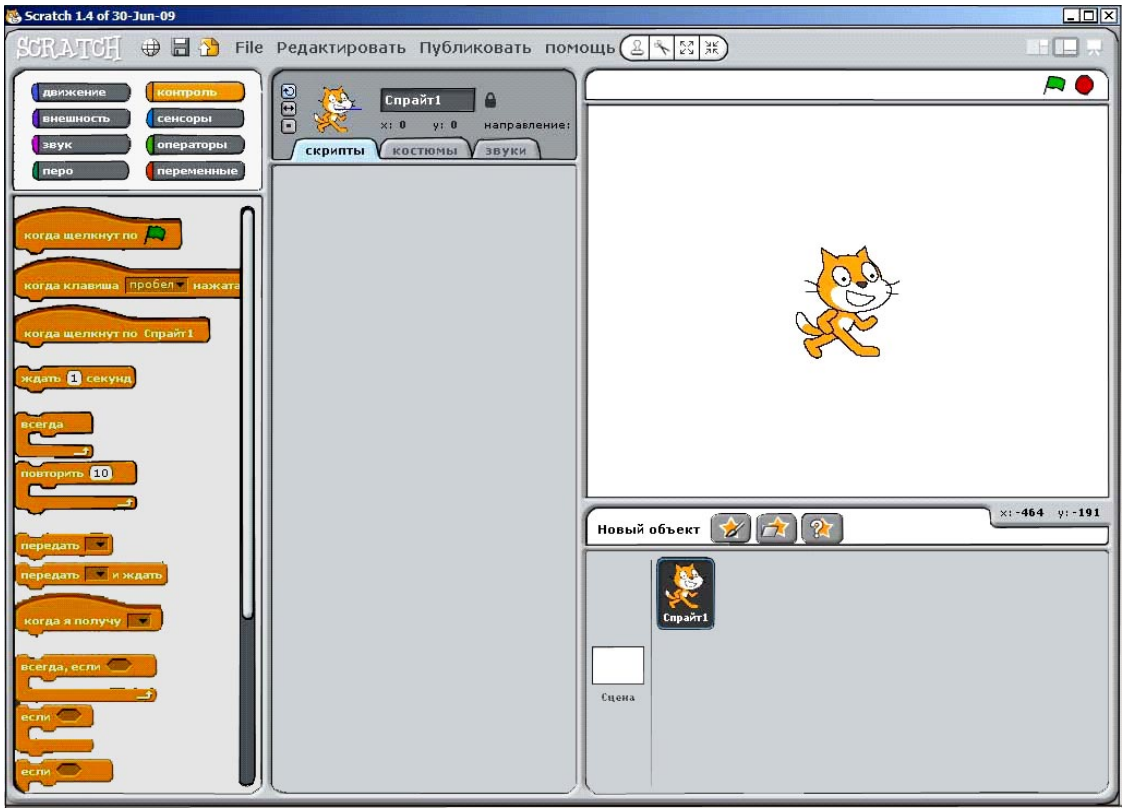


Рис. 4

младших школьников. Подробное руководство по использованию среды Скретч создано Евгением Патаракиным и доступно в интернете по ссылке umr.rcokoit.ru/dld/metodsupport/scratch1.pdf.

Экран программы Скретч имеет вид, представленный на рис. 4.

В верхней части программы находится меню команд.

В левом верхнем блоке сгруппированы команды языка. Таких групп в языке восемь, и каждая из них имеет в интерфейсе среды определенный цвет:

- Движение (синий) – содержит команды перемещения объектов;
- Внешность (фиолетовый) – команды изменения внешнего вида объекта;
- Звук (лиловый) – команды управления звуком;
- Перо (темно-зеленый) – команды рисования на экране;
- Контроль (желтый) – контролирующие операторы, условные операторы и операторы циклов;
- Сенсоры (голубой) – датчики, команды управления мышью, таймером, диалогами;
- Операторы (ярко-зеленый) — операции с числами, логические операторы, вычисления, команды сравнения;
- Переменные (оранжевый) – команды управления переменными.

В центральной части экрана серого цвета собираются все кирпичики команд и формируется программа. Запустить программу на выполнение можно, нажав на зеленый флажок, находящийся под командным меню в правом верхнем углу, или дважды щелкнув на сформированных скриптах программы. Остановить выполнение программы можно, щелкнув мышью на красном восьмиугольнике, который находится там же, рядом с зеленым флажком.

В правой части экрана находится рабочая область белого цвета. Здесь задается фон исполнения, или сцена, и собираются объекты программы, называемые спрайтами. Спрайты имеют свой вид. Они могут видо-

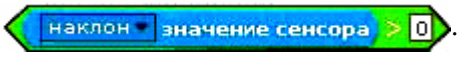
Спрайты... могут видоизменяться... менять свой костюм.

изменяться или, иначе говоря, менять свой костюм. Можно пользоваться готовыми спрайтами, которые можно загружать из библиотек, а можно рисовать самостоятельно, используя достаточно хороший внутренний графический редактор. И героям, и фону можно давать команды.

Команды управления Лего мотором объединены в подгруппу Моторы группы Движение, которая становится доступна после нажатия кнопки «Показать блоки мотора» в меню Редактировать. Программа содержит следующие команды мотора (табл. 1).

Кроме подгруппы мотора программа может получать значения от двух датчиков расстояния и наклона, которые доступны в разделе Сенсоры в команде

Для датчика наклона доступно шесть значений. Значение сенсора 1 соответствует наклону вверх, 3 – наклону вниз, 2 – наклону вправо, а 4 – наклону влево. Число 0 соответствует значению «нет наклона». Еще одно значение «любой наклон» можно задать командой



Значения датчика расстояния изменяются в диапазоне от 0 до 100. На расстояниях, меньших 7 см от светлой поверхности, датчик расстояния показывает ноль. От 7 до 20 см показания датчика меняются в диапазоне от 0 до 75, а затем медленно увеличиваются. На темной поверхности показания будут иными. Таким образом, датчик расстояния измеряет не расстояние, а некоторую освещенность поверхности.

Блоки	Пояснения
включить мотор на 2 сек	Включает мотор на заданное время
мотор вкл	Включает мотор
мотор откл	Выключает мотор
мощность мотора 100	Задает мощность мотора (от 0 до 100) и включает его
мотор в сюда	Задает направление вращения мотора (сюда – по часовой стрелке, туда – против часовой стрелки, перевернуть - меняет направление на противоположное)

Табл. 1

ПРОЕКТЫ НА СКРЕТЧЕ. ОТ ПРОСТОГО ...

Знакомиться со Скретчем удобно, рассматривая готовые проекты, которые можно скачать с сайта http://scratch.mit.edu. Мы отобрали несколько характерных проектов и присвоили им номера по мере усложнения. Проекты для удобства собраны на прилагающемся диске.

Простой, но очаровательный **проект «Новогодняя открытка»** (1 new-year card.sb) представляет собой анимированную новогоднюю открытку, в которой зайчик движется по экрану вправо-влево, мигает гирлянда на елочке, и Дед Мороз поднимает и опускает посох (рис. 5). Чтобы запустить этот проект, вам необходимо установить у себя программу Скретч, открыть этот файл и нажать на зеленый флажок в правой верхней части экрана.

Рассмотрим, как сделан этот проект. Здесь есть три героя, или спрайта: гирлянда, Дед Мороз и кролик. Если

Рис. 5

нажать на спрайт гирлянда, в центральной части экрана появится следующий скрипт для гирлянды (рис. 6).

Он означает, что программа стартует при щелчке мышки по зеленому флажку, запускает бесконечный цикл «всегда», в котором после секундного ожидания отображается следующий костюм для спрайта гирлянды. Щелкнув мышкой по вкладке «костюмы» в верхней средней части экрана, можно увидеть, что у гирлянды имеется восемь разных костюмов, соответствующих разным перемигиваниям лампочек. Смена этих костюмов и создает эффект перемигивания гирлянды (рис. 7).

Рассмотрев скрипт и костюмы для Деда Мороза, можно увидеть, что он выполнен по аналогии. Немного более сложный скрипт относится к кролику. Тут помимо смены костюма добавлено перемещение кролика из одной точки с заданными координатами (x, y) в другую точку. Чтобы перемещение было плавным, вместо команды

, используется команда

.

Рис. 7

Рис. 6

Более сложный **проект «Игра в мяч»** (2 igra ball.sb) выполнен в виде игры, в которой надо, управляя котенком с помощью стрелок на клавиатуре, стараться поймать голубой мяч и не попасться в лапы собачке (рис. 8). В этом проекте снова действуют три персонажа, в скриптах собаки и мяча появляются новые команды «если край, оттолкнуться», что дает инструкции спрайтам отталкиваться от края экрана. Помимо этого появляются действия по условию касания одним спрайтом другого спрайта:

. Самый «сложный» скрипт относится к спрайту «котенок». Здесь помимо команд, задающих движение котенка по экрану при нажатии соответствующей стрелки, имеется подсчет очков в игре, реализованный с помощью переменной «очки» и команд для переменных «поставить» и «изменить» (рис. 9).

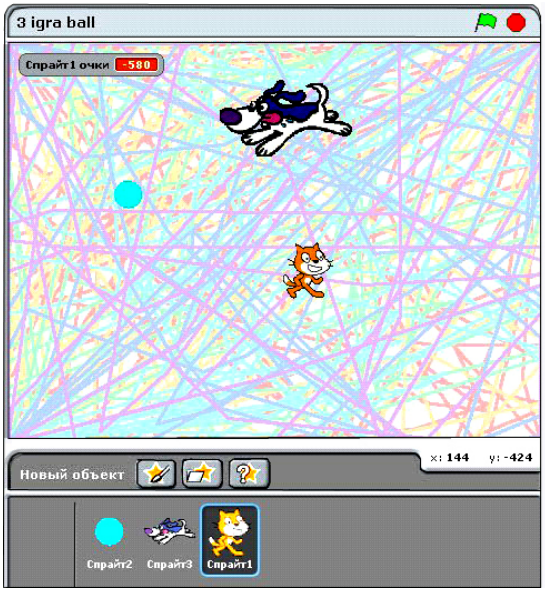


Рис. 8

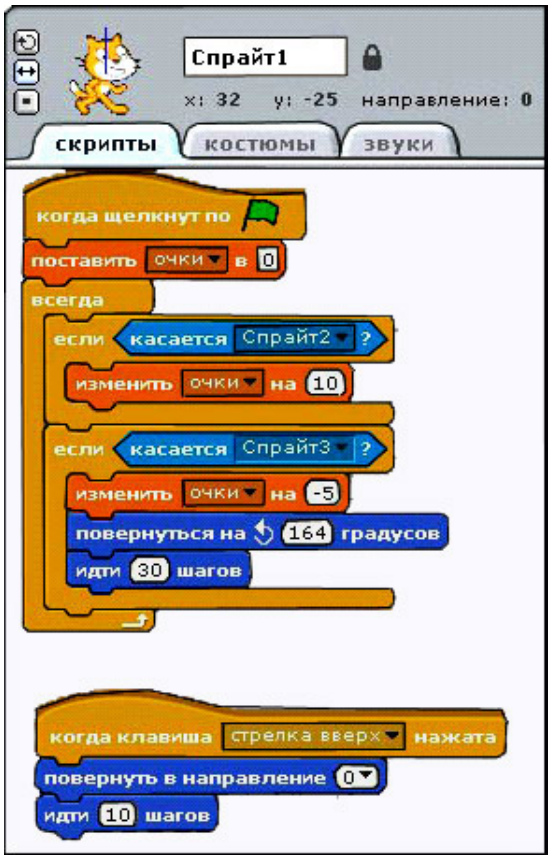


Рис. 9

Проект «Устный счет» (3 ustnyi schet.sb) выстроен на использовании команд для интерактивного диалога, реализуемого с помощью команды «спросить» из раздела Сенсоры и отклика «ответ» в том же разделе. В проекте можно выбрать героя, который будет задавать вопросы, следует отвечать на вопросы, вводя числовые значения с клавиатуры, и отслеживать количество верных и неверных ответов. Вопросы задаются не только письменно командой «спросить», но и озвучиваются. Для этого они были записаны как звуковые файлы, затем загружены во вкладке «звуки» (вкладка рядом с костюмами) для каждого героя. Воспроизведение файлов выполняется командами «играть звук» из раздела «Звук». Вообще в программе Скретч имеется множество встроенных звуков, которые можно использовать для звукового сопровождения проектов.

Графические возможности Скретча иллюстрирует, например, **проект «Круги и квадраты»** (4 circles and squares.sb). Программа отрисовывает круги и квадраты с завораживающими переливами цвета и размера. Например, небольшой скрипт (рис. 10) создаст следующий рисунок из начального спрайта, имеющего форму синего квадрата на белом фоне (рис. 11).

В Скретче можно менять цвет и толщину пера, регулировать тень пера, применять семь различных графических эффектов, отслеживать траектории персонажей или получать отпечатки.

Таким образом, возможности Скретча намного богаче, чем возможности программы Lego Education WeDo Software, так что, за исключением программирования многомоторных и многодатчиковых моделей, все задачи, выполненные на Lego WeDo могут быть реализованы на Скретче. Но далеко не все программы, написанные на Скретче для

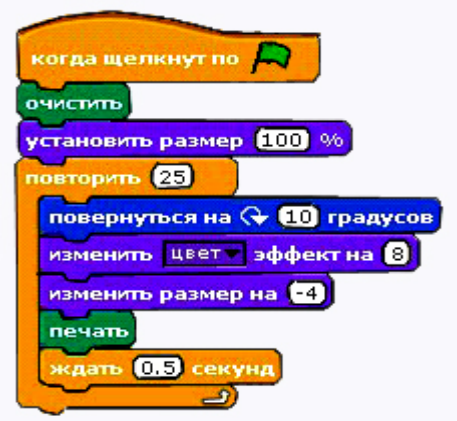


Рис. 10

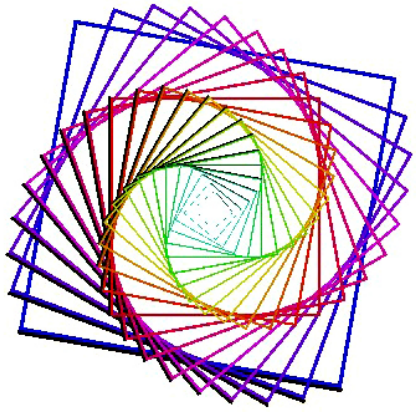


Рис. 11

одного мотора и датчика, могут быть перенесены на Lego Education WeDo.

СКРЕТЧ И РОБОТЫ LEGO WEDO

Для начала было бы интересно управлять персонажем Скретча не клавиатурой, а джойстиком, который мы собирали в первом разделе. На экране программы по умолчанию отображается спрайт-котенок, которым и управляет джойстик согласно программе (рис. 12).

Чтобы было веселее, добавим проигрывание звука. Этот звук предварительно надо импортировать в проект (рис. 13).

Темы для обсуждения, исследований и модернизации модели.

1. Значение сенсора 1 соответствует наклону вверх, 3 – вниз, 2 – вправо, а 4 – влево. А как запрограммировать еще два возможных значения датчика наклона – «любой наклон» и «нет наклона»?

2. Измените программу, добавьте смену костюма так, чтобы котенок шагал.

3. Подключите мотор. Сделайте так, чтобы при движении вперед мощность мотора увеличивалась, а при движении назад – уменьшалась.

4. Попробуйте разнообразить звуки. Запишите нотами мелодию и проигрывайте ее на случайно выбранном инструменте.

5. Придумайте компьютерные игры или задачи для датчика наклона.

Посмотрим теперь, как будут выглядеть программы для некоторых базовых моделей роботов Lego WeDo, рассмотренных в первой статье, на языке Скретч. Примеры отобраны с целью продемонстрировать основные приемы программирования мотора и датчиков, использование переменных и случайного числа. Попробуйте выполнить «перевод» самостоятельно и только потом свериться с нашим решением. Решения могут быть разными, главное, чтобы робот делал одинаковые действия в обеих программах.

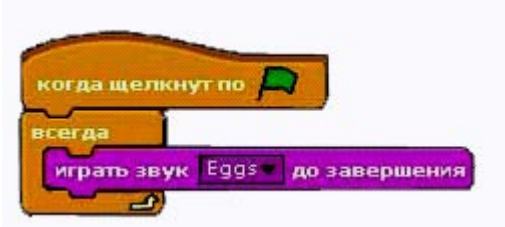


Рис. 13

Рис. 12

Голодный аллигатор

Программа (рис. 14) стартует при нажатии на блоке Начало и запускает цикл, в котором она выполняет ожидание, пока датчик наклона что-нибудь не увидит, включает мотор, чтобы закрыть пасть, воспроизводит звук 17 (хруст), включает мотор в обратном направлении, чтобы открыть пасть, мотор включается на семь десятых секунды.

Следует сделать несколько предварительных замечаний. Во-первых, в Скретче команда мотор в сюда не запустит мотор, так как по умолчанию мощность мотора рав-

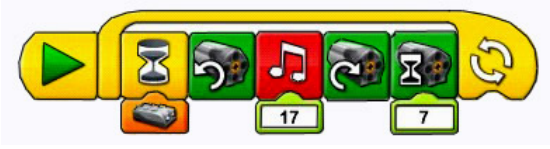


Рис. 14

Рис. 15

на нулю. Перед этой командой надо задать какую-то мощность мотора. Важно помнить, что в Скретче мощность мотора меняется от 0 до 100, а в Lego WeDo от 1 до 10. Временные интервалы в Lego WeDo измеряются в десятых долях секунды, в отличие от Скретча, где время задается в секундах. С учетом сказанного, программа для голодного аллигатора на языке Скретч будет выглядеть так (рис. 15).

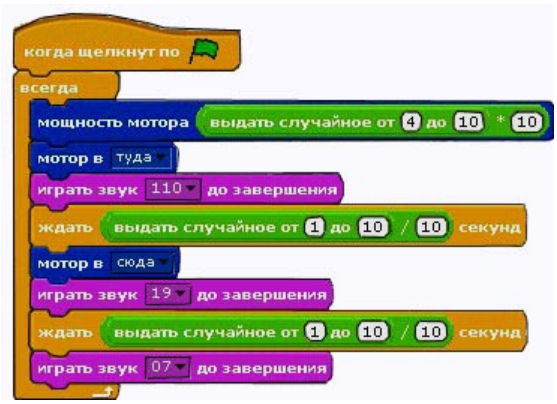


Рис. 17

Рис. 16

Танцующие птички

Программа стартует при щелчке на блоке Начало (рис. 16), чтобы запустить цикл, в котором устанавливает мощность мотора случайным образом, включает мотор по часовой стрелке, воспроизводит звук 18 (хлопанье крыльев), выполняет ожидание в течение случайного промежутка времени, включает мотор в обратном направлении, воспроизводит звук 19 (птичка), выполняет ожидание в течение случайного промежутка времени и воспроизводит звук 7 (волчок).

На языке Скретч подобная программа будет иметь вид (рис. 17).

Звук 110 соответствует звуку 18 из Lego WeDo, просто Скретч не любит числовые имена файлов и присваивает им свою нумерацию. Кроме того, здесь нам потребовалось дополнительное деление на 10, чтобы перевести случайные числа в десятые доли секунды.

Умная вертушка

При щелчке на блоке Начало (рис. 18) программа включает мотор по часовой стрелке, чтобы раскручивать волчок, одновременно с этим воспроизводит звук 15 (мотор), переходит на ожидание, пока датчик расстояния не заметит, что ручку вертушки подняли, и выключает мотор. При этом раскрученный волчок вращается. Чтобы измерить длительность вращения запускается секундомер. Обнуляют Экран и повторяют отсчет секунд, а именно выполняют ожидание в течение одной секунды и прибавляют к Экрану единицу.

Рис. 18

Рис. 20

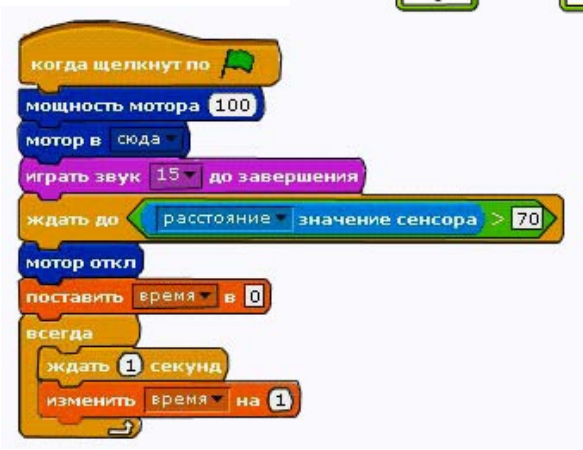


Рис. 19

Рис. 21

Программа на Скретче выглядит так (рис. 19).

В Скретче нет команды типа Ожидать, когда сработает датчик. Датчик непрерывно выдает какие-то значения в диапазоне от 0 до 100. Если объект находится далеко, то значение 100, если близко – ноль или число меньше 100. Поэтому здесь программа ждет, когда значение датчика расстояния будет достаточно большим, и запускает вторую часть с секундомером. Далее, вместо Экрана, в Скретче используются переменные, например «время».

Рычащий лев (рис. 20)

Здесь единственное трудное место – это ожидание любого наклона датчика. Значение датчика наклона равное нулю означает отсутствие наклона. Отрицание этого, то есть не 0, будет соответствовать любому наклону (рис. 21).

Спасение самолета (рис. 22)

Можно сформулировать текст этой программы в виде задачки: используя мотор и датчик наклона, напишите в среде Скретч программу, согласно которой мощность мотора зависит от угла наклона датчика вперед или назад. При смене направления наклона должен меняться звук (рис. 23).

Рис. 23

Рис. 22

Рис. 24

Спасение от великана (рис. 24)

После старта по щелчку на блоке Начало программа воспроизводит звук 13 (храп), пока датчик наклона что-нибудь не заметит, далее включает мотор, чтобы поднять великана, ждет в течение пяти десятых секунды и воспроизводит звук 14 (рычание); после этого выключает мотор (рис. 25).

Здесь использована конструкция «повторять до» из блока Контроль, что обеспечивает воспроизведение цикла, пока не сработает датчик.

Рис. 25

Рис. 26

Можно придумать огромное множество задач для Скретча и роботов Lego WeDo. В качестве примера приведем несложную задачку: используя мотор и датчик расстояния, напишите в среде Скретч программу, согласно которой мощность мотора зависит от расстояния до препятствия. В программе должен меняться звук при приближении и удалении от препятствия. Ключевым моментом тут является «приближение и удаление» (рис. 26).

Видно, что в среде Скретч решение довольно простое и компактное, а в среде Lego WeDo такую задачку не реализовать.

Мы надеемся, что сумели заинтересовать вас средой программирования Скретч (Scratch) и желаем вам успехов в творчестве!

Порохова Ирина Алексеевна, кандидат физико-математических наук, методист Образовательного центра «ИНТОКС».