



РОББО™

Самоучитель
для преподавателей
и родителей

Содержание

Введение	4
Начните работу с РОББО Робоплатформой	6
Начните работу с РОББО Лабораторией	9
Использование программной среды RobboScratch	12
Упражнение 1: Програмируем при помощи RobboScratch	20
Упражнение 2: Цепочки из нескольких команд	24
Упражнение 3: Цикл со счетчиком	28
Упражнение 4: Бесконечное повторение	32
Упражнение 5: Условный оператор	36
РОББО Лаборатория	40
1. Программирование Лаборатории	42
Структура руководства	42
Дополнительный материал	42
Ход программирования при помощи Лаборатории	44
2. Использование Лаборатории как элемента управления игры	48
Упражнение 1: Управляем кнопками Лаборатории 1/3	52
Упражнение 2: Управляем кнопками Лаборатории 2/3	56
Упражнение 3: Управляем кнопками Лаборатории 3/3	60
Упражнение 4: Движение фигурки случайным образом	64
Упражнение 5: Поймайте фигурку!	68
3. Разработка игры	72
Упражнение 6: Подсчет очков	78
Упражнение 7: Завершение игры	82
Упражнение 8: Озвучивание игры	86
Упражнение 9: Увеличение скорости при помощи рычажка	90
РОББО Робоплатформа	94
1. Программирование Робоплатформы	96
Ход программирования с Робоплатформой	98
Датчики	104
Упражнение: Управляем человеком-роботом	105
2. Управляем Робоплатформой с помощью кнопок	106
Упражнение 1: Иди — стой	110
Упражнение 2: Управление стрелками	114
Упражнение 3: Закрепление знаний	118

3. Самостоятельное движение Робоплатформы.....	120
Упражнение 4: Обход препятствия	122
Упражнение 5: Обход коробки	126
Упражнение 6: Движение вдоль фигур	130
Упражнение 7: Закрепление знаний	134
4. Датчик касания	136
Упражнение 8: Включаем Робоплатформу датчиком касания	140
Упражнение 9: Столкновения с препятствием	144
Упражнение 10: Закрепление знаний	148
5. Датчик слежения за линией	150
Упражнение 11: Следует за линией	152
Упражнение 12: Пути и маршруты	156
Упражнение 13: Закрепление знаний	160
6. Датчик освещенности	162
Упражнение 14: Коробка с сюрпризом	164
Упражнение 15: Робоплатформа поворачивается к свету	168
Упражнение 16: Сторожевая собака	172
Упражнение 17: Закрепление знаний	176
7. Датчик расстояния и дополнительные упражнения	178
Упражнение 18: Робоплатформа путешествует	180
Упражнение 19: Применение знаний на практике	184

Используем Робоплатформу и Лабораторию совместно	186
Упражнение 1: Лаборатория как элемент управления Робоплатформой	192
Упражнение 2: Робоплатформа-охранник	196
Упражнение 3: Светофор	200
Упражнение 4: Применение навыков на практике	204

Приложения	206
Приложение 1. Выполненные упражнения, бланк	206
Приложение 2. Команды, использованные в упражнениях	208
Приложение 3. Понятия программирования	214

Введение

Настоящее руководство-самоучитель по использованию программы RobboScratch предназначено для всех, кто интересуется программированием и хочет преподавать программирование. Инструкция состоит из упражнений, протестированных на практике с преподавателями и учениками. Эти упражнения шаг за шагом развивают необходимые в программировании навыки. Во время упражнений дети учатся мыслить логически, с точки зрения разработчика программ, что необходимо во многих бытовых ситуациях. Такими навыками являются, в частности, наглядное представление общего, деление проблемы на мелкие части, разработка простой схемы действий. С помощью оборудования РОББО это происходит следующим образом:

1. Наглядное представление общего

В работе по программированию Робоплатформы и Лаборатории программист обдумывает, какого результата он хочет добиться.

2. Деление проблемы на мелкие части

Программист этап за этапом определяет события, с помощью которых сможет достичь желаемого результата.

3. Разработка программы

Программист пишет программу, используя команды выбранного языка программирования.

ЦЕЛЬ РУКОВОДСТВА

Целью руководства является обучение пользователей базовым знаниям программирования. С помощью задач по программированию и действий с РОББО пользователю прививают интерес к творческим процессам планирования, к работе в команде, к знаниям в области математики и естественных наук, к работе сверх учебных программ. На упражнениях по программированию полезно проводить беседы о том, где используется робототехника в современном обществе, как она влияет на жизнь человека и как ее можно будет использовать в будущем.

С помощью развиваемых на занятиях навыков пользователю прививается интерес к планированию и разработке инновационных решений, связанных с робототехникой. Цель обучения — научиться понимать и использовать технологии и с их помощью научиться придумывать новое.

Упражнения выполняются в парах, поскольку общение помогает лучше освоить полученные знания. На первых упражнениях ученики составляют программу по готовой модели, но по ходу дальнейших упражнений все больший простор оставляют собственным идеям ученика. Наставник оказывает поддержку ребенку при помощи направляющих вопросов, способствующих активному продолжению работы.

СТРУКТУРА РУКОВОДСТВА

В начале руководства изложены инструкции по подготовке РОББО Робоплатформы и Лаборатории к работе, а также даны пять упражнений по использованию среды программирования. Остальная часть руководства разделена на три блока: РОББО Лаборатория, РОББО Робоплатформа и Совместное использование Робоплатформы и Лаборатории. Пакеты упражнений по Робоплатформе и Лаборатории являются самостоятельными комплексами, при этом на первых упражнениях предполагается, что пользователь ранее не был знаком с данной программной средой. В третьей части, в свою очередь, предусматривается, что пользователь ранее работал и с одним, и с другим устройством, поскольку при выполнении упражнений данной части требуются хорошие навыки использования этих устройств.

В качестве введения к упражнениям для преподавателя в сжатом виде приводятся цели, содержание и используемые команды. Также подчеркиваются аспекты того или иного упражнения, наиболее важные с преподавательской точки зрения. Кроме того, предлагаются темы для бесед, которые можно обсуждать во время упражнений или после их выполнения. Упражнения составлены в форме карточек заданий, и их можно делать со всей группой вместе, в небольших группах с продвижением от задания к заданию или на индивидуальных занятиях, проводимых в невысоком темпе. Упражнения можно распечатать и раздать участникам группы, что позволит вести занятие дифференцированно, в соответствии с уровнем подготовки. Ход занятий можно контролировать с помощью таблицы, в которой будут отмечаться выполненные детьми и одобренные преподавателем карточки заданий ([Приложение 1](#)).

РАБОЧАЯ ГРУППА РУКОВОДСТВА

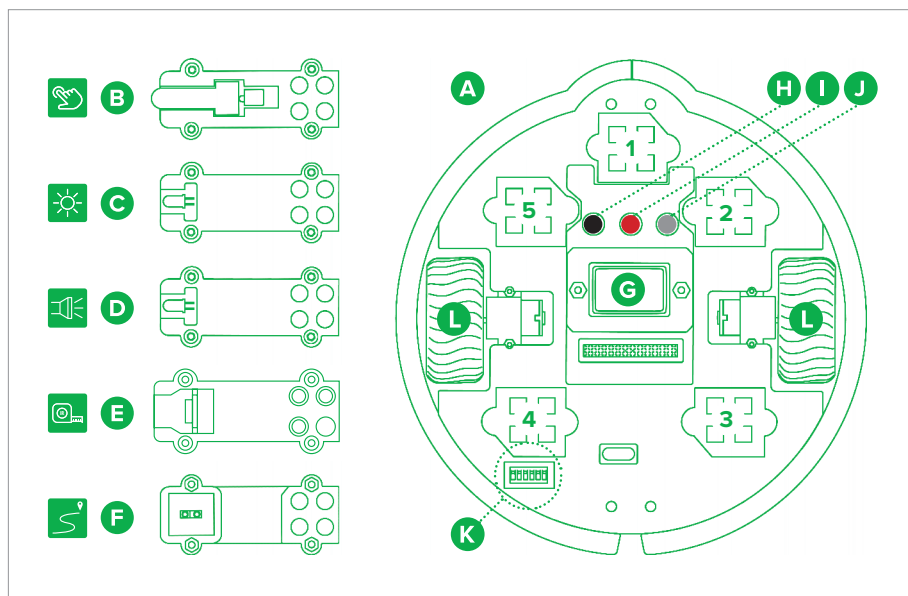
Руководство является результатом совместной работы ROBBO Europe и инновационной сети Innokas университета г. Хельсинки. Педагогическую структуру разработали Кати Сормунен и Томи Хуккалайнен. В составлении упражнений приняли участие также преподаватели сети Innokas Эрки Хаутала и Янне Холлаинен. Творческая проработка идей для упражнений проводилась в школах Vanttila г. Эспоо, Ohkola г. Мянсяля и Pohjois-Hervanna г. Тампере. Руководство содержит разработанные профессиональными педагогами упражнения, которые также были опробованы на практике с группами учеников в школе Viikin Normaalikoulu Университета г. Хельсинки и в школе Muijala г. Лохья.

Желаем весело провести занятия по программированию в компании с Робоплатформой и Лабораторией!

Авторы

Начните работу с Робоплатформой

ЧАСТИ РОБОПЛАТФОРМЫ



A. Робоплатформа

B. Датчик касания идентифицирует прикосновение.

C. Датчик освещенности определяет световой поток.

D. Светодиодный модуль дает свет.

E. Датчик расстояния определяет расстояние до объекта.

F. Датчик линии распознает светлую и темную поверхность.

G. Батарея питания

H. Кнопка Reset (Перезагрузка) перезагружает процессор.

I. Кнопка ON/OFF (Вкл./Выкл.) включает питание от батареи.

J. Кнопка Start (старт) — программируемая кнопка.

K. DIP-переключатели

В Робоплатформе имеются переключатели, регулирующие ее работу.

1. и 2. Энкодер колес (датчик угла поворота)

3. Защита от недостаточного напряжения батареи (используйте вместе с заряжаемой батареей)

4. Блокировка совместного использования USB-провода и батареи

5. Автоматическое время отключения (около 10 минут)

ВНИМАНИЕ! В большинстве случаев все переключатели рекомендуется держать в положении ON (Вкл.)

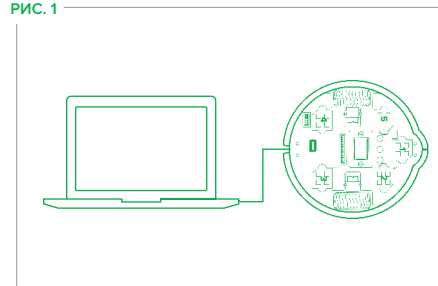
L. Колеса

СОВМЕСТИМОСТЬ

Загрузите программное обеспечение (ПО) для Робоплатформы (Windows, Mac, Linux): <http://files.robbo.world>

ФОРМИРОВАНИЕ СВЯЗИ

РИС. 1

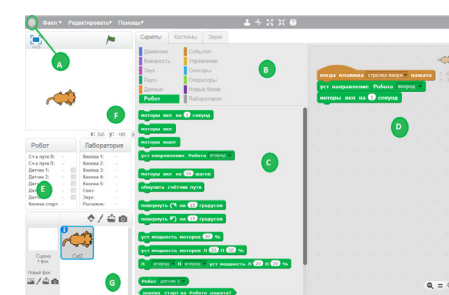


- Подключите один конец USB-кабеля к Робоплатформе, а другой — к компьютеру (РИС. 1).
- Откройте на компьютере ПО RobboScratch.
- Компьютер найдет Робоплатформу автоматически.
- Связь сформируется, когда кружок на панели управления станет зеленым. Если кружок не станет зеленым, нажмите кнопку Перепрошить (Upload firmware) на программном обеспечении прибора.

Bluetooth:

- Подсоедините к Робоплатформе источник питания (батарейку/аккумулятор).
- Включите Робоплатформу, нажав на кнопку ON/OFF.
- Активируйте Bluetooth на вашем компьютере и найдите оборудование поблизости.
- Выберите ROB-R-X-X-X и сформируйте пару (при необходимости используйте код 1234).
- Зеленый светодиодный индикатор Bluetooth будет мигать, пока модуль ожидает связи. При установленной связи светодиод будет гореть постоянно. Модуль Bluetooth находится книзу от сенсорного датчика №3.

ПРОГРАММИРОВАНИЕ



Программирование Робоплатформы (Робот) при помощи RobboScratch

A. Значок «земной шар»: смена языка.

B. Меню команд: здесь находятся группы команд.

C. Группа команд: команды отображаются здесь.

D. Зона программирования: перетяните команды в эту зону, разместив их одна под другой.

E. Значение датчиков можно проследить в этой зоне.

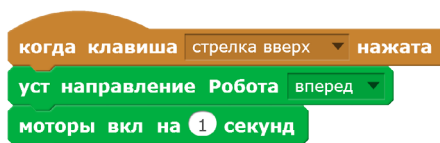
F. Сцена: в этой зоне перемещается Исполнитель (на рисунке — кошка).

G. Зона редактирования Героя и Сцены (фона). Здесь можно изменять и создавать новых героев и фоны.

ПРИВЕДИТЕ РОБОПЛАТФОРМУ В ДВИЖЕНИЕ

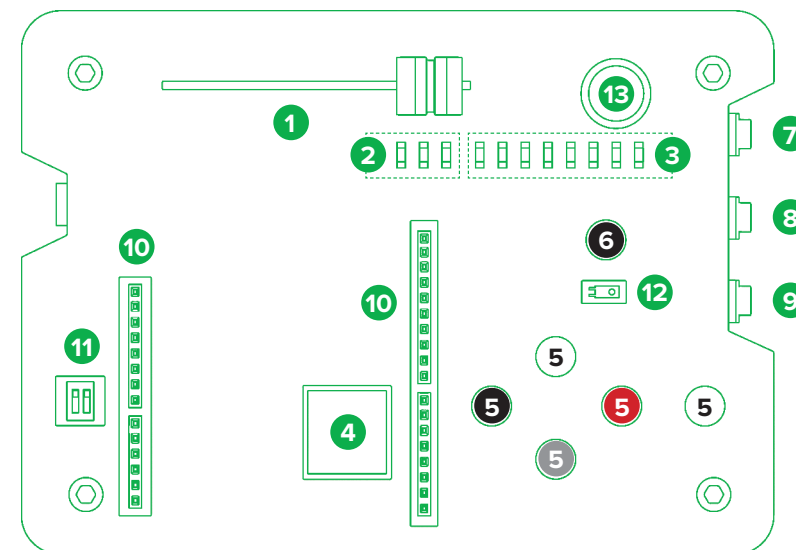
Протестируйте связь между компьютером и Робоплатформой, перемещая команды включения двигателя в зону программирования.

Убедившись в наличии связи, начинайте упражнения с Робоплатформой.



Начните работу с Лабораторией

ЧАСТИ ЛАБОРАТОРИИ



1. **Рычажок** (потенциометр скольжения).
2. **Три светодиода** (красный, желтый, зеленый).
3. **8 программируемых светодиодов.**
4. **Динамик.**
5. **Пять кнопок.**
6. **Кнопка перезагрузки Reset.**
7. **Разъем цифрового ввода/вывода D13:** порт для чтения и записи цифровой информации (0/1).
8. **Разъем аналогового входа A1:** порт для чтения аналоговой информации. Можно также использовать как цифровой вход.
9. **Разъем аналогового входа A0:** линия для чтения аналоговой информации. Можно также использовать как цифровой вход.
10. **Разъемы, интегрированные с Arduino Uno.**
11. **Переключатели DIP:**
 - 1-ON = Часть собственных компонентов лаборатории (№1, №3, №4, №12, №13) отсоединена, при этом Лабораторию

можно использовать как Arduino Uno. Кнопки и внешние разъемы (A0, A1, D13) работают. Могут быть задействованы все линии разъемов №10.

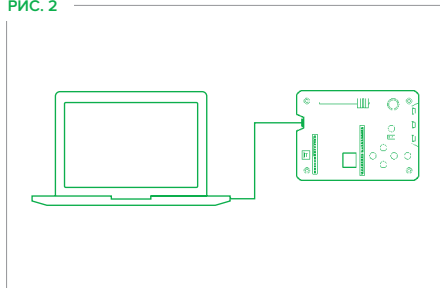
- 1-OFF = Все собственные компоненты лаборатории используются. Часть линий разъемов №10 подключена к собственным компонентам лаборатории.
 - 2-ON = Последовательная связь идет на линии разъема №10 (D0/RX, D1/TX). Контроллер USB (FTDI-плата) отключен.
 - 2-OFF = Последовательная связь идет через провод USB, FTDI-плата подключена.
- ВНИМАНИЕ!** При программировании переключатель необходимо держать в этом положении.
12. **Световой датчик** (фототранзистор).
 13. **Звуковой датчик** (микрофон).

СОВМЕСТИМОСТЬ

Загрузите программное обеспечение для Лаборатории (Windows, Mac, Linux):
<http://files.robbo.world>

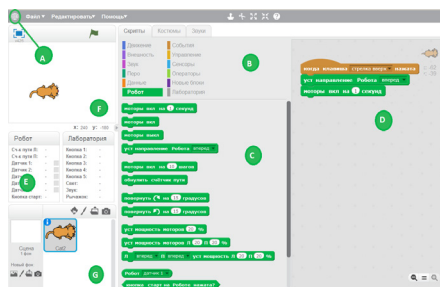
ФОРМИРОВАНИЕ СВЯЗИ

РИС. 2



- Один конец USB-кабеля подключите к Лаборатории, а другой — к компьютеру (РИС. 2).
- Откройте программу RobboScratch на своем компьютере.
- Компьютер найдет Лабораторию автоматически.
- Связь сформируется, когда кружок на панели управления станет зеленым.
- Если кружок не станет зеленым, нажмите кнопку Перепрошить на программном обеспечении прибора.

ПРОГРАММИРОВАНИЕ



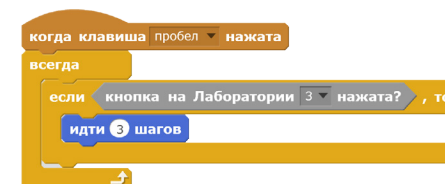
Программирование Лаборатории происходит при помощи программы RobboScratch следующим образом.

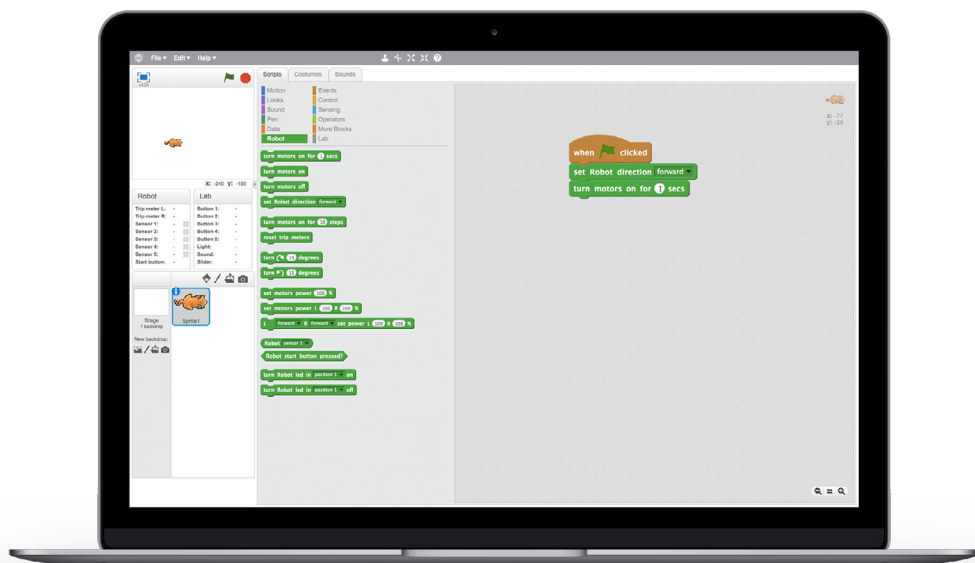
- А. Значок «Земной шар»:** смена языка.
- В. Меню команд:** здесь находятся группы команд.
- С. Группа команд:** команды отображаются здесь.
- Д. Зона программирования:** перетяните команды в эту зону одну под другую.
- Е. Значение датчиков и команд** можно проследить в этой зоне.
- Ф. Сцена:** в этой зоне перемещается Исполнитель (на рисунке — кошка).
- Г. Зона редактирования Героя и Сцены.** Здесь можно изменять и создавать новых героев и фоны.

ЗАПРОГРАММИРУЙТЕ ЛАБОРАТОРИЮ

Протестируйте связь между компьютером и Лабораторией, перемещая команды из блока Лаборатория в зону программирования.

Убедившись в наличии связи, начинайте упражнения с Лабораторией.





ИСПОЛЬЗОВАНИЕ ПРОГРАММНОЙ СРЕДЫ ROBBOSCRATCH

На предлагаемых упражнениях ученики осваивают программную среду RobboScratch. Упражнения по программированию начинаются с небольших заданий, целью которых является ознакомление пользователей с работой программного обеспечения. Выполнение упражнений прививает навыки использования программной среды RobboScratch.

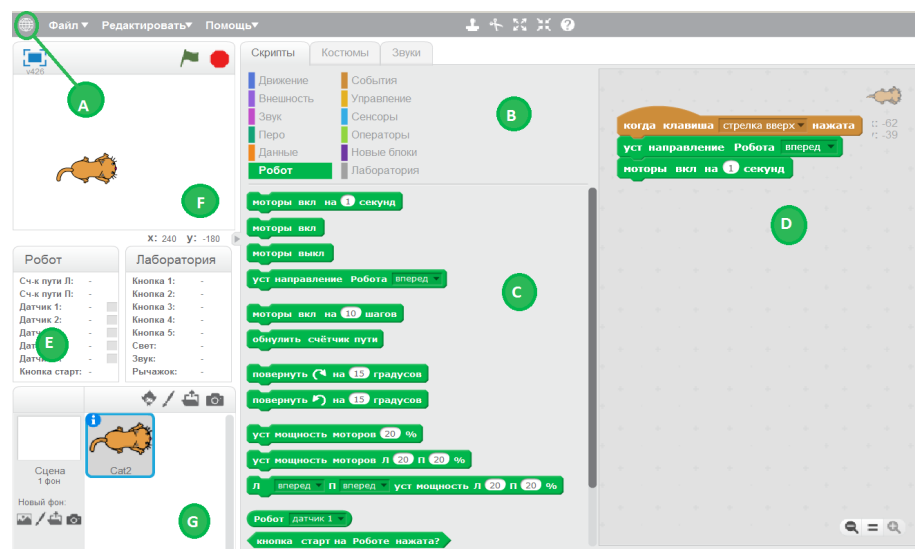
Из заранее подготовленных команд выстраивается соответствующая заданию программа. Упражнения начальной части заучиваются быстро, поэтому упражнения 1–4 рекомендуется выполнить за один раз.

Выполняя упражнения, ученики узнают, что происходит в программе при смене значений или команд.

Все изученное на упражнениях можно использовать в разработке собственной программы. Например, можно двигать не кошку, а какую-либо другую фигурку, имеющуюся в программе.

До выполнения упражнений можно вместе рассмотреть вопрос использования программного обеспечения.

ПРОГРАММИРОВАНИЕ ПРИ ПОМОЩИ ROBBOSCRATCH

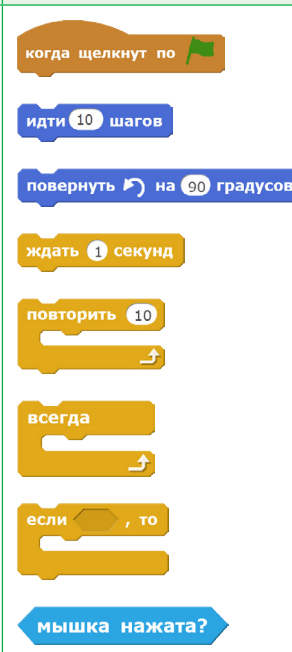


Героя при помощи программы RobboScratch программируют следующим образом.

- А. Значок «земной шар»:** смена языка.
- В. Меню команд:** здесь находятся группы команд.
- С. Группа команд:** отражаются команды.
- Д. Зона программирования:** перетяните команды в эту зону одну под другую.
- Е. Значение датчиков и команд** можно проследить в этой зоне.
- Ф. Сцена:** в этой зоне перемещается Исполнитель (на рисунке — кошка).
- Г. Зона редактирования Героя и Сцены.** Здесь можно изменять и создавать новых Героев или фоны.



ИСПОЛЬЗОВАНИЕ ПРОГРАММНОЙ СРЕДЫ ROBBOSCRATCH

Упражнение	Цели	Изучаемые команды и конструкции	Scratch	RobboScratch
1. Програмируем при помощи RobboScratch 2. Цепочки из нескольких команд 3. Цикл со счетчиком 4. Бесконечное повторение 5. Условный оператор	• Знакомство с программной средой • Изучение структуры программы и команд • Приведение героя на сцене в движение • Использование последовательных команд (линейная серия команд) • Изучение использования циклов (со счетчиком и бесконечного) и условных операторов с условием логического (булевого) типа или логическим выражением	• Начало программы • Приведение героя в движение • идти_шагов • повернуть на_градусов • Циклы • повторить_ • всегда • Условный оператор • если <_>, то • Условие • истина/ложь • ждать_секунд	 когда щелкнут по  идти 10 шагов повернуть на 90 градусов ждать 1 секунд повторить 10 всегда если  то мышка нажата?	

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 1: ПРОГРАММИРУЕМ ПРИ ПОМОЩИ ROBBOSCRATCH

Пользователи изучают программную среду и делают первую программу из двух команд. В этой программе фигурку кошки на сцене перемещают вправо.

Важно:

- Совместно изучить базовую работу программной среды.
- Изучить, что будет происходить при увеличении или уменьшении количества шагов, используемых в команде.

УПРАЖНЕНИЕ 2: ЦЕПочки ИЗ НЕСКОЛЬКИХ КОМАНД

Закрепление умения пользоваться программной средой. Выполнение более продолжительной серии команд.

Важно:

- Вместе изучить и подумать над тем, как меняется движение кошки, если
 - увеличить или уменьшить количество используемых в команде шагов.
- вместо пяти команд выполнить это же движение с помощью одной команды и со значением 50 (движение остается таким же).
- В каких ситуациях это можно применить?
- Найдите команду, с помощью которой кошка будет начинать движение всегда из одной точки в начале программы.

УПРАЖНЕНИЕ 3: ЦИКЛ СО СЧЕТЧИКОМ

Пользователи изучают цикл со счетчиком **повторить_**, а также новые команды, необходимые для движения.

Важно:

- Убедиться, что упражнения выполнены так, как предполагалось.
- Изучить и подумать вместе, как программа меняется, если порядок команд различается, отсутствует команда **ждать** или аргументы команд принимают различные значения.
- Менять величину угла поворота для создания фигур различной формы.

УПРАЖНЕНИЕ 4: БЕСКОНЕЧНОЕ ПОВТОРЕНИЕ

Закрепляется навык использования структуры повторения и изучается команда **всегда**.

Важно:

- В задании 2 используется материал, изученный в предыдущем упражнении.

УПРАЖНЕНИЕ 5: УСЛОВНЫЙ ОПЕРАТОР

Пользователи знакомятся с условными операторами, с используемыми в них условиями логического типа.

Важно:

- С помощью условного оператора задать работу программы по необходимым условиям.
- Ситуации, в которых используется условный оператор (выбор), можно придумать также самостоятельно из бытовых ситуаций. Например: если не чистить зубы, то будет кариес. Если не учить уроки, то...
- Логическое выражение или логическая переменная могут принимать значения **истина/ложь**. Детям не обязательно знать термин «логическое выражение», но оно является важной частью в реализации условных операторов.

ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

	Команда когда щелкнут по приводит в действие серию команд, прикрепленных ниже, если щелкнуть мышью по флажку.
	Команда идти_шагов двигает героя на сцене на заданное количество пикселей вперед (положительное значение) или назад (отрицательное значение).
	Команда повернуть на_градусов меняет направление движения героя на заданное количество градусов.
	Команда ждать_секунд останавливает выполнение серии команд на заданное время (на рисунке — на 1 секунду).
	Команда повторить_ повторяет расположенную внутри нее серию команд желаемое количество раз (на рисунке — 4 раза).
	Команда всегда повторяет расположенную внутри нее серию команд до завершения программы, т.е. с точки зрения программы — вечно.
	Условный оператор если <_>, то выполняет расположенную внутри него серию команд только в том случае, если логическая переменная или логическое выражения принимают значение истина .
	Условие мышка нажата? можно разместить внутри условного оператора. Условие принимает значение истина (клавиша мыши нажата) или ложь (клавиша мыши не нажата) в момент исполнения команды.



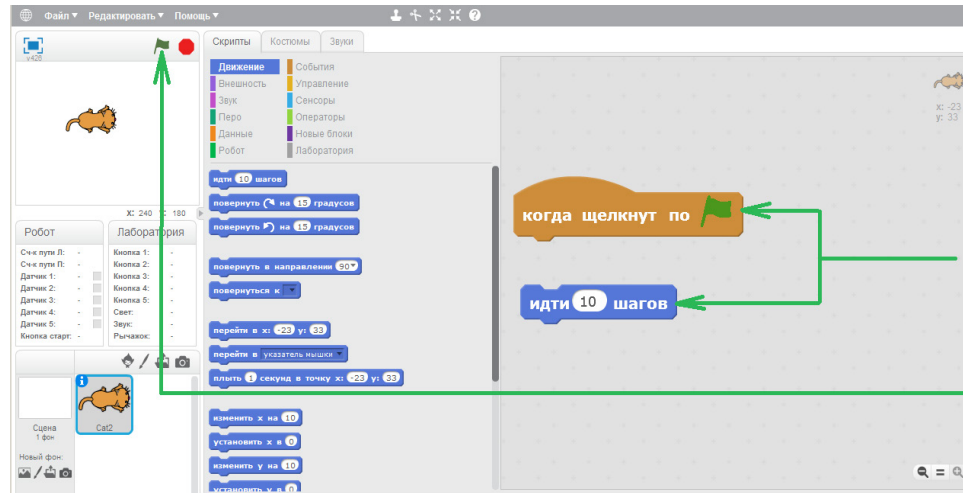
Упражнение 1: Програмируем при помощи RobboScratch

ЦЕЛИ

- Ознакомление с программной средой
- Разработка первой программы по заданным командам
- Изучение структуры программы

ЗАДАНИЕ

1. Найдите и объедините команды.
2. Щелкните по зеленому флажку.
3. Если сделаете все правильно, герой (кошка) будет двигаться вправо.
4. Перетащите кошку на сцене в первоначальное положение.



1. Объедините команды.

2. Щелкните по зеленому флажку.

ПРОСМОТР

Видеоролик об использовании программной среды:

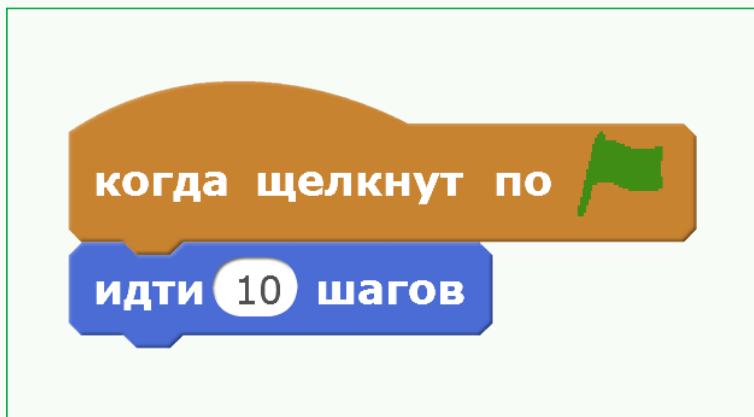
<http://bit.ly/robbolabohje>





Упражнение 1: Програмируем при помощи RobboScratch

РЕШЕНИЕ





Упражнение 2: Цепочки из нескольких команд

ЦЕЛИ

- Последовательное использование команд
- Закрепление навыков использования программной среды

ЗАДАНИЕ

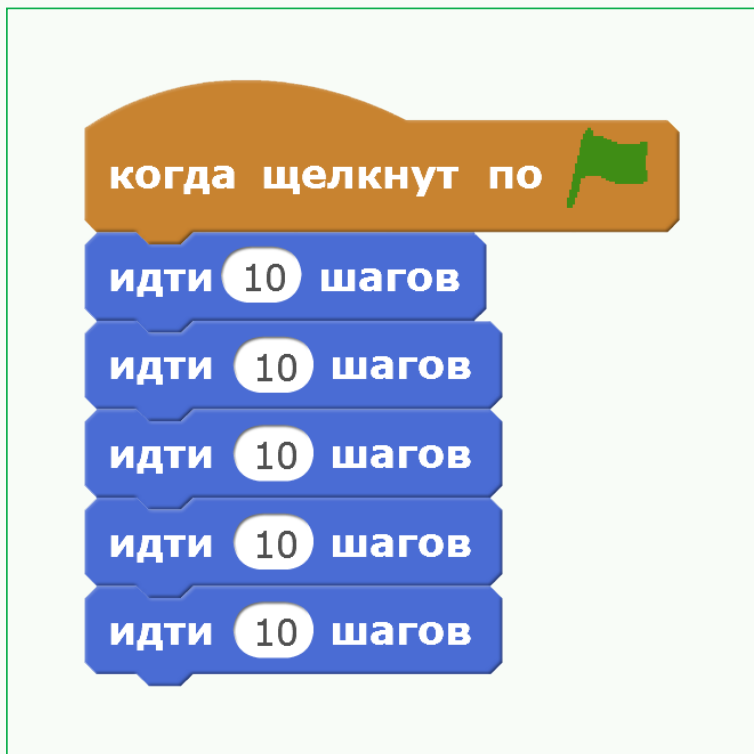
1. Найдите и объедините команды.
2. Щелкните по зеленому флажку.
3. Если сделаете все правильно, то кошка продвигается вправо на большее расстояние, чем в предыдущем упражнении.
4. Перетащите кошку на сцене в первоначальное положение.





Упражнение 2: Цепочки из нескольких команд

РЕШЕНИЕ





Упражнение 3: Цикл со счетчиком

ЦЕЛИ

- Использование в программе цикла со счетчиком
- Использование команды **ждать_секунд** и команды движения

ЗАДАНИЕ 1

1. Найдите и объедините команды.
2. Щелкните по зеленому флажку.
3. Если выполните все правильно, то кошка будет двигаться по маршруту в форме квадрата.
4. Перетащите кошку на сцене в первоначальное положение.

ЗАДАНИЕ 2

Запрограммируйте кошку так, чтобы она двигалась по маршруту в форме треугольника.



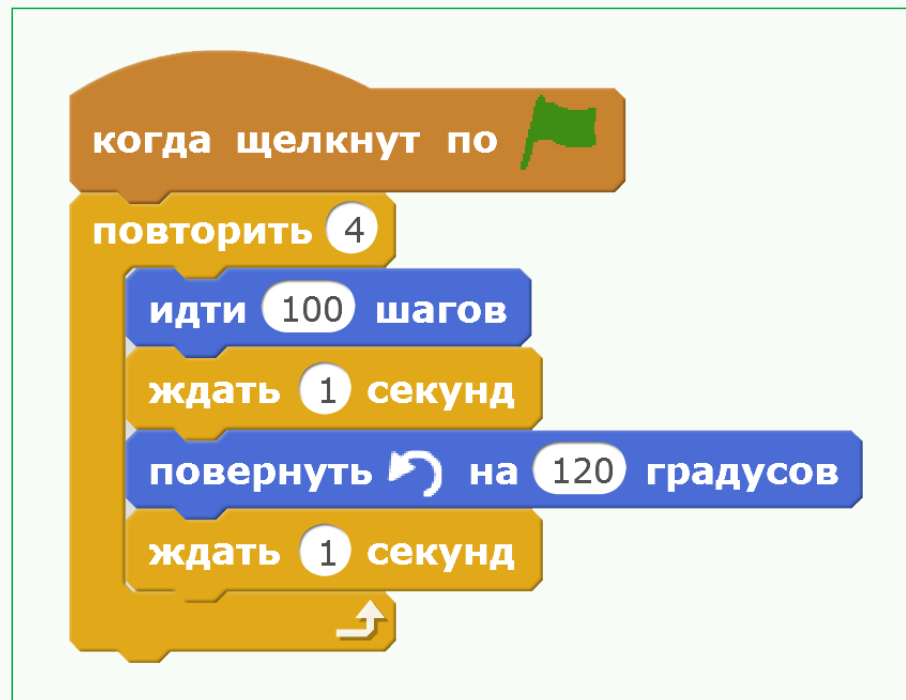


Упражнение 3: Цикл со счетчиком

РЕШЕНИЕ 1



РЕШЕНИЕ 2





Упражнение 4: Бесконечное повторение

ЦЕЛИ

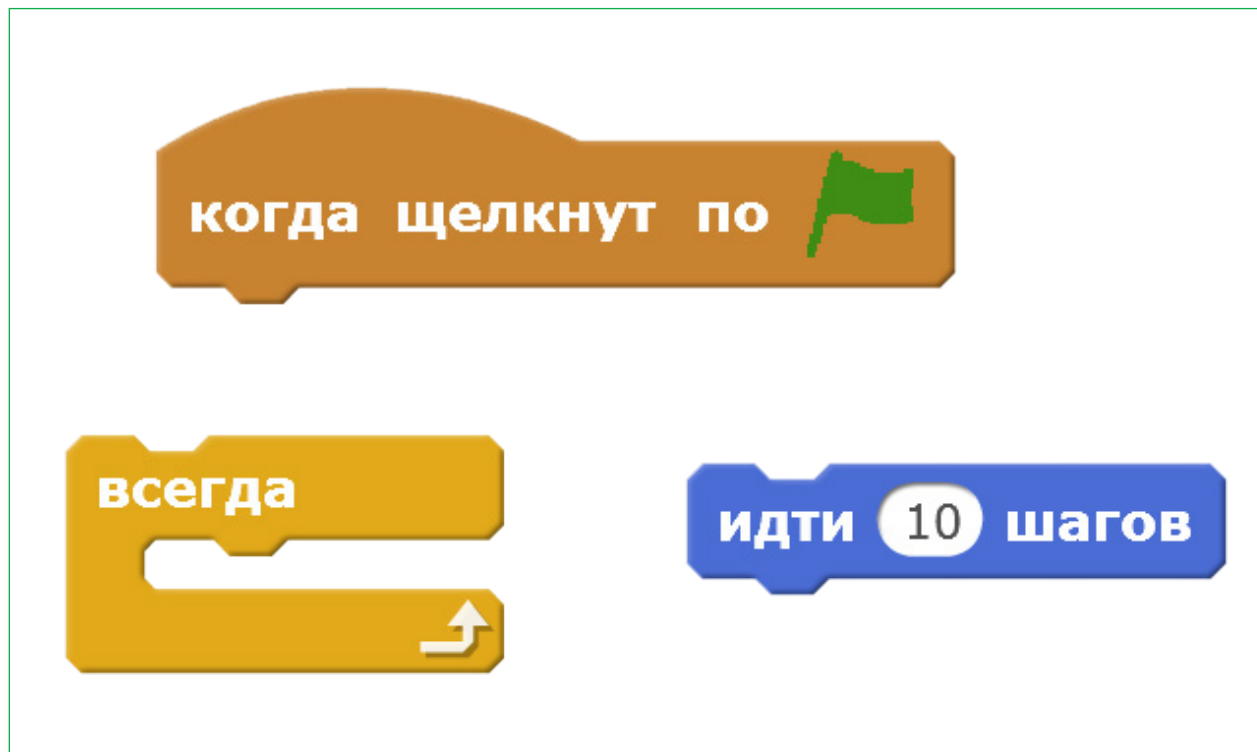
Закрепление навыков использования структуры повторения и различных команд движения

ЗАДАНИЕ 1

1. Найдите и объедините команды.
2. Щелкните по зеленому флажку.
3. Если выполните все правильно, то кошка остановится, когда попадет на край сцены.
4. Перетащите кошку на сцене в первоначальное положение.

ЗАДАНИЕ 2

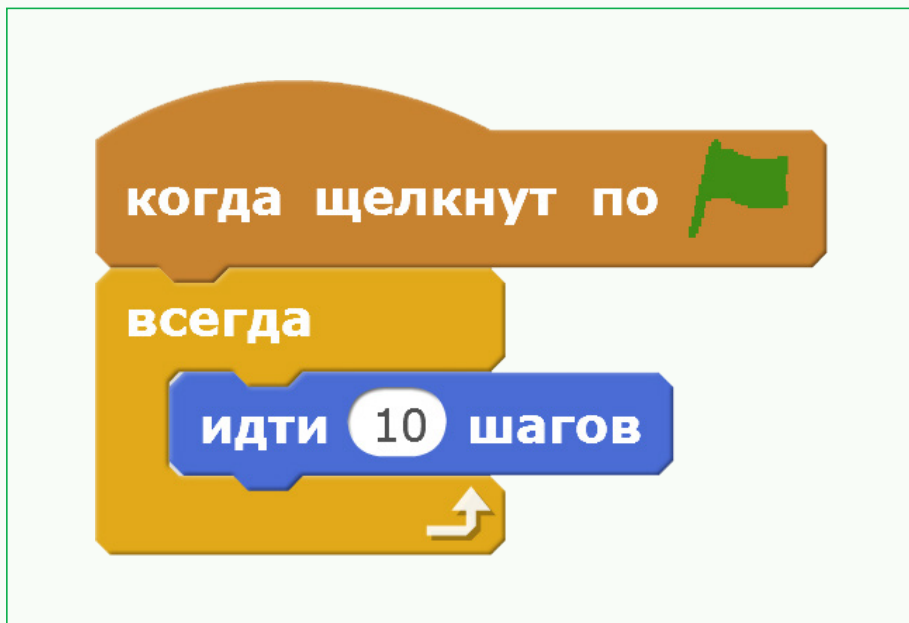
Запрограммируйте кошку так, чтобы она двигалась по кругу бесконечно.





Упражнение 4: Бесконечное повторение

РЕШЕНИЕ 1



РЕШЕНИЕ 2





Упражнение 5: Условный оператор

ЦЕЛИ

- Использование условного оператора **если <_>, то**
- Использование условий, принимающих значения **истина/ложь**

ЗАДАНИЕ

1. Найдите и объедините команды.
2. Щелкните по зеленому флажку.
3. Если выполните все правильно, то кошка будет двигаться направо при каждом нажатии клавиши мыши.
4. Перетащите кошку на сцене в первоначальное положение.

ЭКСПЕРИМЕНТ!

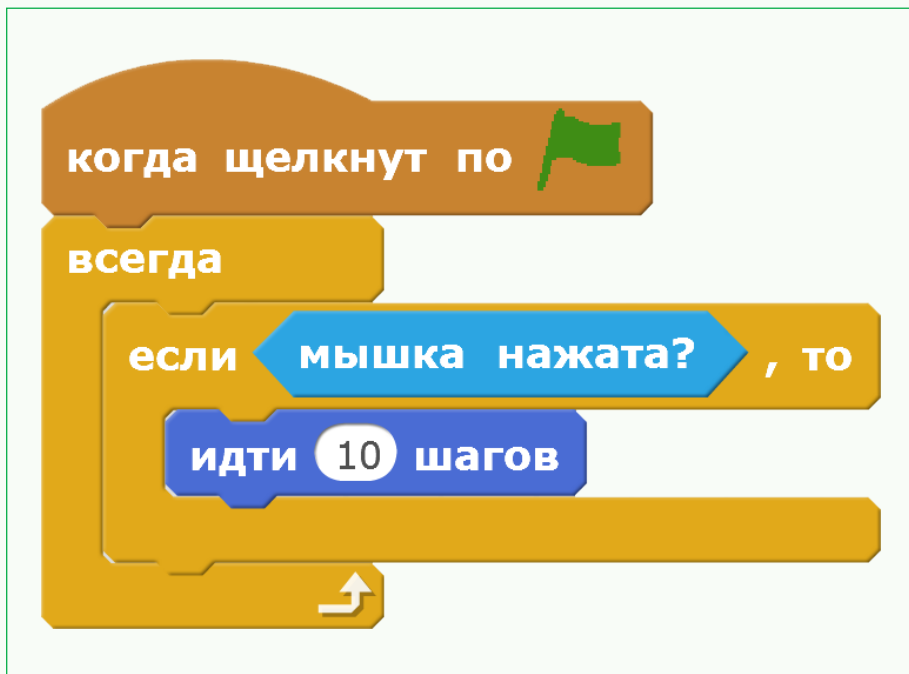
Измените порядок команд. Что вы заметили?



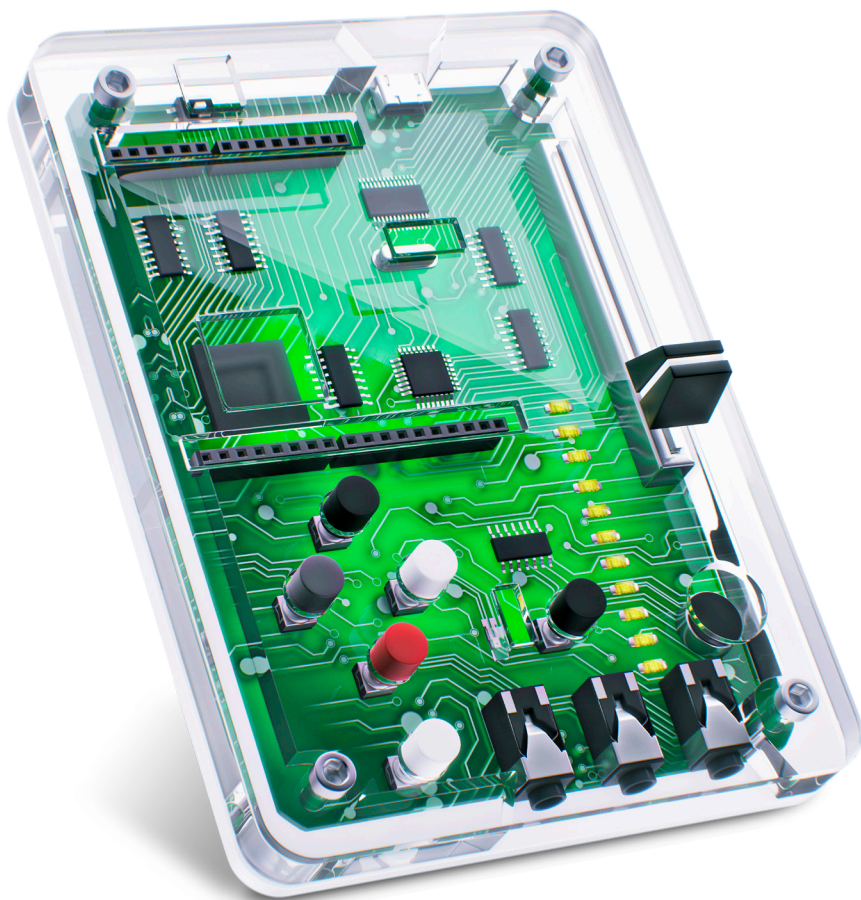


Упражнение 5: Условный оператор

РЕШЕНИЕ



РОББО ЛАБОРАТОРИЯ



ПРОГРАММИРОВАНИЕ ЛАБОРАТОРИИ

ИСПОЛЬЗОВАНИЕ ЛАБОРАТОРИИ КАК ЭЛЕМЕНТА УПРАВЛЕНИЯ ИГРЫ

Упражнение	Цели	Изучаемые команды и структуры	Scratch	RobboScratch
1. Управляем кнопками Лаборатории 1/3 2. Управляем кнопками Лаборатории 2/3 3. Управляем кнопками Лаборатории 3/3 4. Движение фигурки случайным образом 5. Поймайте фигурку!	- Углубление ранее полученных навыков - Разработка более длинных программ - Знакомство с принципами работы Лаборатории - Использование условных операторов в программах - Использование в командах отрицательных значений и случайных чисел - Рисование линии на сцене	- Работа лаборатории - Кнопки - Случайное число - Движение - Измените значение Y и значение X - Плыть в пункт X и Y - Опустить перо - Внешний вид - Сообщения - Звуки - Сыграйте звук - Условие - Истина — ложь		

РАЗРАБОТКА ИГРЫ

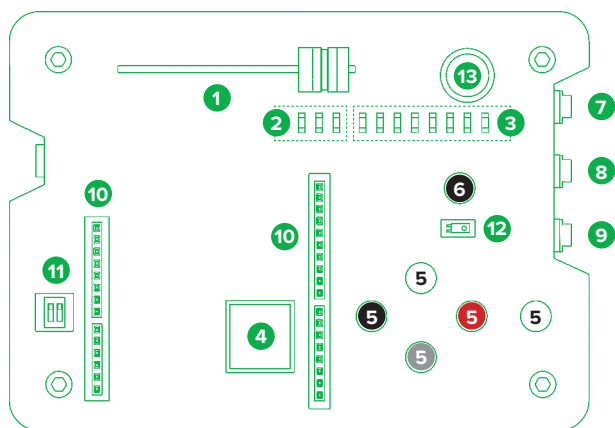
Упражнение	Цели	Изучаемые команды и структуры	Scratch	RobboScratch
6. Подсчет очков 7. Завершение игры 8. Озвучивание игры 9. Увеличение скорости при помощи рычажка	- Закрепление навыков использования цикла - Использование переменной величины - Знакомство с принципами работы датчиков (сенсоров) - Использование в программах условного оператора и условий - Внесение в программы звука	- Переменные величины - Управление - Остановка - Условный оператор - Если — то — иначе - Работа Лаборатории - Сыграй ноту - Рычажок и датчики		



2. Использование Лаборатории как элемента управления игры

После освоения основных функций программного обеспечения приступают к освоению РОББО Лаборатории. Кнопки, расположенные в Лаборатории (РИС. 2, ПУНКТ 5), используются для управления движениями кошки на экране и добавления первого игрового элемента в программу.

РИС. 2



1. Рычажок (потенциометр).
2. Три светодиода (красный, желтый, зеленый).
3. Программируемая матрица из восьми светодиодов.
4. Динамик.
5. Пять кнопок.
6. Кнопка Reset.
7. Вход D13: вход для чтения и записи цифровой информации (0/1).
8. Вход A1: вход для чтения аналоговой информации (0–100). Можно также использовать как цифровой вход.
9. Вход A0: вход для чтения аналоговой информации (0–100). Можно также использовать как цифровой вход.
10. Разъемы, интегрированные с Arduino Uno.
11. Переключатели DIP:
 - 1-ON = Часть собственных компонентов Лаборатории (№1, №3, №4, №12, №13)

При программировании продолжают использовать циклы и условные операторы, а также знакомятся с понятием случайного числа. Также закрепляются навыки использования логических (булевых) выражений.

отсоединена, при этом Лабораторию можно использовать как Arduino Uno. Кнопки и внешние разъемы (A0, A1, D13) работают. Могут быть задействованы все линии разъемов №10.

- 1-OFF = Используются все собственные компоненты Лаборатории. Часть линий разъемов №10 подключена к собственным компонентам Лаборатории.
- 2-ON = Последовательная связь идет на линии разъема №10 (D0/RX, D1/TX). Контроллер USB (FTDI-плата) отключен.
- 2-OFF = Последовательная связь идет через провод USB, FTDI-плата подключена.

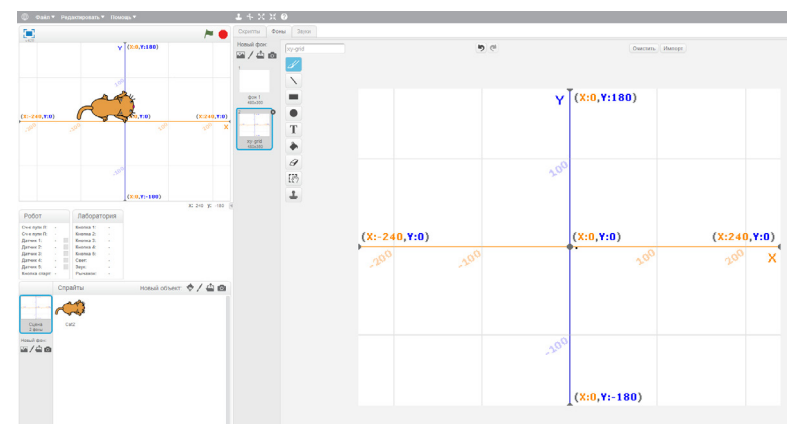
ВНИМАНИЕ! При программировании переключатель необходимо держать в этом положении.

12. Световой датчик (фототранзистор).

13. Звуковой датчик (микрофон).

В приведенных упражнениях задействованы принципы работы системы координат при программировании фигурки (РИС. 3). В карточках заданий не дается отдельного обучения редактированию фона, поскольку после создания новой фигурки этот навык появляется естественным образом. В RobboScratch в качестве фона можно установить сетку координат, которую можно параллельно изучать при программировании:

РИС. 3



1. Активируйте фон в зоне редактирования Сцены (синяя окантовка).
2. Выберите фон ху-grid из библиотеки программы.
3. Назад в область программирования можно перейти, выбрав вкладку Скрипты.

Пройденный на упражнениях материал можно использовать при создании собственной программы. Например, вместо кошки можно двигать какую-либо иную выбранную в программе фигурку.

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 1: УПРАВЛЯЕМ КНОПКАМИ ЛАБОРАТОРИИ 1/3

Пользователи учатся использовать условия, принимающие значения **истина/ложь**, в условном операторе и управлять фигуркой сцены при помощи кнопок Лаборатории.

Важно:

- Научиться находить правильную кнопку путем слежения за изменениями значений истины (**истина/ложь**) и использовать наблюдения в разработке программы.

УПРАЖНЕНИЕ 2: УПРАВЛЯЕМ КНОПКАМИ ЛАБОРАТОРИИ 2/3

Закрепляют полученные знания путем их использования. Используют отрицательные значения в команде **идти_шагов**.

Важно:

- Вместе изучать и размышлять над тем, как меняется движение кошки при увеличении или уменьшении значений команд.

УПРАЖНЕНИЕ 3: УПРАВЛЯЕМ КНОПКАМИ ЛАБОРАТОРИИ 3/3

Учатся рисовать линию на сцене во время движения фигурки.

Важно:

- Убедиться в том, что упражнения выполнены так, как было запланировано.
- Вместе изучать и размышлять над тем, как меняется движение кошки при увеличении или уменьшении значений в командах.

УПРАЖНЕНИЕ 4: ДВИЖЕНИЕ ФИГУРКИ СЛУЧАЙНЫМ ОБРАЗОМ

Учатся создавать новую фигурку, изменять ее и использовать случайные числа в качестве переменного значения.

Важно:

- Закрепление использования координат.
- Понимание понятия «случайное число»: Случайным числом называется случайно выбранное число из определенной области чисел. В данной программе в качестве границ области чисел используются максимальные значения координат.

УПРАЖНЕНИЕ 5: ПОЙМАЙТЕ ФИГУРКУ!

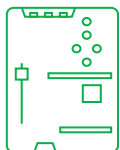
Закрепление использования случайных чисел и условного оператора в программе; повторение того, как управлять фигуркой с помощью кнопок Лаборатории.

Важно:

- В задании 2 пользователи учатся управлять командой **измените Y на_** вместо команды **идти_шагов**.
- Первая команда меняет положение фигурки на заданное в горизонтальном направлении количество шагов, а вторая команда двигает фигурку вперед в том направлении, куда фигурка смотрит в тот момент.
- В карточке задания предлагаются различные звуки для озвучивания. Можно озвучить фигурку также самостоятельно.

ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

	Команда когда щелкнут по приводит в действие серию команд, прикрепленных внизу, если щелкнуть мышью по флажку.
	Команда всегда повторяет расположенную внутри нее серию команд до завершения программы, т.е. с точки зрения программы — вечно.
	Команда если <_>, то (условный оператор) выполняет расположенную внутри нее серию команд только в том случае, если значение истинности условия, заданного для команды на момент выполнения команды, истина .
	Условие кнопка на Лаборатории_нажата говорит о том, нажата ли определенная кнопка Лаборатории в запрашиваемый момент. Условие можно поместить, например, в условный оператор.
	Команда идти_шагов двигает фигурку на сцене на заданное количество точек вперед (положительное значение) или назад (отрицательное значение).
	Команда изменить Y на_ изменяет место фигурки по оси Y на заданное количество пикселей.
	Команда изменить X на_ изменяет место фигурки по оси X на заданное количество пикселей.
	После команды опустить перо при движении фигурки на сцене вырисовывается линия.
	Команда плыть_секунд в точку X:_Y:_ сдвигает фигурку в системе координат в заданное место в течение заданного времени.
	Выражение выдать случайное от_до_ случайным образом выбирает число из заданного диапазона.
	Команда говорить_в течение_секунд создает фигурке текстовую выноску на заданный промежуток времени. Текст, расположенный в белой зоне (S), можно редактировать произвольно.
	Команда играть звук_ воспроизводит выбранный из меню звук.
	Условие касается_? размещают внутри условного оператора. Условие получает значение истина (фигурка касается) или ложь (фигурка не касается) на момент исполнения команды.



Упражнение 1: Управляем кнопками Лаборатории 1/3

ЦЕЛИ

- Управление фигуркой на сцене при помощи кнопок Лаборатории
- Использование значения истинности **если — то** в условном выражении

ПОДСКАЗКА

С помощью нажатия кнопок Лаборатории и наблюдения за меняющимися значениями вы сможете выяснить, что собой представляет Кнопка 3 (**ложь/истина**).

ЗАДАНИЕ

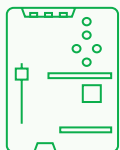
1. Выясните, что представляет собой **Кнопка 3** Лаборатории.
2. Найдите и объедините команды.
3. Нажмите правую кнопку Лаборатории.
4. Если сделаете все правильно, то при нажатии кнопки Лаборатории кошка будет двигаться вправо.
5. Перетащите кошку на сцене в первоначальное положение.

ЭКСПЕРИМЕНТ!

Попробуйте изменить порядок команд. Что вы заметили при этом?

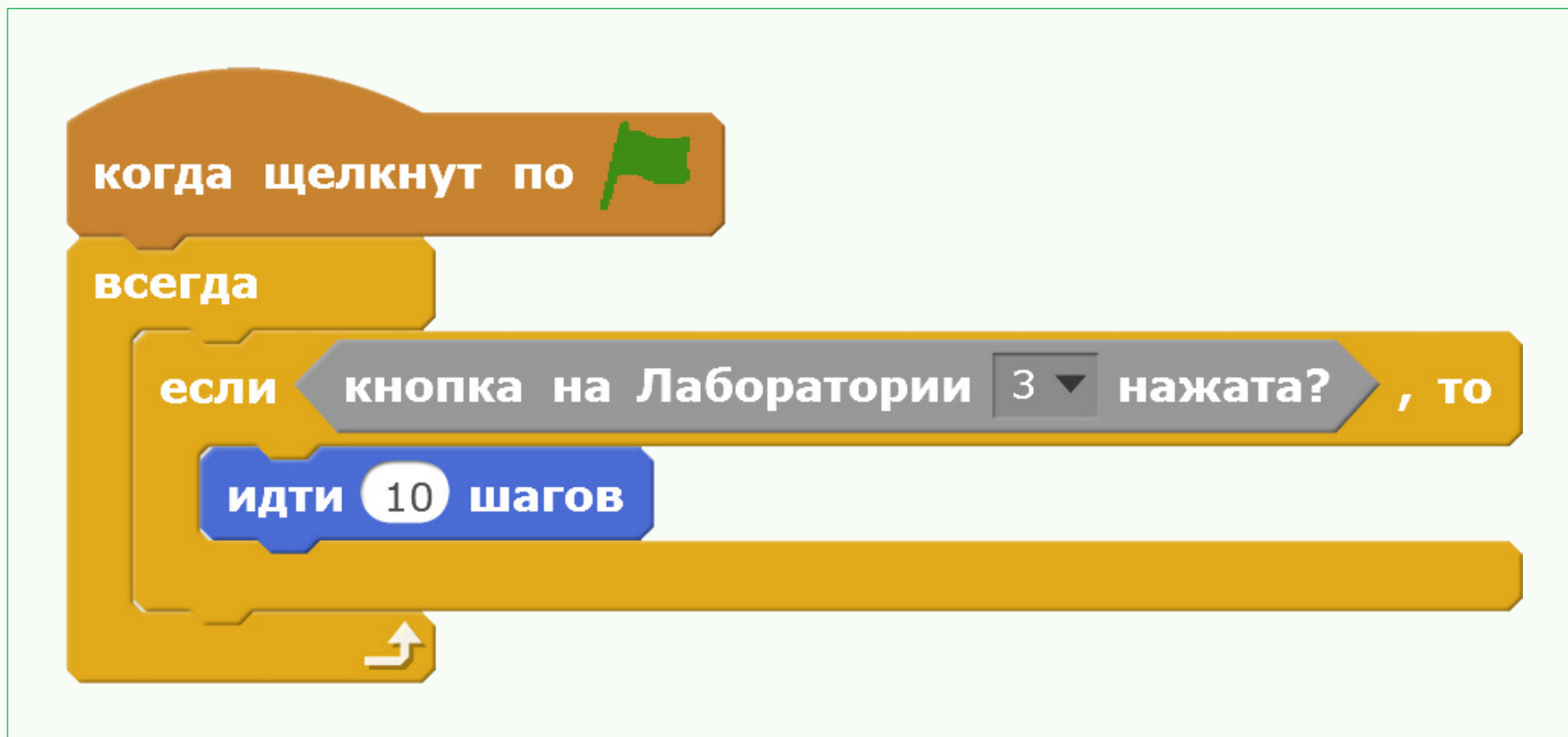
Лаборатория	
Кнопка 1:	ложь
Кнопка 2:	ложь
Кнопка 3:	ложь
Кнопка 4:	ложь
Кнопка 5:	ложь
Свет:	61
Звук:	10
Рычажок:	62

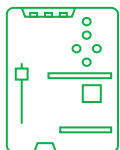




Упражнение 1: Управляем кнопками Лаборатории 1/3

РЕШЕНИЕ





Упражнение 2: Управляем кнопками Лаборатории 2/3

ЦЕЛИ

- Перемещение фигурки на сцене влево и вправо кнопками Лаборатории
- Использование отрицательных значений в команде

ЗАДАНИЕ

1. Выясните, что собой представляют **кнопки 1** и **3** Лаборатории. Попробуйте изменить порядок команд.
2. Найдите и объедините команды.
3. Нажмите на правую кнопку Лаборатории.
4. Если сделаете все правильно, то кошка будет по одной кнопке двигаться вправо, а по другой кнопке — влево.
5. Перетащите кошку на сцене в первоначальное положение.

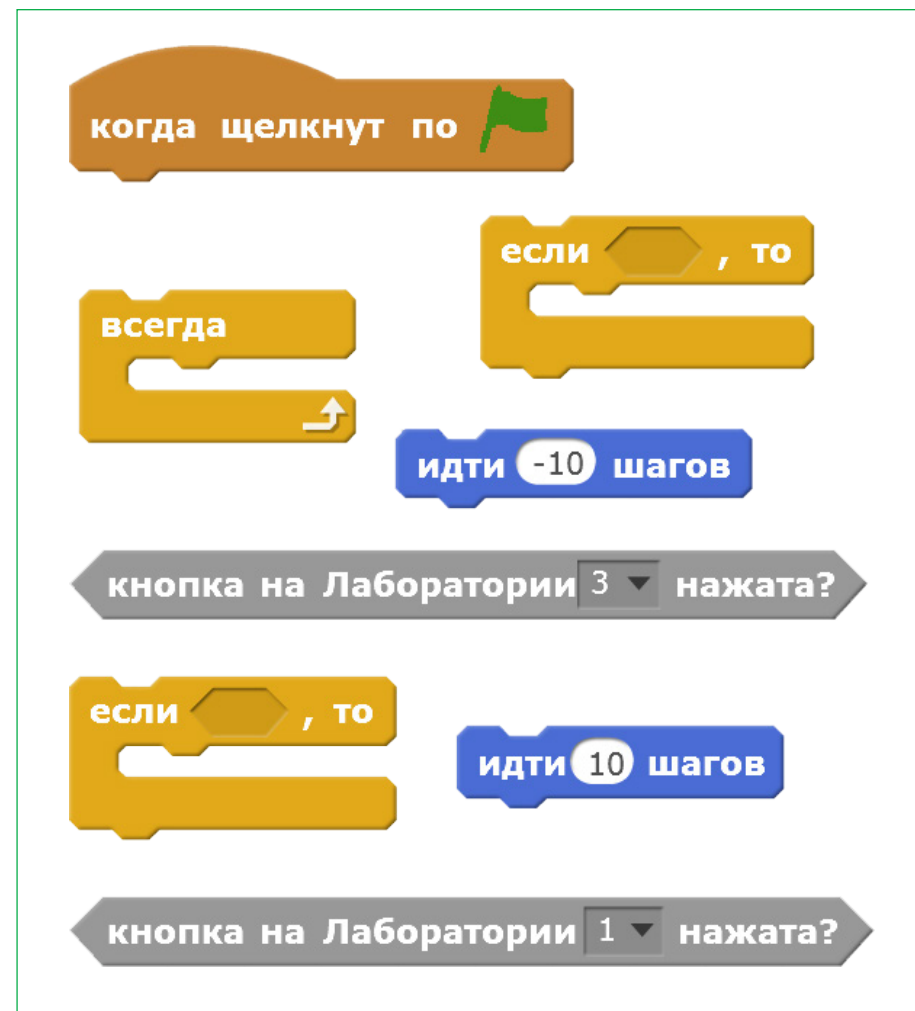
ПОДСКАЗКА

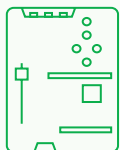
С помощью нажатия кнопок Лаборатории и слежения за изменяющимися значениями вы сможете выяснить, что представляют собой **кнопки 1** и **3** (**ложь/истина**).

ЭКСПЕРИМЕНТ!

Попробуйте изменить порядок команд.

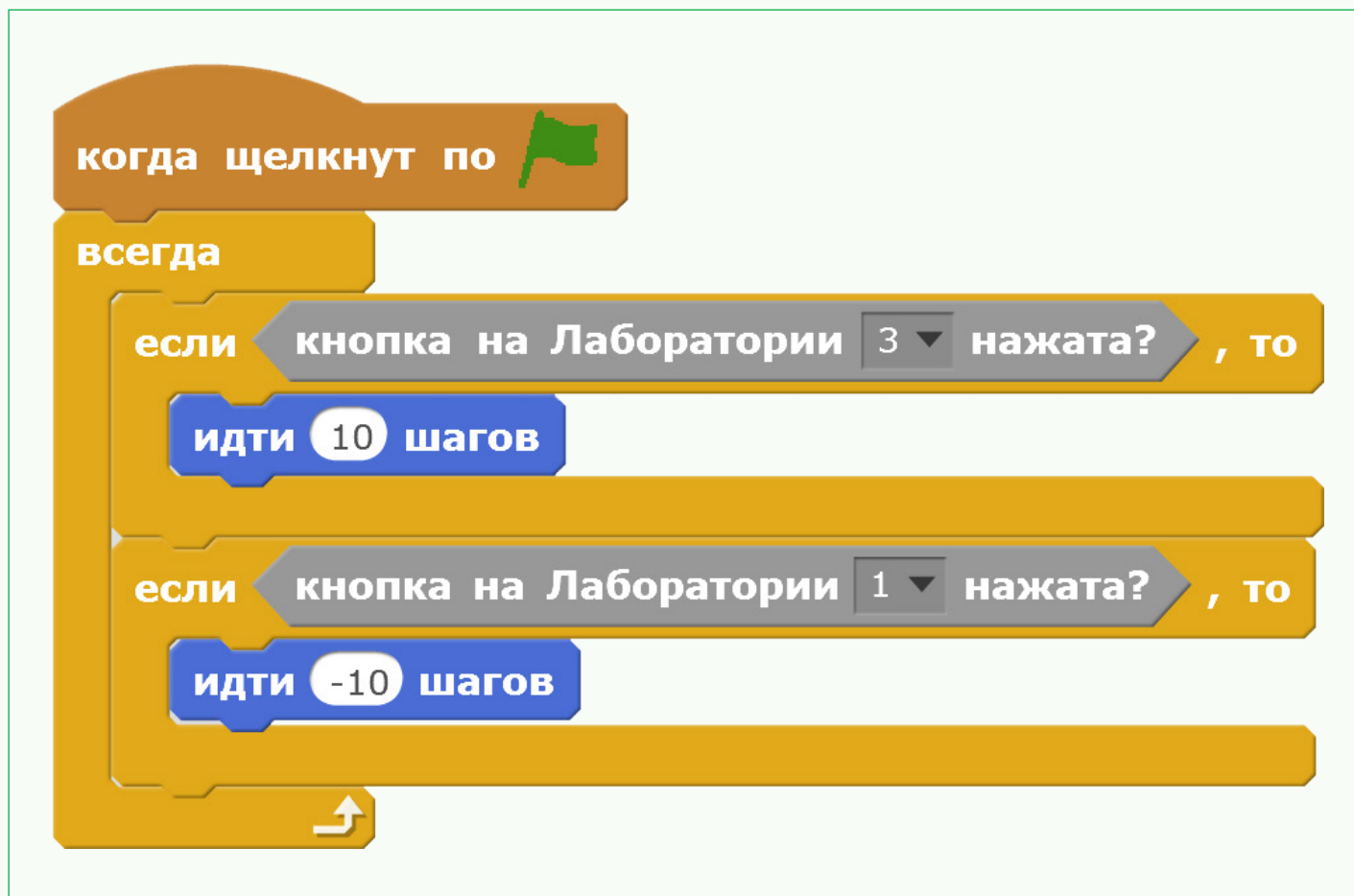
Лаборатория	
Кнопка 1:	ложь
Кнопка 2:	ложь
Кнопка 3:	ложь
Кнопка 4:	ложь
Кнопка 5:	ложь
Свет:	61
Звук:	10
Рычажок:	62

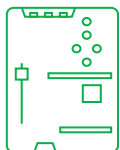




Упражнение 2: Управляем кнопками Лаборатории 2/3

РЕШЕНИЕ





Упражнение 3: Управляем кнопками Лаборатории 3/3

ЦЕЛИ

- Перемещение фигурки на сцене влево, вправо, вверх и вниз с помощью кнопок Лаборатории
- Использование в команде значений по вертикальной оси Y

ЗАДАНИЕ 1

1. Выясните, что представляют собой кнопки Лаборатории 1, 2, 3 и 4.
2. Найдите и объедините команды.
3. Нажимайте кнопки Лаборатории.
4. Если сделаете все правильно, то по нажатию кнопок кошка будет двигаться вправо, влево, вверх и вниз.
5. Убедитесь, что изначально кошка смотрит вправо.

ЗАДАНИЕ 2

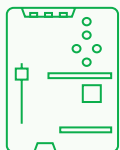
Добавьте команду **опустить перо** из группы команд **перо**.

опустить перо

ЭКСПЕРИМЕНТ!

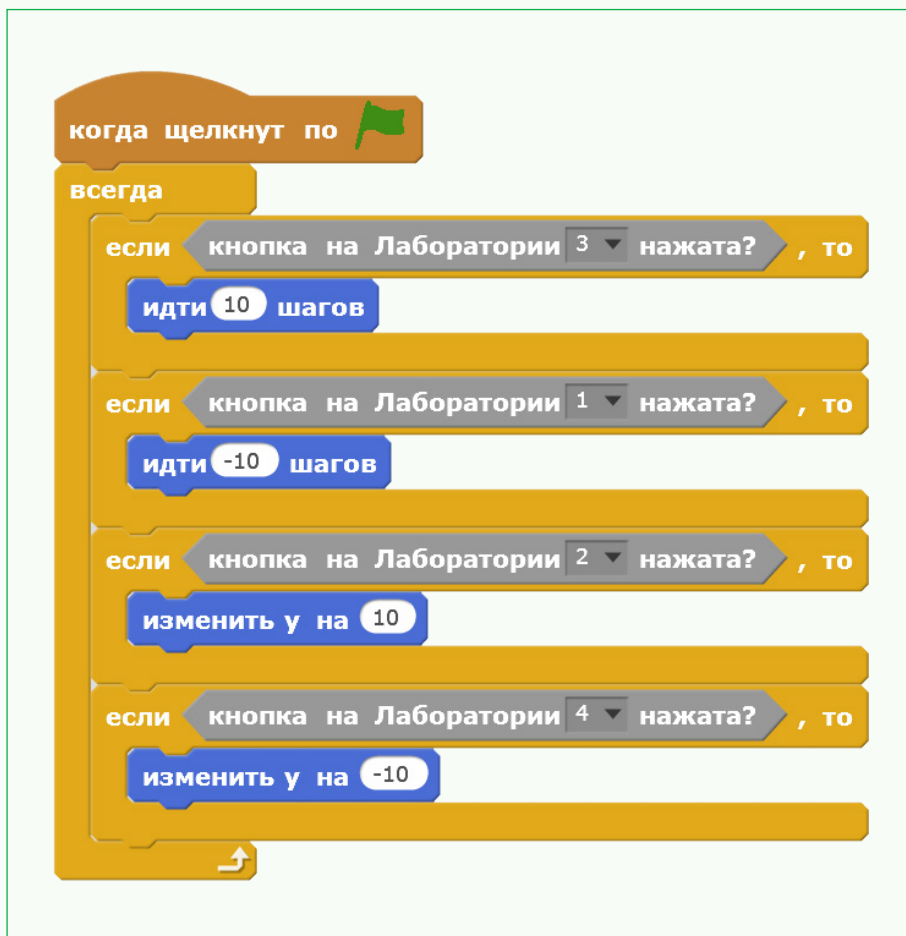
Попробуйте изменить порядок команд. Что вы заметили при этом?





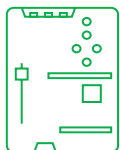
Упражнение 3: Управляем кнопками Лаборатории 3/3

РЕШЕНИЕ 1



РЕШЕНИЕ 2





Упражнение 4: Движение фигурки случайным образом

ЦЕЛИ

- Создание второй фигурки в программе
- Управление произвольным движением фигурки по сцене
- Закрепление навыков управления значениями осей X и Y

ЗАДАНИЕ

1. Создайте новую фигурку нажатием кнопки **Выберите из библиотеки** (РИС. 1).
2. Убедитесь, что созданная фигурка активна (РИС. 2, на примере мышка обведена линией синего цвета).
3. Смените название фигурки в синем кружке «i», расположенном в левом верхнем углу (РИС. 2).
4. Найдите и объедините команды.
5. Нажмите на зеленый флажок.
6. Если сделаете все правильно, то фигурка будет двигаться по сцене произвольно.

ЭКСПЕРИМЕНТ!

Измените значения случайных чисел. Что вы заметили при этом?

РИС. 1

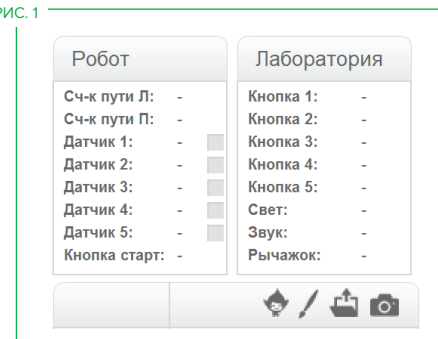
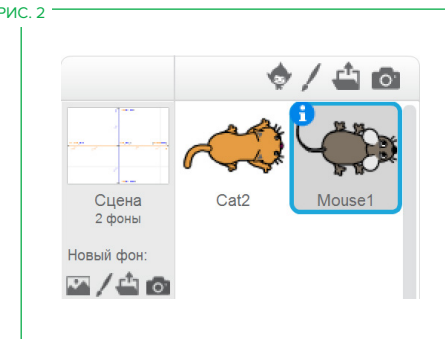


РИС. 2



когда щелкнут по

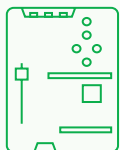
всегда

плыть 0.5 секунд в точку x: 10 y: 10

выдать случайное от -240 до 240

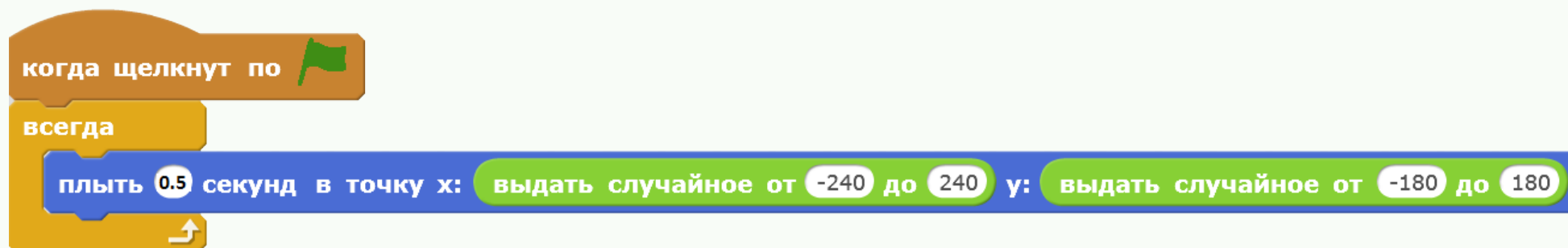
выдать случайное от -180 до 180

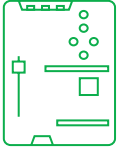




Упражнение 4: Движение фигурки случайным образом

РЕШЕНИЕ





Упражнение 5: Поймите фигурку!

ЦЕЛИ

- Закрепление навыков использования случайных чисел и условного оператора в программе
- Управление движением фигурки по сцене с помощью Лаборатории

ЗАДАНИЕ 1

1. Создайте вторую фигурку (например, мышку).
2. Найдите и объедините команды.
3. Нажмите на зеленый флажок.
4. Если выполните все правильно, мышка произнесет: «Меня поймали!», когда кошка поймает ее.

ЗАДАНИЕ 2

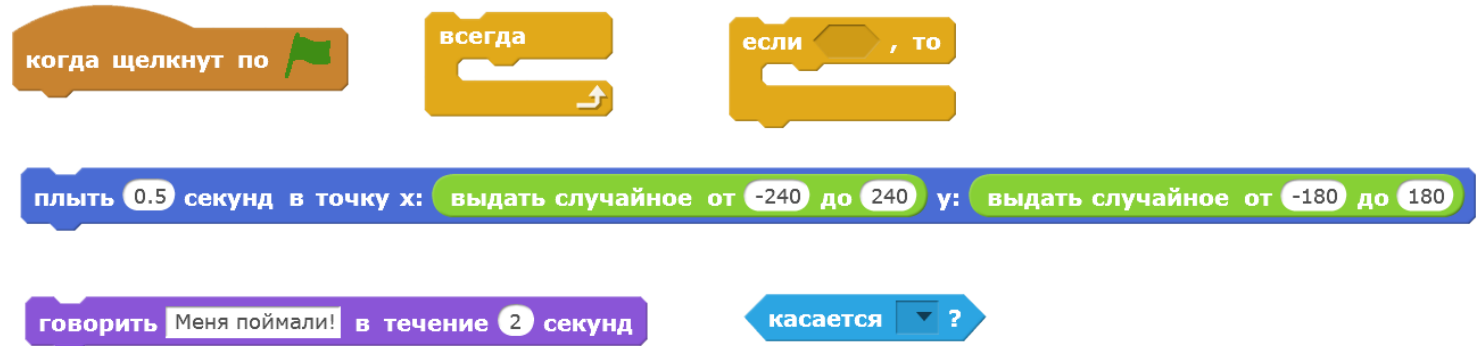
- Запрограммируйте Лабораторию так, чтобы она работала как элемент управления кошки (см. упражнение 3).
- Попробуйте поймать мышку, передвигая кошку с помощью Лаборатории.
- Используйте в программе эту команду.

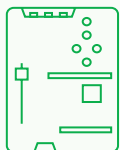
изменить x на 10

ЭКСПЕРИМЕНТ!

- Что еще может сказать мышка? Измените текст речевой выноски.
- Добавьте звук в конце программы. Можете записать собственные звуки.

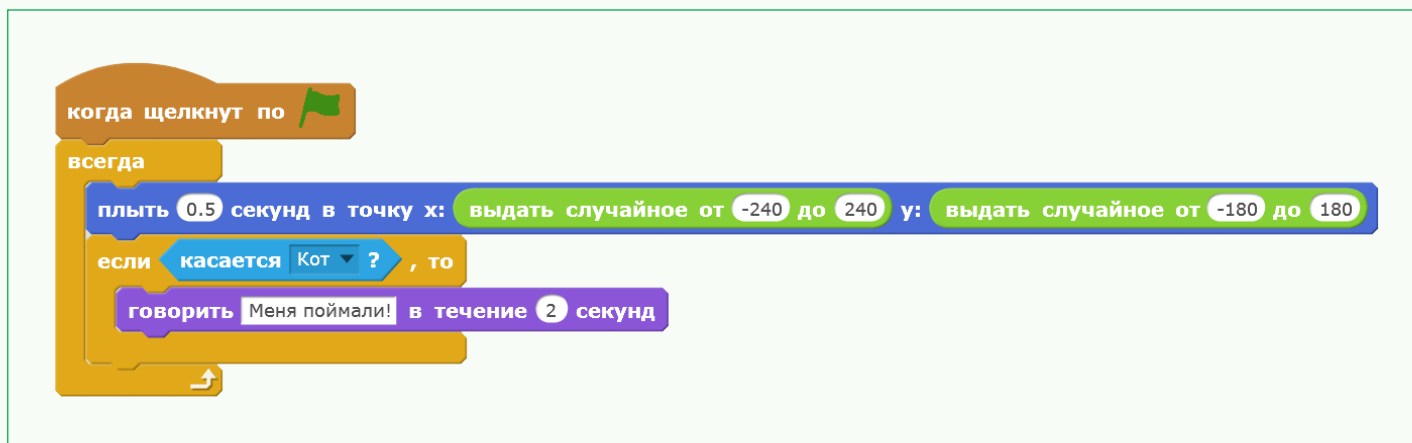
играть звук pop



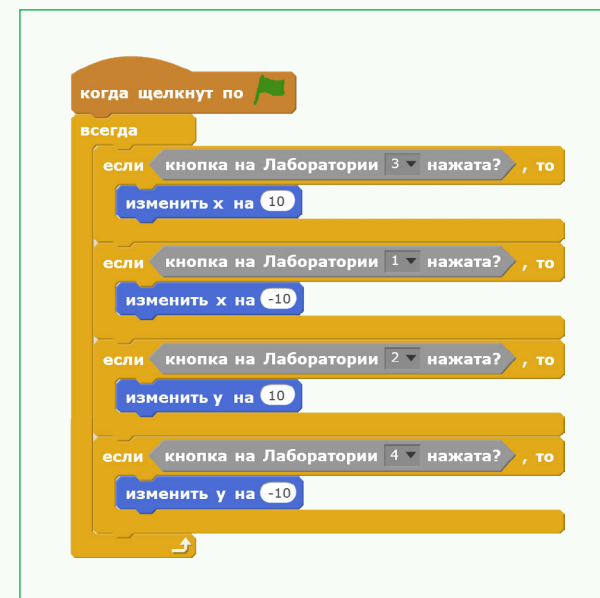


Упражнение 5: Поймите фигурку!

РЕШЕНИЕ 1



РЕШЕНИЕ 2



3. Создание игры

В последней части создаются различные новые элементы игры, которые увеличивают игровую составляющую. Пользователи закрепляют уже полученные знания в программировании и обучаются применению переменной величины. Также закрепляются навыки использования логического выражения.

Упражнения выполняются постепенно, пользователи шаг за шагом продвигаются к готовой игре. При выполнении упражнения 6 рекомендуется рассмотреть функцию сохранения сделанной программы (РИС. 4), а при выполнении упражнения 7 — открытия сохраненной программы (РИС. 5). Сохранение упражнений облегчает работу.

РИС. 4

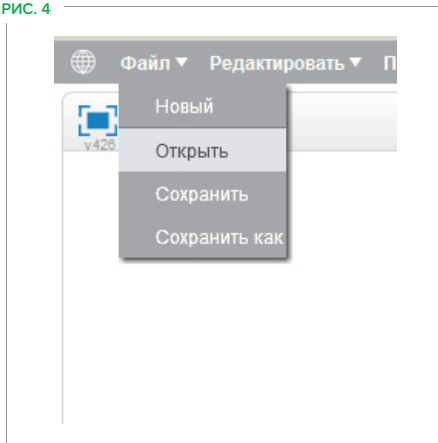
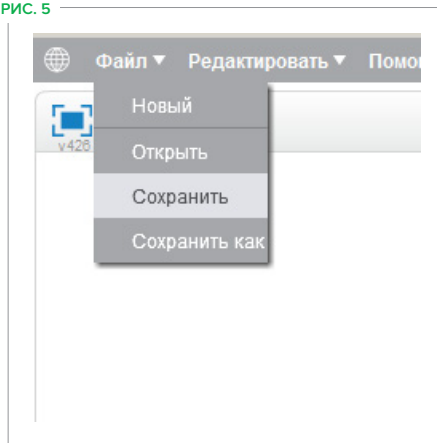


РИС. 5



УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 6: ПОДСЧЕТ ОЧКОВ

Использование переменной величины для подсчета очков (переменная **Счет**).

Важно:

- Переменная величина (или просто переменная) — это база данных, в которой сохраненная информация может храниться в виде цифр, текста или картинки. Подсчет очков — это переменная величина, которая должна быть легко понимаема (рис. 6).

УПРАЖНЕНИЕ 7: ЗАВЕРШЕНИЕ ИГРЫ

Закрепление навыков применения переменной и Лаборатории в программе.

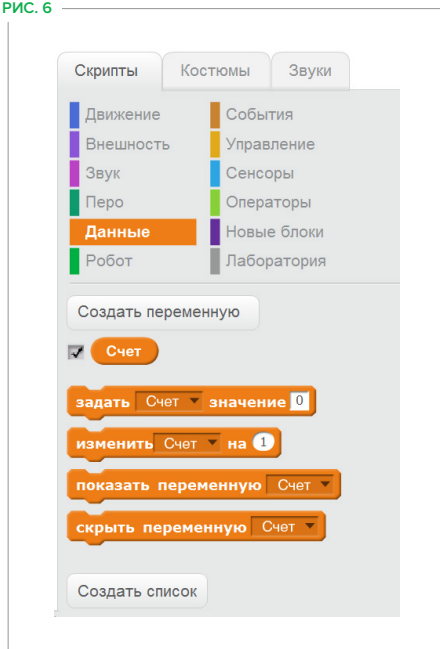
Важно:

- Посмотрите, как влияет на игру изменение значений команд **задать_значение_** и **изменить_на_**.
- В задании 2 не даются готовые к использованию команды, поскольку упражнения по применению данных команд уже выполнялись ранее.

УПРАЖНЕНИЕ 8: ОЗВУЧИВАНИЕ ИГРЫ

В данном упражнении пользователи учатся вносить в игру звук из Лаборатории, а также закрепляют уже полученные знания.

РИС. 6



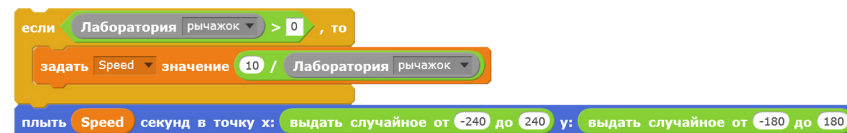
УПРАЖНЕНИЕ 9: УВЕЛИЧЕНИЕ СКОРОСТИ ПРИ ПОМОЩИ РЫЧАЖКА

Пользователи учатся использованию в программе рычажка Лаборатории и вспоминают работу с переменной, а также учатся применять математические операторы сравнения.

Важно:

- В данном упражнении запрограммировать рычажок сложнее, чем в предыдущем задании. Сначала рекомендуется обдумать стратегию решения задачи — например, в первую очередь выполнив уже знакомые части программы (рис. 7).
- Если значение рычажка Лаборатории выше, чем ноль (0 = рычажок находится в левом крайнем положении; 100 = рычажок находится в правом крайнем положении), то следует выполнить команду, находящуюся внутри команды **если <_>, то**.
- Для переменной **Speed** (скорость) устанавливается значение 10 / значение рычажка лаборатории (при нахождении рычажка в левом крайнем положении мышка двигается медленно, и, соответственно, в правом — быстро).
- Фигурка движется по сцене произвольно со скоростью, определяемой переменной **Speed**.
- Рекомендуется вместе подумать над использованием новых условий (операторов) сравнения и попробовать, как их изменение влияет на программу.
- Датчики освещенности и звука работают аналогично рычажку. Рекомендуется в составе группы творчески подойти к их использованию, выслушав различные идеи.

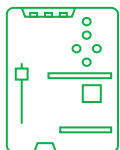
РИС. 7



НОВЫЕ КОМАНДЫ

	Команда задать_значение_ передает выбранной переменной новое значение.
	Команда изменить_на_ изменяет выбранную переменную на указанное значение.
	Переменная. Можно самому назначить имя переменной (на рисунке — Счет). Значение переменной можно в ходе программы менять и также использовать, например, в операторах сравнения.
	Команда стоп все/ этот скрипт/ другие скрипты останавливает необходимые программы, т.е. скрипты.
	Команда если <_>, то_ иначе (разветвленный условный оператор) исполняет находящиеся внутри него серии команд избирательно согласно значению истинности условия, заданного для команды. Если условие — истина , исполняется первое; если ложь , то последнее.
	Условия сравнения. Применяются в сравнении двух значений, например, в условных циклах.
	Математическое выражение. Можно использовать переменными и в условиях сравнения.
	Команда играть ноту на Лаборатории_ включает звучание из динамиков лаборатории звук определенной высоты.
	Команда Лаборатория свет/ звук/ рычажок передает значение датчика в запрашиваемый момент времени. Эти показания можно использовать, например, в условиях сравнения.





Упражнение 6: Подсчет очков

ЦЕЛИ

Создание переменной **Счет**, которая подсчитывает очки

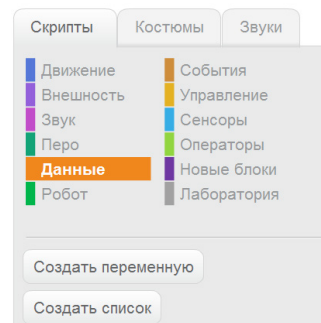
ЗАДАНИЕ

1. Сделайте другую фигурку (например, мышку).
2. Создайте переменную из группы команд **Информация** согласно серии изображений (**РИС. 1 и 2**).
3. Запрограммируйте мышку.
4. Щелкните по зеленому флажку.
5. Если выполните все правильно, то значение переменной **Счет** будет расти, когда кошка будет ловить мышку.

ЭКСПЕРИМЕНТ!

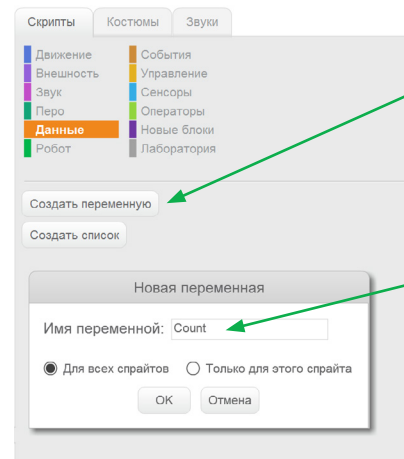
Как изменится счет, если изменить значение переменной?

РИС. 1



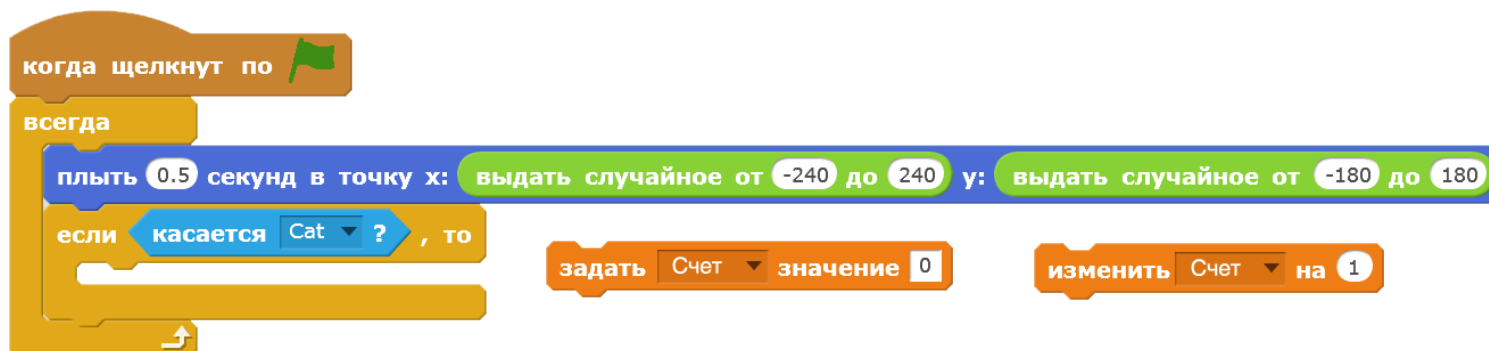
Выберите
информацию

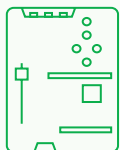
РИС. 2



Выберите
Создайте
переменную.

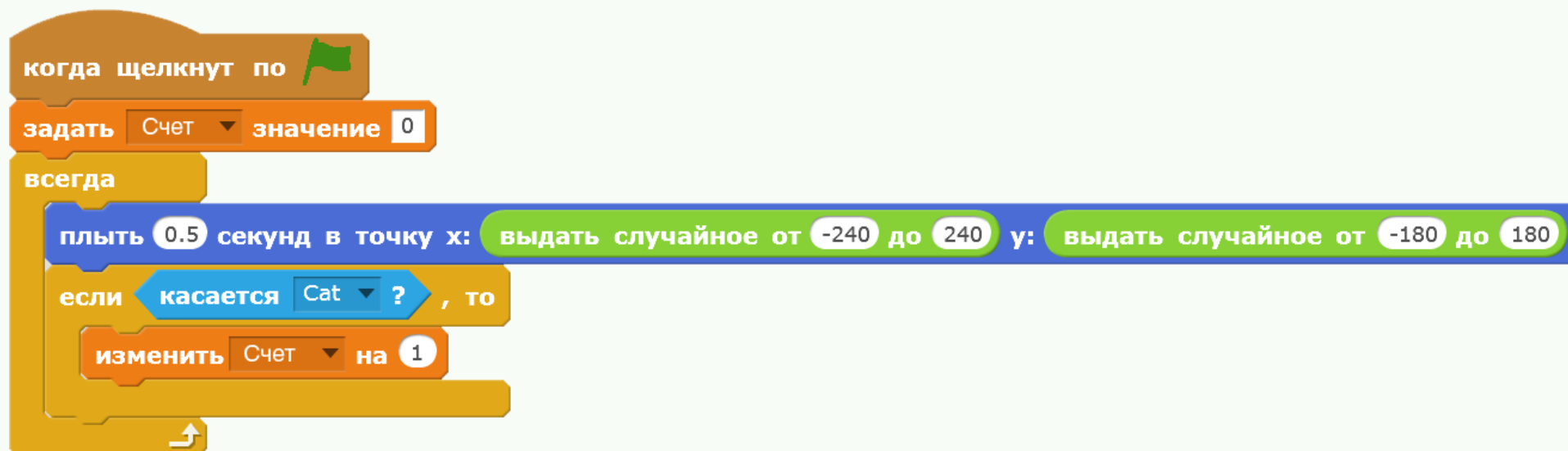
Дайте переменной
имя **Count**
и щелкните по **OK**.

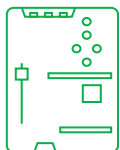




Упражнение 6: Подсчет очков

РЕШЕНИЕ





Упражнение 7: Завершение игры

ЦЕЛИ

- Использование переменной **Счет**
- Закрепление полученных знаний

7. Если выполнили все правильно, то значение переменной **Счет** будет расти при попадании кошки в мышку. Игра заканчивается, когда набрано 10 очков.

ЗАДАНИЕ 1

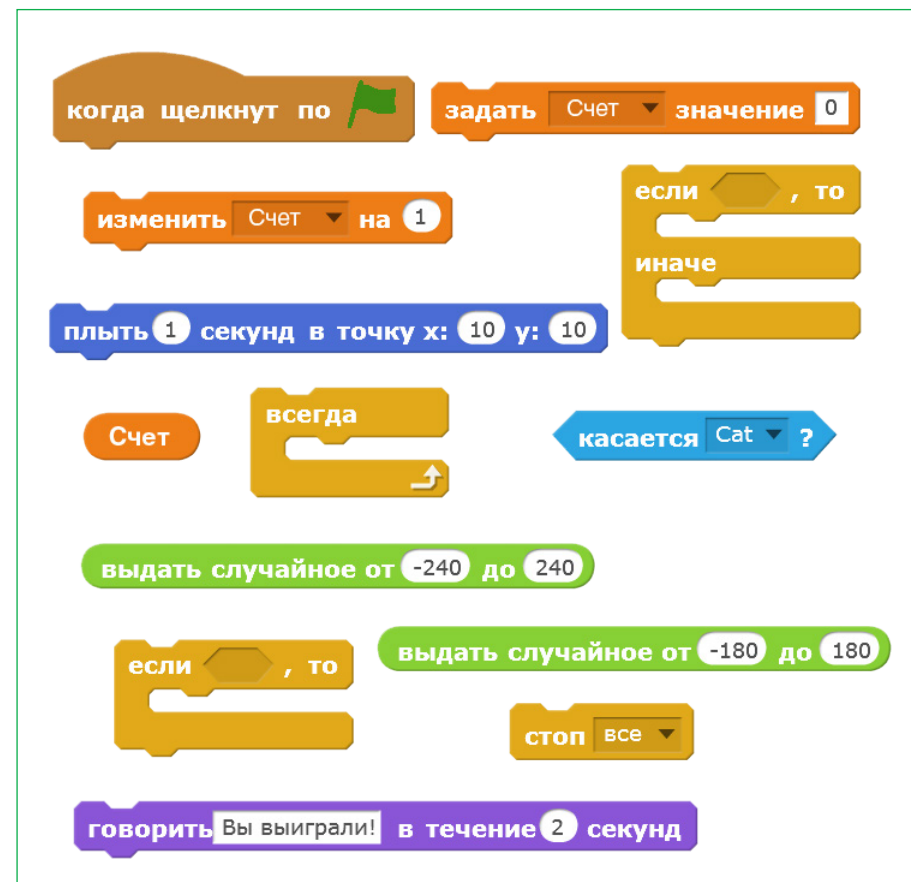
1. Сделайте две фигурки (например, кошка и мышка).
2. Создайте переменную **Счет** (см. упражнение 6).
3. Запрограммируйте мышку (см. упражнение 5).
4. Запрограммируйте Лабораторию для работы в качестве управляющего элемента кошки (см. упражнение 3).
5. Щелкните по зеленому флажку.
6. Управляйте кошкой при помощи Лаборатории и постарайтесь как можно скорее поймать мышку.

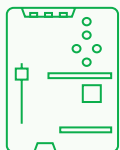
ЗАДАНИЕ 2

Запрограммируйте игру так, чтобы она начиналась при нажатии на белую кнопку.

ЭКСПЕРИМЕНТ!

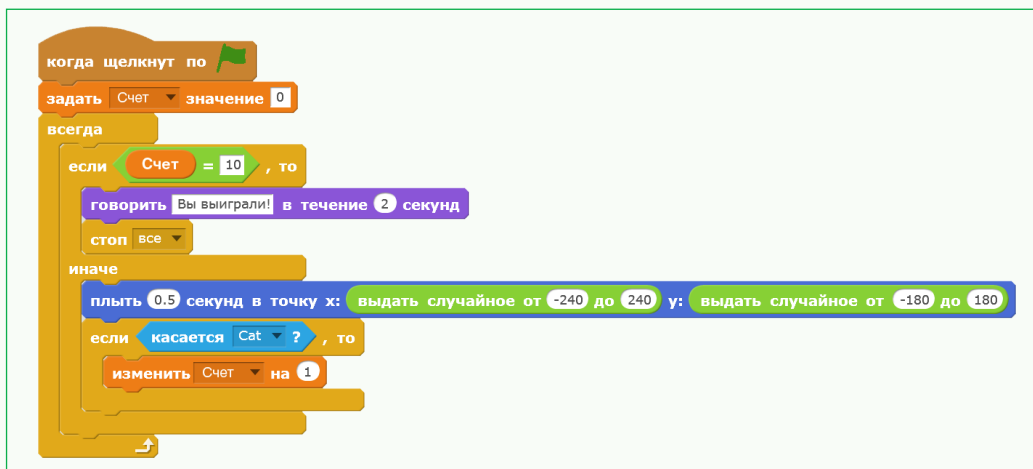
Как меняется игра, когда вы меняете значения в командах **задать_значение_** и **изменить_на_**?



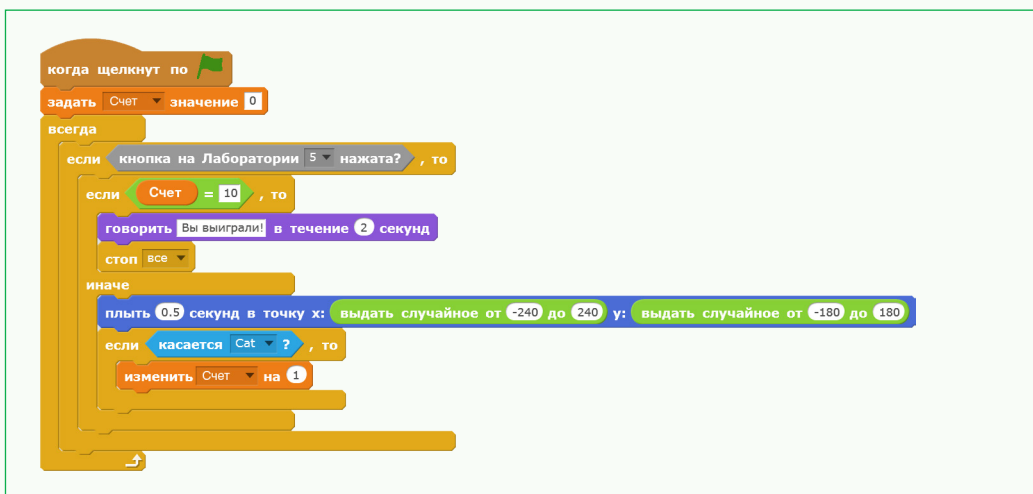


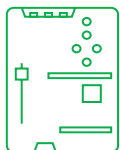
Упражнение 7: Завершение игры

РЕШЕНИЕ 1



РЕШЕНИЕ 2





Упражнение 8: Озвучивание игры

ЦЕЛИ

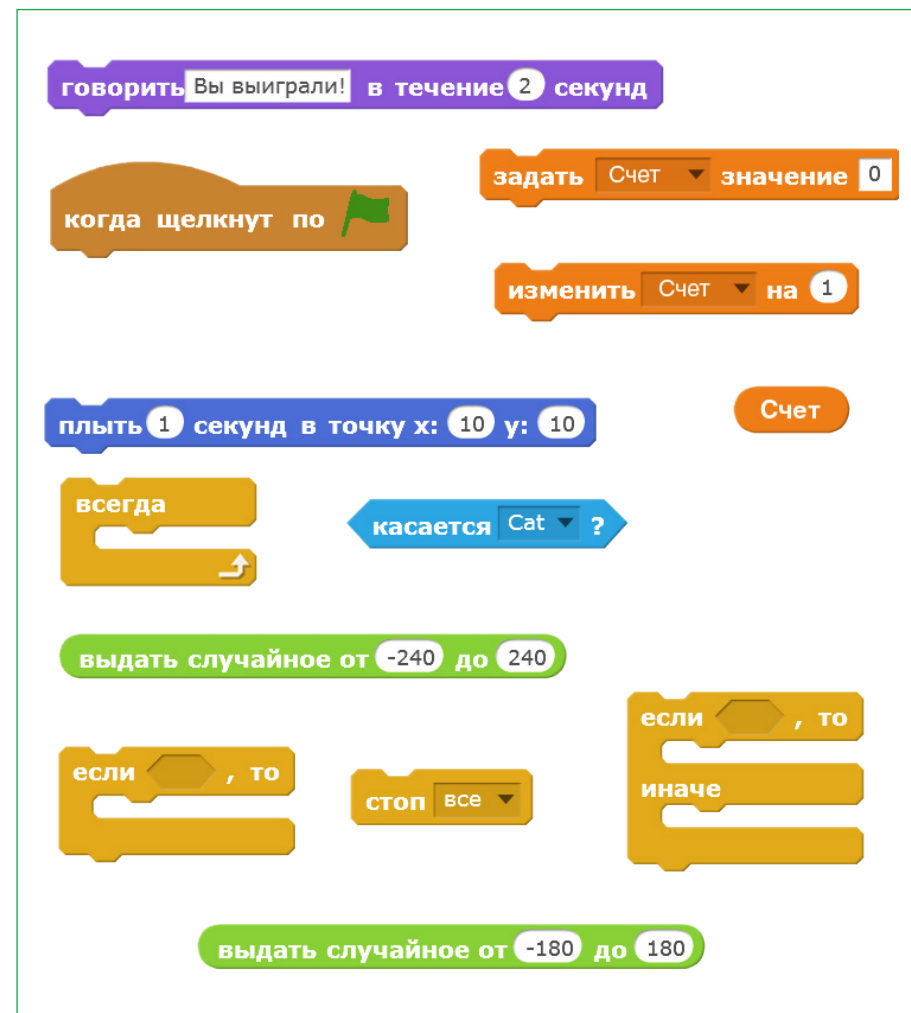
- Создание звука в Лаборатории
- Закрепление пройденного

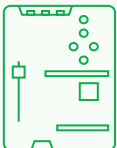
ЗАДАНИЕ

1. Сделайте две фигурки (например, кошку и мышку).
2. Создайте переменную **Счет** (см. упражнение 6).
3. Запрограммируйте мышку (см. упражнение 5).
4. Запрограммируйте Лабораторию для работы в качестве управляющего элемента кошки (см. упражнение 3).
5. Щелкните по зеленому флажку.
6. Управляйте кошкой при помощи Лаборатории и постарайтесь как можно скорее поймать мышку.
7. Если выполнили все правильно, Лаборатория воспроизведет звук при прикосновении кошки к мышке.

ЭКСПЕРИМЕНТ!

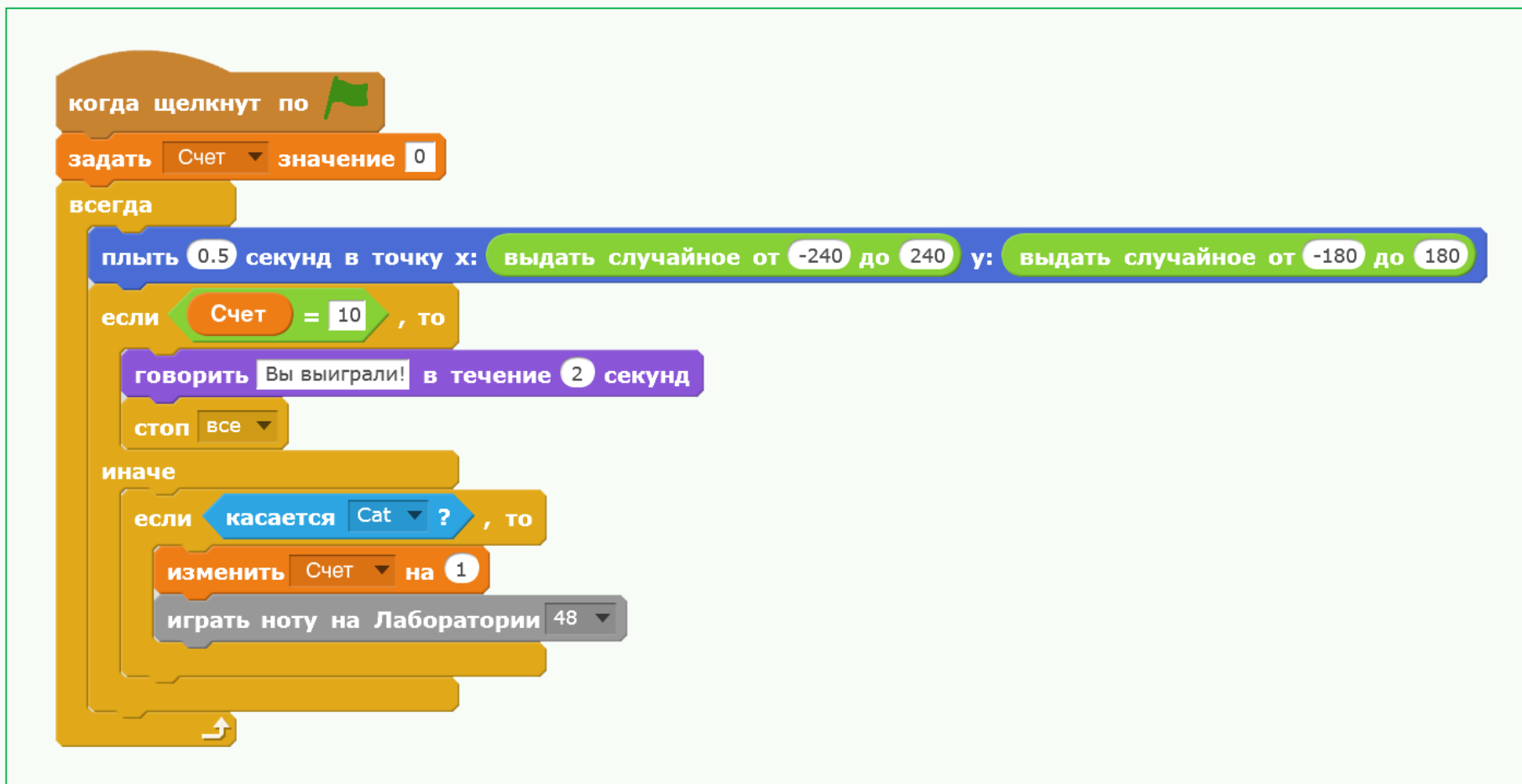
Как меняется звук, когда вы меняете значение команды **играть ноту_**?

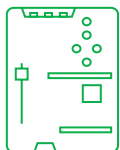




Упражнение 8: Озвучивание игры

РЕШЕНИЕ





Упражнение 9: Увеличение скорости при помощи рычажка

ЦЕЛИ

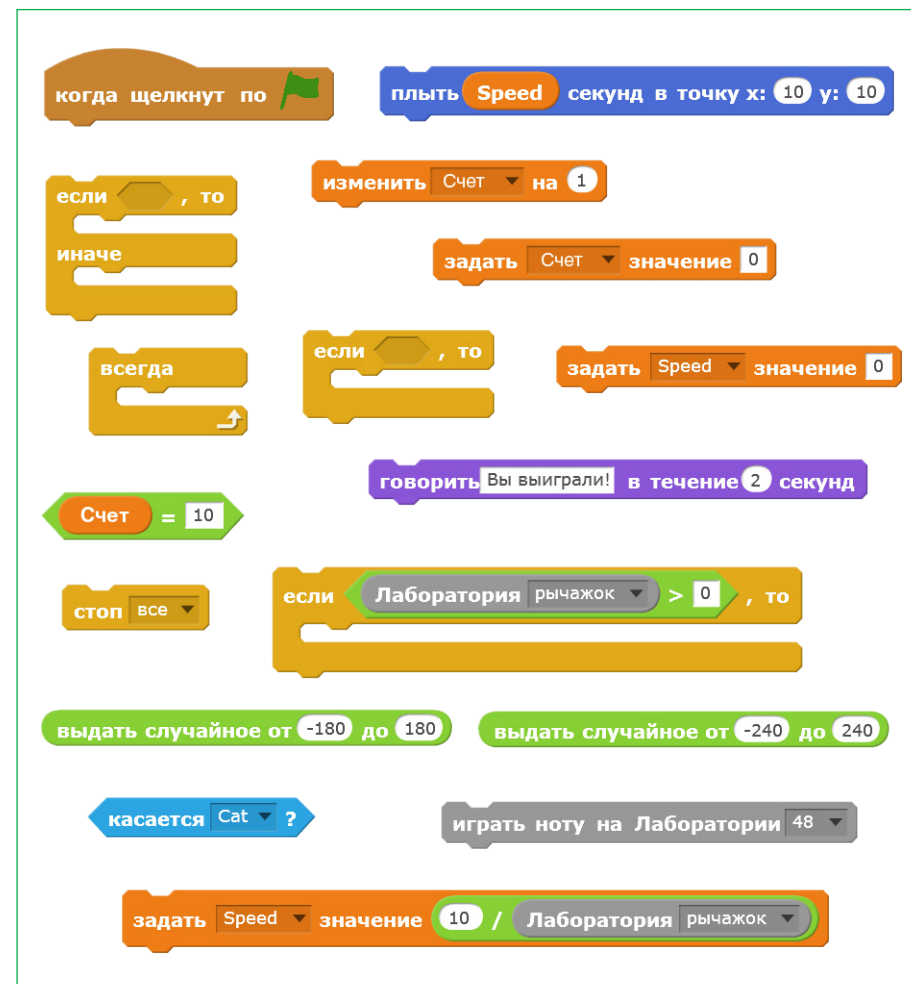
- Использование в программе рычажка
- Использование новых сравнительных условий
- Закрепление полученных навыков

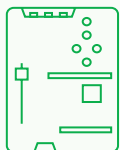
ЗАДАНИЕ

1. Сделайте две фигурки (например, кошку и мышку).
2. Создайте переменные **Счет** и **Speed** (см. упражнение 6).
3. Запрограммируйте мышку (см. упражнение 5).
4. Запрограммируйте Лабораторию для работы в качестве управляющего элемента кошки (см. упражнение 3).
5. Щелкните по зеленому флажку.
6. Управляйте кошкой при помощи Лаборатории и постарайтесь как можно скорее поймать мышку. Скорость мышки можете регулировать рычажком.

ЭКСПЕРИМЕНТ!

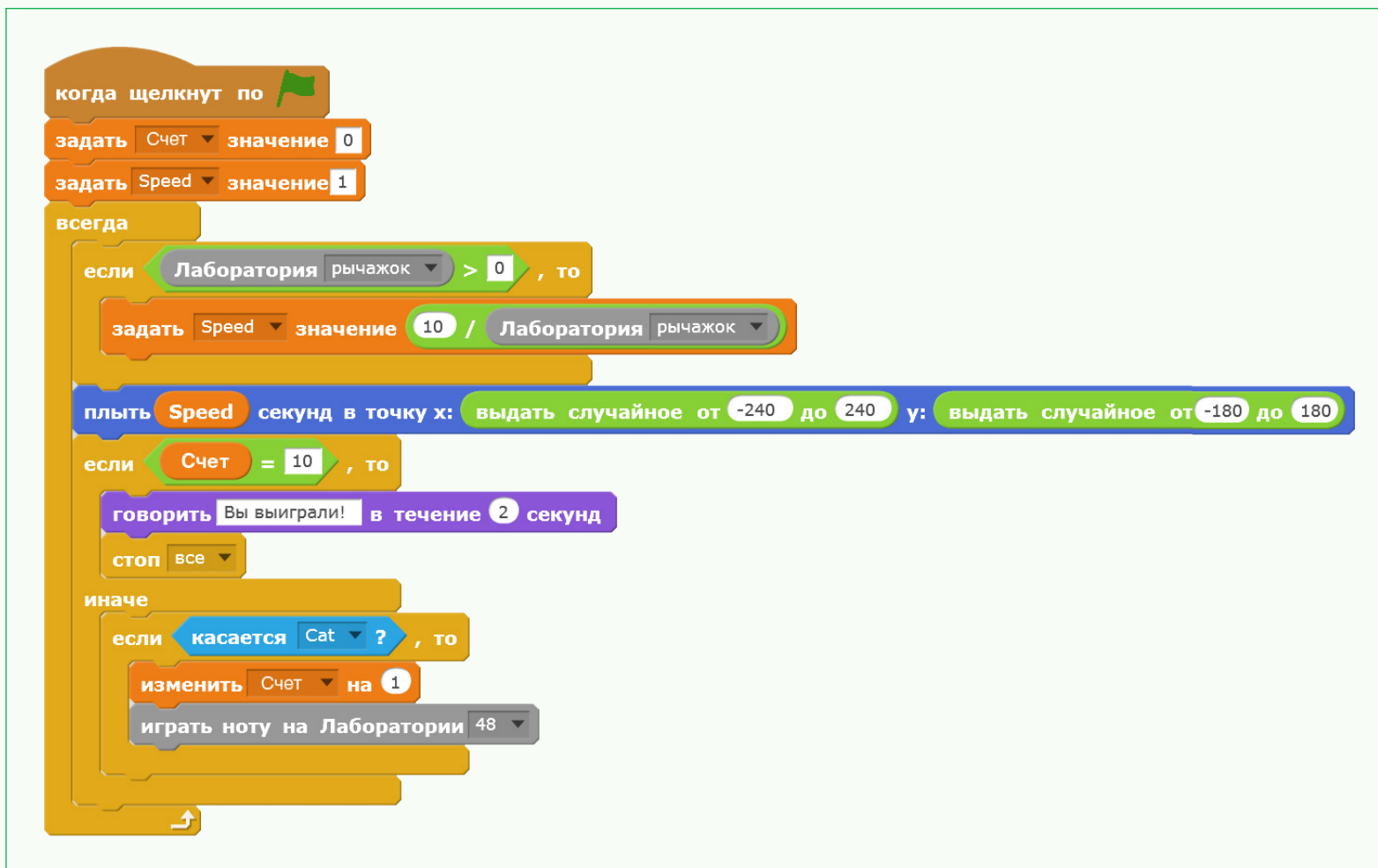
Как изменится игра, если изменить значение переменной **Speed**?

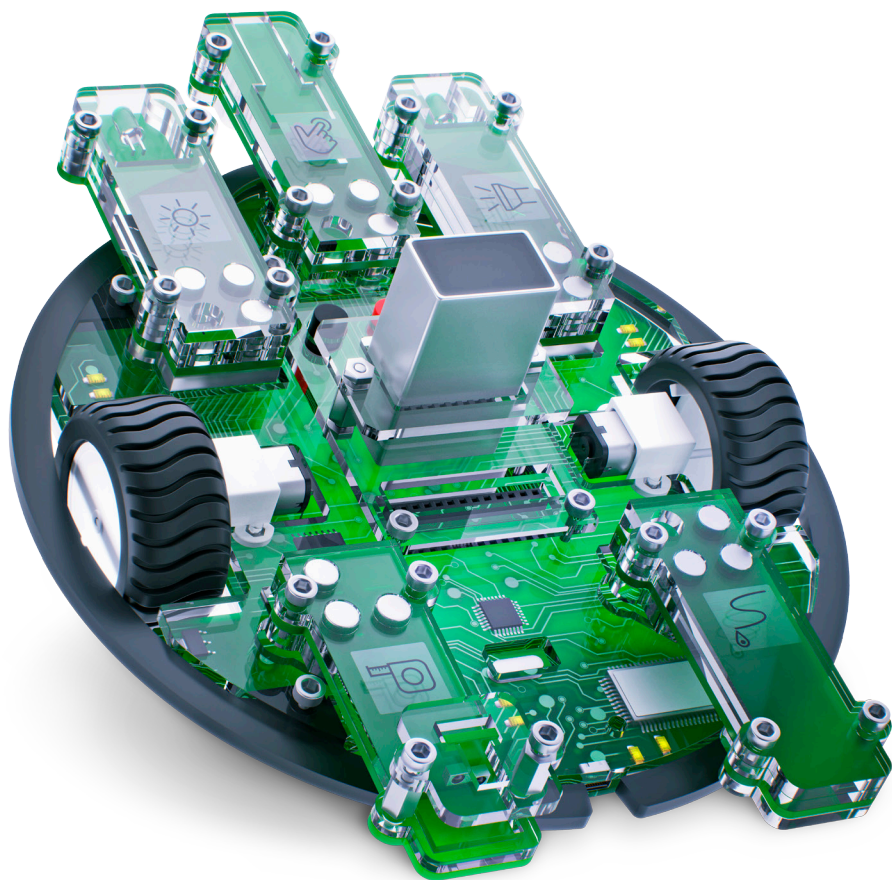




Упражнение 9: Увеличение скорости при помощи рычажка

РЕШЕНИЕ





РОББО РОБОПЛАТФОРМА

1. Программирование Робоплатформы

Упражнения разделены на темы по уровню подготовки обучающихся. В качестве введения к упражнениям для преподавателя в сжатом виде приводятся цели, содержание и используемые команды. Также поднимаются

вопросы того или иного упражнения, наиболее важные с педагогической точки зрения; предлагаются темы для бесед, которые можно обсуждать во время упражнений или после их выполнения.

РИС. 1

[illegible]

Упражнения составлены в форме карточек заданий. Их можно делать со всей группой вместе, в небольших группах со свободным перемещением из группы в группу или на индивидуальных занятиях, проводимых в невысоком темпе. Упражнения можно распечатать и раздать участникам группы, что позволит вести занятие дифференцированно, в соответствии с уровнем подготовки.

Ход занятий можно контролировать с помощью таблицы, в которой дети будут отмечать выполнение заданий (рис. 1).

В конце каждой темы собраны упражнения, на которые рекомендуется затратить чуть больше времени. После выполнения задания полученный навык станет автоматическим.

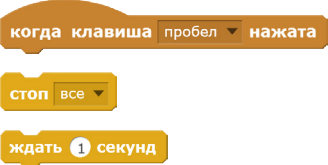
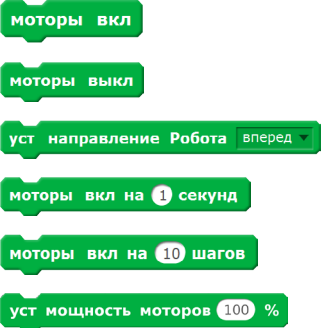
В каждой теме поднимаются явления, связанные с различными бытовыми вопросами. Полученный материал закрепляется лучше, когда он связан с повседневной жизнью. Одновременно ученики получают навыки наблюдения за окружающим миром и представленными в нем компонентами программирования. Темы для более углубленного изучения окружающей среды можно найти в разделе руководства «Робототехника в жизни». Также навыки наблюдения за окружающей средой можно получить с помощью различных заданий, например, путем фотографирования явлений своей повседневной жизни. Кроме того, полезно поразмышлять, для чего человеку нужно то или иное явление и как бы он ощущал себя без него.

Модели решений заданий предназначены для поддержки работы как преподавателя, так и учеников. Модели приводятся на оборотной стороне карточек заданий. Одно и то же упражнение может иметь несколько моделей решения, из которых только одна представлена в карточке. Рекомендуется обсудить с детьми различные модели решений, поскольку обнаружение и понимание различных стратегий решений будет способствовать развитию умственных способностей у детей. Также детей можно призывать размышлять о том, можно ли программу реализовать более коротким путем с меньшим количеством команд или как легко программу можно видоизменить и применить к схожим ситуациям.

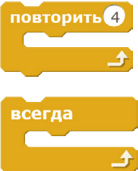
Упражнения следует выполнять постепенно, в соответствии с прилагаемой таблицей «Ход программирования с Робоплатформой». Упражнения разделены на группы по рассматриваемым темам. Они способствуют постепенной выработке навыков работы с аналогичным материалом. Упражнения начального уровня рекомендуется повторять на более поздних этапах обучения, чтобы знания, полученные в начале обучения, не забывались в дальнейшем.

ХОД ПРОГРАММИРОВАНИЯ С РОБОПЛАТФОРМОЙ

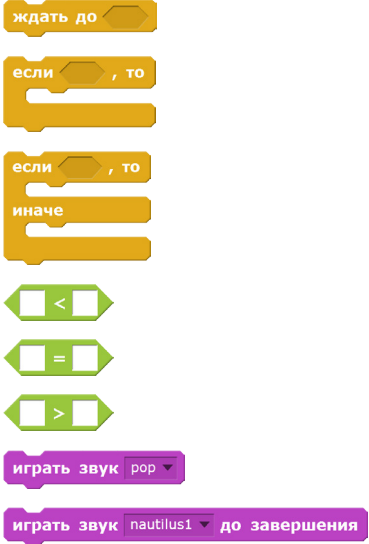
УПРАВЛЯЕМ РОБОПЛАТФОРМОЙ С ПОМОЩЬЮ КНОПОК

Упражнение	Цели	Команды и структуры	Scratch	RobboScratch
1. Иди — стой 2. Управление стрелками 3. Закрепление знаний	- Знакомство с программной средой - Освоение структуры программы и серии команд - Приведение Робоплатформы в движение и ее остановка - Использование последовательных команд (линейная серия команд)	- Начало и окончание серии команд - Управление двигателями: - Включение и остановка - Регулировка мощности - Изменение направления - Поворот - Использование времени и шагов - Задержки		

САМОСТОЯТЕЛЬНОЕ ДВИЖЕНИЕ РОБОПЛАТФОРМЫ

Упражнение	Цели	Команды и структуры	Scratch	RobboScratch
4. Обход препятствия 5. Обход коробки 6. Движение вдоль фигур 7. Закрепление знаний	- Закрепление ранее полученных навыков - Разработка более длинных серий команд - Определение одинаково повторяющихся серий команд и изучение циклов - Изучение, как ограничивать количество повторений и использовать циклы - Планирование серий команд	- Циклы - Повторить_раз - Всегда		

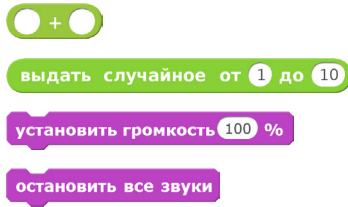
ХОД ПРОГРАММИРОВАНИЯ С РОБОПЛАТФОРМОЙ

ДАТЧИК КАСАНИЯ				
Упражнение	Цели	Команды и структуры	Scratch	RobboScratch
8. Включаем Робоплатформу датчиком касания 9. Столкновения с препятствием 10. Закрепление знаний	- Закрепление навыка употребления циклов - Ознакомление с принципами работы датчиков - Более глубокое знакомство с датчиком касания и изучение его использования - Изучение использования условных операторов в программах - Внесение звуков в программы	- Условные операторы - Если — то - Если — то — иначе - Ждать до - Операции сравнения - Меньше чем - Такой же как - Больше чем - Звуки - Сыграйте звук - Сыграйте звук до конца		

ДАТЧИК СЛЕЖЕНИЯ ЗА ЛИНИЕЙ				
Упражнение	Цели	Команды и структуры	Scratch	RobboScratch
11. Следуем за линией 12. Пути и маршруты 13. Закрепление знаний	- Закрепление употребления структур повторения и условий - Более подробное знакомство с датчиком слежения за линией и изучение его использования - Закрепление полученных ранее навыков			

ХОД ПРОГРАММИРОВАНИЯ С РОБОПЛАТФОРМОЙ

ДАТЧИК ОСВЕЩЕННОСТИ

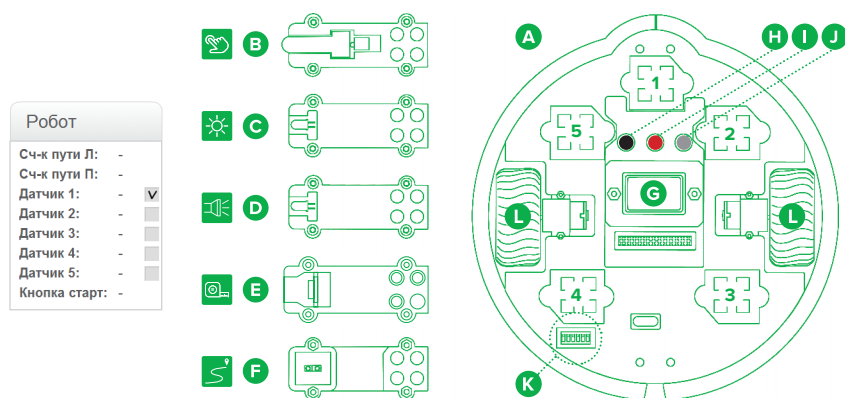
Упражнение	Цели	Команды и структуры	Scratch	RobboScratch
14. Коробка с сюрпризом 15. Робоплатформа поворачивается к свету 16. Сторожевая собака 17. Закрепление знаний	- Закрепление употребления циклов и условий - Более подробное знакомство с датчиком освещенности и изучение его использования - Самостоятельное озвучивание программы - Измерение показаний датчиков и использование их в программах - Использование случайного числа и операции суммирования	- Суммирование - Случайные числа - Звуки: - Установка громкости звука - Остановка звуковой команды		

ДАТЧИК РАССТОЯНИЯ

Упражнение	Цели	Команды и структуры	Scratch	RobboScratch
18. Робоплатформа путешествует 19. Применение знаний	- Закрепление употребления циклов и условий - Более подробное знакомство с датчиком расстояния и изучение его использования - Закрепление полученных ранее навыков			

ДАТЧИКИ

РИС. 2



A. Робоплатформа.

B. Датчик касания опознает прикосновение.

C. Датчик освещенности опознает свет.

D. Светодиодный датчик дает свет.

E. Датчик расстояния опознает расстояние до объекта.

F. Датчик линии опознает светлую и темную поверхности.

G. Батарея.

H. Кнопка Reset перезагружает процессор.

I. Кнопка ON/OFF (вкл./выкл.) включает питание от батареи.

J. Кнопка Start — программируемая кнопка.

K. Переключатели DIP:

В Робоплатформе имеются переключатели, с помощью которых регулируется ее работа.

1. & 2. Кольцевой переключатель.

3. Защита от недостаточного напряжения батареи (используйте вместе с заряжаемой батареей).

4. Блокировка совместного использования провода USB.

5. Автоматическое время отключения (около 10 минут).

Внимание! В большинстве случаев все переключатели рекомендуется держать в положении ON (Вкл.).

L. Колеса.

Датчики — это приборы, которые воспринимают информацию из окружающей среды и оценивают ее. Их много в нашем окружении:

- В раздвижных дверях датчик движения включает механизм открывания дверей (датчик освещенности или датчик расстояния).
- Сумеречный выключатель в стене дома включает светильники во дворе (датчик освещенности).
- В магазине устройство для считывания штрих-кодов распознает темные линии различной толщины на светлой поверхности (датчик линии).

- В современных автомобилях имеется радар заднего хода (датчик расстояния) и датчик пассажира, сидящего на сиденье (датчик касания).

К Робоплатформе можно подсоединить четыре разных датчика и светодиодный модуль LED (РИС. 2):

- Датчик касания опознает прикосновение.
- Датчик освещенности опознает окружающий свет.
- Датчик расстояния опознает расстояние до объекта.
- Датчик линии опознает светлую и темную поверхности.
- Светодиодный модуль дает свет.

Датчики крепятся к Робоплатформе при помощи магнитов. Когда датчик установлен на платформу, квадратик «Датчик места» отмечен в программе галочкой (РИС. 2). Значения, считываемые с датчика, становятся видны на дисплее и обновляются в режиме реального

времени. Все датчики имеют свои минимальные и максимальные значения, которые следует всегда проверять перед началом программирования. Перед выполнением упражнений принципы работы датчиков необходимо разъяснять ученикам.

УПРАЖНЕНИЕ: УПРАВЛЯЕМ ЧЕЛОВЕКОМ-РОБОТОМ

Это первое упражнение в программировании выполняется без Робоплатформы. Цель упражнения — давать простые инструкции и настроить детей на программное мышление.

Цели

- Разделение инструкции на мелкие части
- Выдача простых команд
- Понимание действия робота: программа робота разработана человеком

Ход занятия

Преподаватель просит двоих учеников выйти перед классом — это «робот» и «программист».

- Задача «робота» — действовать по указаниям «программиста». «Программисту» не разрешается общаться, делать подсказки или действовать произвольно.
- «Программист» дает «роботу» однозначные короткие инструкции к действию.
- Преподаватель сообщает об ошибках: если указание «программиста» приводит к разным толкованиям, то преподаватель дает знак.

- Остальные участники группы являются коррективщиками «программы» и дают новую инструкцию после получения сообщения об ошибке.
- «Программисту» дается задание, которое он велит сделать «роботу». Например:
 - Надеть шапочку, перчатки и шарф.
 - Написать имя на доске.
 - Поднять предмет и положить на стол.

Приложения

Упражнение можно усложнить путем использования более сложных команд.

- Например, вместо команды «Сделай два шага» дается указание, состоящее из мелких частей: «подними правое бедро, выпрями правую голень, потяни пальцы левой ноги, опусти левую ступню на пол».
- Данное упражнение имеет целью продемонстрировать, что «простые» команды получаются от того, что кто-то их сложно закодировал.
- Эту идею можно продумать с детьми, например, при работе с командой «включи двигатель».

ТЕМЫ ДЛЯ БЕСЕД ВО ВРЕМЯ ИЛИ ПОСЛЕ УПРАЖНЕНИЙ

- Какие инструкции были хорошими и почему?
- Из какой серии мелких указаний состоит команда?

2. Управляем Робоплатформой с помощью кнопок

Упражнения по программированию Робоплатформы начинаются с небольших заданий, целью которых является ознакомление пользователей с работой программного обеспечения. Вначале пользователи учатся использовать с Робоплатформой программную среду RobboScratch, а также упражняются с принципами построения программ. Готовые

и собственные программы строятся с помощью коротких последовательных команд, из которых формируется линейная серия команд.

В ходе работы изучают, какие мыслительные умения необходимы для программирования: прогнозирование работы готовой программы и изменение ее под свои потребности.

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 1: ИДИ — СТОЙ

Происходит знакомство с разработкой программ, и по образцу выполняют разработку первой программы. Пользователи двигают Робоплатформой и учатся останавливать ее.

Важно:

- Изучить, как двигается Робоплатформа при увеличении или уменьшении мощности привода.
- Совместно поразмышляйте над различными и предназначением команд **моторы вкл**, **моторы вкл на_секунд** и **моторы вкл на_шагов**.

УПРАЖНЕНИЕ 2: УПРАВЛЕНИЕ СТРЕЛКАМИ

Первую программу выполняют сначала по образцу и затем для усвоения пройденного добавляют в программу одинаковые серии команд.

Важно:

- Научить пользователя прогнозировать, какие события произойдут в результате работы программы. И только после этого приступать к созданию и испытанию программы.
- Исследовать и поразмышлять вместе, как разные команды для двигателя влияют на работу Робоплатформы.

УПРАЖНЕНИЕ 3: ЗАКРЕПЛЕНИЕ ЗНАНИЙ

С помощью выполнения различных дополнительных заданий происходит закрепление изученных знаний. Эти задания рекомендуется выполнять отдельно от упражнений 1 и 2. Для усвоения знаний рекомендуется возвращаться к ним повторно на дальнейших этапах обучения.

Важно:

- Напоминать ученикам о необходимости использовать команду **мощность мотора** для установки скорости до включения моторов.
- Во время выполнения упражнений «Управление вслепую» порассуждайте, чем отличается управление человеком от управления роботом. Какие команды лучше других подходят для «программирования» человека?



КОМАНДЫ

	Команда когда щелкнут по приводит в действие серию команд, зафиксированных ниже, если щелкнуть мышью по флажку.
	Команда когда клавиша_нажата включает программу с помощью выбранной из меню кнопки клавиатуры.
	Команда ждать_секунд останавливает выполнение серии команд на заданное время (на рисунке — на 1 секунду).
	Команда стоп_ (все/ этот скрипт/другие скрипты фигурки) останавливает избранные скрипты, т.е. программы.
	Команда моторы вкл включает оба мотора.
	Команда моторы выкл выключает оба мотора.
	Команда уст мощность моторов на_% устанавливает мощность обоих моторов в диапазоне от 0 до 100%.
	Команда моторы вкл на_секунд включает оба мотора на заданное время (на рисунке — на 1 секунду).
	Команда моторы вкл на_шагов включает двигатель на прохождение определенного количества «шагов» — риск на колесе.
	Команда уст направление робота_ задает направление вращения двигателей.

ТЕМЫ БЕСЕД ВО ВРЕМЯ И ПОСЛЕ ВЫПОЛНЕНИЯ УПРАЖНЕНИЙ

Какие различия есть у разных команд для двигателя? Как они влияют на движение Робоплатформы?

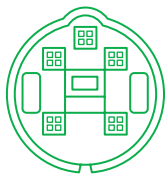
Какие изученные команды используются в повседневной жизни?

- Команды «Включить/выключить» часто используются в быту. Примером является переключатель, которым открывают и закрывают, включают или выключают приборы. Важно также отметить разницу обычного механического переключателя от запрограммированного.

- Механические переключатели:
 - Выключатель света
 - Кнопка включения прибора
 - Различные игрушки
- Запрограммированные переключатели:
 - Пульт дистанционного управления
 - Телевизор
 - Сумеречный переключатель
- Стрелочное управление часто используется в дистанционно управляемых игрушках и в игровых программах. Многие рабочие установки и механизмы (подъемники, погрузчики) имеют приборы со стрелочным управлением.

РОБОТОТЕХНИКА В ЖИЗНИ

- Выясните, как работает мотор, какие существуют виды двигателей.
- Выясните, за счет чего Робоплатформа и программа знают, на сколько шагов прокрутились колеса Робоплатформы.
- Выясните, как работает используемый Робоплатформой регулятор мощности с широтно-импульсной модуляцией (ШИМ) (подсказка: в основании Робоплатформы имеется т.н. энкодер).



Упражнение 1: Иди — Стой

ЦЕЛИ

- Ознакомление с программной средой
- Создание первой программы по образцу
- Изучение структуры программы
- Движение Робоплатформы вперед и ее остановка

ЗАДАНИЕ 1

- В меню команд (1) имеются различные группы команд.
- Команды из «Группы команд» видны в этой зоне (2).
- В зону программирования надо перетащить нужные команды (3).
- Попробуйте, что получится в результате программ, показанных на рисунке.

ЗАДАНИЕ 2

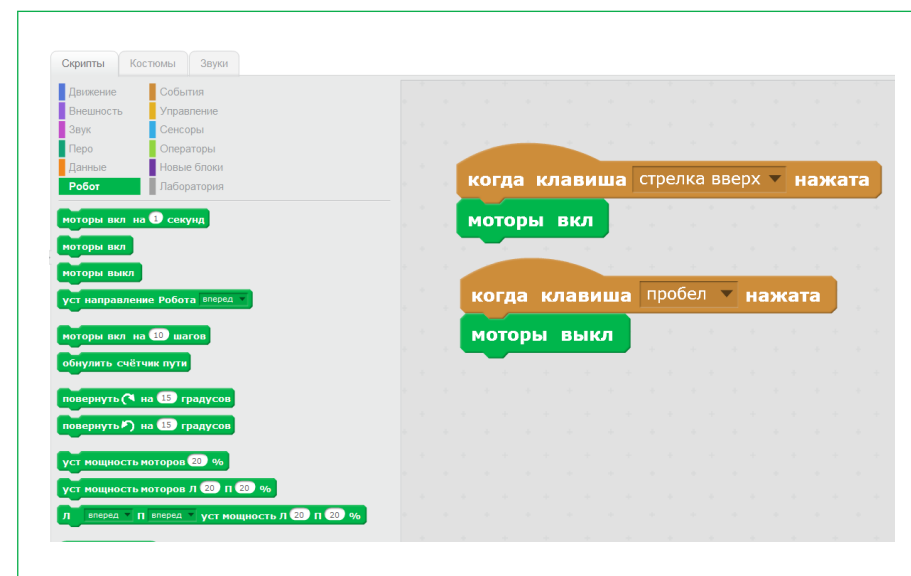
Попробуйте изменить мощность мотора.

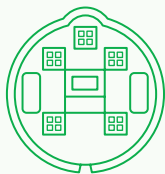
уст мощность моторов 100 %

ЗАДАНИЕ 3

Выясните, как запустить и остановить Робоплатформу.

ждать 1 секунд





Упражнение 1: Иди — Стой

РЕШЕНИЕ 1

При нажатии кнопки «стрелка вверх» мотор Робоплатформы заводится и Робоплатформа движется вперед. При нажатии пробела программа и Робоплатформа останавливаются.

ЭКСПЕРИМЕНТ!

Посмотрите, какая разница между двумя этими командами.

моторы вкл на 1 секунд

моторы вкл на 10 шагов

Попробуйте добавить эту команду в программу остановки Робоплатформы. Что происходит?

СТОП все

РЕШЕНИЕ 2

Приведение в движение на линии

когда щелкнут по



ждать 1 секунд

моторы вкл

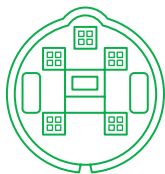
Остановка на линии

когда клавиша пробел нажата

ждать 1 секунд

моторы выкл





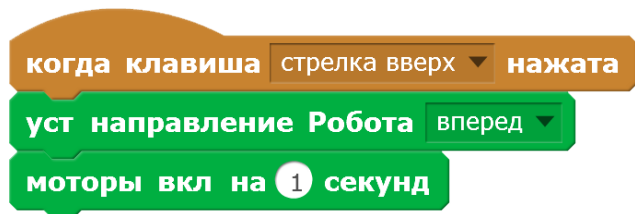
Упражнение 2: Управление стрелками

ЦЕЛИ

- Использование последовательных команд
- Управление Робоплатформой для движения вперед, назад, влево, вправо

ЗАДАНИЕ 1

- Спрогнозируйте, как эта программа будет двигать Робоплатформу.
- Сделайте программу и испытайте ее.



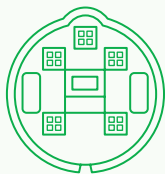
ЗАДАНИЕ 2

Попробуйте подвигать Робоплатформу с помощью кнопок со стрелками вперед и назад, а также поворачивать ее вправо и влево.

ПОДСКАЗКА

В одной и той же зоне программирования могут быть несколько цепочек команд.





Упражнение 2: Управление стрелками

РЕШЕНИЕ 1

При нажатии на клавиатуре кнопки ↑ Робоплатформа двигается вперед в течение 1 секунды.

ЭКСПЕРИМЕНТ!

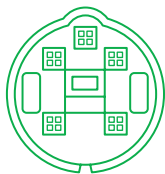
Замените команду **моторы вкл на 1 секунду** указанной ниже командой. Что произойдет?

моторы вкл

РЕШЕНИЕ 2

	когда клавиша стрелка вверх нажата уст направление Робота вперед моторы вкл на 1 секунд
	когда клавиша стрелка влево нажата уст направление Робота налево моторы вкл на 1 секунд
	когда клавиша стрелка направо нажата уст направление Робота направо моторы вкл на 1 секунд
	когда клавиша стрелка вниз нажата уст направление Робота назад моторы вкл на 1 секунд





Упражнение 3: Закрепление знаний

ЦЕЛИ

- Закрепление полученных навыков.
- Подача четких указаний к действию
- Контроль данных указаний к действию

ЗАДАНИЕ 1: МАРШРУТЫ ИЗ МАЛЯРНОЙ ЛЕНТЫ

Создайте на полу с помощью малярной ленты различные маршруты и проведите по ним Робоплатформу.

ЗАДАНИЕ 2: СЛАЛОМНАЯ ТРАССА

- Создайте на полу слаломную трассу. Насколько быстро вы пройдете ее?

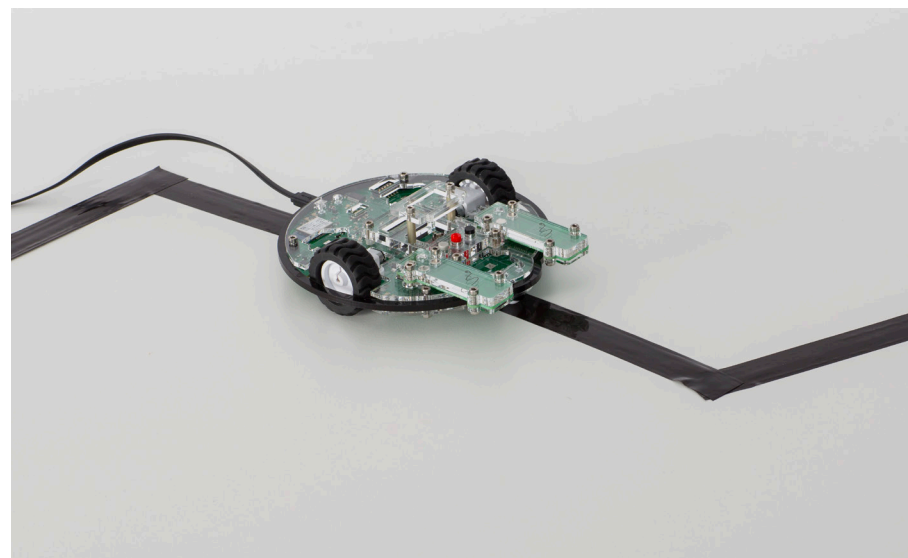
- См. видеоролик: <https://vimeo.com/141896031>.
- С помощью каких команд Робоплатформа движется быстрее всего?

ЗАДАНИЕ 3: УПРАВЛЯТЬ ВСЛЕПУЮ

- Работая в паре, постройте трассу.
- Запрограммируйте Робоплатформу на движение при помощи кнопок-стрелок.
- Разделите задания:
 - Один ученик — водитель, и ему завязывают глаза. Его задача — управлять Робоплатформой вслепую по указаниям от другого ученика.
 - Другой ученик дает указания водителю.
- В конце задания поговорите, какие указания были хорошими и почему.

ЗАДАНИЕ 4: ИЗОБРЕТАЙТЕ!

Придумайте задание, в котором можно использовать полученные навыки.



3. Самостоятельное движение Робоплатформы

После ознакомления с программами, формируемися из коротких основных команд, переходим к составлению более длинных программ.

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 4: ОБОИД ПРЕПЯТСТВИЯ

Закрепляем полученные ранее знания и составляем более длинные программы, а также планируем новые.

Важно:

- При выполнении данного упражнения рекомендуется сравнить программы, разрабатываемые пользователями.
- Какие решения приняты пользователями?
- Какая самая короткая и самая длинная программа?

УПРАЖНЕНИЕ 5: ОБОИД КОРОБКИ

В разработанной программе надо найти одинаковые повторяющиеся структуры и с их помощью изучить понятие цикла.

Важно:

- Руководить процессом поиска одинаковых повторяющихся серий команд в готовой программе.
- Время, заданное в команде, может быть разным у разных Робоплатформ.
- Попробуйте также команды, ограничивающие количество шагов мотора.
- В пункте «Эксперимент!» испытайте команду **всегда**.

В упражнениях применяются те же методы, что и в предыдущей части.

По мере усложнения заданий обращайтесь внимание на циклы, используйте новые команды и структуры программирования.

УПРАЖНЕНИЕ 6: ДВИЖЕНИЕ ВОДЬ ФИГУР

Тренируемся использовать циклы в одинаковых, но более сложных заданиях. Также учимся функции «бесконечный цикл».

Важно:

- Научиться планированию программы до ее построения.

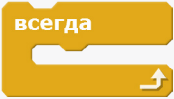
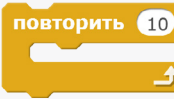
УПРАЖНЕНИЕ 7: ЗАКРЕПЛЕНИЕ ЗНАНИЙ

С помощью выполнения различных дополнительных заданий происходит закрепление изученных знаний. Эти задания рекомендуется выполнять отдельно от упражнений 4, 5 и 6. Для усвоения знаний также рекомендуется возвращаться к ним на дальнейших этапах обучения.

Важно:

- Уже на данном этапе обучения следует начать планирование и реализацию своих заданий и программ. Группы учеников могут придумывать задания друг для друга и распределять их для выполнения другим группам. Создание собственных заданий закрепляет изученные знания и помогает в разработке новых приложений.

НОВЫЕ КОМАНДЫ

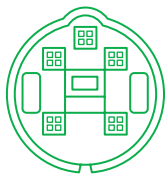
	Команда всегда (выражение повторения) повторяет серию команд, расположенных внутри команды, до тех пор, пока выполнение всей программы не будет завершено, т.е. с точки зрения программы — вечно.
	Команда повторить_ повторяет расположенную внутри серию команд желаемое количество раз (на рисунке — 10 раз).

ТЕМЫ ДЛЯ БЕСЕД ВО ВРЕМЯ И ПОСЛЕ УПРАЖНЕНИЙ

- Какие стратегии решений применялись в упражнениях? Какая самая быстрая и точная программа в каждом упражнении?
- Как планирование влияет на разработку программы? Планирование — это важный этап в различных ситуациях, связанных с программированием. Чем сложнее задание, тем важнее становится этап планирования.

РОБОТОТЕХНИКА В ЖИЗНИ

- Промышленные роботы повторяют одинаковые программы с момента включения агрегата до его выключения. Выясните, какие промышленные роботы существуют.



Упражнение 4: Обход препятствия

ЦЕЛИ

- Использование последовательных команд
- Планирование конструктивной программы

ЗАДАНИЕ 1

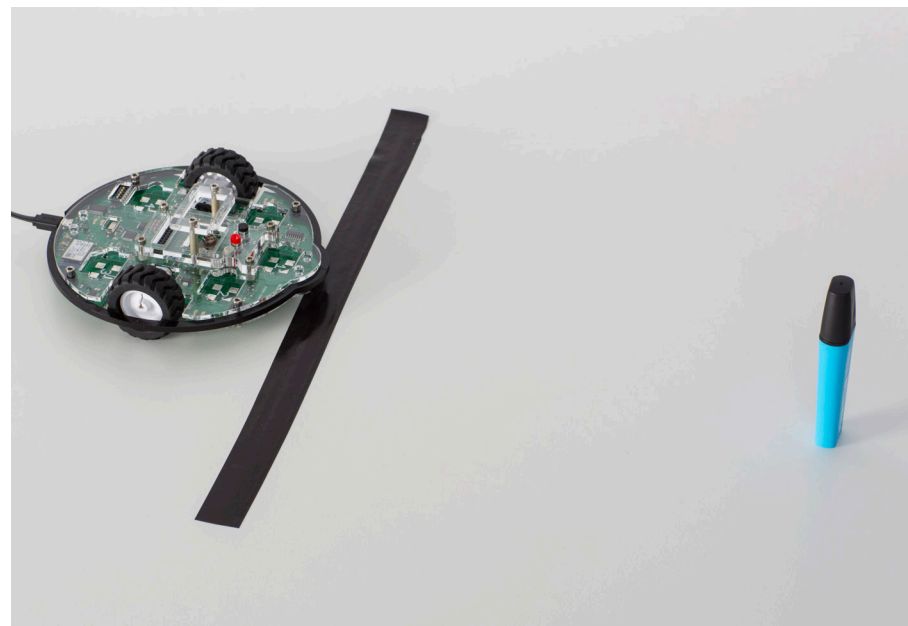
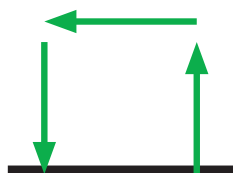
- Сделайте на полу дорожку, как показано на картинке.
- Подумайте, каким должен быть путь движения Робоплатформы.
- Запрограммируйте Робоплатформу так, чтобы она обходила препятствие самостоятельно без управления стрелками и вернулась назад к стартовой черте.

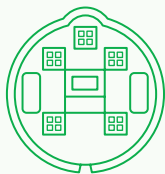
ЗАДАНИЕ 2

Измените трассу, передвигая препятствия или добавляя на трассе новые препятствия.

ПОДСКАЗКА

Робоплатформа сначала подходит к препятствию, поворачивается и обходит его и возвращается за стартовую черту. Подумайте: какие команды понадобятся для выполнения задания?





Упражнение 4: Обход препятствия

РЕШЕНИЕ

Задачу можно решить многими различными способами. Это один из вариантов решения.

когда клавиша пробел ▼ нажата

уст направление Робота вперед ▼

моторы вкл на 3 секунд

уст направление Робота налево ▼

моторы вкл на 0.5 секунд

уст направление Робота вперед ▼

моторы вкл на 2 секунд

уст направление Робота налево ▼

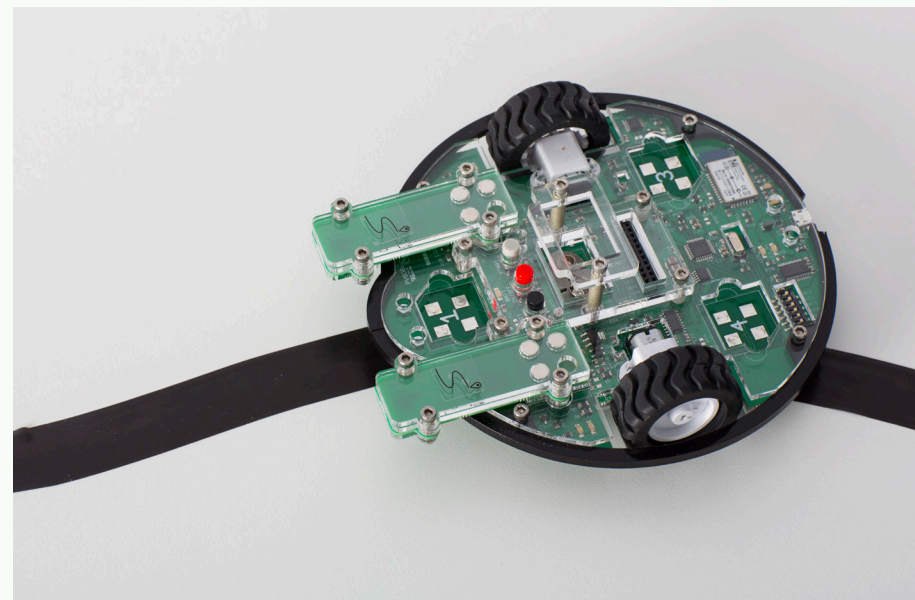
моторы вкл на 0.5 секунд

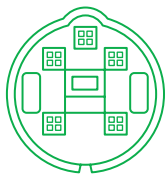
уст направление Робота вперед ▼

моторы вкл на 3 секунд

ЭКСПЕРИМЕНТ!

Сделайте из малярной ленты разные трассы на полу и опробуйте их.





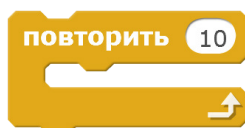
Упражнение 5: Объезд коробки

ЦЕЛИ

Использование цикла в программе.

ПОДСКАЗКА

Команда **повторить 10** находится в меню команд **Управление**.



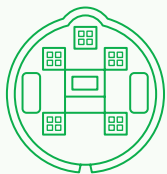
ЗАДАНИЕ 1

- Положите коробку на пол.
- Подумайте, какой должна быть трасса для прохождения ее Робоплатформой.
- Запрограммируйте Робоплатформу так, чтобы она объехала коробку один раз.

ЗАДАНИЕ 2

- Найдите в своей программе команды, которые повторяются несколько раз.
- Сделайте из этих команд, а также из команды **повторить_** новую программу.
- Пробуйте!

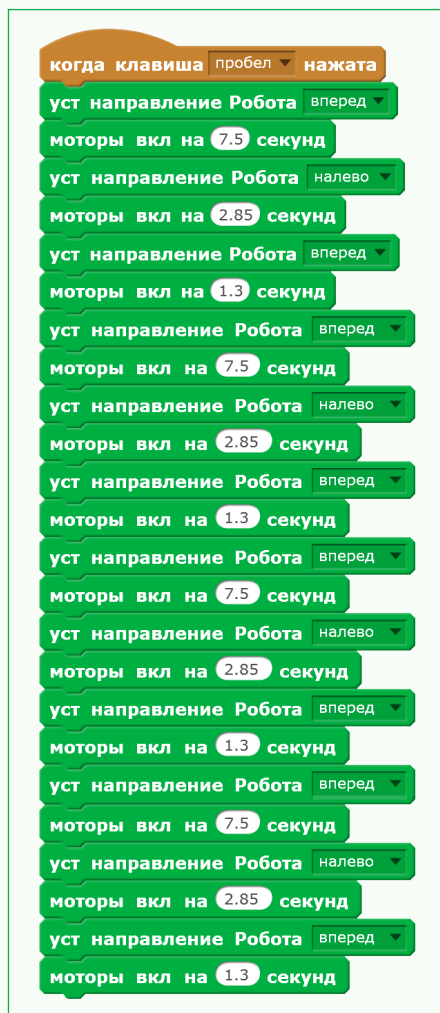




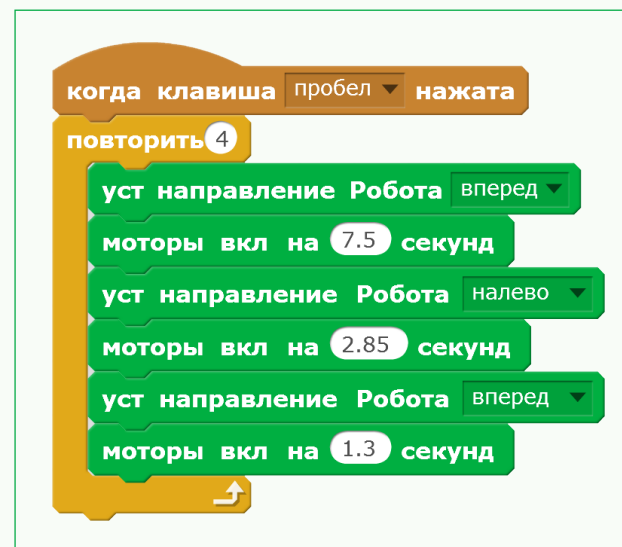
Упражнение 5: Объезд коробки

РЕШЕНИЕ 1

Время зависит от длины трассы.



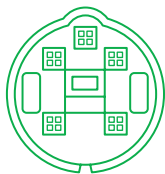
РЕШЕНИЕ 2



ЭКСПЕРИМЕНТ!

Сделайте из малярной ленты разные трассы на полу, и пусть Робоплатформа пройдет эти трассы.





Упражнение 6: Движение вдоль фигур

ЦЕЛИ

- Использование цикла в программе
- Планирование конструктивной программы

ЗАДАНИЕ 1

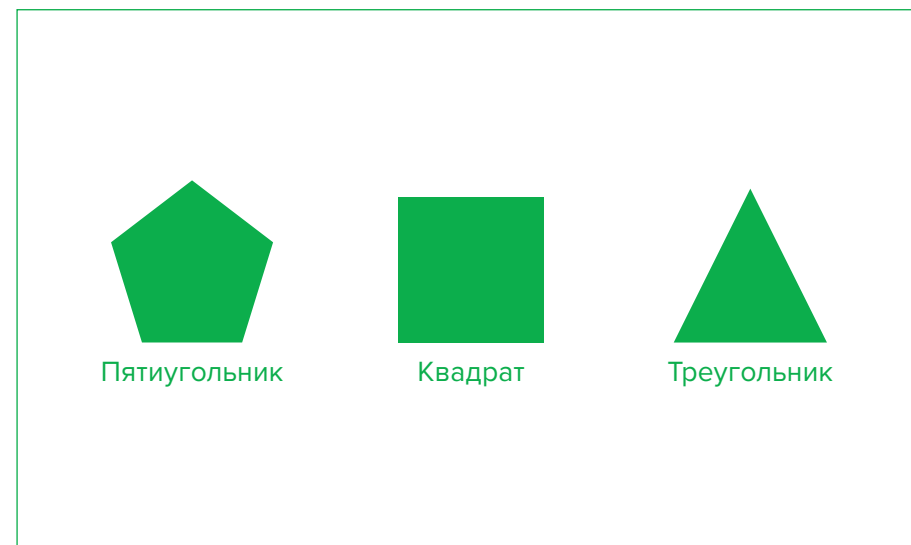
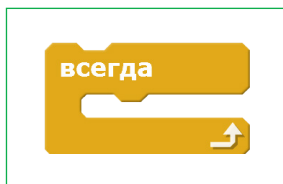
- Сделайте на полу из малярной ленты фигуры с равными сторонами, как показано на рисунке.
- Подумайте, как должна двигаться Робоплатформа для объезда фигур по периметру.
- Запрограммируйте Робоплатформу на движение вдоль фигуры таким образом, чтобы одно из колес Робоплатформы было всегда на ленте.

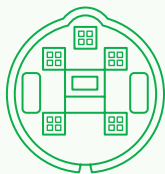
ЗАДАНИЕ 2

Запрограммируйте Робоплатформу так, чтобы она объезжала коробку вечно.

ПОДСКАЗКА

Команда **всегда** находится в меню команд **Управление**.

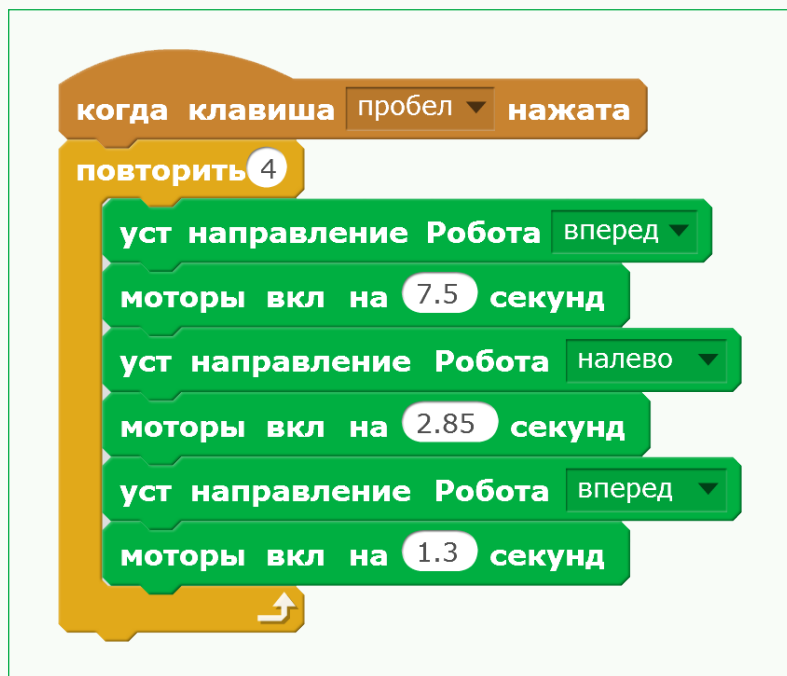




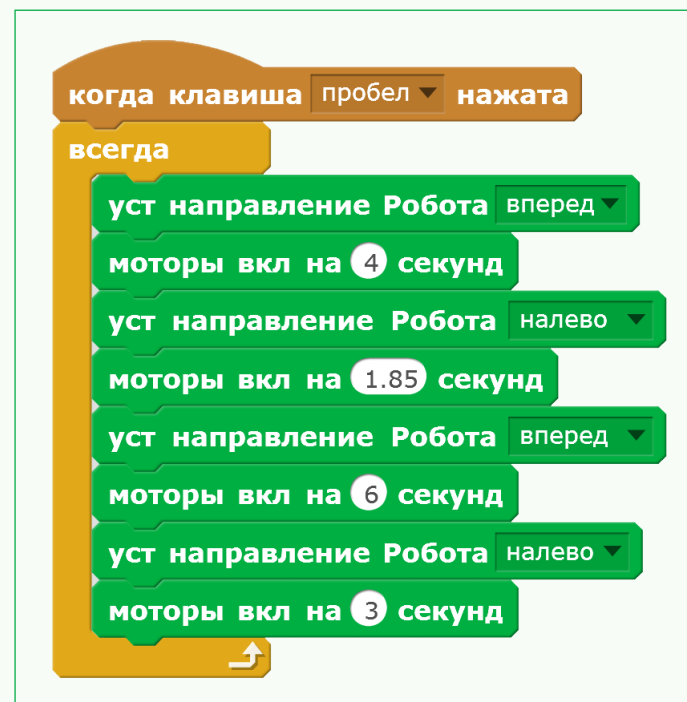
Упражнение 6: Движение вдоль фигур

РЕШЕНИЕ 1

Задаваемое время прохождения Робоплатформой трассы зависит от размера созданной на полу фигуры.



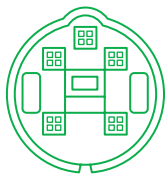
РЕШЕНИЕ 2



ЭКСПЕРИМЕНТ!

Выполните это задание с какой-нибудь другой формой трассы.





Упражнение 7: Закрепление знаний

ЦЕЛИ

- Закрепление навыков
- Оценка и измерение пути

ЗАДАНИЕ 1: ГОНКИ

- Сделайте на полу слаломную трассу и опробуйте ее. Насколько быстро вы проходите трассу?
- См. видеоролик: <https://vimeo.com/141896031>.
- С помощью каких команд вы заставили Робоплатформу двигаться максимально быстро?

ЗАДАНИЕ 2: РАСЧЕТ РАССТОЯНИЯ

- Постройте трассу для объезда препятствия.
- До начала программирования выясните, какое расстояние Робоплатформа проходит за одну секунду по прямой и на поворотах.
- Сделайте расчет по формуле:
Общий путь : путь за 1 секунду = общее время
- Учтите, что в команде можно указать также десятичные числа. Для этого вместо запятой ставьте точку!

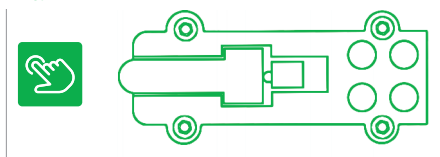
ЗАДАНИЕ 3

Придумайте задание, в котором можно использовать полученные навыки.



4. Датчик касания

РИС. 1



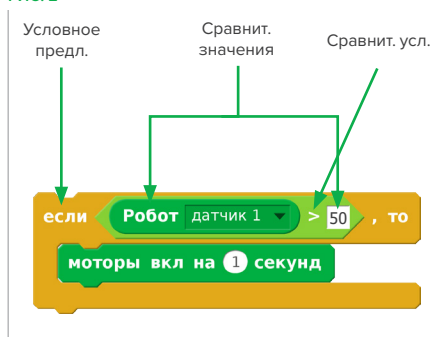
Датчик касания опознает давление на язычок или высвобождение язычка, расположенного на передней части датчика (РИС. 1 — датчик касания). С помощью датчика касания Робоплатформу можно запрограммировать на опознавание окружающей среды и реагирование при соприкосновении с чем-либо.

После ознакомления с циклом ученики знакомятся с условными операторами и используются в них условиями с помощью датчика касания. С помощью условных операторов программа настраивается для работы по заданным условиям. Условия сравнения формируют знакомые из математики знаки сравнения (<, >, =) и два сравниваемых выражения, т.е. значения. При работе с Робоплатформой при помощи датчиков можно легко научиться использовать условные операторы.

Значения датчика касания тестируются следующим образом:

- Закрепите датчик на Робоплатформе.
- В программе выберите квадратик датчика и поставьте крестик.
- Протестируйте, какое значение находится в базовом положении и какое — при нажатом язычке, расположенном на краю датчика.
- Значения колеблются в пределах от 0 до 100 в зависимости от приложенного давления (в качестве сравнительного значения обычно используют значение 50).

РИС. 2



Пример (РИС. 2)

Робоплатформа включает моторы, если значение датчика касания более 50.

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 8: ВКЛЮЧАЕМ РОБОПЛАТФОРМУ ДАТЧИКОМ КАСАНИЯ

Ученики знакомятся с работой датчиков, и в данном случае с работой датчика касания. Они учатся подключать датчик к Робоплатформе, составлять серию команд, которая будет считывать значения датчика, и применять готовые звуковые команды.

Важно:

- Объяснить принципы действия датчиков.
- Найти и соединить друг с другом команды, необходимые для программирования.

УПРАЖНЕНИЕ 9: СТОЛКНОВЕНИЯ С ПРЕПЯТСТВИЕМ

Закрепляется навык использования в программах датчика касания путем знакомства с двумя новыми командами/условными предложениями: **если<_>,то_** и **если<_>,то_иначе**.

Важно:

- Руководить поиском условных предложений из группы команд управления.
- Программу можно исполнять с использованием двух разных условных предложений.
- Поразмышлять над различиями и удобством использования составленных программ. Команда **если<_>,то_иначе** более удобна в использовании почти во всех упражнениях, где необходимы две различные альтернативные функции в зависимости от положения датчика. Важно выучить это!

УПРАЖНЕНИЕ 10: ЗАКРЕПЛЕНИЕ ЗНАНИЙ

С помощью выполнения различных дополнительных заданий происходит закрепление изученных знаний. Эти задания рекомендуется выполнять отдельно от упражнений 8 и 9. Для усвоения знаний к ним рекомендуется возвращаться повторно.

Важно:

- Выполнение собственных заданий по программированию.
- Задание 2 — создать крепление карандаша к Робоплатформе, оно будет использовано в дальнейшем.

НОВЫЕ КОМАНДЫ

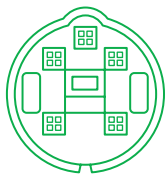
	Команда ждать до<_> прерывает исполнение команд до тех пор, пока заданное условие не получит значение истинности истина .
	Команда (условное выражение) если<_>, то исполняет серию команд только в том случае, если заданным значением в момент исполнения команды является истина .
	Команда если<_>, то...иначе (разветвленное условное выражение) исполняет серии команд избранным образом согласно заданному для команды значению истинности. Если значение — истина , то исполняется первое, если ложь , то исполняется последнее.
	Условия сравнения размещаются в командах условных операторов. Они также являются так называемыми логическими выражениями, определяющими выполнение условия (истина) или невыполнение (ложь). Иными словами, если в данных упражнениях условие не выполнено, то серия команд не исполняется.
	Выражение Робот датчик N возвращает значение выбранного датчика на момент времени исполнения команды. Возвращаемое значение можно использовать в условия сравнения, при этом можно сравнивать значение датчика с каким-нибудь числом или значением другого датчика.
	Команда играть звук_ начинает воспроизведение звука, выбранного в меню, и сразу же продолжает работу программы.
	Команда играть звук_до завершения полностью воспроизводит звук, выбранный в меню, и только потом продолжает работу программы.

ТЕМЫ ДЛЯ БЕСЕД ВО ВРЕМЯ И ПОСЛЕ УПРАЖНЕНИЙ

- Какие аппараты в повседневной жизни включаются посредством датчика касания? А какие выключаются?
- Придумайте условные выражения, применяемые в повседневной жизни. Например: Если не почистишь зубы, то будет кариес.
- Вспомните различные переключатели, которые были рассмотрены при выполнении первых упражнений. Программируемыми переключателями являются, в частности, пульт дистанционного управления и телевизор.

РОБОТОТЕХНИКА В ЖИЗНИ

- В данных упражнениях в качестве сравнительных условий использовался логический тип данных (булев тип). Выясните, что это означает и откуда произошло название.



Упражнение 8: Включаем Робоплатформу датчиком касания

ЦЕЛИ

- Приведение Робоплатформы в движение датчиком касания
- Подсоединение датчика к порту для датчиков на Робоплатформе
- Использование показаний датчика в команде **ждать до_**

ЗАДАНИЕ 2

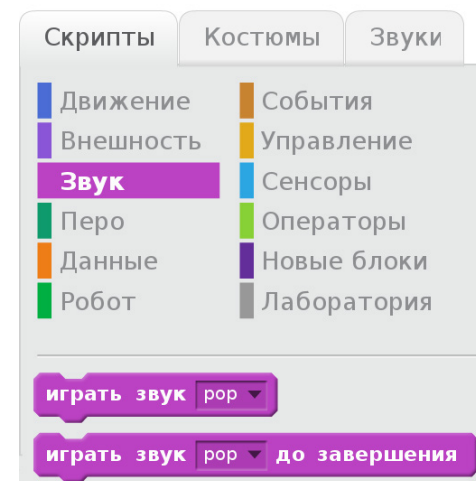
- Когда Робоплатформа придет в движение по нажатию датчика касания, внесите в программу какой-нибудь веселый звук, который будет звучать перед отправлением в путь.
- Готовые звуки находятся во вкладке **Звуки**.

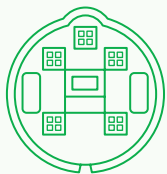
ЗАДАНИЕ 1

- Подсоедините датчик к порту N°1 для датчиков на Робоплатформе.
- В программе поставьте крестик в квадрате **Датчик 1**.
- Составьте программу, по которой Робоплатформа будет двигаться только при нажатии датчика касания.

ПОДСКАЗКА

Надавите на датчик касания. Изучите на дисплее изменения параметра датчика 1. В меню **Операторы** вы найдете подходящую команду, которую можете использовать в упражнениях.

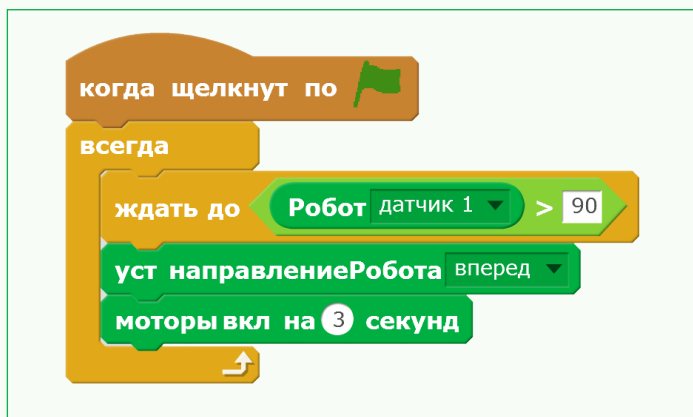




Упражнение 8: Запустите Робоплатформу датчиком касания

РЕШЕНИЕ 1

Время, задаваемое для прохождения трассы Робоплатформой, зависит от длины созданной на полу трассы.



ЭКСПЕРИМЕНТ!

Включите на компьютере звуки и откройте (или запишите) файл Horse.

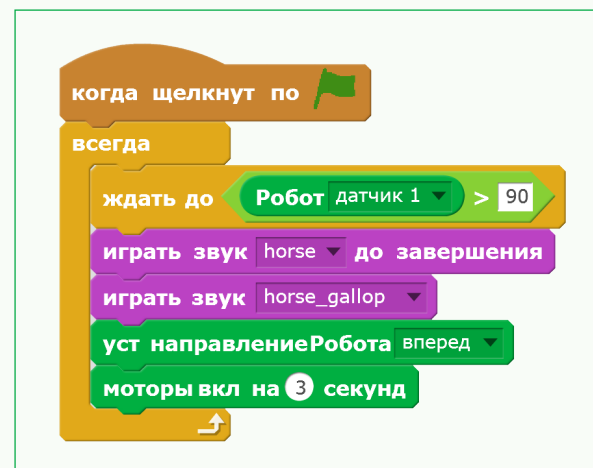
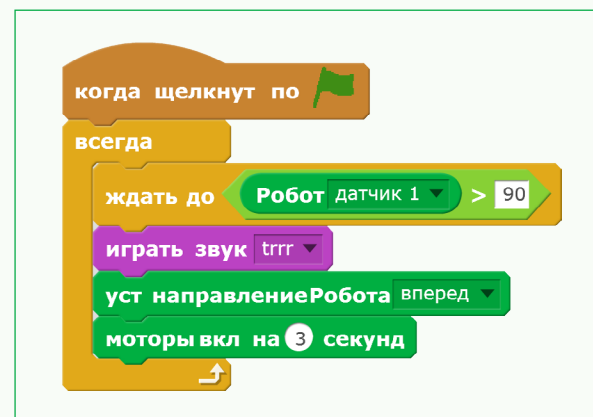
Попробуйте сделать это самостоятельно!

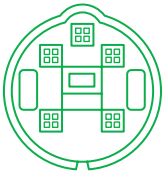
ПРОСМОТР

См. видеоролик «Робот-лошадка»:

<http://1drv.ms/1VmqvKY>

РЕШЕНИЕ 2





Упражнение 9: Столкновения с препятствием

ЦЕЛИ

- Изменение направления движения Робоплатформы при помощи датчика касания
- Использование параметра датчика в условных предложениях **если<_>, то** и **если<_>, то_иначе** (условное выражение можно найти в группе команд **Управление**)

ЗАДАНИЕ 1

- Прикрепите датчик к передней части Робоплатформы, к порту N°1 для датчиков.
- В программе поставьте крестик в квадрате **Датчик 1**.
- Запрограммируйте Робоплатформу так, чтобы она двигалась вперед. Ударившись о препятствие, Робоплатформа отпрянет,

повернется и продолжит движение в новом направлении.

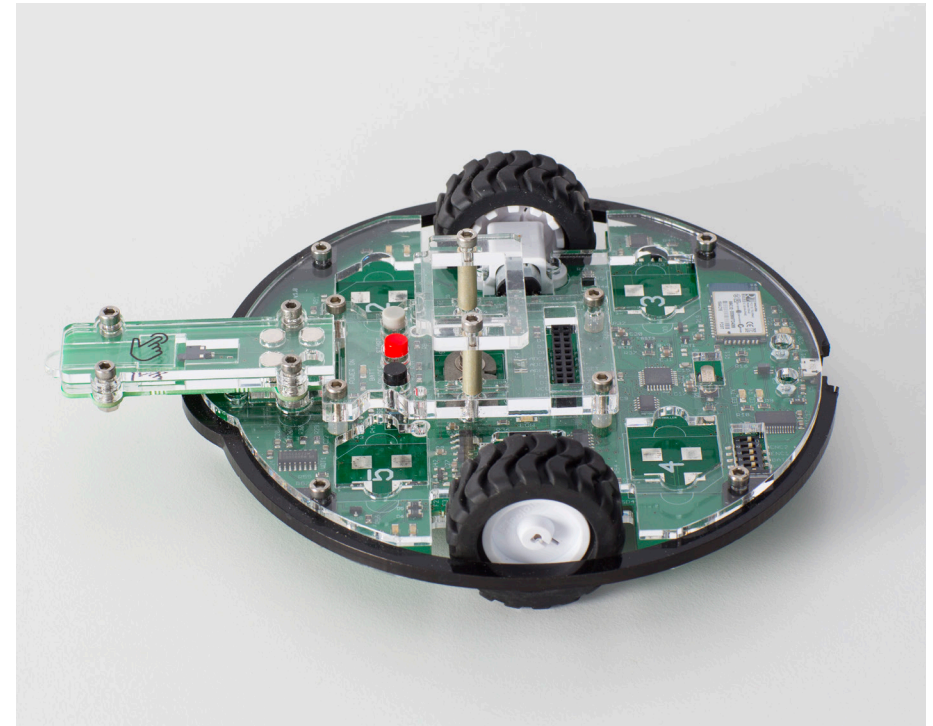
- Используйте условный оператор **если<_>, то**.

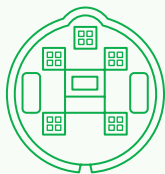
ЗАДАНИЕ 2

- Сделайте то же задание, но используя команду **если<_>, то_иначе**.
- Сравните, чем программы отличаются друг от друга.

ПОДСКАЗКА

Подумайте сначала: какое условие должно быть при ненажатом датчике и что Робоплатформа должна при этом делать?



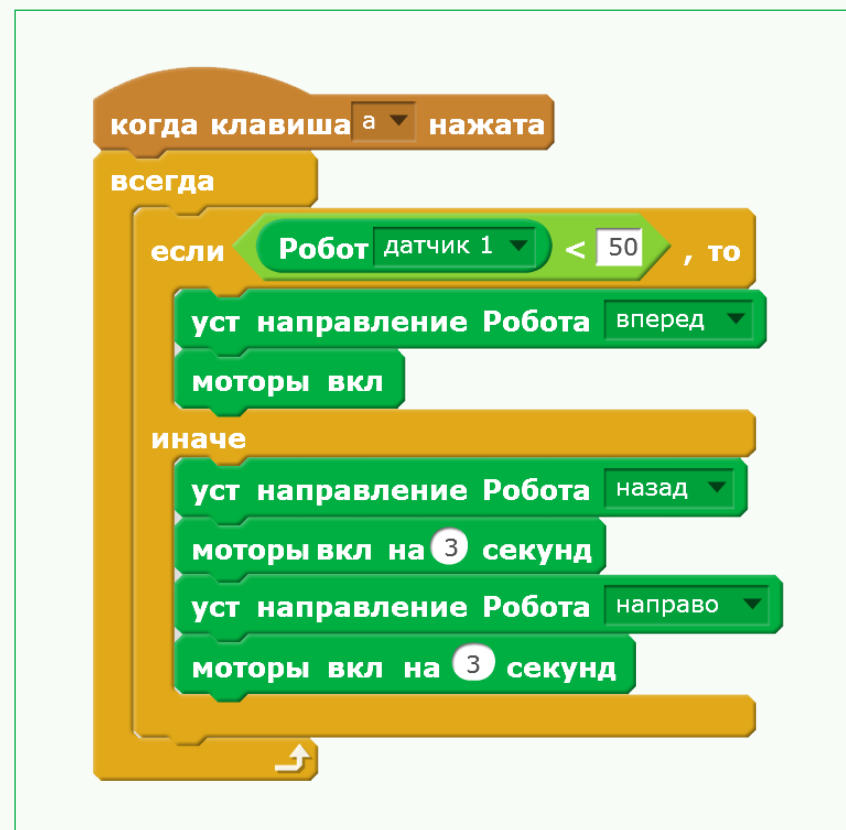


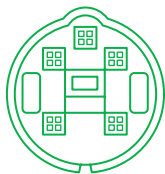
Упражнение 9: Столкновения с препятствием

РЕШЕНИЕ 1



РЕШЕНИЕ 2





Упражнение 10: Закрепление знаний

ЦЕЛИ

Закрепление полученных навыков

ЗАДАНИЕ 1: ОЗВУЧИВАЕМ СТОЛКНОВЕНИЯ

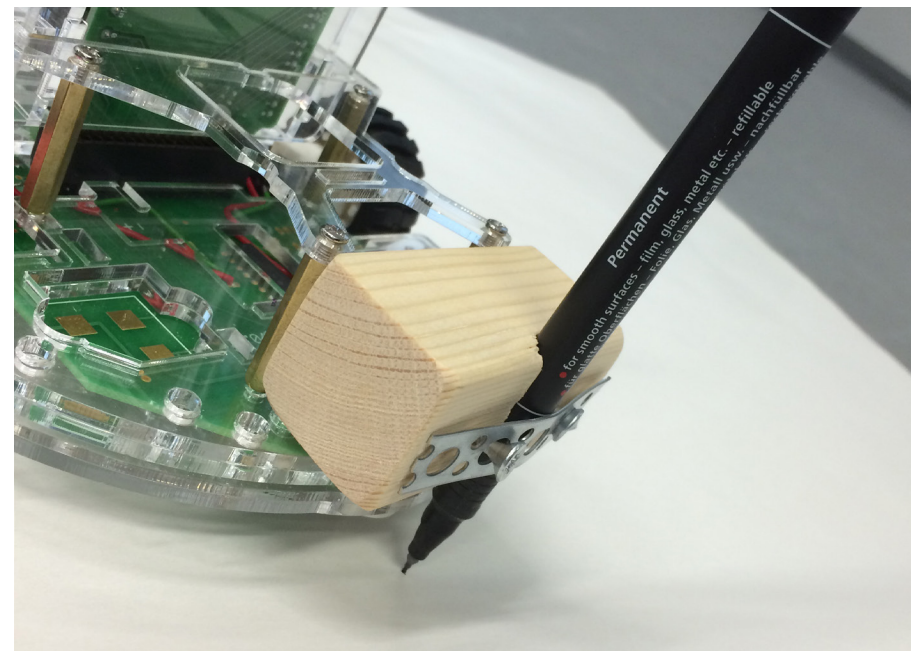
- Создайте для Робоплатформы программу движения.
- Добавьте в программу подходящий звук, который будет раздаваться после столкновения с препятствием.

ЗАДАНИЕ 2: РИСУЕМ МАРШРУТ

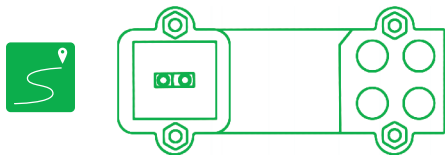
Приделайте к Робоплатформе крепление для карандаша, и Робоплатформа, двигаясь, будет рисовать собственный маршрут движения.

ЗАДАНИЕ 3: ИЗОБРЕТАЙТЕ!

Придумайте задание, в котором можно использовать полученные навыки.



5. Датчик линии



Навыки, полученные с помощью датчика касания, закрепляются путем использования датчика линии. Данный датчик распознает различия оттенков светлого (белого) на поверхности, посылая инфракрасные лучи света и измеряя количество инфракрасного излучения, отраженного от поверхности. Глаз не может наблюдать инфракрасное излучение, но оно окружает нас повсюду. Поэтому датчик реагирует и на другие источники света, например, на солнечный свет (попробуйте!). С помощью переданной датчиком информации Робоплатформу можно запрограммировать так, чтобы она перемещалась вдоль линии на полу.

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 11: СЛЕДУЕМ ЗА ЛИНИЕЙ

Используем знания, полученные на упражнениях с датчиком касания.

Важно:

- Обратить внимание на сходство в программировании различных датчиков.
- Изучить изменения значений датчика на светлой и темной поверхности.
- Дать время на планирование перед программированием, чтобы лучше понять, как Робоплатформа распознает светлую и темную поверхность.

Значения датчика тестируются следующим образом:

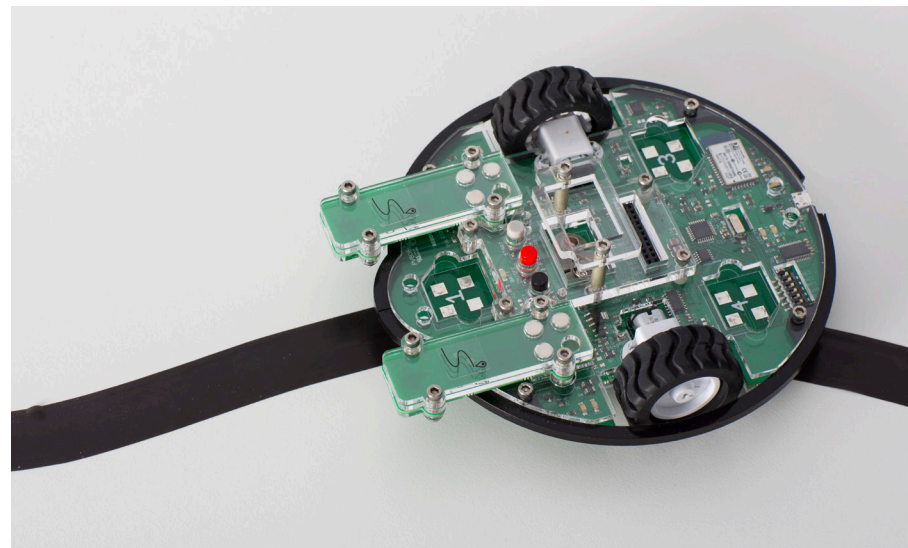
- Прикрепите датчик к Робоплатформе.
- В программе отметьте крестиком квадрат датчика.
- Протестируйте, какое значение датчик покажет на темной поверхности и какое — на светлой.
- Значения датчика колеблются в пределах от 0 до 100, в зависимости от освещенности поверхности.

УПРАЖНЕНИЕ 12: ПУТИ И МАРШРУТЫ

Используем два датчика линии в одной программе.

Важно:

- Координировать использование полученных знаний.



УПРАЖНЕНИЕ 13: ЗАКРЕПЛЕНИЕ ЗНАНИЙ

С помощью выполнения различных дополнительных заданий происходит закрепление изученных знаний. Эти задания рекомендуется выполнять отдельно от упражнений 4, 5 и 6. Для усвоения знаний к ним рекомендуется возвращаться повторно на дальнейших этапах обучения.

Важно:

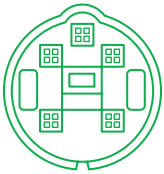
- Выполнять свои задания по программированию.

ТЕМЫ ДЛЯ БЕСЕД ВО ВРЕМЯ И ПОСЛЕ УПРАЖНЕНИЙ

- Чем отличается отражение света от светлой и темной поверхности?
- Что произойдет, если перед датчиком расположить цветную пленку?
- Попробуйте, опознает ли датчик цвета. Какие значения вы получили?
- Как расстояние между датчиком и опознаваемой им поверхностью влияет на значения датчика?

РОБОТОТЕХНИКА В ЖИЗНИ

- Выясните, как работает датчик линии.



Упражнение 11: Следуем за линией

ЦЕЛИ

- Чтение показаний датчика движения Робоплатформы
- Использование значения **если<_>, то_иначе** в условии

ЗАДАНИЕ 1

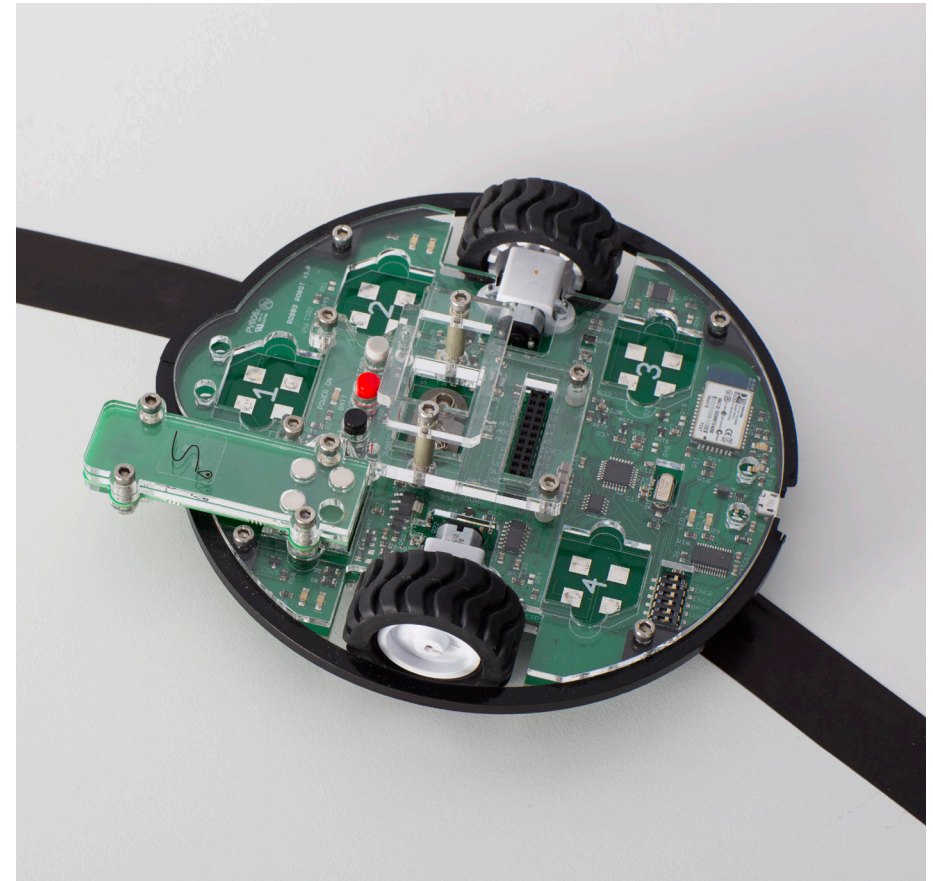
- Установите на Робоплатформе датчик линии.
- Поводите Робоплатформу над черной и белой поверхностью.
- Изучите изменения параметров света.
- Сделайте программу, по которой Робоплатформа с помощью датчика будет следовать по черной линии (см. рисунок справа).

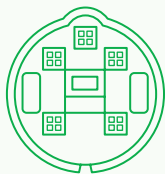
ЗАДАНИЕ 2

Сделайте трассу, в которой повороты черной линии будут более крутыми. Подумайте, как изменить программу, чтобы Робоплатформа проходила и крутые повороты.

ПОДСКАЗКА

- Вначале подумайте, что Робоплатформа будет предпринимать, пока датчик не опознает под собой черную линию.
- Используйте условное выражение **если<_>, то_иначе**.

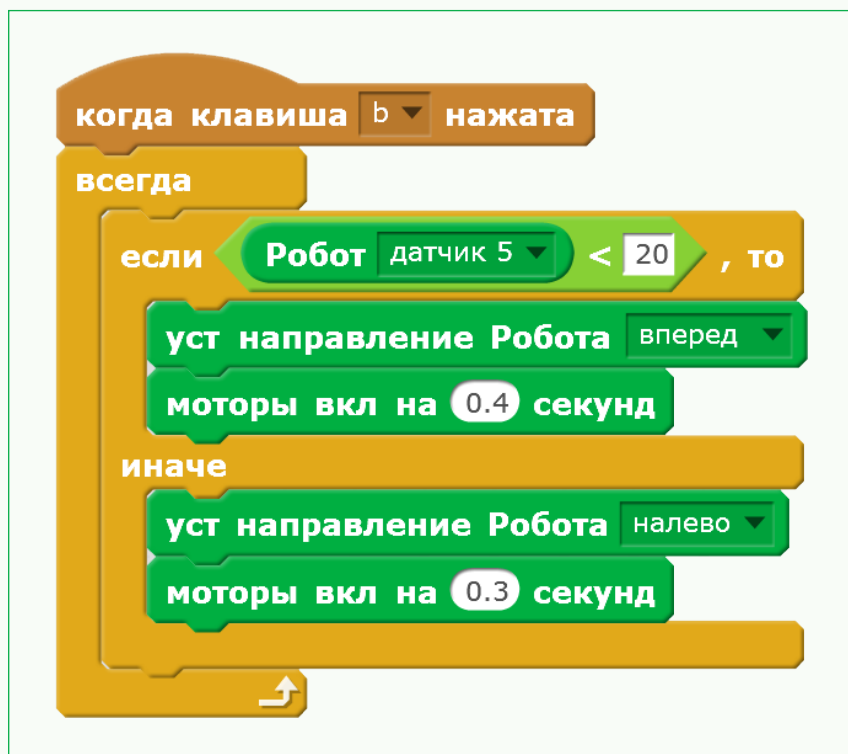




Упражнение 11: Следуем за линией

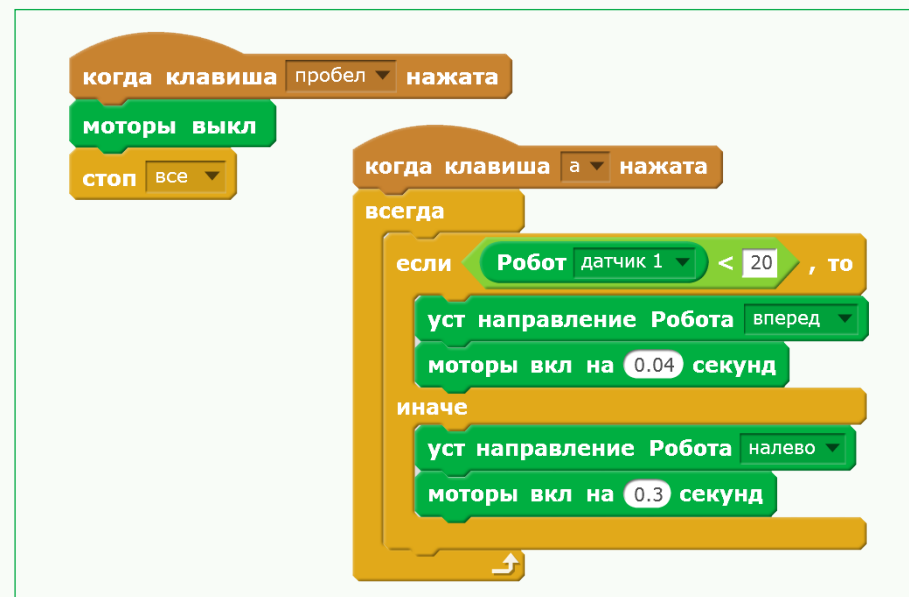
РЕШЕНИЕ 1

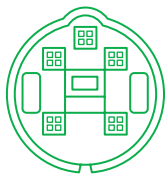
Учтите, что значения датчика могут сильно колебаться в зависимости от окружающего освещения.



РЕШЕНИЕ 2

Изменяя время движения вперед или при поворотах, вы сможете сделать так, чтобы Робоплатформа поворачивалась более резко. Однако учтите, что если указанная выше команда поворачивает Робоплатформу влево за 0,3 секунды, то на крутом повороте Робоплатформа может выехать за линию и двигаться по неправильной стороне линии. Для применения кнопки «Экстренная остановка» используйте пробел.





Упражнение 12: Пути и маршруты

ЦЕЛИ

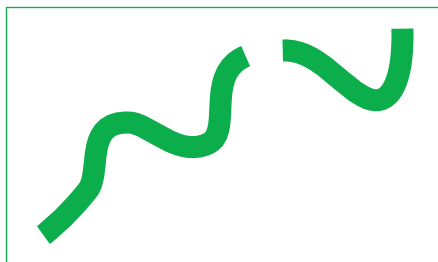
- Использование двух одинаковых датчиков линии в одной программе
- Программирование Робоплатформы на следование по черной линии с изгибами в обе стороны

ЗАДАНИЕ 1

- Подключите датчики линии к портам №2 и 5.
- Отметьте крестиком квадраты **Датчик 2** и **5** в программе.
- Поводите датчики над черным и белым фоном. Исследуйте изменения показаний датчиков на мониторе.
- Запрограммируйте Робоплатформу так, чтобы она следовала по черной линии, нарисованной на белом фоне. Линия может изгибаться в обе стороны.

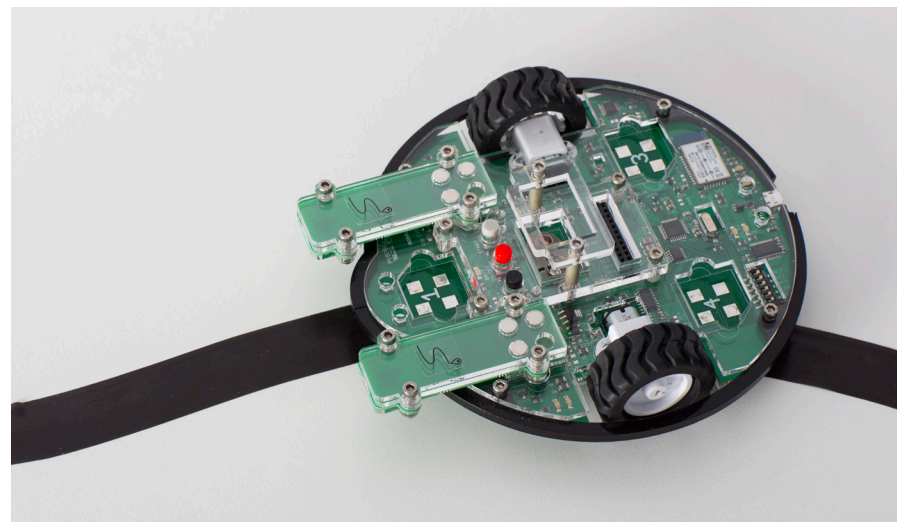
ЗАДАНИЕ 2

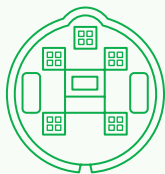
Задача усложнится, если черная линия прервется. Подумайте, какой должна быть линия до того, как она временно прервется.



ПОДСКАЗКА

Решите проблему сначала «с точки зрения» одного датчика!





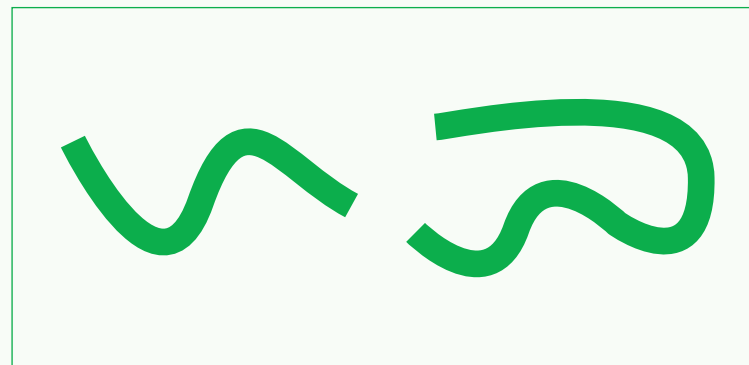
Упражнение 12: Пути и маршруты

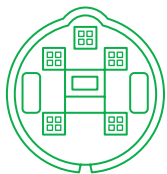
РЕШЕНИЕ 1



РЕШЕНИЕ 2

Линия, прежде чем она прервется, должна быть прямой на достаточно большом расстоянии. Следуя по прямой, Робоплатформа будет продвигаться в верном направлении и в итоге снова попадет на линию. Если линия будет иметь повороты слишком близко от места ее прерывания, Робоплатформа может продолжить движение в неверном направлении и не найдет линию.





Упражнение 13: Закрепление знаний

ЦЕЛИ

Закрепление полученных навыков

ЗАДАНИЕ 1: МАРШРУТ В ФОРМЕ ВОСЬМЕРКИ

Сделайте трассу в виде восьмерки, но так, чтобы в месте пересечения линии отсутствовали.

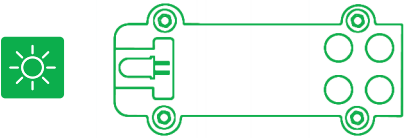


ЗАДАНИЕ 2: ИЗОБРЕТАЙТЕ!

Придумайте задачу, в которой вы сможете использовать полученные навыки.



6. Датчик освещенности



Датчик освещенности измеряет световой поток, поступающий на переднюю часть датчика (угол обзора линзы датчика составляет примерно 30 градусов). При помощи получаемых от датчика данных Робоплатформу можно запрограммировать таким образом, чтобы она реагировала на колебания светового потока.

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 14: КОРОБКА С СЮРПРИЗОМ

Используем полученные навыки в работе с новым датчиком, учимся делать свое озвучивание и повторять его на упражнениях по программированию.

- Важно:**
- Вспомнить команду **ждать до_**.
 - Изучить изменения значений датчика освещенности вблизи и вдали от него.

- Значения датчика тестируются следующим образом:**
- Установите датчик на Робоплатформе.
 - Выберите квадрат датчика освещенности в программе и поставьте в нем крестик.
 - Протестируйте, какое значение датчик показывает в темноте (например, внутри ладоней) и при хорошем освещении.
 - Значения датчика колеблются в пределах от 0 до 100 в зависимости от поступающего на датчик светового потока.

УПРАЖНЕНИЕ 15: РОБОПЛАТФОРМА ПОВОРАЧИВАЕТСЯ К СВЕТУ

Используем два датчика освещенности в одной программе. Производим измерения для программирования и выстраиваем программу на основе наблюдаемых данных путем сравнения двух разных датчиков.

- Важно:**
- Для выполнения задания потребуются два карманных фонарика и помещение, которое можно на время сделать неосвещенным.
 - Руководить решением проблемы сначала без Робоплатформы.
 - Обдумать вместе с детьми связанное с решением логическое выражение сравнения. Понимание улучшится, если одновременно использовать две Робоплатформы, одна из которых имеет логическое выражение сравнения, а другая — нет.

УПРАЖНЕНИЕ 16: СТОРОЖЕВАЯ СОБАКА

Используем ранее полученные навыки и учимся использовать случайное число.

- Важно:**
- Убедитесь, что пользователи понимают понятие «случайные числа» в программировании. Выражение случайных чисел действует наподобие игрального кубика: программа RobboScratch тянет жребий между значениями, заданными для числа.

УПРАЖНЕНИЕ 17: ЗАКРЕПЛЕНИЕ ЗНАНИЙ

С помощью выполнения различных дополнительных заданий ученики закрепляют изученные знания. Данные задания рекомендуется выполнять отдельно от упражнений 14, 15 и 16. Для усвоения знаний к ним рекомендуется возвращаться повторно.

- Важно:**
- Выполнять свои задания по программированию.

НОВЫЕ КОМАНДЫ

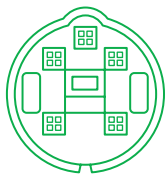
	Математическое выражение. Можно использовать, например, в подсчете переменных значений и в сравнительных условиях.
	Выражение выдать случайное от_до_ дает случайное число из заданного диапазона значений. Выражение можно поместить внутри условия сравнения.
	Команда установить громкость_% устанавливает громкость звука по шкале 0–100%.
	Команда остановить все звуки останавливает все звуки программы.

ТЕМЫ ДЛЯ БЕСЕД ВО ВРЕМЯ И ПОСЛЕ УПРАЖНЕНИЙ

- Чем отличается датчик линии от датчика освещенности?
- Почему в упражнении 15 применяется выражение сравнения?

РОБОТЕХНИКА В ЖИЗНИ

- Выясните вместе, как работает датчик освещенности (фотодиод).



Упражнение 14: Коробка с сюрпризом

ЦЕЛИ

Использование датчика освещенности, измеряющего световой поток, который поступает на переднюю часть

ЗАДАНИЕ 1

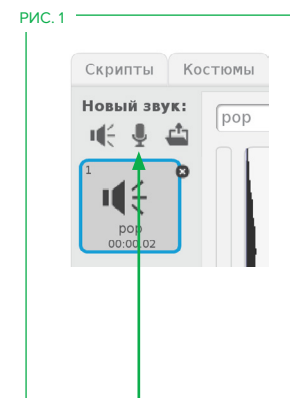
- Подключите датчик освещенности к порту №5.
- Отметьте **Датчик 5** в программе.
- Подвигайте Робоплатформой перед источником света, под столом и т.д. Исследуйте изменение показаний датчика.
- Положите Робоплатформу в пустую картонную коробку с крышкой.
- Составьте программу, в которой Робоплатформа начинает движение, когда открывают коробку; важно, чтобы световой поток был достаточно большим.

ЗАДАНИЕ 2

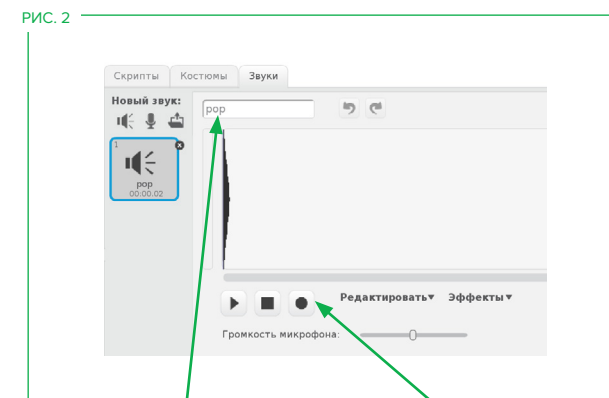
Научитесь делать собственное озвучивание и использовать его в программах (**РИС. 1 и 2**).

ПОДСКАЗКА

- В задании 1 можно использовать команду **ждать до_**.
- Создать собственную звуковую запись в задании 2 помогут рисунки (**РИС. 1 и 2**).



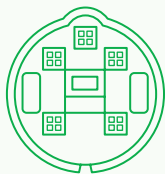
Запишите новый звук



Название записанного звука

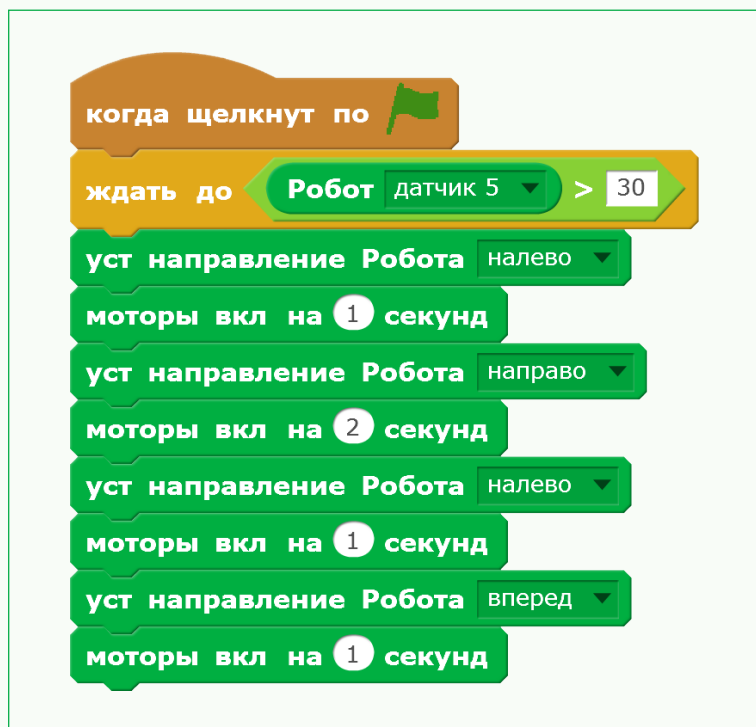
Запись



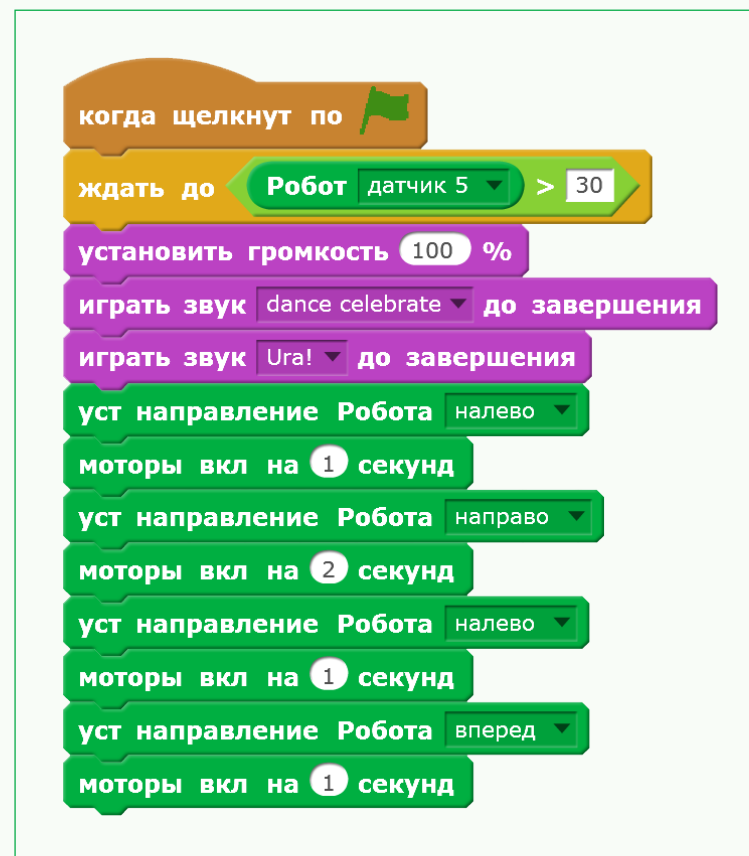


Упражнение 14: Коробка с сюрпризом

РЕШЕНИЕ 1



РЕШЕНИЕ 2

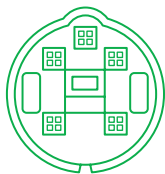


ПРОСМОТР

См. видеоролик «Робот в картонном ящике»:

<http://1drv.ms/1QRydis>





Упражнение 15: Робоплатформа поворачивается к свету

ЦЕЛИ

- Сравнение параметров двух датчиков освещенности между собой
- Управление Робоплатформой по результатам сравнения

будет каждый раз поворачиваться в том направлении, в котором показания датчика освещенности больше. Если свет будет падать на оба датчика, то Робоплатформа будет стоять на месте.

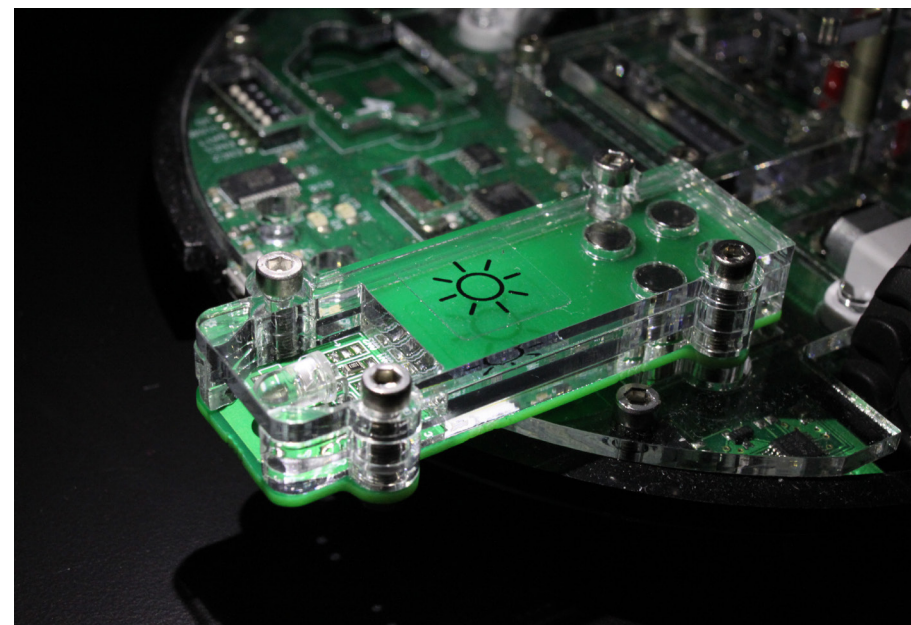
ЗАДАНИЕ

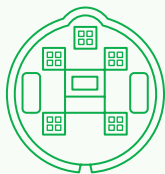
Задание выполняется в темном помещении. Вам потребуется также хороший карманный фонарик.

1. Подключите к Робоплатформе два датчика освещенности. Прочитайте показания датчиков в темном помещении.
2. Направьте свет фонарика по очереди на оба датчика и исследуйте изменение параметров освещенности. Запишите все результаты измерений.
3. Составьте программу, с помощью которой Робоплатформа

ПОДСКАЗКА

- Для задач со сравнением можно установить два разных датчика.
- Сначала создайте программу, с помощью которой Робоплатформа будет поворачиваться только в направлении одного датчика. Когда это заработает, подумайте, как добиться, чтобы Робоплатформа поворачивалась в обе стороны.





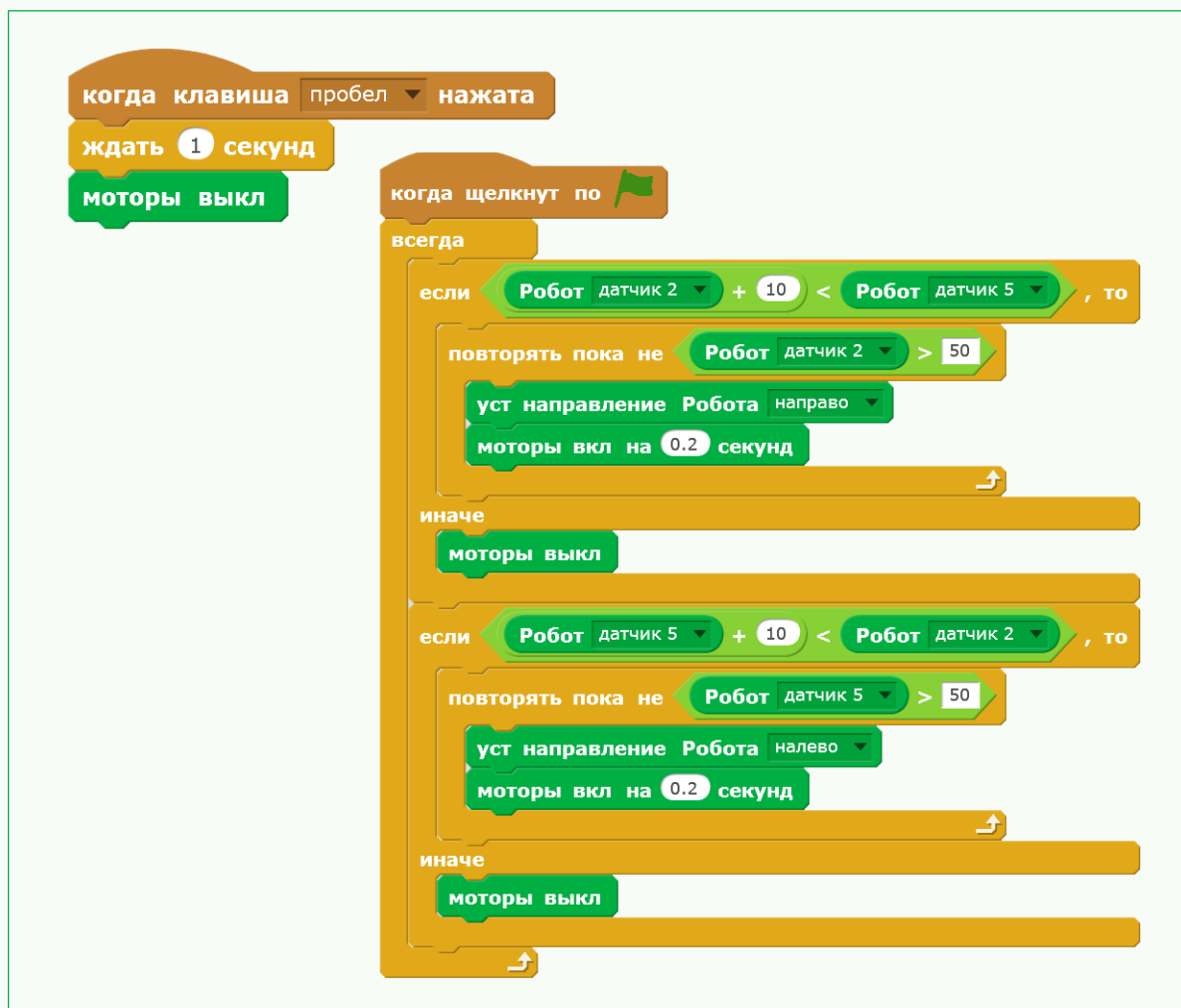
Упражнение 15: Робоплатформа поворачивается к свету

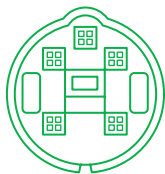
РЕШЕНИЕ

В данном решении к параметру датчика с меньшим результатом измерения добавлено число 10, чтобы Робоплатформа не поворачивалась постоянно туда-сюда, а принимала бы небольшую разницу между датчиками (показание — 10). Тогда Робоплатформа будет двигаться спокойней. Обратите также внимание, что повороты к свету будут продолжаться, пока не будет достигнута определенная освещенность (в данном случае — 50).

ПРОСМОТР

См. видеоролик:
<http://bit.ly/robbo10>





Упражнение 16: Сторожевая собака

ЦЕЛИ

- Использование случайного числа в команде **повторить_**
- Программирование игрушки «сторожевая собака», которая будет лаять голосами двух разных собак, если кто-то подходит к двери

ЗАДАНИЕ 1

- Подключите к Робоплатформе один датчик освещенности.
- Поставьте Робоплатформу перед воображаемой дверью и замерьте параметр света.
- Составьте программу, с помощью которой Робоплатформа начнет лаять при значительном уменьшении показания датчика освещенности (= кто-то приближается к двери).

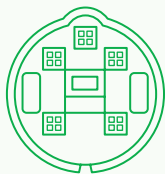
ЗАДАНИЕ 2

- Запрограммируйте Робоплатформу так, чтобы она не всегда лаяла одинаково.
- В **Операторах** вы найдете выражение для случайных чисел.

ПОДСКАЗКА

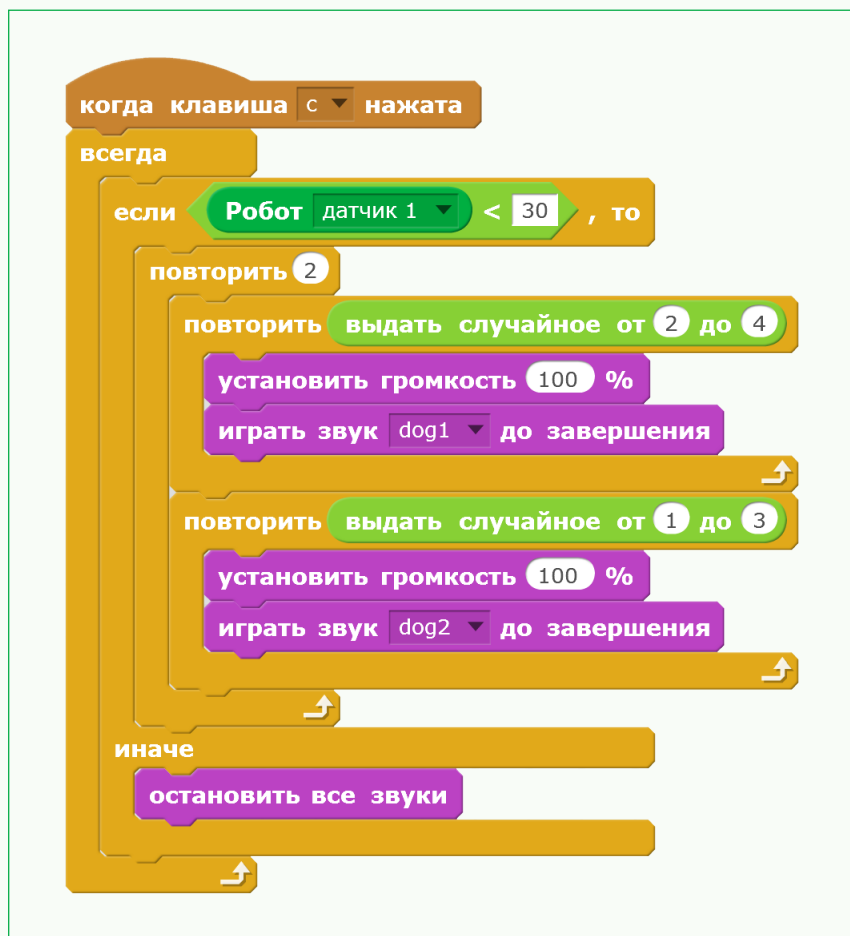
Поработайте с командой **повторить_** и немного разнообразьте лай собаки. Можете также записать свои звуки, например, речь или имитацию домашних хлопот, которыми вы отпугнете «разбойников».

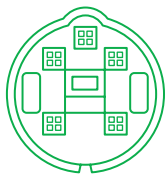




Упражнение 16: Сторожевая собака

РЕШЕНИЕ





Упражнение 17: Закрепление знаний

ЦЕЛИ

Закрепление полученных навыков.

ЗАДАНИЕ 1: ШУТКА

Спланируйте для своего друга какой-нибудь розыгрыш или сюрприз, используя датчик освещенности, голоса, движение Робоплатформы.

ЗАДАНИЕ 2: ОХРАННАЯ СИГНАЛИЗАЦИЯ

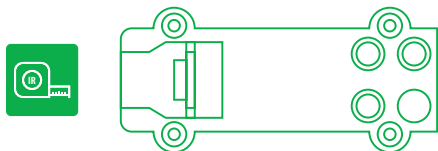
Разработайте сигнализацию от воров с помощью датчика освещенности и своих звуковых записей.

ЗАДАНИЕ 3

Придумайте задание, в котором можно использовать полученные навыки.



7. Датчик расстояния и дополнительные упражнения



В этой последней главе на примере одного упражнения рассматриваются принципы работы датчика расстояния. Датчик расстояния распознает расстояние до объекта по тому же принципу, что и датчик линии опознает степень яркости поверхности, т.е. путем восприятия инфракрасного света. С помощью полученных от датчика данных можно запрограммировать изменение направления движения Робоплатформы при приближении к объекту или, альтернативно, следовать за объектом.

Датчик расстояния и датчик линии содержат аналогичные компоненты (светодиод, действующий в зоне инфракрасного излучения, и фотодиод).

На практике диапазон датчика расстояния не так велик из-за инфракрасного излучения, содержащегося в фоновом свете, который

может легко скрыть сигнал, поступающий с объекта. На самом деле датчик лучше всего работает в темноте.

Значения датчика тестируются следующим образом:

- Прикрепите датчик к Робоплатформе.
- Отметьте крестиком квадрат датчика в программе.
- Протестируйте, как меняются показания датчика, если к датчику подносить ладони.
- Значения датчика колеблются в пределах от 0 до 100, в зависимости от расстояния до объекта (и яркости фоновое освещение).

Для применения полученных знаний после изучения принципов работы датчика расстояния ученики переходят к выполнению расширенных заданий, которые представлены в упражнении 19.

Важно:

- Исследовать изменения показаний датчика вблизи препятствия и вдали от него.
- Под руководством преподавателя ученик учится управлять траекториями движения Робоплатформы при изменении параметров программы (предельных значений датчика, угол поворота Робоплатформы и время обратного хода).
- Сравните данный датчик с датчиком касания и определите, в чем различия.

УПРАЖНЕНИЕ 19: ПРИМЕНЕНИЕ ЗНАНИЙ НА ПРАКТИКЕ

В этом последнем упражнении собраны различные задачи, в которых используются все полученные ранее навыки.

Начиная с данного упражнения, увеличивается значение творческого осмысления, планирования и составления собственных заданий и программ.

ТЕМЫ ДЛЯ БЕСЕД ВО ВРЕМЯ И ПОСЛЕ УПРАЖНЕНИЙ

- Где в быту используется датчик расстояния?
 - Радар заднего хода в автомобилях
 - Раздвижные двери и ворота
 - Двери лифтов

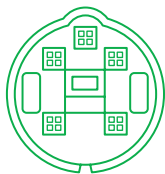
РОБОТОТЕХНИКА В ЖИЗНИ

- Выясните вместе, как работает датчик расстояния.
- Выясните, чем датчики, основывающиеся на ультразвуке, отличаются от датчиков освещенности.

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 18: РОБОПЛАТФОРМА ПУТЕШЕСТВУЕТ

Используем навыки, полученные в предыдущих упражнениях, в работе с новым датчиком.



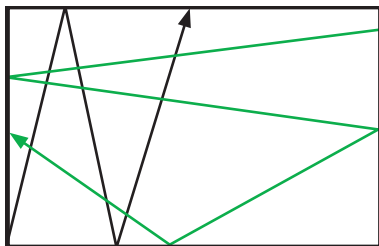
Упражнение 18: Робоплатформа путешествует

ЦЕЛИ

- Управление Робоплатформой на основе показаний датчика расстояния
- Изучение траекторий движения Робоплатформы

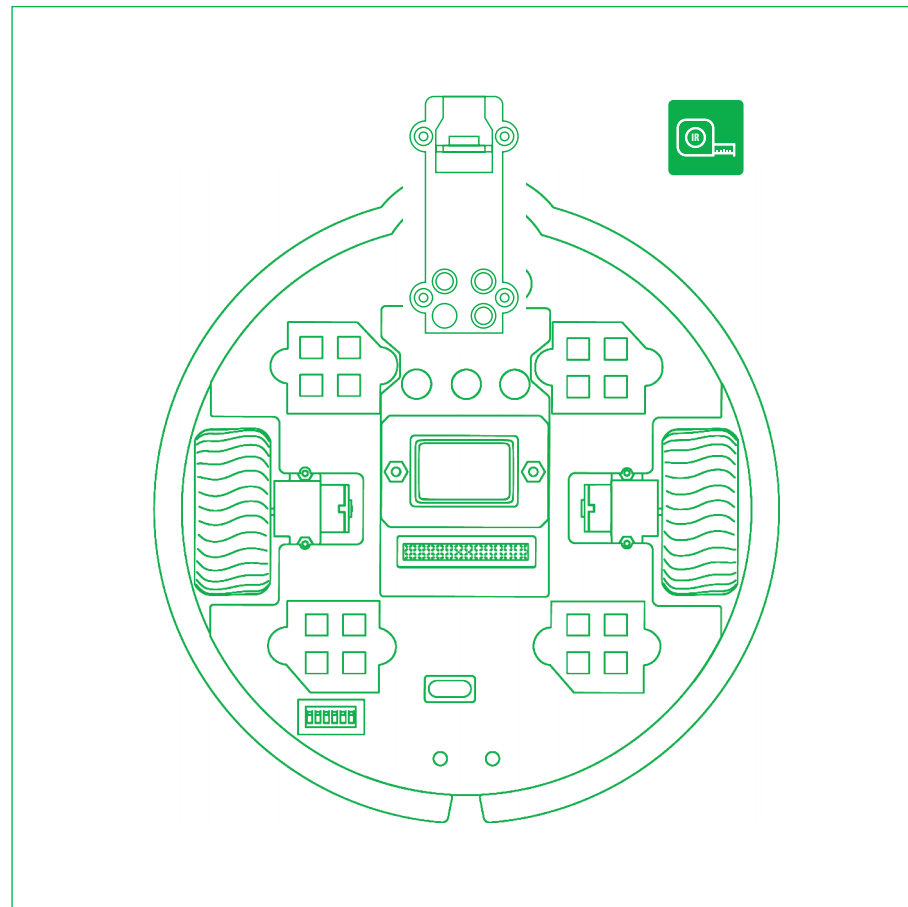
ПОДСКАЗКА

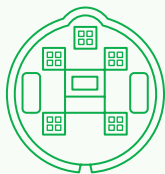
До начала проектирования нарисуйте на бумаге маршрут, по которому Робоплатформа должна пройти (изучите две разные модели на рисунке).



ЗАДАНИЕ

- Подсоедините датчик расстояния к порту №1.
- Отметьте крестиком квадрат **Датчик 1** в программе.
- Помашите рукой перед датчиком и посмотрите изменения показаний датчика. Подумайте, где можно применить полученные результаты.
- Запрограммируйте Робоплатформу таким образом, чтобы она ехала вперед и, заметив впереди препятствие, подала назад, повернулась и продолжила движение в противоположном направлении.





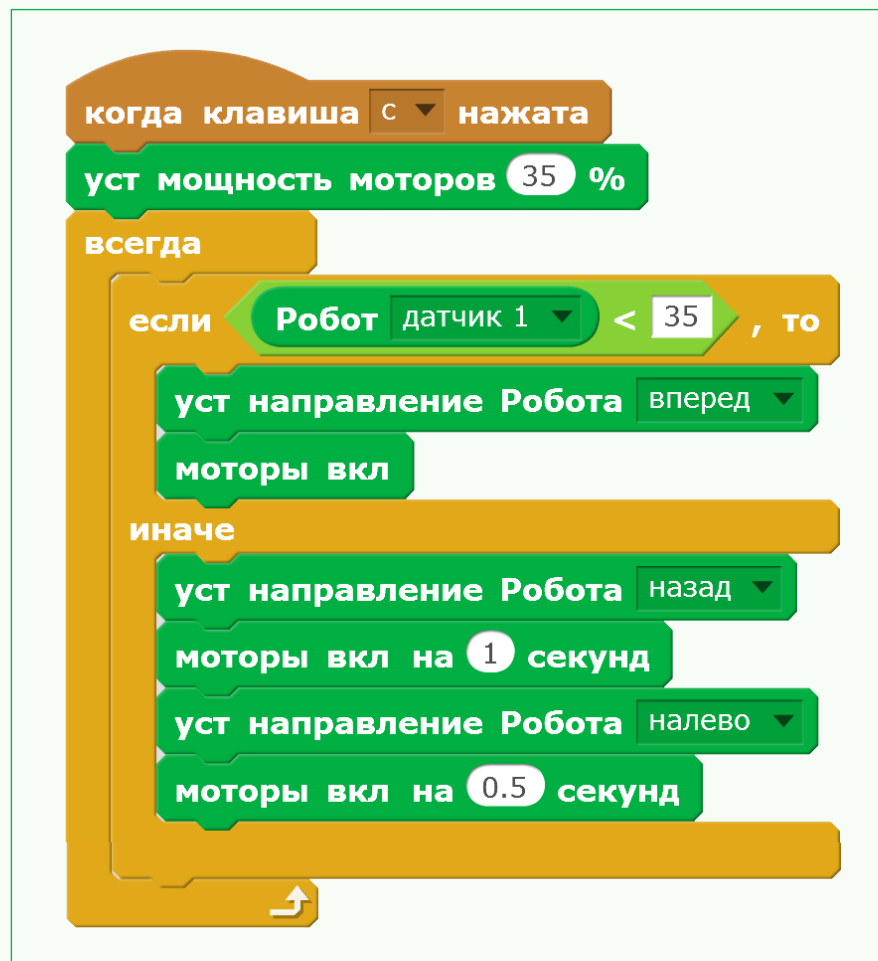
Упражнение 18: Робоплатформа путешествует

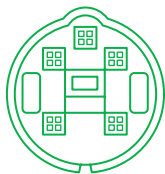
РЕШЕНИЕ

Учитывайте, что показания датчиков могут сильно отличаться в зависимости от освещения.

ЭКСПЕРИМЕНТ!

Попробуйте различные предельные величины датчика. Что произойдет?





Упражнение 19: Применение навыков

ЦЕЛИ

- Использование навыков, полученных на предыдущих упражнениях
- Оценка возможности применения навыков в повседневной жизни

ЗАДАНИЕ 1: ПЫЛЕСОС

Необходимо спроектировать робот-пылесос, который будет убирать небольшие предметы на полу.

- Обозначьте на полу малярной лентой территорию площадью метр на метр.
- Положите туда небольшие предметы.
- Спроектируйте, постройте и прикрепите к Робоплатформе приспособление, которое будет помогать в уборке (например, толкатель и щетка).
- Создайте для Робоплатформы программу, с помощью которой она будет проводить уборку.

ЗАДАНИЕ 2: ПРЕСЛЕДОВАТЕЛЬ

Спроектируйте робот, который будет следовать за движущимся по полу предметом, например, за другим роботом.

ЗАДАНИЕ 3: РАДАР ЗАДНЕГО ХОДА

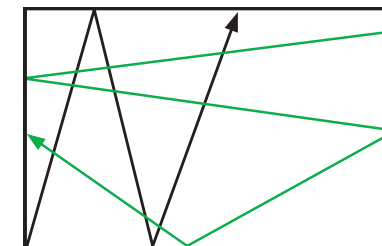
С помощью датчика расстояния и звуков спроектируйте радар заднего хода, с помощью которого будет подаваться звуковой сигнал. Громкость сигнала должна нарастать по мере приближения к объекту.

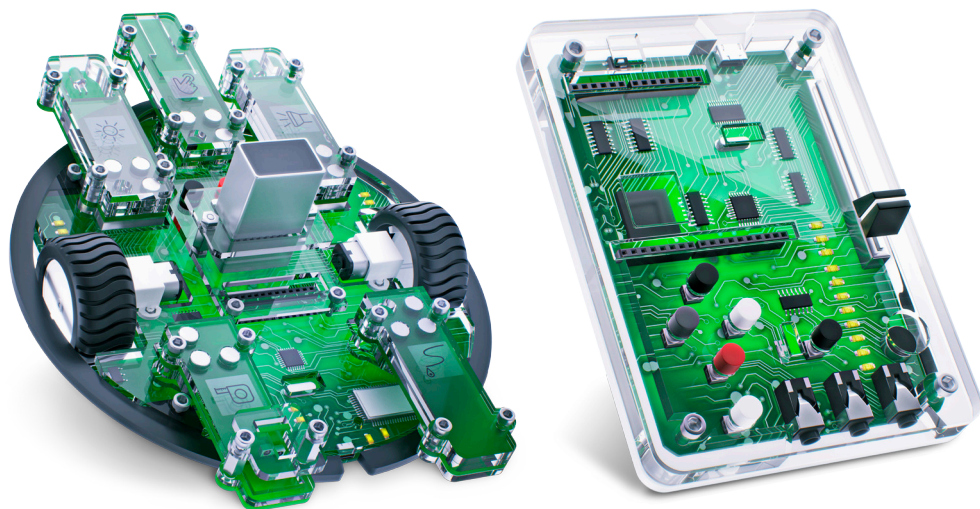
ЗАДАНИЕ 4: ИЗОБРЕТАЙТЕ!

Придумайте задание, в котором можно использовать полученные навыки.

ПОДСКАЗКА

Прежде чем приступить к программированию, начертите на бумаге маршрут, по которому пойдет Робоплатформа (исследуйте две разные схемы, изображенные на рисунке).





**ИСПОЛЬЗУЕМ
РОБОПЛАТФОРМУ
И ЛАБОРАТОРИЮ
СОВМЕСТНО**

Робоплатформой и Лабораторией РОББО можно пользоваться одновременно. Легче всего потренироваться в их совместном использовании, если Лабораторию применить знакомым для детей способом — как элемент управления подвижным устройством. Для выполнения упражнений этой части требуется умение пользоваться и тем, и другим устройством. До выполнения упражнений

УПРАЖНЕНИЯ

УПРАЖНЕНИЕ 1: ЛАБОРАТОРИЯ КАК ЭЛЕМЕНТ УПРАВЛЕНИЯ РОБОПЛАТФОРМОЙ

Учимся использовать Робоплатформу и Лабораторию вместе, применяя ранее полученные навыки.

Важно:

- Совместно обдумать различия и предназначение команд **моторы выкл** и **остановить все**.
- Наблюдать за изменениями значений величин Лаборатории и в программе и использовать наблюдения в разработке программы.

УПРАЖНЕНИЕ 2: РОБОТ-ОХРАННИК

Учимся использовать двигатели Робоплатформы не только для движения, а и для другой работы.

Важно:

- Научить делать наблюдения, что кнопки Лаборатории — это датчики касания. В данном упражнении отпускание кнопки вызывает подачу сигнала.
- Вместе исследовать и подумать, как предметы разного веса влияют на работу.

рекомендуется ознакомиться с инструкциями по эксплуатации и упражнениями по Робоплатформе и Лаборатории.

По структуре упражнения аналогичны другим упражнениям данного руководства.

Они направлены на то, чтобы пользователь увидел и мог установить связь между этими устройствами.

УПРАЖНЕНИЕ 3: СВЕТОФОР

Закрепление изученного в ситуациях, знакомых из повседневной жизни.

Важно:

- Тщательно продумать, в каком порядке происходят разные действия.
- Сделать предварительно план программы на бумаге.

УПРАЖНЕНИЕ 4: ПРИМЕНЕНИЕ НАВЫКОВ НА ПРАКТИКЕ

Закрепите полученные навыки путем проведения различных дополнительных заданий.

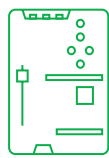
ИСПОЛЬЗУЕМЫЕ КОМАНДЫ

	Команда когда щелкнут по приводит в действие серию команд, прикрепленных внизу, если щелкнуть мышью по флажку.
	Команда когда клавиша_нажата включает программу с помощью выбранной из меню кнопки клавиатуры.
	Команда ждать до_ останавливает выполнение серии команд, пока данное условие не будет реализовано, т.е. пока оно не получит значение истина .
	Команда если<_>, то (условный оператор) выполняет помещенную внутри нее серию команд только в том случае, если значение заданного условия на момент выполнения команды — истина .
	Команда если<_>, то_иначе (разветвленное условие) выполняет одну из двух помещенных в нее серий команд в зависимости от значения условия, заданного для команды. Если условие — истина , исполняется верхняя серия; если ложь , то нижняя.
	Команда всегда повторяет помещенную внутри нее серию команд до прерывания программы, т.е. с точки зрения программы — вечно.
	Команда уст направление Робота устанавливает направление вращения моторов.
	Команда моторы вкл на_секунд включает оба мотора на заданное время (на рисунке — на 1 секунду).

	Условия сравнения используются в командах условных операторов. Они также являются так называемыми логическими выражениями, определяющими выполнение условия (истина) или невыполнение (ложь). Иными словами, если в данных упражнениях условие не выполнено, то серия команд не исполняется.
	Команда установить громкость звука_% устанавливает громкость звука в диапазоне 0–100%.
	Команда доиграть звук_до завершения полностью воспроизводит выбранный из меню звук и только после этого продолжает исполнение программы.
	Условие кнопка на Лаборатории_нажата говорит о том, нажата ли определенная кнопка Лаборатории в запрашиваемый момент. Условие можно поместить, например, в команды условных операторов.
	Функция датчик_ (освещенности/датчик звука/рычажок Лаборатории) показывает вид выбранного датчика на момент выполнения команды.
	Команда играть ноту на Лаборатории_ проигрывает звук определенной высоты из динамиков Лаборатории.
	Команда вкл цветной светодиод на Лаборатории_ включает на Лаборатории светодиод указанного цвета.
	Команда выкл цветной светодиод на Лаборатории_ выключает выбранный светодиод в Лаборатории.

ТЕМЫ ДЛЯ БЕСЕД ВО ВРЕМЯ И ПОСЛЕ УПРАЖНЕНИЙ

- Какие стратегии решений в упражнениях были использованы?
- Как планирование отражается на разработке программы? Планирование — это важный этап в различных ситуациях, связанных с программированием. Чем сложнее задание, тем важнее становится этап планирования.
- Где в повседневной жизни используется автоматика и робототехника?
 - Команды «Включить — выключить» часто используются в быту. Принципом действия является переключатель, которым открывают и закрывают, включают или выключают приборы. Важно также отметить разницу обычного механического переключателя от запрограммированного.
 - Механические переключатели:
 - Выключатель света
 - Кнопка включения прибора
 - Различные игрушки
 - Запрограммированные переключатели:
 - Пульт дистанционного управления
 - Телевизор
 - Сумеречный переключатель
 - Стрелочное управление часто используется в дистанционно управляемых игрушках и в игровых программах. Многие рабочие установки и механизмы (подъемники, погрузчики) имеют приборы со стрелочным управлением.
 - Где в быту используются различные датчики?
 - Радар заднего хода в автомобилях
 - Раздвижные двери и ворота
 - Двери лифтов



Упражнение 1: Лаборатория как элемент управления Робоплатформой

ЦЕЛИ

- Изучение совместной работы Робоплатформы и Лаборатории
- Использование ранее полученных навыков

ЗАДАНИЕ 1

- Кнопки Лаборатории действуют наподобие датчика касания.
- Запрограммируйте Робоплатформу так, чтобы она двигалась вперед по нажатию кнопки 2 и останавливалась по нажатию кнопки 5.

ЗАДАНИЕ 2

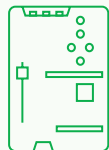
Запрограммируйте Робоплатформу так, чтобы она двигалась вперед и назад и поворачивалась направо и налево по нажатию на цветные кнопки.

ПОДСКАЗКА

Нажмите на кнопку. Исследуйте на дисплее изменения параметров Лаборатории.

кнопка на Лаборатории **1** нажата?



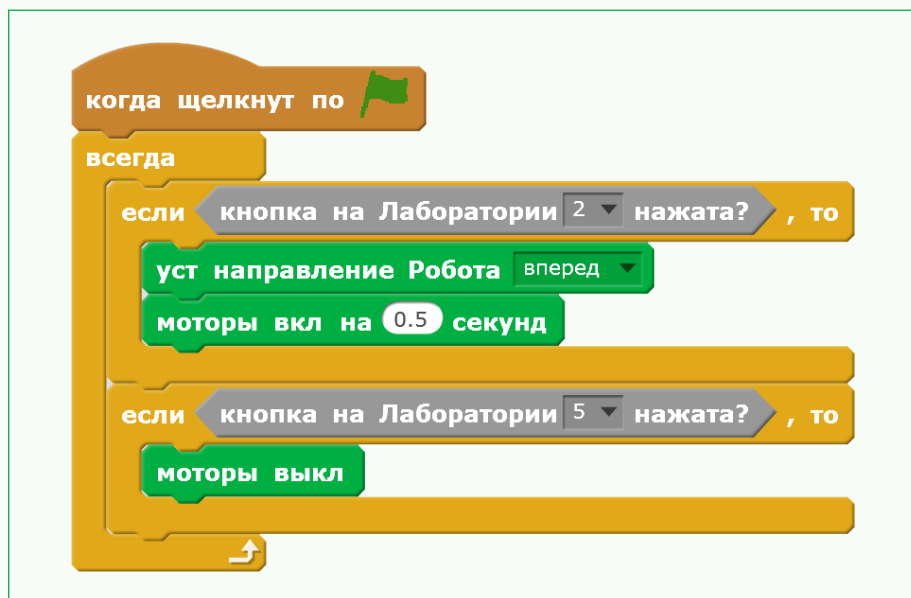


РОББО™

ИСПОЛЬЗУЕМ РОБОПЛАТФОРМУ И ЛАБОРАТОРИЮ СОВМЕСТНО – РЕШЕНИЕ

Упражнение 1: Лаборатория как элемент управления Робоплатформой

РЕШЕНИЕ 1



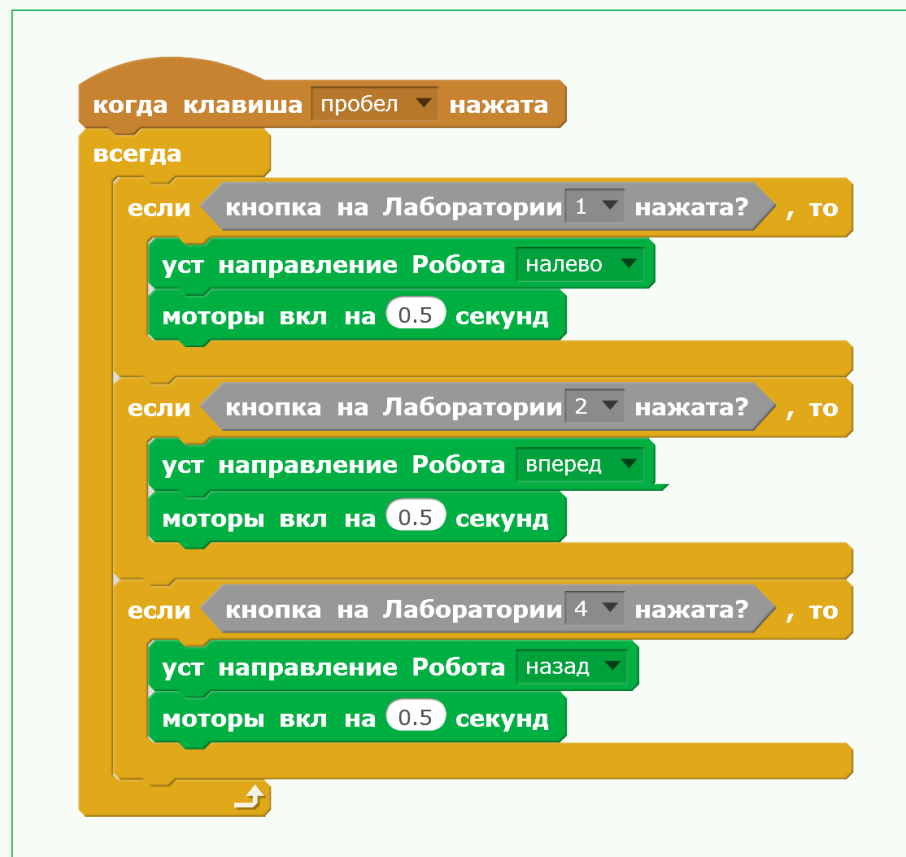
ЭКСПЕРИМЕНТ!

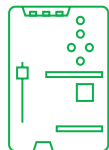
Подключите к программе блокировку рычажка, чтобы предотвратить несанкционированный доступ к элементу управления.

ждать до

кнопка на Лаборатории 5 нажата?

РЕШЕНИЕ 2





Упражнение 2: Робот-охранник

ЦЕЛИ

- Совместное использование Лаборатории и Робоплатформы
- Использование двигателей Робоплатформы для открывания и закрывания шлагбаума

ЗАДАНИЕ 1

- При помощи Лаборатории постройте прибор, который опознает кражу ценного экспоната (например, батарейки LR20) на выставке.
- В случае пропажи экспоната территория выставки закрывается на шлагбаум.

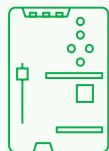
ПОДСКАЗКА

- Поставьте Робоплатформу на какое-нибудь возвышение, чтобы ее колеса оказались повисшими в воздухе. Тогда она не укатится, когда колесо будет поворачивать шлагбаум.
- Экспонат положите на кнопку Лаборатории. Если его взять (т.е. снять с кнопки тяжесть), параметры Лаборатории изменятся.

ЗАДАНИЕ 2

Когда шлагбаум заработает, озвучьте и запрограммируйте также функцию, по которой охранник из диспетчерского пункта (Лаборатории) будет переключать рычажок в положение 100. В этом случае из динамиков выставки будет звучать объявление: «Задержать похитителя!» (выберите для этого звук) до тех пор, пока охранник не переключит рычажок в положение 0.



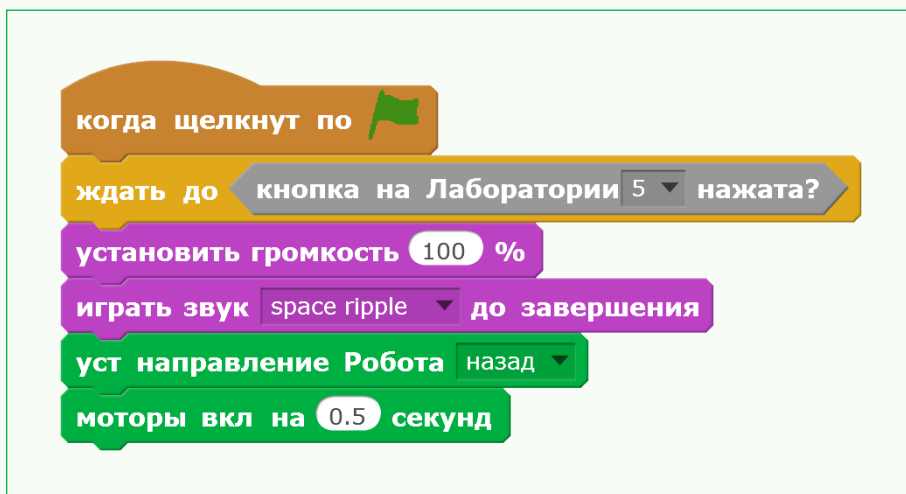


РОББО™

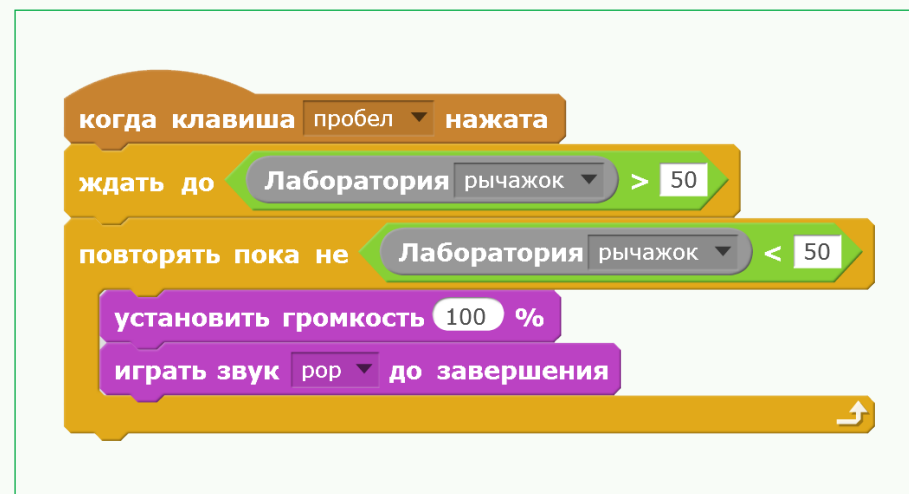
ИСПОЛЬЗУЕМ РОБОПЛАТФОРМУ И ЛАБОРАТОРИЮ СОВМЕСТНО – РЕШЕНИЕ

Упражнение 2: Робот-охранник

РЕШЕНИЕ 1



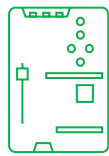
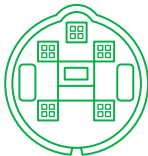
РЕШЕНИЕ 2



ПРОСМОТР

См. видеоролик «Робот-охранник»: <http://bit.ly/robbo1>





Упражнение 3: Светофор

ЦЕЛИ

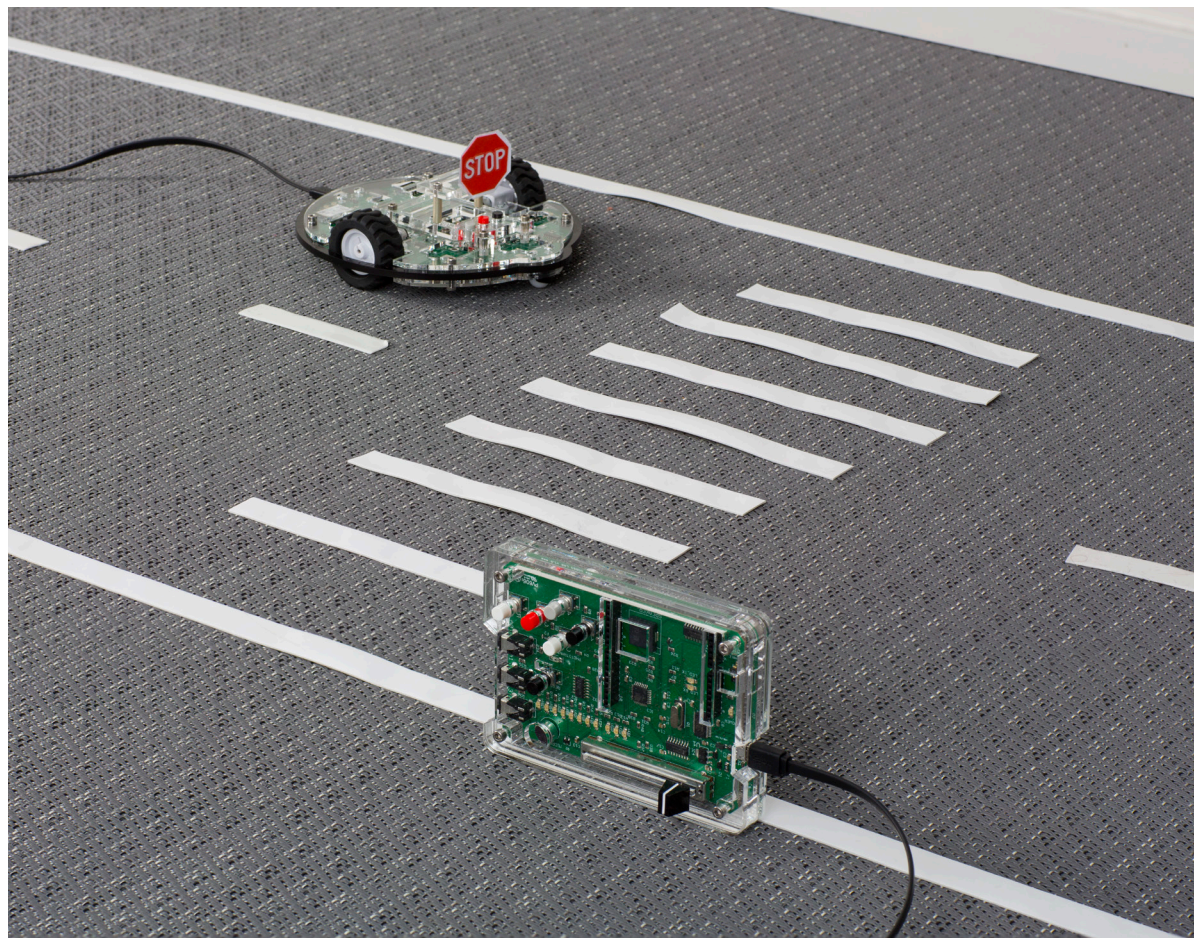
Использование Лаборатории в качестве светофора и выключателя Робоплатформы

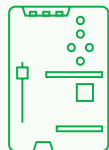
ЗАДАНИЕ

- Постройте светофор для пешеходов и робот, который будет останавливать автомобили.
- В качестве кнопок пешеходов используйте кнопки Лаборатории, а в качестве светофора — светодиоды.
- Сначала подумайте, как включить правильный свет светофора. Потом поразмыслите над принципом действия светофора.

ПОДСКАЗКА

Робоплатформа может останавливать автомобили прикреплённым к ней шлагбаумом или знаком «Стоп».





РОББО™

ИСПОЛЬЗУЕМ РОБОПЛАТФОРМУ И ЛАБОРАТОРИЮ СОВМЕСТНО – РЕШЕНИЕ

Упражнение 3: Светофор

РЕШЕНИЕ

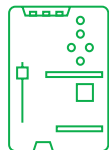


ПРОСМОТР

См. видеоролик «Светофор»:

<http://bit.ly/robbo12>





Упражнение 4: Применение навыков на практике

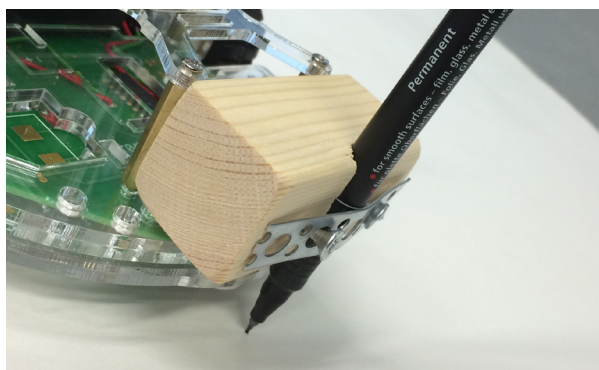
ЦЕЛИ

- Использование полученных навыков
- Использование Робоплатформы для рисования

ЗАДАНИЕ 1: ХУДОЖНИК

Спроектируйте и постройте робот, который под управлением Лаборатории будет рисовать различные узоры.

- Сделайте и прикрепите к Робоплатформе устройство, с помощью которого можно будет чертить маршрут ее движения.
- Каким образом линию, которую чертит Робоплатформа, можно редактировать при помощи кнопок Лаборатории?



ЗАДАНИЕ 2: ИГРУШКА С ДИСТАНЦИОННЫМ УПРАВЛЕНИЕМ

Спроектируйте и сделайте дистанционно управляемую игрушку.

Добавьте в управление функции, в которых скорость будет замедляться и увеличиваться.

ЗАДАНИЕ 3: БАЛЕРИНА

Спроектируйте и сделайте дистанционно управляемую игрушку.

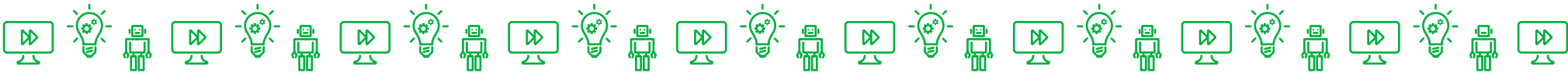
Добавьте в управление функции, в которых скорость будет замедляться и увеличиваться.

ЗАДАНИЕ 4 ИЗОБРЕТАЙТЕ!

Придумайте задание, в котором можно использовать полученные навыки.



ВЫПОЛНЕННЫЕ УПРАЖНЕНИЯ



Имена учеников	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.

ПРИЛОЖЕНИЕ 2

КОМАНДЫ ROBBOSCRATCH, ИСПОЛЬЗУЕМЫЕ В УПРАЖНЕНИЯХ

ДВИЖЕНИЕ

	Команда идти_шагов двигает фигурку на сцене на заданное количество пикселей вперед (положительное значение) или назад (отрицательное значение).
	Команда повернуть на_градусов меняет направление фигурки на заданное значение градусов.
	Команда двигает фигурку в системе координат в заданное место и за заданное время.
	Команда изменить Y на_ изменяет место фигурки по оси Y на заданное количество пикселей.
	Команда изменить X на_ изменяет место фигурки по оси X на заданное количество пикселей.

СОБЫТИЯ

	Стартовая команда, которая приводит в действие серию команд, если щелкнуть мышью по зеленому флажку.
	Стартовая команда, которая приводит в действие серию команд при нажатии клавиши клавиатуры.

ВНЕШНИЙ ВИД

	Команда в течение заданного времени показывает около фигурки «речевую» выноски. Текст выноски (на рисунке — «Hello!») можно редактировать.
--	--

УПРАВЛЕНИЕ

	Команда останавливает выполнение серии команд на заданное время (на рисунке — на 1 секунду).
	Команда (цикл со счетчиком) повторяет вложенную серию команд заданное число раз (на рисунке — 10 раз).
	Команда (бесконечный цикл) повторяет помещенную внутри нее серию команд, пока не будет прервано выполнение всей программы, т.е. с точки зрения программы — вечно.
	Команда (условный оператор) исполняет помещенную внутри нее серию команд только в том случае, если переданное этой команде значение в момент исполнения команды — истина .
	Команда (условный оператор с ветвлением) исполняет одну из двух помещенных внутри нее серий команд в зависимости от заданного для команды значения условия. Если значение — истина , то исполняется верхняя серия команд; если ложь , то исполняется нижняя.
	Команда прерывает исполнение команд до тех пор, пока не реализуется условие, заданное этой команде, т.е. пока оно не получит значение истина .
	Команда останавливает желаемые программы, т.е. скрипты.

ГОЛОС

	Команда начинает проигрывание звука, выбранного в меню, и сразу же продолжает работу программы.
	Команда играть звук_до завершения полностью воспроизводит звук, выбранный в меню, и только потом продолжает работу программы.
	Команда останавливает все звуки программы.
	Команда устанавливает громкость звука по шкале 0–100%.

ОСЯЗАНИЕ

	Условие, которое размещают внутри команды. На момент исполнения команды условие получает значение истинности истина (фигурка касается) или ложь (фигурка не касается).
	Условие, размещаемое внутри команды. Условие получает значение истинности истина (клавиша мыши нажата) или ложь (клавиша мыши не нажата) в момент исполнения команды.

КАРАНДАШ

	Команда опускает карандаш фигурки вниз, так что после подачи команды при движении фигурки на сцене вырисовывается линия.
--	--

ДЕЙСТВИЯ

	Математическое выражение. Можно использовать в командах переменного значения и в сравнительных условиях.
	Выражение производит случайное число из заданного диапазона значений. Выражение можно поместить внутри сравнительного условия.
	Условия сравнения помещают в команды условных операторов. Они также являются т.н. логическими выражениями Boolean, которые определяют исполнение условия (истина) или неисполнение условия (ложь).

ИНФОРМАЦИЯ

	Назначенная переменная. Можно самому назначить имя переменной (на рисунке — Счет). Значение переменной можно в ходе программы менять, а также употреблять, например, в сравнительных условиях.
	Команда задает выбранной переменной новое значение.
	Команда изменяет переменную на заданное значение (на рисунке — +1).

РОБОПЛАТФОРМА

моторы вкл на 1 секунд	Команда включает моторы на установленное время (на рисунке — на 1 секунду).
моторы вкл	Команда включает оба мотора.
моторы выкл	Команда выключает оба мотора.
уст направление Робота вперед	Команда устанавливает направление вращения моторов.
моторы вкл на 10 шагов	Команда включает моторы на время достижения заданного количества шагов (прохождения риска на колесе).
обнулить счётчик пути	Команда устанавливает значение счетчика для измерения количества «шагов» колес на ноль.
повернуть на 15 градусов	Команда поворачивает Робоплатформу на заданное значение градусов.
уст мощность моторов 100 %	Команда устанавливает мощность обоих моторов в диапазоне 0–100%.
уст мощность моторов Л 100 П 100 %	Команда устанавливает отдельно мощность левого и правого моторов в диапазоне 0–100%.
Л вперед П вперед уст мощность Л 100 П 100 %	Команда устанавливает отдельно направления вращения левого и правого моторов и их мощность в диапазоне 0–100%.
Робот датчик 1	Выражение Робот датчик_ возвращает значение выбранного датчика на момент выполнения команды. Возвращаемое значение можно использовать в сравнении, при этом можно сравнивать значение датчика с каким-нибудь числом или значением другого датчика.
кнопка старт на Роботе нажата?	Оператор, которое показывает, нажата ли кнопка Start Робоплатформы в определенный момент времени. Возвращает значение истина или ложь . Условие можно помещать в команды условных выражений.
вкл светодиод на Роботе позиция 1	Команда включает свет (индикатор) светодиодного модуля в выбранном месте Робоплатформы.
выкл светодиод на Роботе позиция 1	Команда выключает свет (индикатор) светодиодного модуля в выбранном месте Робоплатформы.

ЛАБОРАТОРИЯ

вкл светодиод на Лаборатории 0	Команда включает выбранный светодиод Лаборатории.
выкл светодиод на Лаборатории 0	Команда выключает выбранный светодиод Лаборатории.
вкл цветной светодиод на Лаборатории красный	Команда включает выбранный (красный/желтый/зеленый) светодиод Лаборатории.
выкл цветной светодиод на Лаборатории красный	Команда выключает выбранный (красный/желтый/зеленый) светодиод Лаборатории.
играть ноту на Лаборатории 48	Команда включает звук определенной высоты из динамиков Лаборатории.
Лаборатория свет	Команда задает значение выбранному датчику Лаборатории (датчик освещенности/ звука/ рычажок). Значение можно использовать, например, в сравнительных условиях.
кнопка на Лаборатории 1 нажата?	Оператор, который говорит о том, нажата ли определенная кнопка Лаборатории в запрашиваемый момент. Условие можно поместить, например, в команды условных операторов. Возвращаемые значения — истина/ложь .
Лаборатория вход A1	Аналоговый вход, возвращает получаемое числовое значение.
Лаборатория вход D8	Цифровой вход/выход. Возвращаемое значение — истина/ложь .
уст Лаборатория выход D2 в включить	Команда устанавливает выбранный цифровой вход Лаборатории в положение включено/выключено (истина/ложь = 0/5 В).
уст Лаборатория выход D3 в ШИМ значение 0	Команда устанавливает значение широтно-импульсной модуляции выбранного цифрового входа Лаборатории в диапазоне 0–255.

ПРИЛОЖЕНИЕ 3
ТЕРМИНЫ ПРОГРАММИРОВАНИЯ

СЛОВАРЬ ПО ПРОГРАММИРОВАНИЮ

Настоящий словарь прилагается к руководству для преподавателей.
Пояснения по значению терминов рекомендуется давать во время выполнения упражнений.

Термин на финском языке на русском языке	Термин на английском языке	Пояснение	Ссылка
Boolean lauseke Логическое выражение (булево выражение)	Boolean expression	Выражение, возвращаемым значением которого является значение истинности истина/ложь (по-английски — true/false).	Робоплатформа, п. 4 / Датчик касания
Ehto Условие	Condition	В условных операторах используются условия, исполнение которых определяет, какая следующая серия команд будет реализована. Условия являются логическими выражениями, например: «1 < 2». Когда условие оценивается, оно получает значение истинности истина/ложь.	Робоплатформа, п. 4 / Датчик касания
Ehtolause Условный оператор	Conditional statement	Оператор, в котором исполнение программы разветвляется в зависимости от того, будет ли реализовано определенное условие.	Робоплатформа, п. 4 / Датчик касания
Käsäy Команда	Command	В программировании под командой подразумевают указание компьютеру выполнить определенное отдельное задание. Например: S = A + B, т.е. суммируйте два числа A и B и сохраните результат в переменной S.	

Komento Команда	Command	См. команда.	
Komentosarja Серия команд	Sequence	Сформированная серия, в которой заданные команды исполняются подряд.	Робоплатформа, п. 2 / Управляем Робоплатформой с помощью кнопок
Lause Предложение	Statement	См. команда.	
Lauseke Выражение	Expression	Программная структура, которая возвращает определенное значение. Например, значение переменной, значение, считываемое с датчика, или стандартное значение, такое как число 1. Выражения можно соединять знаками операций, такими как плюс (+). Например, 1 + 2 — это выражение, значение которого составляет 3.	Робоплатформа, п. 4 / Датчик касания
Ohjelma Программа	Program	Компьютерная программа (сокращенно, программа) — самостоятельный исполняемый в компьютере набор команд. Короткие программы, содержащие только несколько команд, можно также называть сериями команд или скриптами.	
Sensori Датчик	Sensor	Датчик, прибор, измеряющий значение какой-либо реальной величины, например, освещенности, температуры или расстояния. В программировании параметры или значения датчиков часто можно использовать в выражениях. При этом можно формировать, например, такие сравнения: «расстояние < 50», которые, в свою очередь, можно использовать в условных предложениях.	Робоплатформа, п. 4 / Датчик касания

Skripti Скрипт	Script	См. Серия команд.	
Toistorakenne Цикл (структура повторения)	Loop	Цикл — это управляющая конструкция, повторяющая определенную серию команд вечно (бесконечный цикл), N раз (цикл со счетчиком) либо до исполнения какого-либо условия (цикл с условием).	Робоплатформа, п. 3 / Самостоятельное движение Робоплатформы
Vertailuehto Условное выражение	Condition, Relational expression	Используемое в условном операторе условие, в котором сравниваются между собой два числа. Например, логическое выражение «1 < 2» можно использовать в условии сравнения.	Робоплатформа, п. 4 / Датчик касания
Vertailuoperaattori Оператор сравнения	Relational operator	Операторы сравнения <, >, = можно применять для формирования логического выражения из двух числовых выражений. Например, «1 < 2» — это логическое выражение, всегда имеющее значение «истина».	Робоплатформа, п. 4 / Датчик касания

