

МАЛЕНЬКИЙ, НО ОЧЕНЬ УМНЫЙ

CODEY ROCKY

makeblock

НЕДЕТСКИЙ
КОЗД



DIGIS

Рекомендации

для преподавателей

72 часа

Учебных занятий

ПРЕДИСЛОВИЕ

Набор Codey Rocky предназначен для изучения основ программирования, принципов работы электроники, интернета-вещей (IoT), искусственного интеллекта (AI) в детском саду и школе. Он подходит как для использования на уроках, так и при организации работы кружка и реализации проектной деятельности.

Оснащенный различными типами датчиков и программируемым «мозгом» Codey может работать самостоятельно. Это идеальный партнер для обучения программированию для начинающих. Rocky – это подвижное тело Codey, вместе они становятся лучшими товарищами для игр. Codey Rocky может выполнять все виды миссий, таких как избегание препятствий, определение цвета и следование по линии.

Codey Rocky легко собрать, позволяя ученикам сосредоточиться на превращении своих идей в реальность с помощью кодирования.



Для программирования используется среда mBlock. Кроме изучения основных алгоритмических конструкций, среда поддерживает работу со всеми роботами компании Makeblock. Вы можете подключаться к облачным сервисам и создавать проекты в области искусственного интеллекта (AI) и машинного обучения: распознавание изображений, речи, эмоций, возраста человека и др.

Встроенный модуль Wi-Fi позволяет Codey Rocky осуществлять сбор данных, например, о погоде для приложений интернета-вещей (IoT).

В нашем курсе мы будем рассматривать методику преподавания программирования на основе образовательного робототехнического набора Makeblock Codey Rocky Education Pack - 6 Robots. Содержит 6 программируемых роботов Codey Rocky мультимедийное устройство, 6 наборов карточек и лент для ношения Codey на шее. Это практичное решение для класса



НАШ ПОДХОД

Мы работаем в парах

Парное программирование — это процесс разработки программного обеспечения, в котором два программиста работают вместе. Один отвечает за запуск программы, а другой наблюдает за тем, как она работает. Доказано, что парное программирование помогает студентам учиться и сотрудничать друг с другом.

Обучение через игру

Дидактические игры по кодированию позволяют студентам изучать программирование с помощью игр. Игры живого действия – это учебная деятельность, при которой цветные ручки, карточки, учебники, другие материалы и даже сами ученики используются для изучения принципов программирования и алгоритмики. Даже без компьютера учителя все равно могут преподавать основы информатики с помощью дидактических игр, чтобы лучше понять принципы, заложенные в вычислительных машинах.

Ни дня без практики

Практическое задание должно быть на каждом занятии. Аксиома: для того, чтобы научиться программировать, необходимо писать код. А для того, чтобы писать код было интересно, оставляйте свободу выбора в рамках задачи.

СТРУКТУРА УРОКА



МАТЕРИАЛЫ ДЛЯ ПРЕПОДАВАТЕЛЯ

Рекомендации для проведения занятий

Советы по проведению занятий по рабочей программе на 72 часа.

Раздаточные материалы

Презентации, рабочие листы и формы обратной связи, которые используются при проведении занятий

Идеи для проектов

Банк идей для проектов с Codey Rocky с описанием сути проекта, поясняющим видео и программным кодом

ПРОГРАММА КУРСА

Курс разделен на 6 модулей, по 12 занятий. В каждом модуле есть элементы теории и серия дополнительных заданий на отработку полученных знаний.

Старт курса

Базовый курс (часть 1)

- события, условия, циклы, сенсоры

Базовый курс (часть 2)

- функции, переменные

Гейм-дизайн

- принципы создания игр, разработка контроллеров

Облачные сервисы (AI)

- распознавание изображений, речи, эмоций

Взаимодействие (IoT)

- связь Codey Rocky между собой, интернет-вещей

Творческие проекты

- кросс-предметные проекты

ЗНАКОМСТВО

В этом уроке мы знакомимся с Codey Rocky — крошечным, но универсальным роботом. Учитель может продемонстрировать особенности Codey Rocky через видео или загрузить заранее подготовленные программы в Codey Rocky. Например, это может быть избегание препятствий, игра «Камень, ножницы, бумага» и распознавание эмоций.



ЦЕЛЬ УРОКА:

1. Представить возможности Codey Rocky и замотивировать на обучение
2. Дать возможность поуправлять роботом и понять принципы его программирования
3. Ввести понятие «Программа»

Спросите студентов:

1. Встречали ли вы роботов в повседневной жизни?
2. Как они выполняли задачи?
3. Попробуйте догадаться, для чего используются эти роботы

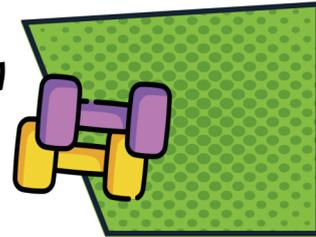


4. Как вы думаете, каким образом роботы могут понять наши инструкции? Дайте ученикам некоторое время для обсуждения.

Подведение итогов:

В процесс обсуждения необходимо прийти к выводу, что все роботы создаются для какой-то конкретной задачи, особенности их конструктива, используемые материалы, все это обусловлено основной функцией робота.

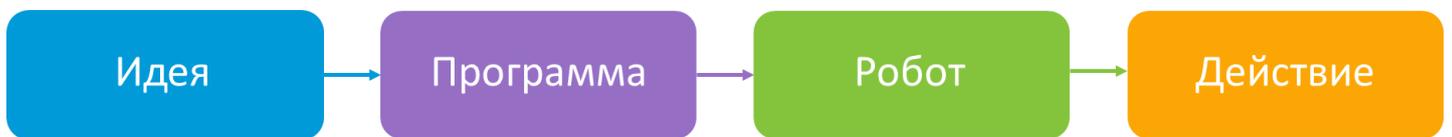
РАЗМИНКА



NEW

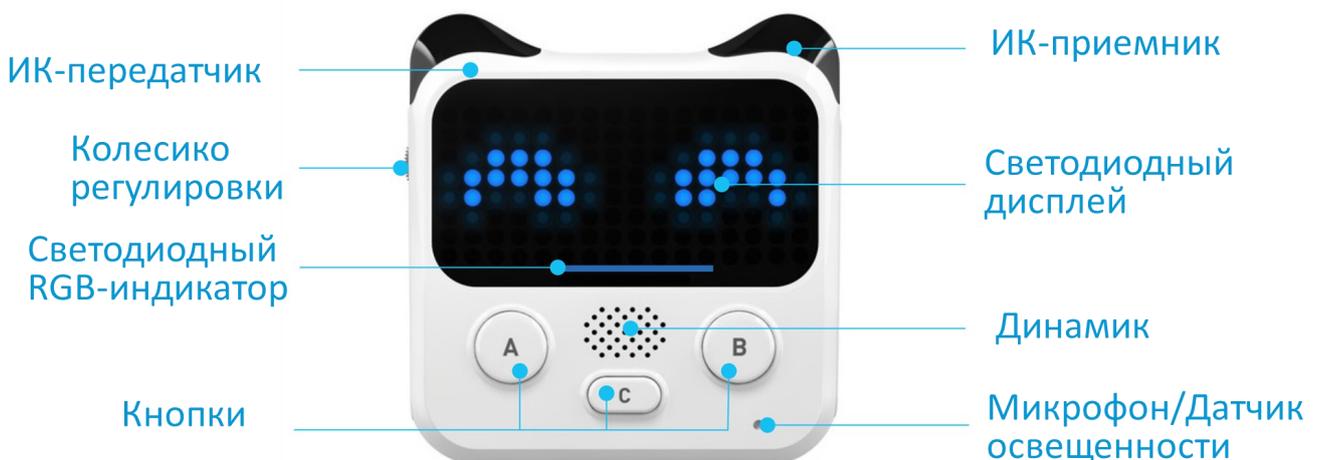
ПОНЯТИЕ «ПРОГРАММА»

Программа — это искусственный язык, который мы используем, чтобы сказать роботу, что делать. Мы переводим наши инструкции в программный код. Затем мы загружаем программу в робота, заставляя его действовать, выполняя указанные в программе команды.



Таким образом, робот выполняет лишь те команды, которые прописаны для него человеком и которые предусмотрены его конструкцией. Например, мы не можем попросить Codey Rocky заварить нам чай, как бы подробно не были описаны наши инструкции. Зато мы можем попросить его переключить канал телевизора. Поэтому для того, чтобы понять на что способен Codey Rocky, необходимо знать как он устроен.

Codey, как мозг робота, оснащен различными датчиками и программируемыми блоками. Он может работать индивидуально, а также может работать с Rocky для выполнения большего количества задач. Давайте взглянем на то, какие датчики у него есть.



Rocky обеспечивает подвижность Codey и оснащен поворотным блоком ИК-датчиков. Он добавляет больше возможностей для Codey, позволяя ему перемещаться, избегать препятствий, определять цвета, следовать по линии и многое другое.

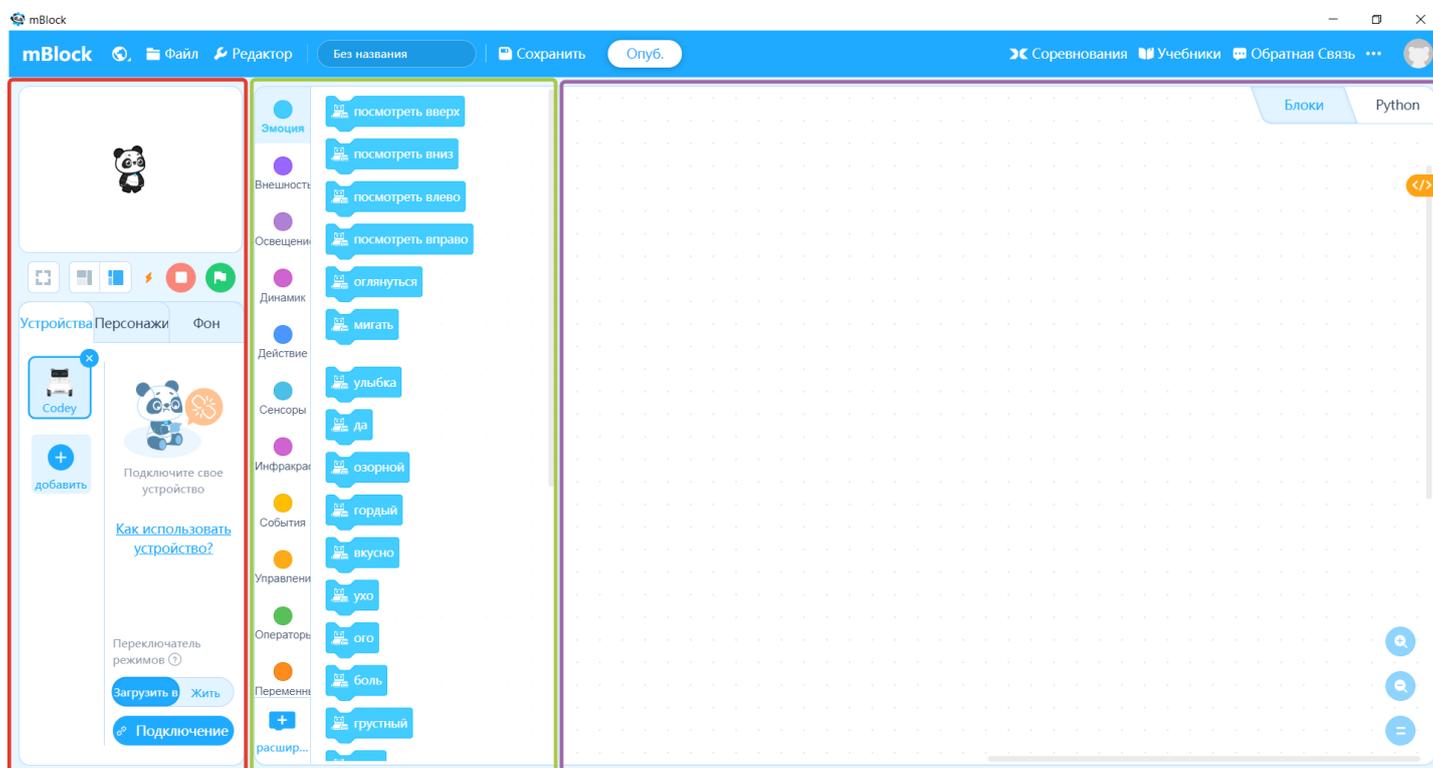
УПРАВЛЕНИЕ РОБОТОМ



mBlock 5 — это среда программирования, которая поддерживает программирование с помощью графических блоков и написание кода на высокоуровневом языке Python. Он разработан на основе Scratch 3.0, программного инструмента с открытым исходным кодом, который разработан совместно MIT и Google. Скачать актуальную версию программы можно [здесь](#).

Используя mBlock 5, вы можете писать программы, как для Codey Rocky, так и для других роботов компании Makeblock. Вы даже можете использовать программное обеспечение для создания историй, игр и анимации. Кроме того, mBlock 5 предоставляет детям такие технологии, как ИИ, машинное обучение и интернет-вещей. Одним словом, mBlock 5 может быть одним из лучших вариантов для начинающих программистов.

ЗНАКОМСТВО С ИНТЕРФЕЙСОМ



Сцена и спрайты

Здесь добавляются устройства, персонажи, сцены и загружаются программы

Палитра блоков

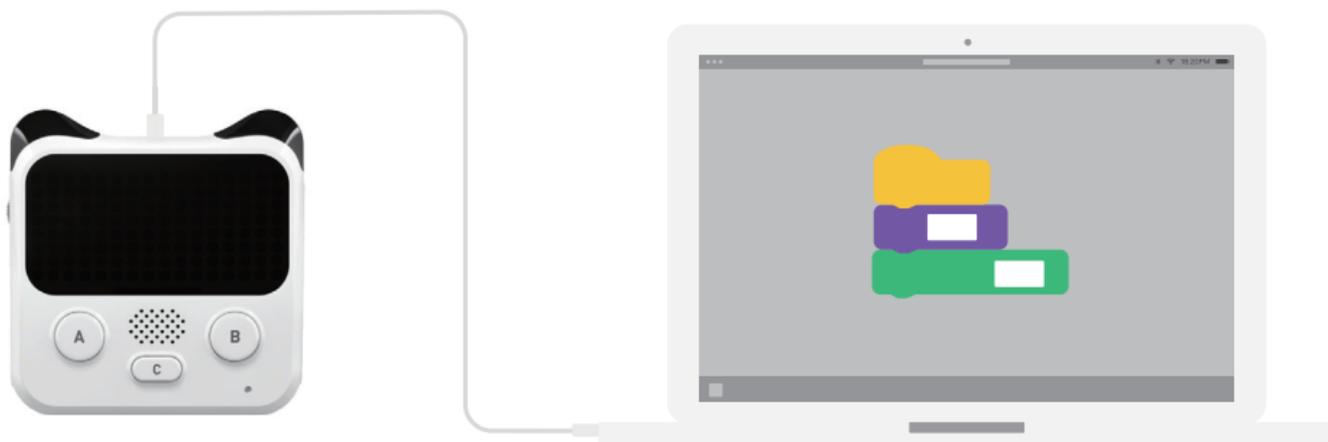
Здесь расположены все блоки, используемые для программирования. Для удобства они сгруппированы по категориям

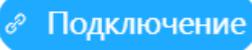
Область кода

В эту область перетаскиваются блоки программы из палитры

ЗАГРУЗКА ПРОГРАММЫ

(1) Подключите Codey Rocky к вашему компьютеру с помощью USB-кабеля и включите

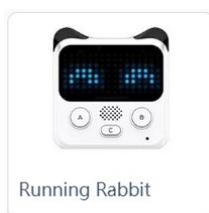


(2) Нажмите на кнопку 

(3) Выберите серийный порт для подключения и нажмите кнопку «Подключение»

Для того, чтобы потренироваться в загрузке программ, можно использовать примеры программ.

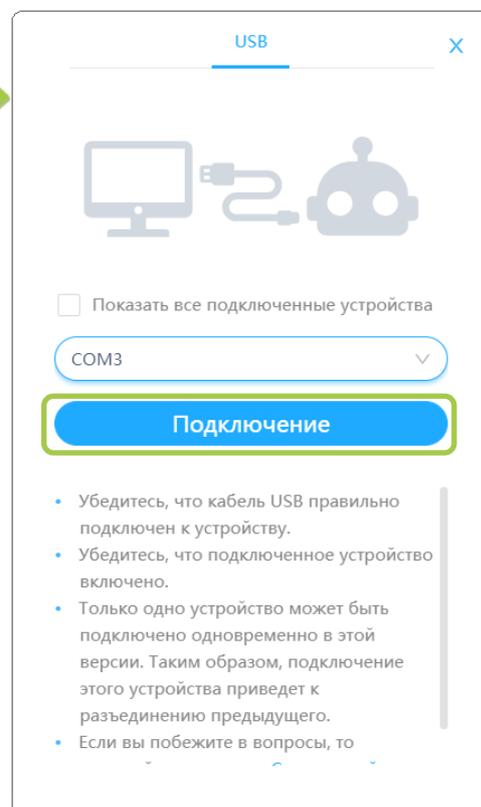
Нажмите на кнопку  и выберите пункт «Примеры программ»



Откройте одну из программ, например, **Running Rabbit**. И нажмите 

На экране Codey Rocky должен пробежать зайчик.

Вы можете сохранять свои программы на компьютере, для этого воспользуйтесь пунктом меню **Файл** — **Сохранить на ваш компьютер**. Или хранить все свои скрипты в облаке, получая доступ к ним с любого устройства из любого места. Для этого необходимо завести аккаунт в Makeblock и выполнить в него вход. Вы можете использовать для этого свой Google-аккаунт.



УЧИМСЯ УПРАВЛЯТЬ

Для начала работы с Codey Rocky полезно дать учащимся возможность поиграться с роботом, поуправлять им, понять как он двигается, меняет внешность и издает звуки. Для этого отлично подходит приложение для смартфона Makeblock app.



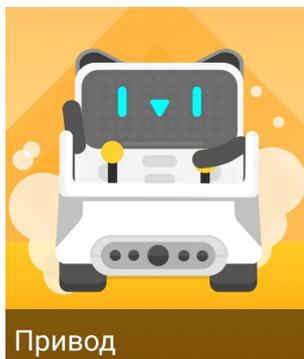
Приложение Makeblock

Предоставляет управление любым роботом от компании Makeblock с возможностью настройки контроллера и написания программ. Для старших конструкторов есть также 3D-инструкции по сборке.

Android

iOS

Приложение работает с Codey Rocky в трех режимах:



Привод

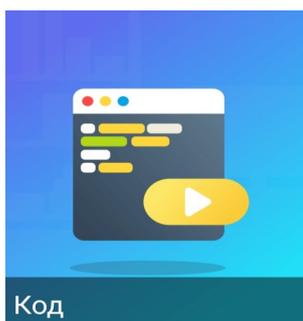
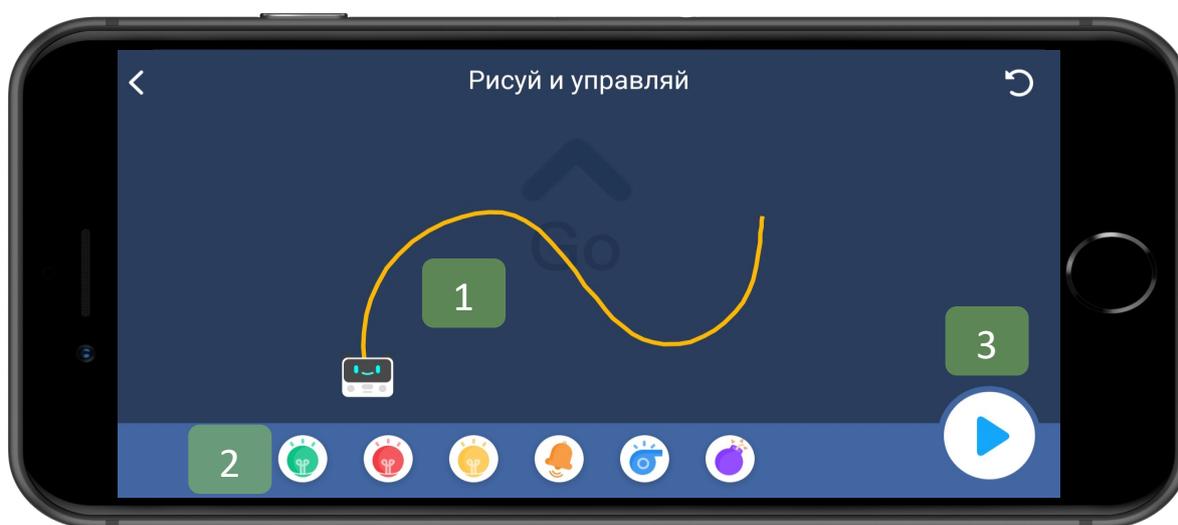
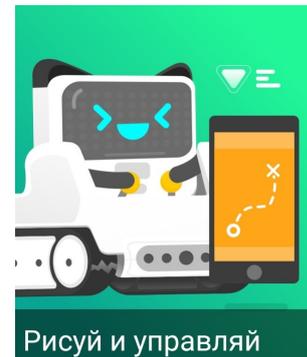
Контроллер **Всё-в-одном**

1. Управление движением Codey Rocky.
2. Изменение изображения на экране
3. Предустановленные действия (бежать, крутиться на месте и др.)
4. Проявление эмоций со звуками и движением



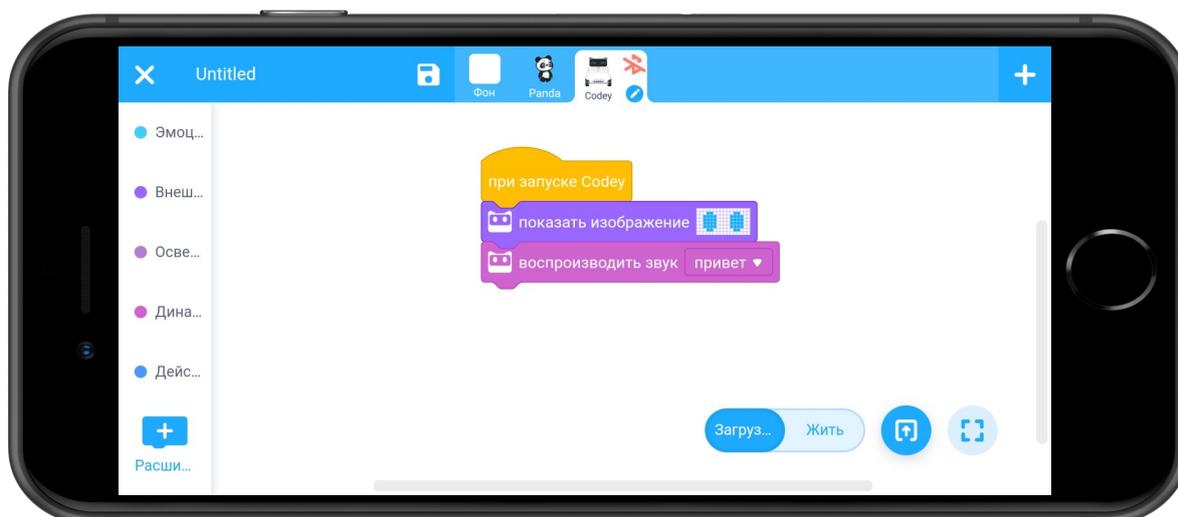
Навигатор **Следи по линии**

1. Нарисуйте маршрут
2. Добавьте точки на маршруте с дополнительными действиями (мигать светодиодом, издавать звуки)
3. Нажмите на кнопку **«Старт»**, чтобы проследовать по маршруту



Среда разработки

Для работы в этом режиме потребуется установка дополнительного приложения mBlock. Оно работает в режиме обучения (прохождение квеста) и кодирования. По своему интерфейсу соответствует версии ПО для компьютера, кроме того синхронизируется с ним, если выполнен вход в личный профиль



1010101001111011



ПЕРВАЯ ТРЕНИРОВКА

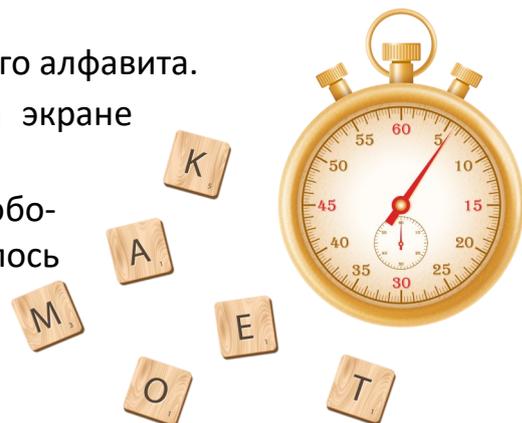
Следующие задания носят игровой характер, их основная задача, научиться управлять и взаимодействовать с Codey Rocky. В зависимости от количества роботов в классе может меняться частота смены оператора.

Задание 1. Лабиринт

На полу малярным скотчем рисуется лабиринт или выкладывается карандашами/фломастерами. Команды по очереди проходят лабиринт. По звуковому сигналу ведущего - меняется оператор Codey Rocky. Побеждает так команда, которая потратила на прохождение лабиринта меньше всего времени.

Задание 2. Эрудит, спортсмен и просто красавец

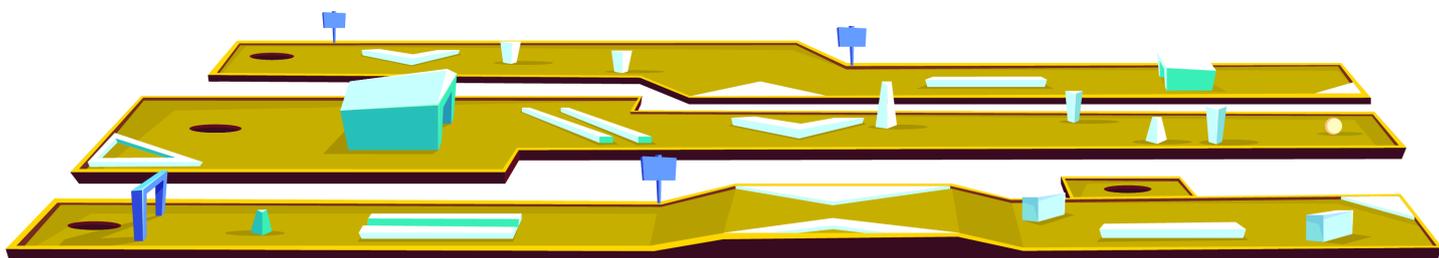
- Каждому оператору дается карточка с буквой русского алфавита.
- Отводится 1 минута, чтобы вывести эту букву на экране Codey Rocky.
- Необходимо как можно быстрее выстроить всех роботов в нужной последовательности, чтобы получилось правильное слово.
- После того, как слово составлено, все Codey Rocky выполняют победное движение



ТВОРЧЕСКОЕ ЗАДАНИЕ

Вместе с учениками создайте полосу препятствий для Codey Rocky из подручных средств.

Вы можете придумать различные соревновательных механики, разработать систему начисления очков и т.д. Проведите это время весело вместе с Codey Rocky, у нас впереди еще много свершений!



ОБРАБОТКА СОБЫТИЙ

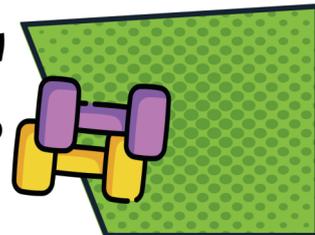
В этом занятии мы вводим новое понятие «Событие». Дело в том, что визуальная среда программирования является событийно-ориентированной, т.е. выполнение программного кода происходит только при возникновении какого-то события, например, работа включили, потрясли, изменилась громкость и т.д.



ЦЕЛЬ УРОКА:

1. Ввести понятие **Событие**
2. Научиться обрабатывать события «**Нажатие на кнопку**»
3. Научиться менять изображение на экране Codey Rocky

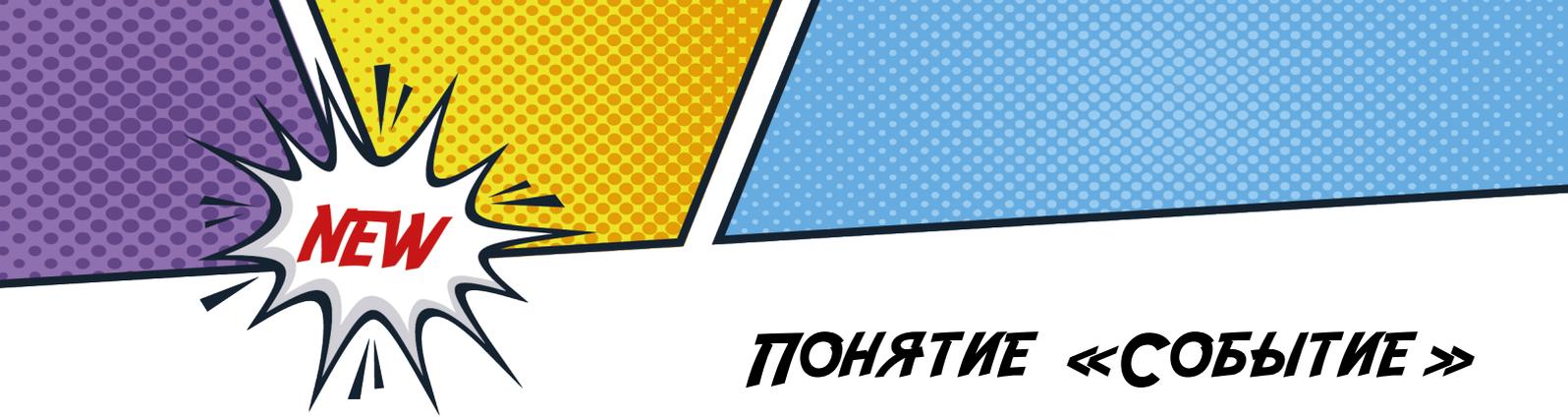
РАЗМИНКА



Предложите ученикам поиграть в игру, чтобы лучше понять концепцию «Событие».

1. Нарисуйте фигуры на доске (или используйте презентацию), например, прямоугольник, треугольник, круг.
2. Разделите учеников на три группы и проговорите правила игры, определив 4 события, например:
 - Когда вы кладете руку на треугольник - 1-ая группа хлопает;
 - Когда вы кладете руку на круг – 2-я группа топает;
 - Когда вы кладете руку на квадрат – 3-ая студентов прыгает;
 - Когда вы кладете руку на звезду – все кричат «Урааа!».
3. Перемещайте руку на фигуры случайным образом и проверяйте правильность реакции студентов. Если студенты реагируют так, как вы ожидаете, то перемещайте руку на другую форму. Если студенты не реагируют неверно, необходимо повторить правила игры снова.
4. Повторите несколько раз, увеличивая скорость переключения между фигурами.

Подведите итог. В этом случае, рука на фигуре служит событием. Когда рука указывает на одну из фигур, исполнитель — группа студентов выполняет инструкции.



NEW

ПОНЯТИЕ «СОБЫТИЕ»

Событие в mBlock—это некое «внешнее» воздействие на персонаж или робота. Так например, мы можем задавать событие на включение, получение сообщения, нажатие на кнопку и т.д.

Мы вводим это понятие на простых примерах из обычной жизни: когда стемнеет и мы входим в комнату, нам нужно включить свет. Чтобы включить свет, нам нужно нажать кнопку включения света. В этом случае нажатие кнопки является событием, и что свет включен является результатом.

Перед началом работы полезно напомнить учащимся основные элементы интерфейса программы. Для программирования в mBlock 5 используются блоки, которые добавляются на область скриптов методом перетаскивания (drag-and-drop). Блоки различаются по цвету и форме. Цвет помогает сориентироваться какую функцию выполняет блок. Форма определяет в каком месте программы можно использовать блок.

Блоки событий имеют желтый цвет и уникальную форму, благодаря которой мы не можем поставить ни один другой блок перед ними.

при запуске Codey

когда Codey вибрирует

при нажатии кнопки A ▼

Таким образом, **событие является началом части программы**. Когда вы пишете программы, первое, что вам нужно сделать, это выбрать событие, при котором должен выполняться блок программы. Например, мы хотим, чтобы при включении Codey Rocky, показал открытые глаза и сказал «Hello!».

Продемонстрируйте ученикам как собирать программу.

Спросите:

1. Какой блок событий надо выбрать?
2. Как вы думаете в какой категории надо искать блок отвечающий за изображение на экране? А где искать звуковые команды?

при запуске Codey

показать изображение 

воспроизводить звук привет ▼

Загрузите программу и покажите результат выполнения.

1010101001111011



УПРАВЛЕНИЕ ЭМОЦИЯМИ

Задание 1. Напишите программу, чтобы Codey Rocky реагировал на нажатие кнопок, так как показано на рисунке:



Задание 2. Добавьте к изменению внешнего вида звуковое сопровождение, соответствующее показанной эмоции.

Проверка выполнения задания (звуки могут быть другими)

при нажатии кнопки **A** ▾
 показать изображение [eyes]

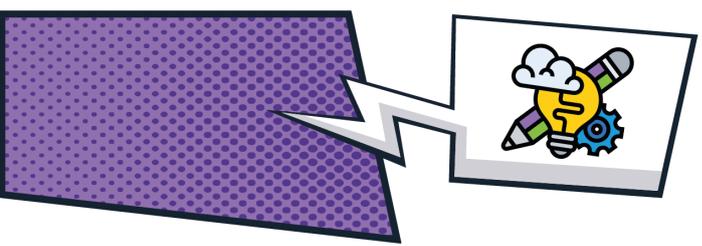
воспроизводить звук **ого** ▾

при нажатии кнопки **B** ▾
 показать изображение [robot]

воспроизводить звук **смех** ▾

при нажатии кнопки **C** ▾
 показать изображение [eyes]

воспроизводить звук **грустный** ▾



ТВОРЧЕСКОЕ ЗАДАНИЕ

Вспомните, какие эмоции вы испытывали сегодня в течение дня.

Покажите как бы реагировал Codey Rocky попав в такую же ситуацию. Поэкспериментируйте, используя блоки движения.

ПОКАДРОВАЯ АНИМАЦИЯ

В этом уроке мы знакомимся с понятием линейного алгоритма, через пример покадровой анимации. Это отличный пример для демонстрации того, что действия выполняются последовательно один за одним, но так как они выполняются очень быстро появляется движение/действие. Робот работает аналогичным образом последовательно выполняя заданные для него команды.

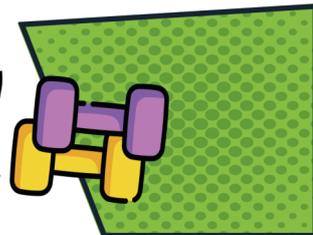


ЦЕЛЬ УРОКА:

1. Ввести понятие **Линейный алгоритм**
2. Изучить принцип покадровой анимации
3. Научиться давать последовательные команды

РАЗМИНКА

Игра «Я—РОБОТ»



Задание: Вы выполняете роль робота. Подойдите от того места, где вы находитесь, к доске и нарисуйте смайлик на ней. Команды для «робота» готовят ученики самостоятельно. Задача преподавателя **строго следовать инструкциям.**

Проведение игры

1. Объясните правила игры. Строго обозначьте место старта
2. Предложите студентам дать инструкции устно или записать на бумаге. Напомните, что инструкции должны быть организованы в правильном порядке (сверху-вниз).
3. Вы должны слушать/читать инструкции студентов и выполнять задачи в строгом соответствии с командой.

ВАЖНО

Инструкции должны быть достаточно подробными и конкретными. Например, инструкция подойдите к доске, не отвечает этим требованиям. Совсем другое дело: поверните налево на 90 градусов и двигайтесь вперед на 4 метра. Когда нет конкретной настройки для времени и угла, робот будет выполнять команды непредсказуемо и не получится достичь нужного результата.

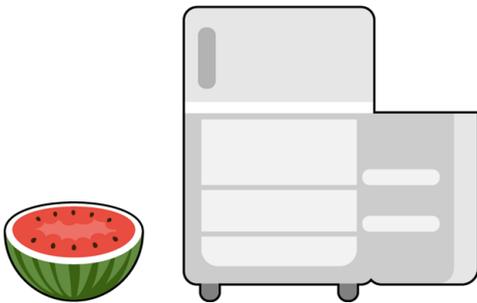
NEW

ЛИНЕЙНЫЙ АЛГОРИТМ

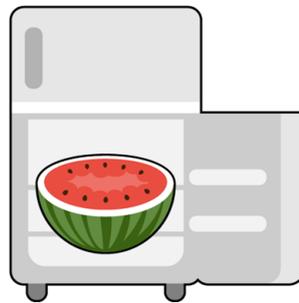
В жизни мы часто сталкиваемся с тем, что для того чтобы достичь какого-либо результата необходимо выполнить несколько шагов. Чаще всего мы даже не замечаем, что каждое действие состоит из последовательности действий. Если нам нужно почистить зубы перед сном, мы автоматически идем в ванну, берем щетку, открываем тюбик зубной пасты и начинаем чистить зубы.

Рассмотрим еще один пример: Какие шаги, чтобы положить арбуз в холодильник? Ученики могут дать вам разные варианты ответов, но в результате вы получите примерно следующий алгоритм: открыть холодильник, положить арбуз в холодильник, закрыть дверь холодильника.

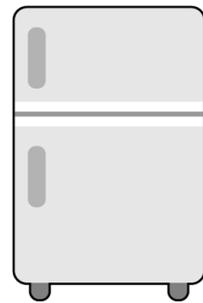
Шаг 1. Открыть холодильник



Шаг 2. Положить арбуз в холодильник



Шаг 3. Закрыть холодильник



Спросите учеников: «Можете ли вы вспомнить случаи, когда вам нужно следовать набору шагов для достижения чего-то?». Дайте некоторое время для ответов, обсудите каждый пример.

Для робота определены только некоторые команды которые он может делать автоматически, остальные нужно задавать программисту. Когда мы программируем, мы располагаем блоки сверху вниз, чтобы задать последовательность шагов.

открыть холодильник

положить арбуз в холодильник

закрыть холодильник

Таким образом, **линейный алгоритм** - это алгоритм, в котором все операции выполняются последовательно одна за другой.

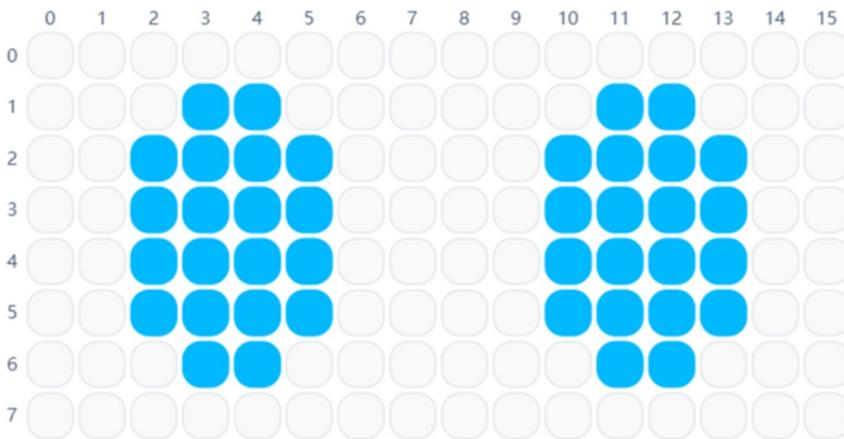
A stylized graphic with the word "NEW" in red, bold, italicized letters, set against a white starburst shape. The background consists of purple, yellow, and blue halftone patterns.

ПОКАДРОВАЯ АНИМАЦИЯ

Покадровая анимация (англ. - «Stop motion») – это техника анимации, при которой отрисовывают каждый кадр.

Фильм или мультфильм состоит из сотен или даже тысяч кадров, каждый из которых прорисовывается отдельно. Каждый новый кадр не похож на предыдущий, и от этого анимация становится более живой.

Нашим холстом для создания анимационных фильмов будет служить экран размером 16 на 8 точек. Рисовать на нем очень просто, выберите инструмент **Рисование**: если вы нажимаете на пустую точку—она закрашивается, если на закрашенную—очищается.



Рисование



Ластик



Очистить холст



Убрать нумерацию



Сохранить



Сохраненные кадры

Напоминаем, что за управление экраном отвечают фиолетовые блоки расположенные в категории «**Внешность**», для создания покадровой анимации нам потребуется блок **показать [изображение] в течение [секунд]**



1010101001111011



АНИМАЦИЯ

Задание 1. Напишите программу, чтобы Codey Rocky подмигивал при нажатии на кнопку А.

при нажатии кнопки А ▾

- показывать изображение в течение 0.2 сек

Задание 3. Напишите программу, анимирующую рост подсолнуха.



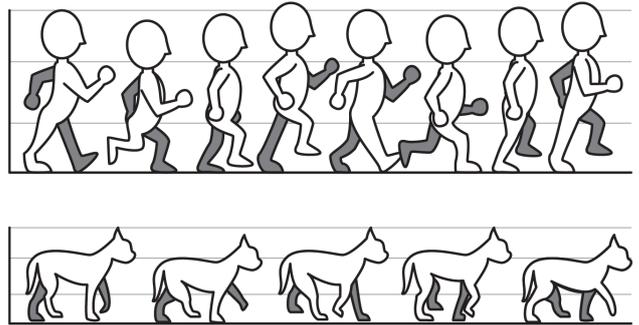
при нажатии кнопки А ▾

- показывать изображение в течение 1 сек

при нажатии кнопки А ▾

- показывать изображение в течение 1 сек
- показывать изображение в течение 1 сек
- показывать изображение в течение 1 сек

Задание 2. Напишите программу, анимирующую бегущего человека. Определите каким образом можно менять скорость забега.

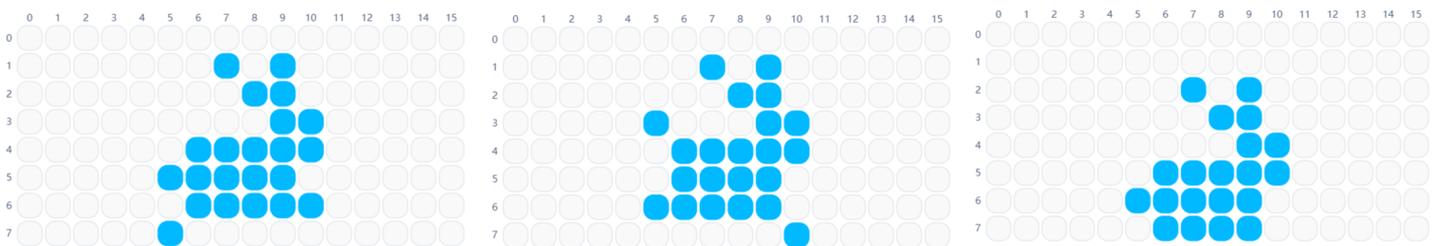




Для лучшего закрепления материала предлагается упражнение для самостоятельного выполнения. Финальный код не выводится на экран для проверки. Ученики в парах создают его самостоятельно, а затем демонстрируют учителю.

Вышел зайчик погулять

Напишите программу, имитирующую движение зайца. В качестве ключевых кадров используйте следующий шаблон.



Вы можете начать свою анимацию с любой стороны: начинать с сидящего зайчика, или постепенно выводить его на экран, остановившись в конце. Можно также добавить звуковое сопровождение прыжков.

Аниматор 80-ого уровня

Для тех кто успел, выполнить все задания из предложенных можно дать полную свободу творчества в создании своей анимации. Это может быть как абстрактная анимация, так и визуализация какого-то реального процесса по аналогии с примерами рассмотренными на уроке.

В качестве подготовки к следующим занятиям, где будут изучаться циклы, а затем и гейм-дизайн можно попросить учеников создать анимацию ожидания.

Анимации ожидания (Idle animations) - это движения игрового персонажа, когда игрок ненадолго оставляет игру без внимания. В этот момент персонаж начинает делать какие-то движения, в ожидании пока игрок вернется в игру.

Анимация ожидания при хорошей реализации становятся важнейшим способом передачи тонких (или не совсем тонких) нюансов персонажей и атмосферы игры.

Подробнее об анимации ожидания можно прочитать в статье [«Что такое качественная idle-анимация? Разработчики делятся своими любимыми примерами»](#)

Хотите получить
доступ ко всем
материалами
курса?

[Регистрация здесь](#)

