

Западное управление Министерства образования и науки Самарской области  
Структурное подразделение «Центр внешкольной работы» государственного  
бюджетного общеобразовательного учреждения Самарской области средней  
общеобразовательной школы №9 имени кавалера ордена Мужества Д.И.  
Герасименко "Центр образования" городского округа Октябрьск Самарской  
области

## «Программирование на языке Scratch\*»

**Номинация:** «Учебные материалы к дополнительным общеобразовательным  
программам технической направленности»

Разработка учебных занятий

Педагог дополнительного образования

Голенкова Татьяна

Владимировна

Октябрьск 2024

## Введение

В современном мире, где технологии проникают во все сферы жизни, навык программирования становится неотъемлемой частью профессиональной компетенции. Он развивает логическое мышление, креативность и умение решать сложные задачи, что положительно сказывается на общем уровне интеллекта.

Одним из языков программирования, который можно изучать с младшего школьного возраста – стал язык Scratch. Это удивительный язык программирования, созданный для обучения детей и подростков основам кодирования. Его визуальный интерфейс позволяет пользователям создавать интерактивные истории, игры и анимации, используя простые блоки кода, которые можно перетаскивать и соединять. Каждый блок представляет собой конкретное действие или команду, что делает процесс программирования интуитивно понятным. Например, с помощью блока "двигаться" персонажа можно переместить по экрану, а блок "звук" добавляет музыкальные эффекты или реплики персонажей. Это создает не только возможность научиться логике и алгоритмическому мышлению, но и развивает креативность, позволяя юным создателям реализовывать свои уникальные идеи. Scratch способствует сотрудничеству, так как пользователи могут делиться своими проектами на платформе, вдохновляя друг друга на новые эксперименты. Таким образом, знакомство с программированием через Scratch становится увлекательным и доступным занятием для каждого, независимо от уровня подготовки. Это шаг к будущему, где навыки программирования будут неотъемлемой частью жизни и работы каждого человека. Важно отметить, что Scratch развивает креативное мышление и навыки решения проблем, что крайне важно в современном мире. Это делает его не только языком программирования, но и инструментом для развития логического мышления и командной работы. В итоге, Scratch открывает двери в мир технологий, вдохновляя новое поколение программистов и творцов.

## **Аннотация**

Для учеников, которые освоили программу технической направленности «Программирование на Scratch» и готовы перейти на следующий уровень, предлагается разработка дополнительного изучения визуальной среды Scratch\*. Данная разработка учебных занятий для детей в возрасте от 12 до 16 лет, которые уже имеют опыт работы с Scratch и хотят расширить свои знания и навыки в области программирования и углубленно изучить возможности среды Scratch.

Целью программы является: повышение мотивации к изучению программирования через создания творческих проектов в среде Scratch, а также развитие творческого и логического мышления.

### **Задачи программы:**

#### Образовательные

- освоить базовые понятия объектно-ориентированного программирования и применять их при создании проектов в среде визуального программирования Scratch;
- приобщить обучающихся к новым технологиям способным им помочь в освоении и развитии творческого потенциала;
- совершенствовать навыки работы на компьютере и повышать интерес обучающихся к программированию.

#### Воспитательные

- способствовать развитию коммуникативных навыков, обучающихся;
- формировать умение демонстрировать свои работы;
- развивать творческие способности и эстетическое восприятие обучающихся.

#### Развивающие

- Способствовать развитию алгоритмического мышления, памяти, умения анализировать;
- формировать потребность в саморазвитии;
- создавать условия повышения самооценки обучающегося, как реализации его личности.

Основная цель разработки учебных занятий заключается в создании увлекательной и образовательно насыщенной среды, где дети смогут расширять свои навыки программирования, учась применять нестандартные решения и развивая креативное мышление.

Задачи данной разработки включают в себя:

1. Изучение более сложных концепций программирования, расширений Scratch с акцентом на практическое применение в проектах.
2. Разработка собственных компьютерных игр и анимаций, что позволит детям не только освоить теорию, но и реализовать свои идеи на практике.
3. Формирование навыков работы в команде через групповые проекты, где участники смогут обмениваться опытом, идеями и способами решения задач.
4. Внедрение игровых элементов в образовательный процесс, что способствует увеличению мотивации и интереса к обучению.

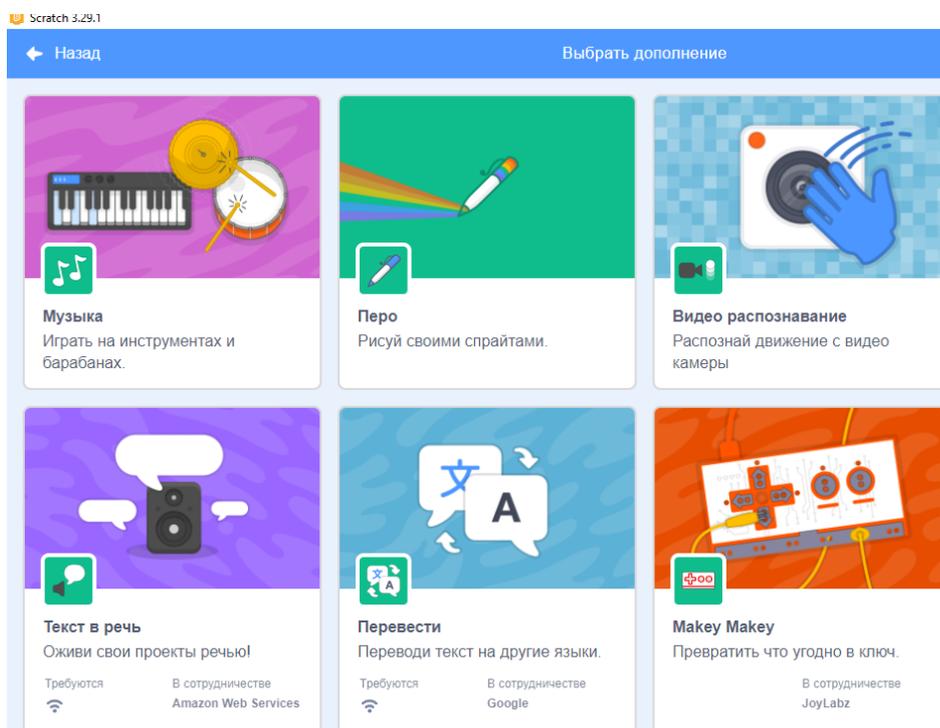
Апробация: углубленное изучение материала открывает двери к новым возможностям, позволяя каждому ребенку исследовать и экспериментировать в безопасной и поддерживающей среде. Эти знания и умения могут стать основой для будущих достижений в области STEM (наука, технологии, инженерия и математика), формируя новое поколение создателей и инноваторов, готовых менять мир. Таким образом, данная разработка

направлена на комплексное развитие навыков программирования, креативности и сотрудничества среди детей.

## Практическая часть

### 1. Дополнения в Scratch

Это набор блоков, которые можно добавить в палитру блоков и расширить возможности Scratch. Это новая функция, которая позволяет скретчеру заставлять программы Scratch взаимодействовать с внешним оборудованием и онлайн-ресурсами. Дополнения могут помочь переосмыслить игры в Scratch. Scratch 3.0 поддерживает одиннадцать различных типов дополнений. Пять из них — веб-дополнения, а именно: Музыка, Перо, Видео распознавание, Текст в речь и Перевод, которые подключают Scratch к дополнительным функциям в Интернете. Остальные шесть — это аппаратные расширения, а именно Makey Makey, micro: bit, LEGO MINDSTORMS EV3, LEGO BOOST, LEGO EDUCATION WeDo2.0 и Go Direct Force and Acceleration, которые подключают Scratch к внешнему устройству. Чтобы добавить дополнений, пользователь должен нажать на синий значок в нижнем левом углу экрана, рядом с категориями блоков. Откроется библиотека дополнений, из которых можно выбрать вариант дополнения. После этого он появится в категориях блока.



## 1.1 Дополнение «Музыка»

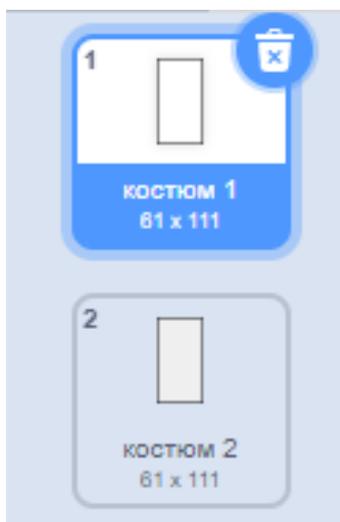
Пользователи Scratch могут использовать дополнение Музыка для воспроизведения нот и сочинения музыки. Звуки в этих блоках создаются с помощью сэмплированных инструментов.

<p><b>Барабану играть</b> 0 0 бита <span style="float: right;">:англ:</span></p> <p>Код:</p> <pre>барабану играть ( ) ( v ) (0.25) бита</pre> <p><b>Описание:</b> Воспроизводит звук указанного барабана в течение указанного количества ударов.</p>	<p><b>Пауза в течение</b> 0 бит <span style="float: right;">:англ:</span></p> <p>Код:</p> <pre>пауза в течение ( ) бит</pre> <p><b>Описание:</b> Приостанавливает этот скрипт на заданное количество ударов.</p>
<p><b>Играй ноту</b> 0 0 бита <span style="float: right;">:англ:</span></p> <p>Код:</p> <pre>играй ноту ( ) ( ) бита</pre> <p><b>Описание:</b> Воспроизводит указанную ноту в течение указанного количества ударов.</p>	<p><b>Изменить инструмент на</b> 0 <span style="float: right;">:англ:</span></p> <p>Код:</p> <pre>изменить инструмент на ( )</pre> <p><b>Описание:</b> Устанавливает инструмент на указанный инструмент.</p>
<p><b>Установить темп</b> 0 <span style="float: right;">:англ:</span></p> <p>Код:</p> <pre>установить темп ( )</pre> <p><b>Описание:</b> Устанавливает темп на указанную величину во всех спрайтах. Влияет на все блоки расширения.</p>	<p><b>Изменить темп на</b> 0 <span style="float: right;">:англ:</span></p> <p>Код:</p> <pre>изменить темп на ( )</pre> <p><b>Описание:</b> Изменяет темп на указанную величину во всех спрайтах. Влияет на все блоки расширения.</p>
<p><b>Темп</b> <span style="float: right;">:англ:</span></p> <p>Код:</p> <pre>( темп )</pre> <p><b>Описание:</b> Этот репортёр сообщает текущий темп.</p>	

## Проект «Пианино в Scratch» с использованием дополнения «Музыка»

Нарисуем новый спрайт – белую клавишу. Добавим расширение музыка.

Клавише сделаем еще один костюм – светло-серый (чтобы имитировать нажатие на клавишу):



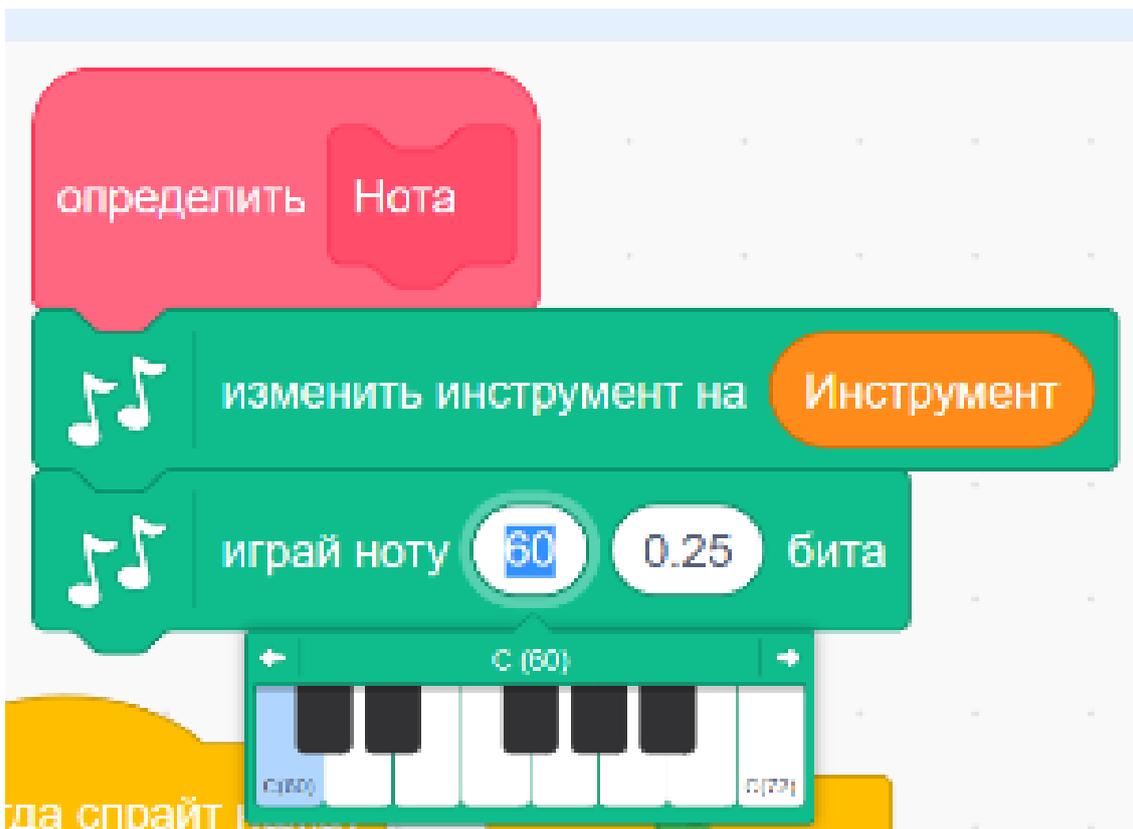
Для программы на клавишу введем другие блоки (см. п. 2):



Зададим переменную инструмент и выберем ее вид – рычажок, для смены инструмента и звучания соответственно. Скопируем спрайт белой клавиши и выставим пианино.



Также добавим черные клавиши, разница в скрипте у спрайтов в ноте:



Попробуем сыграть песенку «В лесу родилась ёлочка»:

«В лесу родилась ёлочка»

В ле-су ро-ди-лась ё-лоч-ка,  
В ле-су о-на рос-ла.  
Зи-мой и ле-том строй-на-я,  
Зе-лё-на-я бы-ла.

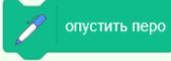
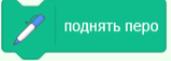
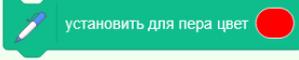
The image shows the song title and four lines of lyrics. Above each line are numbered notes in various colors (red, blue, green, orange, black) representing the pitch and rhythm of the melody. The notes are: Line 1: 1, 6, 6, 5, 6, 4, 1, 1; Line 2: 1, 6, 6, 6|7, 5, 1; Line 3: 1, 2, 2, 6|7, 6|7, 6, 5, 4; Line 4: 4, 6, 6, 5, 6, 4.

Вот так выглядит скрипт песни, если собрать его по нотам:

The image shows a vertical musical score for the song. It consists of 20 staves, each starting with a treble clef and a key signature of one flat (B-flat). The notes are written in a standard musical notation. The score is presented in a vertical orientation, with the staves stacked on top of each other. The notes are: Staff 1: G4, B4, G4, F4, G4, E4, G4, G4; Staff 2: G4, B4, G4, G4|A4, F4, G4; Staff 3: G4, A4, A4, B4|A4, B4|A4, G4, F4, E4; Staff 4: E4, G4, G4, F4, G4, E4.

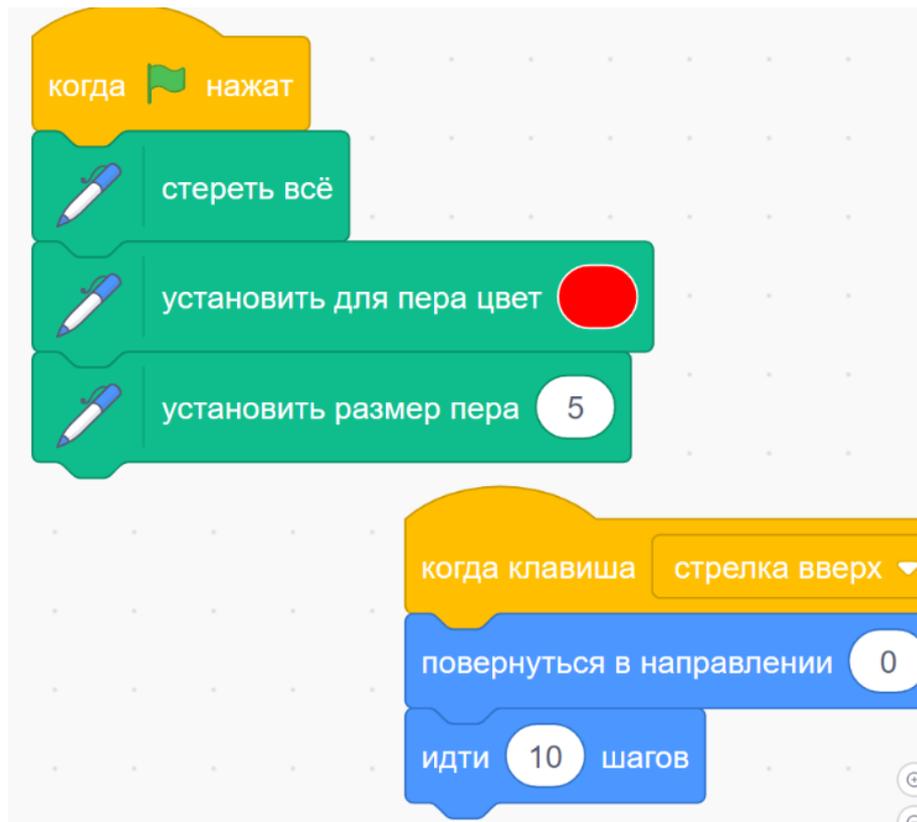
## 1.2 Дополнение «Перо»

Позволяет спрайтам управлять своими ручками, а также отпечатывать себя на сцене и удалять все следы от пера. Его также можно использовать для создания игр и симуляций, а также других проектов.

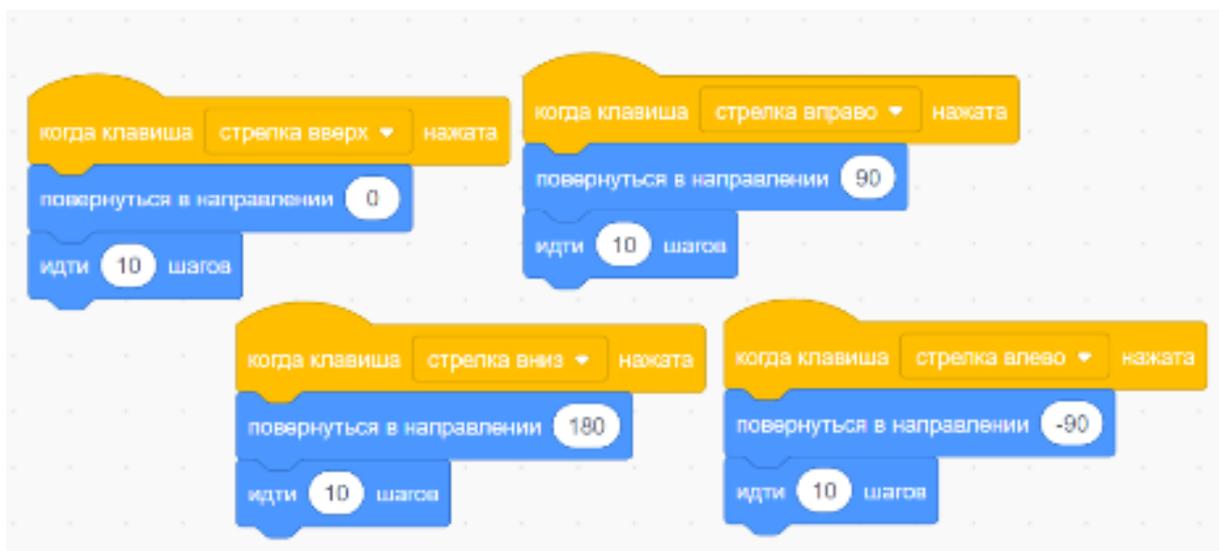
 <p>Код:</p> <input type="text" value="стереть всё"/> <p><b>Описание:</b> Очищает сцену от рисовок, в том числе и от печатей</p>	 <p>Код:</p> <input type="text" value="печатать"/> <p><b>Описание:</b> Отпечатывает текущий костюм спрайта на сцену</p>
<p><b>Опустить перо</b> :англ:</p>  <p>Код:</p> <input type="text" value="опустить перо"/> <p><b>Описание:</b> «Опускает» перо спрайта на сцену — то есть позволяет начать рисовать</p>	<p><b>Поднять перо</b> :англ:</p>  <p>Код:</p> <input type="text" value="поднять перо"/> <p><b>Описание:</b> «Поднимает» перо спрайта со сцены — то есть говорит спрайту, чтобы он перестал рисовать</p>
<p><b>Установить для пера цвет []</b> :англ:</p>  <p>Код:</p>	
<p><b>Изменить (v) пера на ()</b> :англ:</p>  <p>Код:</p> <input type="text" value="изменить ( v) пера на ()"/> <p><b>Описание:</b> Меняет для пера значение цвета, насыщенности, яркости и прозрачности. В Спретч 1.0 — 2.0 были только отдельные блоки для цвета и тени (яркости):</p>  	<p><b>Установить (v) пера ()</b> :англ:</p>  <p>Код:</p> <input type="text" value="установить ( v) пера ()"/> <p><b>Описание:</b> Задаёт для пера значение цвета, насыщенности, яркости и прозрачности. В Спретч 1.0 — 2.0 были только отдельные блоки для цвета и тени (яркости):</p>  
<p><b>Изменить размер пера на ()</b> :англ:</p>  <p>Код:</p> <input type="text" value="изменить размер пера на ()"/> <p><b>Описание:</b> Изменяет размер пера на указанное значение. Для <a href="#">flash-перца</a> значение варьируется от 0 до 255.</p>	<p><b>Установить размер пера ()</b> :англ:</p>  <p>Код:</p> <input type="text" value="установить размер пера ()"/> <p><b>Описание:</b> Задаёт размер пера на указанное значение. Для <a href="#">flash-перца</a> значение варьируется от 0 до 255.</p>

## Проект «Рисуем отрезками» с использованием дополнения «Перо»

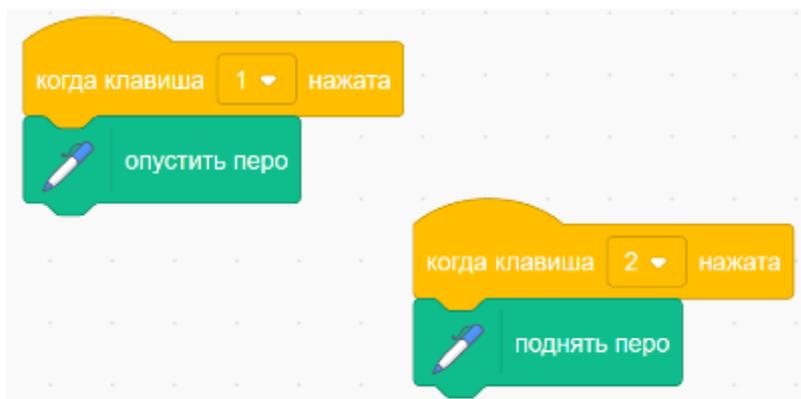
Уменьшим персонажа до 25, напишем скрипт, где выберем цвет пера и размер, а также очистим сцену:



А также клавиши стрелки на движение персонажа:



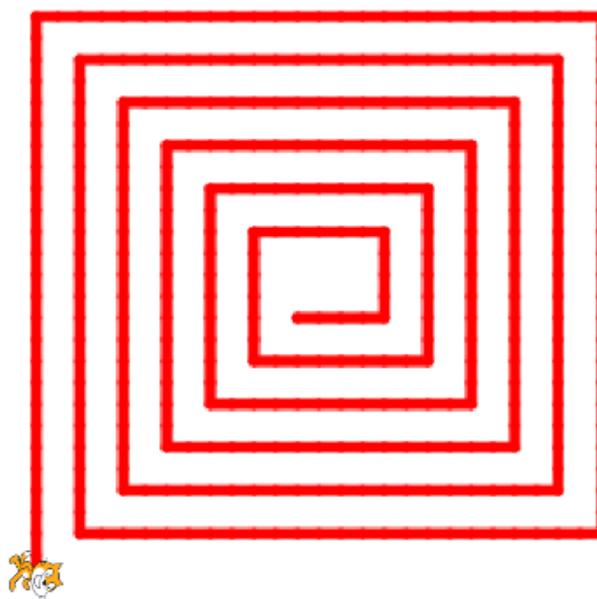
И любые клавиши на то чтобы опустить/поднять перо:



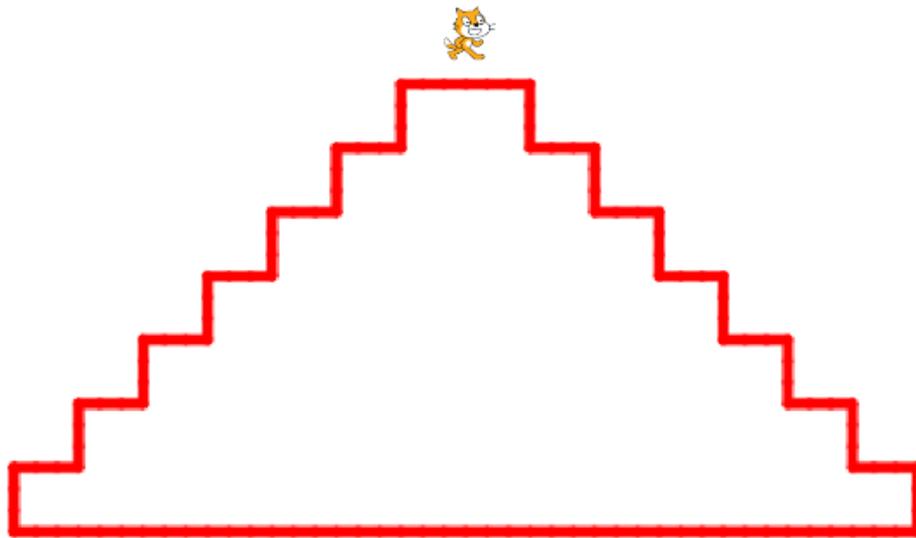
Пробуем нарисовать ряд из квадратов на одинаковом расстоянии (тренируем глазомер):



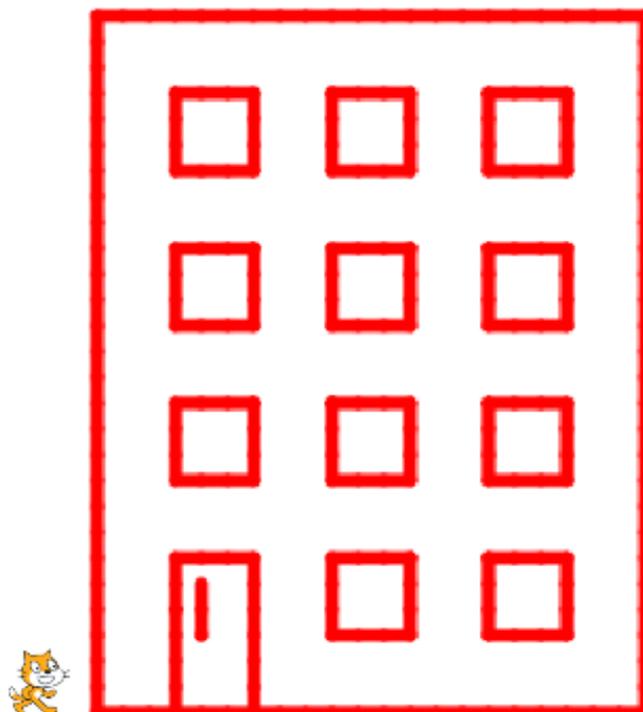
Спираль:



Пирамиду:



Многоэтажку:

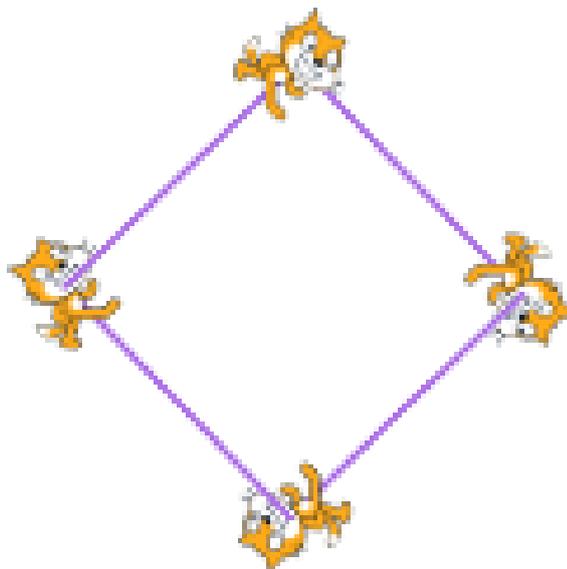
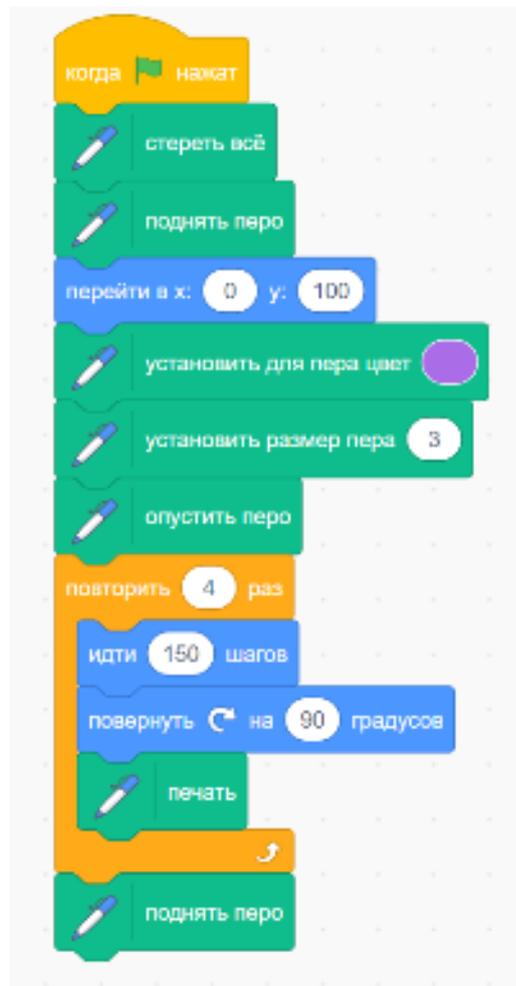


! Если ты случайно нарисуешь неправильный отрезок, как стереть только его?

! Придумай и нарисуй свою разноцветную фигурку

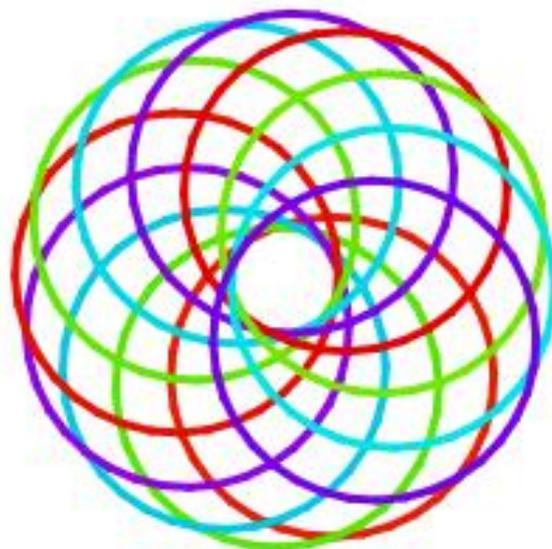
## Проект «Рисуем фигуры» с использованием дополнения «Перо»

Квадрат с котами:

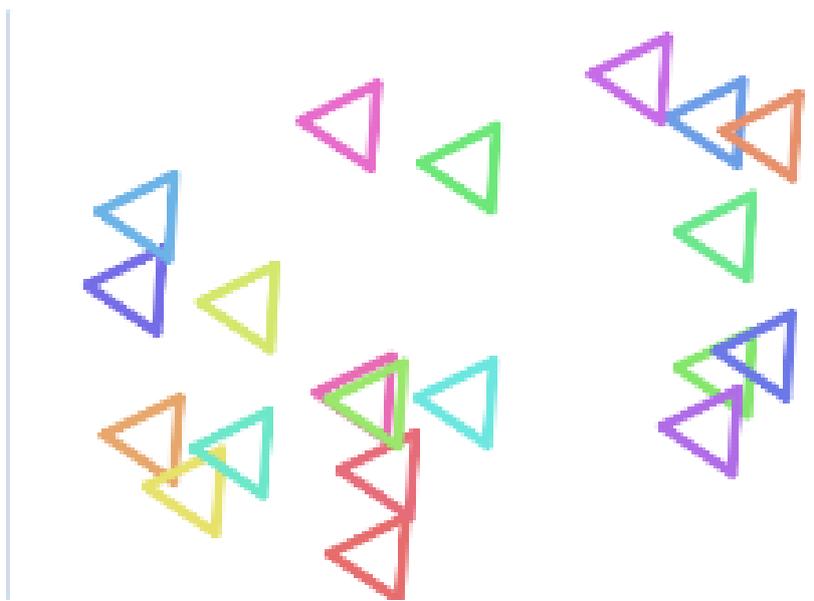


Разноцветные круги по кругу:

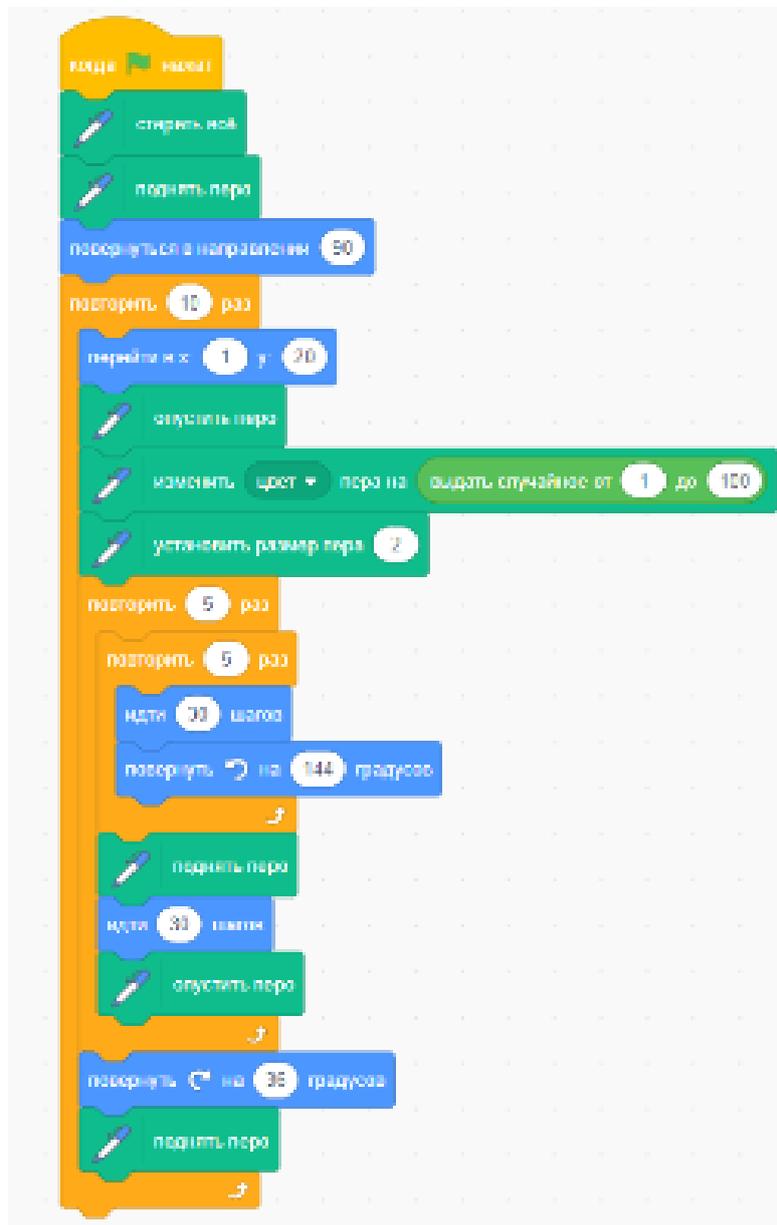
```
когда флажок нажат
  установить для пера цвет [фиолетовый]
  установить размер пера [3]
  повторить 12 раз
    опустить перо
    изменить цвет пера на 25
    повторить 36 раз
      идти 10 шагов
      повернуть на 10 градусов
    поднять перо
    идти 10 шагов
    повернуть на 30 градусов
```



Случайные треугольники:



Звездочки снежинкой:



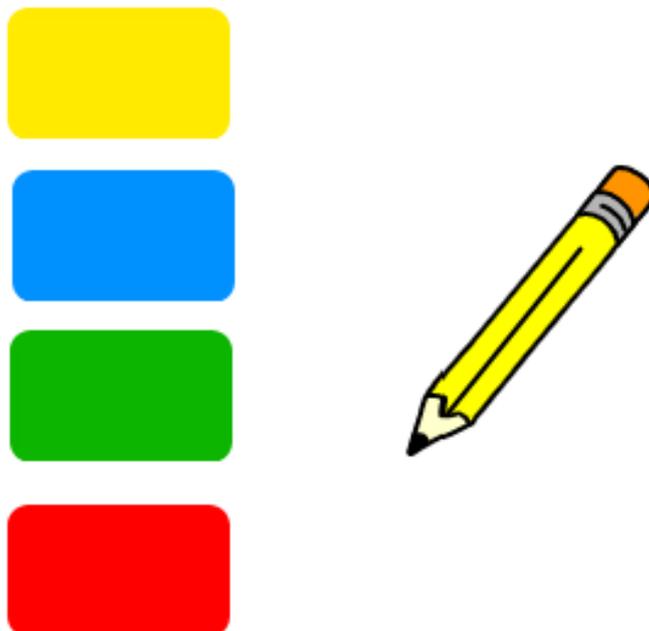
```
when green flag clicked
  start pen
  lift pen
  turn right 90 degrees
  repeat 10 times
    go to x: 1 y: 30
    lift pen
    change cost of pen size by random number from 1 to 100
    set pen size to 2
    repeat 5 times
      repeat 5 times
        move 30 steps
        turn right 144 degrees
      lift pen
      move 30 steps
      lift pen
    turn right 36 degrees
    lift pen
```



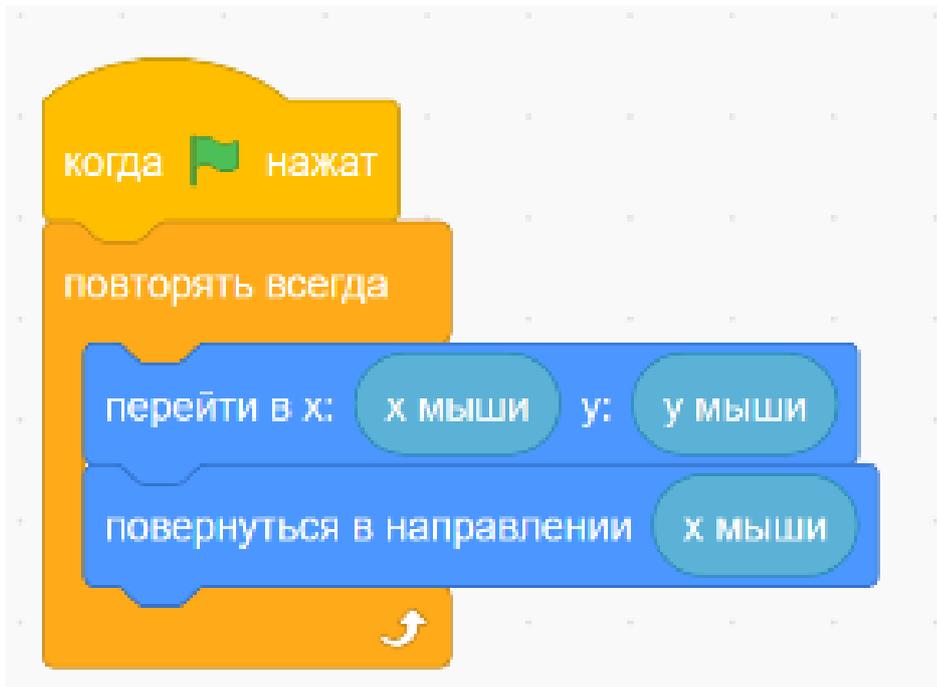
Очень большое количество фигур можно построить с помощью дополнения Перо.

### **Проект «Создаем Paint в Scratch» с использованием дополнения «Перо»**

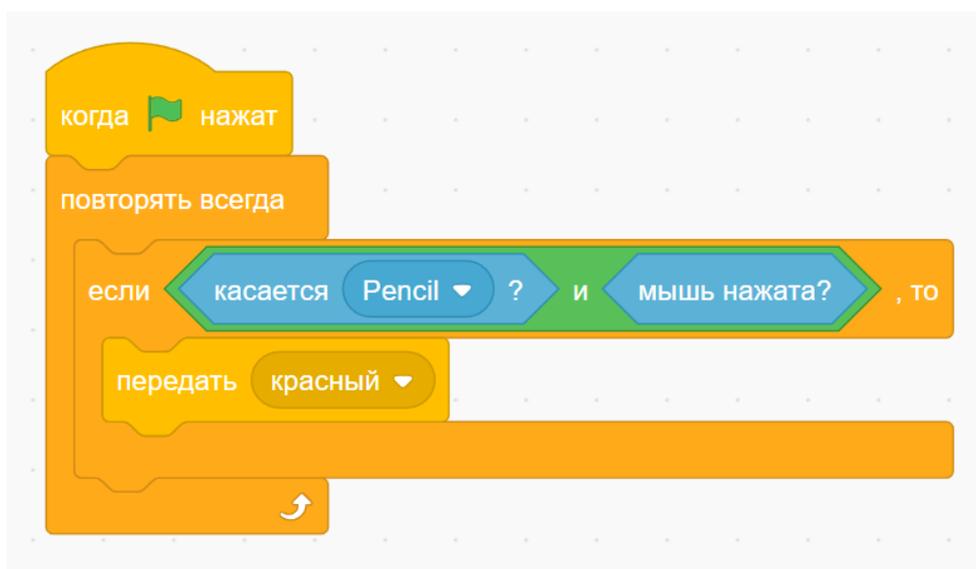
Создаем палитру из нескольких цветов и добавляем из библиотеки карандаш:



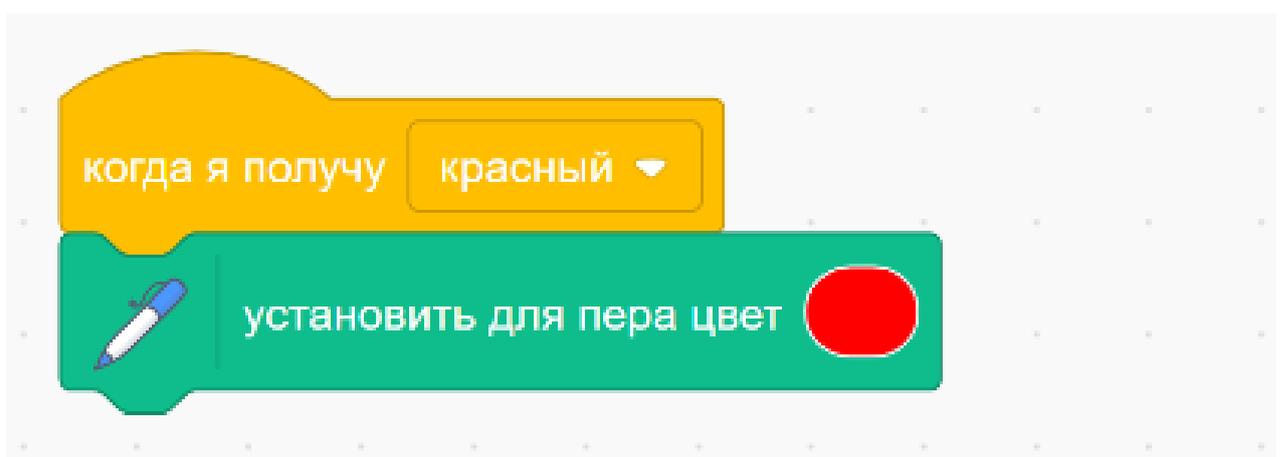
Программа на карандаш для управления им мышью:



Программа на каждый цвет палитры, меняется только сообщение:

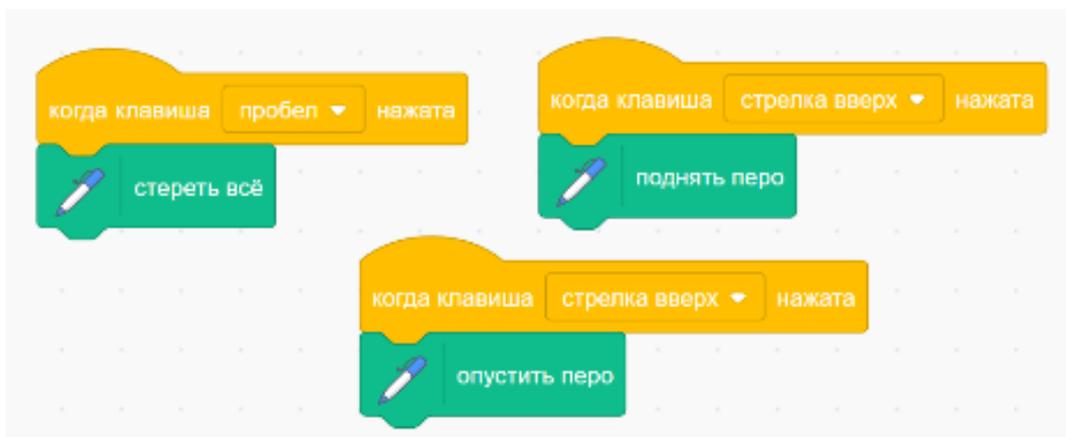


И программа для карандаша, реагирующая на сообщение цвета и меняющая



ЦВЕТ СООТВЕТСТВЕННО:

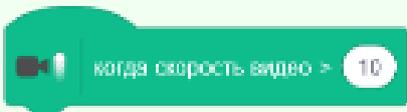
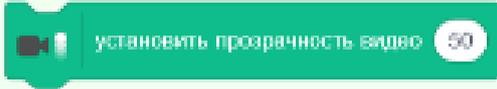
И



дополнить программу на карандаш:

### 1.3 Дополнение «Видео распознавание»

Это расширение позволяет веб-камерам взаимодействовать с проектами.

<p><b>Когда скорость видео &gt; ()</b> :англ:</p>  <p>Версия: 3.0 Код:</p> <pre>когда скорость видео &gt; (10)</pre> <p><b>Описание:</b> Активируется, когда текущее движение видео на сцене превышает введенное значение.</p>	<p><b>Видео (v) на (v)</b> :англ:</p>  <p>Версия: 2.0 Код:</p> <pre>(видео (движение v) на (спрайт v))</pre> <p><b>Описание:</b> Возвращает движение (по шкале от 0 до 100) или направление движения на видео на сцене или текущем спрайте. Если видео нет, блок вернёт -1.</p>
<p><b>Включить видео (v)</b> :англ:</p>  <p>Версия: 2.0 Код:</p> <pre>(включить видео (включить v))</pre> <p><b>Описание:</b> Включает, выключает или переключает веб-камеру в зависимости от аргумента.</p>	<p><b>Установить прозрачность видео ()</b> :англ:</p>  <p>Версия: 3.0 Код:</p> <pre>установить прозрачность видео (50)</pre> <p><b>Описание:</b> Устанавливает прозрачность видео на определенное значение.</p>

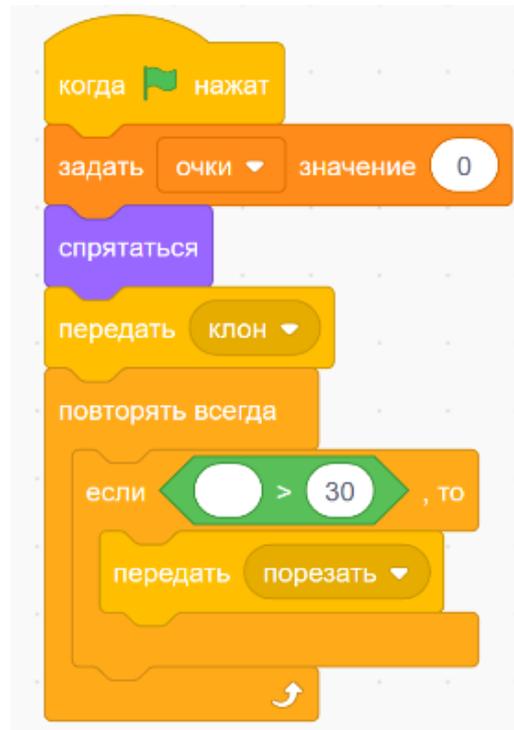
посредством видеосенсора.

## Проект «Дополненная реальность в Scratch» с использованием дополнения «Видео распознавание»

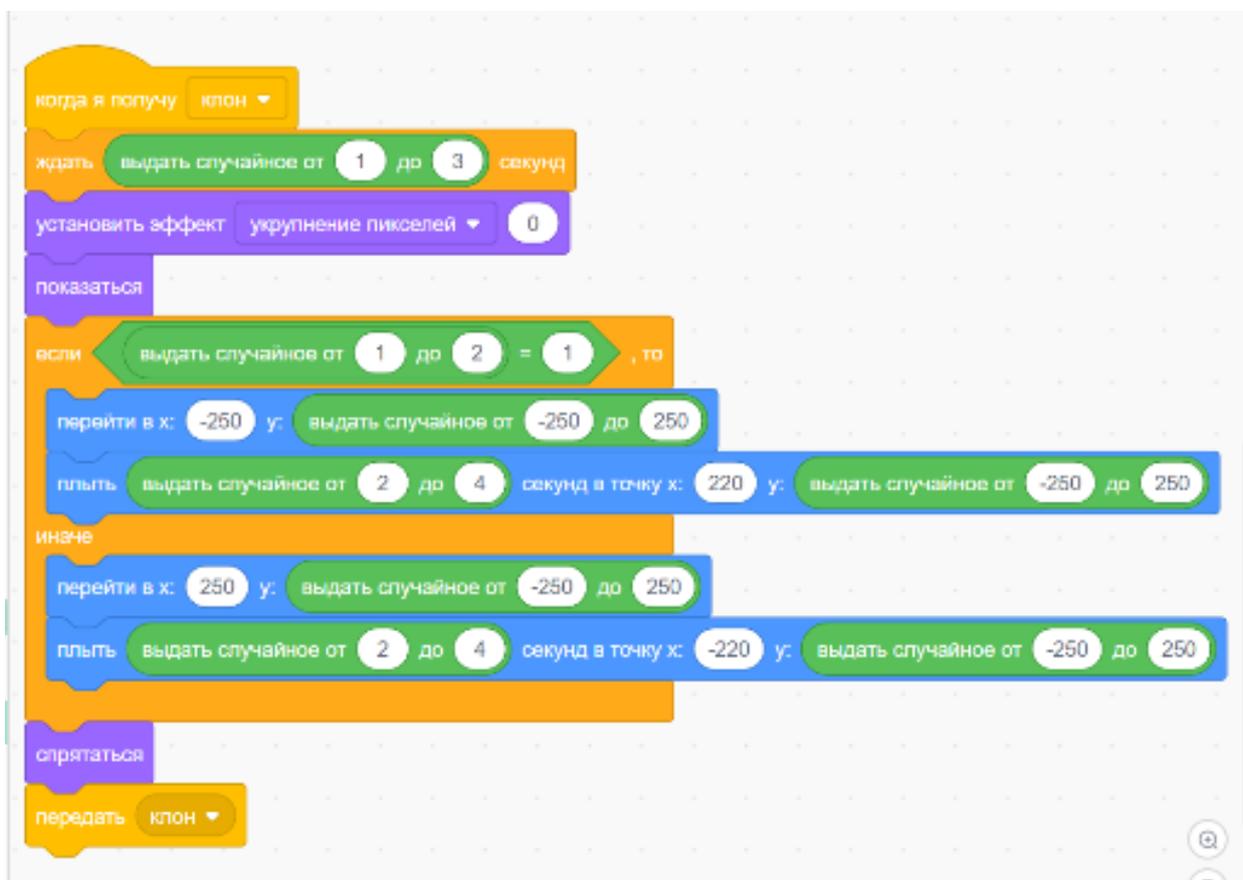
Добавим спрайт (например, мяч):



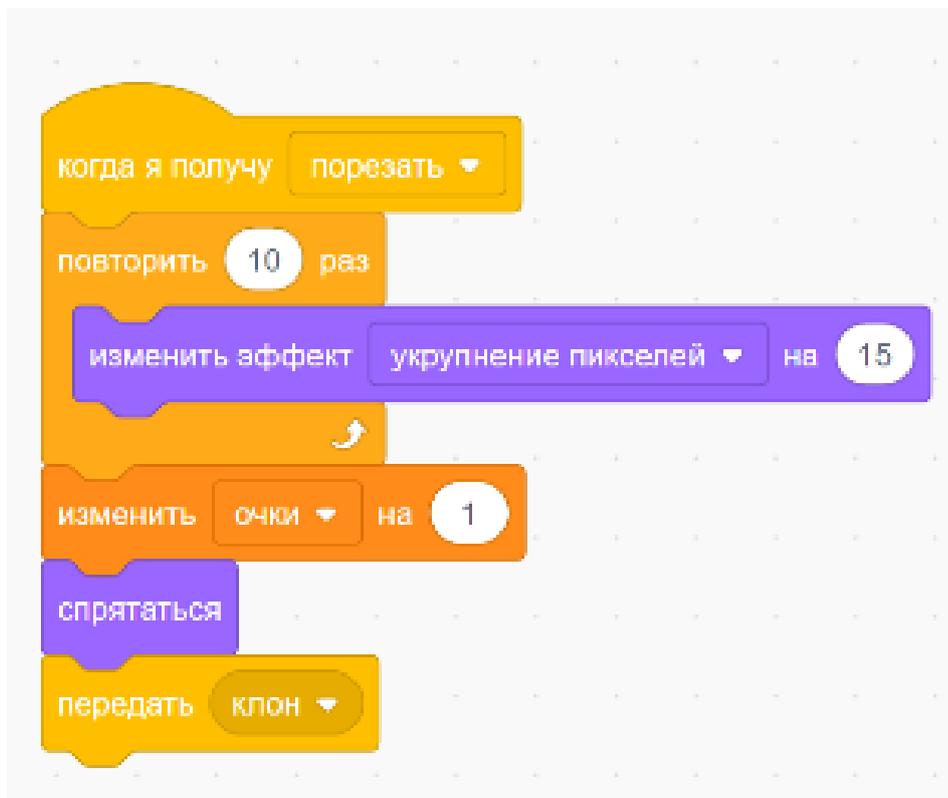
И напишем программу на создание клонов:



Распишем поведение клонов:

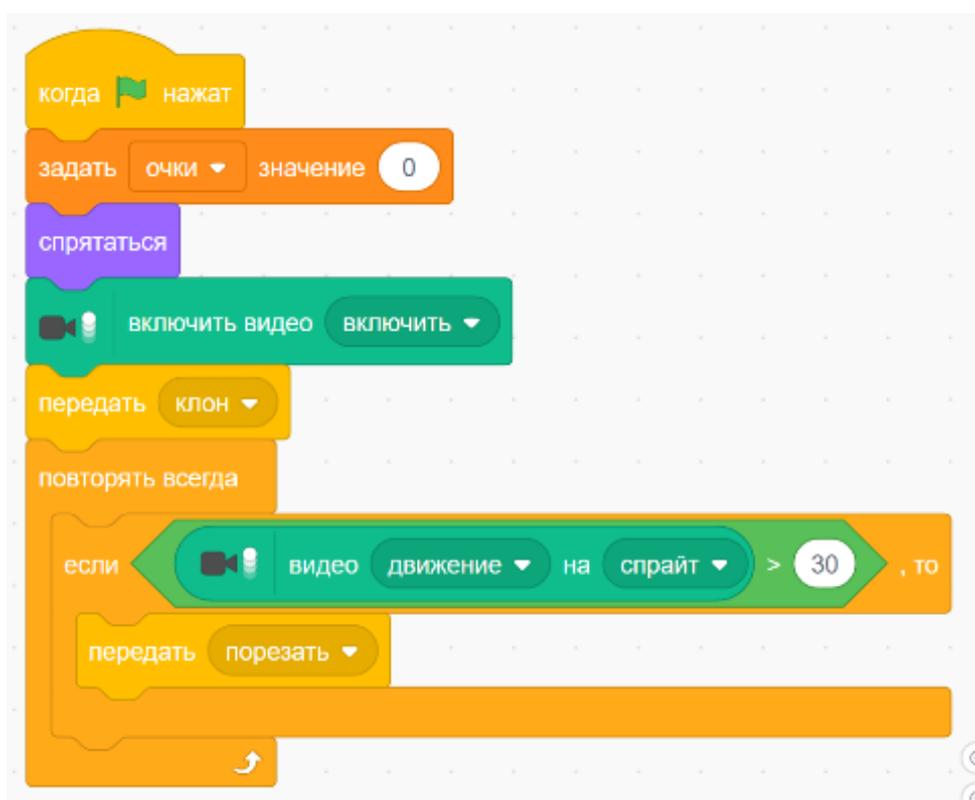


Напишем, что произойдет с клонами при видео касании:



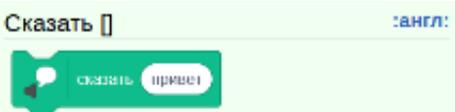
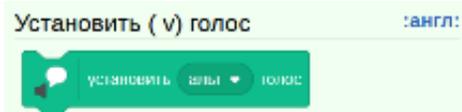
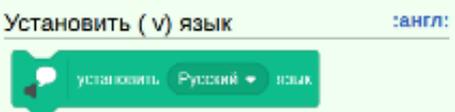
Добавим дополнение видео распознавание и вставим блок в пропущенные.

Тестируем:



## 1.4 Дополнение «Текст в речь»

Проекты, использующие это расширение, могут выводить синтезированную речь. Это расширение позволяет менять язык голоса, например, арабский, китайский, датский и т. д.

<p><b>Сказать []</b> :англ:</p>  <p><b>Версия:</b> 3.0 <b>Код:</b></p> <pre>сказать [привет]::tts</pre> <p><b>Описание:</b> Произнести заданный текст, используя инструмент преобразования текста в речь, работает только при включённом интернете.</p>	<p><b>Установить (v) голос</b> :англ:</p>  <p><b>Версия:</b> 3.0 <b>Код:</b></p> <pre>установить (альт v) голос</pre> <p><b>Описание:</b> Устанавливает текущий голос, используемый для произношения текста.</p>
<p><b>Установить (v) язык</b> :англ:</p>  <p><b>Версия:</b> 3.0 <b>Код:</b></p> <pre>установить (Русский v) язык</pre> <p><b>Описание:</b> Заставляет текст произноситься с использованием произношения указанного языка, но не переводит текст.</p>	

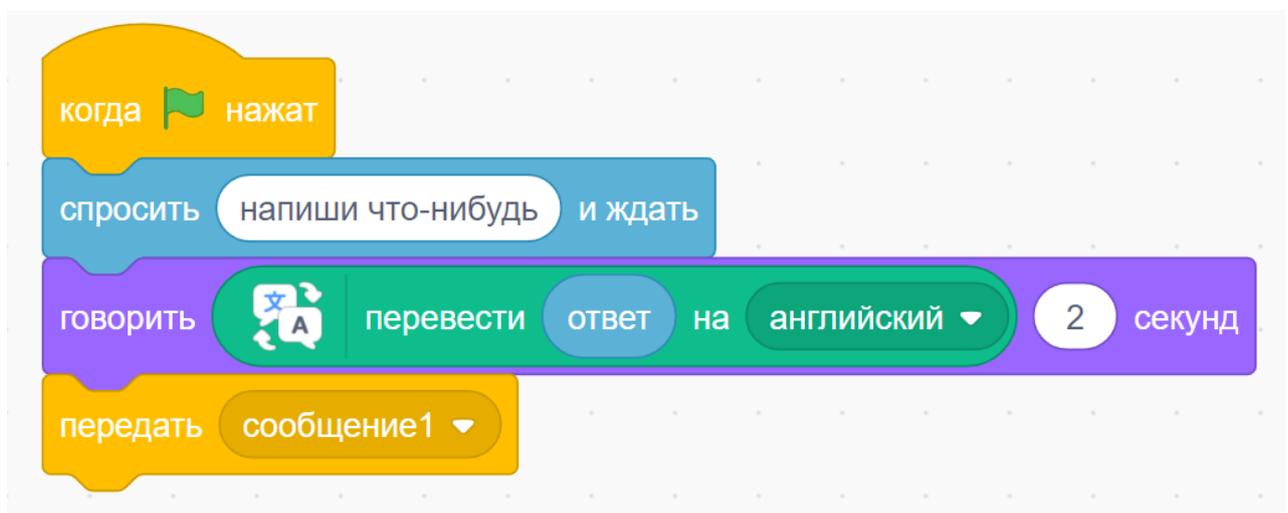
**1.5 Дополнение «Перевести»** - позволяет пользователям использовать Google Translate для перевода слов с одного языка на другой.

<p><b>Перевести [] на (v)</b> :англ:</p>  <p><b>Версия:</b> 3.0 <b>Код:</b></p> <pre>(перевести [привет] на (итальянский v))</pre> <p><b>Описание:</b> Переводит строку на выбранный язык, работает только при включённом интернете.</p>
<p><b>Язык</b> :англ:</p>  <p><b>Версия:</b> 3.0 <b>Код:</b></p> <pre>(язык)</pre>

## Проект «Переводчик» с использованием дополнения «Перевести» и «Текст в речь»

Добавляем спрайт и пишем ему скрипт, который будет переводить всё то что мы ему напишем, и озвучивать на языке, который мы выберем.

Этот скрипт просит написать слово или фразу:



Этот скрипт озвучит ее на языке что мы выбрали:



## 2. Другие блоки в Scratch (пользовательские блоки)

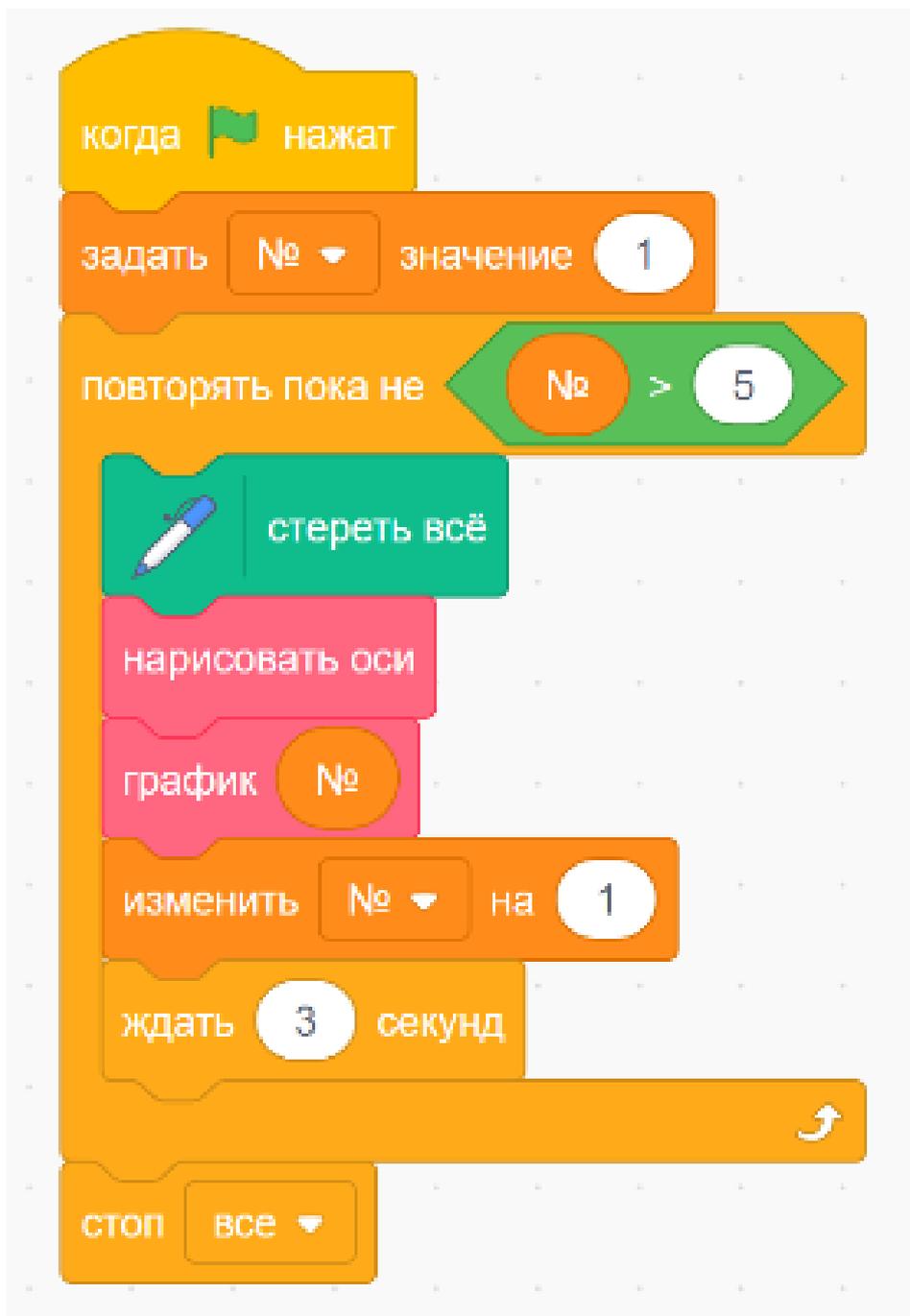
Категория блоков, содержащая процедуры для выбранного спрайта. Блоки этой категории окрашены в розовый цвет. Перед созданием каких-либо блоков категория пуста, за исключением кнопки «Создать блок». Нажатие кнопки «Создать блок» вызывает диалоговое окно, позволяющее пользователю создать процедуру. В диалоговом окне можно добавить строковые и логические входы. После нажатия ОК в палитре появится новый блок, а в области скриптов появится пустое определение. Когда процедура запускается, Scratch запускает блоки ниже соответствующего определения.

Использование пользовательских блоков — единственный способ запуска скриптов без обновления экрана. Обновление экрана — это очень незначительное ожидание, которое происходит в конце каждой итерации блока цикла или, когда используется блок ожидания. Диалоговое окно «Создать блок» предлагает отключить обновление экрана (по умолчанию оно включено). Это может сделать происходящее менее очевидным, но полезно, если вы хотите внести изменения, чтобы пользователь не видел промежуточных шагов или не ждал долгое время. Например, если использовать этот скрипт пользователь может наблюдать, как спрайт рисует картинку. Но если поместить его в пользовательский блок с отключенным обновлением экрана, пользователь увидит только полную картинку, и она нарисуеться гораздо быстрее. Эта функция обычно используется в играх, требующих более быстрой обработки. Это также используется для исправления проблем с синхронизацией, вызванных паузами в скрипте, которые предназначены для того, чтобы сделать движение на экране более заметным, но иногда они могут мешать.

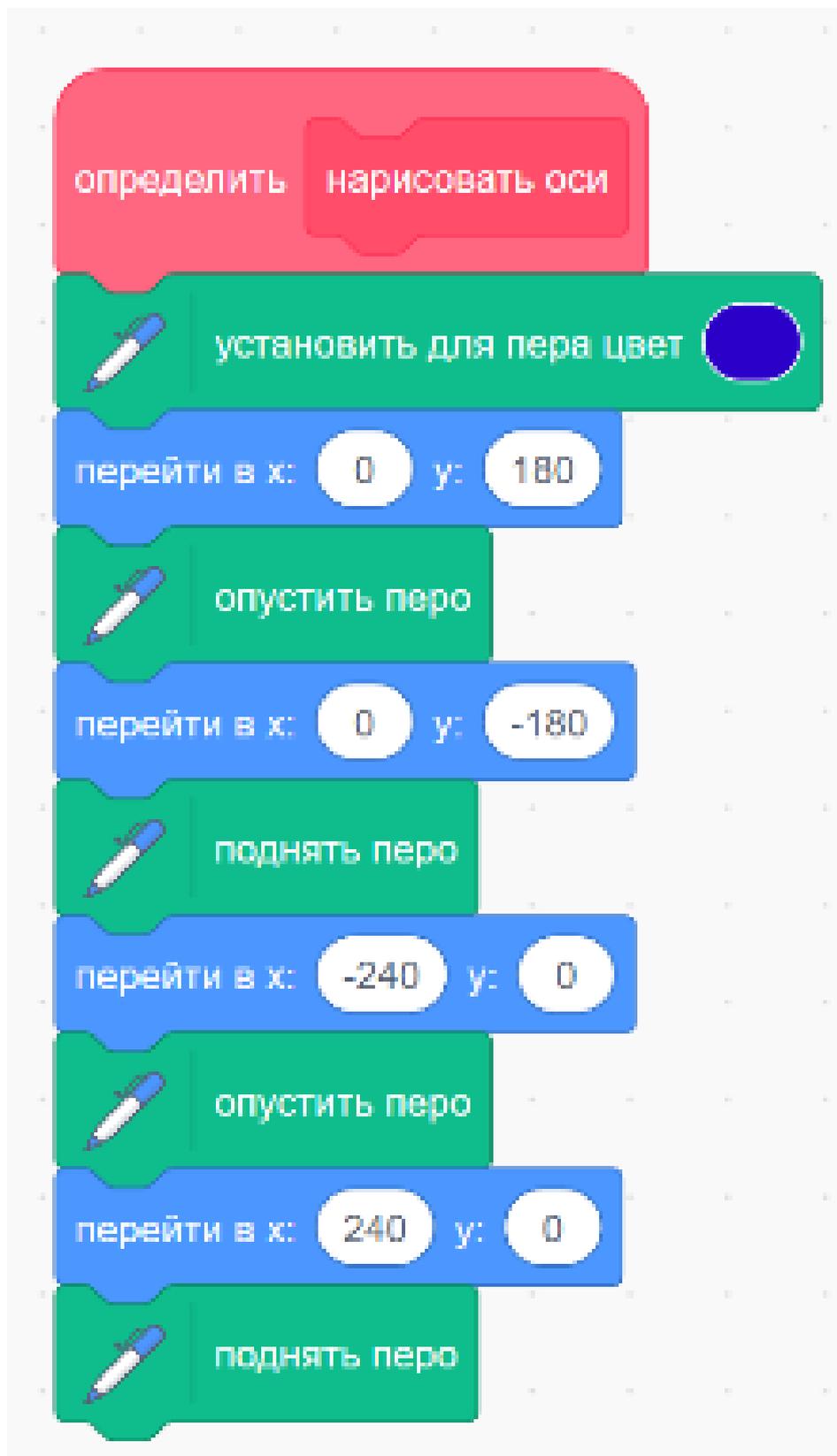
Альтернативой использованию этой функции является включение Турбо-режима, в результате чего все сценарии будут выполняться без обновления экрана. Кроме того, запуск без обновления экрана не следует использовать с пользовательскими блоками, которые содержат какие-либо блоки ожидания, так как это может привести к зависанию редактора и возможному сбою.

### **Проект «Построение графиков различных функций в Scratch» с использованием других блоков**

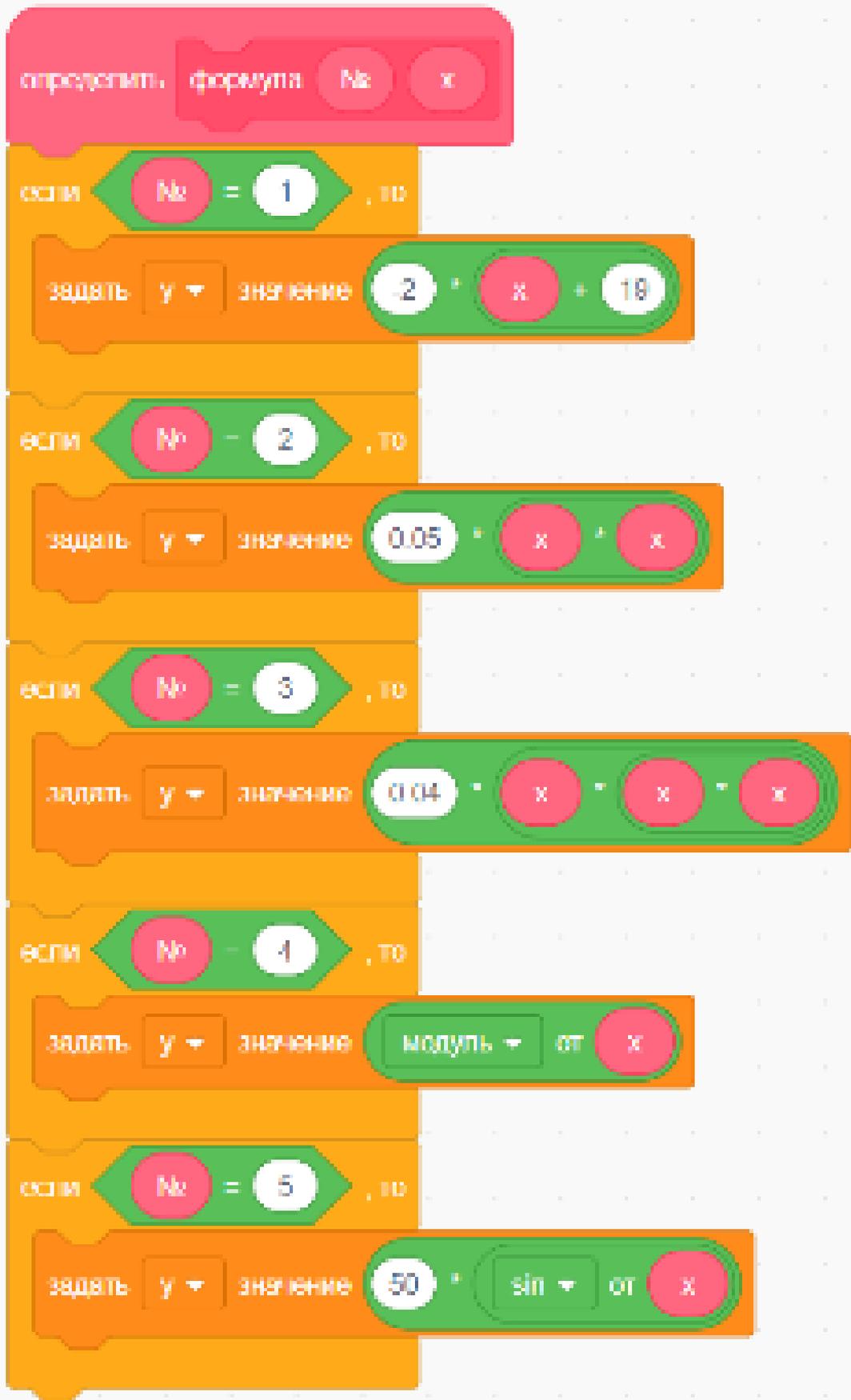
Сначала определим количество графиков, которые будут рисоваться и дадим команду нарисовать оси. У нас будет пять различных графиков функций: прямая, парабола, кубическая парабола, модуль  $x$ , синусоида. Создадим другие блоки: график, нарисовать оси и формула. Также нам понадобится дополнение перо (см.п. 1.2).



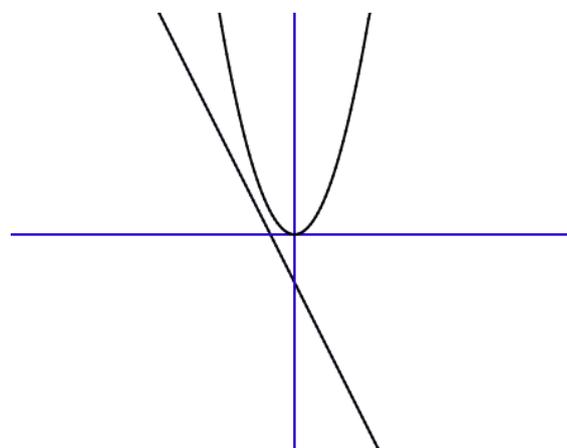
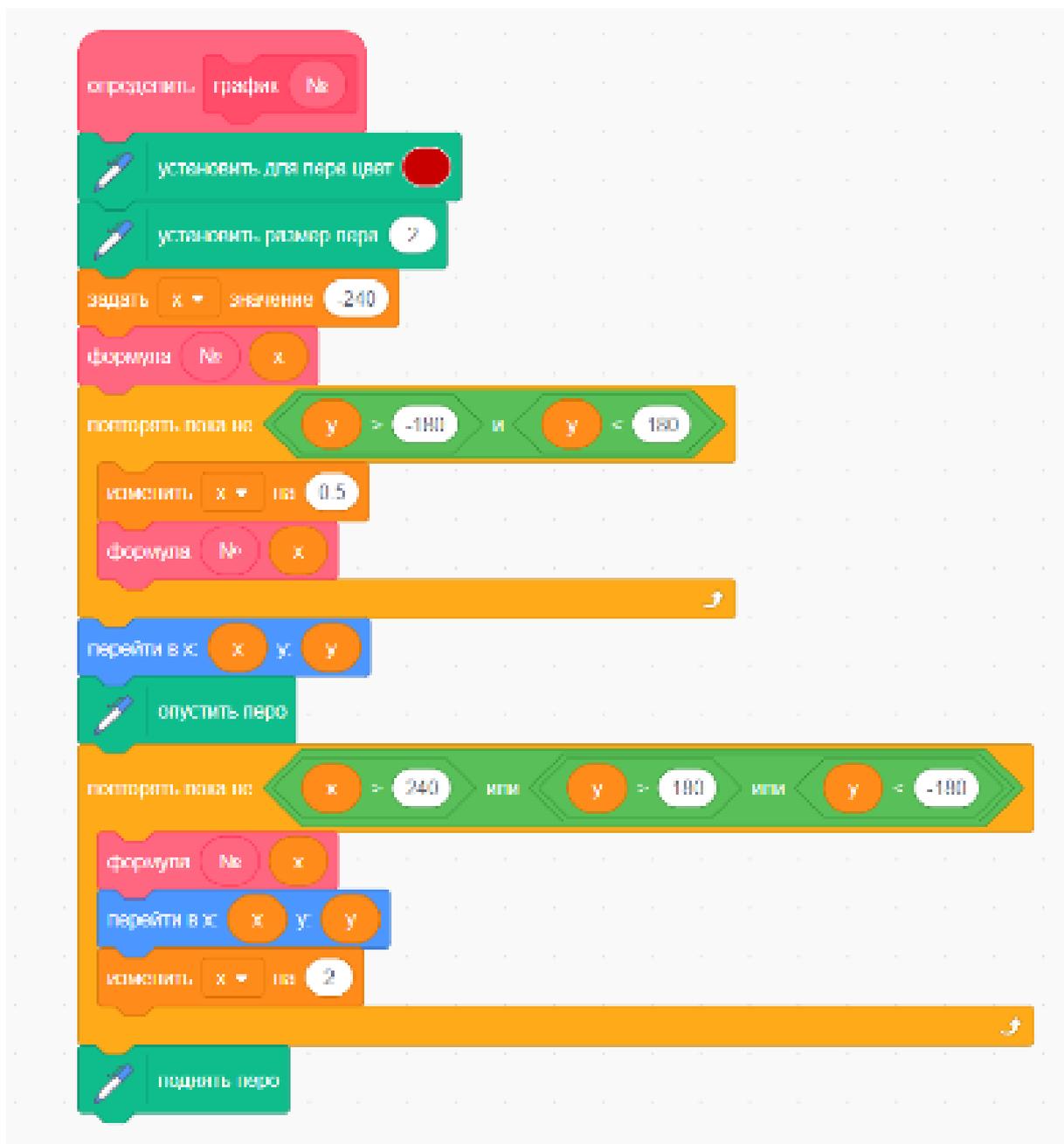
Нарисуем оси:



Напишем формулы графиков, которые хотим построить:



Построим графики:



### 3. Решение математических задач в Scratch

Scratch предоставляет широкие возможности для создания интересных и образовательных математических задач (см. проект выше), и игр. Используя визуальный интерфейс и блоки программирования, можно легко создавать задачи, которые помогут им улучшить свои математические навыки.

#### Проект «Тренировка счета» с использованием математических формул в Scratch

Создадим четыре спрайта – кнопки, которые будут запускать определенное арифметическое действие:



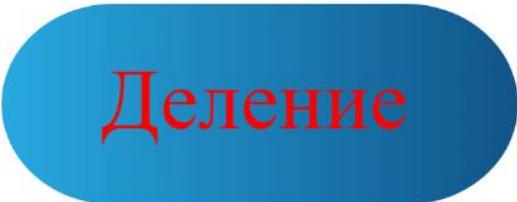
Сложение



Вычитание



Умножение



Деление

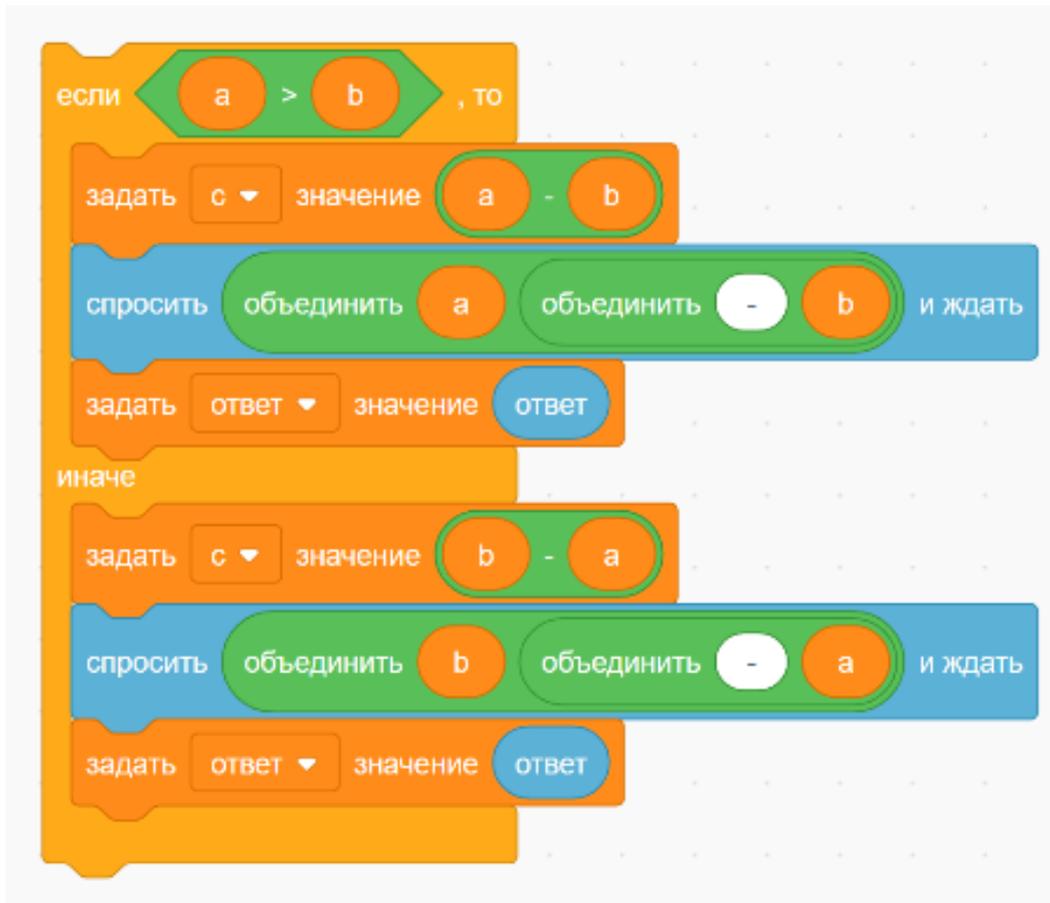
Создаем переменные на числа и на сложение:



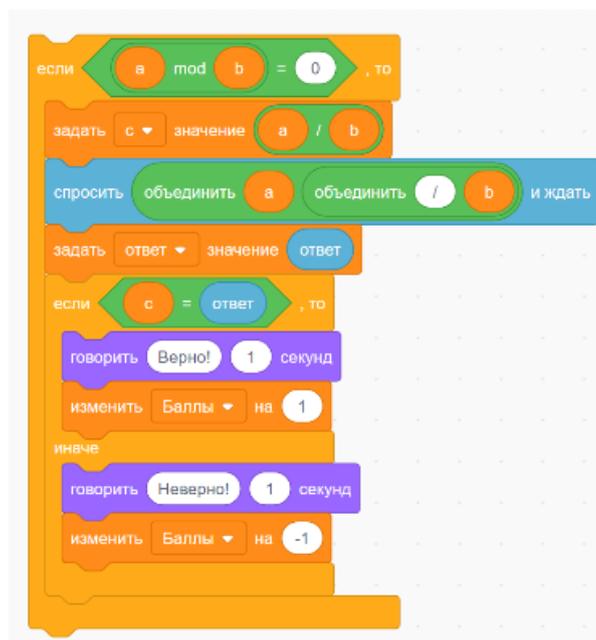
Если ответ, который вводит пользователь, совпадает, то пишется верно, иначе- неверно.

Аналогично пишется скрипт на умножение, только меняется арифметический знак.

С вычитанием сложнее, нужно добавить сравнение чисел, чтобы не получались отрицательные:



А в деление нужно добавить оператор mod, для того чтобы выходили примеры на деление без остатка:



## **Диагностический инструментарий**

Вот несколько ключевых элементов, которые могут быть включены в такой инструментарий:

1. Тестирование знаний: создание тестов или викторин.
2. Проектные задания: предложение учащимся выполнить небольшие проекты, используя полученные знания и навыки в Scratch. Оценка проектов может включать в себя критерии, такие как оригинальность и сложность решения.
3. Решение задач: предложение учащимся решить задачи, которые требуют применения конкретных программистских навыков, например, создание алгоритма для решения конкретной задачи.
4. Портфолио проектов: просьба учащимся создать портфолио своих проектов в Scratch, чтобы продемонстрировать свои достижения и прогресс в области программирования.
5. Самооценка и обратная связь: предоставление учащимся возможности самостоятельно оценить свои знания и навыки, а также получить обратную связь от учителя или своих коллег.

Диагностический инструментарий после изучения Scratch не только помогает оценить уровень компьютерной грамотности учащихся, но и способствует их дальнейшему развитию в этой области. Путем регулярного использования такого инструментария учителя могут отслеживать прогресс учащихся, выявлять слабые места и адаптировать учебный процесс для достижения лучших результатов.

Итак, диагностический инструментарий после изучения Scratch играет важную роль в развитии компьютерной грамотности учащихся, помогая им не только оценить свои знания и навыки, но и продолжать совершенствоваться в области программирования и компьютерной науки.

## Заключение

В заключение, дополнения в Scratch — это мощный инструмент, который может значительно улучшить ваши проекты Scratch. Добавляя новые блоки и функции, вы можете создавать более сложные и интересные проекты, внедрять новые концепции программирования и улучшать пользовательский опыт.

Изучение программы Scratch имеет множество преимуществ и целей, включая:

1. Развитие логического мышления: создание игр и приложений в среде Scratch требует анализа, планирования и решения различных задач, что способствует развитию логического мышления у учащихся.
2. Программирование и алгоритмическое мышление: работа со Scratch позволяет учащимся освоить основы программирования и создания алгоритмов, что может быть полезным в будущем при изучении других языков программирования.
3. Креативное мышление: создание игр в Scratch позволяет учащимся проявить свою креативность, дизайнерские навыки и воображение при разработке уникальных игровых миров и сценариев.
4. Командная работа: работа в Scratch может быть выполнена как индивидуально, так и в группах, что способствует развитию коммуникативных навыков, умению работать в команде и делиться идеями.
5. Повышение интереса к науке и технологиям: язык программирования Scratch предлагает увлекательный и интерактивный способ изучения научно-технических дисциплин, математики, что может помочь стимулировать интерес учащихся к этим областям.

## Оглавление

Введение	2
Аннотация	3
Практическая часть	5
1. Дополнение в Scratch	5
1.1 Дополнение «Музыка»	6
Проект «Пианино в Scratch»	7
1.2 Дополнение «Перо»	10
Проект «Рисуем отрезками»	11
Проект «Рисуем фигуры»	14
Проект «Проект «Paint в Scratch»»	18
1.3 Дополнение «Видео распознавание»	20
Проект «Дополненная реальность в Scratch»	20
1.4 Дополнение «Текст в речь»	23
1.5 Дополнение «Перевести»	23
Проект «Переводчик в Scratch»	24
2. Другие блоки	25
Проект «Рисуем графики функций в Scratch»	26
3. Решение математических задач в Scratch	30
Проект «Тренировка счета»	30
Диагностический инструментарий	33
Заключение	34
Оглавление	35
Список используемой литературы	36

### **Список используемой литературы:**

1. Программирование на Scratch/ Д.В. Голиков, А.В Голиков; 2014.
2. ТРИЗ и алгоритмы мышления/А.В.Кислов; КТК Галактика, 2023. – 336 с.  
(общая педагогика).