

Муниципальное общеобразовательное учреждение  
Иркутского районного муниципального образования  
«Бутырская средняя общеобразовательная школа»

**Инструкция к методической разработке  
по программированию в среде Scratch  
«Образовательные карточки Scratch»**

Образовательная область: Информатика

Разработал(а):

Беликова Светлана Викторовна,

педагог дополнительного образования,

1 категории

2022 г.

## 1. Введение: Что такое Scratch и зачем мы его изучаем?

### 1.1. Знакомство с программой Scratch: простой и понятный интерфейс.

- **Рабочая область:** Здесь будет создаваться наш проект – будем видеть спрайты (персонажи), фон и результат работы программы.
- **Панель спрайтов:** Здесь хранятся все наши персонажи – их можно добавлять, удалять, изменять внешний вид.
- **Блоки команд:** Это “кирпичики”, из которых мы будем строить программы. Они разделены на группы по цветам (движение, внешний вид, звук, события и переменные) – каждая группа отвечает за определённую функцию. Мы подробно разберём каждую из них в следующих разделах.
- **Сцена:** Это фон, на котором будут действовать наши спрайты. Мы научимся менять фон и добавлять новые.

Главное – не бояться экспериментировать! Scratch создан для того, чтобы учиться на практике, просто перетаскивая блоки и видя, что получается.

### 1.2. Что мы будем создавать: веселые игры и анимации.

В этом разделе мы научимся применять знания о движении, внешности, звуке, событиях и переменных в Скретч для создания собственных игр и анимаций! Мы не просто будем изучать отдельные блоки, а соединим их воедино, чтобы оживить наши проекты.

Например, мы создадим:

- **Простую игру “поймай предмет”:** Будем управлять персонажем, используя клавиши стрелок, и ловить падающие предметы, меняя их скорость и внешний вид с помощью переменных. Звуки будут сопровождать успешные и неуспешные попытки.
- **Весёлую анимацию с танцующим персонажем:** Персонаж будет менять костюмы (внешность), двигаться под музыку (звук), реагировать на нажатие мыши (события) и менять свои движения в зависимости от переменной (например, счётчик шагов).
- **Мини-игру “лабиринт”:** Персонаж должен пройти по лабиринту, используя заданные события и изменяя свои координаты (движение). В конце лабиринта будет ожидать награда, сопровождающаяся соответствующим звуком и изменением внешности персонажа.

Все эти проекты помогут вам закрепить пройденный материал и почувствовать себя настоящими разработчиками игр! Вы сможете проявить свою фантазию и создать уникальные игры и анимации, используя изученные блоки Скретч.

## **2. Блок “Движение”: Учим нашего спрайта ходить, прыгать и танцевать!**

### **2.1. Основные блоки движения: шаг вперед, поворот, скольжение.**

**Что мы будем делать:** Научимся управлять движением нашего спрайта (персонажа) в Scratch, используя три основных блока: “Шаг вперед”, “Повернуть” и “Скользить”.

**Блок “Шаг вперед”:** Этот блок заставляет спрайта сделать шаг на определённое расстояние. Попробуйте разные значения расстояния – на сколько шагов ваш спрайт передвинется? Экспериментируйте!

**Блок “Повернуть”:** Этот блок поворачивает спрайта на заданный угол влево или вправо. Попробуйте повернуть спрайта на 90 градусов, 180 градусов, 360 градусов. Что происходит? Можно ли повернуть спрайта “назад”?

**Блок “Скользить”:** Этот блок плавно перемещает спрайта в указанную точку на сцене за определённое время. Попробуйте переместить спрайта в разные углы сцены, меняя время скольжения. Как меняется скорость движения спрайта?

#### **Задания:**

1. **Простая прогулка:** Создайте скрипт, который заставит спрайта сделать 10 шагов вперед, повернуться на 90 градусов вправо, и снова сделать 10 шагов вперед.
2. **Квадрат:** Напишите скрипт, который заставит спрайта нарисовать квадрат, используя блоки “Шаг вперед”, “Повернуть” (на 90 градусов) и повторы (цикл).
3. **Гонки:** Создайте двух спрайтов и напишите скрипты, которые заставят их “соревноваться” в скорости, используя блоки “Шаг вперед” и “Скользить”. Какой спрайт быстрее? Попробуйте изменить значения в блоках, чтобы изменить скорость.

**Дополнительное задание (для самых смелых):** Создайте лабиринт и напишите скрипт, который заставит спрайта пройти его, используя только блоки “Шаг вперед”, “Повернуть” и условные операторы (если...то...иначе).

### **2.2. Практическое задание 1: Создание анимации бегущего персонажа.**

**Цель:** Закрепить знания о блоках движения, внешности, и переменных в среде программирования Scratch путём создания анимации бегущего персонажа.

**Задание:** Создайте в Scratch анимацию бегущего персонажа. Персонаж должен:

- **Движение:** Бежать по сцене слева направо (или справа налево). Используйте блоки движения для перемещения персонажа. Движение должно быть плавным, можно использовать циклы для повторения анимации.

- **Внешность:** У персонажа должен быть изменяющийся внешний вид во время бега (например, смена спрайтов, изменение костюмов, или эффекты). Используйте блоки внешности для достижения этого эффекта.
- **Переменные:** Создайте переменную, отслеживающую расстояние, пройденное персонажем. Эта переменная должна изменяться в зависимости от пройденного расстояния. Выведите значение переменной на сцену (можно использовать блок “показать”).

#### **Дополнительные возможности (необязательно):**

- Добавить звуковое сопровождение бегу. (Блоки звука)
- Создать препятствия, которые персонаж должен преодолевать. (Блоки движения, события)
- Добавить счётчик времени бега. (Блоки события, переменные)

#### **Критерии оценки:**

- Корректность использования блоков движения для создания анимации бега.
- Эффективное использование блоков внешности для изменения внешнего вида персонажа.
- Правильное использование переменных для отслеживания пройденного расстояния.
- Общее качество и оригинальность анимации.

**Время выполнения:** [Указать время, например, 20-25 минут].

### **2.3. Более сложные движения: изменение скорости, использование координат.**

В предыдущих заданиях мы научились управлять движением спрайтов с помощью простых команд. Теперь давайте сделаем движение более интересным! Мы научимся изменять скорость движения спрайта и управлять его положением на сцене с помощью координат.

**Изменение скорости:** В Scratch есть блоки, позволяющие задать скорость движения. Мы можем сделать спрайта быстрым или медленным, плавно менять его скорость во время движения. Попробуем создать сценарий, где спрайт сначала медленно ползёт, а потом резко ускоряется.

**Использование координат:** Каждый спрайт на сцене имеет свои координаты (X и Y). Координата X показывает горизонтальное положение, а координата Y – вертикальное. С помощью блоков координат мы можем точно указать, куда нужно переместить спрайта. Например, мы можем запрограммировать спрайта так, чтобы он двигался по заданной траектории, используя конкретные значения X и Y. Попробуем создать сценарий, где спрайт перемещается между несколькими точками с заданными координатами.

## 2.4. Практическое задание 2: Создание лабиринта с движущимся спрайтом.

**Цель:** Закрепить знания о блоках движения, внешности, событиях и переменных в Scratch, создав интерактивный лабиринт.

**Задание:** Создайте в Scratch лабиринт с помощью фонов и спрайтов. Один спрайт будет представлять собой персонажа, который должен пройти лабиринт.

### Необходимые элементы:

- **Фон:** Создайте или выберите фон, изображающий лабиринт. Можно использовать готовые изображения или нарисовать свой собственный.
- **Спрайт:** Выберите или создайте спрайт, который будет двигаться по лабиринту.
- **Блоки движения:** Используйте блоки движения (например, “переместиться на X шагов”, “повернуть на ... градусов”, “перейти в X: Y:”) для управления движением спрайта.
- **Блоки внешности:** Можно использовать блоки внешности для изменения внешнего вида спрайта (например, изменение костюмов для анимации движения).
- **Блоки событий:** Используйте блок “когда флаг нажат” для запуска игры и, возможно, другие события (например, “когда спрайт касается края”).
- **Переменные (опционально):** Для более сложного варианта задания можно использовать переменные для отслеживания прогресса персонажа в лабиринте (например, количество пройденных шагов или количество столкновений со стенами). Это позволит добавить счетчик или индикатор успеха.

### Дополнительные сложности (для более продвинутых учеников):

- Добавить таймер, отслеживающий время прохождения лабиринта.
- Добавить препятствия (например, другие спрайты), с которыми спрайт-персонаж не должен сталкиваться.
- Создать систему подсчета очков или жизней.
- Использовать звуковые эффекты для обозначения столкновений или завершения игры.

### Инструкция для учителя:

- Разделите учеников на пары или небольшие группы для совместной работы.
- Предоставьте ученикам время на экспериментирование с различными блоками и на создание собственных лабиринтов.
- Помогите ученикам, если они столкнутся с трудностями.
- Продемонстрируйте примеры использования переменных и звуковых эффектов для более сложных вариантов задания.

- Организуйте презентацию созданных лабиринтов и обсуждение решений.

Это задание позволит ученикам не только закрепить пройденный материал, но и проявить креативность и навыки решения задач.

### **3. Блок “Внешность”: Меняем внешний вид нашего спрайта**

#### **3.1. Выбор костюмов для спрайта: изменение внешнего вида.**

В Скретче каждый спрайт может иметь несколько костюмов – это как разные картинки для одного и того же персонажа. Благодаря костюмам вы можете легко менять внешний вид вашего спрайта во время выполнения программы. Например, если у вас есть спрайт-кот, вы можете создать для него костюм с улыбающимся лицом, а другой – с грустным. В программе вы будете переключаться между этими костюмами, чтобы показать изменение настроения кота. Для этого в редакторе Скретча найдите вкладку “Костюмы” и загрузите или нарисуйте новые костюмы для вашего спрайта. Затем, используя блоки программирования, вы научитесь управлять переключением между этими костюмами, создавая анимацию!

#### **3.2. Изменение размера и эффектов: масштабирование, поворот, цвет.**

Этот раздел посвящен практическим заданиям, направленным на освоение инструментов изменения внешнего вида спрайтов в Scratch. Учащиеся научатся изменять размер спрайта, используя блок “изменить размер на ...”, и понимать разницу между увеличением и уменьшением. Далее, задания будут ориентированы на использование блока “повернуть ... градусов” для вращения спрайта и создание анимации с помощью последовательных поворотов. Наконец, дети научатся менять цвет спрайта, используя блоки изменения цвета и эффектов, таких как “изменить эффект [цвет] на ...” и “сбросить эффекты”. Примеры заданий могут включать создание анимации растущего и вращающегося цветка или создание эффекта “дыхания” с изменением размера и цвета персонажа.

#### **3.3. Практическое задание 3: Создание анимации смены костюмов.**

**Цель:** Закрепить навыки работы с костюмами спрайта в среде Scratch для создания простой анимации. Учащиеся научатся использовать блок “Когда флажок нажат” и блок “Следующий костюм” для управления сменой костюмов и создадут анимацию, демонстрирующую изменение внешнего вида персонажа.

**Задание:** Создайте анимацию, в которой спрайт (например, кошка Scratch) изменяет свои костюмы с определенной периодичностью, создавая эффект движения или изменения состояния.

#### **Необходимые блоки:**

- **События:** “Когда флажок нажат”
- **Внешность:** “Следующий костюм” (можно использовать и “Переключить в костюм [номер костюма]”)
- **Управление:** (по желанию) “Подождать (секунды)” для регулировки скорости смены костюмов.

### **Пример:**

Пусть у кошки Scratch есть три костюма: кошка стоит, кошка идёт, кошка прыгает. Задача – создать анимацию, где кошка последовательно меняет эти костюмы, создавая иллюзию движения.

### **Алгоритм:**

1. Выберите спрайт (например, кошку Scratch) и добавьте ему необходимые костюмы (минимум 2).
2. В области скриптов выберите блок “События” → “Когда флажок нажат”.
3. Подключите к нему блок “Внешность” → “Следующий костюм”.
4. (Дополнительно) Для регулировки скорости смены костюмов, добавьте блок “Управление” → “Подождать (секунды)” между блоками “Следующий костюм”. Поэкспериментируйте с разными значениями времени ожидания.

### **Критерии оценки:**

- Анимация работает корректно: костюмы спрайта меняются.
- Анимация имеет логическую последовательность.
- (Дополнительно) Скорость смены костюмов подобрана оптимально.

### **Варианты усложнения:**

- Добавить звуковые эффекты к смене костюмов (использование блоков “Звуки”).
- Добавить другие события, запускающие смену костюмов (например, нажатие клавиши).
- Использовать несколько спрайтов с синхронной или асинхронной сменой костюмов.
- Создать анимацию, рассказывающую простую историю.

Это задание помогает детям понять, как использовать костюмы для создания динамики в своих проектах Scratch, связывая блок “Внешность” с блоком “События” и, опционально, “Управление”.

## **4. Блок “Звук”: Добавляем звуковые эффекты**

### **4.1. Выбор звуков: добавление готовых звуков.**

В этом разделе мы научимся добавлять готовые звуки в наши Scratch-проекты. Рассмотрим, как найти нужный звук в библиотеке звуков Scratch и как добавить его в спрайт.

#### **4.2. Изменение громкости и длительности звуков.**

Этот раздел посвящен практическому применению звуковых эффектов в Scratch, а именно изменению их громкости и длительности. Умение управлять этими параметрами позволяет создавать более выразительные и динамичные проекты.

##### **Практические задания:**

1. **Регулировка громкости:** Создайте спрайт, который издает какой-либо звук. Используя блок “изменить громкость на [число]”, заставьте спрайт постепенно увеличивать и уменьшать громкость своего звука, имитируя, например, приближение и удаление объекта. Экспериментируйте с различными значениями, чтобы понять, как изменение числа влияет на громкость.
2. **Изменение длительности:** Создайте мелодию из нескольких нот с помощью блоков “играть звук [звук]”. Затем, используя блоки “ждать [число] сек” между воспроизведением отдельных нот, измените длительность между ними. Попробуйте создать эффект замедления или ускорения мелодии.
3. **Комбинированное изменение:** Объедините приемы из первого и второго задания. Создайте спрайт, который издает звук, чей объем и длительность меняются в зависимости от какого-либо события (например, нажатия клавиши или перемещения мыши).

##### **Дополнительные задания (для более продвинутых учеников):**

- Создайте программу, которая генерирует случайную громкость и длительность для игаемого звука при каждом нажатии пробела.
- Используйте переменные для хранения значений громкости и длительности, позволяя пользователю управлять ими в реальном времени.

Эти задания помогут учащимся освоить основные приемы работы со звуком в Scratch и применять их для создания интерактивных и интересных проектов.

#### **4.3. Практическое задание 4: Добавление звуков к анимации бегущего персонажа.**

#### **4.3. Практическое задание 4: Добавление звуков к анимации бегущего персонажа.**

Создайте в Scratch анимацию бегущего персонажа (например, кота, собачки или собственного персонажа). Персонаж должен двигаться по сцене слева направо. Добавьте к анимации звуковые эффекты, которые будут воспроизводиться во время бега персонажа. Например, это могут быть звуки шагов, дыхания или другие подходящие звуки.

Экспериментируйте с разными звуками и их громкостью, чтобы добиться наилучшего эффекта. Обратите внимание на синхронизацию звуков с анимацией бега.

## **5. Блок “События”: Запускаем наши программы!**

### **5.1. Блок “Когда флажок нажат”: начало программы.**

Блок “Когда флажок нажат” — это специальный блок в Scratch, который запускает программу. Он является стартовым блоком и позволяет начать выполнение всех последующих команд только после того, как пользователь нажмет на зеленый флажок в правом верхнем углу рабочей области Scratch. Без этого блока, ни одна из написанных вами команд не выполнится. Этот блок очень важен, потому что он определяет момент начала работы вашей программы, связанной с движением спрайта, изменением внешности, воспроизведением звука, использованием переменных и событий.

### **5.2. Другие события: нажатие клавиши, столкновение со спрайтом.**

В Scratch, помимо событий, запускающихся при старте проекта (зелёный флажок), есть множество других событий, которые реагируют на действия пользователя или взаимодействие спрайтов. Сегодня мы рассмотрим два из них:

**1. Нажатие клавиши:** Это событие запускает скрипт, когда пользователь нажимает определённую клавишу на клавиатуре.

- **Как использовать:** В палитре “События” найдите блок “Когда нажата клавиша [пробел]”. Вы можете выбрать любую клавишу, заменив “[пробел]” на нужную. После этого блока вы можете добавить любые действия, которые должны произойти при нажатии этой клавиши. Например, можно заставить спрайта прыгнуть, изменить его костюм или сказать что-то.
- **Пример:** Создадим скрипт, который заставляет кота двигаться вправо при нажатии стрелки вправо, и влево при нажатии стрелки влево.

**2. Столкновение со спрайтом:** Это событие запускает скрипт, когда один спрайт соприкасается с другим.

- **Как использовать:** В палитре “События” найдите блок “Когда [кошка] касается [мышь]”. Здесь “[кошка]” и “[мышь]” - это примеры имён спрайтов. Вы можете выбрать любые спрайты из вашего проекта. После этого блока вы можете добавить действия, которые должны произойти при столкновении. Например, один спрайт может исчезнуть, изменить свой размер или цвет, а другой – начать двигаться в другом направлении.

- **Пример:** Создадим двух спрайтов – кота и мышку. При столкновении кота с мышкой, кошка будет говорить “Я поймал тебя!”, а мышка исчезнет.

Эти два типа событий открывают широкие возможности для создания интерактивных игр и анимаций в Scratch. Поэкспериментируйте с ними, комбинируя их с другими блоками и создавая собственные уникальные проекты!

### **5.3. Практическое задание 5: Создание интерактивной игры с использованием событий.**

**Цель:** Закрепить понимание работы событий в Scratch и научиться создавать интерактивную игру, используя события для управления игровыми объектами и действиями.

**Задание:** Создайте простую игру, например, “ловлю рыбок”, “собираю яблоки” или “стрельбу по мишеням”. В игре обязательно должны присутствовать:

- **События:** Используйте событие “Когда флажок нажат” для начала игры. Для управления игровыми объектами (например, движение персонажа, выстрел) используйте события, связанные с нажатием клавиш (стрелки, пробел) или кликом мыши. Включите события, реагирующие на столкновение объектов (например, персонаж касается яблока/рыбки/мишени).
- **Внешность:** Используйте различные спрайты (персонаж, яблоки/рыбки/мишени) с соответствующей внешностью. Измените их внешний вид (костюмы) в зависимости от событий (например, изменение цвета яблока после сбора, исчезновение мишени после попадания).
- **Движение:** Определите движение игрового персонажа (например, плавное перемещение, прыжки) с помощью блоков движения, реагирующих на нажатие клавиш или события. Аналогично, пропишите движение для других объектов игры (например, падающие яблоки, движущиеся мишени).
- **Звук:** Добавьте звуковые эффекты, соответствующие игровым событиям (например, звук сбора яблока, звук выстрела, звук столкновения).
- **Переменные:** Для усложнения (по желанию) можно использовать переменные для подсчёта очков, жизней или времени. Изменения значений переменных должны быть связаны с игровыми событиями (например, при сборе яблока увеличивается счёт очков).

**Критерии оценки:** Работа должна быть выполнена корректно, игра должна быть функциональна и содержать все перечисленные элементы (события, внешность, движение,

звук, и по желанию - переменные). Оценивается логика построения игры, использование различных событий и качество оформления.

## **6. Блок “Переменные”: Сохраняем и используем данные**

### **6.1. Что такое переменная: хранение информации.**

В программировании Скретч переменные – это как специальные контейнеры для хранения информации. Представьте себе коробочки, в которые можно положить числа, слова или даже логические значения (правда/ложь). Эти коробочки имеют имена, по которым мы к ним обращаемся.

Например, мы можем создать переменную с именем “счёт”, и положить в неё число очков игрока. Каждый раз, когда игрок набирает очки, мы увеличиваем значение в этой переменной. Или переменная “имя” может хранить имя персонажа в игре.

Важно понимать:

- **Имя переменной:** Выбирайте понятные имена, отражающие хранимую информацию (например, “скорость”, “жизни”, “цвет”).
- **Тип данных:** Хотя в Скретч не нужно явно указывать тип данных (число, текст), переменная может хранить только один тип данных за раз. Нельзя в одной переменной хранить одновременно число и текст.
- **Изменение значения:** Значение переменной можно изменять в процессе работы программы. Это позволяет создавать динамичные и интерактивные игры.

В Скретч переменные создаются в разделе “Переменные” и используются в блоках скриптов для чтения и изменения их значений. Например, блок “изменить [переменная v] на [1]” увеличивает значение переменной на 1, а блок “показать [переменная v]” отображает текущее значение переменной на экране. Правильное использование переменных делает программы более гибкими и управляемыми.

### **6.2. Создание и использование переменных в программах.**

В Скретч мы можем хранить информацию, которую используем в наших программах, в специальных контейнерах — переменных. Представьте переменную как ящичек с этикеткой, на которой написано имя. В этот ящичек мы можем положить число, текст или другие данные. Зачем это нужно? Переменные позволяют делать наши программы более гибкими и интересными!

#### **Как создать переменную:**

1. Найдите блок “Переменные” в палитре блоков.

2. Нажмите “Создать переменную...”.
3. Придумайте понятное имя для вашей переменной (например, “счёт”, “очки”, “размер”). Важно выбирать имена, которые описывают, что хранится в переменной.
4. Выберите, будет ли переменная доступна для всех спрайтов или только для одного.

#### **Как использовать переменную:**

- **Присвоение значения:** Используйте блок “присвоить [значение] переменной [имя переменной]”, чтобы положить данные в вашу переменную. Например, “присвоить 0 переменной счёт” установит значение переменной “счёт” равным нулю.
- **Изменение значения:** Вы можете изменять значение переменной в процессе работы программы, например, прибавляя к ней единицу каждый раз, когда происходит какое-то событие. Это можно сделать с помощью блоков математических операций.
- **Вывод значения на экран:** Используйте блок “отобразить [переменная]”, чтобы показать значение переменной на сцене. Это поможет вам отслеживать, как меняются данные в вашей программе.

#### **Примеры использования переменных:**

- **Счётчик очков в игре:** Переменная “очки” будет увеличиваться каждый раз, когда игрок выполняет какое-то действие.
- **Изменение размера спрайта:** Переменная “размер” будет определять размер спрайта, который будет меняться в зависимости от событий в игре.
- **Сохранение имени игрока:** Переменная “имя” будет хранить имя игрока, введённое в начале игры.

Использование переменных делает ваши программы в Скретч намного мощнее и позволяет создавать действительно интересные интерактивные истории и игры!

### **6.3. Практическое задание 6: Создание игры с подсчетом очков с использованием переменных.**

Создайте игру в Scratch, где игрок управляет спрайтом (например, кошкой), собирающим объекты (например, рыбок). За каждый собранный объект игрок получает одно очко. Используйте переменную для подсчёта очков. На экране должна отображаться текущая сумма очков. Игра заканчивается, когда игрок собирает определенное количество объектов (например, 10 рыбок) или проходит отведённое время.

#### **Обязательные элементы:**

- **Спрайт игрока:** Спрайт, управляемый стрелками или клавишами WASD.

- **Спрайты-объекты:** Несколько спрайтов, которые игрок должен собирать.
- **Переменная “Очки”:** Переменная, отображающаяся на экране, которая увеличивается при сборе каждого объекта.
- **Условие окончания игры:** Либо достижение определенного количества очков, либо ограничение по времени (используя блок “ждать” и таймер).
- **Обработка столкновений:** Используйте блоки “если <касается > тогда” для увеличения счетчика очков при столкновении игрока с объектом. После столкновения объект должен исчезать или перемещаться в другое место.
- **Звуковое сопровождение (необязательно):** Добавьте звуки для сбора объектов и окончания игры.

**Дополнительные возможности (для более продвинутых учеников):**

- Добавить различные типы объектов с разным количеством очков.
- Добавить препятствия, которые уменьшают количество очков при столкновении.
- Создать таймер обратного отсчёта.
- Добавить начальный экран с описанием игры и кнопкой “Старт”.
- Сохранение рекорда лучшего результата с помощью переменной.