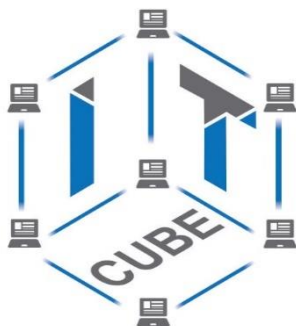


Государственное бюджетное учреждение
дополнительного образования
«Центр дополнительного образования
Липецкой области»



ЦЕНТР ЦИФРОВОГО ОБРАЗОВАНИЯ ДЕТЕЙ
«IT-КУБ»

Методика преподавания программирования на Scratch в системе дополнительного образования детей

Автор:

Трошина Виктория Николаевна,
педагог дополнительного образования

Липецк, 2024

СОДЕРЖАНИЕ

Пояснительная записка	3
1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИМЕНЕНИЯ SCRATCH	5
1.1. Рекомендации по использованию среды Scratch.	5
Технические требования.....	5
1.2. Установка Scratch.....	5
2. СПЕЦИФИКА ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ В СРЕДЕ ПРОГРАМИРОВАНИЯ SCRATCH	13
3. РАЗРАБОТКИ ЗАНЯТИЙ ПО ОСВОЕНИЮ НАВЫКОВ ПРОГРАММИРОВАНИЯ В СРЕДЕ SCRATCH.....	17
Тема № 1 «Знакомство со Scratch».....	17
Тема № 2 «Сенсоры»	21
Тема № 3 «Передача сообщений».....	27
Тема № 4 «Таймер. Случайные числа».....	31
Тема № 5 «Переменные»	37
Список литературы	42

Пояснительная записка

Наш мир очень сильно развивается в IT индустрии, технологии растут, человечество становится информационным, что говорит о наступлении новой фазы развития человечества, где человек думает, а машина работает. В наши дни почти всё является в той или иной степени результатом программирования. Одна из главных проблем общества, что не так много людей занимается программированием, а последующие 10 лет рабочие места в ITсфере будут только увеличиваться, поэтому задача современной школы – обеспечить вхождение обучающихся в информационное общество, научить каждого пользоваться информационно-коммуникационными технологиями.

Для организации учебной и внеурочной деятельности внимание сосредоточено на развитие у учащихся универсальных учебных действий, путем выбора содержания, методов и средств обучения, которые охватывают познавательные процессы (самостоятельность мышления, способы нахождения и решения проблемы), развитие качеств личности, умение прогнозировать и оценивать результаты своей деятельности, умение взаимодействовать со сверстниками и взрослыми.

Научно-познавательная деятельность ученика, организованная в форме выполнения проектов, является наиболее приемлемым методом для формирования универсальных учебных действий. Включение ребенка в проектную деятельность творческого характера позволяет сформировать у него познавательный интерес и исследовательские навыки. А это в свою очередь требует использования определенных средств (инструмента). В качестве такого инструмента идеально подойдет среда программирования Scratch

Во-первых Scratch – это среда программирования для детей в возрасте от 8 до 16 лет, хотя первоначально создавался не для изучения программирования, а для творчества детей. Поэтому он, красив, прост в изучении, имеет мощный инструмент, который позволяет детям создавать собственные анимированные и интерактивные истории, игры и другие произведения. Scratch позволяет проявить ребенку свой личный потенциал, реализовать свои возможности с ориентиром на любую предметную область, т.е. самовыразиться.

Во-вторых, это среда моделирования. Моделирование является универсальным методом познания окружающего мира. Составляя мысленный образ модели ученику, необходимо провести ее анализ и представить с помощью средств некоторого языка своими элементами и связями. Это способствует формированию у ребенка причинно-следственных связей, логического и наглядно-образного мышления.

В-третьих, в Scratch можно проектировать. Для этого среда имеет: редактор кода, компилятор, отладчик, графический редактор для создания и модификации объектов, библиотеку готовых объектов (спрайтов) и звуков, много готовых примеров. Благодаря наличию таких средств программа Scratch «собирается» из разноцветных блоков-команд так же, как собираются из разноцветных кирпичиков в конструкторах Лего различные объекты. Имеется возможность внесения изменений в программу даже тогда, когда она запущена, что позволяет экспериментировать с новыми идеями по ходу решения задачи. В результате выполнения простых команд создаётся сложная модель, в которой взаимодействуют множество объектов, наделенных различными свойствами. Поэтому данную среду можно еще рассматривать и как возможность организации проектной деятельности, как единоличной, так и групповой. Любая проектная деятельность предполагает этапы работы над проектом - это проблематизация, целеполагание, планирование, реализация, рефлексия. Даже самые простые проекты, созданные в Scratch, предусматривают поэтапное их выполнение для достижения цели. Т.е. среда Scratch имеет инструментальные возможности, которые способны формировать у учащихся регулятивные универсальные действия.

В-четвертых, созданные скретч-проекты можно опубликовывать в сети Интернет, а также обмениваться ими внутри международной среды в рамках сообщества Scratch (<http://scratch.mit.edu>). Все участники сообщества могут между собой общаться на форуме, послушать похвалу и критику, найти единомышленников, подчеркнуть новые идеи. Это возможность среды Scratch позволяет формировать у ученика коммуникативные универсальные действия.

Таким образом, перечисленные особенности среды программирования Scratch позволяют формировать у ученика универсальные учебные действия.

Цель данного методического пособия: оказать помощь учащимся и педагогом дополнительного образования по введению в среду программирования Scratch. В методическом пособии представлены теоретические основы применения Scratch, даны рекомендации по использованию Scratch и технические требования по установке и описание среды программирования Scratch, описана специфика проектной деятельности в среде программирования Scratch. Так же пособие содержит серию конспектов занятий и пошаговые инструкции создания проектов. Материалы к занятиям снабжены достаточным количеством иллюстраций и пояснений к ним, дающих возможность обучать детей технологии разработки игровых программ.

Предлагаемый методический материал может быть использован педагогами дополнительного образования и учителями информатики и ИКТ.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИМЕНЕНИЯ SCRATCH

1.1. Рекомендации по использованию среды Scratch.

Технические требования

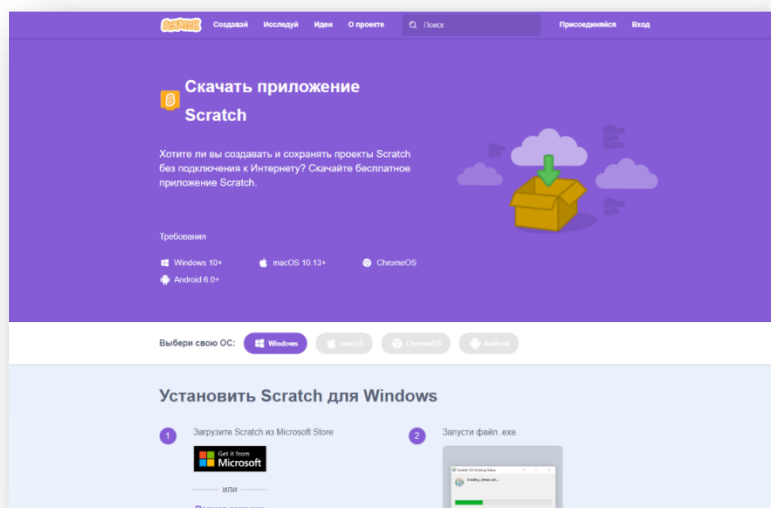
Для организации работы в аудитории потребуется:

- ✓ помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования;
- ✓ качественное освещение;
- ✓ столы, стулья по количеству учащихся и 1 рабочим местом для педагога;
- ✓ специальный шкаф под компьютер и оргтехнику;
- ✓ персональный компьютер;
- ✓ МФУ;
- ✓ доступ к сети Интернет.

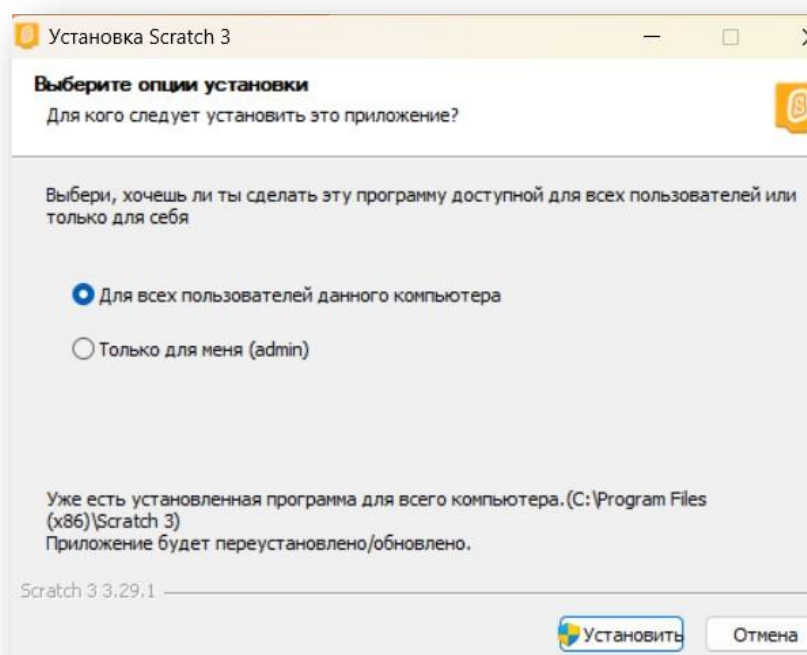
Особых требований к ресурсам компьютера среда программирования Scratch не предъявляет: процессор должен быть минимум типа Pentium или Celeron, свободного пространства на жёстком диске для установки Scratch потребуется 120 Мбайт, разрешение экрана 1024x768, с цветностью 16-бит, любой браузер, операционная система: Windows XP/Vista/7/8/10,11 Mac OS X 10.3 (и выше), Ubuntu Linux. Установка последней версии Adobe AIR и других стандартных драйверов.

1.2. Установка Scratch

Программа Scratch бесплатно скачивается из интернета. Из любого браузера, установленного на вашем компьютере, перейдите по адресу <https://scratch.mit.edu/download> Скачивайте только с официального сайта (рисунок 1), чтобы работать на самой актуальной версии программного обеспечения и остаться в безопасности от заражения вирусами. Если вдруг, по каким либо причинам вы не хотите или не можете установить Scratch на компьютер вы всегда можете программировать онлайн, без установки чего-либо <https://scratch.mit.edu/projects/editor/?tutorial=getStarte>

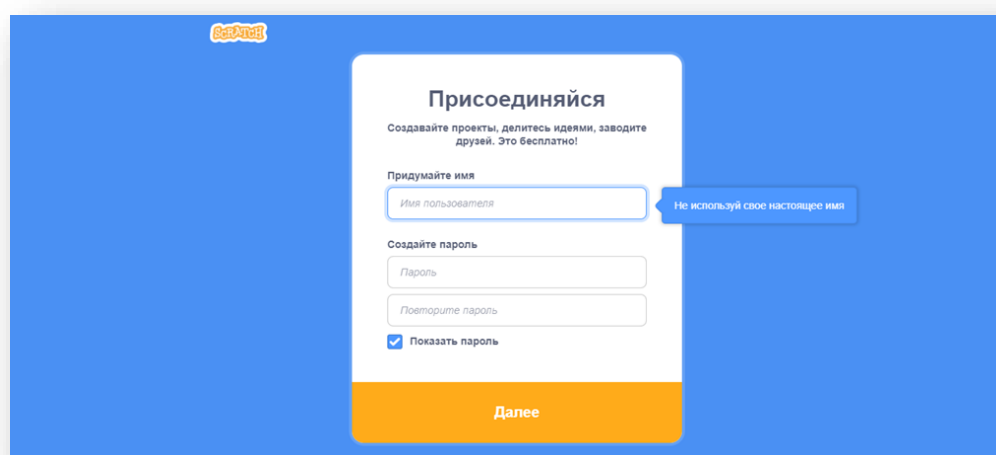


После скачивания дистрибутива установки, запускаем Scratch 3.29.1 Setup (ваша версия может отличаться) и нажимаем кнопку «Установить», после чего ждем завершения.



По умолчанию программа установится в папку ProgramFiles, и будет сохранять ваши проекты в папке «Мои документы», рекомендую создать папку «Проекты Scratch» - и там сохранять созданные вами проекты.

Для онлайн – работы, нужно пройти регистрацию на этом же сайте:



Я рекомендую обеспечить для себя обе возможности. Работать в офлайн вы сможете без подключения к сети Интернет, а пройдя регистрацию на сайте, вы не только получите доступ к онлайн среде разработчика, но и вольетесь в общество скрейчеров – людей, публикующих свои проекты и обсуждающих результаты своей деятельности.

Сообщество носит открытый характер. Любой желающий может посмотреть публикуемые материалы и использовать их.

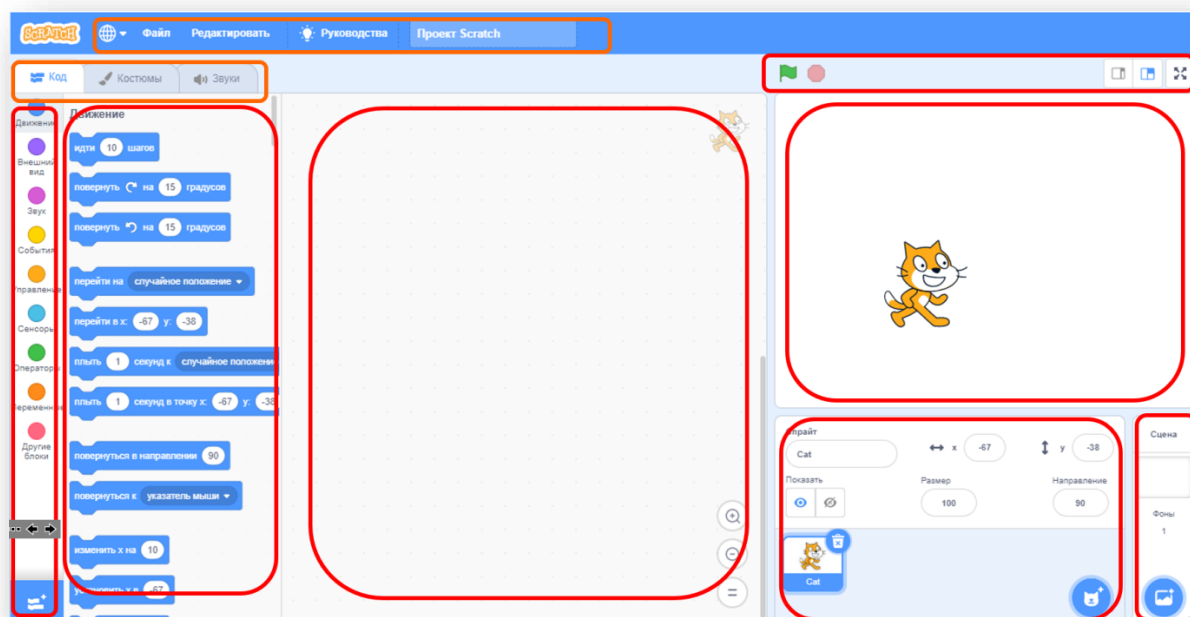
1.3 Описание среды программирования Scratch

Scratch имеет приятный и понятный пользовательский интерфейс. В нем ученики в полной мере могут раскрыть свои творческие таланты, так как в Scratch можно легко создавать игры, презентации, анимированные открытки, мультфильмы, обучающие программы, тренажеры, интерактивные тесты: придумывать и реализовывать различные объекты, определять, как они выглядят в разных условиях, перемещать по экрану, устанавливать связь между объектами; выдумывать истории, рисовать и оживлять на мониторе своих выдуманных персонажей, осваивая при этом технологии обработки графической и звуковой информации, анимационные технологии, мультимедийные технологии.

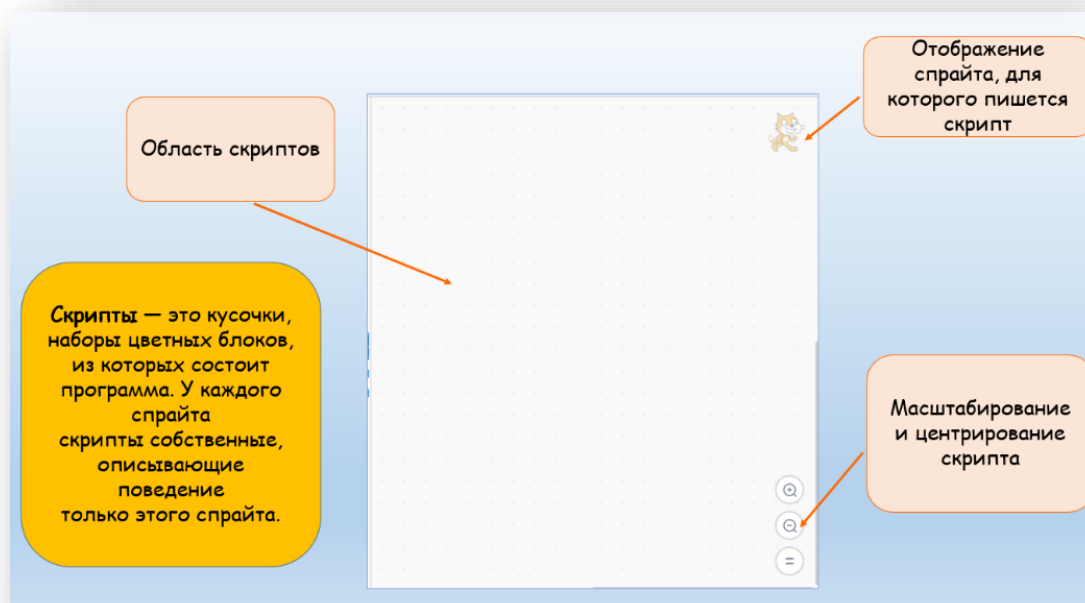
Scratch — это визуально объектно– ориентированная среда, в которой блоки с командами собираются в одно целое, подобно LEGO. Ориентация при работе со Scratch на результаты образования на основе системно-деятельностного подхода, который лежит в основе концепции развития универсальных учебных действий, являющихся основным понятием ФГОС нового поколения и обеспечивающих способность обучающихся к саморазвитию путем сознательного и активного освоения нового опыта.

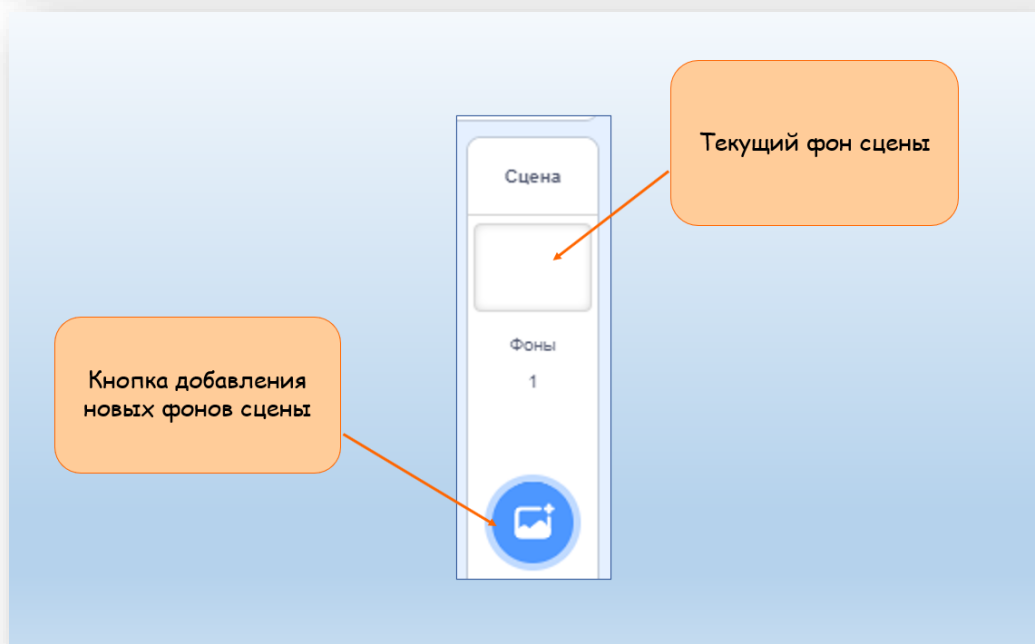
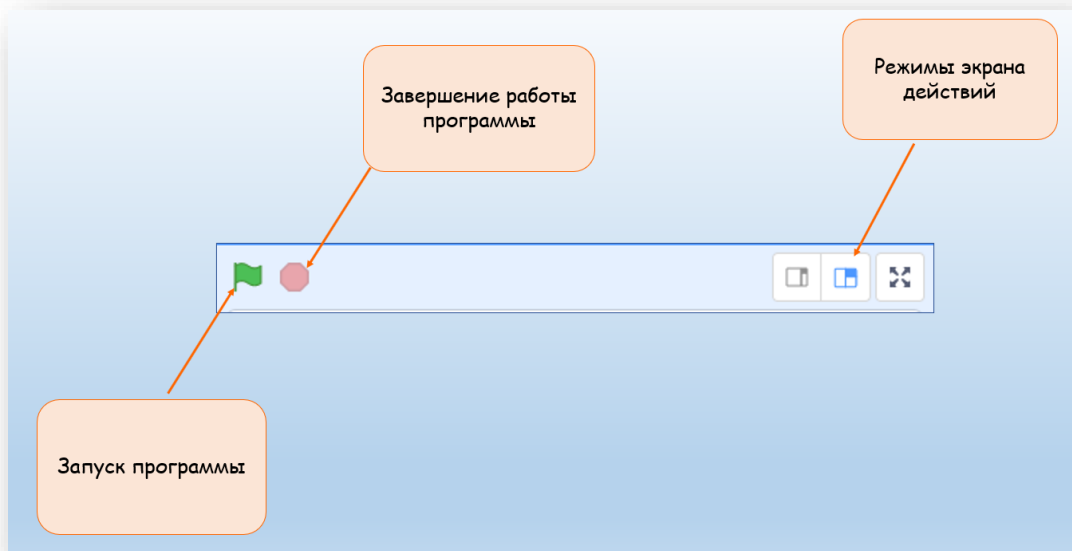
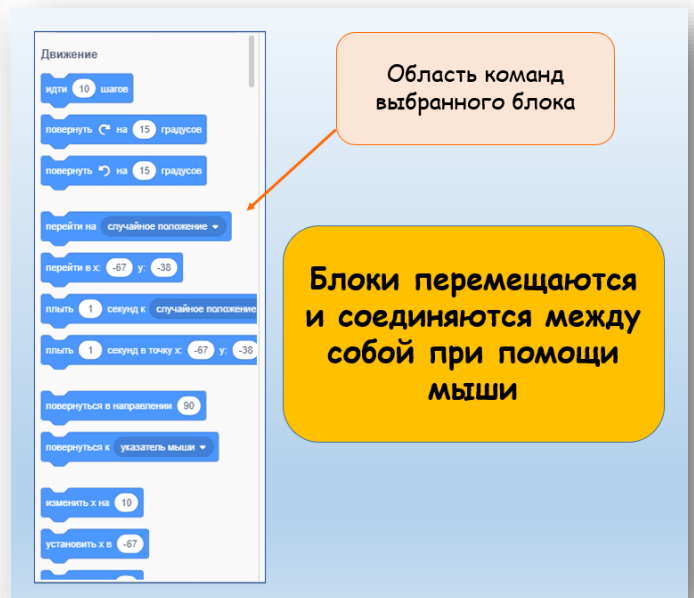
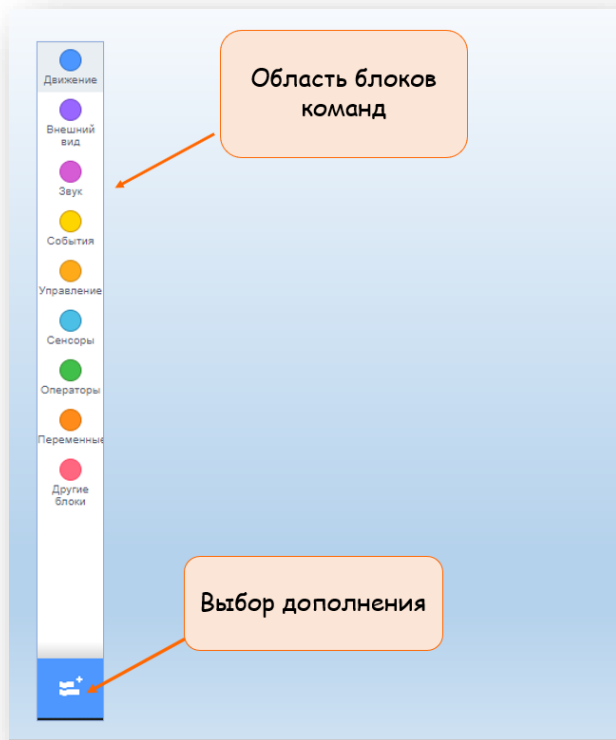
Scratch имеет две версии офлайн и онлайн. Интерфейс десктопной версии и онлайн-сервиса <https://scratch.mit.edu/> абсолютно одинаковые, тем самым учитель, в зависимости от оборудования класса может выбирать.

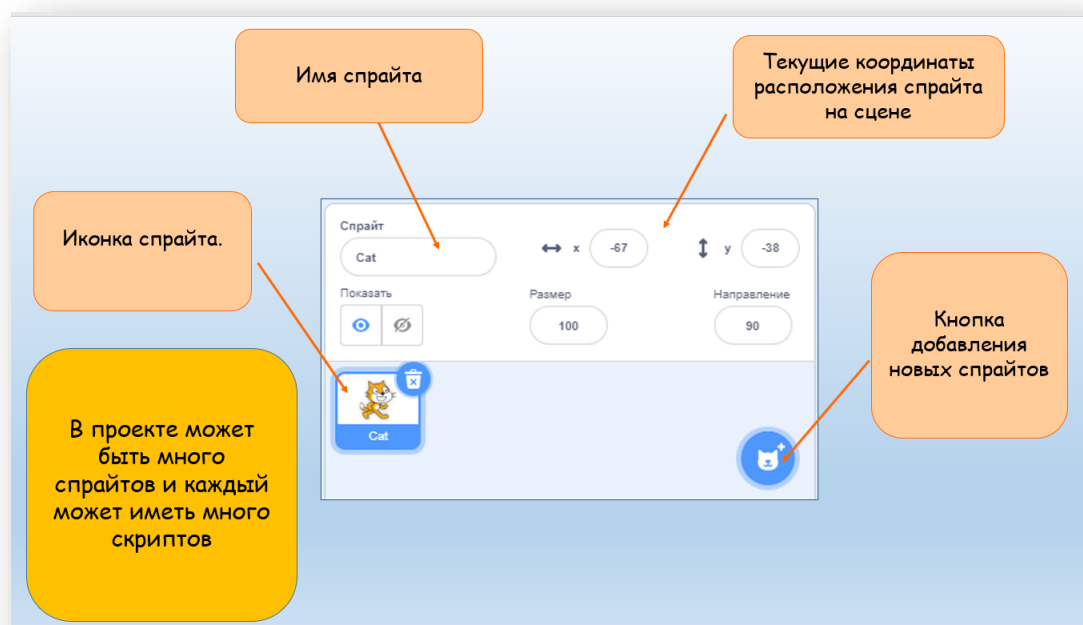
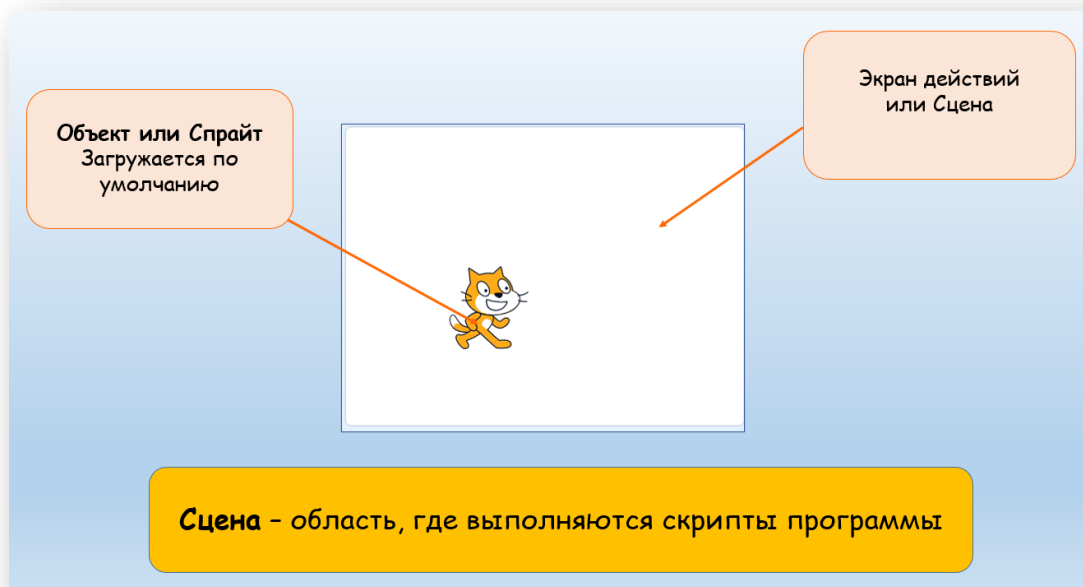
Интерфейс Scratch



Давайте более подробно рассмотрим каждый элемент интерфейса Scratch, чтобы начинающему пользователю сразу стало понятно предназначение каждой кнопки, опции и области.

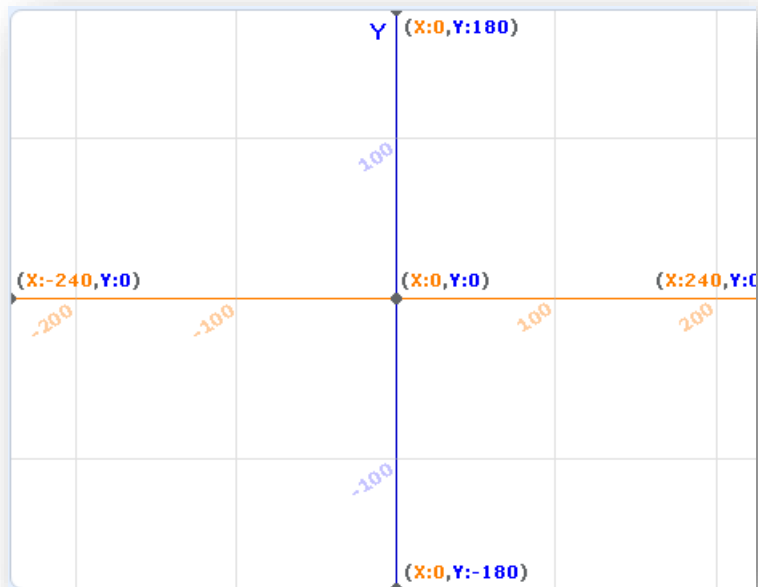






Спрайты и фоны

Основными компонентами программы являются объекты-спрайты. Спрайт состоит из графического представления — набора кадров-костюмов (и сценария-скрипта). Для редактирования костюмов спрайтов в Scratch встроен графический редактор. Действие программы происходит на сцене размером 480×360 (условных) пикселей с центром координат в середине сцены



В проекте может быть много спрайтов, и у каждого спрайта свой скрипт!

Программировать будет легче, если ты дашь спрайтам понятные имена

Переключайся между спрайтами, кликая по ним

Чтобы скопировать спрайт и его скрипты, сделай по нему правый клик и выбери «дублировать». Выбери «удалить», если спрайт не нужен

Выбор изображения с диска

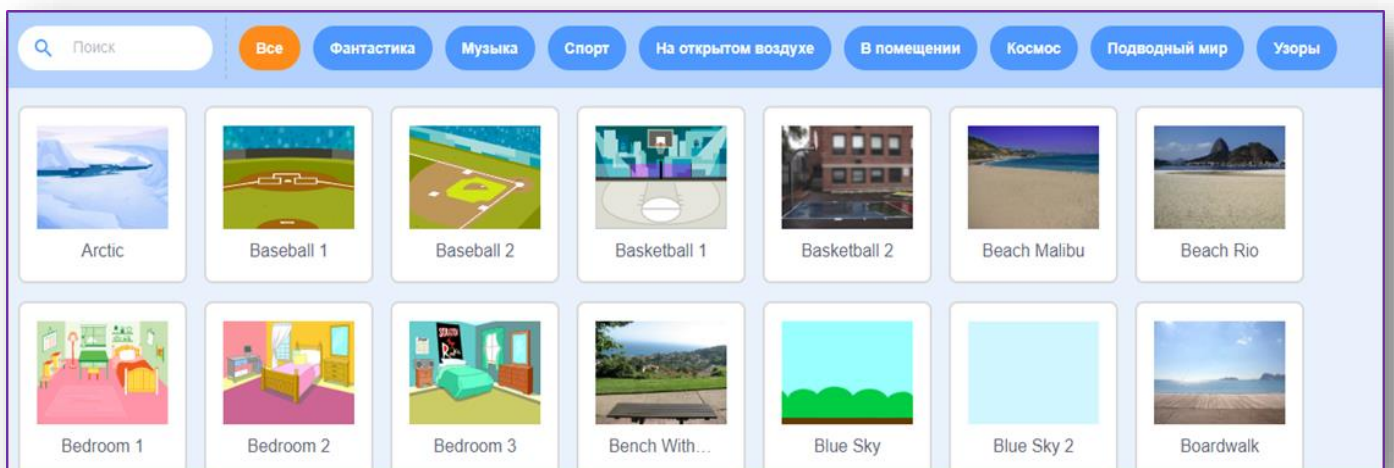
Рандомный спрайт из библиотеки

Создание собственного спрайта

Выбор из библиотеки

Фон — это задний план проектов на Scratch. У него намного меньше функций в отличии от спрайтов, но его также можно редактировать и добавлять новые костюмы.

Автоматически в ваш проект добавляется стандартный белый фон размером 480*360 пикселей.



Аналогично со спрайтами фоны можно добавлять с диска, создавать самостоятельно или выбрать из библиотеки. В Scratch добавлена библиотека фонов, состоящая из 8 категорий. Всего в библиотеке фонов 85 различных костюмов для сцены.

Цветные блоки и скрипты

Поведение каждого объекта в среде программирования Scratch определяется его программой – скриптом.

При создании скриптов используется палитра блоков, расположенная в левой части экрана.

Здесь можно выбрать нужную вам группу команд:

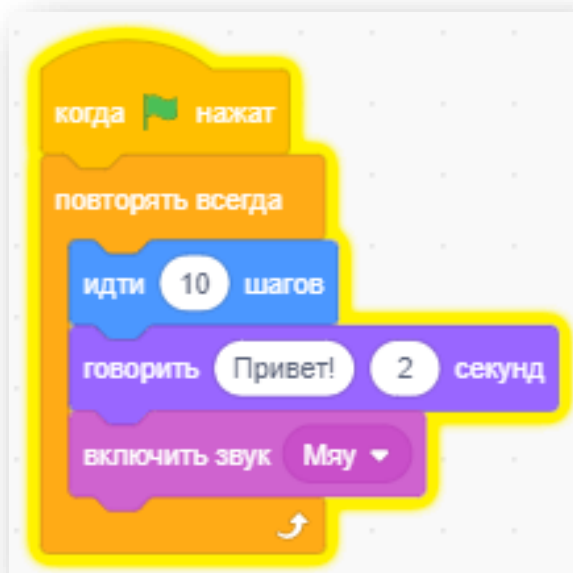
- ✓ *Движение*
- ✓ *Внешний вид*
- ✓ *Звук*
- ✓ *События*
- ✓ *Управление*
- ✓ *Сенсоры*
- ✓ *Операторы*
- ✓ *Переменные*

Блоки раскрашены в разные цвета в зависимости от их назначения. При их соединении получаются скрипты, в которых блоки выполняются по очереди.

Блоки разных типов имеют различные функции в программе. Некоторые из них двигают спрайты, некоторые управляют звуками, некоторые определяют, что должно происходить.

Порядок выполнения скриптов таков: когда программа запущена, Scratch выполняет инструкции блоков. Он делает это, начиная с верхнего блока в скрипте и далее к низу.

Когда скрипт работает, он подсвечивается. Используйте зеленый флажок для запуска или кликайте по верхнему блоку события, чтобы запустить его.



2. СПЕЦИФИКА ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ В СРЕДЕ ПРОГРАМИРОВАНИЯ SCRATCH

Главной целью дополнительного образования на сегодняшний день является формирование и развитие творческих способностей ребенка. На занятиях педагог стремясь создать оптимальные условия для вовлеченности обучающихся в познавательный- творческий процесс использует различные педагогические технологии. Одной из активных форм педагогических технологий является проектная деятельность.

Проектная деятельность обучающихся – это совместная учебно-познавательная деятельность, имеющая общую цель, согласованные методы и способы деятельности, направленные на достижение общего результата.

Познакомившись со средой Scratch, мы понимаем, что она наилучшим образом подходит для проектной деятельности, поскольку даже самый простой детский проект, который выполнен в этой среде, будет иметь свою цель, и по ее достижению, за определенное количество шагов, ребенок предоставит результат, являющийся или его личным продуктом, или частью большого проекта, выполняемого группой.

Несмотря на кажущуюся простоту Scratch является полноценной средой программирования, где результаты действий визуализированы, что делает работу с ней более понятной, увлекательной и интересной.

Перед началом работы над проектом ученику необходимо на своем рабочем месте создать информационное пространство (папку) в которое он будет помещать материалы к занятиям, выдаваемые преподавателем, а также непосредственно сами проекты. Работу можно организовывать как индивидуально, так и разделившись на группы по 3- 4 человека, учитывая при этом желание ребят.

Условно все проекты, выполненные в среде Scratch, можно разделить на следующие типы:

- ✓ *игра;*
- ✓ *анимация;*
- ✓ *музыкальный проект;*
- ✓ *учебная презентация;*
- ✓ *обучающая программа;*
- ✓ *учебная модель, демонстрационный эксперимент.*

Планируя проект на Scratch, важно продумать следующие моменты:

- ✓ название проекта;
- ✓ сложность проекта в зависимости от возраста учащихся, которые его выполняют;

- ✓ предметную область, которой будет посвящен проект;
- ✓ тип проекта по виду деятельности.

Выделяют следующие этапы проектирования на Scratch:

1. Подготовительный этап. На этом этапе происходит постановка цели и составление плана деятельности:

- ✓ выделяются все объекты предстоящего проекта, их свойства и взаимодействия;
- ✓ выделяются отдельные подзадачи и последовательность их выполнения.

На этом этапе очень удобно раздать детям карточки, визуализирующих этапы планирования их проекта:

1.Подумайте, какой проект вы хотели бы сделать?	
2.Создайте общий план вашего проекта, используя любую либо каждую из следующих подсказок:	
Какая область вас интересует? Как бы выглядела ваша идея, если бы вы перенесли её в Scratch?	
Это была бы игра? Рассказ? Загадка? Музыкальное видео? Нечто совершенно иное?	
Какие ресурсы (например, люди, примеры проектов) могут вам понадобиться для создания своего проекта?	
Каков был бы ваш первый шаг?	

--	--

2. Организационный этап:

В групповом проекте происходит распределение ролей по виду деятельности: сценарист, программист, художник, музыкальный редактор; и по подзадачам- каждый учащихся группы разрабатывает свою часть проекта.

В индивидуальном проекте должно четко прослеживаться последовательное выполнение плана деятельности подготовительного этапа.

3. Реализация проекта.

На этом этапе разрабатывается наглядное представление проектов: выбор сцены и спрайтов, рисование новых объектов и создание костюмов. Пишутся скрипты и происходит отладка кода.

4. Этап презентации и оценки результатов проекта.

На этом этапе происходит демонстрация проекта в группе, обсуждается и оценивается проект, делаются выводы по проделанной работе.

При оценивании проекта следует обратить внимание на такие элементы как:

- ✓ художественное оформление;
- ✓ техническая сложность;
- ✓ практическая значимость проекта;
- ✓ продуманность интерфейса, многоуровневость и этапа подведения итогов (для игр);
- ✓ наличие музыкального сопровождения;
- ✓ наличие заставки с указанием творческого названия и автора проекта.

Важно познакомить детей с *алгоритмом творчества Митчелла Резника*, по мнению которого, создание творческих проектов протекает по спирали, виток которой состоит из пяти шагов:

1. **Представь себе будущую программу?** Кто, когда и что будет делать в проекте
2. **Сделай проект.**
3. **Испытай.** Все ли получилось так, как было задумано?
4. **Поделись с другими.** Как оценили твою работу?
5. **Обдумай** их оценки: нужно ли что- то изменить, улучшить или упростить? Что можно сделать еще?

С этапами проектирования необходимо познакомить ребят, начиная со второго занятия. Большую часть времени каждого учебного занятия ученик

работает самостоятельно: индивидуально или в небольшой группе, учитель лишь выступает в роли тьютера, поэтому самостоятельность ребенка надо всячески поощрять: добрым словом, похлопыванием по плечу, демонстрацией готового проекта данного ученика.

На этапе планирования у детей может возникать масса идей, которые в силу нехватки знаний программирования в среде Scratch, они не смогут выполнить. Поэтому своевременно обратить на это внимание и не дать ребенку войти в заблуждение и встать в «тупик», но при этом не предотвратить процесс исследования, который ребята в Scratch выполняют с удовольствием, иначе ребенок потеряет интерес к изучению среды.

Таким образом мы пришли к выводу, что метод проектной деятельности в Scratch, предполагает образовательное определение будущего результата, мотивы его необходимости, последовательное выполнение шагов реализации проекта, которые по мере своего свершения будут расширяться и корректироваться.

3. РАЗРАБОТКИ ЗАНЯТИЙ ПО ОСВОЕНИЮ НАВЫКОВ ПРОГРАММИРОВАНИЯ В СРЕДЕ SCRATCH

Тема № 1 «Знакомство со Scratch»

Цель занятия: Знакомство со средой программирования Scratch.

Задачи занятия:

Образовательные:

- ✓ познакомиться с понятием «язык программирования», «визуальное программирование»;
- ✓ получить представление о преимуществах языка программирования Scratch;
- ✓ познакомиться с интерфейсом программной среды Scratch;
- ✓ познакомиться с основными типами блоков;
- ✓ изучить блоки «Движение», «Внешность», «Управление»;
- ✓ получить представление о структуре программы в среде Scratch;
- ✓ научиться добавлять на сцену новые фоны и спрайты;
- ✓ получить навык управления программами в среде Scratch.

Планируемые результаты:

Предметные:

Знать

- ✓ понятия «язык программирования», «визуальное программирование»;
- ✓ интерфейс среды программирования Scratch;
- ✓ основные типы блоков;
- ✓ структуру программы в среде Scratch;

Уметь

- ✓ переключать язык среды программирования;
- ✓ запускать и останавливать программу;
- ✓ добавлять на сцену новые фоны и спрайты;
- ✓ редактировать программу;
- ✓ сохранять готовый проект;

Метапредметные:

- ✓ учащиеся обретут навыки осуществлять самонаблюдение, самооценку, самоконтроль;

- ✓ научатся планировать последовательность и прогнозировать итоги действий и всей работы в целом, анализировать полученные результаты;

План занятия:

1. Знакомство с понятиями «язык программирования», «визуальное программирование».
2. Знакомство с интерфейсом среды программирования Scratch.
3. Знакомство с основными типами блоков.
4. Команды управления программами.
5. Практическая работа «Знакомство»

Ход занятия.

1. Организационный момент. Приветствие.

2. Знакомство с понятиями «язык программирования», «визуальное программирование».

Педагог проводит беседу с учащимися о языках программирования, рассказывает о преимуществах языка программирования Scratch

3. Знакомство с интерфейсом среды программирования Scratch.

Педагог рассказывает теоретический материал о том, как изменить язык интерфейса на русский, а также о различных элементах окна среды программирования.

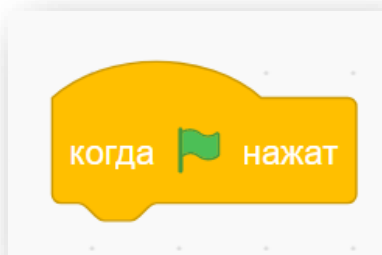
4. Команды управления программами.

Педагог рассказывает о командах запуска и остановки скрипта, а также показывает, как работают данные команды в среде программирования Scratch.

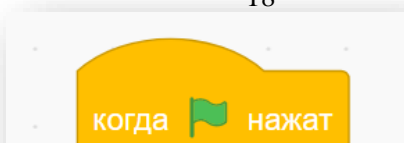
5. Практическая работа «Знакомство».

1) *Перейдите во вкладку скриптов «События» (оранжевого цвета)*

2) *Перетащите на рабочее поле блок «Когда нажат флажок»*



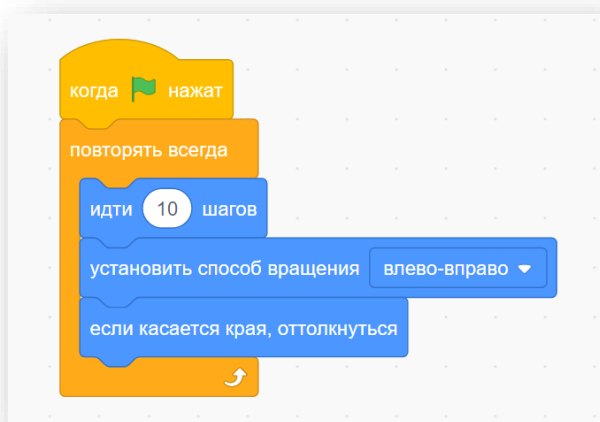
3) *Перейдите во вкладку скриптов «Движение» – перетащите блок «идти 10 шагов» на рабочее поле и присоедините его к блоку «Когда нажат флажок».*



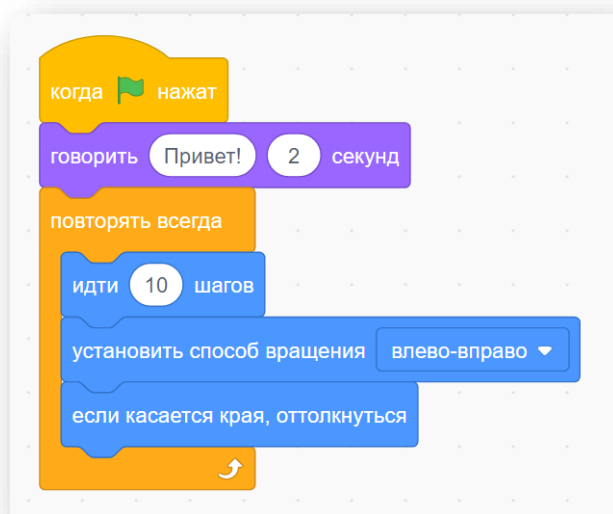
Нажмите на зеленый флажок в верхней части сцены, и спрайт переместится на 10 шагов вправо.

4) Перейдите во вкладку «Управление» и добавьте блок «Повторять всегда» в скрипт. Обратите внимание, что блок «идти 10 шагов» должен находиться внутри него. Добавьте из вкладки «Движение» блоки – «Установить способ вращения...» и «если касается края, оттолкнуться».

Запустите скрипт и посмотрите, что будет происходить на сцене.



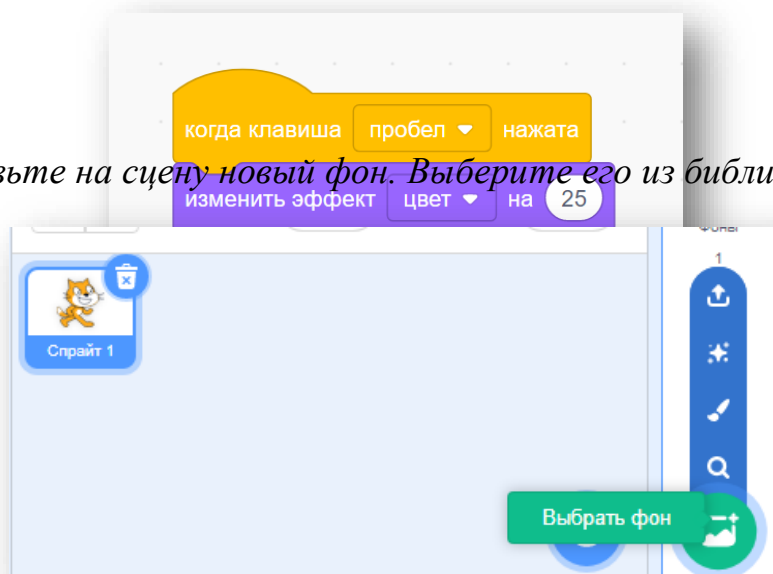
5) Научим Кота говорить. Перетащите на поле из вкладки скриптов «Внешность» блок «Говорить: «Привет!» в течение 2 секунд». Помести этот блок между шапочкой и блоком «Повторять всегда»



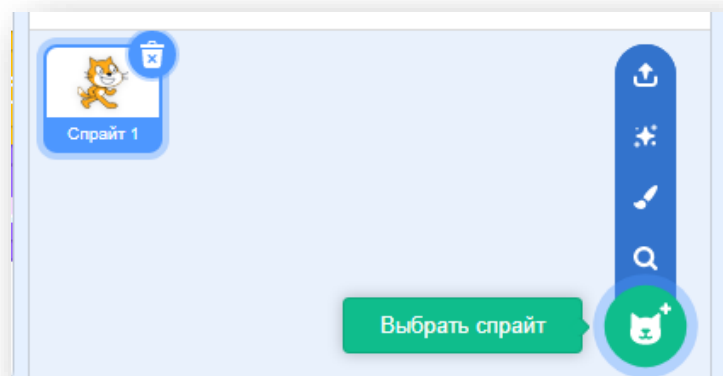
б) Изменим цвет шерсти кота. Перетащим на поле из вкладки скриптов «Внешность» блок «изменить цвет эффект на 25».

Поместим над этим блоком блок «**Когда клавиша пробел нажата**».
Запусти программу флажком. Посмотри, что будет происходить, когда вы нажимаете «Пробел».

7) Добавьте на сцену новый фон. Выберите его из библиотеки.



8) Добавьте на сцену другие спрайты и запрограммируйте изменение их цвета при нажатии на другие клавиши



9) Запрограммируйте изменение фона при нажатии на клавиши вверх или вниз.

6. Подведение итогов занятия.

Методические рекомендации

1. Следует обратить внимание детей на то, что ненужные блоки лучше сразу убирать, перетаскивая их в ту часть окна, где находится область скриптов.
2. В ходе выполнения практической работы ученики самостоятельно более подробно знакомятся с блоками «Движение», «Внешность», «Управление», а также с тем, как добавлять новые спрайты и осуществлять смену фона.

3. Большую часть данной практической работы учащиеся выполняют самостоятельно, изредка прибегая к помощи педагога.

Тема № 2 «Сенсоры»

Цель занятия: Освоение методов создания простейшей анимации с использованием блоков «Сенсоры», «Движение», «Внешность», «Управление».

Задачи занятия:

Образовательные:

- ✓ получить представление об области информации спрайта;
- ✓ познакомиться с циклом «Повторять всегда»;
- ✓ научиться создавать анимацию смены костюмов с использованием команд блока «Внешность» и блока события;
- ✓ познакомиться с командой, позволяющей спрайту оттолкнуться от края при столкновении;
- ✓ изучить блоки «Сенсоры», а также команду «Если-то», с которой используются сенсоры;
- ✓ научиться программировать движение спрайта с помощью команды «Перейти в»;

Планируемые результаты

Предметные:

Знать

- ✓ понятие «область информации», «бесконечный цикл», «условный оператор»;
- ✓ команды блока «Сенсоры»;
- ✓ команды других блоков, с которыми используются сенсоры;

Уметь

- ✓ настраивать спрайт с использованием области информации;
- ✓ создавать простейшую анимацию, используя команды, изученные на данном занятии;

Метапредметные:

- ✓ учащиеся обретут навыки осуществлять самонаблюдение, самооценку, самоконтроль;

- ✓ научатся планировать последовательность и прогнозировать итоги действий и всей работы в целом, анализировать полученные результаты;

План занятия:

1. Знакомство с областью информации спрайта.
2. Знакомство с командами, позволяющими создать анимацию движения и смены внешности у спрайта.
3. Практическая работа «Анимация. Кот говорит, думает, меняет цвет».
4. Знакомство с командами блока «Сенсоры» и командами других блоков, которые используются вместе с ними.
5. Практическая работа «Анимация с сенсорами».
6. Подведение итогов

Ход занятия.

1. Организационный момент. Приветствие

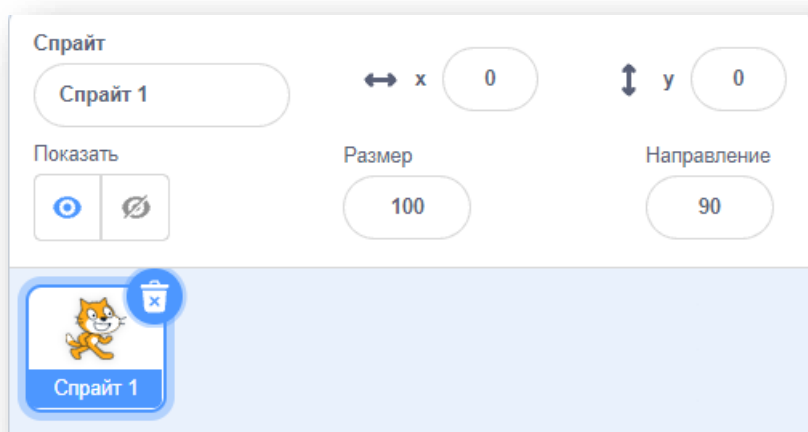
2. Знакомство с областью информации спрайта.

Педагог рассказывает о том, как можно настроить спрайт, обращая особое внимание учеников на стиль вращения спрайта.

3. Знакомство с блоками, позволяющими создать анимацию движения и смены внешности у спрайта.

Педагог обращает внимание учеников на то, что у спрайта может быть несколько костюмов, рассказывает о командах, которые позволяют изменить цвет и костюм спрайта, о командах блока событий, которые позволяют управлять сменой внешности у спрайта, а также о том, как заставить спрайт отталкиваться от края при столкновении с ним и разворачиваться на определенное количество градусов.

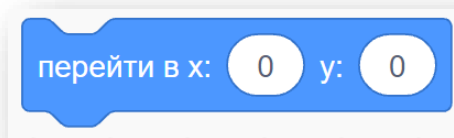
4. Практическая работа «Анимация. Кот говорит, думает, меняет цвет»



- ✓ *Имя спрайта – называйте спрайт осмысленно*

- ✓ *X, Y – координаты текущего положения спрайта*
- ✓ *Направление – текущий угол наклона спрайта*
- ✓ *Размер – изменение размера спрайта*

Блок «Перейти в»:



Блок «**Перейти в**» перемещает спрайт в точку.

Для простоты определения координаты точки перемещения необходимо:

- ✓ переместить спрайт по сцене в то место, куда его нужно будет переместить в скрипте
- ✓ перетащить блок «**Перейти в**» в скрипт

Блок будет содержать именно те координаты, куда переместили спрайт на сцене!

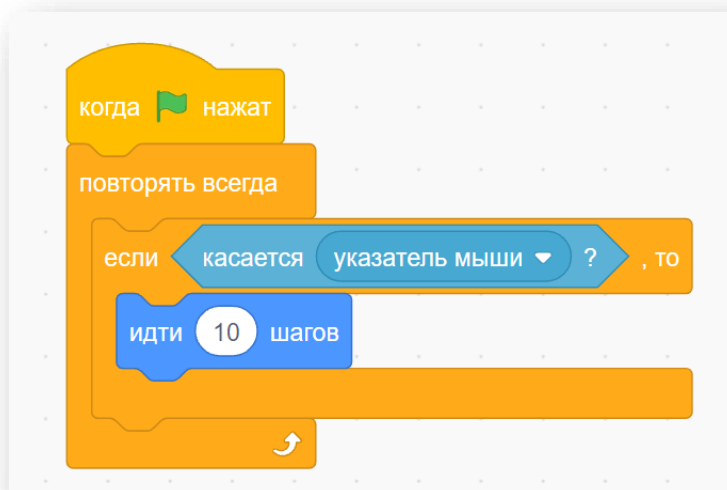
Использование сенсоров

Место для вставки сенсора выглядит как шестиугольник:

Чаще всего сенсоры используются с блоками

ЕСЛИ ТО, ЖДАТЬ ДО, ПОВТОРЯТЬ ПОКА НЕ

Спрайт, убегающий от курсора мыши:



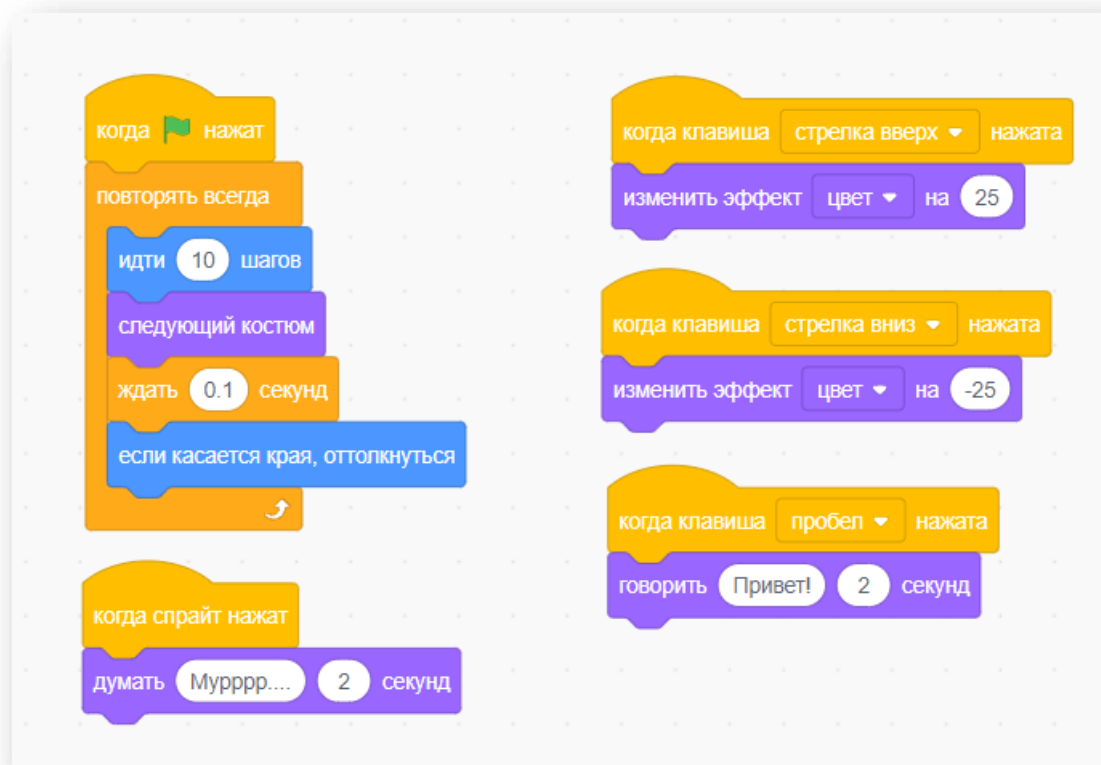
Анимация. Кот говорит, думает, меняет цвет

Бесконечное движение Кота с отражением от стен и сменой костюмов.

Задача:

1. Кот бежит от края до края
2. При нажатии на стрелку вверх Кот меняет цвет или применяется другой эффект

3. При нажатии на стрелку вниз эффект должен отмениться
4. При нажатии на пробел Кот говорит: «Привет!»
5. При щелчке мышью по Коту он мурлычет



5. Знакомство с командами блока «Сенсоры» и командами, которые используются вместе с ними.

Педагог знакомит учащихся с командами, входящими в блок «Сенсоры». Педагог обращает внимание учеников на то, что команда «Если- то» всегда используется вместе с командой «Повторять всегда», в противном случае программа не будет работать. Также ребята знакомятся с еще одним способом создания анимации движения спрайта – командой «Перейти в...»

6. Практическая работа «Анимация с сенсорами».

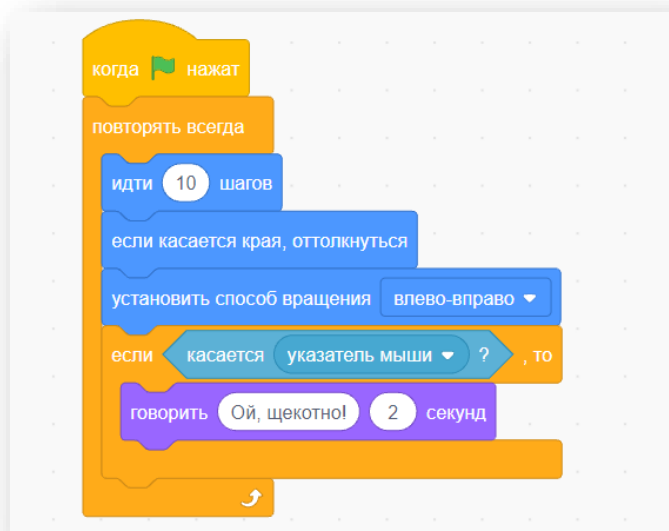
Педагог предлагает ученикам выполнить практические задания на программирование спрайта таким образом, чтобы он изменял внешность или отодвигался от указателя мыши при соприкосновении с ним. Также ребятам предлагается запрограммировать спрайты таким образом, чтобы они разбежались в разные стороны при соприкосновении друг с другом.

Задача 1:

Бесконечное движение Лошадки с отражением от стен и реакцией на наведение мыши.

1) Добавить спрайт Лошадки, которая скачет вперёд-назад

2) При касании указателя мыши Лошадка говорит: «Ой, щекотно!»



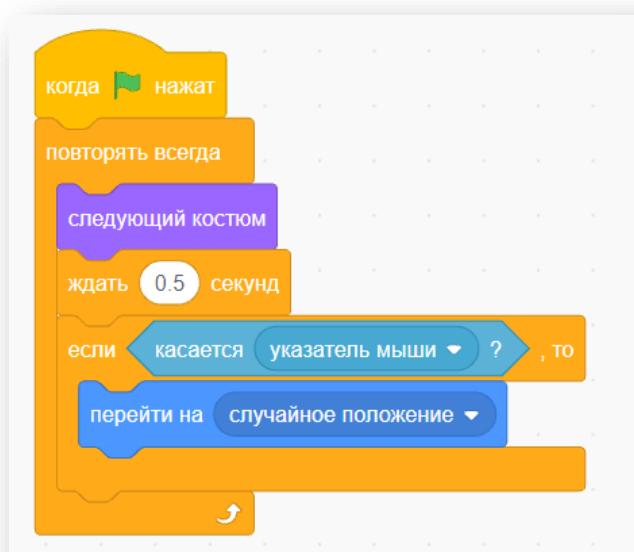
Задача 2:

Реакция Летучей мыши на наведение указателя.

1) Добавить на сцену спрайт Летучая мышь.

2) В цикле «Повторять всегда» назначить ей смену костюмов с ожиданием в 0.5 секунд.

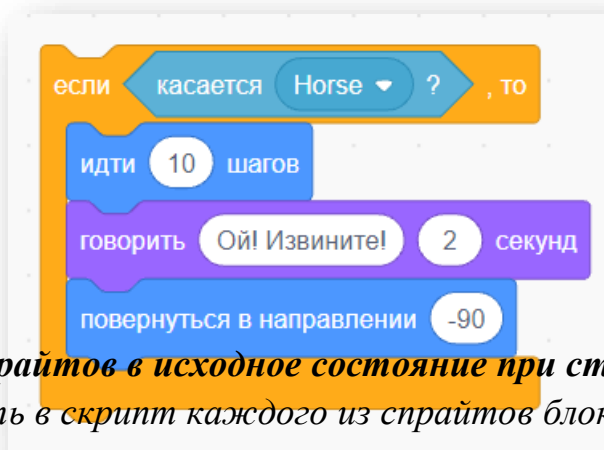
3) Задать скрипт летучей мыши: при касании указателя мыши перемещается на случайное положение на сцене



Задача 3:

Реакция Летучей мыши и Лошадки на столкновение друг с другом.

- 1) При столкновении Летучей мыши и Лошадки они должны говорить: «Ой! Извините» и разбежаться в противоположном направлении.



Для возврата спрайтов в исходное состояние при старте:

- 1) Добавить в скрипт каждого из спрайтов блок «Перейти в...»

7. Подведение итогов занятия.

Методические рекомендации

1. Педагогу следует обратить особое внимание учащихся на то, что команда «Если то» работает только вместе с командой «Повторять всегда».
2. Также следует обратить внимание на то, что для корректной работы программы важно правильно настроить задержку времени, так как если учащиеся забывают добавлять в программу команду «Ждать», программа может выполняться мгновенно и результат не всегда виден. Например, в том случае, когда спрайты при столкновении должны разбежаться в разные стороны.
3. Следует обратить внимание детей на то, в скриптах находится большое количество блоков, их нужно структурировать, чтобы не запутаться.
4. Педагог должен обратить внимание учащихся на то, что при использовании команды «Отталкивание от края» важно правильно настроить свойства спрайта, чтобы он не ходил «вверх ногами».

Тема № 3 «Передача сообщений»

Цель занятия: Освоение методов создания анимации с использованием команд передачи сообщений из блока «События».

Задачи занятия:

Образовательные:

- ✓ познакомиться с командами передачи сообщений;
- ✓ научиться осуществлять обмен сообщениями между спрайтами;
- ✓ научиться создавать анимацию с участием нескольких спрайтов с использованием команд передачи сообщений;
- ✓ закрепить навыки использование команд блока «Движение».

Планируемые результаты

Предметные:

Знать

- ✓ команды «Передать сообщение» и «Когда я получу сообщение» блока «События»
- ✓ команды «Плыть», «Перейти», «Повернуться» блока «Движение»

Уметь

- ✓ осуществлять обмен сообщениями между спрайтами;
- ✓ создавать анимацию с участием нескольких спрайтов с использованием команд передачи сообщений;
- ✓ создавать анимацию, реагирующую на положение курсора мыши на экране;

Метапредметные:

- ✓ ученики обретут навыки осуществлять самонаблюдение, самооценку, самоконтроль;
- ✓ работы в целом, анализировать полученные результаты;

План занятия:

1. Практическая работа «Эстафета». Знакомство с командами передачи сообщений.
2. Практическая работа «Скачки».
3. Знакомство с методами слежения и прилипания с использованием блоков «Движение».
4. Практическая работа «Прилипание»
5. Подведение итогов.

Ход занятия.

1. Организационный момент. Приветствие.

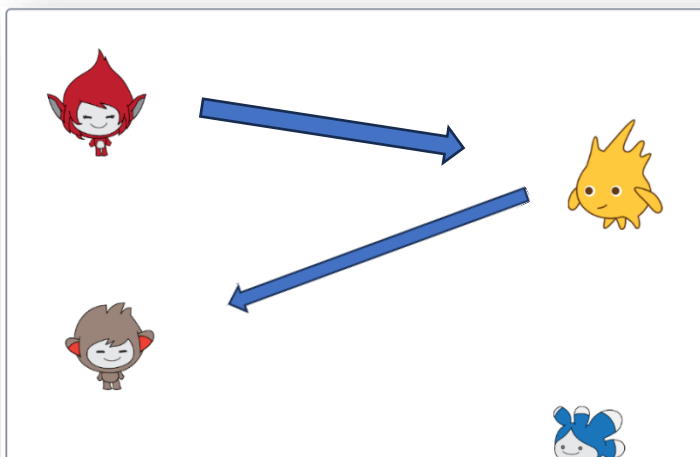
2. Практическая работа «Эстафета». Знакомство с командами передачи сообщений

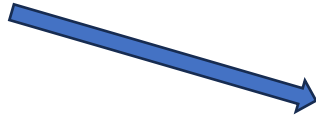
Педагог предлагает учащимся создать анимацию движения от одного спрайта к другому с использованием блоков из группы «Движение». Поясняет, что начало движения каждого спрайта зависит от того, когда закончит движение предыдущий участник эстафеты. Обсуждает с учениками, каким образом можно это реализовать (расчет задержки начала движения), обсуждает недостатки этого метода. Рассказывает, что для синхронизации этих действий в среде программирования Scratch существуют сообщения.

Задача:

Реализовать эстафету между спрайтами.

- 1) Разместить на сцене четырех героев, установить их в нужном направлении.
- 2) У каждого спрайта на событие «**Когда нажат флажок**» прикрепить блок «**Перейти в**»
- 3) Перетащить каждый спрайт в точку, где он закончит эстафету и установить в программу спрайта блок движения в выбранную точку, используя «**Плыть ... секунд в точку x: ... y: ...**».
- 4) После блока «**Плыть...**» добавить блок передачи сообщения следующему спрайту для начала движения.





```
когда флажок нажат
  перейти в x: -177 y: 106
  плыть 1 секунд в точку x: 158 y: 53
  передать Эстафета передана Gobo
```

```
когда флажок нажат
  перейти в x: 178 y: 67
  когда я получу Эстафета передана Gobo
  плыть 1 секунд в точку x: 122 y: -145
  передать Эстафета передана Nalo
```

3. Практическая работа «Скачки».

Изучение взаимодействия объектов на основе обмена сообщениями

Задача:

- 1) Добавить спрайт Лошадки, которая скачет вперёд-назад и при касании края сцены говорить: «Ой!»
- 2) Добавить спрайт Зайчика
- 3) При касании Лошадки о край Зайчик должен сказать: «Бедняжка!» и сменить костюм.

```
когда флажок нажат
  повторять всегда
    идти 10 шагов
    если касается края, оттолкнуться
    установить способ вращения влево-вправо
    если касается край ?, то
      говорить Ой! 1 секунд
      передать Ой!
```

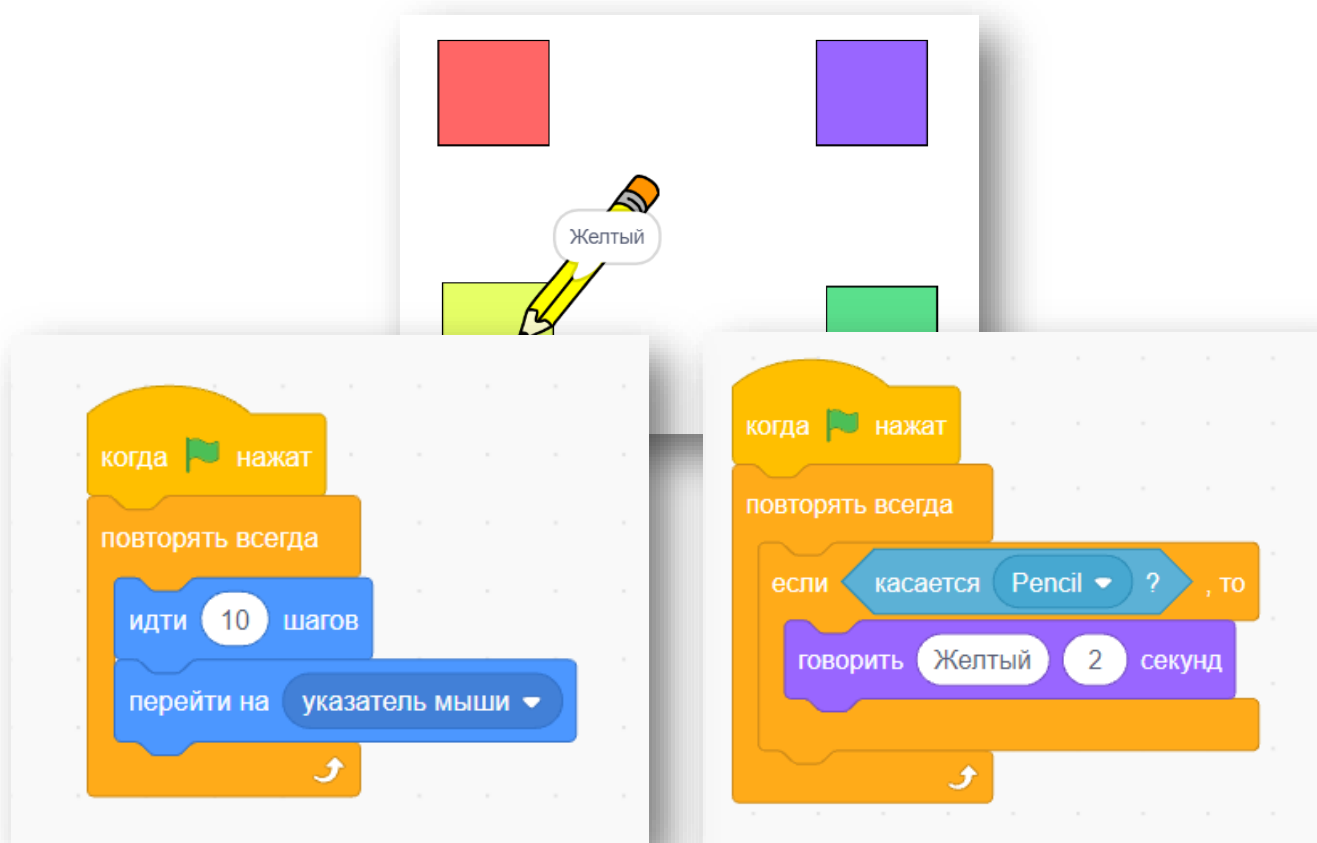
```
когда я получу Ой!
  изменить костюм на hare-a
  говорить Бедняжка! 1 секунд
  ждать 0.5 секунд
  изменить костюм на hare-b
```

4. Практическая работа «Прилипание»

Задача:

Используя блок «Перейти в указатель мыши» реализовать следующий игровой сценарий:

- ✓ Добавить на сцену квадраты пяти цветов и спрайт карандаша.
- ✓ При запуске программы карандаш прилипает к указателю мыши и движется вместе с курсором. При соприкосновении курсора с квадратом, квадрат называет свой цвет.



5. Подведение итогов занятия.

Методические рекомендации

1. Педагог должен объяснить, что программа будет и должна работать после условного сигнала в виде передаваемого сообщения.
2. Следует обратить внимание детей на то, что каждый спрайт должен реагировать именно на отправленное ему сообщение, чтобы спрайты выполняли действия в правильном порядке (порядке передачи сообщений). Часто возникает ошибка, когда одно и то же сообщение отправляется одновременно нескольким спрайтам и поэтому программа работает неправильно (например, при выполнении проекта Эстафета).

3. Стоит отметить, что учащимся нужно научиться правильно определять координаты точек, в которые должен «приплыть» спрайт при программировании его движения.

Тема № 4 «Таймер. Случайные числа»

Цель занятия: Освоение методов создания простейших игр с использованием оператора «Случайные числа», а также сенсора «Таймер».

Задачи занятия:

Образовательные:

- ✓ получить представление о координатах экрана программной среды Scratch;
- ✓ научиться программировать движение кота с помощью смены координат;
- ✓ научиться создавать простейшие игры с использованием сенсора «Таймер» и оператора «Случайные числа»;

Планируемые результаты:

Предметные:

Знать

- ✓ понятие «координаты»;
- ✓ команды «Изменить x » и «Изменить y » блока «Движение»;
- ✓ сенсоры «Таймер» и «Касается цвета»;
- ✓ оператор «Случайные числа»;

Уметь

- ✓ определять координаты спрайта;
- ✓ создавать простейшие игры, используя блоки, изученные на данном занятии;

Метапредметные:

- ✓ учащиеся обретут навыки осуществлять самонаблюдение, самооценку, самоконтроль;
- ✓ научатся планировать последовательность и прогнозировать итоги действий и всей работы в целом, анализировать полученные результаты;

План занятия:

1. Знакомство с координатами экрана программной среды Scratch.
2. Знакомство с командами, позволяющими запрограммировать движение спрайта с помощью смены координат.
3. Знакомство с сенсором «Таймер».
4. Практическая работа «Минное поле».
5. Знакомство с оператором «Случайные числа».
7. Самостоятельная работа «Игра Лабиринт».
8. Подведение итогов.

Ход занятия.

1. Организационный момент. Приветствие.

2. Знакомство с координатами экрана программной среды Scratch.

Педагог рассказывает о размерах сцены, обращая особое внимание учеников на то, что начало координат располагается в центре сцены и на максимальные и минимальные значения, которые могут принимать координаты x и y .

3. Знакомство с командами, позволяющими запрограммировать движение спрайта с помощью смены координат.

Педагог обращает внимание учеников на то, что при изменении координаты x на положительное число спрайт смещается вправо, а на отрицательное – влево. Соответственно при изменении координаты y на положительное число спрайт смещается вверх, а на отрицательное число – вниз.

4. Практическая работа «Минное поле».

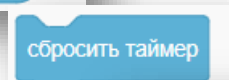
Педагог предлагает учащимся создать простейшую игру, в которой спрайт должен дойти до финиша, пройдя по минному полю. В начале игры должен стартовать таймер, а в конце игры должно выводиться его значение.

Свойства таймера:

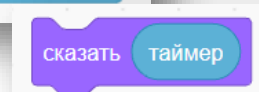
Включение таймера на экране:



Перезапуск таймера в начале программы:



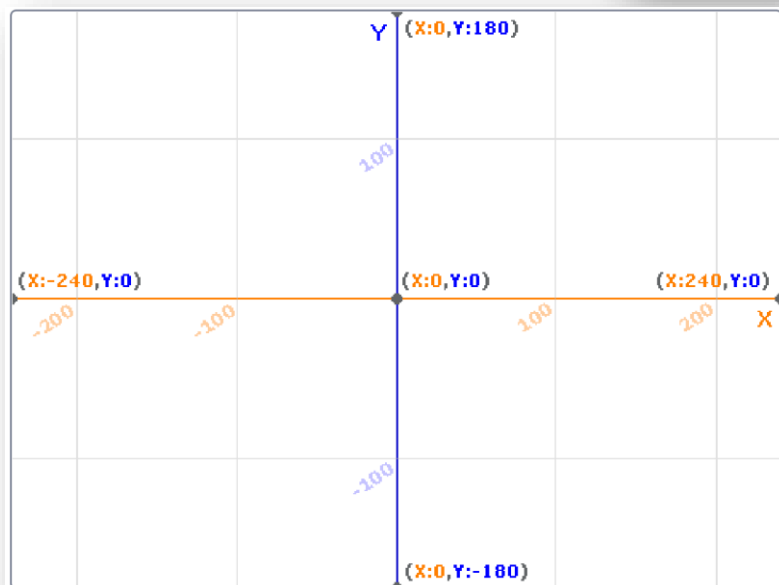
Вывод текущего значения таймера в сообщение:



ВНИМАНИЕ!!! Чтобы остановить таймер при достижении финиша, нужно коснуться финиша и остановить все скрипты

СТОП все ▾

Размеры сцены:



Задача:

Провести кота по минному полю:

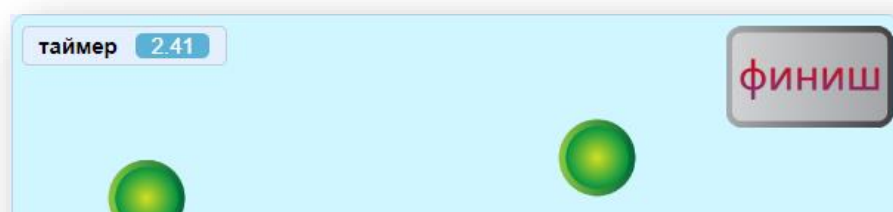
1) Кот управляется стрелками на клавиатуре

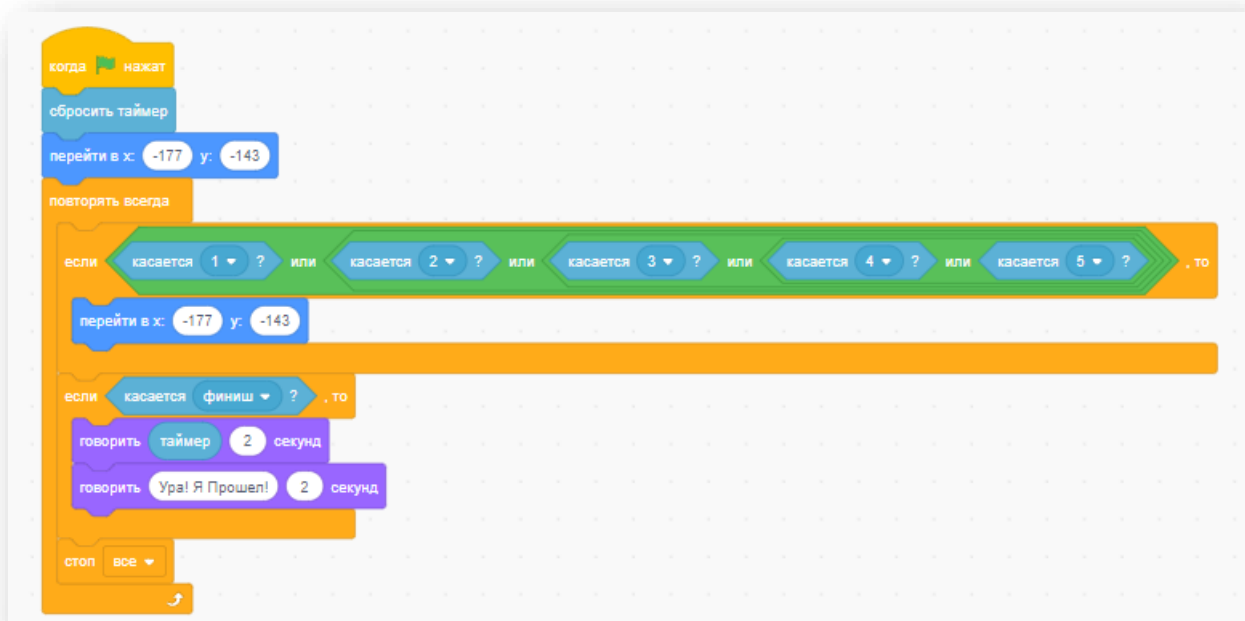
2) При запуске программы перезапускаем таймер.

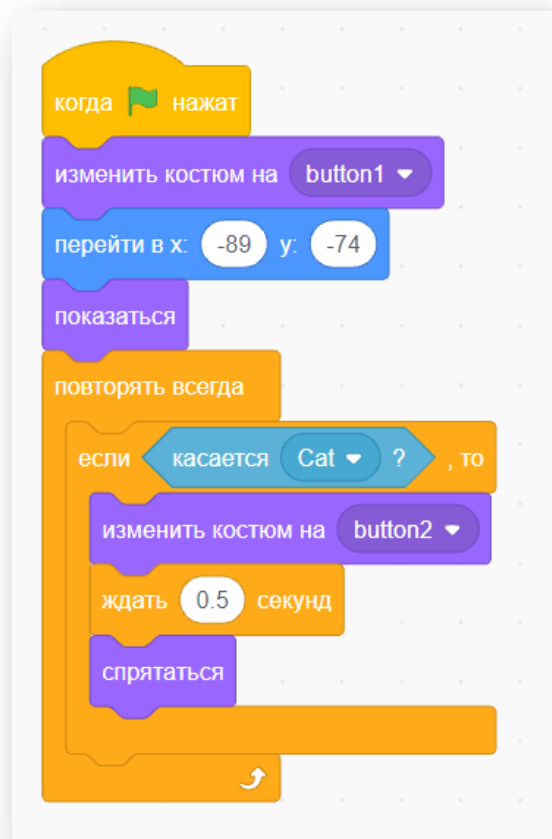
3) Мина должна иметь 2 костюма – мина и взрыв.

4) Если Кот наступит на Мину, то она взорвется и исчезнет, а кот вернется на **Старт**.

5) Если кот коснется финиша, кот произносит значение таймера и возглас кота «Ура, прошел!». Для **остановки таймера** при достижении финиша, нужно **остановить все скрипты**.



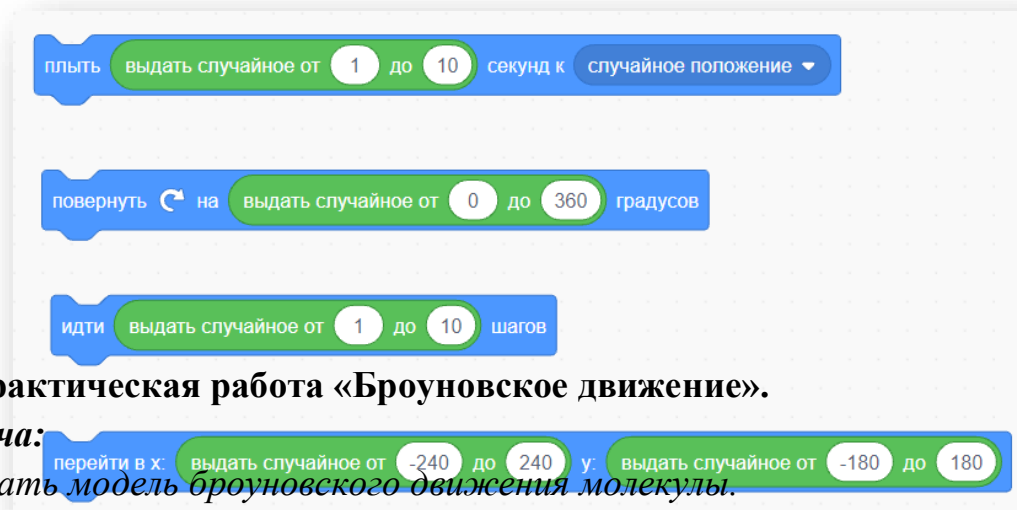




6. Знакомство с оператором «Случайные числа».

Педагог рассказывает воспитанникам об операторе «Случайные числа», его назначении и о тех блоках, с которыми данный оператор можно использовать.

Где можно использовать:

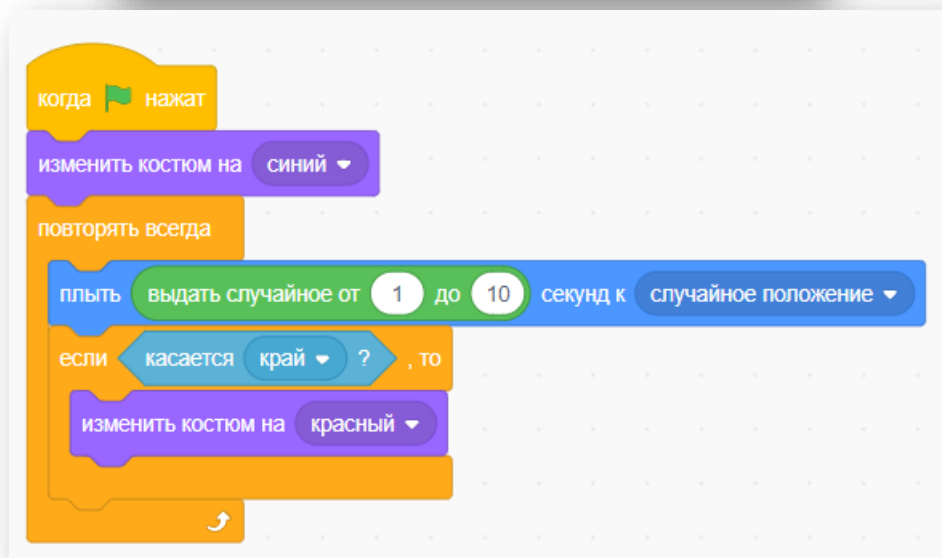
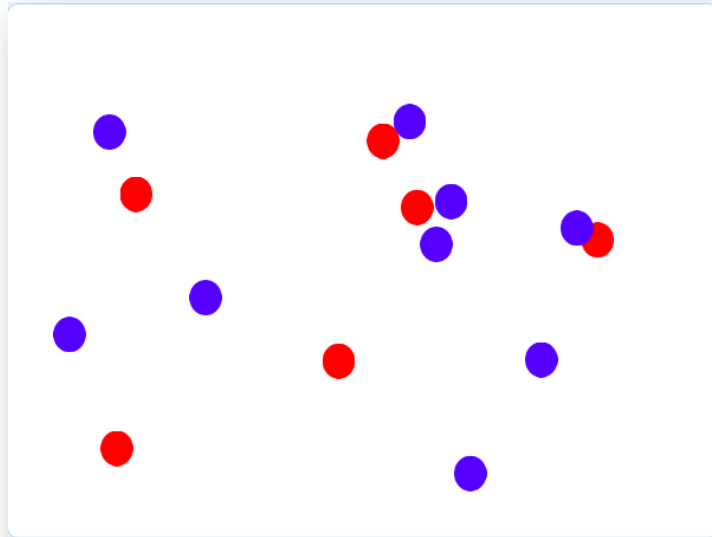


6. Практическая работа «Броуновское движение».

Задача:

Создать модель броуновского движения молекулы.

1. Создать спрайт молекулы синего цвета
2. Сымитировать плавное движение молекулы в случайные точки с использованием блока «Плыть...»
3. Настроить так, чтобы при касании края молекула меняла цвет.
4. Дублировать созданную молекулу на поле.



7. Самостоятельная работа «Игра Лабиринт».

Задача:

Создать модель игры «Лабиринт».

- 1) Загрузите готовый фон лабиринта.
- 2) Создать спрайт героя.

Начальное положение: **«смотрит вправо»**.

- 3) Создать скрипт перемещения героя по сцене по нажатию на стрелки. Каждое нажатие перемещает на 10 шагов. Для перемещения использовать **«изменить x на ...»**, **«изменить y на ...»**
4. Для поворота героя в сторону движения, нужно при каждом перемещении менять направление с помощью **«повернуть в направлении ...»**.
5. При касании черного цвета, необходимо отойти назад, изменив значение количества шагов на противоположное: **10 на -10**.



6. При касании финиша вывести значение таймера и сообщение о победе.
7. Ввести в игру спрайт дракона, который перемещается произвольно по полю и при касании героя выводит сообщение «Проиграл!!», а герой при этом возвращается в исходную позицию.

8. Подведение итогов занятия.

Методические рекомендации

1. Педагогу следует обратить внимание на систему координат, размеры сцены, какие максимальные и минимальные значения могут принимать координаты x и y , а также как изменяется расположение спрайта при их изменении с использованием команд из блока «Движение».
2. Следует обратить внимание учащихся на то, что команда «Стоп все» не должна располагаться внутри бесконечного цикла, в противном случае программа работать не будет.
3. В ходе выполнения практической работы ученики забывают ставить блок «Ждать», из-за чего программа выполняется мгновенно и результат не всегда виден. Блок «Ждать» не должен быть совсем длинным.

Тема № 5 «Переменные»

Цель занятия: Знакомство с переменными в среде программирования Scratch.

Задачи занятия:

Образовательные:

- ✓ познакомиться с понятием «переменная»;
- ✓ получить представление о понятиях «глобальная переменная», «локальная переменная»;

- ✓ изучить блоки «Переменные» и «Операторы»;
- ✓ получить представление о различиях растровой и векторной графики;
- ✓ научиться создавать новые спрайты, используя возможности графического редактора Scratch.

Планируемые результаты:

Предметные:

Знать

- ✓ понятия «переменная», «глобальная переменная», «локальная переменная»;
- ✓ возможности сенсора «Ответ»;
- ✓ арифметические и логические операции;
- ✓ конструкцию операторов «Если-То», «Если-То-Иначе»;

Уметь

- ✓ создавать и редактировать векторную графику;
- ✓ создавать переменные;
- ✓ задавать значения переменным, изменять значения в процессе выполнения программы;
- ✓ использовать арифметические и логические операторы, в том числе со сложными условиями;
- ✓ сравнивать значения переменных и принимать решение и дальнейшем ходе программы (конструкции «Если-то», «Если-то-иначе»)

Метапредметные:

- ✓ учащиеся обретут навыки вести самонаблюдение, самооценку, самоконтроль;
- ✓ научатся планировать последовательность и прогнозировать итоги действий и всей работы в целом, анализировать полученные результаты;

План занятия:

1. Знакомство с понятиями «переменные», «глобальная переменная», «локальная переменная».
2. Знакомство с растровой и векторной графикой.
3. Создание новых спрайтов, с использованием возможностей графического редактора Scratch.
4. Творческие задания.
5. Подведение итогов.

Ход занятия.

1. Организационный момент. Приветствие.

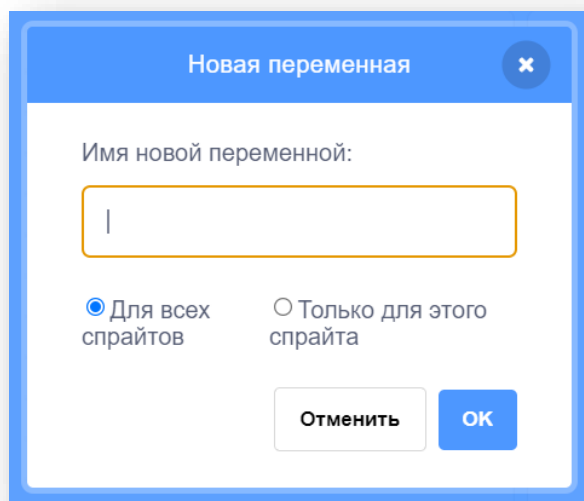
2. Знакомство с понятиями «переменные», «глобальная переменная», «локальная переменная».

Педагог проводит беседу с учащимися о необходимости хранить данные промежуточных вычислений, рассказывает о локальных и глобальных переменных, их отличиях, где и какие переменные необходимо использовать. Дает рекомендации, как задать тип переменной.

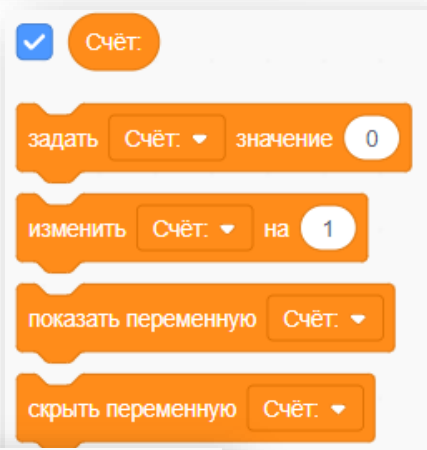
Переменные – это специальные ячейки в компьютерной памяти, в которых можно хранить разные числа и слова.

У каждой переменной есть своё имя. Создание переменной:

«Данные» - «Создать переменную»

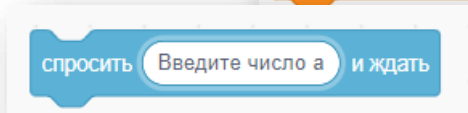


Настройки переменной:

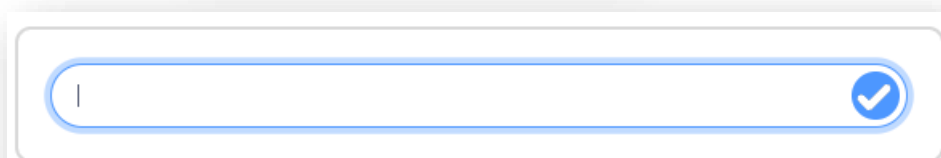


Ввод данных с клавиатуры

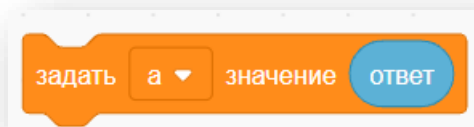
В скрипте:



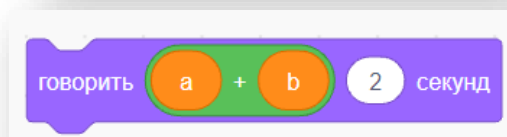
На сцене:



Введенное с клавиатуры число помещается в сенсор «**Ответ**», который можно использовать в программе:



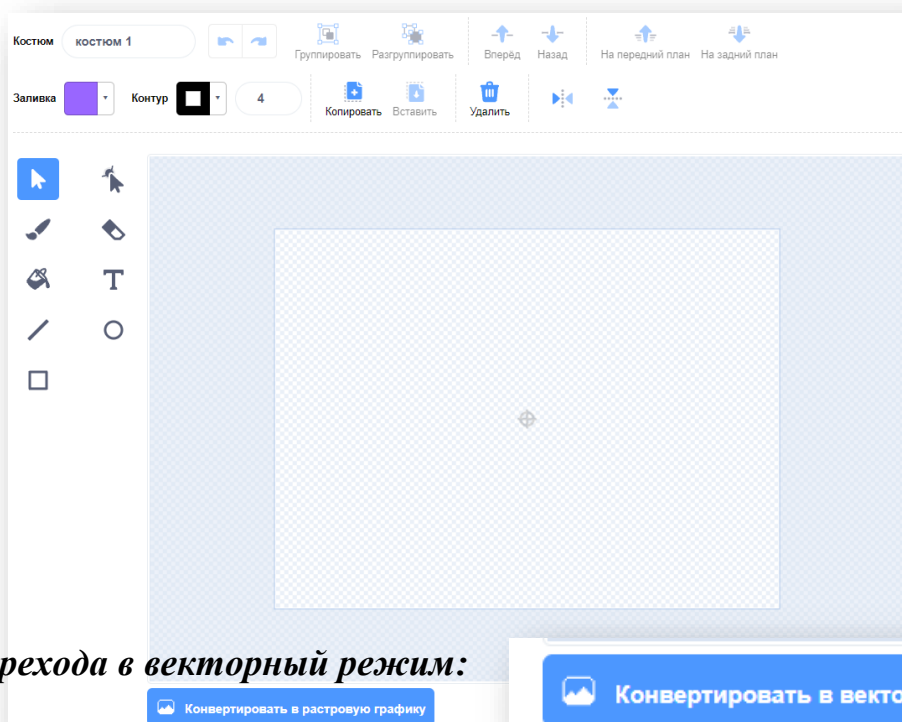
Для вывода результат сложения двух переменных можно использовать блок:



3. Знакомство с растровой и векторной графикой

Педагог задает наводящие вопросы, выясняя, сталкивались ли дети с векторной и растровой графикой. Рассказывает теоретический материал о том, как описывается изображение в том и другом случае. Объясняет, для каких целей лучше использовать растровую графику, а когда – векторную. Показывает, как в графическом редакторе Scratch создать новый спрайт в векторной графике.

Рисование собственных спрайтов:



Для перехода в векторный режим:

Конвертировать в векторную графику

4. Творческие задания

Проект «Простой калькулятор»

Закрепление умения пользоваться переменными.

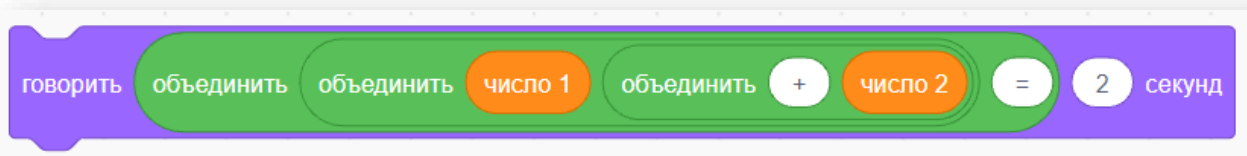
Задача: создать мини-калькулятор, выполняющий 4 основные операции

- 1) Создать две переменные *A* и *B*.
- 2) **Все действия назначать спрайту Кота!!**
- 3) Создать возможность ввода числа *A* с клавиатуры:
- 4) Занести полученное число в переменную *A*
- 5) Создать возможность ввода числа *B* с клавиатуры и занести полученное число в переменную *B*.
- 6) Вывести результат.

Проект «Реши пример»

Задача: создать программу, выводящую на экран 15 примеров и проверяющую умение складывать числа в диапазоне от 1 до 20

- 1) Создать переменные *Ответ*, *Число 1* и *Число 2*
- 2) Назначить спрайту Кота:
 - ✓ Создать 2 случайных числа от 1 до 20 и записать их в переменные *Число 1* и *Число 2*
 - ✓ Спрайт кота говорит пример $\text{Число 1} + \text{Число 2}$



- 3) Создать спрайт-кнопку «Проверить» и назначить ей скрипт:
 - ✓ Создать поле для ввода числа с клавиатуры с помощью команды «Спросить»
 - ✓ Занести полученное число в переменную *Ответ*
 - ✓ Сравнить число в переменной *Ответ* с верным, ответом, который должен был бы получиться в этом примере и вывести на экран сообщение о том, верный или неверный введен ответ.
- 4) **Модифицируйте скрипт** так, чтобы за каждый верно данный ответ добавлялось одно очко и количество очков было видно на экране.

Самостоятельная работа. Проект «Ребусы»

Задача: создать программу, позволяющую отгадывать ребусы-картинки.



5. Подведение итогов занятия.

Методические рекомендации.

1. Для корректного объяснения работы желательно предварительно выполнить задания самостоятельно, либо ознакомиться с готовыми проектами-примерами.
2. Часто возникает ошибка занесения значения, введенного с клавиатуры, в какую-либо переменную. Необходимо помнить, что сначала мы всегда получаем введенное значение в сенсор «Ответ», затем значение «Ответ» записываем в ту переменную, с которой нам необходимо произвести какие-либо действия.
3. Для выполнения проекта «Ребусы» можно предложить ученикам использовать заранее приготовленные изображения ребусов или предложить скачать картинки из сети интернет.

Список литературы

1. Программирование для детей. Видеоигры на Scratch/Йохан Алуден и др.; пер с англ. Д. В. Голикова. – М.: РОСМЭН, 2018. – 126с.: ил.;

2. Программирование для начинающих. Привет Scratch. Моя первая книга по программированию. Дубовик Е.В., Иркова Ю.А., Русин Г.С.-Наука и техника ,2018. – 240с.;
3. Scratch для юных программистов. Голиков Д.В. – ВHV, 2017г. – 192с.
4. Scratch для детей. Самоучитель по программированию. Мажет М., МАНН, ИВАНОВ и ФЕРБЕР ООО, 2018г.-288с.;
5. Юлия Торгашева: Программирование для детей. Учимся создавать игры на Scratch СПб.: Питер, 2018 г., 128 с.;
6. Путина А.С., Scratch 2.0: от новичка к продвинутому пользователю. Пособие для подготовки к Scratch-Олимпиаде [Электронный ресурс] / Путина А.С. - М.: Лаборатория знаний, 2019. - 90 с. - ISBN 978-5-00101-618-2 - Режим доступа:
<http://www.studmedlib.ru/book/ISBN9785001016182.html>
7. 2. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch. — Оренбург: Оренб. гос. ин-т. менеджмента, 2009.;
8. «Раннее обучение программирование в среде Scratch», В.Г. Рындак, В.О. Джинжер, Л.В. Денисова, 2018.;
9. Голиков Д.И. «Scratch для юных программистов», «БХВ-Петербург», Санкт-Петербург, 2017.;
10. Портал «Дополнительное образование детей» [Электронный ресурс]. – Режим доступа: <http://vidod.edu.ru>