*Государственное бюджетное общеобразовательное учреждение Самарской области*

*средняя общеобразовательная школа «Образовательный центр» имени 81 гвардейского мотострелкового полка п.г.т. Рощинский муниципального района Волжский Самарской области*

**443539, Самарская область, Волжский район, п. г. т. Рощинский, школа.**

**Официальный сайт учреждения:** [**http://**[**roshchaschool.minobr63.ru**](http://roshchaschool.minobr63.ru/)](http://mou-rs.narod.ru),

**адрес электронной почты: roshinsky\_sch\_vlg@samara.edu.ru**

**Контактная информация: телефоны: 932 – 82 – 58 (ф), 932 – 82 – 50**

**План-конспект по внеурочной деятельности «Знакомство со средой программирования Scratch»**

**Учитель информатики**

**высшей квалификационной категории**

**Проскурякова И.В.**

**Тема:**«Знакомство со средой программирования Scratch»

**Используемые программные приложения из пакета СПО:**Среда программирования Scratch.

**Цели урока:**

Образовательная:

* Ввести понятия «среда программирования».
* Познакомить с объектами среды программирования Scratch, а так же правилами записи команд.

Развивающая:

* Развитие приемов умственной деятельности (обобщение, анализ, синтез), внимание, памяти, творческой активности.
* Способствовать развитию умения сравнивать, анализировать информацию, ориентироваться на плоскости.
* Пробудить у детей желание экспериментировать.

Воспитательная:

* Способствовать развитию умения аргументировать свою точку зрения.
* Развитие познавательного интереса у учащихся, основ коммуникативного общения, уверенности в собственных силах, аккуратность.

**Задачи урока:**

* Приобретение умения создавать свою первую программу в среде Scratch.
* Рассмотрение примеров использования возможностей программной среды Scratch для создания анимированных моделей.
* Приобретение умения использовать встроенные библиотеки персонажей.

**Планируемые образовательные результаты:**

Предметные — формирование представлений о программной среде Scratch.

Метапредметные — формирование представления о компьютере как универсальном устройстве для создания анимационных моделей.

Личностные — понимание роли компьютера в жизни современного человека в качестве инструмента для решения задач моделирования процессов и явлений.

**УУД:**

Регулятивные:

* Саморегуляция.
* Оценка - выделение и осознание учащимся того, что уже усвоено и что еще подлежит усвоению, осознание качества и уровня усвоения.

Коммуникативные:

* Планирование учебного сотрудничества с учителем и сверстниками.
* Выражение своих мыслей; аргументация своего мнения; учёт разных мнений.
* Оценка действий партнера.
* Умение слушать и слышать.
* Аргументация своего мнения и позиции в коммуникации.

Познавательные:

* Самостоятельное выделение и формулирование познавательной цели.
* Выбор наиболее эффективных способов решения задач в зависимости от конкретных условий.
* Рефлексия способов и условий действия; контроль и оценка процесса и результатов деятельности.

Личностные:

* Оценивание усваиваемого содержания.
* Самооценка.
* Адекватное понимания причин успеха или неуспеха.
* Следование в поведении моральным нормам и этическим требованиям.

**План урока:**

1. Организационный момент.
2. Постановка темы и цели урока.
3. Актуализация знаний.
4. Изложение новой темы
5. Практическая работа «Первая программа».
6. Самоанализ урока.
7. Выставление оценок.
8. Домашнее задание.

**Организационный момент**

Приветствие учащихся, создание позитивного учебного настроя. Проверка присутствующих.

**Постановка темы и цели урока**

Для достижения поставленных целей выбрана традиционная форма урока, с включением элементов интерактивных технологий и отработкой практических навыков.

Огромное значение имеет целевая установка на серьёзную работу в течение урока. Важно, чтобы каждый ученик поставил перед собой цель и постарался добиться наилучшего результата.

**Актуализация знаний**

Что такое алгоритм? (это последовательность действий, строгое исполнение которого приводит к определенному результату)

Какие основные алгоритмические конструкции вы знаете. (линейные, ветвление, циклические) приведите пример

Кто может составлять алгоритмы? (человек)

А кто исполняют эти алгоритмы? (различные исполнители (роботы, вычислительные машины))

**Изложение новой темы**

Приложение Scratch представляет собой среду для создания программ и анимации. Среда Scratch позволяет делать очень многое. Поэтому ее интерфейс (внешний вид после запуска) достаточно сложный. Однако когда мы разберемся с принципами работы в Scratch, поймем как составлять программы и создавать несложную анимацию, то нам откроются большие просторы для самостоятельного творчества и исследования.

Блоки команд в Scratch

На этом уроке нас будут интересовать только команды, связанные с кнопками движение и контроль.



Свойства объекта: Здесь показаны свойства объекта, которым мы будем управлять (писать программы для него). Сейчас - это кот. Его имя написано в поле - Спрайт1. Странное имя для кота, не так ли? Спрайтами в Scratch называются все объекты по умолчанию, отличаются лишь их номера. Обычно предполагается, что мы сами их будем переименовывать и называть более адекватными именами. Например, для кота лучше в поле имени вписать кот или Вася или что-нибудь еще. Сделайте это.



Ниже поля имени отображены три свойства объекта (в данном случае, кота) — это его положение (координаты x и y) и направление.

Внизу мы видим три кнопки-вкладки — скрипты, костюмы и звуки. От того какая из них нажата, зависит ячейка внизу. Если нажата кнопка скрипты, то нижняя ячейка второго столбца покажет программы (скрипты) для объекта, который отображен в верхней ячейки. Сейчас нажата кнопка скрипты и мы видим пустую ячейку внизу. Это значит, что для кота пока нет никакой программы. Мы составим ее чуть позже. Кнопки костюмы и звуки позволяют настраивать и менять соответственно внешний вид объекта и издаваемые им звуки. Оставьте включенной кнопку скрипты.

Перейдем к третьему столбцу и опишем в нем только верхнюю ячейку, где видим белое поле и кота на нем. Белое поле — это холст. Многие действия, которые задает программист на вкладке скрипты, объект выполняет именно на холсте. Так, если мы запрограммируем ходьбу кота, то он будет перемещаться по холсту.

Если зажать левую кнопку мыши над котом, а затем не отпуская ее перемещать мышь, то кот будет перемещаться. Таким образом, мы можем менять его положение на холсте. Переставьте кота в другое место и посмотрите в ячейку свойств объекта (второй столбец). Заметьте, что координаты x и y изменили свои значения.

Над холстом находятся две вот такие кнопки:



Кнопки запуска и остановки анимации. Когда для объектов (кота и других) будут составлены какие-нибудь скрипты (программы) на вкладке скрипты, то чтобы объекты начали их выполнять, надо нажать зеленый флажок. Чтобы остановить выполнение, надо нажать красный круг. Не забывайте останавливать свои скрипты!

Над флагом и кругом мы видим еще три кнопки:



Кнопки режимов окна Scratch. Они предназначены для изменения всего окна приложения Scratch. Сейчас у нас включена вторая кнопка, поэтому окно Scratch примерно поровну разделено на три столбца. Однако если мы составляем какую-нибудь очень сложную программу и у нас получается очень большой скрипт, то лучше нажать первую кнопку. При этом второй столбец займет существенную часть окна. Третья кнопка предназначена исключительно для просмотра созданной программы-анимации. Если ее нажать, то холст занимает центр экрана, и доступными остаются только три кнопки: запуск программы (зеленый флажок), остановка программы (красный круг) и выход из режима презентации (стрелка), осуществляющий возврат к предыдущему окну. Попробуйте переключаться между режимами работы.

**Практическая работа «Первая программа»**

Мы уже рассмотрели интерфейс достаточно, чтобы написать первую программу для кота.

Сейчас включена кнопка движение и видны команды, отвечающие за перемещение объекта. Если перетащить мышью любую их этих команд в ячейку скрипты, то она станет командой для кота. Например, иди 10 шагов заставит животное переместиться на 10 точек экрана. После того как команда перемещена в скрипты, можно посмотреть, как она работает, совершив двойной клик по ней. Кот шагнет на холсте. Задайте команду иди 10 шагов для объекта и проверьте, как она работает.



У нас есть программа для кота, состоящая из одной команды. По идее при нажатии на кнопку запуска (зеленый флаг) она должна срабатывать. Но на самом деле этого не произойдет. Чтобы программа запускалась при нажатии флажка, надо в ее начало поместить специальную команду, которая связана с кнопкой контроль. На ней написано когда щелкнут по и изображен зеленый флажок. Если взять эту команду, перетащить в поле скриптов и соединить с командой иди 10 шагов, то программа станет запускаться при нажатии на кнопку запуска. Сделайте это.

Обратите внимание на то, как соединяются между собой команды: как блоки в конструкторе Лего. В каких-то случаях это может служить своего рода подсказкой. Какие-то команды можно соединять между собой, а какие-то не соединяются, и, значит, мы составляем не очень правильную программу.

Чтобы разорвать блоки, надо потянуть за нижнюю команду, чтобы переместить весь блок - за самую верхнюю. Попробуйте и затем верните все на место.

Сделаем некоторые выводы. Наш код даже не шагает, а просто прыгает на 10 точек вперед и останавливается. Конец программы. Хорошо бы было, чтобы он все-таки шагал и проделывал путь несколько больше. Самое первое, что может прийти в голову, — это увеличить количество шагов, например, до 100. Для этого надо в команде иди 10 шагов щелкнуть по числу 10 и вписать в поле число 100. Переместите кота ближе к левой границе холста, поменяйте число шагов и запустите программу.

Теперь кот шагает больше. Но разве это похоже на шаги? Он просто скачет на большее расстояние. Поэтому вернем число 10 и подумаем над другим решением.



В командах контроля есть такая команда как повторить 10. Это цикл, который прокручивает то, что в него вложено такое количество раз, какое число написано в его поле. По умолчанию — это 10. Если команду иди 10 шагов поместить внутрь этого цикла, то он будет ее «крутить». Поместите кота ближе к левой границе холста и составьте вот такую программу для него. Посмотрите, как теперь наш объект будет двигаться. Согласитесь, что у нас получилась настоящая анимация объекта.



Продолжим улучшать и видоизменять нашу первую программу. Допустим, мы не знаем, сколько раз надо повторять шаги. Мы хотим, чтобы объект двигался всегда, пока мы не нажмем кнопку остановки (красный круг). Для этого надо поступить так: вытащить команду иди 10 шагов из цикла повтори ..., убрать цикл повтори ... (перетащить его в левый столбец), взять команду всегда и собрать такую программу:



Теперь программа должна работать вечно, пока не будет остановлена. Так и происходит. Если нажать кнопку запуска, то зеленый флажок будет гореть до тех пор, пока не будет нажат красный. Это значит, что программа не может сама остановиться. Но посмотрите, что происходит с котом. Он останавливается, достигнув границы холста. Программа работает, а кот стоит. Тут что-то не так. Останавливаем программу! Возвращаем кота на место.

С кнопкой движение связана такая команда как если край, оттолкнуться. Если ее поместить в цикл всегда, то объект уже не остановится достигнув края, а оттолкнется от него и пойдет в другую сторону. Составьте вот такой скрипт и посмотрите, как он работает.

Скрипт работает замечательно, но кот не совсем нормален при движении справа налево. Он идет вниз головой. Для исправления этого недочета остановим программу и снова обратим свой взгляд на ячейку свойств объекта.



В левой части этой ячейки есть три маленькие кнопки: со скругленной стрелкой, двуголовой стрелкой и квадратной точкой. По умолчанию нажата первая кнопка и поэтому объект при столкновении поворачивается так, как мы наблюдали. Если нажать вторую кнопку, то он будет поворачиваться так, как нам надо в данный момент (слева направо). Третья кнопка вообще запрещает какие-либо повороты. Нажмите вторую кнопку и посмотрите, как ходит кот. Не забудьте после этого остановить программу.

Посмотрите на изображение кота в ячейке свойств. У него есть какой-то синий отрезок. Зажав на нем мышь, его можно поворачивать. При этом меняется значение направления. Это означает, что объект при запуске программы будет двигаться в установленном таким образом направлении. Поэкспериментируйте с выбором направления движения кота.