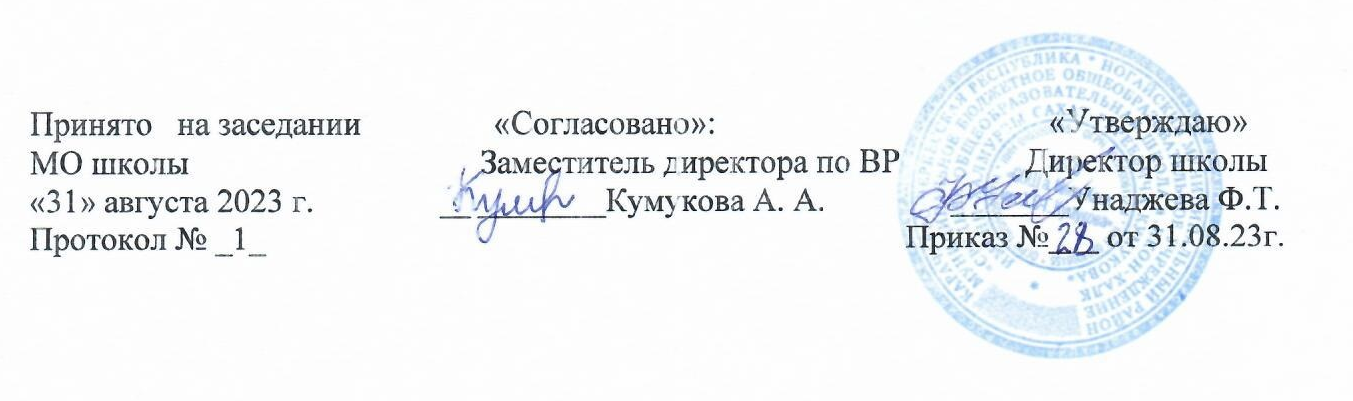
**Муниципальное бюджетное общеобразовательное учреждение**

**«Средняя общеобразовательная школа а. Икон-Халк им. Х. С-Г. Кумукова»**

****

**Дополнительная общеразвивающая программа**

**общего образования**

**научно-технической направленности**

**«Программирование в среде Scratch»**

ID программы в Навигаторе:\_\_\_

Уровень программы: стартовый

Возраст обучающихся: 9-11 лет

Срок освоения: 1 год

Объем 68 часов

**Автор составитель:**

Джанарсланова А. М.

**2024 г.**

**Икон-Халк**

**Содержание**

**I. Целевой раздел** 3

1.1. Пояснительная записка 3

1.2. Цели и задачи программы 3

1.3. Актуальность, отличительные особенности программы 4

1.4. Методические подходы к обучению программированию школьников 5

1.5. Планируемые результаты 5

Ожидаемые результаты и способ их проверки 6

**II. Организационный раздел** 7

2.1. Продолжительность и этапы реализации программы 7

2.2. Материально – техническое обеспечение программы 7

**III.Содержательный раздел** 8

3.1. Содержание программы 8

3.2.Учебно-тематический план 12

**IV. Дополнительный раздел** 13

4.1. Используемая литература 13

**I. Целевой раздел**

**Пояснительная записка**

*Рабочая программа внеурочной деятельности* «Программирование в среде «Scratch» *составлена в соответствии с требованиями Федерального государственного общеобразовательного стандарта начального общего образования.*

Направление прогрaммы - научно-техническая. Обучение по данной прогрaмме направлено на приобретение учащимися базовых знаний в области программирования и умению создавать творческие проекты, а также привлечение их к современным информaционным технологиям.

**Цель программы:** формирование инженерно-технического мышления, творческих способностей обучающихся посредством изучения среды программирования Scratch.

**Задачи:**

*Образовательные:*

− обучить основам программирования в среде Scratch;

− сформировать навыки разработки, тестирования и отладки проектов в Scratch;

− научить использованию приемов векторной и растровой графики;

− познакомить с алгоритмом «Творчества»;

− научить самостоятельному созданию продуктов в среде программирования Scratch;

− познакомить с правилами участия в олимпиадах по программированию в среде Scratch.

*Развивающие:*

− развивать воображение, аналитическое, логическое мышление и творческие способности;

− развивать интерес к занятиям технической направленности;

− формировать мотивацию к выбору профессий инженерно-технической направленности.

*Воспитательные:*

− воспитывать самостоятельность, ответственность;

− воспитывать усидчивость, умение доводить начатое до конца;

− формировать коммуникативные умения и навыки командной работы.

**Актуальность**. Программирование сейчас является очень актуальной темой, так как ни одно устройство в наше время не работает без написанной программы. А в рамках школьного курса информатики на него отводится достаточно немного времени. Программирование на scratch доступно детям начиная с младшего школьного возраста. Научившись программировать на scratch, дети, которые всерьез заинтересуются программированием, смогут в дальнейшем с легкостью обучаться уже более серьезным языкам программирования и в будущем создавать свои продукты (например, программы для роботов и квадрокоптеров).

***Отличительные особенности программы***

Что такое Scratch и в чем его преимущество перед другими языками? В первую очередь, это визуальный язык программирования, то есть такой, в котором не нужно соблюдать строгие правила написания кода. Более того, код здесь писать не нужно вовсе! Вместо этого есть блоки, которые достаточно соединить в верной последовательности, чтобы получить желаемый результат — будь то небольшая игра или танцующий котик.

Если вы хотите, чтобы ваш ребенок начал изучать программирование с самого раннего детства, то Scratch — это отличный выбор. Scratch — язык программирования, который позволит сделать первый шаг в мир алгоритмов, интересных задач и созданных своими руками программ.

В Scratch всё зависит только от фантазии и навыков пользователя: можно создать анимированного персонажа, мультфильм или полноценную игру. Благодаря широкому функционалу программы, ребенок не только развивает творческие способности, но и прокачивает алгоритмическое мышление и навыки создания игр.

Из-за яркого интерфейса и мультяшных персонажей некоторые родители считают Scratch несерьезным — мол, это всего лишь развлечение, не стоит тратить на него время. Но как раз удобный интерфейс, веселые персонажи и понятные действия — особенности программы, которые позволяют ребятам младшего школьного возраста постигать азы программирования и при этом весело проводить время. Так что Scratch не стоит недооценивать.

***Методические подходы к обучению программированию школьников.***

1. **Системный подход**. Ориентирует на раскрытие целостности объекта, выявление многообразных типов связей в нём и сведение их в единую теоретическую картину.

1. **Деятельностный подход**. Предполагает формирование готовности школьника к активному, адекватному и эффективному применению информационных и телекоммуникационных технологий в процессе обучения информатике.
2. **Когнитивный подход**. Основан на таксономии Б. Блума и предполагает овладение учащимися операциями программирования, начиная с элементарных, постепенно продвигаясь к сложным.
3. **Проблемный подход**. Построение курса программирования на основе системы специально разработанных задач или ситуаций, содержание которых интересно для учащихся и расширяет их опыт программирования.
4. **Семиотический подход**. Направлен на целенаправленное развитие знаково-символических действий обучающ

***Планируемые результаты.***

***Личностные результаты***

К личностным результатамосвоения информационных и коммуникационных технологий как инструмента в учёбе и повседневной жизни можно отнести:

* критическое отношение к информации и избирательность её восприятия;
* уважение к информации о частной жизни и информационным результатам других людей;
* осмысление мотивов своих действий при выполнении заданий с жизненными ситуациями;
* начало профессионального самоопределения, ознакомление с миром профессий, связанных с информационными и коммуникационными технологиями.

***Метапредметными результатами*** изучения курса «Программирование в среде Scratch» являются формирование следующих универсальных учебных действий:

***Регулятивные УУД*:**

* планирование последовательности шагов алгоритма для достижения цели;
* поиск ошибок в плане действий и внесение в него изменений.

***Познавательные УУД*:**

* моделирование – преобразование объекта из чувствен­ной формы в модель, где выделены существенные характе­ристики объекта (пространственно-графическая или знаково-символическая);
* анализ объектов с целью выделения признаков (суще­ственных, несущественных);
* синтез – составление целого из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
* выбор оснований и критериев для сравнения, сериации, классификации объектов;
* подведение под понятие;
* установление причинно-следственных связей;
* построение логической цепи рассуждений.

***Коммуникативные УУД*:**

* аргументирование своей точки зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
* выслушивание собеседника и ведение диалога;
* признавание возможности существования различных точек зрения и права каждого иметь свою.

***Предметные результаты***:

Обучающийся:

− Овладеет этапами проектной деятельности.

− Научится использованию различных методов создания, отладки и корректировки проектов в среде Scratch.

− Научится использованию инструментов встроенного графического редактора (создание и сохранение изображений и спрайтов).

− Получит знания основных видов и задач творческих олимпиад по креативному программированию.

**Ожидаемые результаты:**

В результате реализации программы, обучающиеся получат возможность

знать:

− устройство программной среды Scratch (базовый уровень);

− процесс составления алгоритма, блок-схемы;

− алгоритм составления проекта; − процесс составления алгоритма программы и её отладки; уметь:

− рисовать в графическом редакторе Scratch и с использованием команд блока Перо;

− использовать сенсоры, списки, переменные, случайные числа для составления скриптов; − создавать алгоритмы словесно и на языке Scratch;

− создавать мультфильмы и игры;

− создавать свои проекты и уметь презентовать их;

− работать в команде.

Результативность освоения программы отслеживается на практических занятиях, на которых выполняются определенные задания и после каждого изученного раздела заполняется диагностическая карта успеваемости.

***Продолжительность и этапы реализации программы***

Программа рассчитана на 68 часов в год (2 часа в неделю, 68 учебных недели). Предназначена для обучающихся 2-3 классов МБОУ «СОШ а. Икон-Халк им. Х. С-Г. Кумукова»

**Материально-техническое обеспечение**

*Перечень оборудования:*

− учебный кабинет, учебные столы, стулья ;

− проектор, экран;

− компьютеры с установленной операционной системой Linux или Windows для каждого обучающегося и для педагога.

*Перечень инструментов*:

-программы Adobe AIR и Scratch 2 Offline Editor, (бесплатно скачиваются с https://scratch.mit.edu ).

**Cодержание программы**

**I. Интерфейс программы Scratch (2 ч).**

**1.** **Введение. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch.  
Теория.** История создания среды Scratch. Основные базовые алгоритмические конструкции (линейные алгоритмы, с условным оператором, циклического типа с предусловием и постусловием) и их исполнение в среде Scratch. Понятие исполнителя, алгоритма и программы, их назначение, виды и использование. Виды управления исполнителем. Способы записи алгоритма. Основные характеристики исполнителя. Система команд исполнителя. Понятие проект, его структура и реализация в среде Scratch. Основные компоненты проекта Scratch: спрайты и скрипты. Принцип создания анимации и движения объектов. Листинг программы. Сцена. Текущие данные о спрайте. Стиль поворота. Закладки. Панель инструментов, Новый спрайт. Координаты мышки. Режим представления. Окно скриптов. Окно блоков.Блоки стека. Блоки заголовков. Блоки ссылок. Самодостаточные и открытые скрипты.

**II. Начало работы в среде Scratch (4 ч).**

**2.** **Сцена. Редактирование фона. Добавление фона из файла.**

**Теория.** Сцена. Широта и высота сцены. Текущие координаты объекта. Редактирование текущего фона. Вставка нового фона из файла. Вставка стандартного фона из библиотечного модуля среды. Рисование фона в графическом редакторе. Создание нескольких фонов в одной сцене.

**Практика.** Создание фона сцены на выбранную учащимся тему.

**3.** **Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.**

**Теория.** Стандартный объект. Спрайты. Список спрайтов. Редактор рисования для создания новых спрайтов. Инструменты рисования (кисточка, линия, текст, эллипс, ) и редактирования объекта (ластик, заливка, поворот, выбор, печать, пипетка). Центрирование костюма. Масштабирование спрайта. Загрузка на сцену спрайтов из стандартной коллекции среды Scratch. Вставка спрайтов из файлов форматов JPG, BMP, PNG, GIF. Выбор случайного спрайта. Удаление спрайтов.

**Практика.** Создание фона сцены и прорисовка основных спрайтов для Scratch-истории.

**III. Основные скрипты программы Scratch (27 ч).**

**4.** **Синий ящик – команды движения. Темно-зеленый ящик – команды рисования.**

**Теория.** Команды – *идти*; *повернуться направо* *(налево)*; *повернуть в направлении*; *повернуться к; изменить х (у) на; установить х (у) в; если край, оттолкнуться.* Принципиальное различие действия команд *идти в* и *плыть в*. Назначение сенсоров *положение х, положение у* и *направлении.* Команды – *очистить, опустить перо, поднять перо, установить цвет пера, изменить цвет пера на, установить цвет пера, изменить тень пера, установить тень пера, изменить размер пера на, установить размер пера, печать*.

**Практика.** Создание программ для передвижения спрайтов по сцене. Создание программ для рисования различных фигур.

**5. Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов.**

**Теория.**Костюмы спрайта. Копирование и редактирование костюма спрайта с помощью редактора рисования. Переупорядочивание костюмов. Команды – *перейти к костюму, следующий костюм, говорить…в течении…секунд, сказать, думать, думать…секунд, изменить ….эффект на, установить эффект…в значение, убрать графические эффекты, изменить размер на, установить размер, показаться, спрятаться, перейти в верхний слой, перейти назад на…1 слоев.* Назначение сенсоров *костюм* и *размер.* Понятие раскадровки движения. Изменение костюма спрайта для имитации движения.

**Практика.** Создание программы для управления внешним видом объекта. Создание Scratch-историй с имитацией хождения и движения объектов.

**6. Желтый ящик – контроль. Лиловый ящик – добавление звуков.**

**Теория.** Кнопка с зеленым флажком и ее назначение. Управление последовательностью выполнения скриптов. Понятие управляющих сообщений. Команды – *передать, передать и ждать, когда я получу.* Скрипты для создания условных конструкций программы – *если, если…или*. Скрипты для управления циклами – *всегда,* *повторить,* *всегда, если, повторять до.*. Команды – *когда клавиша…нажата,* *когда щелкнут по, ждать…секунд, ждать до,*  *остановить скрипт, остановить все.* Загрузка звуков из стандартной коллекции и из файлов жесткого диска. Запись звука через микрофон. Принципиальная разница работы команд *играть звук* и *играть звук до завершения.* Команды – *остановить все звуки, барабану играть…тактов, оставшиеся…тактов, ноту…играть…тактов, выбрать инструмент, изменить громкость, установить громкость, изменить темп на, установить темп.* Назначение сенсоров *громкость* и *темп*.

**Практика.** Создание программ с элементами управления объектом. Озвучивание Scratch-историй.

**7. Использование в программах условных операторов.**

**Теория.** Базовая конструкция ветвление, назначение, виды (полная и неполная форма). Понятие условия. Изменение порядка выполнения скриптов в зависимости от условия. Разветвление листинга программы. Скрипты условных операторов. Использование неполной формы ветвления в системе Scratch.

**Практика.** Создание программ с изменением последовательного выполнения скриптов при наличии условий.

**8. Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий.**

**Теория.** Циклы с фиксированным числом повторений. Заголовок цикла. Тело цикла. Циклы с условным оператором. Заголовок цикла. Тело цикла. Предусловие и постусловие. Зацикливание.

**Практика.** Создание программ с использованием циклов с фиксированным числом повторений. Создание программ с использованием циклов с предусловием и постусловием.

**9. Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления.**

**Теория.** Числа. Стринги. Логические величины. Логические выражения. Арифметические операции. Логические операции. Операции сравнения. Команды для работы со стрингами – *слить, буква…в, длинна строки.* Команда *выдать случайное от…до.* Использование арифметических и логических блоков в листинге программы. Просмотр полученного результата.

**Практика.** Создание программ с использованием операций сравнения данных. Создание программ с использованием арифметических данных и логических операций.

**10. События. Оранжевый ящик – переменные.**

**Теория.** События в проектах Scratch. Понятие переменных и необходимость их использования в листинге программы. Глобальные и локальные переменные. Имя переменной и правила его формирования. Команды для переменных - *поставить…в, изменить…на, показать переменную, спрятать переменную.* Удаление переменных. Создание счетчиков с помощью переменных.

**Практика.** Разработка сценария Scratch-историй с несколькими событиями. Создание проектов с использование глобальных и локальных переменных.

**11. Списки.**

**Теория.** Создание списков и необходимость их использования в проектах Scratch. Добавление в список данных. Удаление данных из списка. Удаление списка. Команды работы со списками – *добавить…к*, *удалить…из, поставить…в…из, заменить элемент…в…на, элемент…из, длина списка*.

**Практика.** Создание программ-тестов по принципу сравнения данных из нескольких списков.

**12. Голубой ящик – сенсоры. Ввод-вывод данных.**

**Теория.** Понятие сенсора. Правила применения и область действия команд *касается, касается цвета* и *цвет. касается.* Функционал команды *спросить…и**ждать.* Сенсоры *мышка по х, мышка по у, мышка нажата?, клавиша…нажата?, расстояние до, перезапустить таймер.* Сенсоры, значение которых можно выводить на экран – *ответ, таймер, громкость, громко?, …значение сенсора* и *сенсор….* Необходимость ввода данных для их обработки в программе. Ввод данных с помощью команды *спросить.* Вывод конечного результата обработки с помощью команд *говорить* и *сказать* .

**Практика.** Создание проектов с использованием значений сенсоров и команды *спросить.* Создание программ для обработки данных пользователя с выводом на экран конечного результата.

**IV. Работа с несколькими объектами. Синхронизация их работы (10 ч).**

**13. Последовательность и параллельность выполнения скриптов.**

**Теория.** Последовательные и параллельные потоки в программах Scratch. Одновременная и попеременная работа нескольких исполнителей.

**Практика.** Создание Scratch-историй с одновременной и попеременной работой нескольких исполнителей.

**14. Взаимодействие между спрайтами. Управление через обмен сообщениями.**

**Теория.** Решение проблемы появления новых исполнителей только после того, как старые исполнители выполнили свои действия. Взаимодействие спрайтов с неподвижными объектами с помощью команд *касается* и *касается цвета.* Взаимодействие спрайтов с помощью команд *передать* и *когда я получу.* Использование сообщений для создания событий.

**Практика.** Создание Scratch-историй с взаимодействием нескольких исполнителей и неподвижных объектов. Создание Scratch-историй с взаимодействием нескольких исполнителей.

**V. Использование программы Scratch для создания мини-игр (21 ч).**

**15.** **Виды компьютерных игр. Алгоритмическая разработка листинга программы.**

**Теория.** Компьютерные игры – вред или польза. Виды компьютерных игр. Этапы разработки игр программистами.

**Практика.** Алгоритмическая разработка проекта, запись на естественном языке событий и точек взаимодействия героев будущей игры.

**16.** **Разработка базовых спрайтов для игры. Формирование базовых скриптов.**

**Теория.** Логика создания персонажей для игры. Перевод алгоритма, написанного на естественном языке, в коды Scratch.

**Практика.** Разработка и создание основных спрайтов и их костюмов для будущей игры. Разработка скриптов для спрайтов и объектов.

**17. Синхронизация работы скриптов для разных спрайтов.**

**Практика.** Доработка основного листинга программы с целью установления связей между спрайтами. Тестирование и отладка программы.

**18. Переход из одной сцены в другую. Создание интерфейса игры.**

**Теория.** Односторонний (без возможности вернуться назад) переход из одного пространства в другое. Понятие интерфейса. Элементы интерфейса. Основные принципы дизайна интерфейсов. Обратная связь. Необходимые элементы меню.

**Практика.** Создать программу для перемещения объектапо игровой карте и разработать интерфейс для Scratch-проекта.

**19. Сообщество Scratch в Интернете. Просмотр и публикация проектов.**

**Теория.** Правила работы в сети.Интернет-сообщества. Сообщество Scratch. Регистрация на сайте. Использование заимствованных кодов и объектов. Авторские права. Публикация проектов Scratch.

**Практика.** Регистрация на сайте сообщества Scratch. Просмотр проектов сообщества и публикация собственных проектов.

**VI. Разработка творческого проекта (4 ч)**

**20.** **Разработка и защита творческого проекта.** Разработка и создание программы с использованием подготовленных материалов. Тестирование и отладка проекта. Защита проекта**.**

**Календарно – тематическое планирование по внеурочной деятельности «Увлекательное программирование в среде Scratch»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **НАИМЕНОВАНИЕ ТЕМЫ** | **ТЕОРИЯ** | **ПРАКТИКА** | **ВСЕГО** |
| **I. Интерфейс программы Scratch (2 ч)** | | | | |
| 1 | Введение. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch. | 1 | 1 | 2 |
| **II. Начало работы в среде Scratch (4 ч)** | | | | |
| 2 | Сцена. Редактирование фона. Добавление фона из файла. | 1 | 1 | 2 |
| 3 | Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов. | 1 | 1 | 2 |
| **III. Основные скрипты программы Scratch (27 ч)** | | | | |
| 4 | Синий ящик – команды движения. Темно-зеленый ящик – команды рисования. | 0,5 | 2,5 | 3 |
| 5 | Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов. | 0,5 | 2,5 | 3 |
| 6 | Желтый ящик – контроль. Лиловый ящик – добавление звуков. | 0,5 | 2,5 | 3 |
| 7 | Использование в программах условных операторов. | 0,5 | 2,5 | 3 |
| 8 | Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий. | 0,5 | 2,5 | 3 |
| 9 | Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления. | 0,5 | 2,5 | 3 |
| 10 | События. Оранжевый ящик – переменные. | 0,5 | 2,5 | 3 |
| 11 | Списки. | 0,5 | 2,5 | 3 |
| 12 | Голубой ящик – сенсоры. Ввод-вывод данных. | 0,5 | 2,5 | 3 |
| **IV. Работа с несколькими объектами. Синхронизация их работы (10 ч)** | | | | |
| 13 | Последовательность и параллельность выполнения скриптов. | 1,5 | 2,5 | 4 |
| 14 | Взаимодействие между спрайтами. Управление через обмен сообщениями. | 2 | 4 | 6 |
| **V. Использование программы Scratch для создания мини-игр (21 ч)** | | | | |
| 15 | Виды компьютерных игр. Алгоритмическая разработка листинга программы. | 1 | 2 | 3 |
| 16 | Разработка базовых спрайтов для игры. Формирование базовых скриптов. | 1 | 4 | 5 |
| 17 | Синхронизация работы скриптов для разных спрайтов. | 0 | 5 | 5 |
| 18 | Переход из одной сцены в другую. Создание интерфейса игры. | 0,5 | 5,5 | 6 |
| 19 | Сообщество Scratch в Интернете. Просмотр и публикация проектов. | 0,5 | 1,5 | 2 |
| **VI. Разработка творческого проекта (4 ч)** | | | | |
| 20 | Разработка и защита творческого проекта | 0 | 4 | 4 |
| **Итого:** | | **14** | **54** | **68** |

**Список используемой литературы**

1. Базисный учебный план общеобразовательных учреждений Российской Федерации. М.: Просвещение, 2008. 25 с. (Стандарты второго поколения).
2. Герасимова Т. Б. Организация проектной деятельности в школе. // Преподавание истории в школе. 2007. № 5. С. 17–21.
3. Краля Н. А. Метод учебных проектов как средство активизации учебной деятельности учащихся: Учеб-но-методическое пособие / Под ред. Ю. П. Дубенского. Омск: Изд-во ОмГУ, 2005. 59 с.
4. Матвеева Н. В. Информатика и ИКТ. 3 класс: методическое пособие / Н. В. Матвеева, Е. Н. Челак, Н. К. Конопатова, Л. П. Панкратова. М.: БИНОМ. Лаборатория знаний, 2009. 420 с.
5. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008. 61 с.
6. Скретч [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: http://letopisi.ru/index.php/Скретч
7. Хохлова М. В. Проектно-преобразовательная деятельность младших школьников. // Педагогика. 2004. № 5. С. 51–56.
8. Школа Scratch [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: http://letopisi.ru/index.php/Школа\_Scratch
9. Scratch | Home | imagine, program, share [cайт]. URL: http://scratch.mit.edu

Scratch | Галерея | Gymnasium №3 [cайт]. URL: http://scratch.mit.edu/galleries/view/54042