



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГПУ»)

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Изучение основ программирования на языке Scratch во внеурочной деятельности по
информатике

Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование (с двумя профилями
подготовки)


Направленность программы бакалавриата
«Информатика. Экономика»

Проверка на объем заимствований:
92,93 % авторского текста

Работа рекомендована к защите
рекомендована/не рекомендована

«07» июня 2018 г.

Зав. кафедрой И, ИТ и МОИ


 Рузаков А.А.

Выполнил:

Студент группы ЗФ-513/094-5-1
Ашмарин Виктор Евгеньевич


Научный руководитель:

Ст. преподаватель кафедры ИИТиМОИ


Боровская Елена Владимировна

Челябинск
2018



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Изучение основ программирования на языке Scratch во внеурочной деятельности по
информатике
Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование (с двумя профилями
подготовки)
Направленность программы бакалавриата
«Информатика. Экономика»

Проверка на объем заимствований:
_____ % авторского текста

Работа _____ к защите
рекомендована/не рекомендована

« ___ » _____ 20__ г.
Зав. кафедрой И, ИТ и МОИ

_____ Рузаков А.А.

Выполнил:
Студент группы ЗФ-513/094-5-1
Ашмарин Виктор Евгеньевич

Научный руководитель:
Ст. преподаватель кафедры ИИТиМОИ

Боровская Елена Владимировна

Челябинск
2018

Оглавление

Введение.....	3
ГЛАВА 1. Особенности изучения основ программирования.....	5
1.1. Программные среды и языки программирования для изучения основ программирования	5
1.2. Описание программной среда «Scratch»	19
1.3. Особенности изучения программирования в основной школе.....	23
Вывод по 1 Главе.....	30
ГЛАВА 2. Разработка факультативного курса «Основы программирования на языке Scratch» для 6 классов	31
2.1 Пояснительная записка факультативного курса «Основы программирования на языке Scratch»	31
2.2 Программная поддержка курса	49
2.3 Апробация результатов исследования в школе	51
Вывод по 2 Главе.....	52
Заключение	53
Список литературы	54
Приложение	57

Введение

С каждым днем растет потребность в высоко квалифицированных специалистах для разработки программного обеспечения, области информационных технологий. Но тем не менее, школа делает акцент на обучение пользовательскому подходу работы с компьютером, и не обучает алгоритмическому мышления. В школе отсутствует системный подход к изучению программирования. В основу обучения ложится изучения языка программирования, а не основы разработки программного обеспечения, которые в дальнейшем нужно использовать в решении.

При изучении программирование, учащиеся лучше осознают работу компьютера, его возможности, границы применения, что помогает в решении творческой, но в тоже время, точно сформулированной задачи. Всем, без исключения, учащимся это помогает в решении задача и развитии навыков мышления. Без владения умением в решении задач практически невозможно оставаться культурным человеком и являться полноценным, востребованным членом общества. Ученики, вне зависимости от возраста, получают уникальную возможность поэтапно развивать свои логико-алгоритмические способности, способности в области точных наук. Иначе говоря, основы программирования являются не только наукой или учебной дисциплиной в её обычном понимании, но и средством развития личности, способностей детей, побуждает интеллектуальные задатки освоения других дисциплин.

Цель работы: разработать факультативный курс «Основы программирования на языке Scratch» для 6 классов.

Объект исследования: изучение программирования во внеурочной деятельности.

Предмет исследования: процесс обучения программированию в программной среде Scratch.

В соответствии с целью работы были поставлены следующие **задачи:**

1. Изучить формы и методы обучения основам программирования во внеурочной деятельности.
2. Разработать факультативный курс по программированию в среде Scratch 6 классов.
3. Разработать программно-методическую поддержку курса в виде электронного пособия.

Гипотеза: если в программу общего образования, в качестве внеурочной деятельности, включить вопросы, связанные с умением программировать это будет способствовать овладению учащимися логического и алгоритмического мышления.

ГЛАВА 1. Особенности изучения основ программирования

1.1. Программные среды и языки программирования для изучения основ программирования

Учебные языки программирования – это языки программирования и среды программирования, предназначенные для введения в основы алгоритмизации, программирования и развития логико-алгоритмического (операционного) мышления у детей. На сегодняшний день существует множество различных языков программирования – начиная от самых простых (тех, на которых обучаются дети младших классов), и заканчивая мощными инструментами разработки, как Java, C#, C++, OpenGL, JavaScript, Python, PHP, Java и т.д. Однако причисленные языки программирования не подходят детям, которые только начинают изучать программирование, так как у них еще не сформировано абстрактное мышление. Языки программирования, разработанные специально для школьников, учитывают их психофизические и интеллектуальные особенности.

Некоторые из них:

1. «Guido van Robot»

«Guido van Robot» – язык программирования с простым синтаксисом (рис. 1), помогает легко понять учащимся сущность процедур и циклов. Одно из достоинств - это мгновенная визуальная связь. Данный язык хорошо подходит для изучения базовых концепция программирования, используемых в любом высокоуровневом языке программирования. Суть Guido van Robot в управлении роботом, который представлен в виде треугольника, он перемещается по миру содержащем в себе улицы, проспекты, звуки и стены которые робот может собирать или устанавливать. Все его действия руководствуются программой, полностью написанной пользователем. К

минусам можно отнести необходимость строгого соблюдения синтаксиса и отсутствие русского языка [8].

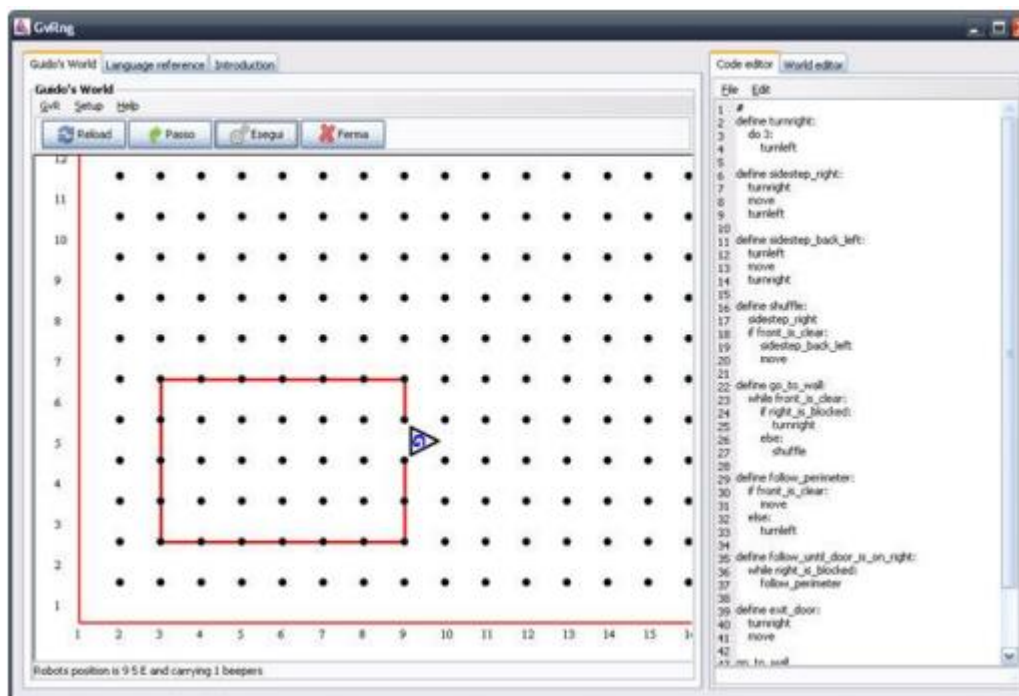


Рис. 1. Guido van Robot

2. «Scratch»

«Scratch» - визуальная объектно-ориентированная среда программирования (рис. 2), разработанная для обучения детей в возрасте от 6 – 14 лет. Он подойдет детям, которые уже уверенно работают и выполняют привычные и повторяющиеся алгоритма, так и для тех, кто только начинает с ними знакомиться. В Scratch можно создавать интерактивные проекты, составляя блоки алгоритмов и присваивать их каждому объекту. Программа включает большую библиотеку животных, героев, различных предметов на любой вкус. Основная цель - познакомить детей с понятиями программирования в ходе создания проектов. Существует большое количество бесплатных методических поддержек, которые помогут ребенку при большом желании освоить программу самостоятельно, без участия родителей и учителей. Так же существует международное сообщество пользователей Scratch, что дает возможность не только создавать проекты, но

и делиться новыми идеями и разработками по всему миру. Еще к плюсам можно отнести отсутствие строгого синтаксиса, все блоки представлены графически и имеют очень простой интуитивно понятный синтаксис [12].

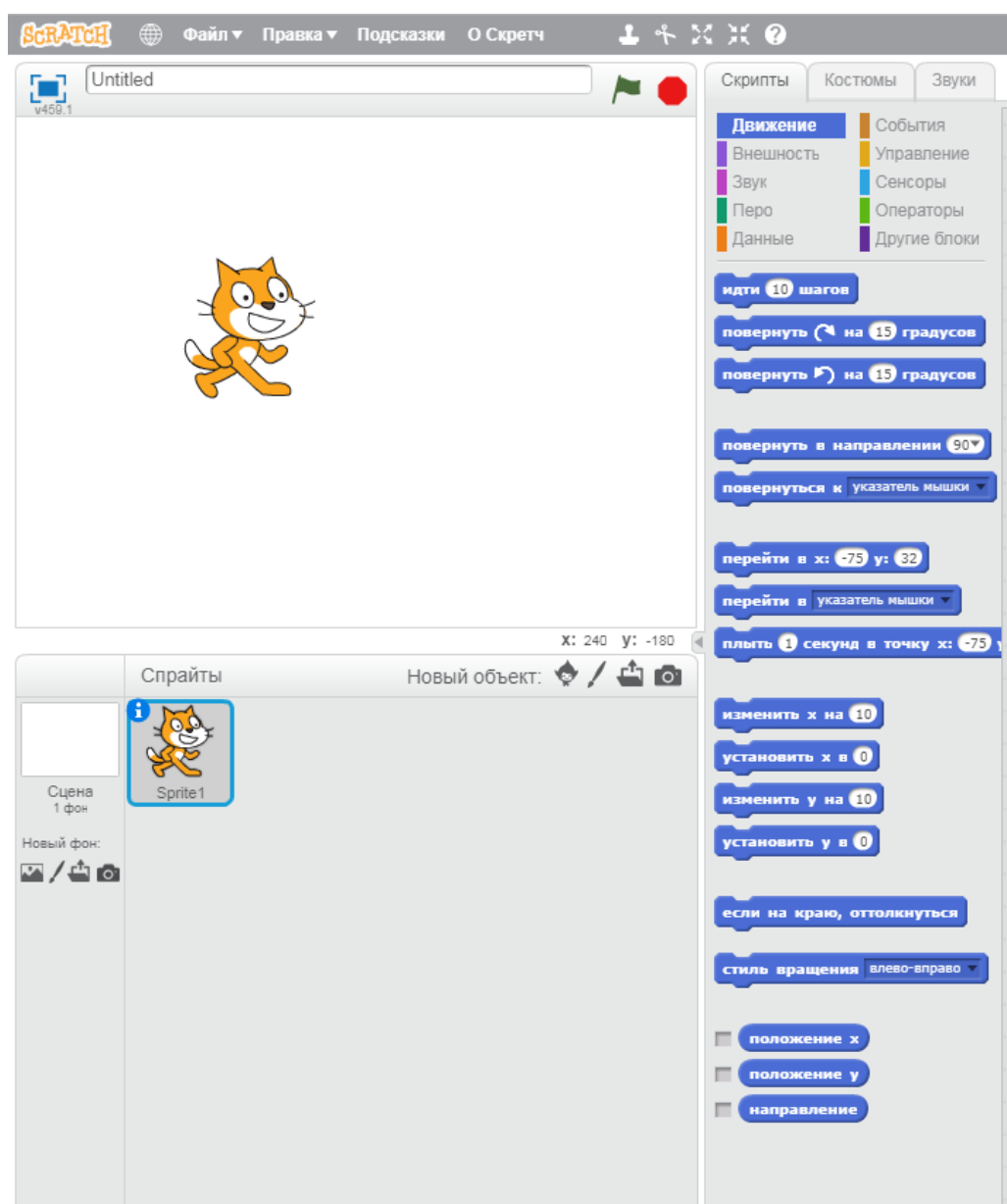


Рис. 2. Scratch.

3. «Lightbot»

«Lightbot» - является средой программирования, которая напоминает головоломку (рис. 3). Пользователю необходимо управлять роботом программируя его движение для выполнения простых задач – нужно преодолеть все преграды и добраться до цели. Программа создается по средствам переноса подходящих команд в клетки памяти робота. Хотя Lightbot

кажется простой, но при ее помощи можно сформировать операционное мышление, понятие о программе, цикле, процедурах и т.п. Анализ тематических форумов говорит о том, что данная среда популярна среди программистов, не говоря уже о детях школьного возраста. К минусам описанной среды можно отнести необходимость наличия подключения к сети интернету [17].

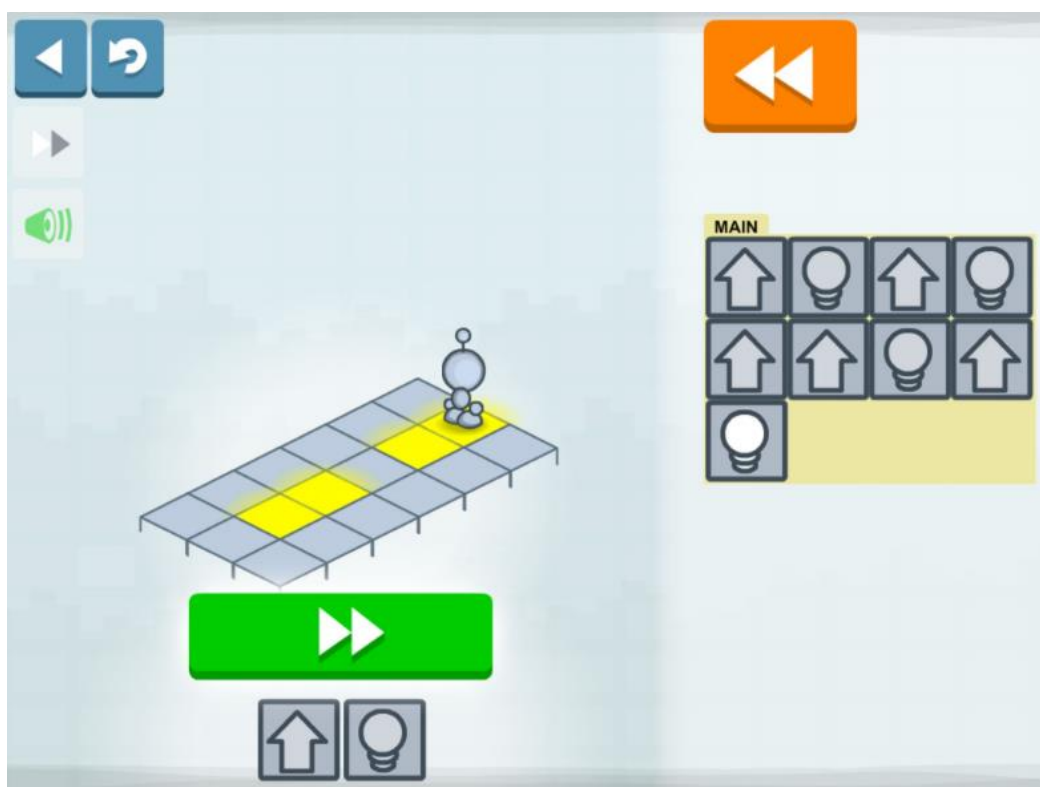


Рис. 3. Lightbot

4. «Alice»

«Alice» - является средой программирования (рис. 4), которая предназначена для учащихся 8-12 лет. Она одна из самых увлекательных программ, для изучения объектно-ориентированного программирования. Alice это интерактивная среда разработки, работая в которой, в игровой форме можно создавать, изменять, анимировать и запрограммировать поведение игрового мира, его персонажей, объектов, настроить камеру и т.п. Управление представлено двумя режимами: редактирования кода и визуальный. В режиме

редактирования можно создавать процедуры, алгоритмы и изменять классы. Исполнение алгоритмов протекает, как и в интерпретируемом языке программирования т.е. буквально и последовательно. Во втором режиме есть возможность настраивать различные ракурсы камеры, добавлять трехмерные модели (в программу входит весьма большая библиотека с животными, людьми, транспортными средствами и различными объектами окружающего мира), настраивать сцену и собирать композицию. Каждая модель интерпретируется как объект, к которому, в дальнейшем, пишется алгоритм действий. В среде присутствует русскоязычный интерфейс, благодаря которому ее освоение не вызовет серьезных затруднений. Так же в интернете есть не малое количество обучающих материалов и готовых проектов [15].



Рис. 4. Alice

5. «Logo»

«Logo» - представляется из себя высокоуровневый язык программирования (рис. 5), который предназначен для детей младше школьного и дошкольного возрастов. Используется в работе системы ЛогоМиры. Например, с черепахой или другим персонажем, которого может

выбрать ребенка, на много быстрее и лучше понять логику работы программы, из-за присутствия направления, которое можно изменить. Обучение персонажа каким-нибудь действия приводит к осмыслению своих мыслей и действий, что гарантирует более быстрый и эффективный процесс обучения программированию. Основным недостатком является отсутствие бесплатных версий, а следственно и не большое количество учебных заведений, которые могут позволить приобрести лицензию на данную среду [10].

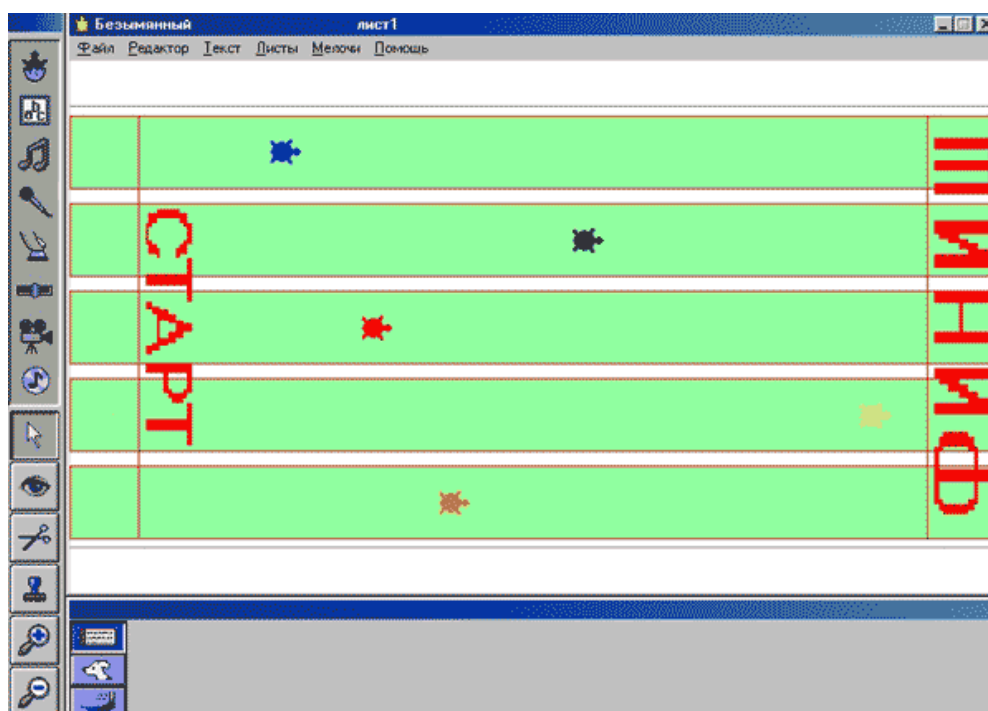


Рис. 5. Logo

6. «Baltie»

«Baltie» - графическая среда программирования которая предназначена для учащихся 6 – 16 лет (рис. 6). Не требует первоначальных знаний, что является весьма привлекательным для большого количества людей из разных стран для изучения программирования. Программа предлагает управлять, при помощи определенного набора кнопок, спрайтом волшебника. Набор включает: цифры, изменение движения, пользовательский ввод, математические и логические функции, добавление анимации, управление координатами, объявление переменных, работа с файлами, работа со строками

и преобразование типов данных, управление структуры (процедуры, циклы). Последние версии уже поддерживают язык программирования C#. Описанная среда программирования используется во многих странах, за исключением России [5].

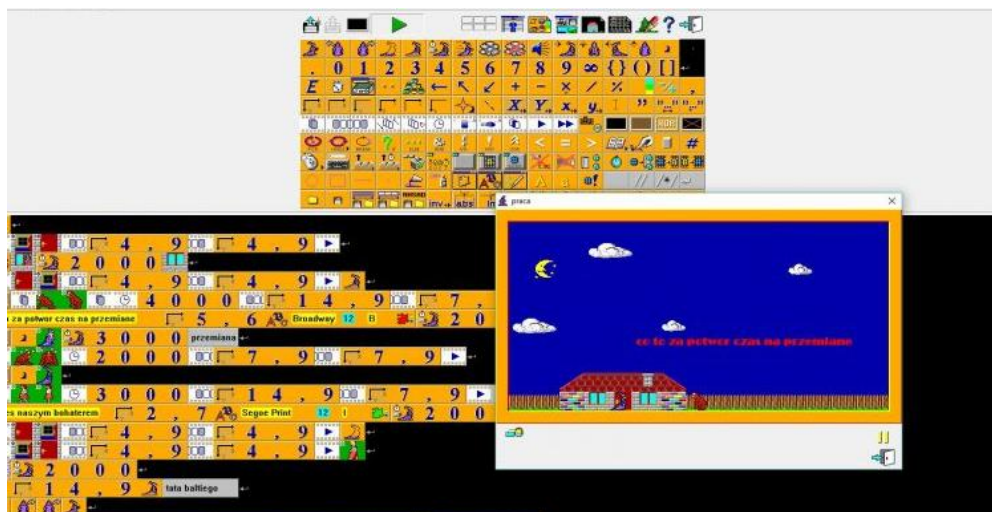


Рис. 6. Baltie

7. «Greenfoot»

«Greenfoot» - среда программирования созданная на языке Java и выступает как конструктор игр (рис. 7). Программирование в этой среде рекомендуется от 10 – 14 лет. В ней можно создавать собственный мир, изменять расположение элементов, добавлять актеров, программировать их поведение при помощи доступных различных кнопок и визуальных диалоговых окон. Интерфейс удобен и прост, и имеет имеет вся для удобной разработки. Главным недостатком является отсутствие русскоязычного интерфейса [7].

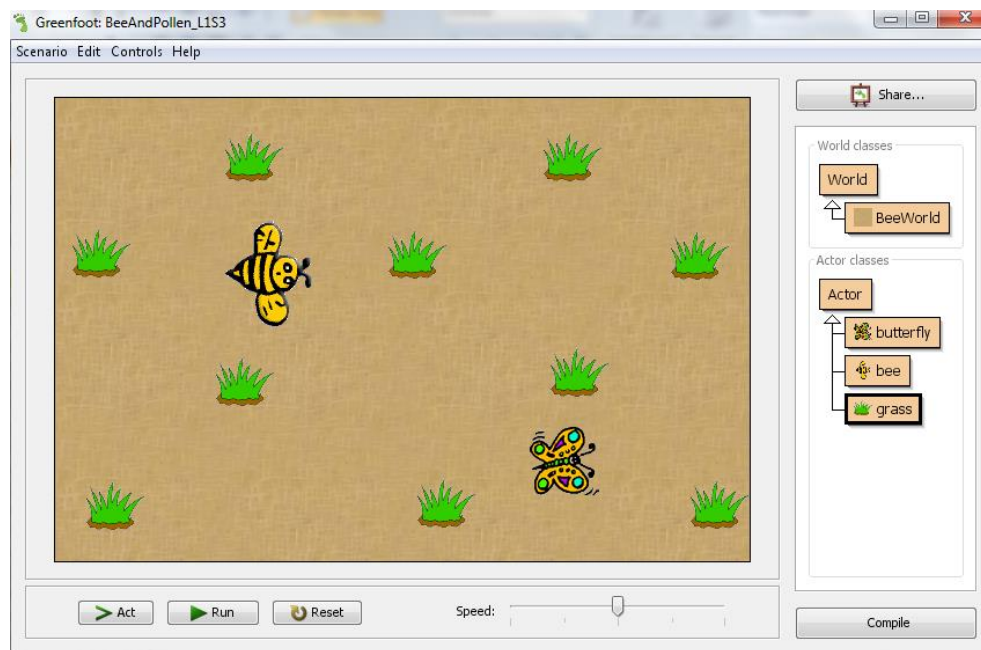


Рис. 7. Greenfoot

8. «Little Wizzard»

«Little Wizzard» - сред программирования предназначенная для изучения основ программирования (рис. 8). Для разработки программ используется лишь компьютерная мышь, благодаря этому дети получают умение составлять программы, но и узнают такие основы как ветвление, переменная, логические блоки, условие, выражения. Процесс составления программы весьма прост, все элементы языка представлены в интуитивно понятных блоках с символами [9].

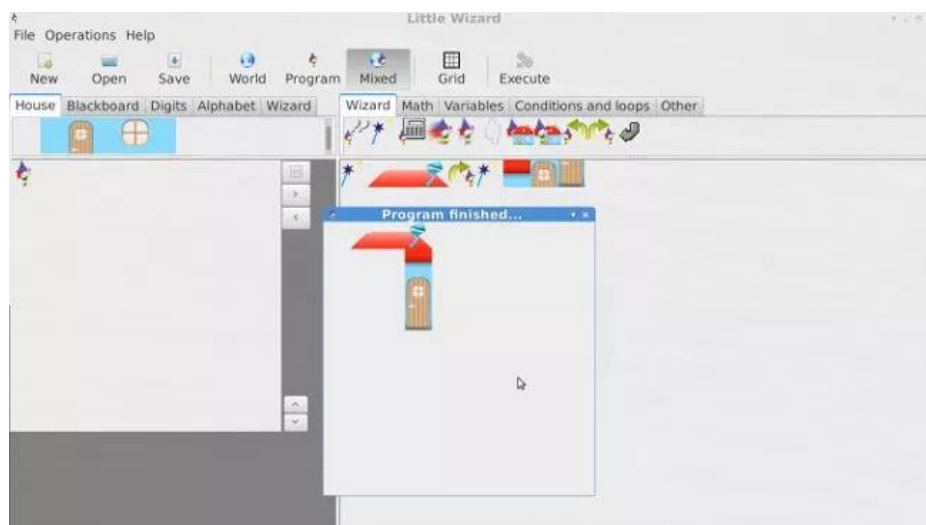


Рис. 8. Little Wizzard

9. «Etoys»

«Etoys» - является интерактивной компьютерной средой предназначенной для детей от 9 до 12 лет (рис. 9). Имеет весьма мощные возможности мультимедиа для создания программ и может работать на различных платформах. Основная идея заключается в том, что все окружающее представляет из себя объекты, т.е. программа это не просто код, который компилируется а потом исполняется как это устроено в других учебных средах для изучения программирования, а представляет из себя возможность общения с данными объектами и изменять их. Etoys представляет из себя простую и в тоже время достаточно мощную среду для изучения методологии работы с моделью объекта, с объектами, которые были созданы учащимися и могут включать в себя html, обычный текст, трехмерные и двумерные изображения, видео, звук и в том числе взаимодействовать в режиме реального времени с другими пользователями. Программа помогает изучить не только основы программирования, но и использовать для моделирования физические явления, решение логически и математических задач. Etoys это мульти язычная среда, благодаря чему получила большое внимание со стороны Японии, Южной Америки, Индии, США но в России, к сожалению, большой популярности не получила [6].

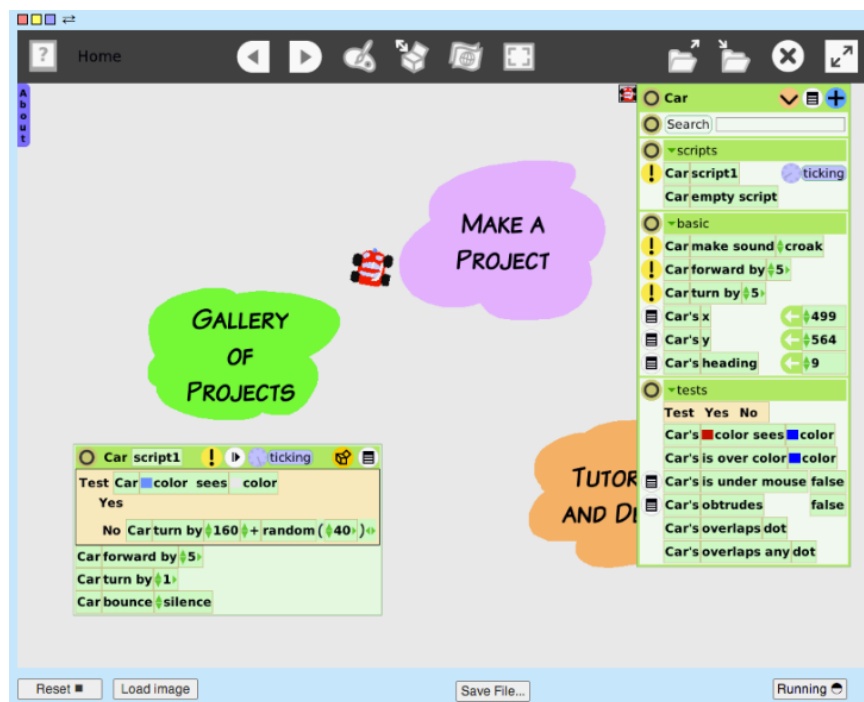


Рис. 9. Etoys

10. «Squeak»

«Squeak» - представляет собой открытую среду программирования, диалект языка SmallTalk (рис. 10). Среда имеет множество возможностей – от создания простых мультимедийных платформ до веб-серверов. Код компилируется и выполняется виртуальной машиной Squeak, благодаря чему программа может работать практически под любой платформой. К недостаткам можно отнести наличие платных компонентов и не простая установка виртуальной машины [13].



Рис. 10. Squeak

11. «Karel», «Karel J.», «Karel ++», «Robot»

«Karel», «Karel J.», «Karel ++», «Robot» - представляют из себя среды программирования для учащихся от 10 – 14 лет (рис. 11). Смысл в составлении программы, благодаря которой можно управлять роботом. В Karel для управления роботом разработан собственный язык программирования – Karel. Уже в следующей версии, а именно Karel++, для программирования используется C++, в Karel j – Java, в Robot – Python. Что позволило разработчикам объединить вместе доступный и простой интерфейс с изучением не простых понятий высокоуровневых языков программирования. Среда распространяется бесплатно. К недостаткам можно отнести отсутствие русскоязычного интерфейса [16].

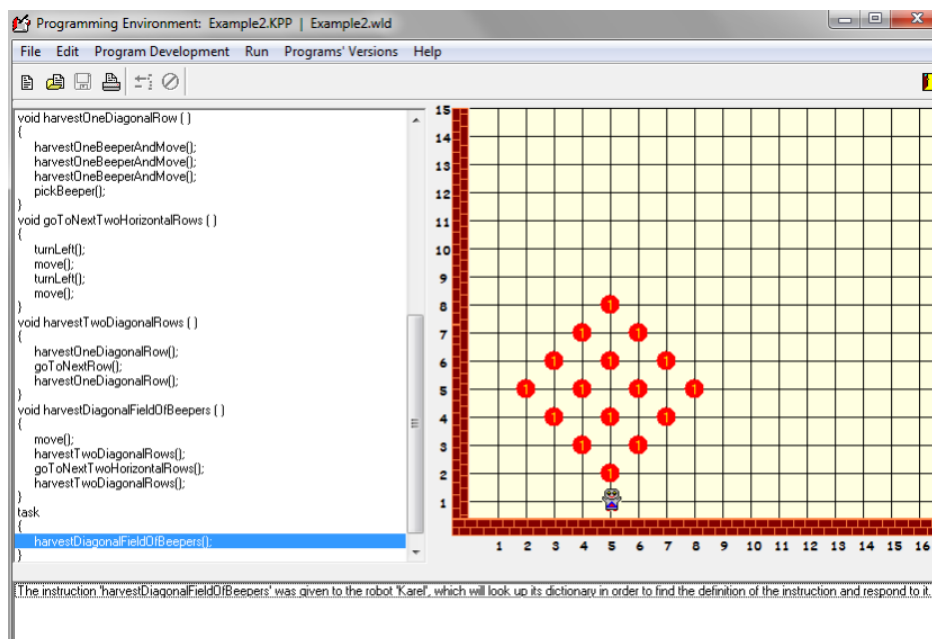


Рис. 11. Karel

12. «RoboMind»

«RoboMind» - среда для обучения основам программирования (рис. 12). Для работы используется достаточно простой интерпретируемый язык. Среда позволяет познакомиться с основами программирования, разобрать основы с сфере искусственного интеллекта и робототехники. В рамках обучения данной среды, управление происходит роботом, который выполняя алгоритмическую последовательно перемещается по доступной карте и собирает, осматривает, различные предметы, окружающие его, и отмечает пройденный по карте путь. В том числе программа направлена на изучение такого языка программирования как Robo, который используется для задания основных команд для движения робота. Программа распространяется бесплатно в случае домашнего использования, для всех остальных целей - на возмездной основе. В версию 4.0 была добавлена возможность экспортирования скриптов в RoboMind для программирования существующих роботов, таких как Lego Mindstorms NXT 2.0. К минусом можно отнести язык программирования Robo, который используется только в описанной среде программирования [11].



Рис. 12. RoboMind

13. Система «Исполнители»

Система «Исполнители» представляет собой интегрированную оболочку для начального обучения по теме "Алгоритмы и исполнители" в школьном курсе информатики (рис. 13). Исполнители (Робот, Чертёжник и Черепаха) выполняют программу, которая вводится в текстовом редакторе. Все действия исполнителя отображаются на экране. Существуют также широкие возможности для создания независимых программ, не использующих команды исполнителей. Система содержит возможность составления и решения разных типов задач по темам "Исполнитель и его команды", "Процедуры", "Функции", "Циклы", "Условные операторы", "Переменные", "Арифметические выражения", "Логические операции и логические переменные", "Глобальные переменные", "Операторы ввода и вывода", "Символьные строки", "Массивы", "Матрицы" и др. Имеется поддержка базовых конструкций языка Си. К плюсам можно отнести полностью русифицированный интерфейс и сам язык программирования. Встроенный отладчик с возможностью выполнения программ в пошаговом режиме с трассировкой процедур и функций [18].

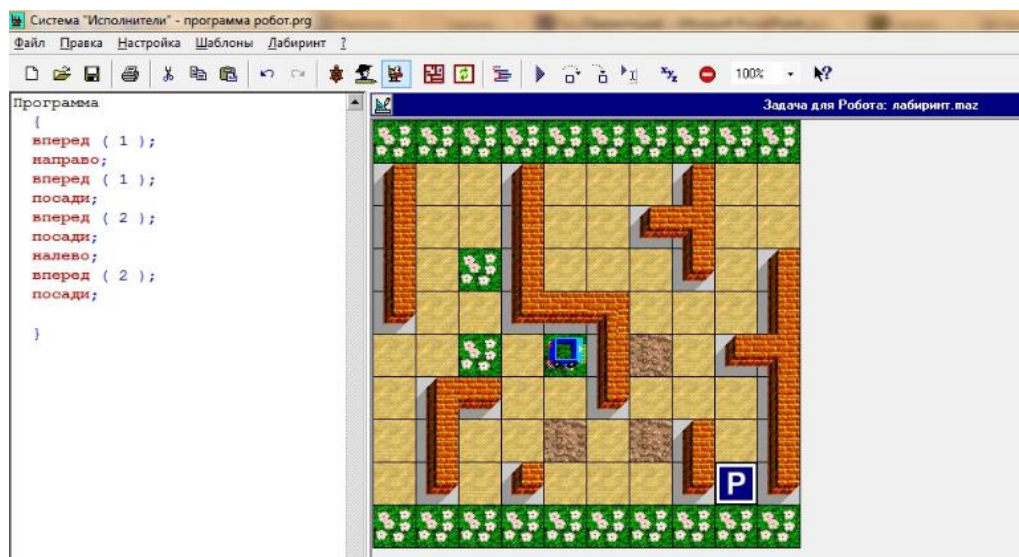


Рис. 13. Система «Исполнители».

14. КуМир

КуМир (Комплект Учебных МИРов) - система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе (рис. 14). В системе КуМир используется школьный алгоритмический язык с русской лексикой и встроенными исполнителями Робот и Чертёжник. При вводе программы КуМир осуществляет постоянный полный контроль ее правильности, сообщая на полях программы об всех обнаруженных ошибках. При выполнении программы в пошаговом режиме КуМир выводит на поля результаты операций присваивания и значения логических выражений. Это позволяет ускорить процесс освоения азов программирования. КуМир работает в операционных системах Windows или Linux [14].

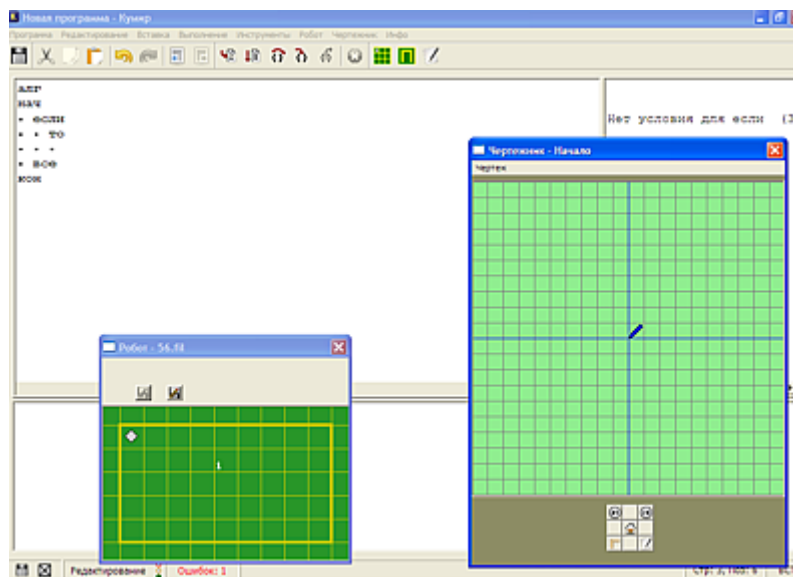


Рис. 13. КуМир.

На основе рассмотренных языков программирования и программных средств, можно утверждать, что у каждого средства есть свои плюсы и минусы. Достаточно большой выбор программных средств дает более расширенные возможности для основ программирования в игровой форме. Однако можно сделать вывод что из всех рассмотренных средств среда Scratch лучше остальных подходит для обучения основам программирования т.к. имеет полностью графический и интуитивно понятный и русскоязычный интерфейс. Не требует написания кода со строгим синтаксисом, но при этом очень мощная и гибкая среда для обучения и разработки свои учебных проектов и полностью интерактивный. Она способна заинтересовать основами программирования достаточно широкую массу учащихся благодаря красочности и простоте освоения.

1.2. Описание программной среда «Scratch»

Scratch представляет из себя среду для изучения основ программирования и имеет полностью графический интерфейс.

Среда разработана как инструмент для простого и интуитивного изучения программирования. Целью которой является помощь тем детям, у

которых отсутствует знания и опыт программирования и изучить принципы многопоточно, императивного и объектно-ориентированного программирования по средствам игры.

Основное окно для разработки среды Scratch состоит из нескольких частей и разделена на три колонки (рис. 14).

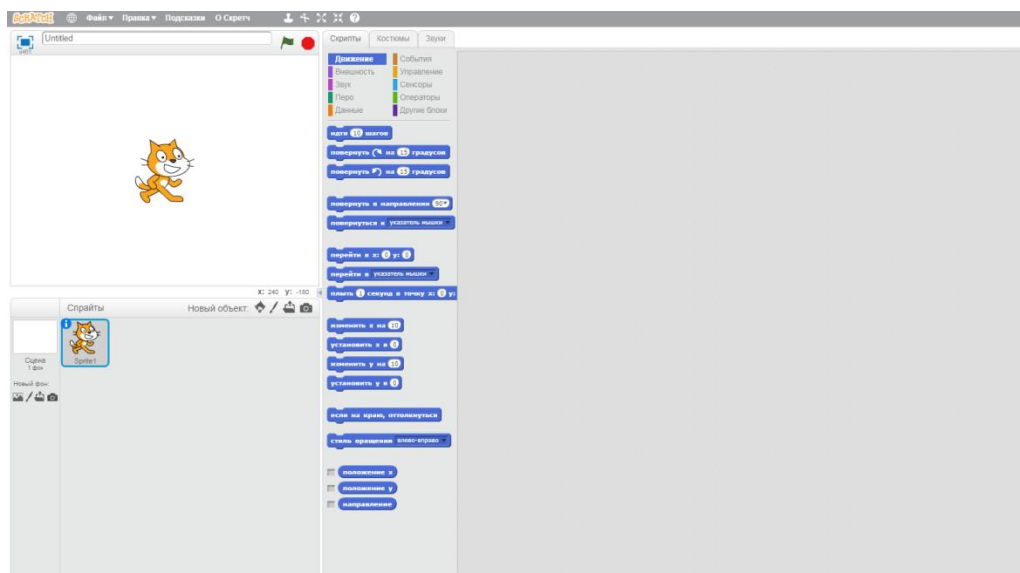


Рис. 14. Основное окно Scratch

Колонка слева содержит блоки. Блок представляет из себя кусочек программы в Scratch и может являться управляющей функцией, оператором, переменной, функцией. Сами блоки разделена 8 категорий:

- Sound – звуки;
- Sensing – регистрирует сигналы от устройства ввода;
- Looks – визуальные эффекты;
- Motion – управление движениями спрайта;
- Control – управляющая структура;
- Operations – логические, строковые, математических функции и операторы;
- Variables – векторные и скалярные переменные и операции;
- Pen – черчение линий.

В колонке в центре располагается текущий спрайт и его действия (скрипты). Спрайтами в Scratch называются объекты, которые ассоциируются с изображением и имеют набор переменных и действий.

Программа задается при соединении нескольких блоков последовательно, либо размещая в теле другого блока, например в цикле. У одного объекта возможны несколько скриптов, которые могут запускаться независимо друг от друга или в зависимости от таймера, полученной информации от другого объекта или действий пользователя.

Scratch иллюстрирует несколько парадигм:

- Объектно-ориентированная;
- Структурная;
- Многопоточная.

Синтаксис:

- `Set varname to value` (установить переменной значение) – присваивание значения переменной
- `«=»` - равенство
- `«<>»` - сравнение
- `If (condition) trueBlock` (если выражение в скобках принимает истинное значение, то выполняется инструкция внутри блока) – если то
- `if (condition) trueBlock else falseBlock` (если выражение в скобках принимает истинное значение, то выполняется инструкция внутри блока, иначе внутри блока `else` – от англ. иначе) – если то иначе
- `forever if (condition) loopBody` (выполняется инструкция внутри блока «forever» и если значение `condition` принимает истинное значение, то снова выполняется инструкция внутри блока «forever») - цикл с предусловием схож с циклом в Си `do...while`

- `repeat until (condition) loopBody` (повторять цикл до тех пор, пока выражение «condition» принимает истинное значение) - цикл с постусловием схож с циклом `while` в Си.

К плюсам можно отнести графический интуитивно понятный интерфейс, отсутствие строгого синтаксиса и среда представлена на русском языке. К минусам можно отнести обязательное, постоянное подключение к сети интернет.

1.3. Особенности изучения программирования в основной школе

ФГОС – федеральный государственный образовательный стандарт, включает в себя требования, которые обязательны для образовательных учреждений, имеющих аккредитацию и реализующих основные образовательные программы: начальное общее, основное общее, среднее полное общее, начальное профессиональное, среднее профессиональное, высшее профессиональное образования.

ФГОС гарантирует:

1. Целостность образовательного пространства Российской Федерации
2. Построение единой линии основных образовательных программ начального, общего, среднего полного, начального профессионального, среднего профессионального, высшего профессионального образования.

Воспитание – одного из главных составляющих образовательного процесса, равно как и обучение. Педагогическая задача состоит в воспитании компетентного, нравственного, инициативного, ответственного гражданина Российской Федерации. В соответствии с ФГОС основная образовательная программа реализуется образовательными учреждениями так же и через внеурочную деятельность [24].

Внеурочная деятельность является составной частью учебно-воспитательного процесса, а так же одной из форм организации свободного времени учащихся, направленная на оказании помощи ученику в преодолении возникших трудностей в освоении нового учебного материала и создании в учебной мотивации деятельности, интереса к дисциплине [24].

Для осуществления внешкольной деятельности, образовательные учреждения так же используют возможности образовательных организаций дополнительного образования детей. Для того, чтоб процесс обучения являлся непрерывным, во время каникул нужна организация отдыха детей и их оздоровление, например, летние школы или лагеря. В случае с детьми с

ограниченными способностями или одаренными, учебные планы лучше разрабатывать совместно с родителями и их детьми.

Целью внеурочной деятельности является – создание развития и раскрытия учеников, их интересов, основанных на свободном выборе, знакомство с нравственными ценностями, культурными традициями, создание условий для физического и интеллектуального развития учащихся.

Принципы организации внеурочной деятельности:

1. Гуманистический – учитывает в полной мере интересы, желания, потребности детей для организации внеурочной деятельности, так же создаются условия для формирования умений и навыков самореализации и рефлексии у учащихся;
2. Системность – создается системность внеурочной деятельности, где устанавливается взаимосвязи между:
 - Всеми участниками образовательного процесса: педагогами, учащимися, родителями;
 - Региональной, муниципальной, школьной, классной, индивидуальной системой дополнительного образования и обучения школьников;
 - Внеурочной и урочной деятельностью;
 - Основными компонентами образовательного процесса: оценочно-результативным, содержательно-деятельностным, оценочным.
3. Креативность. Задача, во внеурочное время развивать творческие способности учащихся, создание благоприятной среды для занятий по коллективным и индивидуальным программам;
4. Вариативность. Выделяется несколько методов, направлений, форм которые способствуют организации внеурочной деятельности, на основе которых дети имеют свободу выбора и добровольное участие той или иной деятельности.

5. Успешность и социальная зависимость. Стремление педагога направлено на удовлетворение потребностей ребенка в самореализации и достижения успеха. Обязательно, результаты учебной деятельности были лично ценными, а так же, значимыми для окружающих, например, родителей, одноклассников, учительского состава и т.д.

Описанные принципы подразумевают разные способы организации внеурочной деятельности, они тесно связаны с основным образованием.

Внеурочная деятельность по федеральному государственному стандарту состоит из пяти направлений развития личности:

- Общекультурное;
- Социальное;
- Духовно-нравственное;
- Спортивно-оздоровительное;
- Обще интеллектуальное.

Определены следующие цели для достижения задачи:

1. Формирование инициативности у учащихся;
2. Поляризация здорового образа жизни;
3. Привлечение учащихся к организации научного труда;
4. Сформировать интерес к научно-исследовательской деятельности;
5. Подведение учащихся на самостоятельное выполнение заданий;
6. Формирование личности каждого школьника в условия творческого и креативной среды, по средствам самоопределения и самореализации;
7. Формирования инициативности детей.

Методисты П.В. Степанов и Д.В. Григорьев считают в содержание внеурочной деятельности нужно включить такие виды деятельности как:

1. Познавательная;
2. Игровая;
3. Художественно-творческая;

4. Проблемно-ценностное общение;
5. Досугово-развлекательная деятельность;
6. Трудовая;
7. Туристско-краеведческая;
8. Спортивно-оздоровительная;
9. Художественное творчество.

Самая сложная задача - выявлении результатов, продуктивности внеурочной деятельности. Тем не менее существуют различные методы, упрощающие анализ результатов (тестирование, беседа, анкетирование, анализ портфолио, соревнования и т.п.).

Во ФГОС описаны следующие результаты, которые можно ожидать при обучении во внеурочной деятельности:

- Самореализация и творческое самоопределение детей;
- Осуществление игрового сюжета в учебной деятельности;
- Социальная защищенность и психологический комфорт каждого из учащихся;
- Формирование навыков организаторской и коллективной деятельности;
- Создание оптимальных условий для отдыха и развития учащихся.

Во ФГОС отражены требования к предметной области «Информатика и математика», которые обеспечивают изучение темы «Программирование».

Метапредметные результаты:

- Готовность выступить с видео, графически и аудио сопровождением;
- Умение анализировать, изображение, звуки, вводить текст с клавиатуры, записывать измеряемые величины в цифровой форме.

Предметные результаты:

- Умение анализировать, интерпретировать и представлять данные;
- Умение строить простейшие алгоритмы и действовать в соответствии с ними;
- Приобретение основных понятий о компьютерной грамотности;
- Овладение основами пространственного воображения, алгоритмического и логического мышления.

При изучении предметной области «Информатика и математика» учащиеся получают понимание о основе логико-алгоритмического компонента программирования.

Одной, из наиболее сложных тем в курсе информатики является «Алгоритмизация и программирование». На сегодняшний день существует достаточное количество проблем, связанных с этим направлением.

Практически каждый из учителей информатики сталкивается с достаточно большим количеством вопросов при составлении темы «Алгоритмизация и программирования»: каким образом выстроить подачу материала, каким материалом пользоваться учащимися при изучении материала и т.п. Описанные вопросы вытекают из отсутствия в полном объеме учебно-методических пособий и материалов для изложения указанной темы.

В таблице 1 представлены результаты анализа связанных с программированием тем и учебников, представленных в федеральном перечне с 2014/2015 по 2017/2018 учебные годы: Гейн А.Г., Угринович Н.Д., Поляков К.Ю., Семакин И.Г., Босова Л.Л [14, 15, 21, 22, 23, 24, 25].

Анализ учебников по теме «Алгоритмизация и программирование»

Класс	Программа	Параграф
Основная школа		
6 класс	Босова Л.Л.	§ 4.19. Графические исполнители (LINE, DRAW, CIRCLE)
7 класс	Угринович Н.Д.	§ 3.3 Растровая и векторная анимация
	Босова Л.Л.	§ 4.3.2 Форматирование символов
	Поляков К. Ю. Еремин Е. А.	§ 29 Алгоритмы и исполнители § 30 Способы записи алгоритмов § 33 Линейные алгоритмы § 36 Переменные § 37 Разветвляющиеся алгоритмы § 44 Анимация
8 класс	Угринович Н.Д.	§ 3.1 Кодирование и обработка звуковой информации § 6.1 Передача цифровой информации § 6.4 Разработка веб-сайтов с использованием языка разметки HTML
	Босова Л.Л.	§ 2.1 Алгоритмы и исполнители § 2.2 Способы записи алгоритмов § 2.3 Объекты алгоритмов § 2.4 Основные алгоритмические конструкции § 3.3 Программирование линейных алгоритмов § 3.4 Программирование разветвляющихся алгоритмов

		§ 3.4 Программирование циклических алгоритмов
	Поляков К. Ю. Еремин Е. А.	§ 2 Управление роботами § 19 Ветвления § 20 Программирование циклических алгоритмов
9 класс	Босова Л.Л.	§ 2.3.2. Разработка алгоритма методом последовательного уточнения для исполнения «Робот» § 2.4. Запись вспомогательных алгоритмов на языке паскаль
	Семакин И.Г.	Глава 6. § 35. Знакомство с языком Паскаль. § 37. Программирование ветвлений на языке Паскаль
	Угринович Н.Д.	§ 4.7. Графические возможности объектно-ориентированного программирования Visual Basic 2005
	Поляков К. Ю. Еремин Е. А.	§ 9. Логика и компьютеры § 11. Логические выражения § 18. Игровые стратегии

Как можно наблюдать из таблицы, авторы используют различные языки и среды программирования для обучения. Что дает учителю свободу выбора, но в то же время учителям необходимо осваивать большое количество языков и сред программирования.

Вывод по 1 Главе

В главе рассмотрены сведения об основах изучения программирования в школе и были проанализированы программные средства и языки программирования для изучения темы «Программирования» в школе, а именно: Час кода, Guido van Robot, Scratch, Lightbot, Alice, Logo, Baltie, Greenfoot, Little Wizzard, Etoys, Squeak, Karel, «Karel J., Karel ++, Robot, RoboMind.

На основе рассмотренных программных средств и языков можно сделать вывод, что для изучения основ программирования среда Scratch подходит наилучшим образом, т.к. представляет из себя оболочку с полностью графическим, интуитивно понятным интерфейсом, не требует соблюдения строгого синтаксиса, а все конструкции программы представлены в виде блоков, что сильно упрощает понимание и увеличивает вовлеченность в учебный процесс.

На основе рассмотренный в главе материала разработан курс «Основы программирования на языке Scratch».

ГЛАВА 2. Разработка факультативного курса «Основы программирования на языке Scratch» для 6 классов

2.1 Пояснительная записка факультативного курса «Основы программирования на языке Scratch»

Одной из основных частей в информатике является программирование, особенно для подготовки специалистов для IT-сферы. Тем более основные сведения о программировании присутствуют в образовательных программах гуманитарных специализаций, например как психология и социология. Следовательно, у разработчиков стандартов и программ образования отсутствуют сомнения в необходимости изучения раздела «программирования», даже для гуманитарных специальностей, пусть даже и на начальном уровне. Для обеспечения достаточно высоких результатов при изучении программирования в старших классах и высших учебных заведениях, нужно заложить прочную концепцию на начальном этапе изучения. Тем самым изучение основ программирования в школе доказывает актуальность факультатива «Основы программирования на языке Scratch».

Предлагаемый факультатив нацелен на организацию внеурочной деятельности в школе, основанный на программной среде Scratch. Занятия проводятся в группах примерно 10 – 15 учеников и рассчитан на одно занятие в неделю по 45 минут.

Цель факультатива: сформировать основы и программирования логико-алгоритмического мышления, овладения навыками работы с компьютером.

Задачи факультатива:

Образовательные:

- Ознакомить с техникой безопасности при нахождении компьютерном кабинете, а также при работе с компьютером;
- Научить применять знания, полученные на практике, при решении задач;

- Дать первоначальные знания о «Программировании»;
- Познакомить с программной средой «Scratch»;
- Сформировать практические навыки программирования в «Scratch»
- Сформировать основные навыки и умение пользоваться компьютерной техникой;
- Сформировать начальные навыки тестирования, отладки, разработки программ.

Воспитательные:

- Воспитать навык работы в коллективе, паре, группе и разумно распределять обязанности;
- Воспитать уважительные отношения к учителям и сверстникам.

Развивающие:

- Развить творческие способности у детей;
- Развить ответственность и самостоятельность
- Развить внимание, логику, способность к концентрации и анализу, память;
- Сформировать алгоритмическую и информационную культуру;
- Развить причинно-следственные и логику.

Требования к освоению курса:

Описанные задачи факультатива способствуют обеспечению выполнения требований к уровню знаний учащихся прошедших факультатив:

Личностные результаты:

- Большой познавательный кругозор, любознательность, ответственность, тяга к творчеству;
- Стремление учеников к самореализации и саморазвитию;
- Интерес к теме «Информатика и ИКТ», желание применить полученные знания, умения и навыки на других учебных предметах и повседневной жизни.

- Стремление к развитию творческих способностей и навыков за счет сформированного алгоритмического и логического мышления;
- Тяга к увеличению знаний в области ИКТ;
- Способность принять решение с долей ответственности;
- Настрой на достижение цели индивидуально и коллективно;
- Понимание важности информационных технологий в условиях информатизации общества;
- Умение критически оценивать и способность выделить основную мысль при помощи анализа информации;
- Понимание основных технических и гигиенических правил безопасности при использовании ИКТ.

Предметные результаты:

- Понимание основных терминов – программа, информация, данные, алгоритм;
- Умение создавать и различать циклические, ветвящиеся и линейные алгоритмы;
- Навык формального выполнения алгоритмов, сформированных с использованием циклов и условных операторов;
- Умение написать простые программы в среде Scratch.

Метапредметные результаты:

- Навык рационального распределения времени и организации собственной учебной деятельности;
- Умение устанавливать промежуточные цели и задачи направленные на достижение конечного результата;
- Прогнозирование возможных результатов;
- Соотнесение полученных результатов с установленными целями и задачами, самоконтроль;
- Готовность к исправлению данных при возникновении и обнаружении ошибки;

- Выделении важно информации и использование методов поиска информации;
- Представление и визуализация информации;
- Подбор более эффективных способов решения задач;
- Готовность к продуктивному и дружескому сотрудничеству в паре или группе;
- Использование информационных средств в учебной и повседневной деятельности.

Способы контроля знаний:

- Отзывы родителей, учителей;
- Проекты;
- Итоговая контрольная работа.

В таблице 2 представлен учебно-тематический план курса:

Таблица 2

Тематическое планирование курса

№	Тема занятия	Количество часов		
		Всего	Теория	Практика
1	Введение и техника безопасности	1	1	-
2	Алгоритмы. Линейные алгоритмы	1	1	-
3	Блок схема. Ветвление	1	1	-
4	Решение задач	1	1	-
5	Блок схема. Циклические алгоритмы	1	1	-
6	Решение задач	1	1	-
7	Знакомство со Scratch	1	-	1
8	Анимация имени	1	-	1
9	Исполнитель Scratch. Цвет и размер пера	1	-	1
10	Основные инструменты графического редактора программной среды Scratch	1	-	1

11	Рисование исполнителем Scratch.	1	-	1
12	Рисование квадратов и прямоугольников линейно	1	-	1
13	Цикл. Исполнитель рисует линии и фигуры	1	-	1
14	Исполнитель Scratch рисует несколько линий и фигур. Конечный цикл	1	-	1
15	Циклический алгоритм. Проект узор	1	-	1
16	Рисование пунктирных линий. Проект светофор	1	-	1
17	Разработка собственного проекта	1	-	1
Итого:		17	6	11

Поурочное планирование факультатива.

Урок 1. Введение и техника безопасности.

Цель урока: ознакомить учащихся с техникой безопасности при работе с ПК и правилам нахождения в компьютерном классе.

Задачи:

Образовательная: систематизировать знания по технике безопасности;

Развивающая: развить эстетические чувства; научить соблюдать дисциплину и бережно отношение к окружающим предметам;

Воспитательные: развить коммуникабельность и сплоченность в коллективе; развить творческие способности.

Основные понятия: безопасность, техника безопасности, ПК.

Методическая рекомендация: Учитель в игровой форме и с помощью технической поддержки рассказывает детям о технике безопасности при работе с ПК и при нахождении в компьютерном кабинете. В конце урока проводится викторина, направленная на проверку усвоенной информации.

Вопросы для контроля:

1. Что категорически запрещается делать в кабинете информатики?
2. Что следует проверить перед работой с ПК?
3. Что делать, в случае не исправности компьютера?

Урок 2. Алгоритмы. Линейный алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования у обучающихся представление о понятии «Алгоритм».

Задачи:

Образовательная: познакомить учащихся с понятием «алгоритм», «линейный алгоритм»; научить составлять и выполнять алгоритмы.

Развивающая: развивать логическое мышление учащихся, память, внимание, воображение, познавательную активность, способность быстро воспринимать информацию.

Воспитательная: воспитывать у детей активность, правила поведения, самостоятельность, интерес к предмету.

Основные понятия: алгоритм, линейный алгоритм, команда.

Методические рекомендации: Учитель организует урок в виде путешествия, где учащиеся сами ставят цели и задачи урока.

Вопросы для контроля:

1. Что такое алгоритм?
2. Какой алгоритм называется линейным?
3. С какой команды начинается алгоритм?
4. Какой командой заканчивается алгоритм?

Урок 3. Блок – схемы. Условный алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования у учащихся представление об условных алгоритмах и построении блок-схем.

Задачи:

Образовательная: дать первоначальное представление об условных алгоритмах, блок-схемах; научить формировать условные алгоритмы и формально исполнять их; научить выделять главную информацию, анализировать содержание алгоритма; научить составлять блок-схемы условного вида.

Развивающая: развивать творческие способности детей; развивать умение прогнозировать результаты.

Воспитательная: формировать стремление в приобретение новых знаний; воспитывать коммуникативные качества учащихся;

Основные понятия: блок-схема, алгоритм, условный алгоритм.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Дайте определение блок-схемы.
2. Какой алгоритм называется условным?
3. С помощью каких фигур строить блок-схема условного алгоритма?
4. Как начинается любой алгоритм?
5. Как заканчивается любой алгоритм?

Урок 4. Решение задач.

Тип урока: урок применения предметных умений.

Цель урока: создать условия для практического применения знаний и умений по теме условные алгоритмы.

Задачи:

Образовательная: научить решать задачи с использованием циклических блок-схем.

Развивающая: развивать способность к самостоятельному решению поставленной задачи; формировать логическое мышление, развивать память;

Воспитательная: способствовать повышению интереса к предмету; воспитывать у учеников коммуникабельность, аккуратность и терпение; объективное оценивание своих достижений и сопоставление их с целями.

Основные понятия: блок-схема, цикл, циклический алгоритм.

Основные понятия: блок-схема, алгоритм, условный алгоритм.

Методические рекомендации: На данном уроке учитель демонстрирует способы решения задач с помощью условных алгоритмов. При необходимости организует не только коллективную работу, но и индивидуальную. В конце урока формирует некоторые вопросы, демонстрирующие результаты освоения темы.

Вопросы для контроля:

1. Дайте определение блок-схемы.
2. Какой алгоритм называется условным?
3. С помощью каких фигур строить блок-схема условного алгоритма?
4. Как начинается любой алгоритм?
5. Как заканчивается любой алгоритм?

Урок 5. Блок-схемы. Циклический алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования представления о циклических алгоритмах и построении блок-схем.

Задачи:

Образовательная: дать первоначальное представление о циклических алгоритмах, блок-схемах; научить формировать циклические алгоритмы и формально исполнять их; научить выделять главную информацию, анализировать содержание цикла; научить составлять блок-схемы циклического вида.

Развивающая: развивать такие компоненты как память, логическое мышление, анализ; научить школьников формировать свое собственное мнение и аргументируя, высказывать его.

Воспитательная: вырабатывать познавательный интерес к теме курса; развивать коммуникативные качества, такие как, умение слушать, высказывать свое мнение, и обосновывать его.

Основные понятия: блок-схема, цикл, циклический алгоритм, цикл с предусловием, цикл с постусловием.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Что такое цикл?
2. Чем отличается циклический алгоритм от других видов алгоритмов?
3. Какие фигуры используются при формировании циклического алгоритма?
4. Как начинается любой алгоритм?

Урок 6. Решение задач.

Тип урока: урок применения предметных умений.

Цель урока: создать условия для практического применения знаний и умений по теме циклические алгоритмы.

Задачи:

Образовательная: научить решать задачи с использованием циклических блок-схем.

Развивающая: развивать способность к самостоятельному решению поставленной задачи; формировать логическое мышление, развивать память.

Воспитательная: способствовать повышению интереса к предмету; воспитывать у учеников коммуникабельность, аккуратность и терпение; объективное оценивание своих достижений и сопоставление их с целями.

Основные понятия: блок-схема, цикл, циклический алгоритм.

Методические рекомендации: На данном уроке учитель демонстрирует способы решения задач с помощью циклических алгоритмов. При необходимости организует не только коллективную работу, но и индивидуальную. В конце урока формирует некоторые вопросы, демонстрирующие результаты освоения темы.

Вопросы для контроля:

1. Что такое цикл?
2. Чем отличается циклический алгоритм от других видов алгоритмов?
3. Какие фигуры используются при формировании циклического алгоритма?
4. Как начинается любой алгоритм?

Урок 7. Знакомство со Scratch.

Тип урока: урок усвоения новых знаний.

Цель урока: познакомить учащихся с интерфейсом программы Scratch.

Задачи:

Образовательная: дать представление программы Scratch, определение скрипта и спрайта.

Развивающая: развивать у обучающихся интерес к предмету, умения и навыки при работе с программой Scratch.

Воспитательная: воспитывать трудолюбие, дисциплинированность, доброжелательное отношение друг другу и умение преодолевать трудности.

Основные понятия: интерфейс программы, спрайт, скрипт, блок.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Каким цветом представлен блок «контроль»?
2. Можно ли изменить костюм персонажа и как это сделать?

3. Что такое скрипт?
4. Дайте определение спрайта.
5. Сколько различных командных блоков существует в Scratch?
6. Как можно поменять фон?

Урок 8. Анимация имени.

Тип урока: урок усвоения новых знаний.

Цель урока: научить создавать простейшую анимацию в среде Scratch.

Задачи:

Образовательная: создать условия для отработки навыков и умений работы в среде Scratch.

Развивающая: создать условия для развития внимания, памяти и воображения.

Воспитательная: содействовать развитию культуры речи и эстетического вкуса.

Основные понятия: анимация, исполнитель в Scratch, область сцены, объекты.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. С помощью какого блока можно добавлять звук?
2. Какие существуют способы добавления новых спрайтов?
3. Как изменить угол наклона объекта?
4. Как изменить цвет объекта?

Урок 9. Исполнитель Scratch. Цвет и размер пера.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условия для формирования у детей представления об исполнителе Scratch.

Задачи:

Образовательная: ознакомить учащихся с возможностями использования пера: его цвет и размер; обеспечить представление о написании алгоритма и выполнение поставленной задачи.

Развивающая: развивать умение анализировать, выделять основное и второстепенное, умение слушать.

Воспитательная: воспитывать уважительное отношение к сверстникам и преподавателям, тактичность, вежливость, коммуникабельность.

Основные понятия: блок контроля, блок управления пером, исполнитель.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие характеристики существуют у объекта?
2. С помощью, каких команд начинается и заканчивается алгоритм?
3. С какой целью используется команда «очистить»?
4. В какой цвет окрашен блок «перо»?

Урок 10. Основные инструменты графического редактора программной среды Scratch.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирований представлений об инструментах графического редактора, встроенного в программную среду Scratch.

Задачи:

Образовательная: дать первоначальные сведения о графическом редакторе программной среды Scratch, инструментах и возможностях, при создании графических изображений.

Развивающая: совершенствовать учебно-интеллектуальные умения, а именно мотивацию к изучению предмета, внимание при восприятии информации, логическое осмысление предложенного материала, самостоятельное выполнение заданий, навыки самоконтроля.

Воспитательная: воспитывать ответственное отношение к работе и уважение не только к учителям, но и к сверстникам.

Основные понятия: графический редактор, инструменты графического редактора, блок внешности, блок управления пером.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие вы знаете инструменты графического редактора?
2. Какой блок окрашен в фиолетовый цвет?
3. Какой блок окрашен в зеленый цвет?

Урок 11. Рисование исполнителем Scratch линий. Линейный алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для закрепления знаний о линейном алгоритме и применение знаний и навыков в ходе практического задания.

Задачи:

Образовательная: формировать навыки создания блок–схем линейного алгоритма; приобретать навыки решения конкретной задачи.

Развивающая: развивать учебно-организационные навыки, а именно воспринимать информацию, определять цель и намечать задачи;

Воспитательная: воспитывать упорность, трудолюбие, аккуратность, терпение при выполнении заданий.

Основные понятия: блок-схема, алгоритм, линейный алгоритм, виды элементов блок-схемы алгоритма.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие вы знаете инструменты графического редактора?
2. Какой блок окрашен в фиолетовый цвет?
3. Какой блок окрашен в зеленый цвет?

Урок 12. Рисование квадратов и прямоугольников линейно.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирования представлений о способах начертания фигур линейно.

Задачи:

Образовательная: формировать пространственное представление; закрепление знаний о линейном алгоритме.

Развивающая: развивать творческие способности детей, креативность в подходе решения вопросов.

Воспитательная: формировать ответственное отношение к учебе, дисциплинированность.

Основные понятия: команда «повернуть в направление», отладка программы.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какие вы знаете геометрические фигуры?
2. Чем отличается квадрат от прямоугольника?
3. С помощью какой команды начинается выполнение алгоритма?

Урок 13. Цикл. Исполнитель рисует линии и фигуры.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирования представлений начертания фигур с помощью циклов.

Задачи:

Образовательная: создать представление о циклическом алгоритме;

Развивающая: совершенствовать умственные свойства и возможности учащихся;

Воспитательная: устранять небольшие недостатки в воспитании младших школьников, например неорганизованность, нетактичность и т.д.

Основные понятия: цикл, циклический алгоритм, блок-схема циклического алгоритма.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Каким способом можно открыть ранее созданный проект?
2. Сколько градусов имеет прямой угол?

Урок 14. Исполнитель Scratch рисует несколько линий и фигур. Конечный цикл.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирования представлений начертания фигур с помощью циклов.

Задачи:

Образовательная: совершенствовать и закрепить знания и умения в построении циклических алгоритмов.

Развивающая: развивать волевые качества младших школьников, умения самостоятельно преодолевать трудности.

Воспитательная: способствовать трудовому обучению школьников.

Основные понятия: цикл, конечный цикл, блок-схема конечного цикла, бесконечный циклический алгоритм, имя спрайта, изменение костюма объекта.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какое отличие программы, которая рисует 3 ступеньки, от той которая рисует 8 ступенек?
2. Что характерного в бесконечном цикле?

Урок 15. Циклический алгоритм.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для закрепления представлений о циклическом алгоритме.

Задачи:

Образовательная: обеспечить в течение урока усвоение основных понятий циклического алгоритма.

Развивающая: формировать способность рассуждать логически, четко и кратко формулировать свои мысли.

Воспитательная: способствовать физическому воспитанию, а именно организовывать физкультминутки в целях снятия напряжения и усталости.

Основные понятия: цикл, блок-схема циклического алгоритма.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Какую команду следует использовать при дальнейшем перемещении по экрану, без следов рисования?
2. Где осуществляется выбор готовых костюмов для спрайта?

Урок 16. Рисование пунктирных линий.

Тип урока: урок усвоения новых знаний.

Цель урока: создать условие для формирования представлений о циклических алгоритмах.

Задачи:

Образовательная: обеспечить контроль знаний и умений в ходе выполнения заданий.

Развивающая: развивать умение анализировать, выделять основное и второстепенное, умение слушать.

Воспитательная: воспитывать упорность, трудолюбие, аккуратность, терпение при выполнении заданий.

Основные понятия: цикл, пунктирные линии, алгоритм рисования пунктирных линий.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоения знаний предыдущего урока. Он выступает в качестве помощника, корректируя действия ученика, при необходимости. В конце урока школьники выполняют самостоятельную работу.

Вопросы для контроля:

1. Дайте определение цикла.
2. Чему вы научились на сегодняшнем уроке?

Урок 17. Разработка собственного проекта

Тип урока: Использование полученных ранее знаний и умений

Цель урока: Создание условий для закрепления полученных знаний

Задачи:

Образовательная: обеспечить контроль знаний и умений в ходе выполнения заданий.

Развивающая: развивать умение анализировать, выделять основное и второстепенное, умение слушать.

Воспитательная: воспитывать упорность, трудолюбие, аккуратность, терпение при выполнении заданий.

Методические рекомендации: В начале урока учитель проводит опрос на проверку усвоенных знаний. Он выступает в качестве помощника, корректируя действия ученика, при необходимости.

Вопросы для контроля:

1. Каким способом можно открыть ранее созданный проект?
2. Чему вы научились при изучении Scratch?

2.2 Программная поддержка курса

В роли программной поддержки факультатива «Изучение основ программирования в школе» выступает веб-сайт разработанный на платформе WordPress. Пособие находится по адресу <http://i95252tn.beget.tech>.

Главная страница представлена на рисунке 14

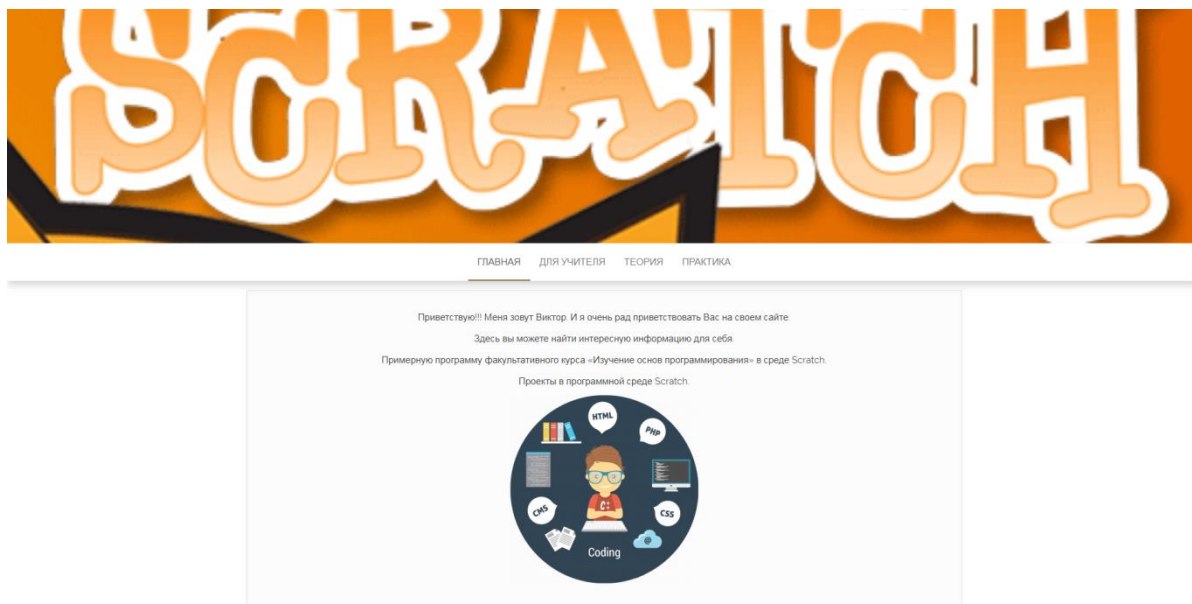


Рис. 14. Главная страница

В разделе «Для учителя» (рис. 15) находится примерная программа для составления факультатива.

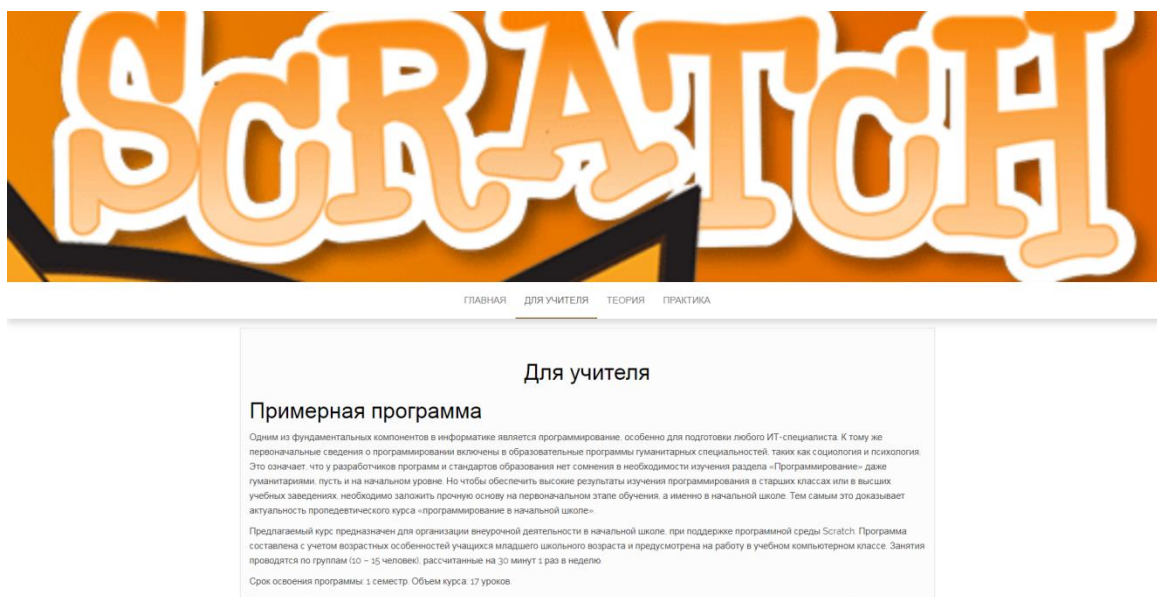


Рис. 15. Раздел «Для учителя»

В разделах «Теория» и «Практика» расположены теоретические и практически сведения по работе со Scratch (рис. 16 и 17).



Рис. 16. Раздел «Теория»

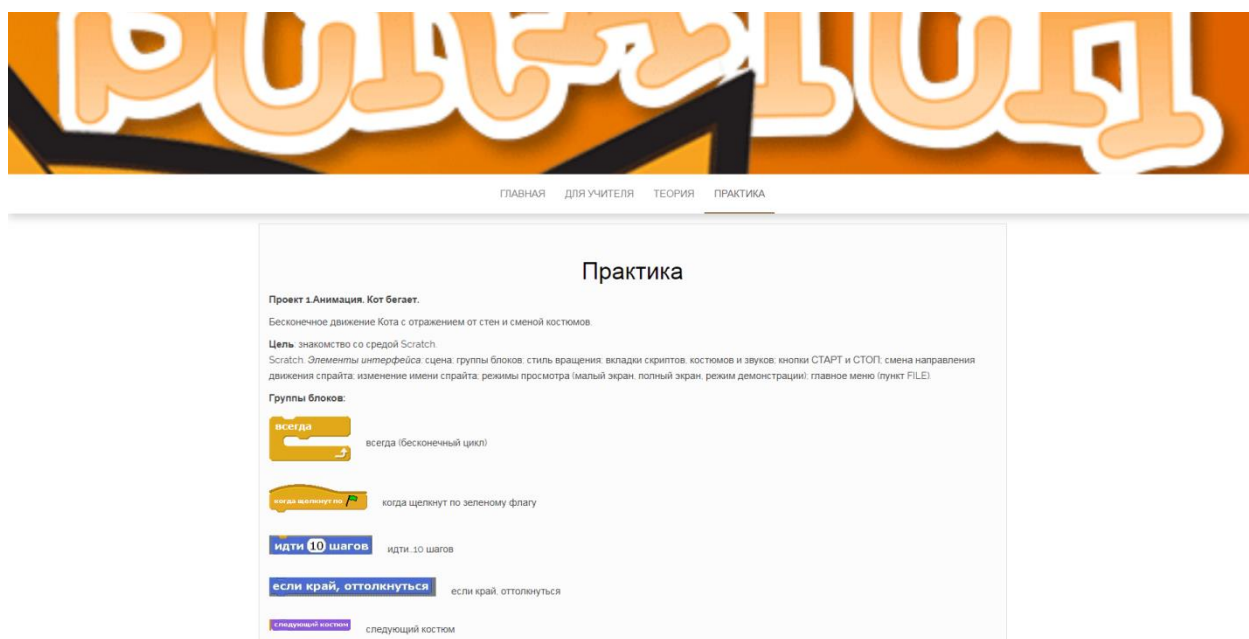


Рис. 17. Раздел «Практика»

2.3 Апробация результатов исследования в школе

Апробация производилась в рамках научно-исследовательской практики в МБОУ СОШ № 92 г. Челябинска. Курс проводился в 6 – х классах, и в течении 2 занятий были рассмотрены темы «Знакомство со Scratch», «Анимация имени».

Апробация завершилась успешно. Этому способствовали правильно поставленные цели и задачи, а также, мотивация.

Тема курса была не знакома учащимся, но это не сказалось на их работе, они быстро включились в изучение темы, т.к. в таком возрасте дети с удовольствием получают новые знания если поданная информация наглядна и проста.

Вывод по 2 Главе

В рамках второй главы рассмотрен ФГОС, описан факультативный курс «Основы программирования на языке Scratch» и программная поддержка в виде веб-сайта.

Исходя из анализа федеральной государственной программы можно заключить – внеурочная деятельность является как составной частью учебно-воспитательного процесса, так и одной из форм организации досуговой деятельности, направленная оказание помощи учащимся в освоении нового материала в учебной деятельности, а также создание стимула и интереса к предмету.

Апробация производилась в рамках научно-исследовательской практики МБОУ СОШ № 92

В рамках второй главы был разработан факультативны курс «Изучение основ программирования в школе» и программную поддержку.

Заключение

Цель исследования достигнута – изучены методы преподавания основ программирования, разработан курс «Изучение основ программирования в школе».

Разработанный факультатив рассчитан на применение во внеурочное время, это позволяет изучить основы программирования в школьном возрасте, развитию и формированию логико-алгоритмического мышления, а также обеспечить учителя освоением новой программной среды Scratch.

В рамках исследования решены задачи:

1. Изучены основы по проблеме исследования;
2. Разработан факультативный курс занятий по изучению основ программирования в среде Scratch;
3. Создана программная поддержка факультативного курса в виде веб-сайта.

В подтверждении поставленной гипотезы можно сказать – факультативный курс по изучению основ программирования дает учащимся возможность овладеть алгоритмическим и логическим мышлением.

Все поставленные задачи в рамках исследования были выполнены, что свидетельствует о правильно поставленной гипотезе.

Список литературы

1. Босова Л., Босова А. Информатика. [Текст]: Учебник для 6 класса / Л. Босова, А. Босова – М.:БИНОМ. Лаборатория знаний, 2016. – 224с.
2. Босова Л., Босова А. Информатика. [Текст]: Учебник для 9 класса / Л. Босова, А. Босова – М.:БИНОМ. Лаборатория знаний, 2016. – 208с.
3. Ершов А.П., Звенигородский Г.А., Первин Ю.А. Школьная информатика (концепции, состояния, перспективы) [Текст] // Информатика и образование. 1995. – 226с.
4. Ибрагимов Г.И., Андриянова Т.М. Теория обучения [Текст]: Теория обучения: учебное пособие / Под редю Г.И. Ибрагимова. [Ибрагимова Е.М., Андриянова Т.М.] – М.:Гуманитар. Издю центр ВЛАДОС, 2011. – 383 с.
5. Информационный портал Alice. Alice [Электронный ресурс] – Режим доступа: <http://www.alice.org/>. - (Дата обращения 3 Апрель 2018).
6. Информационный портал Baltie [Электронный ресурс] – Режим доступа: <http://progopedia.ru/language/baltie/>. - (Дата обращения 4 Апреля 2018 г).
7. Информационный портал Etoys [Электронный ресурс] – Режим доступа: <http://www.etoys.com/about/>. - (Дата обращения 20 Марта 2018).
8. Информационный портал Greenfoot [Электронный ресурс] - Режим доступа: <https://www.greenfoot.org/about/>. - (Дата обращения 15 Апреля 2018).
9. Информационный портал Guido van Robot [Электронный ресурс] - Режим доступа: <http://gvr.sourceforge.net/about.php/>. – (Дата обращения 10 Марта 2018).
10. Информационный портал RoboMind [Электронный ресурс] - Режим доступа: <https://www.robomindacademy.com/go/robomind/home>. - (Дата обращения 17 Апреля 2018).
11. Информационный портал Squeak [Электронный ресурс] – Режим доступа: <https://squeak.org/documentation/>. – (Дата обращения 25 Марта 2018).

12. Информационный портал КуМир [Электронный ресурс] - Режим доступа: <https://www.niisi.ru/kumir/>. – (Дата обращения 28 Апреля 2018).
13. Образовательный портал Scratch [Электронный ресурс] - Режим доступа: <https://scratch.mit.edu>. – (Дата обращения 16 Апреля 2018).
14. Образовательный ресурс Little Wizzard [Электронный ресурс] - Режим доступа: <http://littlewizard.sourceforge.net/tutorial.html>. - (Дата обращения 3 Апреля 2018 г).
15. Онлайн документация Logo [Электронный ресурс] - Режим доступа: <https://myrobot.ru/logo/aboutlogo.php>. – (Дата обращения 22 Апреля 2018).
16. Официальный портал Karel J. Robot [Электронный ресурс] - Режим доступа: <http://karel.sourceforge.net> - / Дата обращения 20 Марта 2018 г..
17. Официальный портал Lightbot [Электронный ресурс] - Режим доступа: <http://lightbot.com> - / Дата обращения 1 Апреля 2018 г..
18. Портал учителя кроlyaков [Электронный ресурс] - Режим доступа: <http://kpolyakov.spb.ru/school/robots/robots.htm>. - (Дата обращения 25 Апреля 2018).
19. Семакин И., Информатика. [Текст]: 10 класс. Углубленный уровень: учебник в 2 ч. / И.Г. Семакин, Т.Ю. Шеина, Л.В. Шестакова - М. : БИНОМ. Лаборатория знаний, 2016. – 436с.
20. Семакин И., Информатика. [Текст]: Учебник для 9 класса /И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова. – М.: БИНОМ. Лаборатория знаний, 2016. – 208с.
21. Угринович Н. Д. Информатика и ИКТ. [Текст]: Учебник для 7 класса / Н.Угринович. – М.: БИНОМ. Лаборатория знаний, 2016. – 168с.
22. Угринович Н. Д. Информатика и ИКТ. [Текст]: Учебник для 8 класса / Н.Угринович. – М.: БИНОМ. Лаборатория знаний, 2016. – 160с.
23. Угринович Н. Д. Информатика и ИКТ. [Текст]: Учебник для 9 класса / Н.Угринович. – М.: БИНОМ. Лаборатория знаний, 2012. – 295с.

24. Федеральный государственный образовательный стандарт основного общего образования 2010г [Электронный ресурс] - Режим доступа: <http://минобрнауки.рф/документы/543>. - (Дата обращения 25 Марта 2018).

Приложение

Конспект урока на тему «Знакомство со Scratch».

Тема урока: Знакомство со Scratch.

Тип урока: урок усвоения новых знаний.

Цель урока: познакомить учащихся с интерфейсом программы Scratch.

Задачи:

Образовательная: дать представление программы Scratch, определение скрипта и спрайта.

Развивающая: развивать у обучающихся интерес к предмету, умения и навыки при работе с программой Scratch.

Воспитательная: воспитывать трудолюбие, дисциплинированность, доброжелательное отношение друг другу и умение преодолевать трудности.

Этапы урока:

1. Организационный момент – 1 мин.
2. Изучение нового материала – 10 мин.
3. Физкультминутка – 2 мин.
4. Закрепление изученного материала – 17 мин.

Перечень средств ИКТ, используемых на уроке:

- Интерактивная доска, компьютеры для каждого ученика, программа Scratch.

СТРУКТУРА И ХОД УРОКА

1. Организационный момент 1 мин.

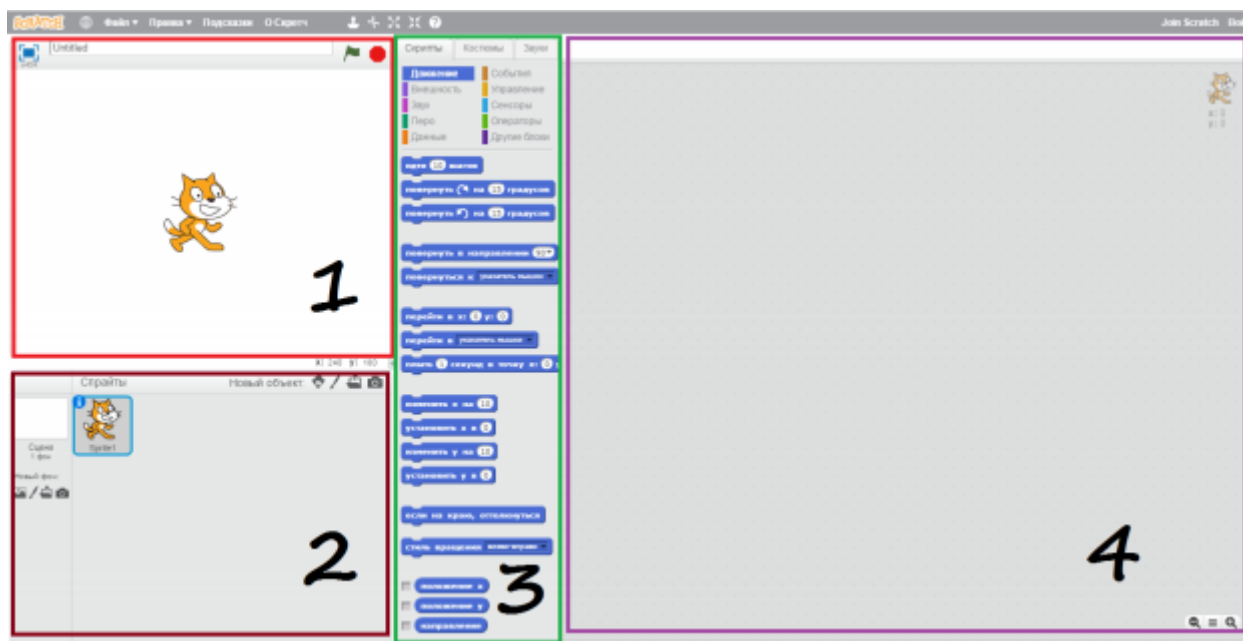
Учитель: Приветствует класс, отмечает присутствующих.

Здравствуйте, ребята. Сегодня мы с вами познакомимся с программой Scratch и даже напишем первую программу.

2. Изучение нового материала 10 мин.

Но для начала рассмотрим интерфейс программы.

Так выглядит экран программы, которая состоит из четырех частей.



Область под номером 1 – это сцена, на которой видно как работает программа.

Область под номером 2 – это список спрайтов. Спрайт – это объекты в программе, такие как животные, автомобили, люди, буквы и т.д.

Область под номером 3 – набор блоков, которые задают спрайту различные свойства, например движение, звук, контроль, внешность и т.д.

Область под номером 4 – это рабочая область (область скриптов), куда мы будем складывать блоки из 3 окна. Скрипт – это программа, которая состоит из блоков и описывает какое-либо действие спрайта.

Предлагаю область под номером 3 рассмотреть внимательней.

В Scratch большое количество блоков разного цвета. Мы познакомимся со всеми постепенно в течении курса, но для начала будем использовать синие, фиолетовые, розовые и желтые.

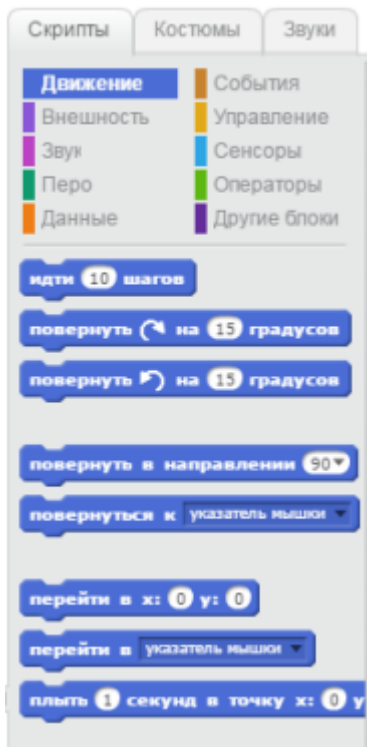
В данном случае мы выбрали раздел движение. Скажите мне, пожалуйста, что наш спрайт может делать, если мы воспользуемся этими блоками?

Ученики:

Идти 10 шагов, повернуть направо/налево на 15 градусов, повернуть в направлении, повернуться к чему-либо, плыть, менять расположение, вращаться.

Учитель:

Верно. Хочу заметить числа в окошечках мы можем задавать сами, например идти не 10 шагов, а 7 или 40 шагов.



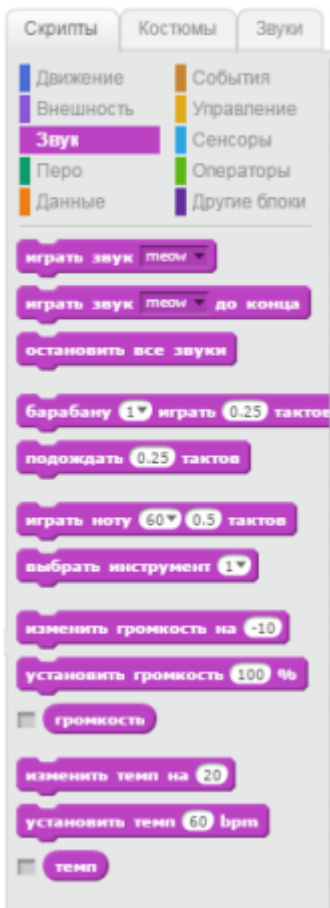
Перейдем к розовым блокам. Этот раздел отвечает за звуки. Прочитайте по очереди.

Ученики:

Играть звук, останавливать все звуки, подождать несколько тактов, изменять громкость, изменять темп.

Учитель:

Молодцы. На самом деле мы не только можем выбирать различные звуки из библиотеки программы, но и записывать их сами.



Рассмотрим раздел внешность. Прочитайте по очереди.

Ученики:

Говорить слово в течении 2 секунд, сказать слово, думать, показаться, прятаться, сменить костюм, поменять фон, изменять размер.



Учитель:

Хорошо. И последний раздел, который мы сегодня рассмотрим, называется события. Данные блоки отличаются от предыдущих тем, что они являются вспомогательными к блокам из других разделов. Например блок «когда спрайт нажат» мы не можем использовать самостоятельно, обязательно к нему прикрепим другой например «идти 1 шаг». В результате получится – когда спрайт нажат, идти 1 шаг.



Ребята, мы переходим к практической части. Но перед тем как сесть за компьютеры, предлагаю выполнить несколько физических упражнений.

3. Физкультминутка 2 мин.

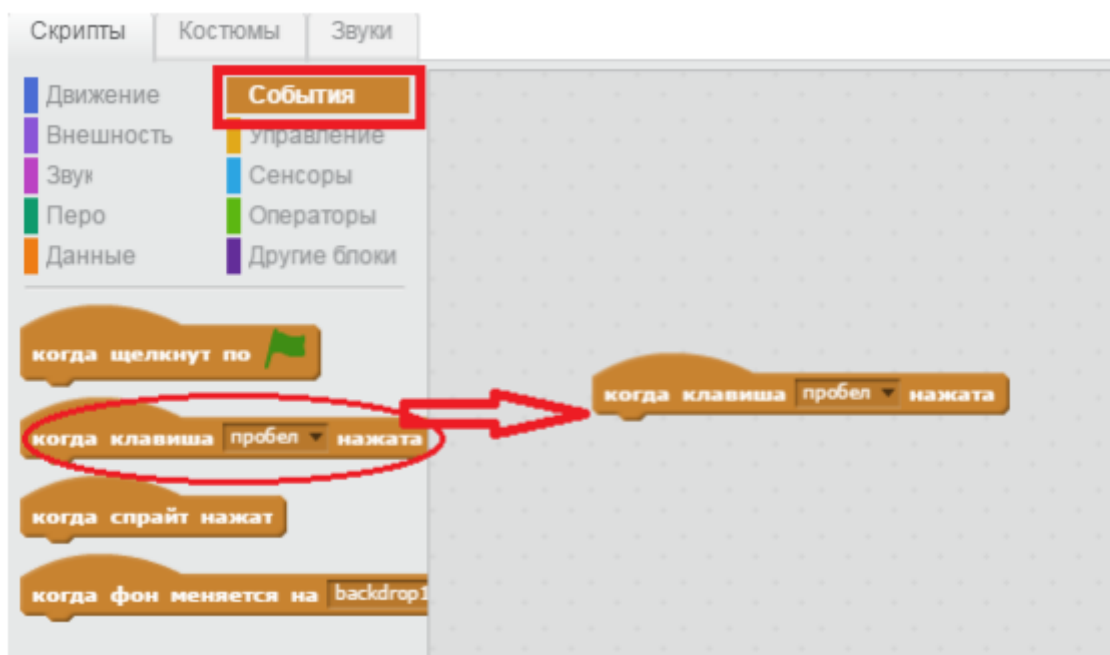
Встаем из-за парт. Руки вверх, тянемся за ручками. Руки в стороны, затем вниз и расслаблено скрещиваем перед грудью, голову наклоняем вперед. Повторяем 5 раз (темп средний).

4. Закрепление изученного материала.

Молодцы. Теперь садимся за компьютеры. Запускаем программу Scratch с рабочего стола.

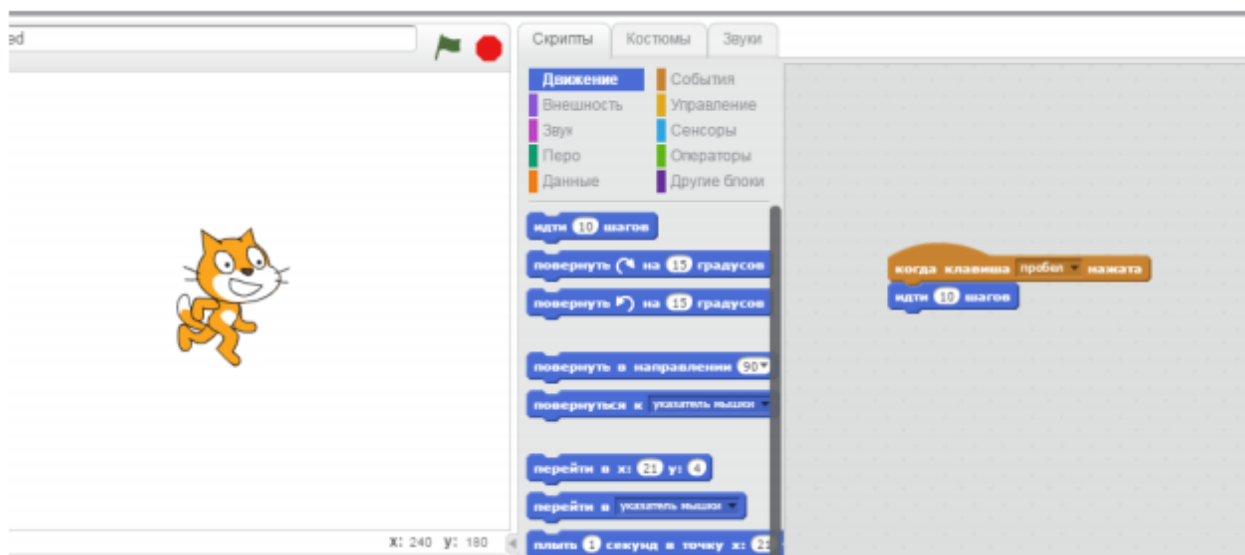
Задание 1. Учим кота ходить.

Выбираем вверху желтые блоки «события», находим блок «когда клавиша пробел нажата», нажимаем левой кнопкой мыши и, не отпуская, перетаскиваем блок вправо в область скриптов.



Затем выбираем синие блоки «движение», тянем блок «идти 10 шагов» в область скриптов и прицепляем под желтый блок «когда клавиша пробел нажата».

В результате у вас должно получиться так.



Если появились затруднения, поднимайте руку, и я подойду к вам. Вот наша первая программа! Теперь необходимо протестировать ее. Нажимайте на клавиатуре пробел и скажите что происходит с котиком?

Ученики:

Кот двигается вправо.

Учитель:

Верно. Если много раз нажимать на пробел, то он дойдет до края экрана и почти скроется.

Его можно вытащить, потянув за хвост.

Задание 2. Учим кота мяукать.

Чтобы наш кот начал издавать звуки, например мяукать, нужно добавить соответствующий блок. Как вы думаете в каком разделе он находится?

Ученики:

В разделе «звуки».

Учитель:

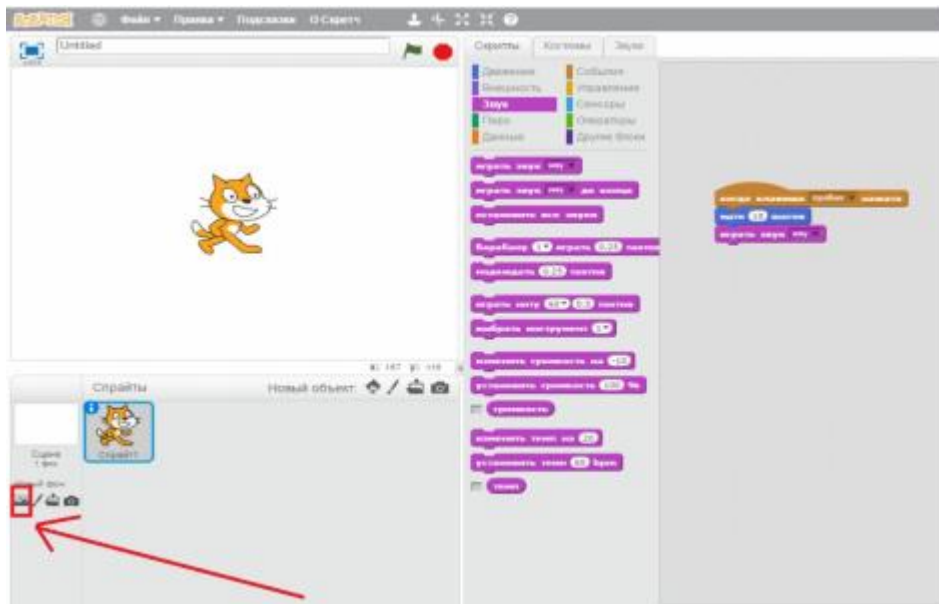
Верно. Переходим в раздел «звуки» и находим там блок «играть звук meow». Затем перетаскиваем его к уже имеющимся блокам и прикрепляем к синему блоку.



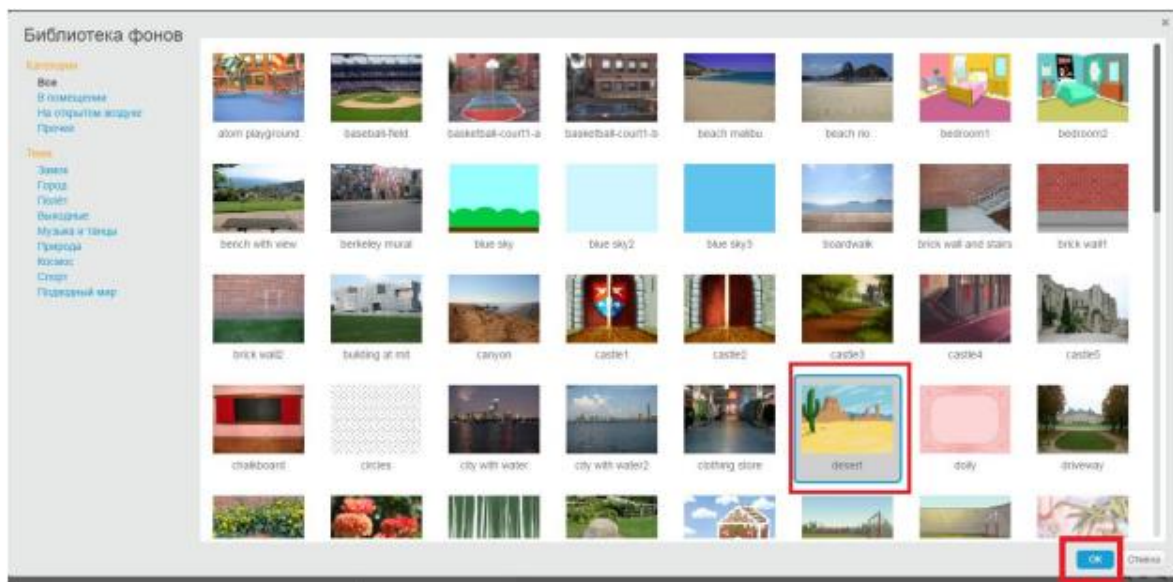
Теперь при нажатии на пробел кот не только идет, но и мяукает. У кого есть вопросы, спрашивайте.

Задание 3. Меняем фон.

Ребята согласитесь, что белый фон – это скучно. Давайте выберем подходящий фон, например пустыню или пляж. Для этого необходимо в левом нижнем углу найти кнопку «Выбрать фон из библиотеки».



Нажимаем на кнопку, и появляется окно – Библиотека фонов. Там выбирайте любой понравившийся фон (я выбрала пустыню) и нажимайте кнопку «ОК» в правом нижнем углу.



Фон изменился. Согласитесь со мной, что так куда более интересней.



У всех получилось? Если есть затруднения, то учитель подходит и помогает ученику.

Если вдруг вы нечаянно добавили не тот блок или просто захотели удалить блок, то необходимо открепить его и перетащить обратно.

Самостоятельно!

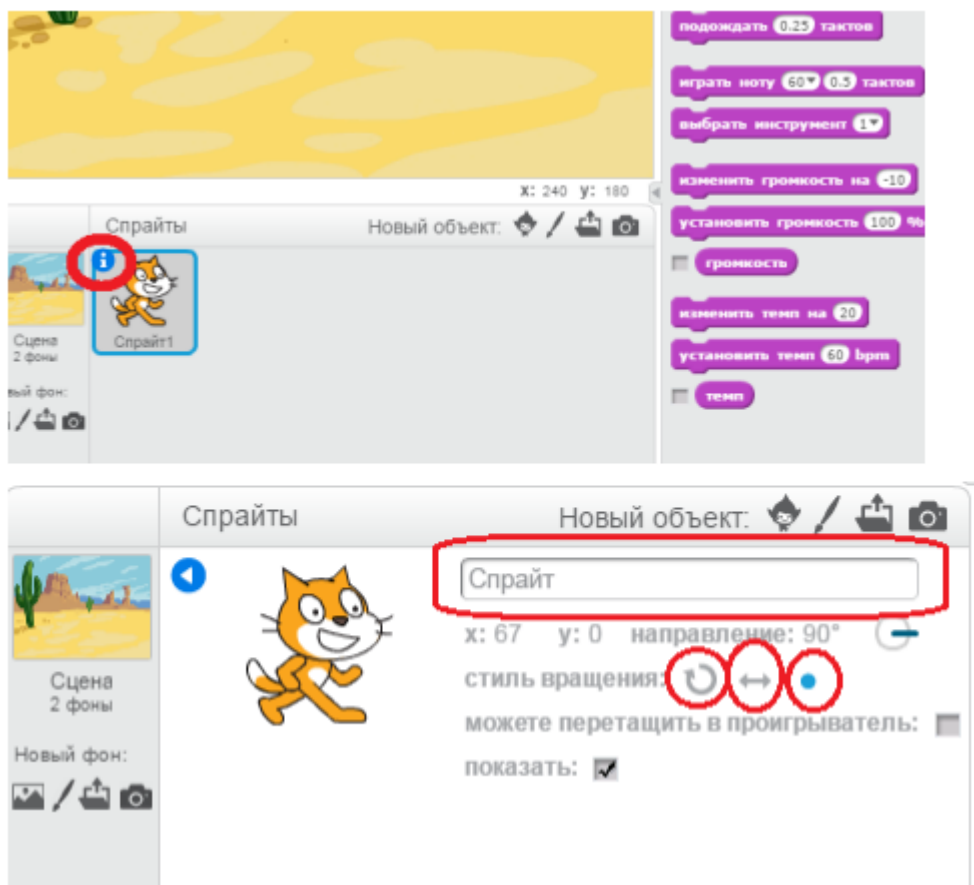
Попробуйте самостоятельно убрать блок «играть звук», а вместо него поставить новый «если на краю, оттолкнуться» из раздела движение. Что в итоге получилось?

Ученики:

Кот шагает при нажатии на пробел, но теперь не скрывается, когда доходит до края экрана, а поворачивается и движется в другую сторону и не мяукает.

Учитель:

Хорошо. В левом нижнем углу в поле спрайтов найдите кота и рядом значок «i». Нажмите на него.



Измените имя спрайта на кот. А так же поменяйте стиль вращения. 1 стиль означает, что кот будет ходить по кругу, когда достигнет края экрана. 2 стиль более правдив, ведь кот развернется, когда дойдет до края. А что означает 3 стиль? Поэкспериментируйте.

Ученики:

Кот по достижению края пойдет задом.

Учитель:

Хорошо. А теперь предлагаю сохранить вашу программу на компьютере.

Для этого в левом верхнем углу найдите кнопку «Файл» и нажмите «Сохранить как». В окне выбирайте «Рабочий стол» и «Сохранить». А теперь закрываем программу, нажав на красный крестик в правом верхнем углу.

Учитель:

Ребята, вы сегодня все большие молодцы. Расскажите, чему вы научились?

Ученики:

Знаем что такое скрипт и спрайт, научились перетаскивать блоки, знаем, как поменять фон сцены, научились задавать звук спрайту.

Учитель:

Всем спасибо, урок закончен.

Конспект урока на тему «Анимация имени».

Тема урока: Анимация имени.

Тип урока: урок усвоения новых знаний.

Цель урока: научить создавать простейшую анимацию в среде Scratch.

Задачи:*Образовательная:* создать условия для отработки навыков и умений работы в среде Scratch.

Развивающая: создать условия для развития внимания, памяти и воображения.

Воспитательная: содействовать развитию культуры речи и эстетического вкуса.

Этапы урока:

1. Организационный момент – 3 мин.

2. Практическая работа – 12 мин.
3. Физкультминутка – 2 мин.
4. Практическая работа (продолжение) – 13 мин.

Перечень средств ИКТ, используемых на уроке:

- Интерактивная доска, компьютеры для каждого ученика, программа Scratch.

СТРУКТУРА И ХОД УРОКА

1. Организационный момент 3 мин.

Учитель: *Приветствует учеников, отмечает присутствующих.*
Здравствуйте ребята. На предыдущем уроке мы с вами познакомились с простейшими действиями в программе Scratch. Давайте вспомним, чему мы учили нашего котика?

Ученики: Учили кота ходить, мяукать, отталкиваться от края экрана. Задавали новый фон сцены.

Учитель: Верно. Ответьте мне на вопрос что такое спрайт и скрипт?

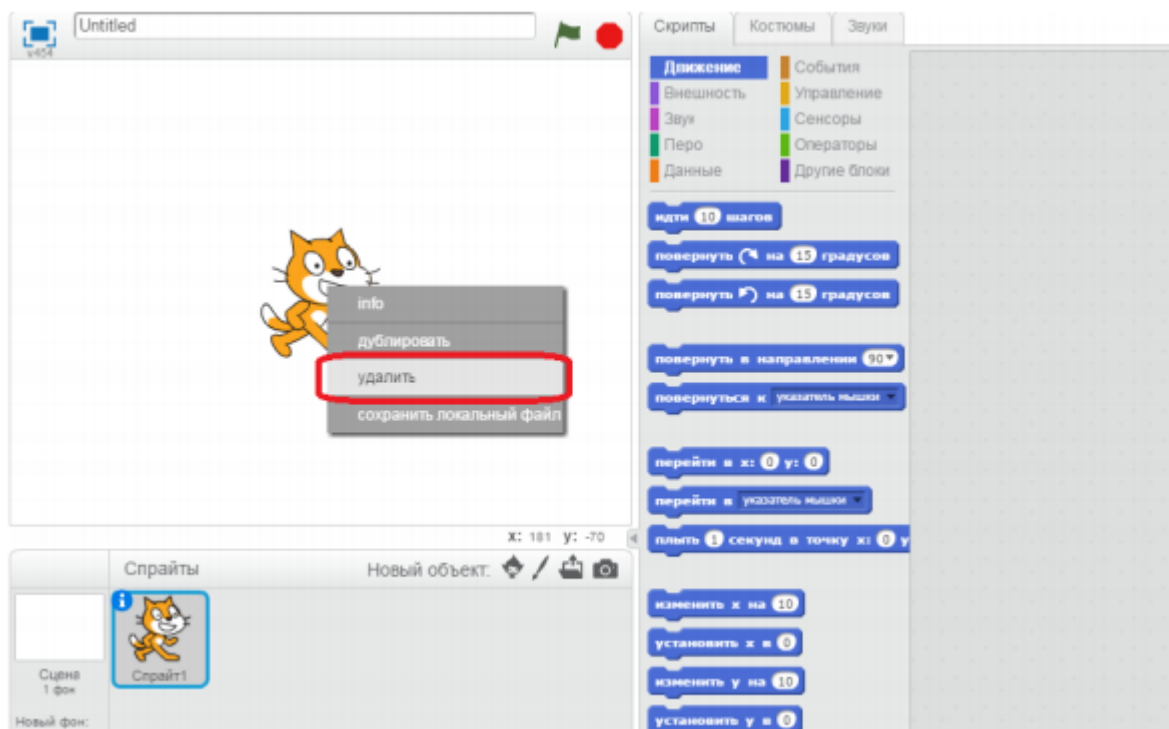
Ученики: Спрайт – это графический объект, выполняющий какие-либо действия в проекте (кот, машина, человек и т.д.)
Скрипт – это программа, которая описывает действия объекта (спрайта)

Учитель: Молодцы. Сегодня мы продолжим работу в Scratch. Вы почувствуете себя настоящими волшебниками, ведь мы будем оживлять буквы. Занимайте рабочие места за компьютерами.

2. Практическая работа 12 мин.

Запускайте программу.

Перед тем как писать буквы, нашего котика нужно удалить. В области сцены нажимаем на кота правой кнопкой мыши и нажимаем «удалить».

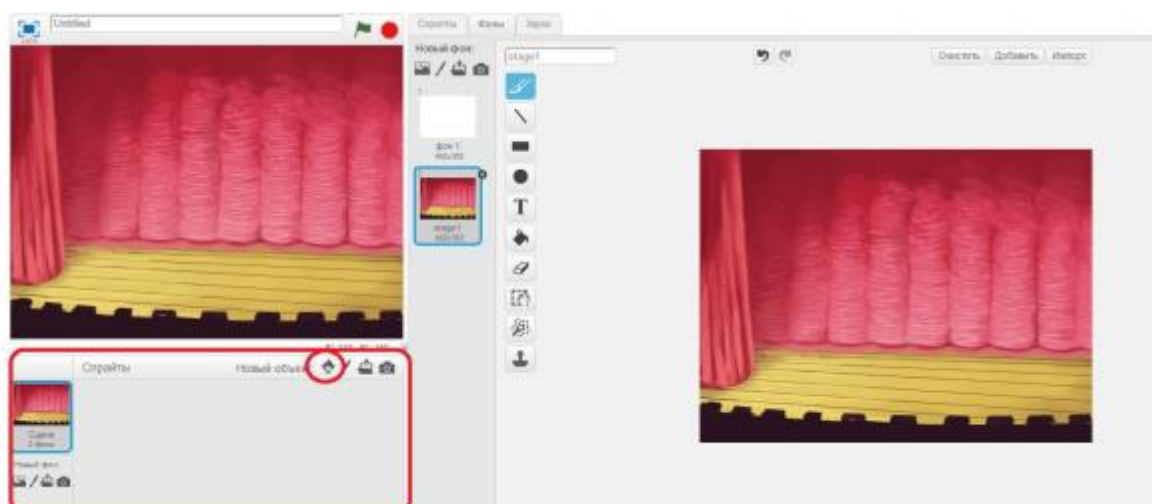


Теперь сцена пустая.

Задание 1. Фон и объекты

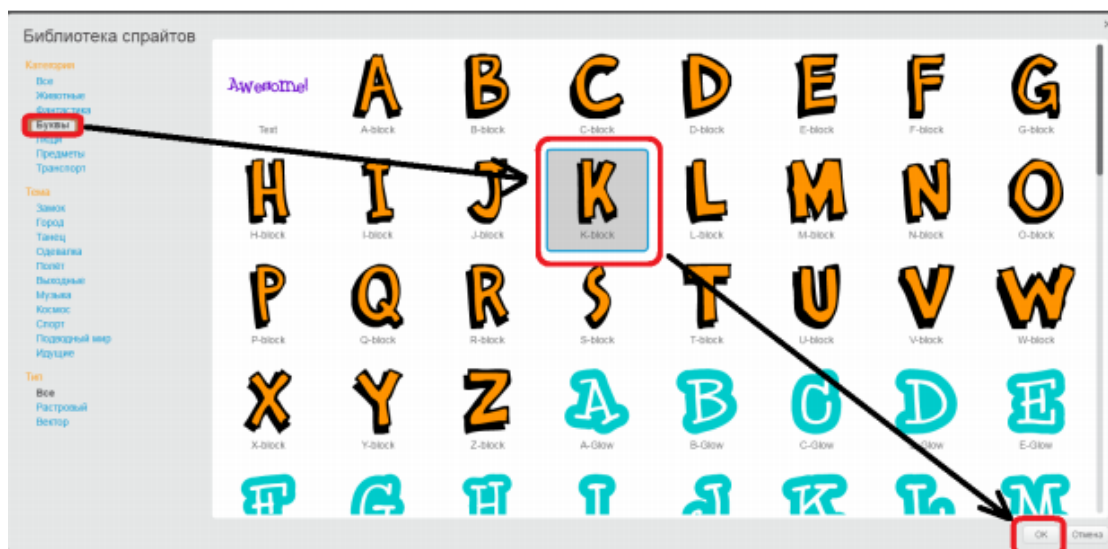
Задаем любой понравившийся фон. На предыдущем уроке мы делали это, вспоминайте. *Если есть затруднения, учитель подсказывает.*

Я выбрал розовую сцену. Далее добавляем объекты. Как это сделать? В левой нижней части находим окно спрайтов. Рядом с «Новый объект» находим человечка и кликаем на него.

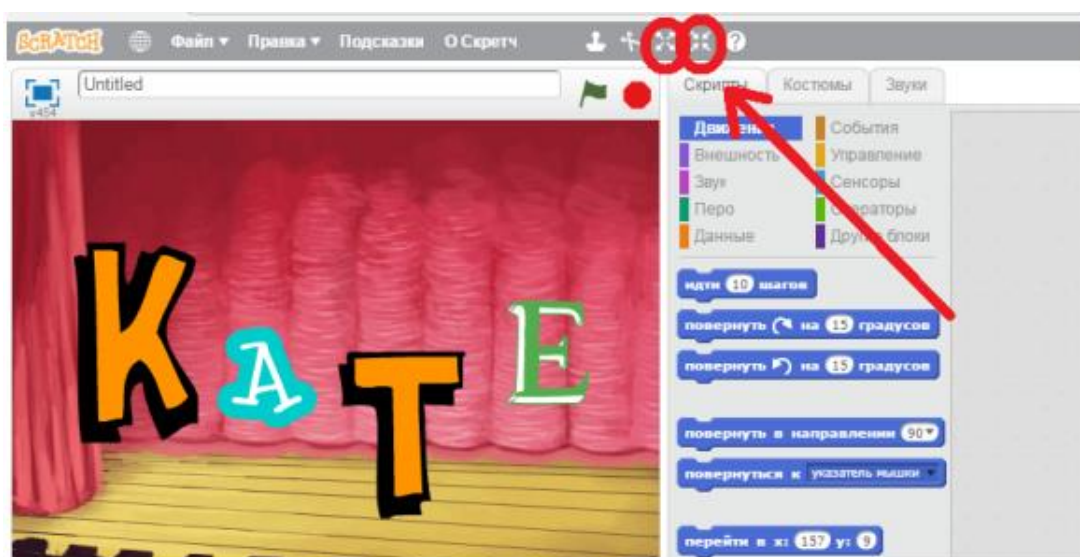


Откроется библиотека спрайтов. Посмотрите, какие бывают графические объекты.

Слева из списка выбирайте «Буквы». У каждого из вас получится разные слова. Давайте из букв составим ваши имена. *Учитель помогает детям написать имена на латинице. (Русских букв в программе нет).* Выбираем букву и нажимаем кнопку «ОК».



!Самостоятельно добавьте остальные буквы. После того как все спрайты добавили, вы можете менять расположение и их размер. Для того чтобы перетаскивать их с места на место, нужно зажимать левую кнопку мыши и не отпуская переместить на новое место. Для того чтобы изменять размер спрайтов нужно вверху в меню найти кнопочки «Увеличить» и «Уменьшить».



Нажимаем один раз на кнопку, например «Увеличить», ведем курсор на нужную букву и нажимаем на нее несколько раз, пока буква не станет

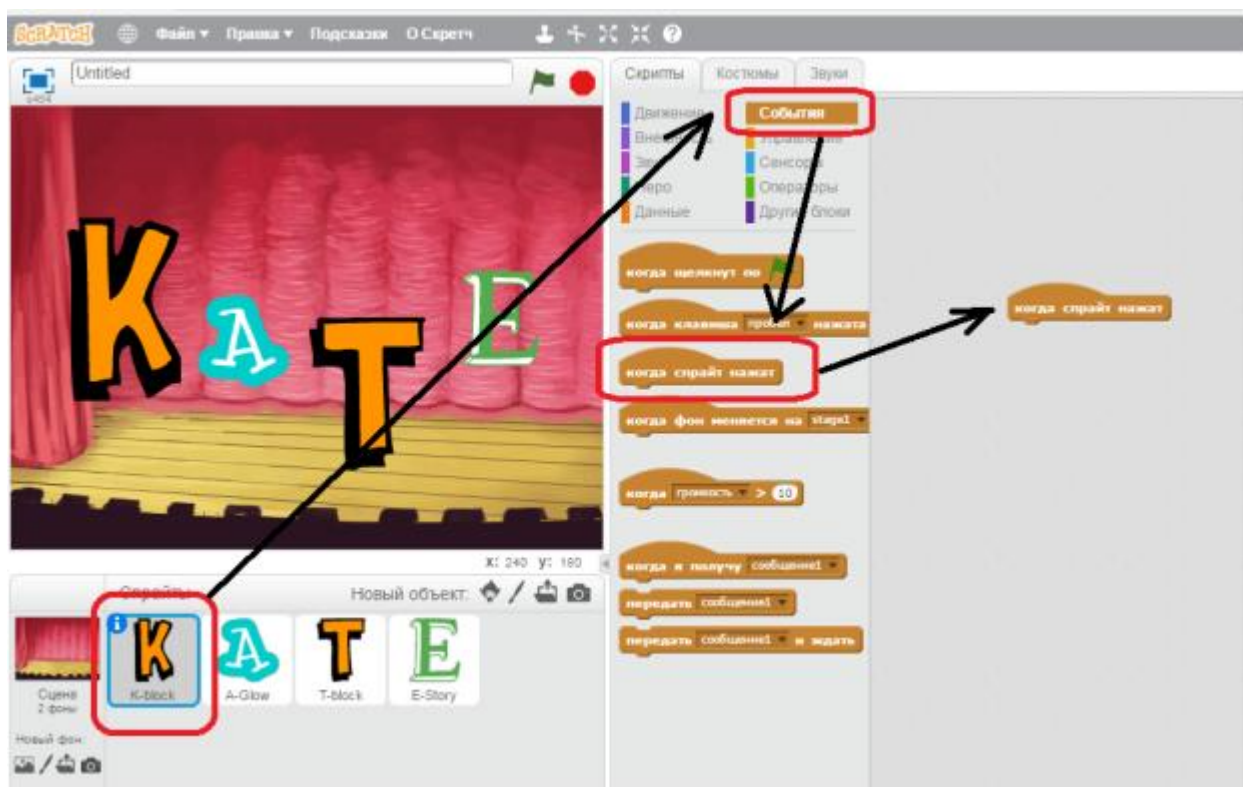
подходящего размера. Поэкспериментируйте с кнопками. У всех получилось? Если есть затруднения, учитель помогает.

Задание 2. Написание скрипта для первой буквы

Чтобы анимировать наши буквы, необходимо каждой написать свою программку (скрипт).

Выбираем в области скриптов первую букву (снизу слева).

Вверху посередине из раздела «события» выбираем блок «когда спрайт нажат» и перетаскиваем в рабочую область.



Из раздела «внешность» добавляем блок «изменить цвет эффект на 25». Посмотрите что получается в результате нажатия на данную букву в области сцены?

Ученики:

Буква меняет свой цвет.

Учитель:

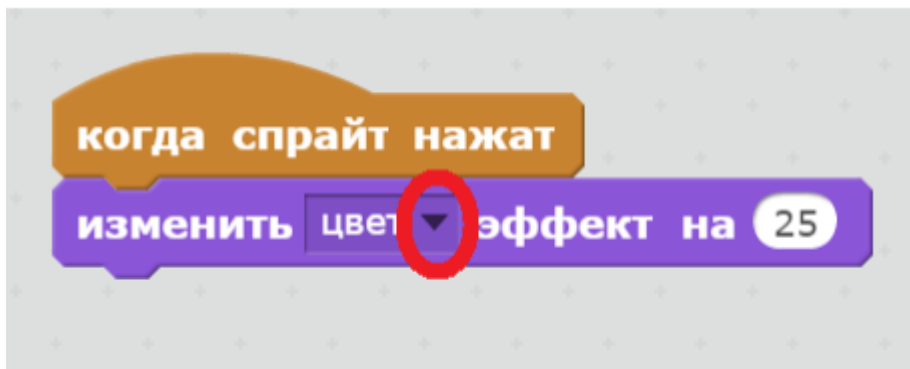
Верно. Попробуйте вместо числа 25 ввести цифру 5. В чем разница?

Ученики:

Буква по-прежнему меняет цвет, но переход от цвета к цвету стал плавней.

Учитель:

Молодцы. Теперь на блок «изменить цвет эффект на 25», раскройте список, нажав на стрелочку.



Рядом с красной кнопкой – зеленый флажок. Эта кнопка запускает наш проект. Давайте оставим из предложенных эффектов – цвет. Ребята, давайте прервемся и проведем физкультминутку.

3. Физкультминутка 2 мин.

Встаем из-за парт. Повторяем за мной.

Ручки кверху поднимаем, а потом их опускаем. А потом их развернем и быстрее к себе прижмем. А теперь скорее скорее, хлопай хлопай веселей. Приседай скорее со мной, держим руки за спиной. Встали прямо, руки вбок, вправо - влево поворот. Молодцы, давайте повторим еще раз, а вы постарайтесь повторять не только движения, но и слова.

4. Практическая работа (продолжение) 13 мин.

Продолжаем нашу работу, садимся за рабочие места. Мы остановились на том, что написали скрипт для первой буквы. Теперь добавим скрипты для остальных букв.

Задание 3. Скрипт для второй буквы.

Скрипт для второй буквы будет отличаться. При нажатии на спрайт, он будет менять наклон.

1. Для этого выбираем в области спрайтов вторую букву.

2. В окне «Скрипты» выбираем раздел «событие» и блок «когда спрайт нажат». Перетаскиваем в рабочую область.



3. Из раздела «движение» выбираем блок «повернуть вправо на 15 градусов».

4. Посмотрите результат. Измените число 15 на 100. Что получится?

Ученики:

Буква сильно изменит положение.

Учитель:

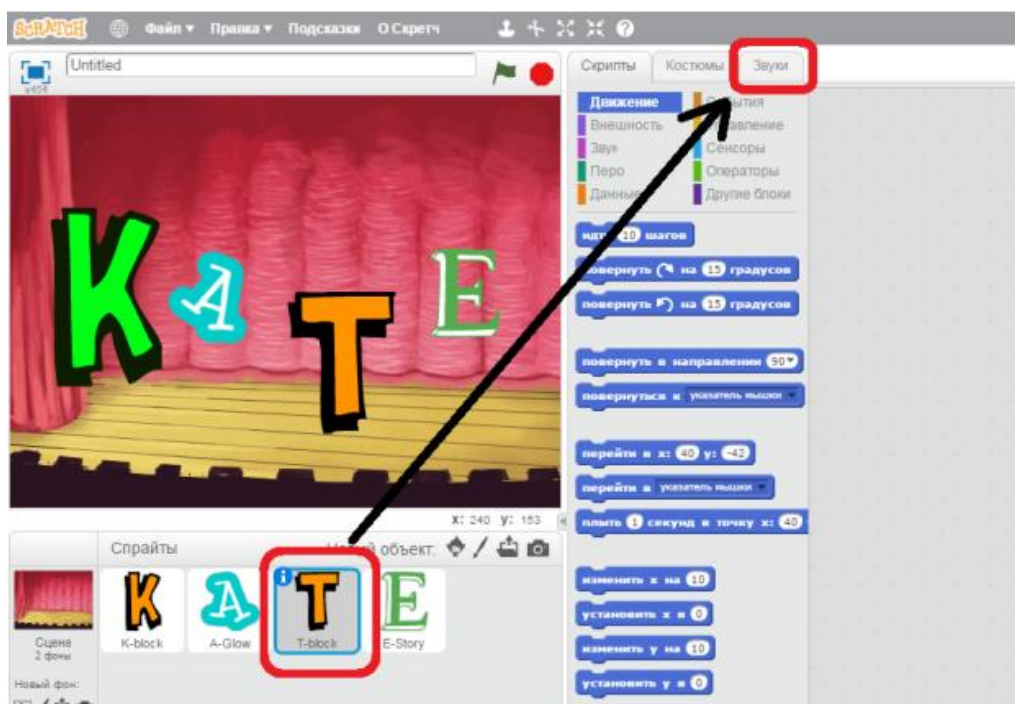
Верно: 360 градусов – это полный круг. Мы не увидим результат. 180 градусов – это половина круга. 90 градусов – это четверть круга. Поэкспериментируйте с градусами.

Задание 4. Скрипт для 3 буквы.

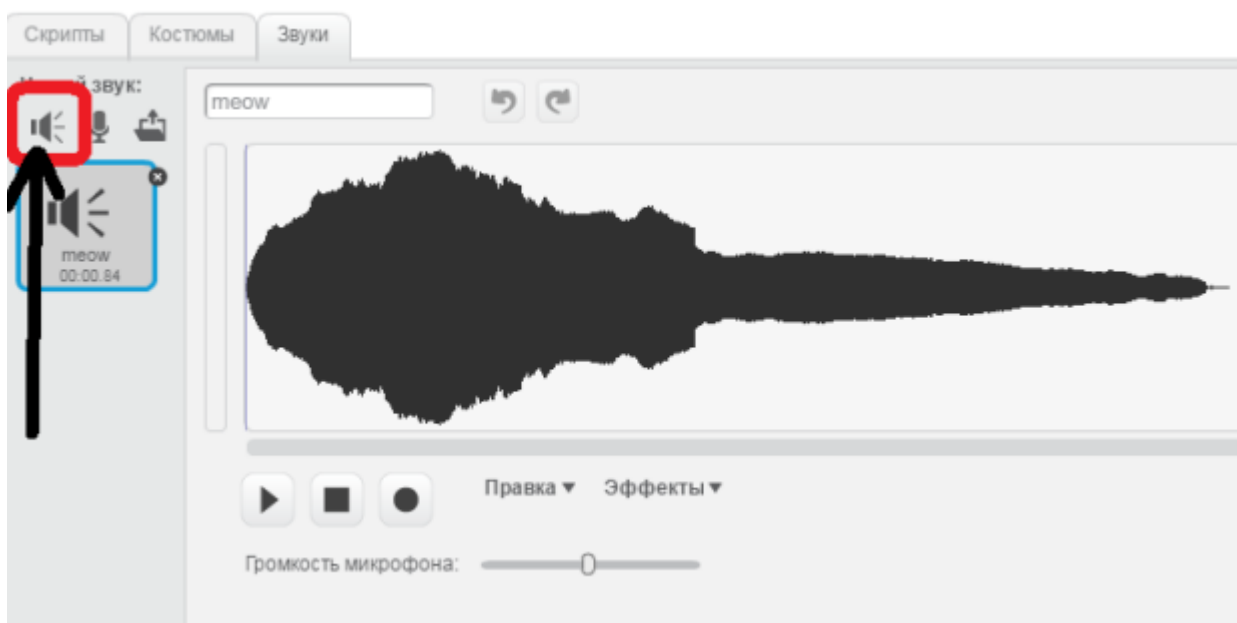
К третьей букве мы добавим звук. Помните, на предыдущем уроке мы учили кота мяукать?

1. В области скриптов выбираем третью букву.

2. Переходим на вкладку «звуки».



4. У нас добавлен только звук «meow». Нажмите на кнопку «библиотека звуков».



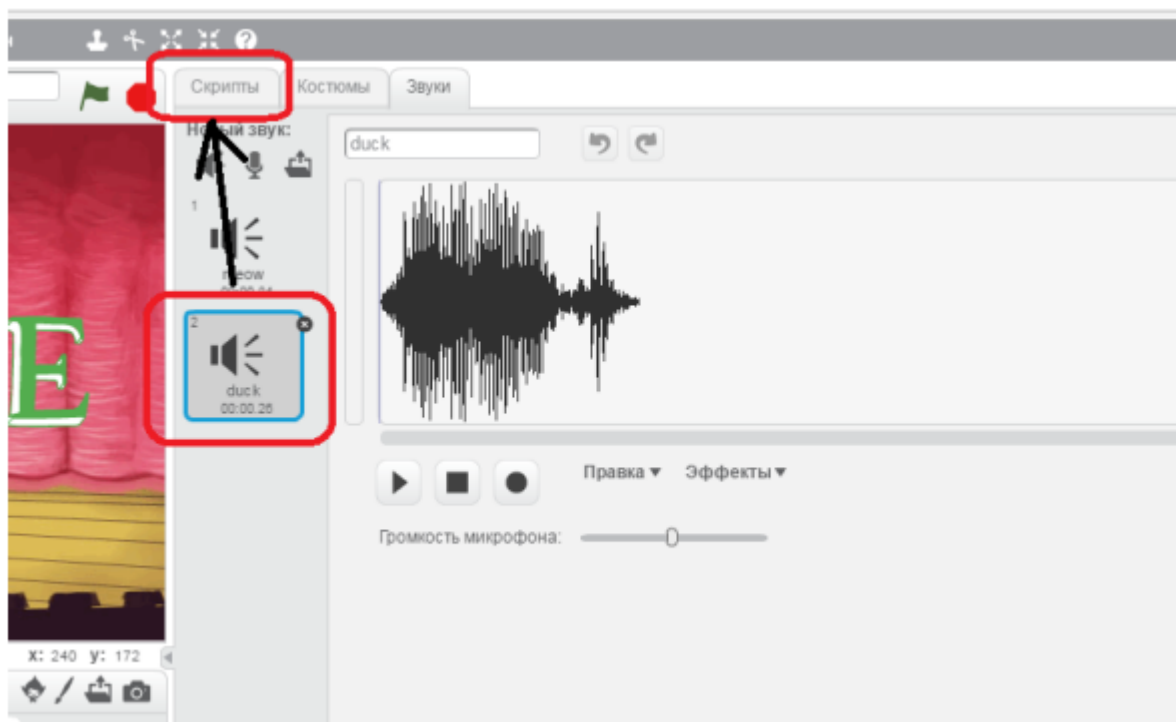
5. Посмотрите какие могут быть звуки. Вы можете их прослушать, нажав на треугольник.



6. Выберите любой понравившийся звук и запомните название. Нажмите на кнопку «ОК».



7. Посмотрите, кроме звука «теow» у вас добавился новый звук.



8. Переходим на вкладку скрипты. Из раздела «события» снова добавляем блок «когда скрипт нажат». Из раздела «звук» добавляем блок «играть звук». Попробуйте в области сцены нажать на третью букву. Получилось? *Если есть затруднения, учитель помогает.*

! Самостоятельно.

Теперь самостоятельно выберите скрипты для оставшихся букв и напишите их. Изменить цвет буквы, изменить наклон или добавить звук. После того,

как все скрипты написаны, необходимо изменить имя проекта и сохранить на рабочий стол. Кнопка «Файл», «Сохранить как». В окне выбирайте Рабочий стол» и «Сохранить». А теперь закрываем программу, нажав на красный крестик в правом верхнем углу.

Учитель:

Ребята, вы сегодня все большие молодцы. Расскажите, чему вы научились?

Ученики:

Научились: добавлять новые объекты (спрайты), задавать звук, менять угол наклона спрайта, изменять цвет объекта.