

Муниципальное бюджетное общеобразовательное учреждение средняя
общеобразовательная школа с. Русский Камешкир

**Разработка проекта «Cats Man»
в среде визуально-интуитивного
программирования Scratch**

(научно-исследовательская работа)

Работу выполнил:
ученик 8 «А» класса
Чубаров Даниэль.

Руководитель работы:
учитель высшей
квалификационной категории
Макина И. В.

«Старт в науку»

2020

ОГЛАВЛЕНИЕ

Введение	3
Цель и задачи	6
Основные компоненты проекта Scratch: спрайты и скрипты	
1. Интерфейс Scratch.....	7
2. Сцена.....	7
3. Добавление и удаление спрайта.....	9
4. Окно блоков и область спрайта.....	10
5. Костюмы.....	12
6. Переменные.....	13
7. Числа, строки, выражения.....	15
8. Ввод значений с клавиатуры.....	16
Проект «Cats Man» в среде программирования Scratch	
1. Первый уровень.....	18
2. Второй уровень.....	18
3. Третий уровень.....	19
4. Четвертый уровень.....	19
5. Пятый уровень.....	20
6. Шестой уровень.....	21
Заключение	22
Литература	24
Приложения	
Приложение 1. Анкетирование одноклассников.....	25
Приложение 2. Среда программирования Scratch 2.0.....	26
Приложение 3. Компьютерные игры.....	27
Приложение 4. Группа «Звук».....	28
Приложение 5. Группа «Переменные».....	29
Приложение 6. Группа «Операторы».....	31
Приложение 7. Диплом ГАОУ ДПО ИРР ПО.....	32

Введение

Алгоритмизация и программирование... Это самое сложное, что есть в курсе информатики в основной школе. Изучая языки программирования, дети теряют интерес к предмету в процессе изучения синтаксиса и грамматики языка. Можно сказать, что они «за лесом не видят деревьев». Синтаксические проблемы описания циклов и ветвлений многим кажутся непреодолимыми. В среде Scratch все эти проблемы снимаются, к тому же новая версия русскоязычная, поэтому процесс освоения базовых алгоритмических конструкций ускоряется на порядок.

В Scratch можно смешивать различные типы информации: графику, фотографии, музыку и звуки в свободном творческом полете. Скретч берет все лучшее от вычислительной техники и дизайна интерфейсов для того, чтобы сделать процесс программирования более привлекательным и доступным для детей, подростков и вообще всех тех, кто хочет научиться программированию. В основе Scratch лежит графический язык, отличающийся от всех прочих, прежде всего своей исключительной простотой.

Scratch базируется на традициях языка Лого и Лего-Лого. Если первый язык, быть может, не всем еще знаком в нашей стране, то о конструкторах Лего имеют представление абсолютно все. Скретч использует своеобразную метафору кирпичиков Лего, из которых даже самые маленькие дети могут собрать простейшие конструкции. Затем, кирпичик за кирпичиком, ребенок может совершенствовать свои навыки программирования.

После того, как проект создан, вы можете поделиться им со всем миром на сайте Скретч. Также можно выложить созданные видео-ролики и фотографии на YouTube. Или же можете вставить ваш Scratch-проект в любую другую веб-страницу - например, Facebook. Если вам понравился

какой-либо персонаж или картинка, просто используйте его целиком или частично в собственном проекте.

В результате выполнения простых команд у нас может возникнуть сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Такой начальный уровень программирования столь прост и доступен, что сейчас Скретч рассматривается в качестве средства обучения не только старших, но и младших школьников. В нашей школе Скретч изучают и в 5-м, и в 10-м классах. Но, не смотря на такую разницу в возрасте, всем одинаково интересно выполнять проекты в этой среде.

Актуальность темы и заключается в том, что многие дети и взрослые любят играть в компьютерные игры, которые остаются востребованными и по сей день. В настоящее время, в связи с увеличением спроса на различные гаджеты, технические устройства, растет и спрос на программистов. «Только для созидания должны вы учиться!» - сказал Фридрих Ницше. Вот смысл учения – вырасти и изменить мир, сделать его лучше. Чтобы создать какую-нибудь программу, надо учиться программировать.

Я провел опрос среди одноклассников по вопросам: часто ли вам приходится работать с компьютерной техникой, и все ответили, что каждый день. Любите ли вы играть в компьютерные игры? Кто может создать свою компьютерную игру? Результат я оформил в виде диаграммы (см. Приложение 1).

Изначально Скретч создавался для детей и подростков до 16 лет, позволяя им самостоятельно заниматься компьютерным творчеством на базе подростковых компьютерных клубов. Причем разработчики этой технологии (Митч Резник и Алан Кей) ориентировались на малообеспеченные слои населения и ставили своей целью увести подростков с улицы с ее часто отрицательным влиянием. В мае 2013 года официальный сайт Scratch перешел на версию 2.0. В данной версии: новый интерфейс, новая онлайн-среда, которая доступна всем пользователям. Скретч 2.0 позволяет создавать, редактировать

и просматривать проекты прямо в браузере (см. Приложение № 2).

Таким образом, благодаря использованию технологии Скретч, школьники получают возможность:

- постепенно учиться программированию;
- реализовать свои творческие порывы;
- участвовать в интерактивном процессе создания игр и анимирования разнообразных историй как индивидуально, так и вместе со своими сверстниками из разных стран;
- получать живой отклик от единомышленников;
- оценить свои творческие способности.

С данной программой я познакомился на занятиях, посещая кружок «Программирование на Scratch». Выполнил несколько проектов по созданию компьютерных игр (см. Приложение № 3), захотелось создать что-то свое, усложнив изученные проекты, добавить более сложные элементы и посмотреть, как это будет работать. Так возникла идея создания компьютерной игры «Cats Man». Этот проект рассчитан на младший и средний школьный возраст. Он носит развлекательный характер, позволяет развивать внимательность, логическое мышление и координацию.

Цель исследования - применение современных инноваций в области визуально - интуитивного программирования Scratch.

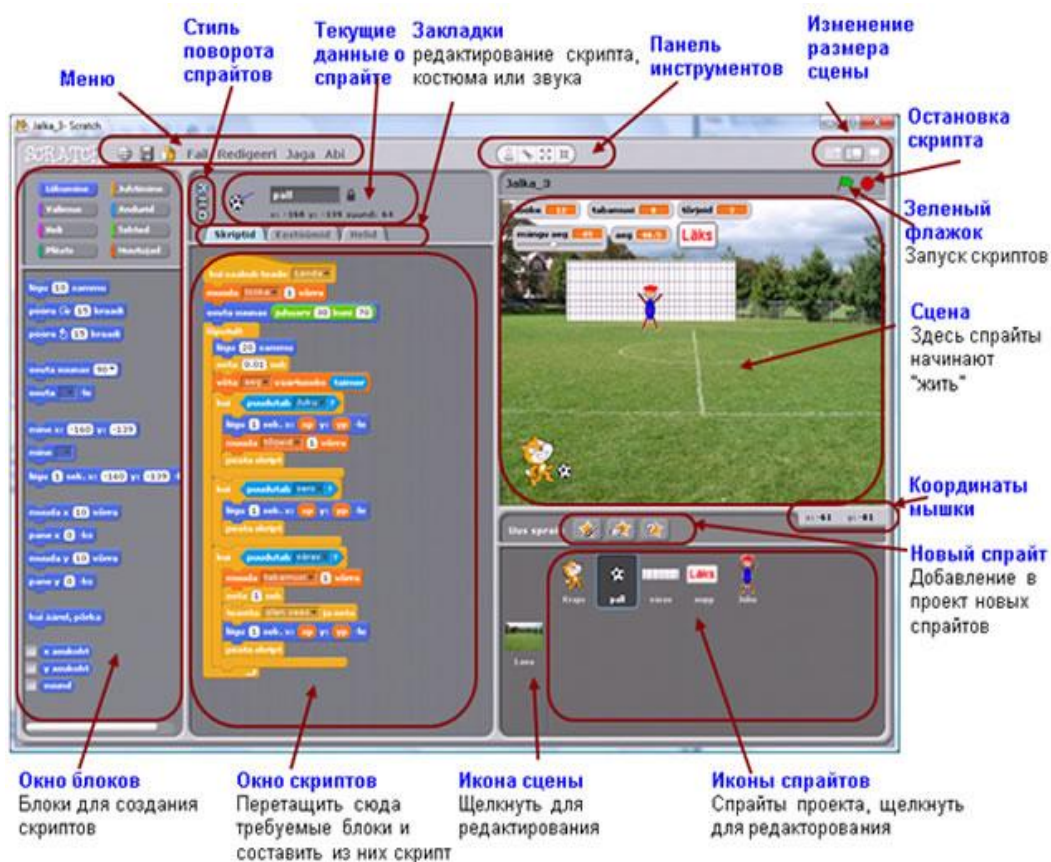
Для достижения поставленной цели были сформулированы следующие **задачи**:

- изучить среду программирования Scratch, ее возможности и характеристики;
- разработать свой проект и создать готовый программный продукт «Cats Man» в среде Scratch;
- развивать интеллектуальные, познавательные и творческие способности, используя возможности программирования на Scratch.

Гипотеза: каждый может научиться создавать игры с помощью программы Scratch, проявляя своё воображение и творческие способности.

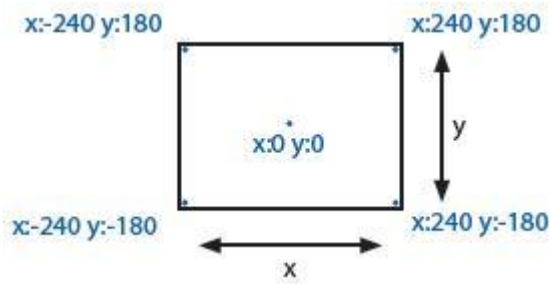
Основные компоненты проекта Scratch: спрайты и скрипты

1. Интерфейс Scratch



2. Сцена

Сцена (*stage*) - это место, на котором осуществляется выполнение созданных проектов: игры, истории, анимации и т.п. На сцене спрайты передвигаются и общаются между собой. Ширина сцены 480 единиц и высота 360 - единиц. Единица является условной и ей не соответствует никакая единица измерения. Значения x и y координат центральной точки сцены равны 0.



Если курсор мышки передвигать по сцене, то под сценой в правом углу видны текущие значения координат x-y в виде:

x: 84 y: -89

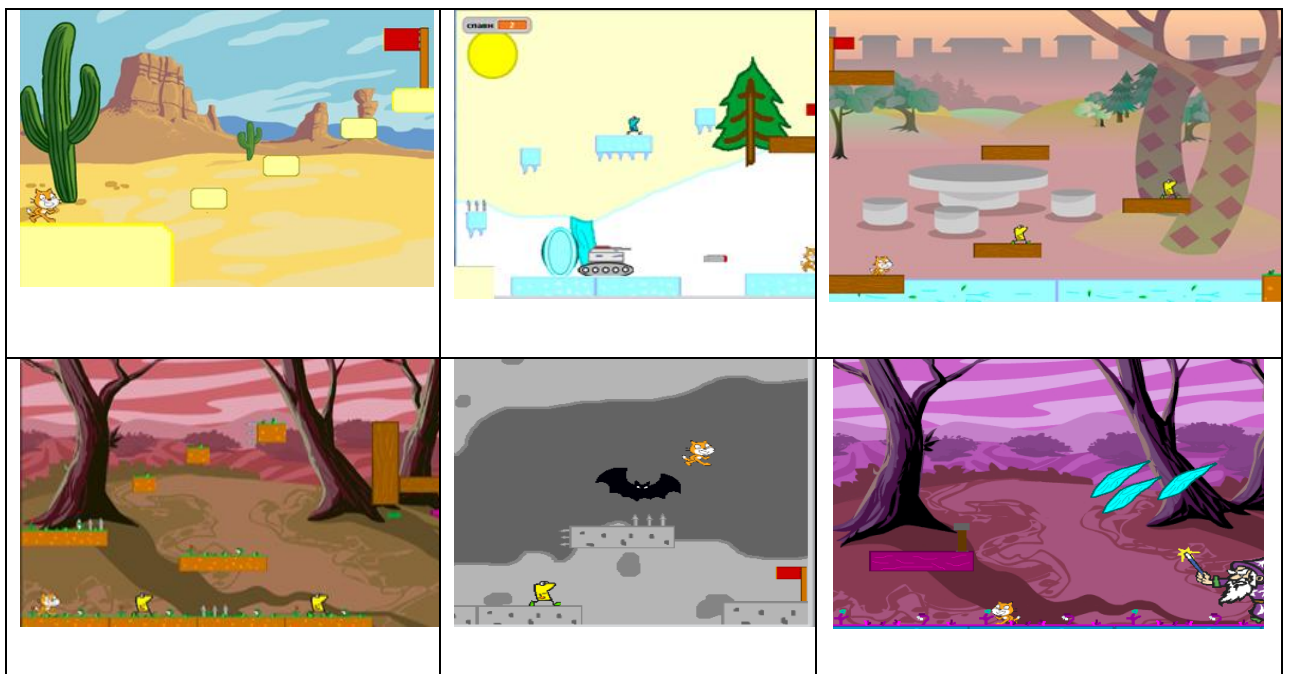


Если щелкнуть по кнопке режима представления, то можно видеть и использовать результат проекта в режиме презентации. Для выхода из этого режима следует щелкнуть по клавише **Esc** или по стрелке в левом верхнем углу сцены.



Если щелкнуть по кнопке, задающей изменение состояния сцены, то можно выбрать большую или маленькую сцену. При выборе маленькой сцены увеличивается область скриптов, что полезно в случае больших программ.

В моем проекте используются следующие сцены:



3. Добавление и удаление спрайтов (объектов)

При запуске нового Scratch-проекта на сцене имеется только стандартный объект - кот. Для добавления (создания) нового спрайта можно использовать приведенные ниже кнопки.



Нарисовать новый спрайт. Нарисовать новый костюм для спрайта с помощью Редактора для рисования (Paint Editor).



Выбрать новый спрайт из файла. Можно выбрать новый спрайт в каталоге спрайтов Scratch (папка **Костюмы**) или свой графический файл из произвольной папки (типы JPG, BMP, PNG, GIF и т.п.).



Выбрать случайный спрайт. Scratch предлагает случайный спрайт из своего каталога.

В списке спрайтов отображаются криптограммы всех используемых в проекте объектов. Для каждого спрайта указано его имя и количество скриптов (процедур). По умолчанию спрайтам задаются имена Имя спрайта можно изменить в окне текущей информации в области имени

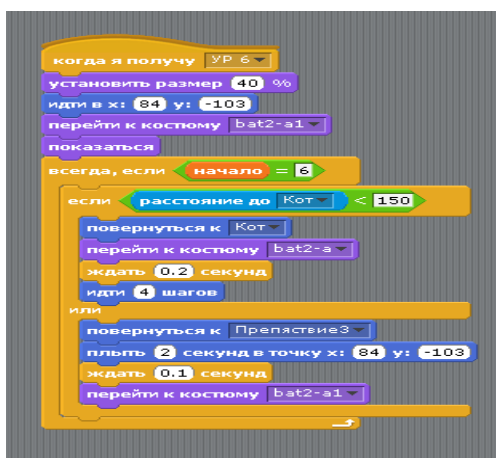


4. Окно блоков и область скриптов

Для программирования спрайта блоки из *Окна блоков* перетаскиваются в область скриптов. Здесь возникает копия блока, оригинал остается на своем месте. Запуск блока осуществляется с помощью щелчка по нему. Щелкнув по блоку правой кнопкой мышки и выбрав из появившегося меню *помощь*, можно получить вспомогательную информацию о блоке. Там же находится и команда дублирования.

Для создания скрипта (процедуры) блоки объединяются в стек (stack). Специальные элементы интерфейса позволяют объединять блоки только одним способом. Щелчок по скрипту в любом месте, запускает последовательное выполнение его блоков сверху вниз.

Ниже приведен пример скрипта (на русском языке). Он выполняет следующий фрагмент программы. Если Кот, перемещаясь по глыбам, попадает в зону обитания Мыши, а это окружность радиусом 150 единиц, Мышь начинает преследовать Кота, тем самым создает препятствия для достижения цели. В случае касания Кота и Мыши, главный герой окажется в исходной точке шестого уровня.



При перетаскивании блока в существующий скрипт (стек), появляется светлая белая полоска, которая показывает место, в которое можно

поместить блок так, чтобы он объединился с другими блоками. Можно добавлять блоки перед стеком, в середину и в конец. Для удаления блока из *Области скрипта* можно просто его перетащить в любое место *Окна блоков* и блок исчезнет.


У некоторых блоков, таких как , имеется раскрывающееся меню. Если щелкнуть по стрелке (треугольнику), то открывается меню, в котором можно выбрать необходимое значение, например, в данном случае номер ноты (см. Приложение № 4). С данными блоками я работал, когда создавал звуковое сопровождение. Например, при прохождении Уровня 4 я использовал инструмент 51 и создал мелодию (см. Рис. 1), а для работы Уровня 5 – инструмент 89 (см. Рис. 2).



Рис. 1

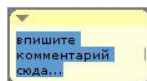


Рис. 2

Для очищения области скриптов следует щелкнуть правой кнопкой мыши в любом месте области скриптов и в появившемся меню выбрать команду *Очистить*. Для сохранения скриптов в виде картинки следует щелкнуть правой кнопкой мыши в любом месте области скриптов и в появившемся меню выбрать команду *Сохранить изображение скрипта*. Для добавления комментария щелкнуть правой кнопкой мыши в любом месте области скриптов и в появившемся меню выбрать команду *Добавить*

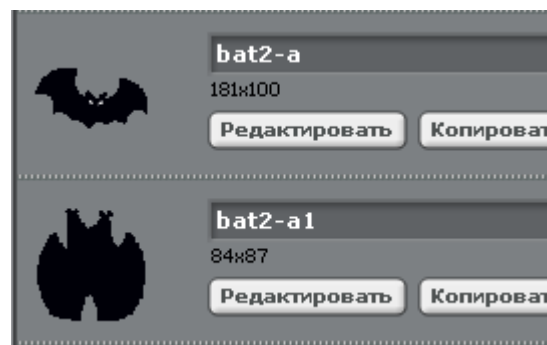
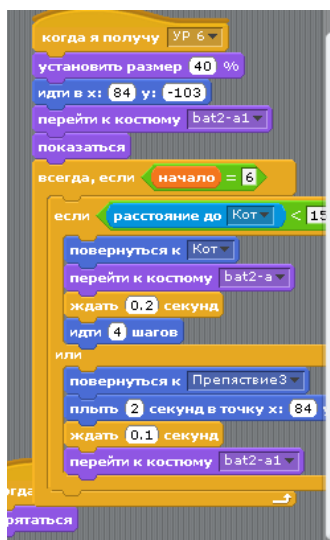
комментарий. Появится желтая область для комментария, в которую можно

ввести пояснения.



5. Костюмы


Костюм представляет собой "двойника" для данного объекта, который имеет другой внешний вид. Если имеется несколько костюмов, то спрайт может их менять. В первую очередь это используется для создания эффекта анимации. Для просмотра и редактирования костюмов следует щелкнуть по закладке *Костюмы*.



У данного спрайта два костюма, которые представляют разные положения спрайта и которые можно использовать для имитации полета. Текущий костюм выделен и окружен синей рамкой. Для переключения на другой костюм следует просто щелкнуть по его криптограмме. Смена костюмов обычно происходит при выполнении скриптов, обычно с помощью блока *следующий костюм*.

Для создания нового костюма имеется четыре возможности:

- Нарисовать его с помощью **Редактора рисования**, который можно запустить кнопкой **Рисовать** (Paint)

- Импортировать его из каталога спрайтов (костюмов) Scratch или из графического файла: кнопка **Импорт** (*Import*). Scratch признает несколько графических форматов: JPG, BMP, PNG, GIF (включая анимированный GIF).
- Щелкнуть по кнопке **Камера**, если к компьютеру подсоединена веб-камера. Каждый раз при щелчке по кнопке  или нажатии на пробел, делается фото, которое становится новым костюмом.
- Перетащить одну или несколько картинок с рабочего стола или из Интернета.

6. Переменные

Переменная – это поименованное место в памяти компьютера: поле памяти или ячейка памяти. Программа (скрипт) может во время выполнения записывать туда значения, изменять их и использовать (считывать) для нахождения новых значений (см. Приложение 5). Значение переменной в Scratch можно изменять и вручную. Значением переменной может быть число, текст или логическое значение (истина - true, ложь - false). В каждый момент времени переменная может иметь только одно значение. Запись в ячейку значения переменной называется присваиванием. При присваивании переменной нового значения ее предыдущее значение теряется.

Переменная может быть связана только с одним спрайтом - локальная переменная, или быть общей для всех спрайтов проекта - глобальная переменная. К локальной переменной имеется непосредственный доступ только из скриптов данного спрайта. Область действия переменной или ее видимость (доступность) задается во время ее создания.

Создание переменной происходит с помощью команды Создать переменную в группе Переменные. При выполнении этой команды Scratch предлагает определить имя и область действия новой переменной с помощью диалогового окна. Имена локальных переменных могут быть одинаковыми у нескольких спрайтов, но они не могут совпадать с именем глобальной переменной.

После создания первой переменной появляются соответствующие блоки, связанные с переменными: блок ссылки с именем переменной, блоки стека: поставить в (set to), изменить на (change), показать переменную (show variable) и спрятать переменную (hide variable). При добавлении следующих переменных появляются только блоки ссылок с именем переменной.

Блок ссылки с именем переменной можно использовать в скриптах в блоках стека для ссылки на нее, в том числе и в выражениях. Перед блоком расположен флажок, с помощью которого можно выводить на сцену или прятать монитор (окно со значением) с текущим значением переменной. Внутри блоков стека имеется раскрывающееся меню, с помощью которого можно выбрать имя требуемой в скрипте переменной. Два первых блока (*поставить в* и *изменить на*) имеют поле, в которое можно ввести значение или поместить блок со ссылкой на имя переменной, выражение и т.п.

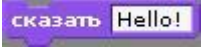

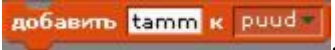


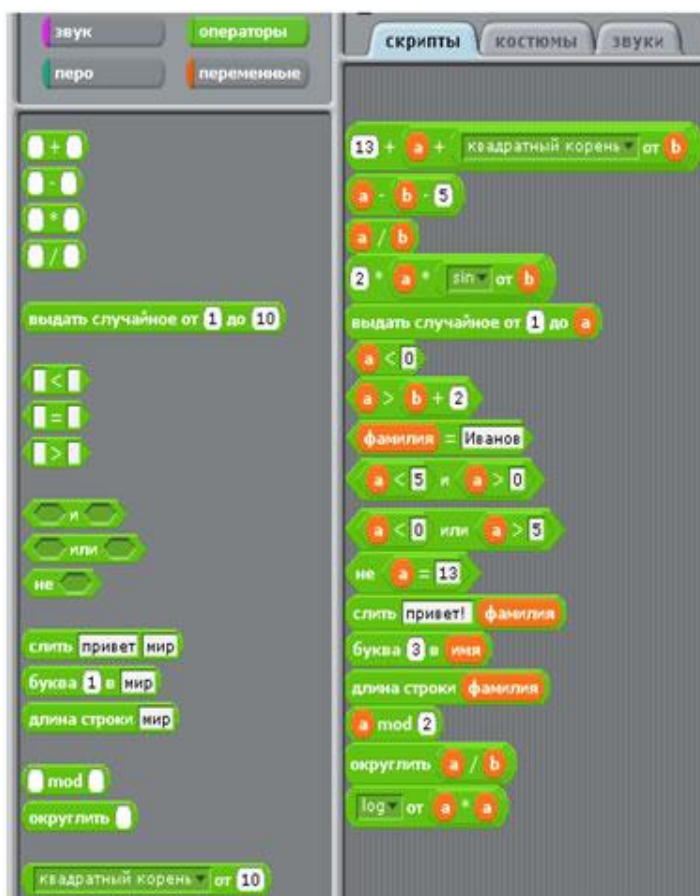
Текущее значение переменной можно отобразить в окне **Сцены** с помощью соответствующего монитора, если включить флажок в **окне блоков** команд перед именем переменной.

Вид монитора переменной в окне **Сцены** (нормальный, большой, слайдер) можно задать с помощью его меню (щелчок правой кнопки мыши). Значение переменной можно изменять вручную, используя **слайдер**.

7. Числа, строинги (тексты), выражения

В Scratch можно непосредственно использовать значения двух типов: **числа** и **строинги**. Неявно участвуют и **логические** величины, которые являются результатом сравнений и логических выражений. **Числа** могут быть представлены константами, которые вводятся в соответствующие места блоков, как, например, , , , а также могут быть значениями переменных и элементов списков. С числами можно выполнять арифметические действия (+, -, *, /), сравнения (<, =, >). Для числовых значений предусмотрены и некоторые функции.

Строинги (или тексты) - это произвольные последовательности знаков. Они также могут использоваться как константы, как, например, в блоке , а также присваиваться переменным и элементам списков, как в приведенных блоках , . Строинги можно между собой сравнивать. Для строингов определены три операции (объединить, выделить и нахождение длины), которые реализованы с помощью блоков ссылок.



С помощью выражения можно задать правило для нахождения значения. Выражение состоит из операндов и знаков операций. Операнды могут быть константами, переменными и функциями. Операции подразделяются на **арифметические**: $+$, $-$, $*$, $/$; **сравнения**: $<$, $=$, $>$ и **логические**: **и**, **или**, **не**. Для представления выражений используются специальные блоки, входящие в состав группы **Операторы** (см. Приложение № 6).

8. Ввод значений с клавиатуры

С помощью блока **спросить** **и ждать** скрипт может задать вопрос, на который пользователь может ответить, введя данные с клавиатуры. При выполнении блока выполнение скрипта приостанавливается, выноска с вопросом выводится рядом с соответствующим спрайтом и снизу на сцене появляется поле для ввода текста, в которое можно с клавиатуры набрать требуемое значение (текст или число). После нажатия на клавишу Enter или на флажок справа от поля ввода блок **ответ** получает введенное пользователем значение и работа скрипта продолжается. Значение блока **ответ** можно рассматривать как глобальную переменную специального вида, которая доступна всем спрайтам.

При выполнении следующего блока **Спросить** новое введенное значение заменяет предыдущее значение блока **ответ**. Если введенное

значение необходимо в дальнейшем использовать, то его можно присвоить переменной.

Пример.



После выполнения блока **Спросить** значение блока **ответ** присваивается введенное значение. Если это число равно 2, то начинается игра со второго уровня.

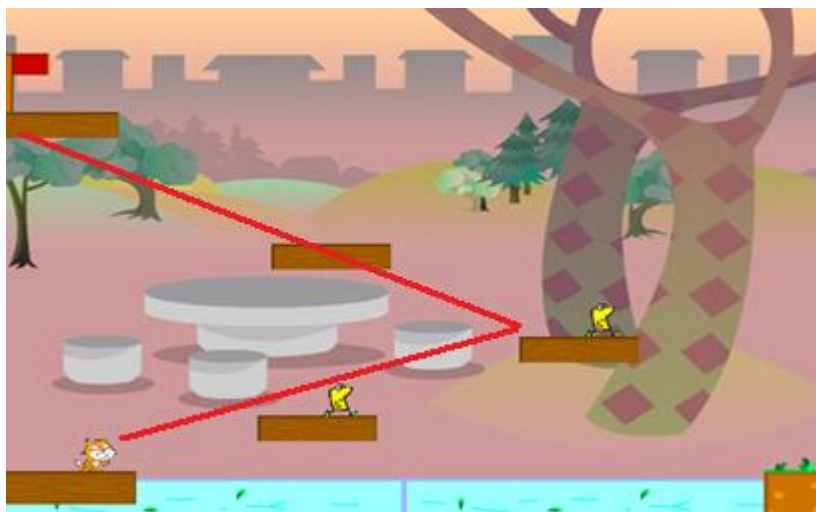
Разработка проекта «Cats Man» в среде программирования Scratch

1. Первый уровень



Цель: Кот должен дойти до флажка по платформам. Управление объектом осуществляется с помощью стрелок на клавиатуре и пробела.

2. Второй уровень



Цель: Кот также должен дойти до флажка, его маршрут должен пройти по деревянным платформам. Сложность прохождения заключается: 1) на платформе дежурит Головастик. Если до него дотронешься, окажешься в исходной точке, и надо начинать все заново; 2) водоем, в который Кот можно провалиться и, также, вернет главного героя на начало второго уровня.




3. Третий уровень



Объект перемещается по зеленым полянам, с целью – добраться до флажка. Прохождение усложняется. Кроме Головастика, на сцене появляется Стрелочник, который вертикально перемещается и каждые 9 с выпускает стрелу, которая может поразить нашего героя, что вернет его в исходное положение. Кроме этого, на поляне может появиться Ловушка – из земли появляются иглы, при касании с ними, Кот возвращается в начало пути.

4. Четвертый уровень

Перед нами зимний пейзаж. Коту необходимо по льдинам добраться до флажка. Помехи на данном уровне более существенные. Есть Портал, из которого появляются в хаотичном порядке Бегоножка, Танк и Летун. Бегоножка вертикально перемещается по льдине и, тем самым, создает помехи для главного героя. Танк выпускает снаряды, которые направлены на Кота, а Летун перемещается по сцене, с целью догнать Кота и вернуть его в начальное положение. На предпоследней льдине дежурит Головастик, Скорость его движения намного выше, чем на предыдущих уровнях. Все помехи созданы только с той целью, чтобы Кот не смог дойти до красного флажка.

		
1. Бегоножка	2. Танк	3. Летун

5. Пятый уровень



Мы оказываемся в пещере, под землей. Перемещаться Кот может по глыбам. Но пройти путь к флажку мешают:

- 1) Стрелочник, который выпускает стрелы в горизонтальном направлении;
- 2) Иглы, которые появляются на глыбе каждые 4 с;
- 3) Головастик, который перемещается по глыбе;
- 4) Мышь, которая начинает догонять Кота, когда тот попадает в зону ее обитания.

В любом случае касания Кота кого-то из злодеев, происходит его возврат в начальную точку уровня.

6. Шестой уровень



Шестой уровень – заключительный. Цель – успешно его пройти и победить Звездочета. Для этого надо добраться до Рычага и трижды его нажать. При этом, после каждого нажатия, рычаг пропадает на 5 с. А это время надо как-то продержаться. Кругом полно злодеев: 1) камни, которые падают сверху через определенный промежуток времени, и это время нефиксированное, оно задается случайным образом - от 2с и более; 2) ледяные стрелы, которые перемещаются в зависимости от местоположения кота и направлены на его поражение; 3) световые молнии, которые выпускает Звездочет. Движение всех объектов осуществляется со звуковым сопровождением. После успешного прохождения Звездочет со словами: «Ты победил меня!» исчезает, и реликвия Кота к нему возвращается.

Заключение

Scratch не только язык программирования, но и удачная среда для проектной деятельности, поскольку все необходимое для такой деятельности включено в его состав. Можно выделить следующие имманентные свойства Scratch, имеющие значительный потенциал:

- Простота и дружелюбность интерфейса позволяют начинать изучение программирования, как только дети научатся читать.
- Редактор текстов как конструктор дает возможность на подсознательном уровне превратить «учебу» в «не учебу», а, кроме того, сократить количество ошибок в программе.
- Ориентированность на графику, так как доказана эффективность обучения с опорой на наглядно-образное мышление.
- Объектная ориентированность позволяет изучить основные способы создания программ с объектами.

Работать над проектом было несложно, увлекательно, все свои идеи удалось воплотить в этой компьютерной игре. Мои одноклассники постоянно интересовались: какой уровень следующий, кто еще из главных героев появился? Соревновались на время, у кого быстрее получится пройти игру. Они мне оказали большую поддержку.

Компьютерная игра «Cats Man» была отправлена на областной фестиваль-конкурс по программированию на Скретч в качестве домашнего задания. По итогам работа получила высокую оценку жюри, что позволило команде МБОУ СОШ с. Р. Камешкир занять 2 место (см. Приложение № 7).

Данный проект позволил не только повысить свой уровень в программировании, но и решать задачи на стратегию и прогнозирование. Я считаю, что любой ученик, выполняя проекты в Scratch, больше изучает, развивается разносторонне, становится стратегом.

Перечисленные особенности Scratch оказывают влияние на развитие таких личностных качеств: ответственность и адаптивность; творчество и любознательность; критическое и системное мышление; умения работать с информацией и медиасредствами; межличностное взаимодействие и сотрудничество; умения ставить и решать проблемы; направленность на саморазвитие.

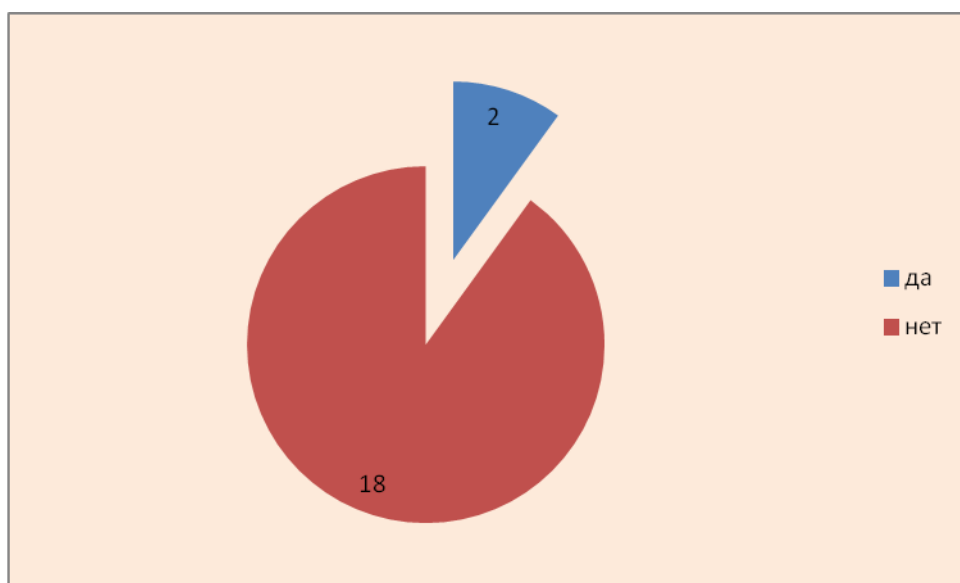
Поставленные задачи в работе выполнены: я изучил основные возможности и характеристики среды программирования Scratch и на основе этого разработал свой проект «Cats Man».

Литература

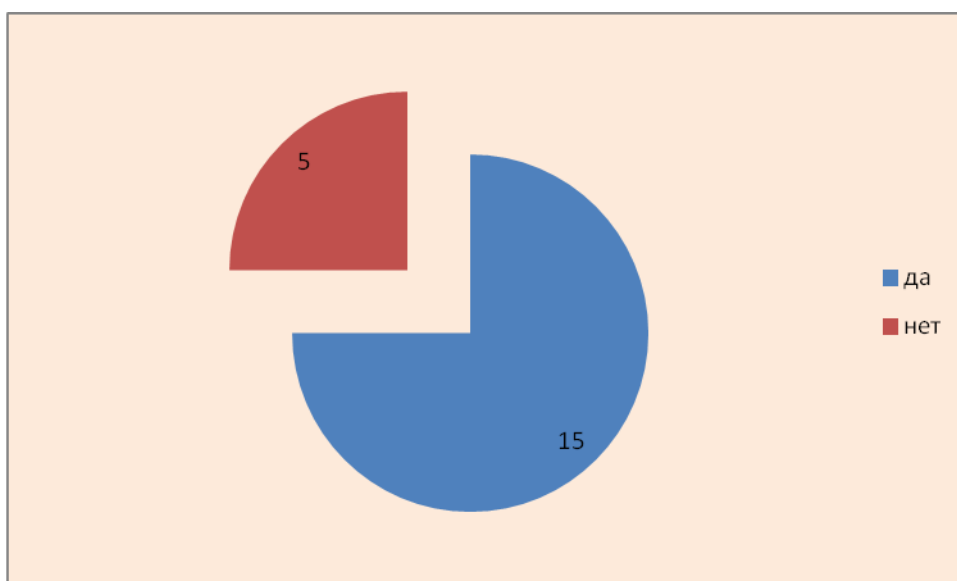
1. Патаракин Е. Руководство для пользователя среды Scratch // http://www.supercode.ru/download/scratch_by_patarakin.pdf
2. Школа Scratch [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: http://letopisi.ru/index.php/Школа_Scratch
3. Scratch | Home | imagine, program, share [сайт]. URL: <http://scratch.mit.edu>
4. Scratch | Галерея | Gymnasium №3 [сайт]. URL: <http://scratch.mit.edu/galleries/view/54042>
5. <http://ru.wikipedia.org/wiki/>
6. <http://scratch.mit.edu/>

Приложение 1. Анкетирование одноклассников

1. Умеете ли вы создавать компьютерные игры?



2. Хотели бы вы научиться создавать игры с помощью компьютера?



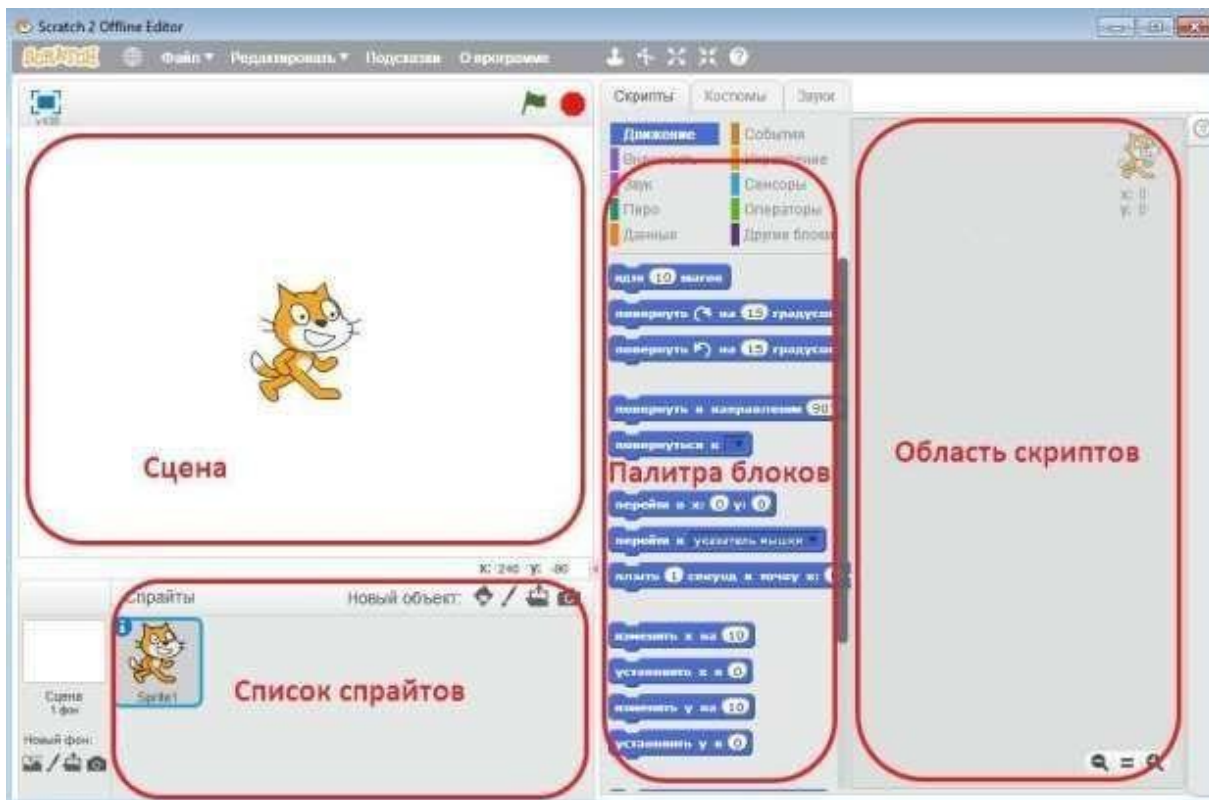
Приложение 2. Среда программирования Scratch 2.0

Новый графический редактор включает векторную графику и теперь можно изменять размер картинки без ухудшения качества. Войдя в систему, можно открыть свой рюкзак внутри любого проекта (он находится внизу экрана). Рюкзак позволяет копировать и перемещать спрайты, костюмы, фоны и скрипты из одного проекта в другой.

Можно создавать свои собственные блоки программы.

Более детальная функция масштабирования для поля со скриптами и графического редактора даёт удобство использования. В **Scratch** обычном мы могли только лишь увеличить/уменьшить размер поля, в котором собирается скрипт.

Инструменты редактирования в звуковом редакторе дают возможность работать с фрагментами записанных звуковых файлов, а благодаря эффектам, теперь можно усиливать, ослаблять звук или вообще переворачивать. И еще много других новых и полезных функций



Интерфейс программы Scratch 2.0

Приложение 3. Компьютерные игры

В настоящий момент существует несколько типов компьютерных игр, которые могут быть классифицированы по нескольким признакам. **По жанрам:**

Шутер - в играх данного типа игрок, как правило, действуя в одиночку, должен уничтожать врагов при помощи холодного и огнестрельного оружия

Файтинг - поединки двух и более противников с применением рукопашного боя.

Аркада - игры, в которых игроку приходится действовать быстро, полагаясь в первую очередь на свои рефлексы и реакцию.

Симулятор - основным принципом симулятора является точное воспроизведение особенностей какой-то тематической области

Стратегия - игра, требующая выработки стратегии, например, для победы в военной операции.

Квест - игра-повествование, в которой управляемый игроком герой продвигается по сюжету и взаимодействует с игровым миром посредством применения предметов, общения с другими персонажами и решения логических задач.

Ролевые игры - у главного героя (героев) и других персонажей и врагов (чаще в меньшей степени) присутствует некоторое количество параметров (умений, характеристик, навыков) которые определяют их силу и способности.

Головоломки и настольные - компьютерная реализация настольных игр, например шахматы, карты, шашки, «Монополия» и другие.

Развивающие игры - это компьютерные программы т. н. “открытого” типа, предназначенные для формирования и развития у детей общих умственных способностей, эмоционального и нравственного развития, целеобразования, способности соотносить свои действия по управлению игрой с создающимися изображениями на экране. Они развивают фантазию, воображение. В них нет явно заданной цели — они являются инструментами для творчества, самовыражения ребенка.

Обучающие игры. К ним относятся игровые программы дидактического (“закрытого”) типа, в которых в игровой форме предлагается решить одну или несколько дидактических задач.

Приложение 4. Группа «Звук»

Блоки	Пояснения
	Начинает играть выбранный из меню звук и сразу переходит к следующему блоку, звук продолжается
	Воспроизводит звук и переходит к следующему блоку, когда звук закончен
	Прекращает все звуки
	Ударные играют указанное число тактов
	Пауза в течение заданного числа тактов
	Определенная нота воспроизводится указанное количество времени (в секундах). Ноты записаны в цифрах, но против каждой цифры стоит ее звучание.
	Выбирает тип инструмента, который использует спрайт в блоке играй ноту . (Каждый спрайт имеет свой инструмент)
	Изменяет громкость звучания на заданную величину
	Задаёт громкость звучания с помощью заданного значения
	Возвращает значение громкости звука
	Изменяет темп звучания на заданную величину
	Задаёт темп игры (количество тактов в минуту)
	Возвращает значение темпа игры (количество тактов в минуту)

Приложение 5. Группа «Переменные»

Блоки	Пояснения
 	<p>Создает переменную, задается ее имя и область действия (глобальная или локальная). С первой переменной появляются соответствующие блоки.</p>
 	<p>Удаляет переменную</p>
	<p>Имя переменной: ссылается на значение переменной. Если флажок включен, то на сцене отображается монитор со значением переменной</p>
 	<p>Присваивает переменной заданное значение, которое может быть постоянной, переменной или значением выражения. Оператор присваивания в других языках программирования переменная = выражение</p>
 	<p>Изменяет значение выбранной из меню переменной на заданную величину переменная = переменная + выражение</p>
 	<p>Отображает на сцене монитор со значением переменной</p>
 	<p>Прячет (делает невидимым) на сцене монитор со значением переменной</p>
 	<p>Создает список, задается его имя и область действия (глобальный или локальный). С первым списком появляются соответствующие блоки</p>
 	<p>Удаляет список</p>
	<p>Имя списка. Ссылается на весь список. Если флажок включен, то список отображается на сцене</p>
 	<p>Добавляет заданный элемент в конец списка. Элемент может быть числом, любым текстом или задан с помощью выражения</p>

	<p>Удаляет один или все элементы из списка. Можно задать порядковый номер удаляемого элемента (число или переменная) или выбрать подходящий вариант в раскрывающемся меню. Вариант "последний" (last) удаляет последний элемент, вариант "все" (all) удаляет все элементы.</p>
	<p>Добавляет элемент в заданную позицию (последний, любой из, порядковый номер) в выбранный список. Длина списка увеличивается на 1. Вариант "последний" (last) добавляет элемент в конец списка, вариант "любой" (any) добавляет элемент в произвольную позицию.</p>
	<p>Заменяет элемент в заданной позиции (последний, любой из, порядковый номер) в выбранном списке. Вариант "последний" (last) заменяет элемент в конце списка, вариант "любой" (any) заменяет элемент в произвольной позиции. Длина списка не меняется.</p>
	<p>Ссылается на элемент списка с заданным порядковым номером или с соответствующим вариантом раскрывающегося меню. Может помещаться в другой блок</p>
	<p>Показывает количество элементов в списке</p>
	<p>Получает значение Истина, если данное значение содержится в списке. NB! Различаются большие и маленькие буквы</p>

Приложение 6. Группа «Операторы»

Блоки	Пояснения
	Операция сложения двух чисел
	Вычитание из первого числа второго
	Умножение двух чисел
	Деление первого числа на второе
	Возвращает случайное число в заданном интервале
	Возвращает значение Истина , если первое число меньше второго
	Возвращает значение Истина , если первое число равно второму
	Возвращает значение Истина , если первое число больше второго
	Возвращает значение Истина , если оба условия истинны
	Возвращает значение Истина , если хотя бы одно условие истинно
	Возвращает значение Истина , если значение условия ложно и наоборот
	Объединяет два текста. Могут быть выражениями
	Возвращает длину данного текста. Текст может быть задан выражением
	Возвращает букву (символ) в заданной позиции. Может быть выражением
	Возвращает найденное значение функции (abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e^, 10^)
	Возвращает остаток от деления первого числа на второе
	Возвращает ближайшее целое число

Государственное автономное образовательное учреждение
дополнительного профессионального образования
ИНСТИТУТ РЕГИОНАЛЬНОГО РАЗВИТИЯ ПЕНЗЕНСКОЙ ОБЛАСТИ

ДИПЛОМ

НАГРАЖДАЕТСЯ КОМАНДА

Хакеры

МБОУ СОШ с.Р. Камешкир

занявшая 2 место в региональном
фестивале-конкурсе по программированию
в визуальной среде программирования
“Scratch” для обучающихся 5–9 классов.

Ректор

О.Ф. Федосеева



23 декабря 2019 г.