

Краснодарский край Динской район Муниципальное Автономное
Общеобразовательное Учреждение Муниципального Образования
Динской район "Средняя Общеобразовательная Школа № 35 Имени
46-го гвардейского орденов красного знамени и Суворова 3-й
степени ночного бомбардировочного авиационного полка"

СБОРНИК ЗАНЯТИЙ ПО ПРОГРАММИРОВАНИЮ В СРЕДЕ SCRATCH

методическое пособие



Автор:
Пак Екатерина Васильевна
Педагог дополнительного
образования
МАОУ МО Динской район СОШ
№35

2024г.

СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИМЕНЕНИЯ SCRATCH	4
1.1. Анализ программного обеспечения для изучения программирования в школьном курсе	4 6
1.2. Целесообразность использования Scratch в школьном курсе	8
1.3. Описание среды программирования Scratch	
ГЛАВА 2. РАЗРАБОТКИ ЗАНЯТИЙ ПО ОСВОЕНИЮ НАВЫКОВ ПРОГРАММИРОВАНИЯ С СРЕДЕ SCRATCH	10
2.1. Занятие №1 Работа с командами на центральном поле	10
2.2. Занятие №2 Путешествие в космос	12
2.3. Занятие №3 Работа с циклами	17
2.4. Занятие №4 Ветвление. Команда "если". Блок ОПЕРАТОРЫ	23
2.5. Занятие №5 Кошки-мышки	26
2.6. Занятие №6 Стреляем по шарикам	27
2.7. Занятие №7 Движение фона	28
2.8. Занятие №8 Управление с помощью клавиатуры	29
2.9. Занятие №9 Взаимодействие графических объектов	31
ЛИТЕРАТУРА	36

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

По всему миру десятки миллионов учеников активно участвуют в различных мероприятиях, направленных на получение качественных знаний, которые помогут в будущем получить высокооплачиваемую работу в IT - среде. Они ходят на кружки, факультативы по информатике, посещают различные курсы, изучают самостоятельно, смотрят ролики в YouTube с одной целью – получить дополнительные знания в IT сфере. Это модно, престижно, увлекательно, очень развивающее, и это позволит в будущем мире высоких технологий быть всегда в тренде. Даже если ученики в будущей профессиональной деятельности не планируют работать в сфере IT, они все равно развиваются и приобщаются к миру технологий, к тому же программирование развивает аналитическое и критическое мышление и память.

Возникает проблема: как правильно внедрять программирование Scratch в курс дополнительного образования, какой методикой воспользоваться, чтобы его применение было эффективным. Данная проблема является следствием того, что в настоящее время вопрос о применении современных технологий и мультимедиа в школах приобретает всё большую значимость, что, в свою очередь, подтверждает актуальность методического пособия.

Цель данного методического пособия: оказать помощь учащимся и педагогом дополнительного образования по введению в среду программирование Scratch.

В методическом пособии представлены теоретические основы применения Scratch, даны рекомендации по использованию Scratch и технические требования по установке и описанию среды программирования Scratch.

Так же пособие содержит серию конспектов занятий и пошаговые инструкции проектирования игр по Scratch-программированию. Материалы к занятиям снабжены достаточным количеством иллюстраций и пояснений к ним, дающих возможность обучать детей технологии разработки игровых программ.

Во время занятия происходит частая смена деятельности: восприятие материала на экране и на слух, участие в обсуждении поставленной задачи, работа с опорными конспектами, работа на компьютере, защита проекта.

Каждый ребенок, участвующий в работе по выполнению задания, имеет возможность не только выполнить поставленную задачу, но и творчески доработать свой мини-проект, предложить свой план выполнения задания.

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

Глава 1. Теоретические основы применения Scratch в школьном курсе

на уроках информатики и ИКТ

1.1. Анализ программного обеспечения для изучения программирования в школьном курсе

Все языки программирования можно разделить на два вида: текстовые и визуальные. Текстовые языки программирования - это языки, введённые с клавиатуры, выводятся в виде текстовых файлов. Графические или визуальные языки программирования - это языки, которые основаны на перетаскивании, а не вводе и знании команд. Команды обычно выглядят как ярлыки, кнопки, картинки, имеющие текстовые метки. При обучении детей младшего возраста целесообразней начать изучать визуальную среду программирования, а затем переходить к текстовым.

Традиционные языки программирования, такие как Pascal, Java, C++, Delphi и другие, сложны для изучения детьми. Scratch разработан для того, чтобы это изменить. При изучении алгоритмизации и программирования недостаточно показать и объяснить работу различных условий, циклов, переменных. Необходимо научить школьников мыслить критически, понимать суть алгоритмов. Ученики должны визуально-наглядно видеть результаты своих действий и Scratch, как одна из визуальных сред программирования решает эти проблемы. Scratch - это инструмент для творчества, где обучающийся сам не замечает, как учится программированию. [1]

Целесообразно сравнить Scratch с другими проектами на рынке программного обеспечения. Традиционные языки программирования рассматривать нет необходимости, вследствие этому написано выше. Рассмотрим основные особенности программирования на Scratch:

1. Блочное программирование. Для программирования в Scratch, необходимо совмещать графические блоки вместе, подобно Лего конструированию. Блоки сделаны так, чтобы их можно было собрать только в синтаксически верных конструкциях, это исключает ошибки. Различные типы данных имеют разные формы, подчеркивая несовместимость. Можно сделать изменения в стеках, даже когда программа запущена, что позволяет больше экспериментировать с новыми идеями.

2. Манипуляции данными. Со Scratch вы можете создать программы, которые управляют и смешивают графику, анимацию, музыку и звуки. Scratch расширяет возможности управления визуальными данными, которые популярны в современной культуре.

3. Совместная работа и обмен. Сайт проекта Scratch предлагает вдохновение и аудиторию: вы можете посмотреть проекты других людей, использовать и изменить их картинки и скрипты, и добавить ваш собственный проект. Самое большое достижение — это общая среда и культура, созданная вокруг самого проекта.

Аналогичная среда программирования с интересным названием Blockly (Блокли) написанная на языке программирования JavaScript поддерживает

единственную платформу для работы - браузер. Блокли включает в себя графический редактор, позволяющий составлять программы, как и в Scratch из блоков, визуально обе среды программирования схожи между собой. Функционал и возможности расширения и подключения дополнительных блоков шире в Scratch.

Рис. 1. Интерфейс Blockly



Kodu первоначально названный Voku, разработчиком которой выступила компания Microsoft еще в 2009 году. Kodu- это игра, в которой можно создавать игру. Преимущество этой визуально событийно ориентированной среды программирования в том, что она работает в 3D-моделями в 3D-мире. Kodu работает на платформе Windows XP/Vista/7/8/10 и Xbox 360. Так же как и в Scratch учащиеся научатся не только программированию, но и сотрудничеству, логике и творчеству.



Рис. 2. Интерфейс Kodu

Подводя итоги, стоит отметить, что каждая из перечисленных сред программирования имеет место быть, но не все возможно реализовать в рамках учебного процесса и российских школ. Так, например Kodu требует более технические требования от компьютера, обязательное наличие

хорошего графического процессора в системном блоке. В школах, как правило, закупаются бюджетный класс компьютеров, тем самым ставя задачу перед лаборантом или учителем задачу установки на персональные компьютеры невозможным или частично невозможным.

Работа в сети Интернет актуальна как никогда, но наличие только вебверсии, как было рассмотрено выше в среде программирования Blockly сильно ограничивает, так как не все школы оборудованы сетью Интернет.

Наглядное сравнение программного обеспечения изображено в табл. 1., где видно, что Scratch наиболее актуален.

Таблица 1.

Сравнения программного обеспечения для изучения программирования

	Бесплатно	Низкие технические требования	Онлайн/Оф лайн	Модули расширения	Общая среда. Культура	Привлекат ельный дизайн
Scratch	+	+	+	+	+	+
Blockly	+	+	-	-	-	-
Kodu	+	-	-	-	-	+

Таким образом, анализируя среды программирования, не сложно заметить, что все они (среды программирования) на уроках информатики и ИКТ способствуют развитию у учащихся рабочей памяти, внимания, логики, математических знаний, мышления, развитию самостоятельности. А используя в обучении эффективные средства программирования, такие как Scratch, которые повышают интерес и мотивацию учеников, постигают всю технологию решения задач - от постановки проблемы до представления результата.

1.2. Целесообразность использования Scratch в школьном курсе

С помощью Scratch можно программировать собственные интерактивные истории, игры и мультфильмы, а затем делиться своими творениями с другими участниками как офлайн, так и онлайн сообщества, Scratch доступен каждому. Разработчики - группа ученых Массачусетского технологического института (MIT) Lifelong Kindergarten предоставляют его бесплатно. Глобальная задача учёных - стать частью образовательной программы.

Scratch помогает молодым людям учиться мыслить творчески, рассуждать системно и работать совместно - это необходимые навыки для жизни в 21 веке [2]. Поэтому, на уроках информатики использовать Scratch нужно, делая при этом акцент на следующие цели:

1. Формирование умений и навыков программирования.

2. Развитие коммуникативных способностей.
3. Подготовка личности "информационного общества" 21 века.
4. Обеспечение ребёнка максимально возможным количеством учебного материала, которое он может усвоить.
5. Формирование умения принимать оптимальные решения.

Способ достижения дидактической цели в программировании на Scratch происходит с помощью детальной разработки (технологии), которая должна завершиться практическим результатом, тем или иным образом оформленным. Получившийся результат можно увидеть, осмыслить, применить в реальной практической деятельности. Чтобы добиться такого результата, необходимо научить детей самостоятельно мыслить, привлекая для этой цели знания из разных областей (математики, физики, технологии, изобразительного искусства и др.), способность прогнозировать результаты и возможные последствия разных вариантов решения, умения устанавливать причинно следственные связи.

Возможность организовать работу методом проектов, которая сориентирована на самостоятельную деятельность учащихся - индивидуальную, парную групповую, которую учащиеся выполняют в течение определённого отрезка времени. Внедрение и изучение Scratch на уроках информатики и ИКТ органично сочетается с групповым подходом к обучению [3].

Также программирование на Scratch предполагает решение какой-то проблемы. Решение многих задач на уроках информатики трудно назвать проблемными. Учащиеся, выполняя задания на уроках информатики, решают не проблему, а выполняют определённые алгоритмы действий .

В сети интернет на сегодня имеется не очень много учебников и учебных руководств по Scratch, при чем, большая часть из них на английском языке, потеряли свою актуальность из-за глобальных изменений при обновлений Scratch или не адаптированы под школьную программу. Нужно сказать, что Scratch находит все большее распространение в системах образования разных стран мира. Причин тому несколько, и корни их лежат не только в сфере собственно педагогики, но, главным образом, в сфере социальной, так как возникает необходимость не столько передавать ученикам сумму тех или иных знаний, сколько научить приобретать эти знания самостоятельно, уметь пользоваться приобретёнными знаниями для решения новых творческих, познавательных и практических задач [4].

Ученик, создавая проекты в среде программирования Scratch осваивает множество навыков 21 века, полезные для его развития:

1. Творческое мышление;
2. Проектирование;
3. Постоянное обучение;
4. Системный анализ;
5. Эффективное взаимодействие;
6. Критическое мышление.

Педагогический потенциал среды программирования Scratch позволяет рассматривать её как перспективный инструмент организации междисциплинарной проектной научно-познавательной деятельности обучающегося, направленной на личностное и творческое развитие ребёнка. Именно междисциплинарность позволит школьникам создать единую картину мира, проводя взаимосвязи с различными науками [26].

1.3. Описание среды программирования Scratch

Scratch- это визуальный язык программирования, созданный специально для детей, в котором программа складывается из цветных блоков. С помощью этого языка ребята понимают, что такое программирование и создают свои первые программы.

Scratch был разработан специально для того, чтобы в простой и наглядной форме дети

учились анимации, созданию игр, музыки, интерактивных изображений. Главное его преимущество – программы не пишутся кодом, а составляются из разноцветных блоков. Каждый блок - это функция, к которой добавляются её параметры.

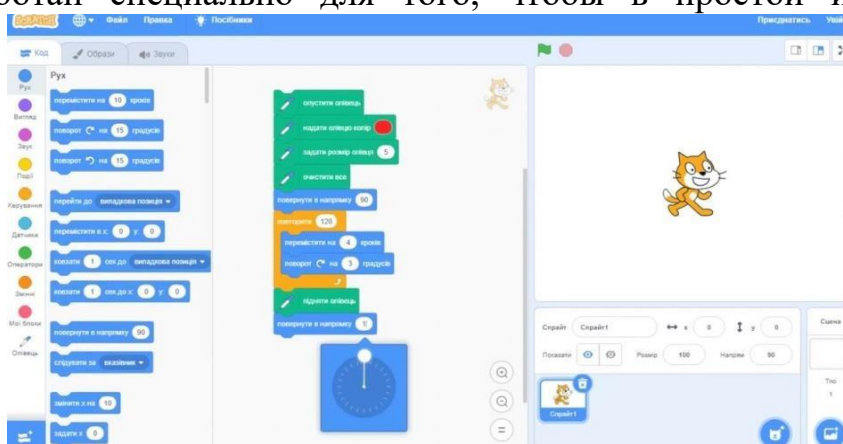
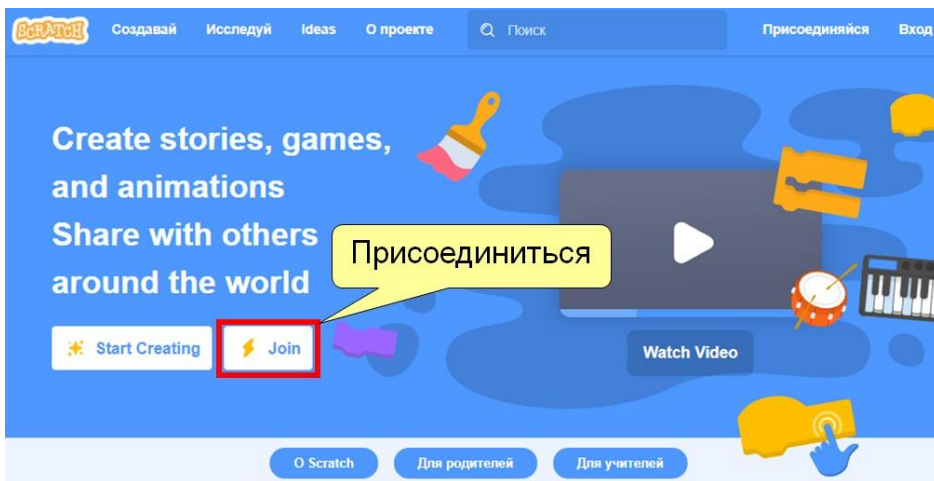


Рисунок 3. Блоки Scratch

Цель программы «Лаборатория компьютерных игр» - научиться программированию через создание творческих проектов в среде программирования для детей Scratch.

У языка Scratch есть свой бесплатный редактор. Его можно установить на свой компьютер или работать сразу в интернет-браузере.

Среда программирования Scratch <https://scratch.mit.edu/>



Регистрация в системе

Для удобства обучения Scratch и сохранения проектов лучше зарегистрироваться на сайте <https://scratch.mit.edu/>.

Придумайте и запомните логин и пароль, под этими данными вы сможете заходить на сайт Scratch с любого компьютера, где есть доступ к сети Интернет – из дома, на наших занятиях, с планшета, телефона и т.д.

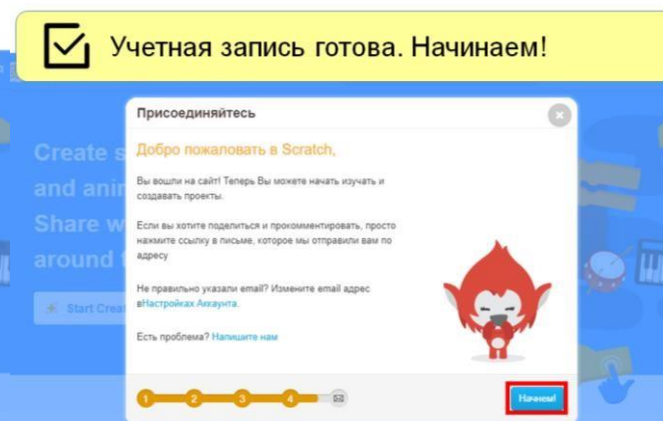
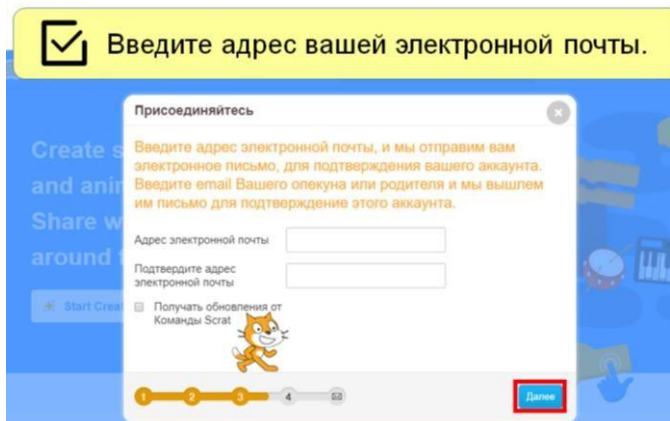
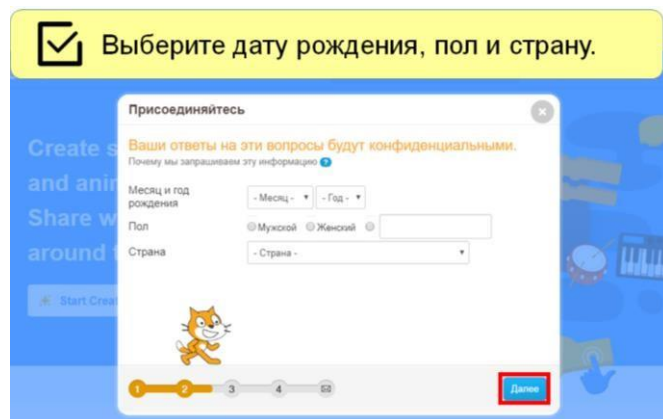
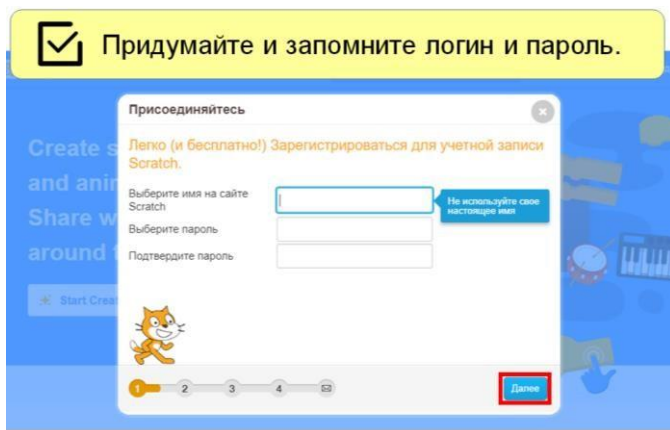
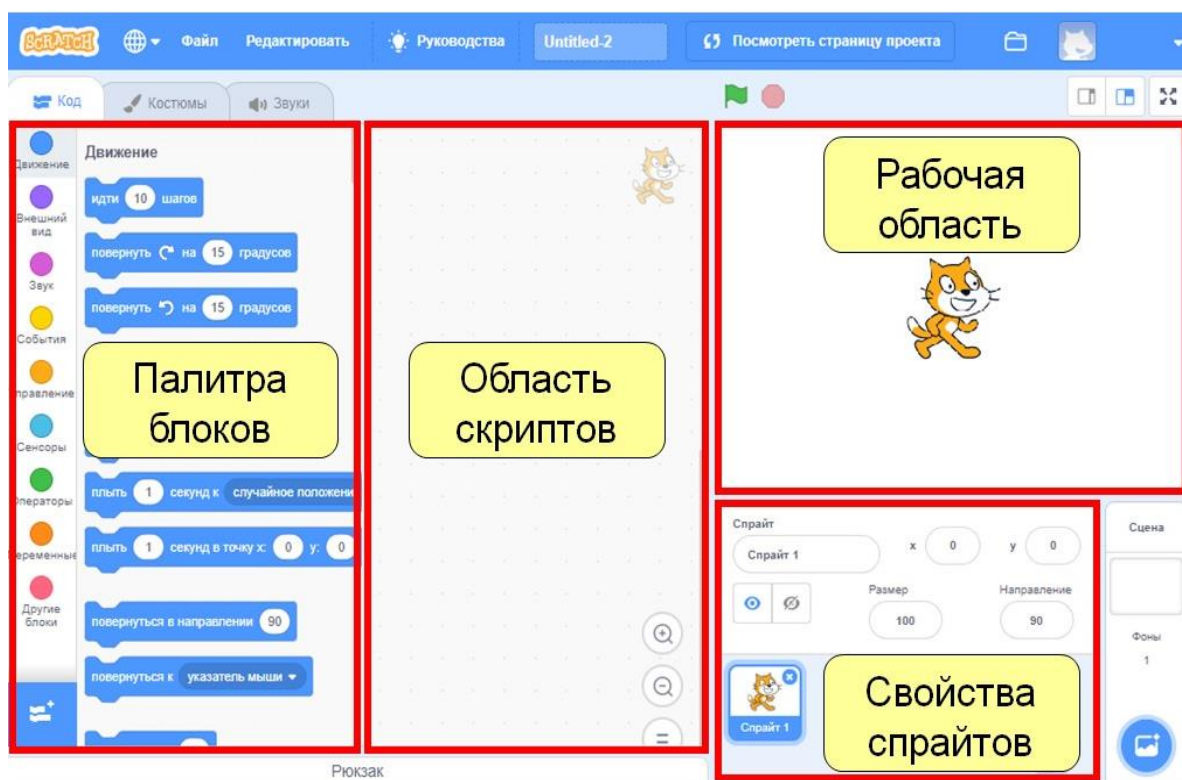


Рисунок 4 установка программы Scratch

Рисунок 5. Элементы редактора Scratch



ГЛАВА 2. РАЗРАБОТКИ ЗАНЯТИЙ ПО ОСВОЕНИЮ НАВЫКОВ ПРОГРАММИРОВАНИЯ С СРЕДЕ SCRATCH

2.1. Занятия №1 Работа с командами на центральном поле

Ход работы:



- 1) Запускаем Scratch с рабочего стола
- 2) Для каждого нового объекта по умолчанию активен блок ДВИЖЕНИЕ (высвечивается синим цветом вся его кнопка, и загружены команды этого блока).
- 3) Зажмем левой кнопкой мыши (ЛКМ) первую команду "идти 10 шагов" и перетащим ее на белое пустое поле по центру, как указано в рисунке 19.

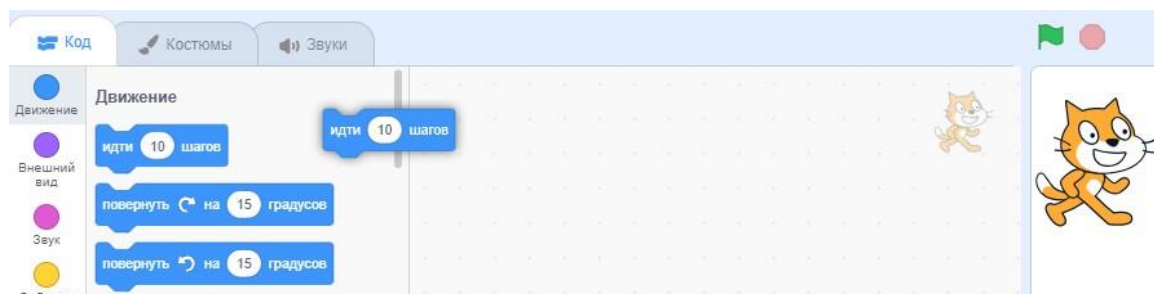
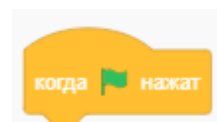


Рисунок 6. Вставка команд

- 3) Отпустим ЛКМ.
- 4) Все! Элементарная программа готова.
- 5) Запустим ее двойным нажатием на ЛКМ по команде, которую мы перетащили в центр – Кот продвинулся вперед к правому краю экрана.
- 6) Рассмотренным способом запуска программ невозможно запустить

скрипты сразу нескольких объектов или запустить просмотр действия даже этой одной команды в режиме презентации.



7) Поэтому начинают скрипт стартовой командой: блока СОБЫТИЯ.

8) Наждем ЛКМ по блоку СОБЫТИЯ. Из области его команд перетащим в центральную область эту команду, присоединив к верхней части команды «иди 10 шагов»:

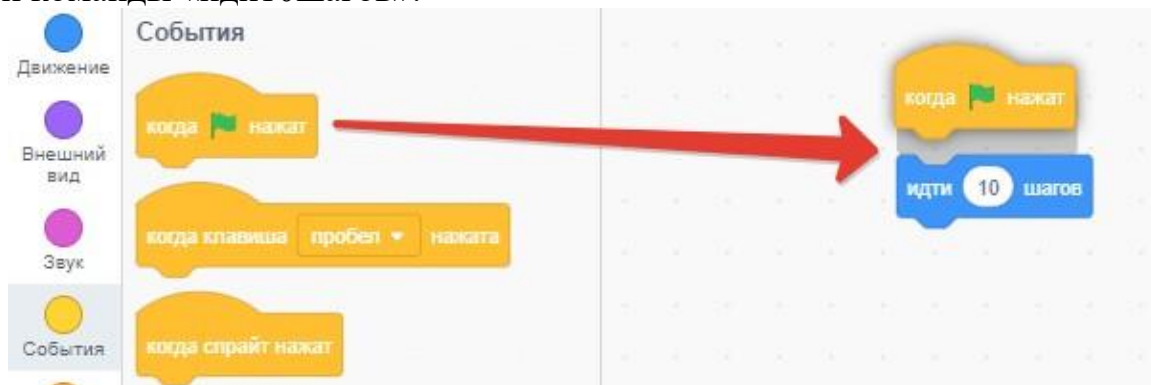


Рисунок 7. Стартовая команда

9) Щёлкнем ЛКМ по блоку ВНЕШНИЙ ВИД и из области его команд перетащим в центральное поле команду «сказать: 'Привет!'» (можно этот текст заменить на любой другой), присоединив к нижней части команды «иди 10 шагов»:

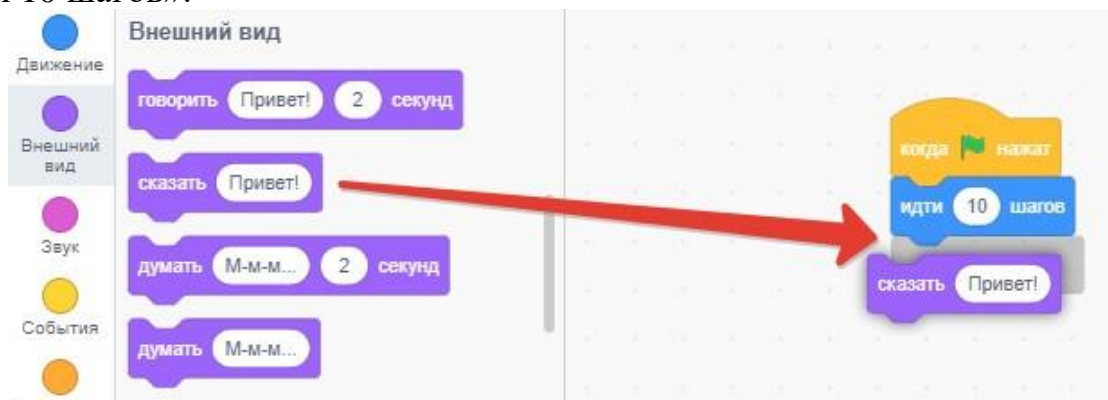


Рисунок 8. Команда говорить

При перемещении команды в область скриптов появляется серая полоса, подсказывающая, куда прилепится команда, если ее отпустить.

10) Также присоединим команду «изменить эффект...»:

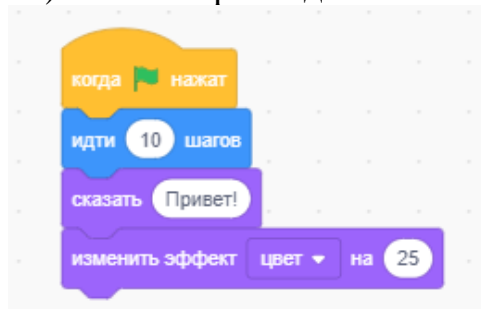


Рисунок 9. Команда изменения цвета

11) Чтобы перетащить не одну, а сразу несколько команд, «хватаемся» за верхнюю команду этой группы и перетаскиваем (например, «разрываем» скрипт).

12) Нажав ПКМ поверхней части этой группы команд выдает интерактивное меню: дублировать, добавить комментарий удалить блок:

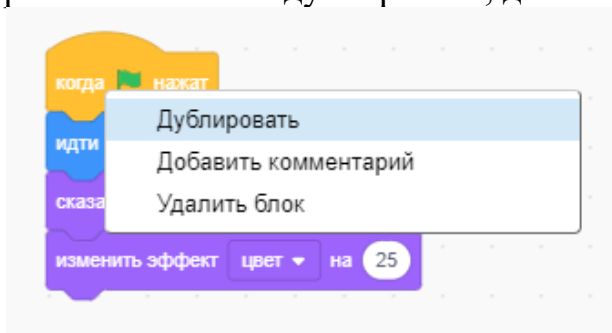


Рисунок 10. Свойства по нажатию ПКМ

13) Запускаем нашу программу и проверяем что получилось



Рисунок 11. Запуск проекта

14) Кот передвинулся на 10 шагов по оси "x", сказал "Привет!" и поменял свой цвет на зелёный.

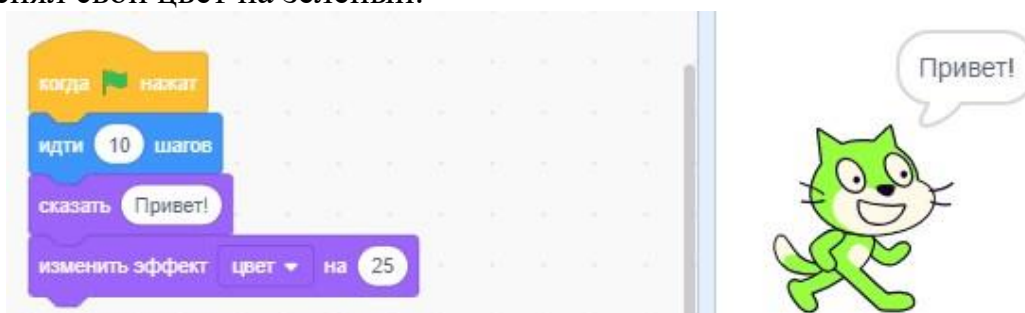


Рисунок 12. Результат выполнения команд

Задания для самостоятельной работы:

- 1) Измените количество шагов так, чтобы Кот передвигался от одного края экрана до другого.
- 2) Найдите команду говорить "Привет" 2 секунды и посмотрите что произойдет, в чем отличие от команды сказать "Привет!"?
- 3) Подумайте, что нужно сделать, чтобы изменить цвет кота на синий?
- 4) Покажите проделанную работу своему учителю!

2.2. Занятие №2 Путешествие в космос

Тема: анимация объектов в среде программирования Scratch.

Цель: научиться анимировать объекты в среде программирования Scratch.

Оборудование: ПК с установленными ОС и Scratch, (данная) инструкция.

Структура урока:

1. Организационный момент.
2. Актуализация опорных знаний.
3. Изучение нового материала.
4. Инструктаж по ТБ.
5. Практическая работа.
6. Подведение итогов урока.
7. Домашнее задание.

Ход урока

1. Инструктаж по ТБ

2. Закрепление изученного материала

Примечание. После выполнения каждого из практических задач обучающийся должен сообщить учителю о своей готовности показать результаты своей работы.

3. Практическая работа проект «Путешествие в космос».

Шаг1. Подготовка

1. Создайте новый проект (меню Файл-Новый);
2. Удалите существующий спрайт (клик правой кнопкой мышки, удалить);
3. Добавьте новое тело (откройте библиотеку, изображение «Stars»);

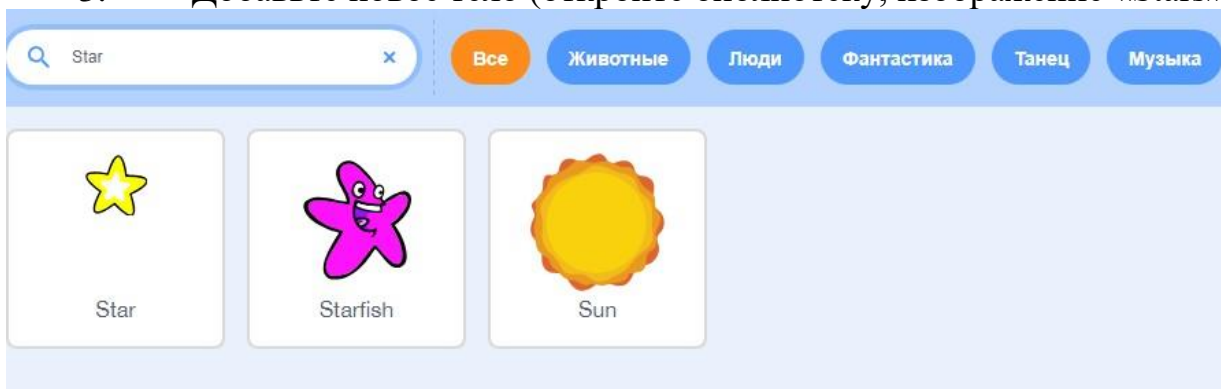
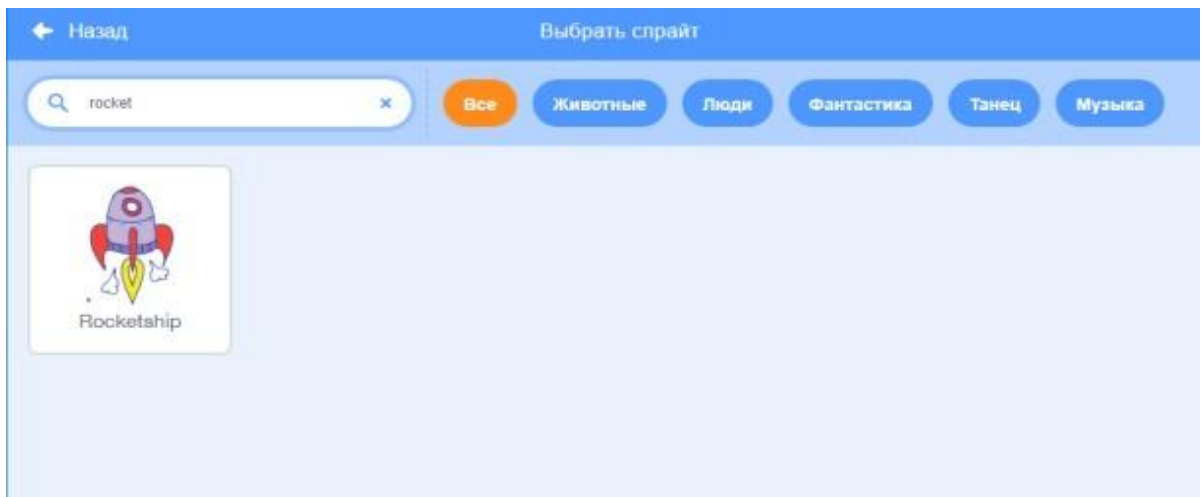


Рисунок 13. Выбор рисунка

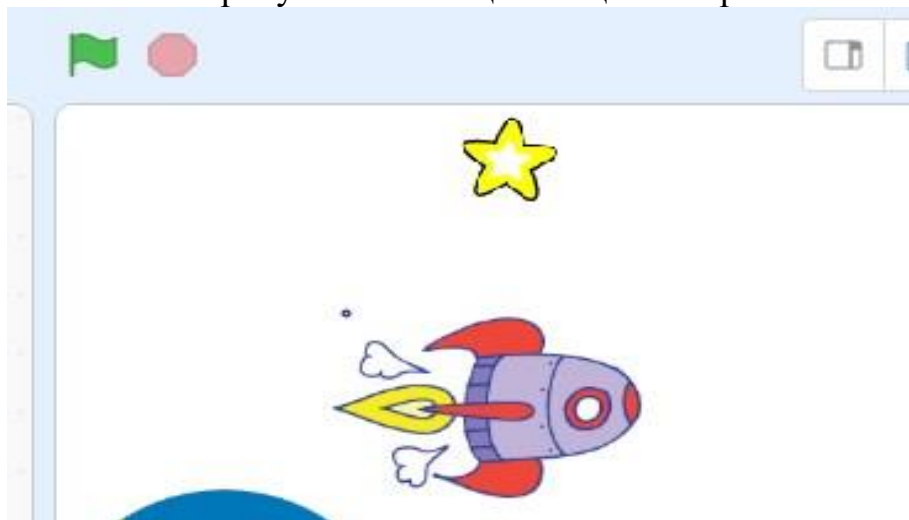
4. Удалите начальную сцену (клик правой кнопкой мыши, Удалить)

Шаг 2. Создание звездолета

1. Добавьте новый спрайт звездолета (категория «Транспорт», спрайт «Rocketship»);



2. Перейдите на вкладку образы, выберите рисунок ракеты и поверните его так как показано на рисунке с помощью опции Направление.



3. Создайте скрипт, чтобы ракета находилась в левом нижнем углу и двигалась сначала к звезде, а потом в центр экрана со звуком полета.

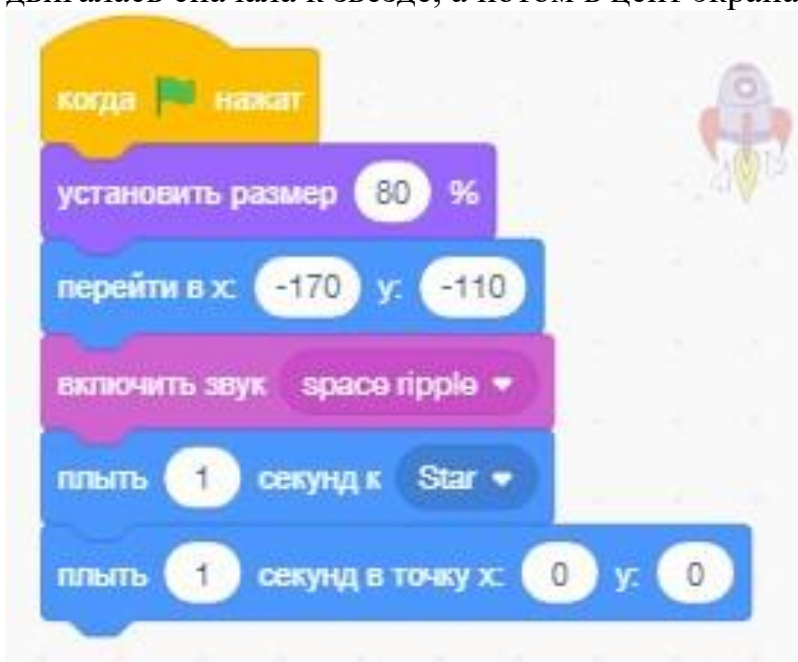


Рисунок 14. Создание скрипта

Шаг 3. Создание Земли

1. Добавьте новый спрайт (категория «Вещи» спрайт Земля «Earth»).



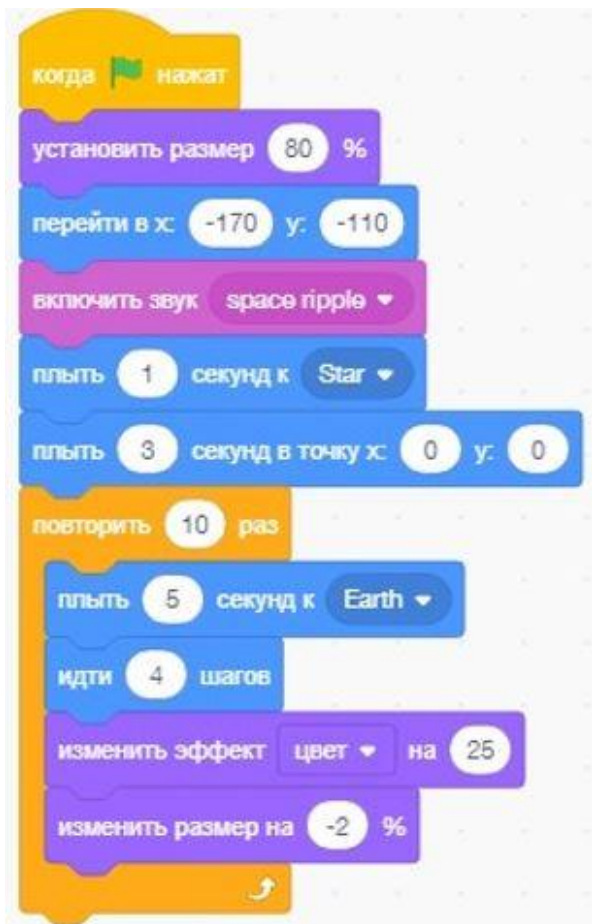
2. Создайте скрипт Земля, который перемещает спрайт в правый верхний угол сцены и устанавливает ее размер в 150%.

Шаг 4. Ракета летит на Землю

1. Добавьте команды, чтобы звездолет полетел на Землю.



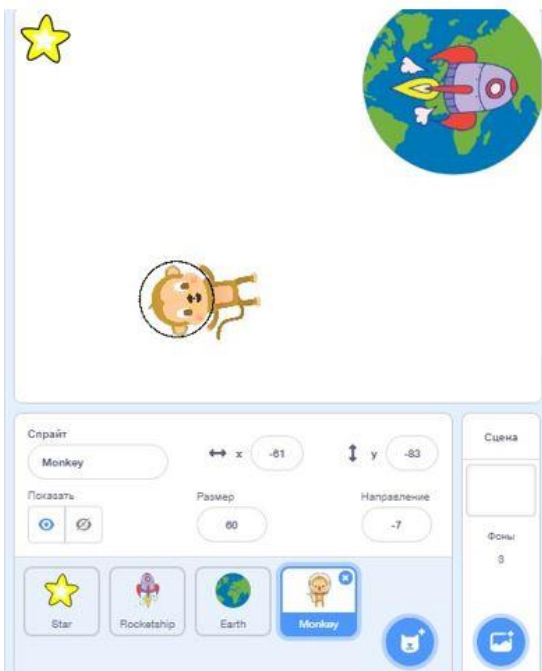
2. Добавьте анимацию уменьшения ракеты с расстоянием и изменением его цвета.



Шаг 5.

Обезьянка-космонавт, потерялась в космосе

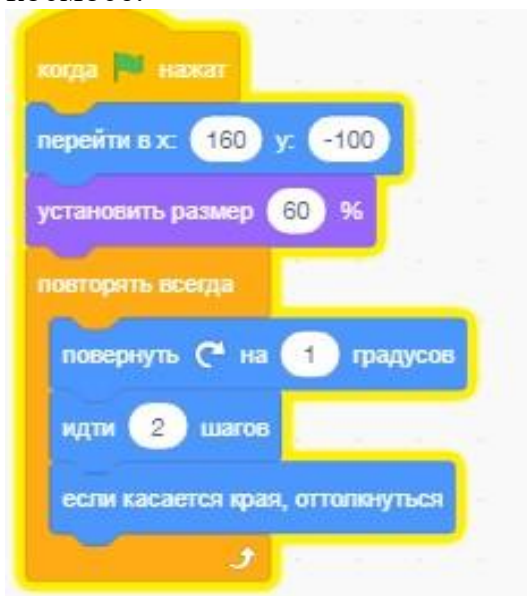
1. Добавьте новый спрайт обезьянка (категория «Животные», «Monkey1»).



2. С ПОМОЩЬЮ

нарисуйте ему шлем.

3. Создайте скрипт, который будет показывать движение нашего космонавта в космосе.



Домашнее задание.

Создайте спрайт летучей мыши «Bat» и движение с уменьшением, как она улетает к звезде, можно добавить изменение цвета.

2.3. Занятие №3 Работа с циклами

Цель: Познакомиться с действием работы циклов блока УПРАВЛЕНИЕ; рассмотреть команды блока СЕНСОРЫ.

Как мы могли убедиться, что программа выполняется всего один раз, а как сделать так, чтобы наш Кот ходил чуть дольше и дальше, как нам захочется. В этом нам поможет команда запускающая цикл.

Цикл - запускает кругооборот одних и тех же действий. В среде Scratch циклы находится во вкладке УПРАВЛЕНИЕ и они бывают разных типов: повторить 10 раз, повторять всегда, повторять пока не....

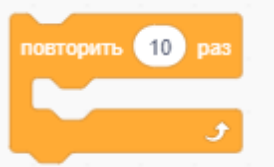


Рисунок 15. Выполнить команду 10 раз

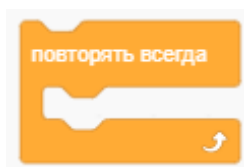


Рисунок 16. Выполняет команду бесконечно

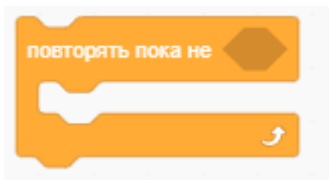


Рисунок 17. Будет выполнять команду пока не случится что-то

Ход работы:

1) Воссоздайте работу из предыдущей работы:

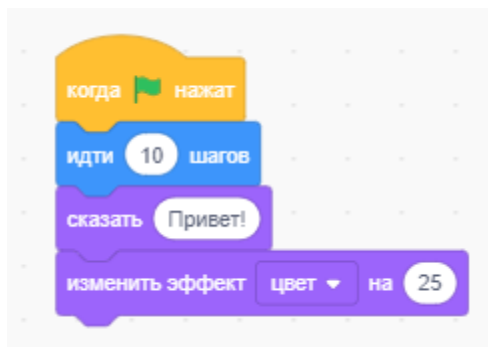


Рисунок18. Пример из работы.№1

2) Перетащите ЛКМ команду повторить 10 раз таким образом, чтобы все наши предыдущие команды, за исключением запуска проекта, оказались внутри цикла. Следите за серой областью, которая показывает, куда подставиться новый блок:

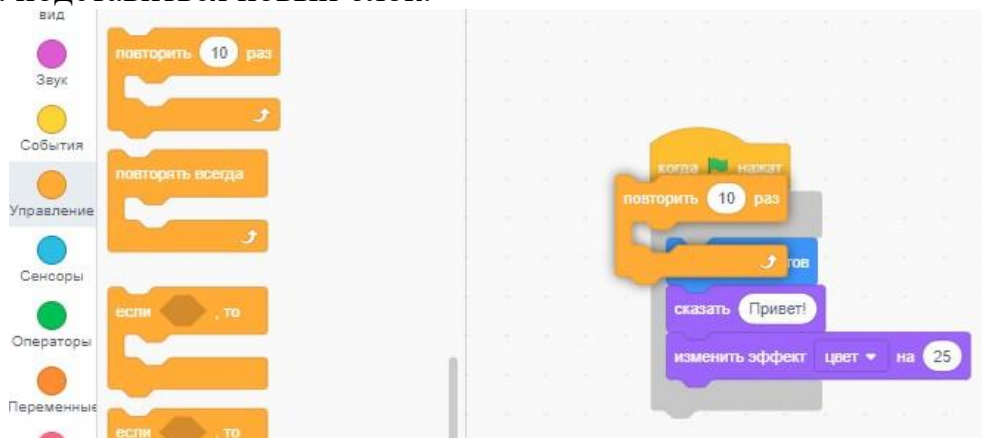


Рисунок19. Вставка цикла в программу

3) Должно получиться следующее:

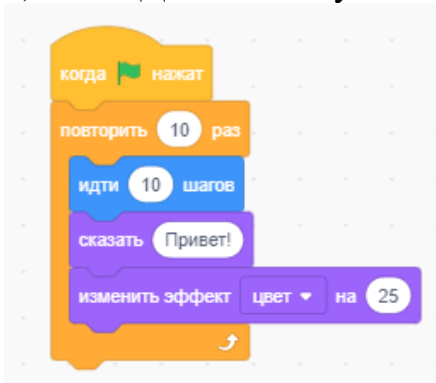


Рисунок 20. Результат вставки цикла

4) Запустите программу и посмотрите, наш Кот теперь движется дольше и дальше.

5) Давайте посмотрим второй тип цикла повторять всегда. Для начала давайте удалим предыдущий цикл, для этого зажмите ЛКМ по блоку повторять 10 раз и перетащите чуть ниже, тем самым мы разорвем связь между блоками:

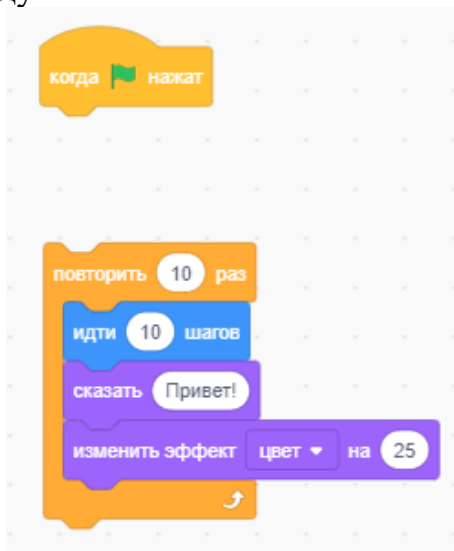


Рисунок 21. Разрыв блоков

6) Далее сделайте тоже самое ухватившись за блок идти 10 шагов:

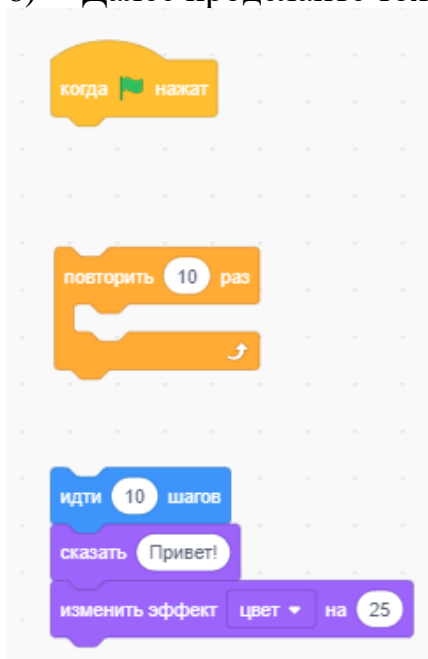


Рисунок 22. Вытаскиваем наши команды из цикла повторить 10 раз

7) Удаляем блок повторить 10 раз зажав ЛКМ и перенесем блок обратно в левую область, где находятся все команды.

8) Перетаскиваем блок повторять всегда и вставляем в него наши команды, как показано на рисунке ниже.



Рисунок 23. Цикл повторять всегда

9) Запускаем и смотрим, что у нас получилось. Кот прошел от одного края до другого уперся в стенку и завис, продолжая идти, а идти уже некуда, прошу заметить, что при этом программа не останавливается и продолжает работать, мы это можем заметить потому, как меняется цвет у Кота.

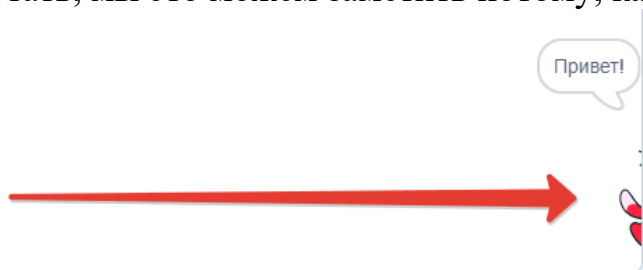


Рисунок 24. Работа цикла повторять всегда

10) Останавливаем программу, в ручную нажав на стоп, хватаем ЛКМ за торчащий хвост кота и перетаскиваем его в начальную точку:

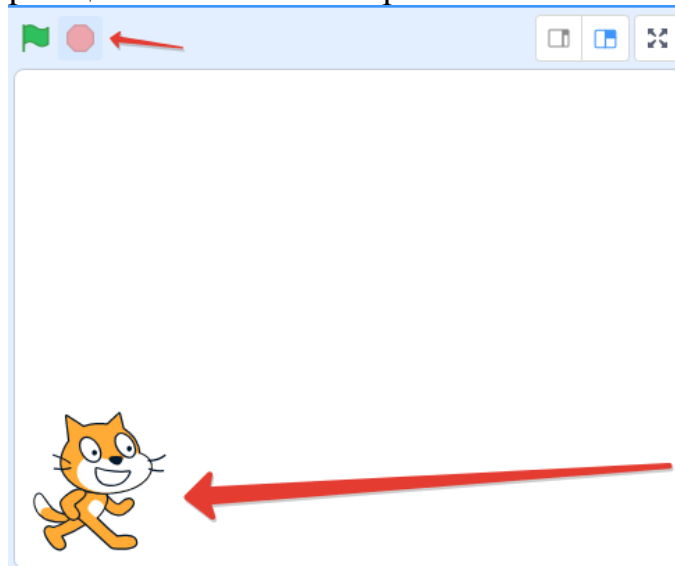


Рисунок 25. Остановка и возвращение в исходную точку

11) Рассмотрим третий тип цикла с наименованием повторять пока не...
Проделайте все такие же операции, как мы делали ранее, чтобы код программы получился как на рисунке ниже:

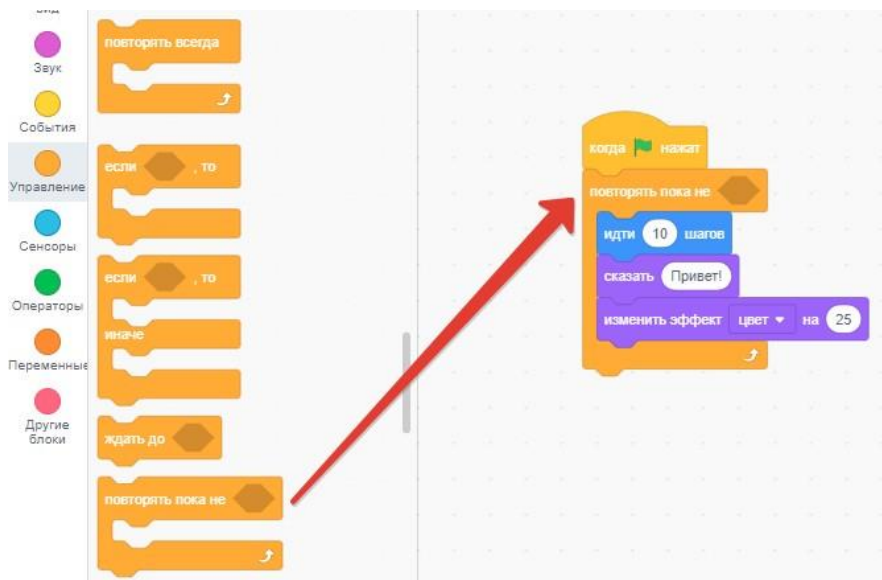


Рисунок 26. Блок повторять пока не...

12) Хочу обратить внимание на то, что в новом блоке присутствует вырезанная геометрическая фигура, это означает, что туда нужно, что-то вставить, тем самым задать условие, при котором будет выполняться цикл.

13) Теперь наша задача сделать так чтобы наш Кот доходил до края, и программа останавливалась сама, как только Кот коснётся края, за это отвечают сенсоры. Зайдите в блок СЕНСОРЫ:

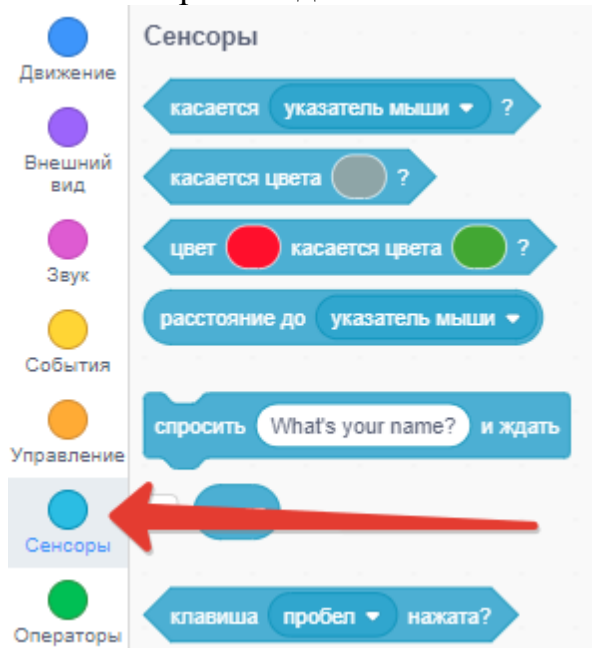


Рисунок 27. Блок СЕНСОРЫ

14) Нажмите ЛКМ на самую верхнюю команду, касается и перетащите в пустую ячейку:

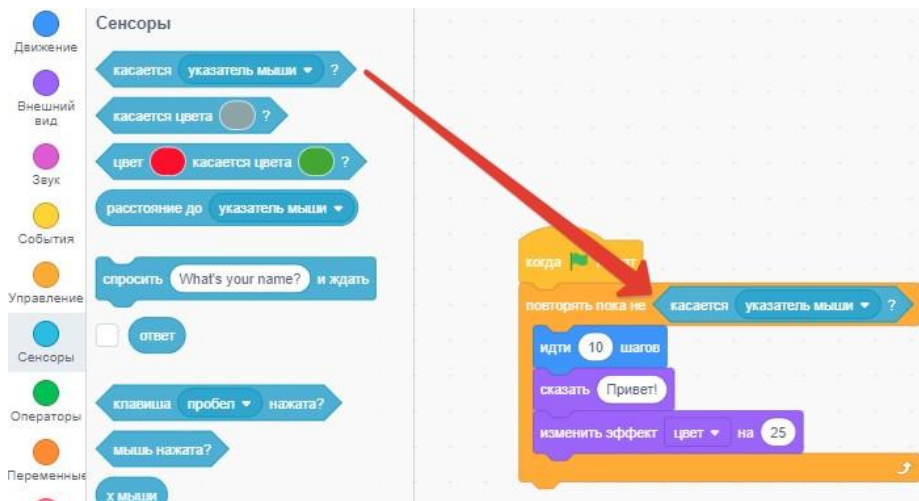


Рисунок 28. Вставка команды *касается*.

14) Теперь нам необходимо нажать на выплывающий список:

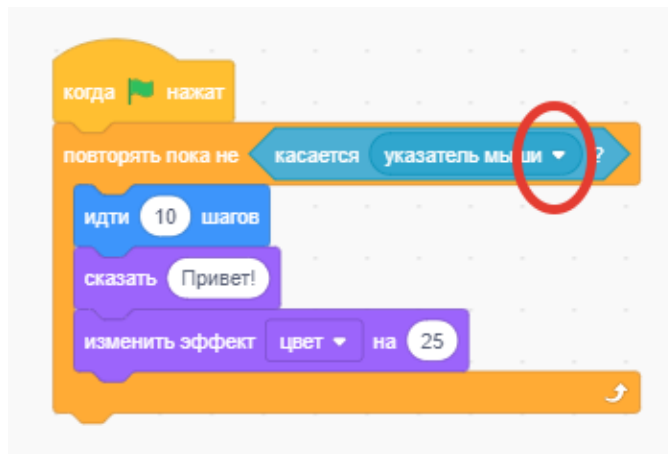


Рисунок 29. Выплывающий список

15) Выбрать край:

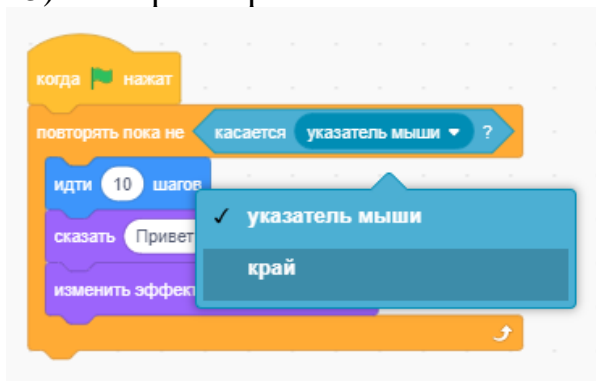


Рисунок 30. Выплывающий список-край.

16) Запускаем и тестируем нашу новую программу. Наш Кот сделал все строго по инструкции, которую мы ему дали:

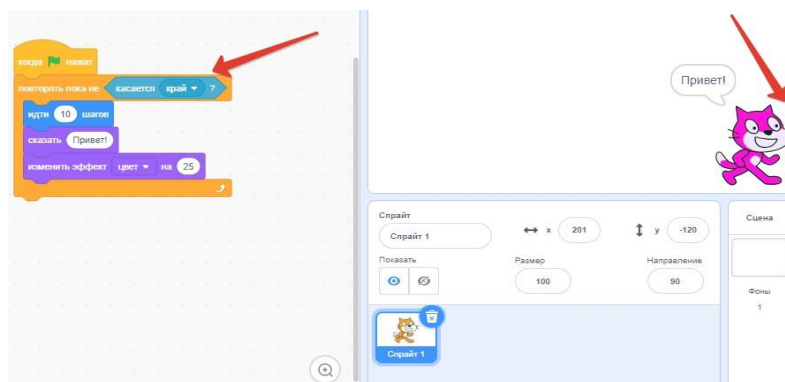


Рисунок 31. Результат выполнения программы

Задания для самостоятельной работы:

- 1) Подумайте, что нужно изменить, чтобы снизить скорость передвижения Кота, как сделаете, покажите учителю вашу работу!

2.4. Занятие №4 Ветвление. Команда "если". Блок ОПЕРАТОРЫ

Цель: Познакомиться с действием команд «Если...» блока УПРАВЛЕНИЕ; простыми и составными условиями, логическими операциями И, ИЛИ, НЕ; рассмотреть команды блока СЕНСОРЫ; познакомиться с элементами блока ОПЕРАТОРЫ, с техникой разветвления анимированных действий.

Команды ветвления «Если...»

- 1) Команда ветвления полной формы:

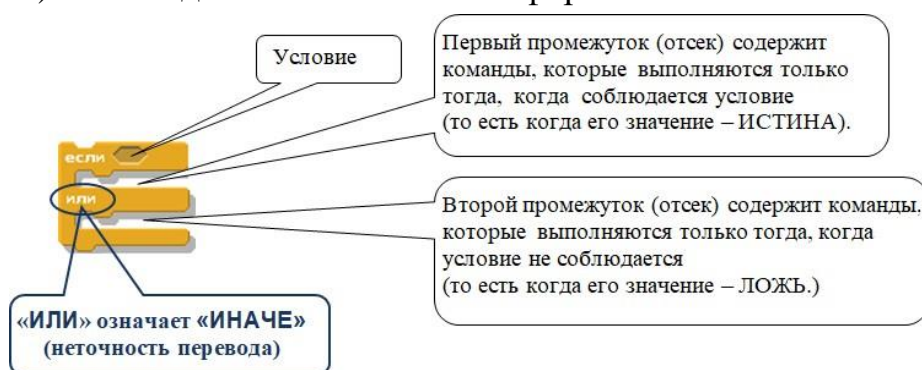


Рисунок 32. Обозначение команды "Если...иначе..."

Команды, следующие за командой «Если...иначе...» (прикрепленные по дней), выполняются не зависимо от условия сразу после выполнения команды «Если...иначе...».

- 2) Команда ветвления сокращенной формы:



Рисунок 33. Обозначение команды "Если..."

Если условие соблюдается, то выполняются вставленные в нее команды, а если нет–команды, следующие за этой командой ветвления.

Условия

1) Простое – может состоять из вопросительных элементов блока СЕНСОРЫ:

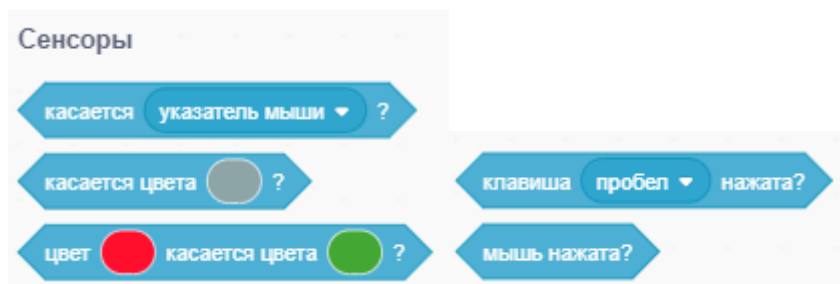


Рисунок 34. Вопросительные сенсоры или элементов сравнения (отношения) блока ОПЕРАТОРЫ:



Рисунок 35. Операторы сравнения

2) Составное – состоит из двух или нескольких вопросов/отношений, соединенных логическими операциями И, ИЛИ, НЕ. Для этого используются следующие команды блока ОПЕРАТОРЫ:



Рисунок 36. Составное условие. Блок ОПЕРАТОРЫ

Задания для самостоятельной работы:

1) Составить скрипт для произвольной анимации, в которой герой задает вопрос и проверяет правильность ответа на него. По примеру:



Рисунок 37. Пример работы программы

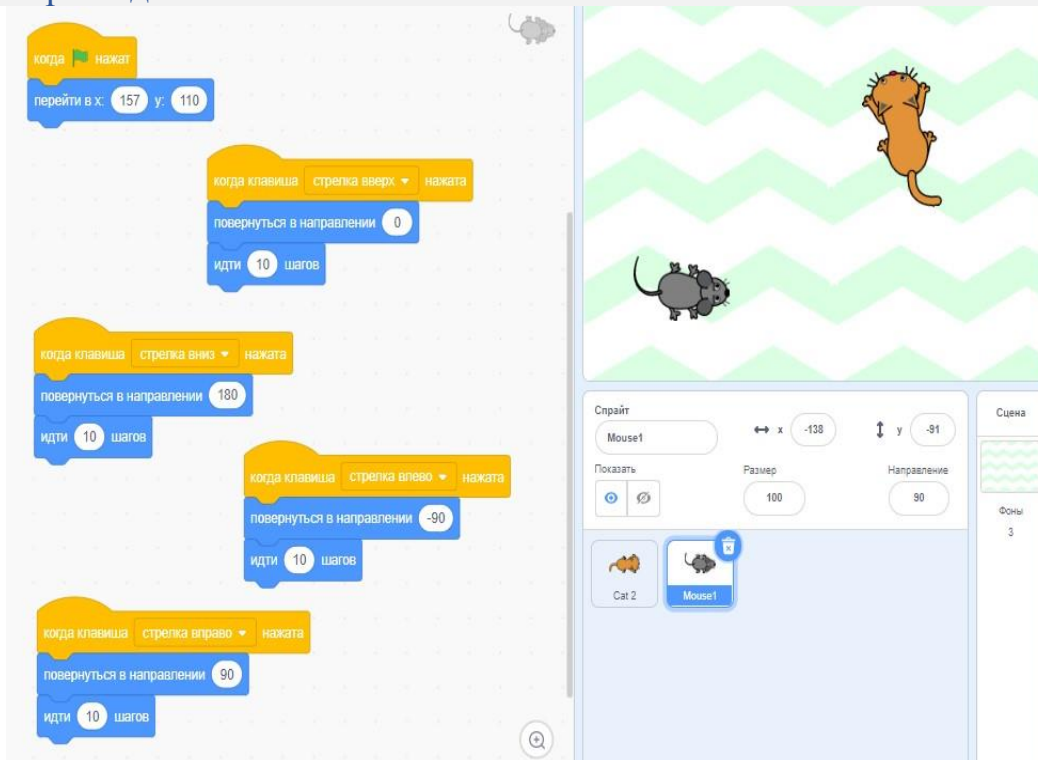
2) Проверьте, как работает данная программа. Затем измените, скрипт так, чтобы в случае неправильного ответа герой говорил: "Неверно попробуй еще раз" и до тех пор, пока не будет дан правильный ответ. Для этого нужно использовать полную форму команды ветвления.

3) После удачного тестирования, покажите свою работу вашему учителю!

2.5. Занятие №5 Кошки-мышки

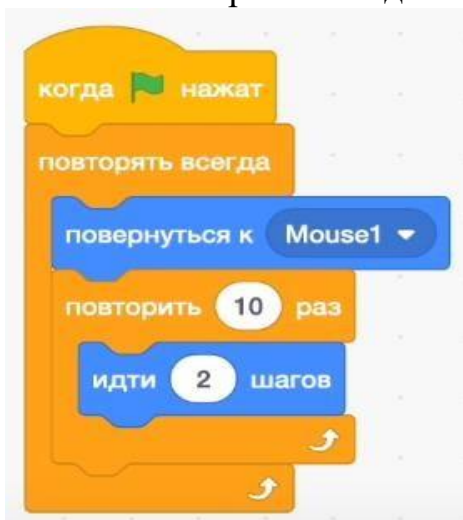
Средствами программы Scratch создай первую простую игру (см. код программы ниже).

Скрипт для Мышки



Скрипт для Кота

1 вариант
/простой код/



2 вариант
/посложнее код/

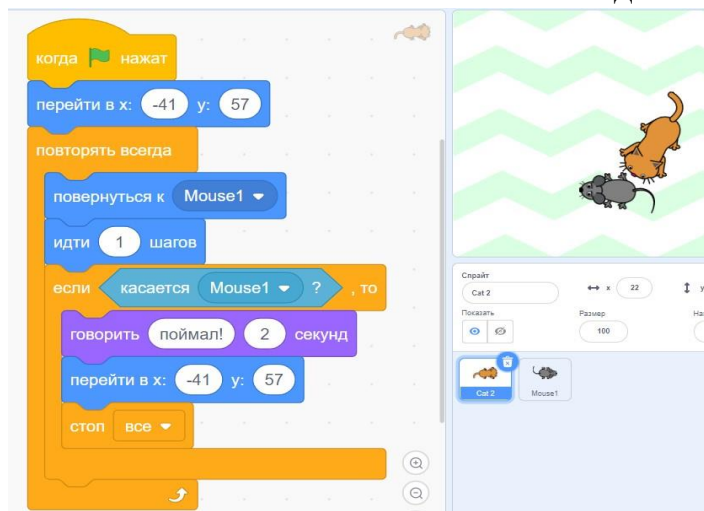
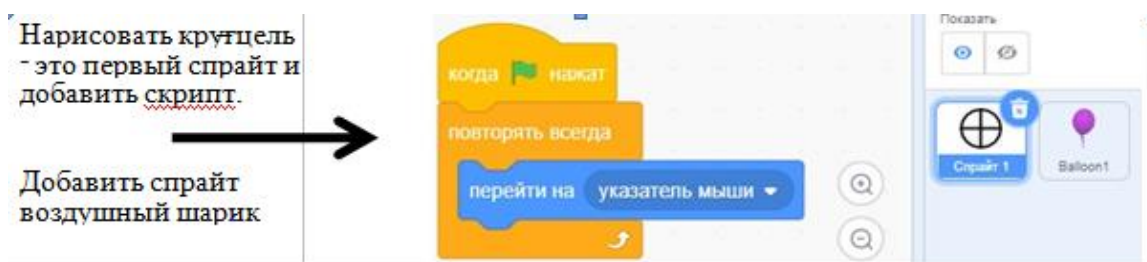


Рисунок 38. Создание игры Кошки-мышки


2.6. Занятие №6 Стреляем по шарикам

Стрелять по появляющимся шарикам, шарики лопают и исчезают, нарисовать спрайт – цель и добавить спрайт Шарик (см. код программы ниже).



Для воздушного шарика:

3 костюма
 1 – просто шарик
 2 – шарик и внутри отверстие
 3 – шарик с большими отверстиями внутри



Скрипт для шарика:

```

        когда нажат
        спрятаться
        повторить всегда
        ждать выдать случайное от 0.3 до 1 секунд
        создать клон самого себя
        когда я начинаю как клон
        показаться
        перейти на случайное положение
        изменить эффект цвет на выдать случайное от 1 до 200
        установить у в -180
        повторить пока не положение у = 180
        изменить у на 5
        удалить клон

        когда я начинаю как клон
        повторить всегда
        если мышь нажата? и касается Спрайт 1? то
        играть звук Pop до конца
        изменить костюм на balloon1-c2
        ждать 0.2 секунд
        изменить костюм на balloon1-c3
        ждать 0.2 секунд
        удалить клон
    
```

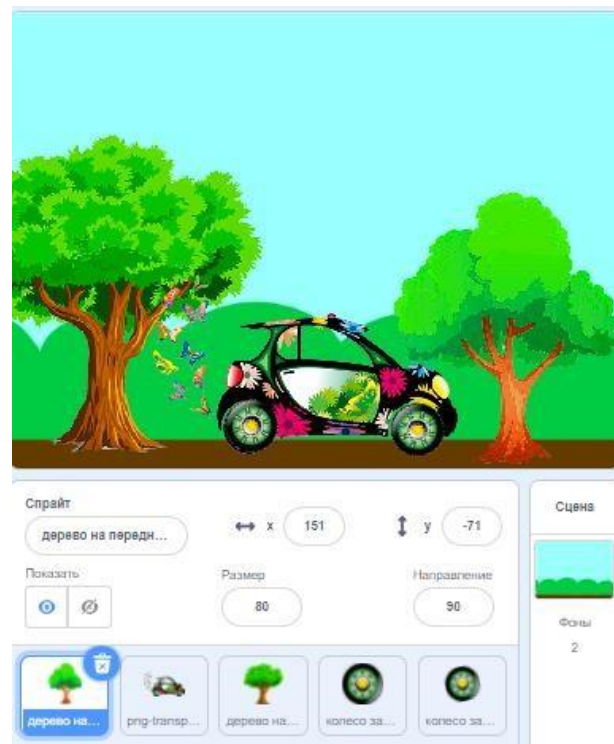
Итого

Рисунок 39. Создание игры Стреляем по шарикам

2.7. Занятие № 7 Движение фона

Машина едет, крутя колесами, мимо нее справа налево двигаются деревья, создавая эффект смены фона. **Задание** – добавь фон, спрайты: 2 дерева, машину и 2 колеса (см. код программы ниже).

1. Создайте новый проект в Scratch.
2. Назовите «Движение фона №1»
3. Поставьте фон для сцены и добавьте спрайтов:



Машина
Дерево 1
Дерево 2
Колесо 1 – заднее
Колесо 2 – переднее

Скрипты для спрайтов

Для дерева Для дерева на заднем на переднем плане

Рисунок 40. Движение фона

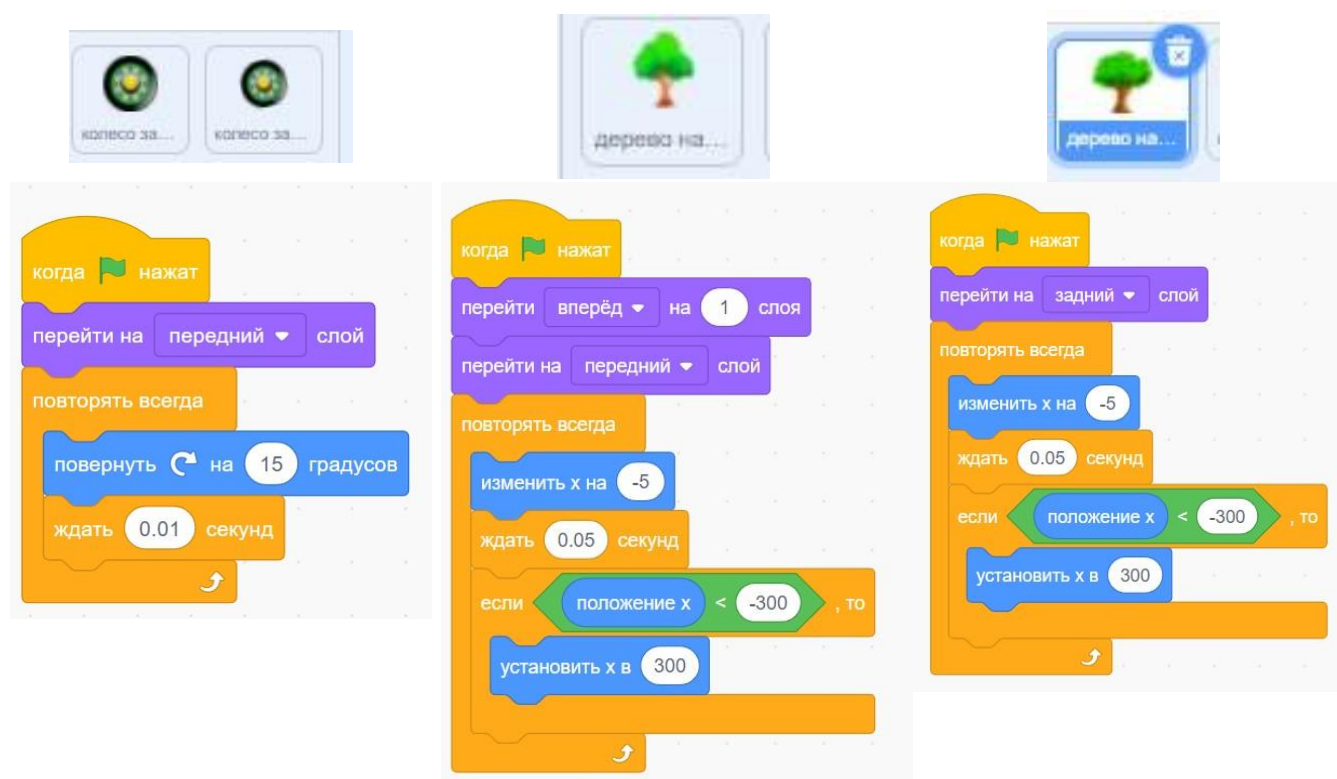


Рисунок 41. Команда для движения фона

Самостоятельно:

добавить 3 дома на задний план и создать скрипт как для дерева на заднем плане

2.8. Занятие №8 Управление с помощью клавиатуры

В среде программирования Scratch можно создавать не только анимации, но и полноценный игры. Чем отличается анимация от игры? Тем, что анимация работает сама по себе без участия пользователя, игра же наоборот требует от пользователя взаимодействия, например: передвижение

главного героя, выстрелы, прыжки и т.д. Мы рассмотрим, как управлять объектами с помощью клавиатуры, прям как в компьютерной игре.

1) Для начала давайте заменим нашего Кота на другой объект, пусть это будет танк, который мы нарисуем сами с помощью геометрических фигур.

2) Удаляем Кота, смотри в правый нижний угол:



Рисунок 42. Удаление спрайта "Кот"

3) Далее, там же справа в нижнем углу наводим курсор на круглую кнопку с изображением кота и выбираем из выплывающего списка "Нарисовать":

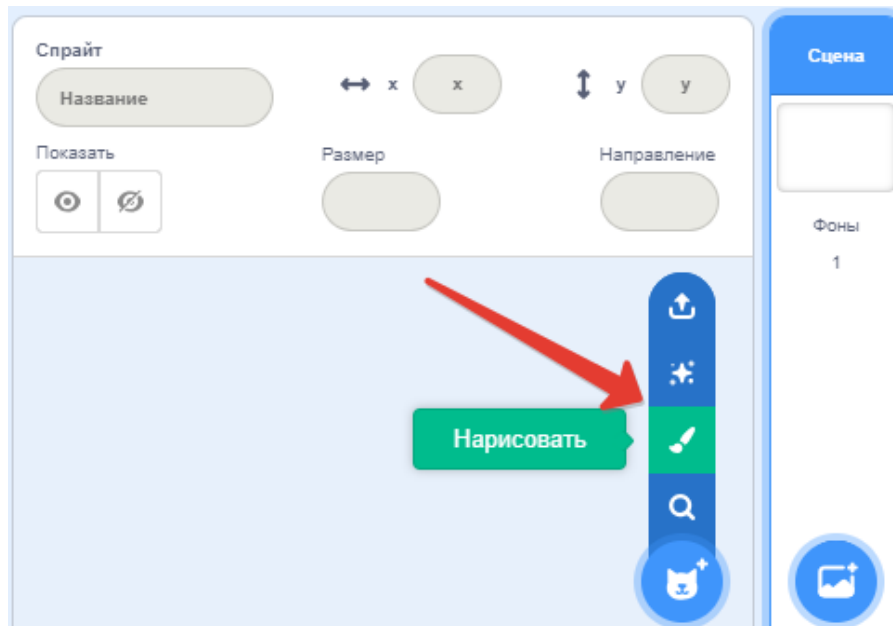


Рисунок 43. Меню рисования

4) В открывшемся окне изменяем цвет, выбираем геометрическую фигуру квадрат, и делаем контур 0:

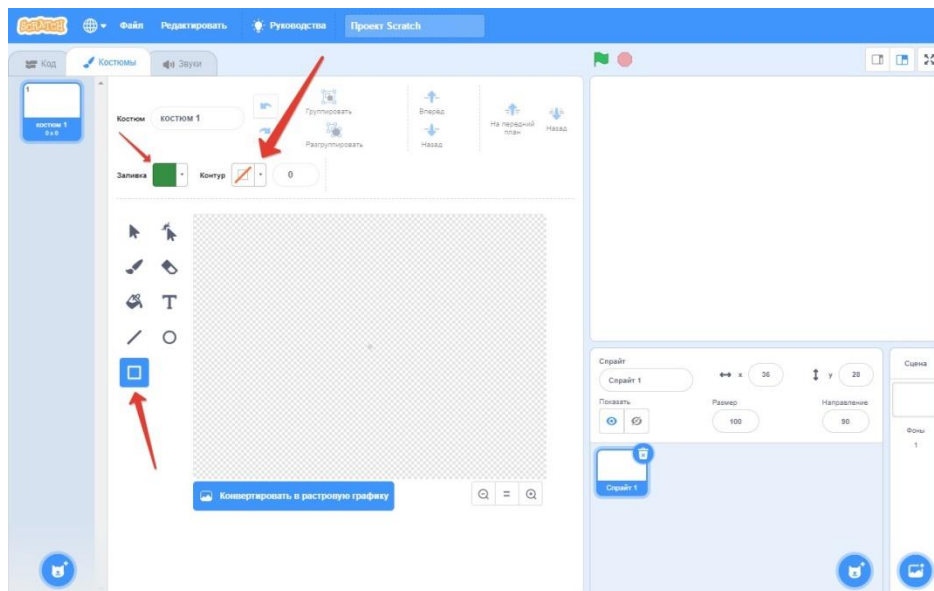


Рисунок 44. Меню рисования с настройками

5) Изменяя эти настройки, самостоятельно нарисуйте примерно такой танк:

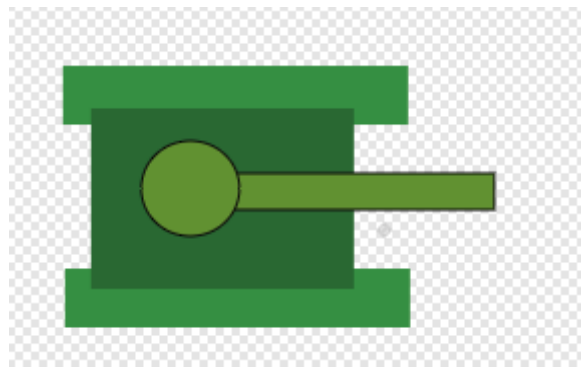


Рисунок 45. Пример нарисованного танка

6) Вернитесь обратно на вкладку Код, что бы приступить программировать движение танка:

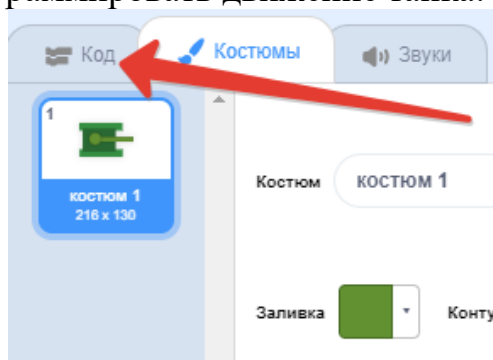


Рисунок 46. Возврат в исходное меню

7) А теперь посмотрите, как легко заставить танк двигаться ТОЛЬКО ЛИШЬ тогда, когда мы нажимаем на определенную клавишу (в нашем случае это клавиша – стрелка вверх):

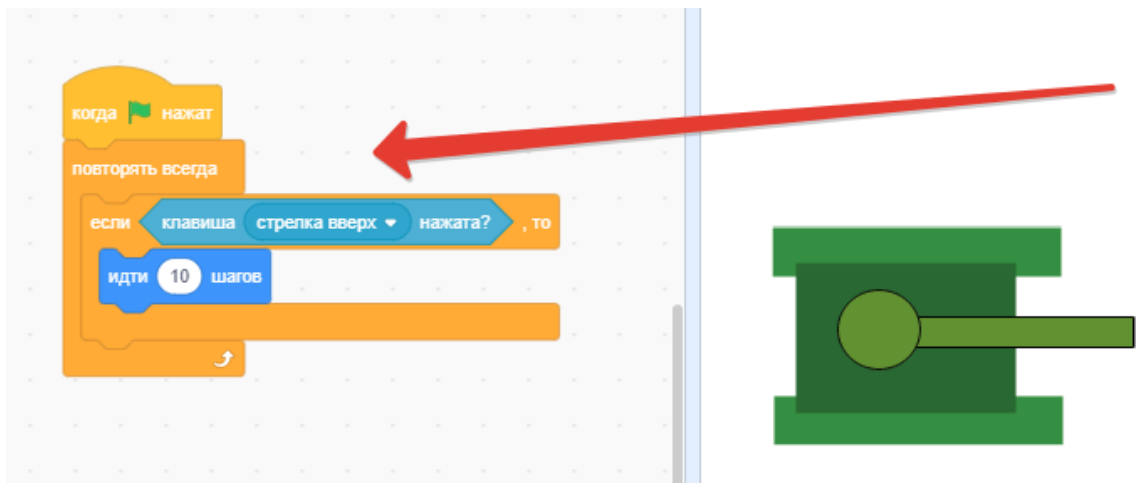


Рисунок 47. Движение вперед

8) Для того чтобы заставить танк поворачиваться вокруг своей оси, точно также ничего сложного нет, также давайте уменьшим размер танка до значения 40 и не забудьте изменить значение градусов со стандартного 15 на 4 радиуса:

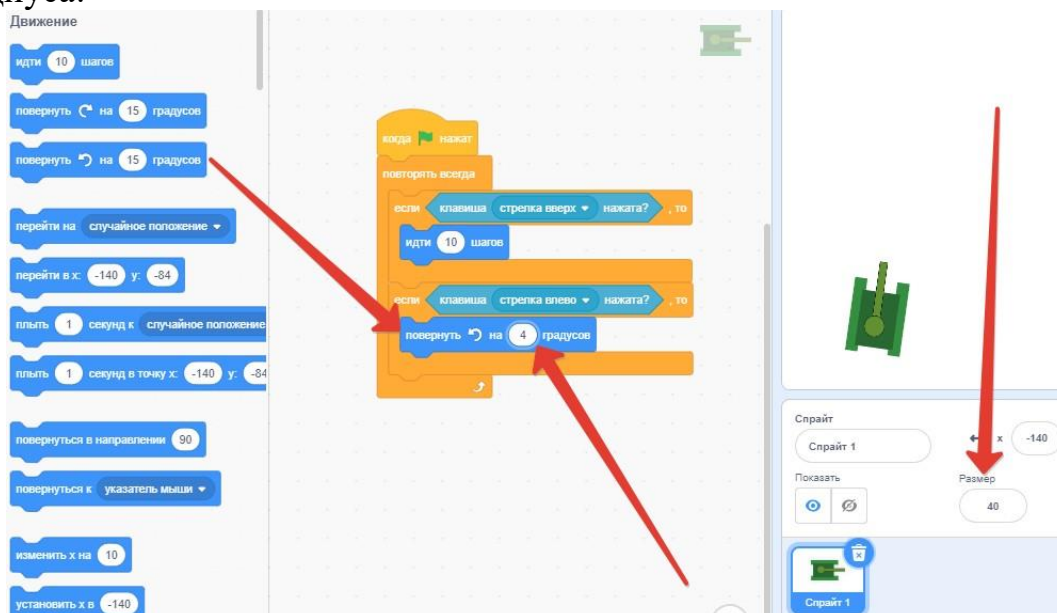


Рисунок 48. Изменение размера и команда поворота

Задания для самостоятельной работы:

- 1) Наш танк едет вперед и поворачивает влево, оставшиеся два направления сделайте самостоятельно (движение назад и поворот вправо);
- 2) Добавьте подходящий фон.

2.9. Занятие №9 Взаимодействие графических объектов

Ответь сам себе на вопрос, разве интересно просто кататься по периметру прямоугольника? Какой же танк и без снарядов. Наша цель на сегодняшний урок сделать так, чтобы при нажатии на пробел из дула танка вылетал снаряд и летел, вперед уничтожая все на своём пути.

- 1) Первым делом давайте нарисуем снаряд, точно так же, как рисовали танк и изменим ее размер, чтобы снаряд и дуло танка были примерно

ОДИНАКОВЫ:

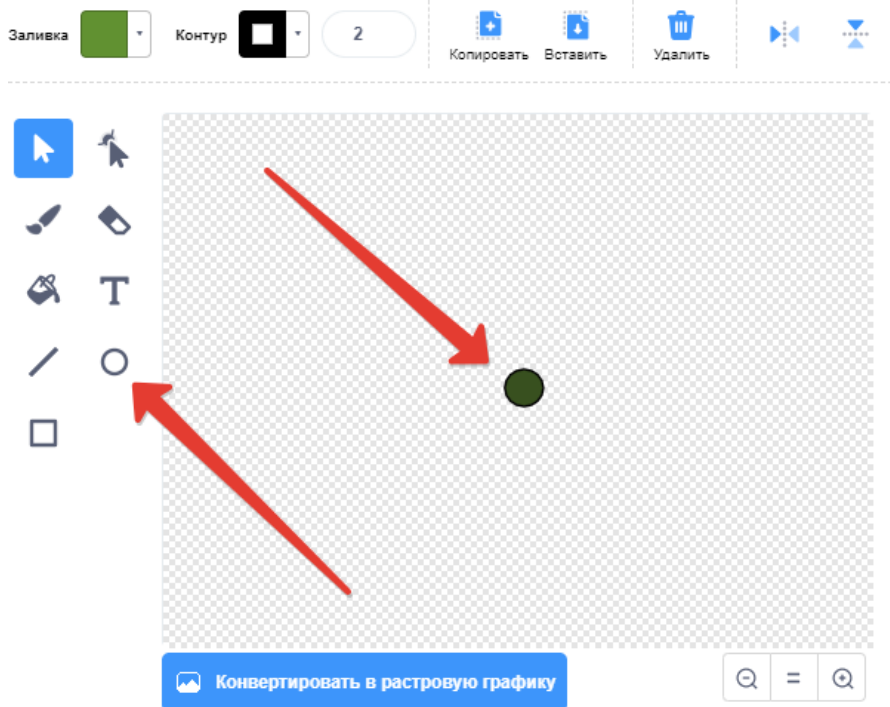


Рисунок 49. Рисуем снаряд

2) Как и раньше готовим стандартный шаблон, для того чтобы наш снаряд работал при запуске программы, не забудьте переключиться с танка на снаряд:

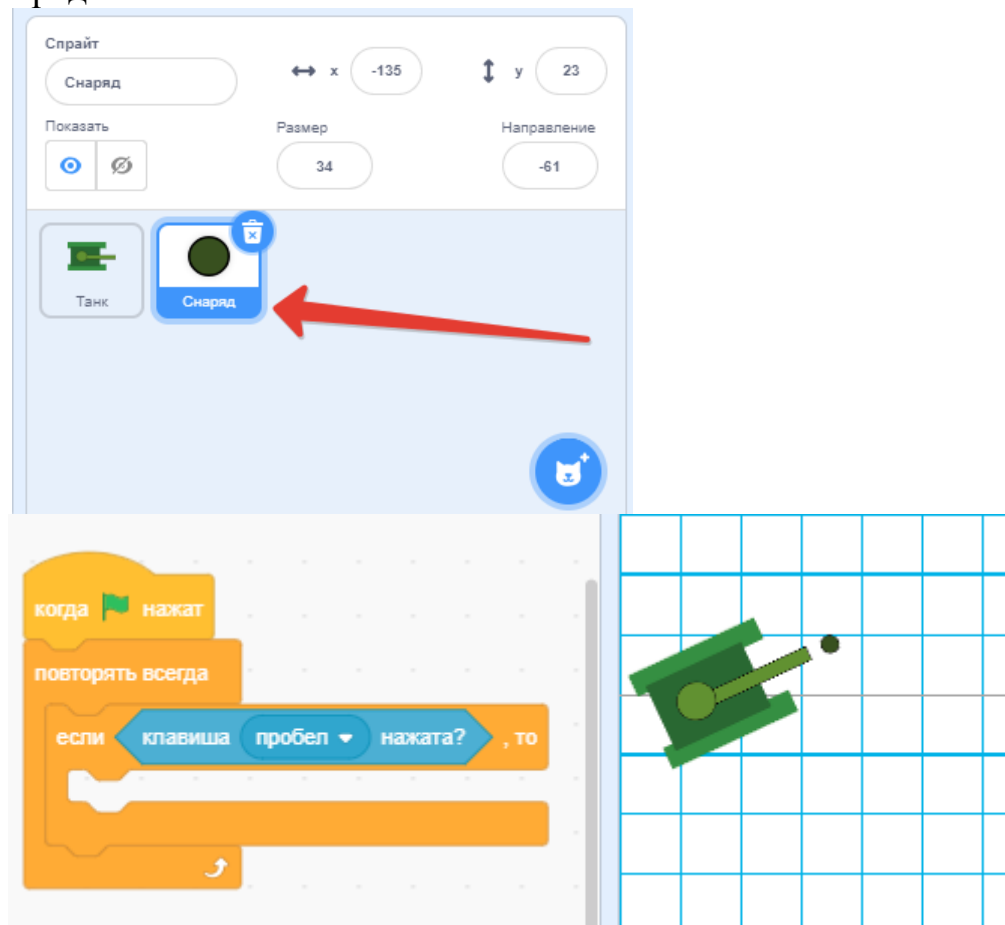


Рисунок 50. Шаблон для запуска по нажатию клавиши

3) Снаряд должен вылетать непосредственно из танка, поэтому первым

деломнам нужно сделать так, чтобы при нажатии на пробел пуля перемещалась в положение танка, для этого используем следующую команду из блока ДВИЖЕНИЕ и вставляем ее в тело цикла "Если...":

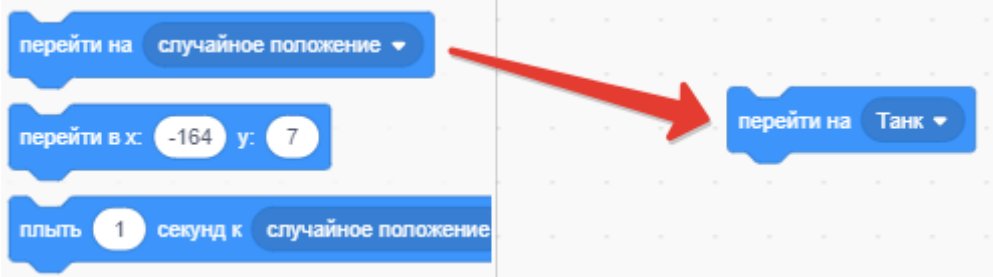


Рисунок 51. Команда быстрого перемещения объекта

4) Далее необходимо повернуть снаряд так, чтобы он смотрел в том же направлении, что и дуло танка, для этого все в том же блоке ДВИЖЕНИЕ находим команду повернуться в направлении 90 и вставляем ниже, в тело цикла:

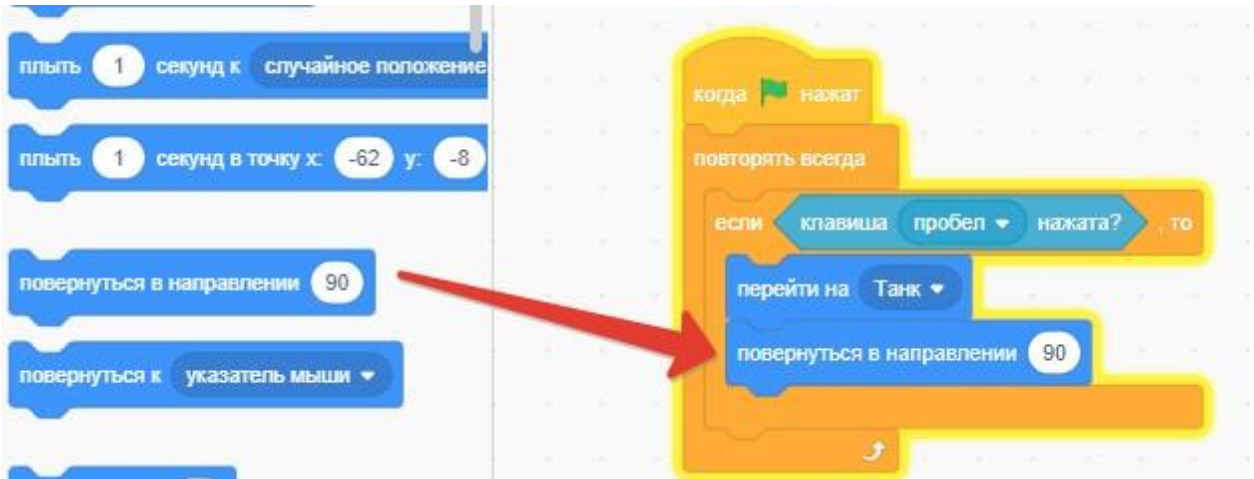


Рисунок 52. Поворот в направлении

5) Если мы оставим все, так как есть, пуля будет вылетать неправильно. Видите овальную область, где написано число 90? Именно в него мы вставим еще одну команду, её вы найдете в блоке СЕНСОРЫ (по умолчанию она называется фон # от Сцена):

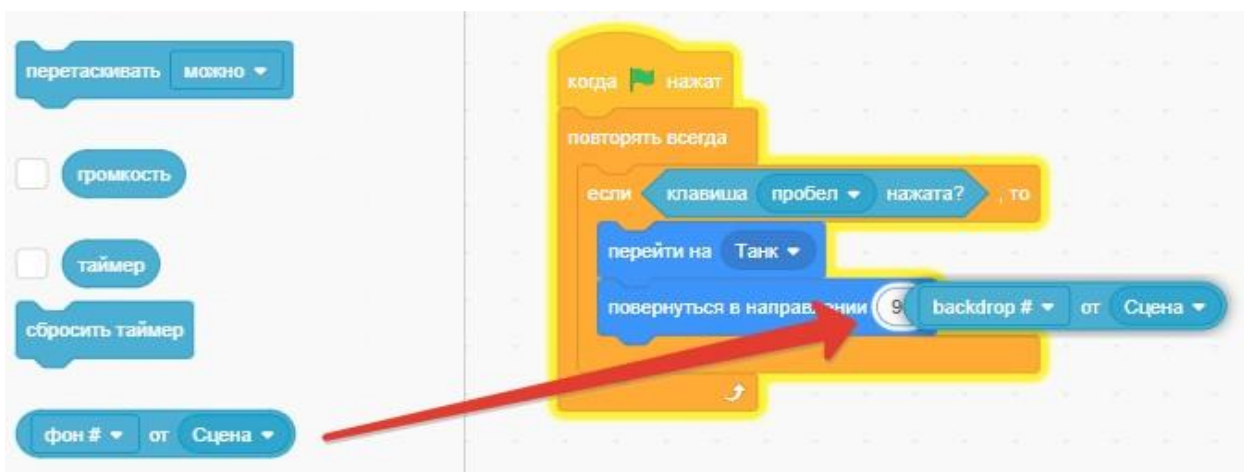


Рисунок 53. Поворот в направлении

6) Нажимаем на надпись Сцена и выбираем имя нашего спрайта Танк (если вы не переименовывали у вас он будет называться Спрайт1). Далее кликаем по второму всплывающему меню, где написано положение x и выбираем направление:

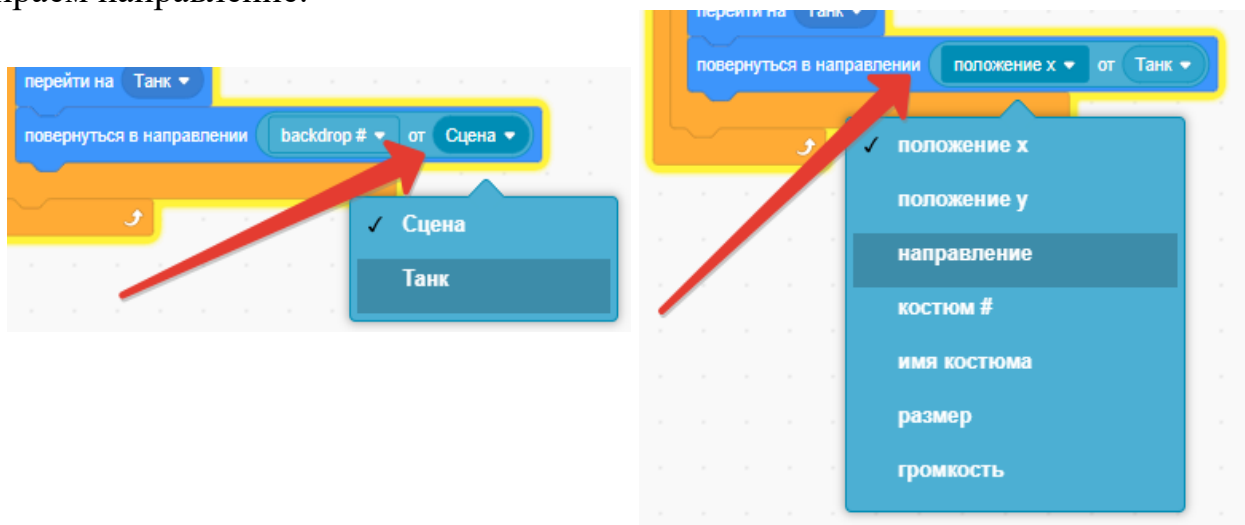


Рисунок 54. Настройка поворота в направлении от Танка

7) Ниже добавим команду идти 20 шагов, чтобы увидеть, что наш снаряд находится и смотрит в правильном направлении от танка, где бы танк не находился. Попробуйте отъехать в другой угол и нажать пробел, снаряд должен переместиться рядом с дулом танка:

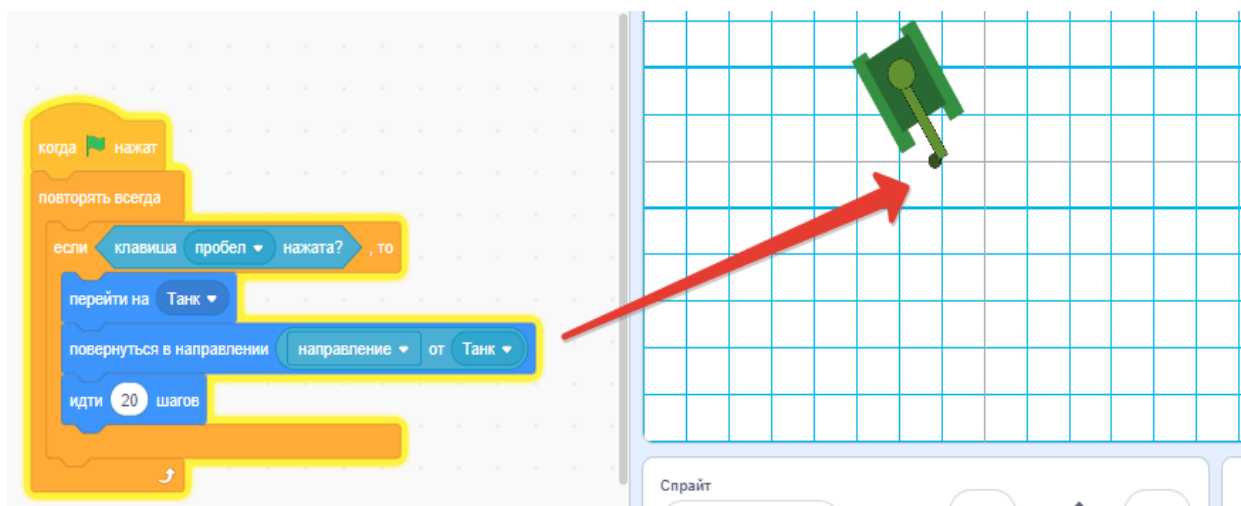


Рисунок 55. Проверка положения снаряда

8) Для того, чтобы снаряд летел до самого края, где бы не находился танк, добавим ниже еще один блок с циклом повторять пока не и сенсором касается край и в цикл вставим идти 20 шагов, протестируй, что получилось:

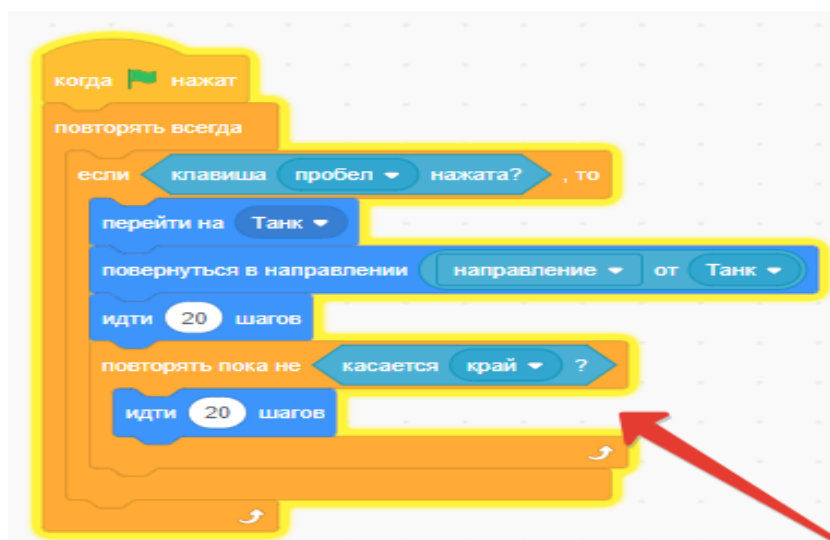


Рисунок 56. Полет пули до края фона

9) Заметили что пуля долетает до края и останавливается, мы этого и добивались, но пуля остается как бы врезавшись в край, это противоречит физике, давайте сделаем так, чтобы пуля вылетала из танка долетая до края - исчезала, тем самым мы добьемся эффект, будто бы снаряд пролетел дальше, для этого есть такие команды в блоке ВНЕШНИЙ ВИД - спрятаться и показаться, добавим их, спрятаться в самом конце, показаться в самом верху:

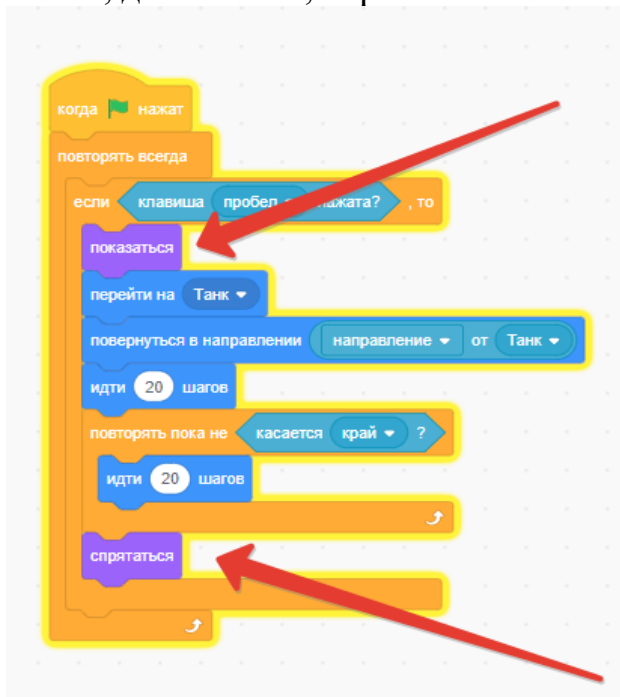


Рисунок 57. Команды показаться и спрятаться

Задания для самостоятельной работы:

- 1) Протестируйте созданную игру, все ли работает, как хотелось бы?
- 2) Добавьте еще один танк и снаряд, назначив клавиши управления на другие. Сделав это, вы сможете играть со своим другом одновременно, например: вына стрелочках, он на клавишах a,w,s,d. Не забудьте про второй снаряд для второго танка.
- 3) Покажите свою работу учителю и получите оценку!

Список используемой литературы

1. Денисова Л.В. Проектная деятельность школьника в среде программирования Scratch / Л.В. Денисова, В.А. Дженжер, В. Рындак // Оренбург. 2009. Режим доступа: <https://sites.google.com/site/orenscratch/nasirazrobotki>. - Название с экрана.
2. Денисова Л.В. Среда Scratch в практике учителя начальной школы / Л.В. Денисова, В.О.Дженжер // Начальная школа. 2012. - № 5. С. 31-35.
3. Кончакова Р.Б. Анимационные программы в курсе информатики для студентов гуманитарных специальностей / Р.Б. Кончакова, М.Ю. Сидляр // Психологопедагогический журнал Гаудеамус. 2014 - № 2 (24). - С. 137-143.
4. Кузнецов А.А., Апатова Н.В. Основы информатики. 8–9 классы. – М.: Дрофа, 2003.