

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ К ДОПОЛНИТЕЛЬНОЙ
ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ
“ПРОГРАММИРОВАНИЕ В СРЕДЕ SCRATCH”**

Пдо Зарыпова Г.Х

Практикум для учащихся по модулю: «Создание игр в среде Scratch

Тема «Линейный алгоритм. Реализация линейного алгоритма исполнителем»

Практическая работа. «Пилотируем самолет»

1. Нарисовать фон: взлетную полосу и небо с облаками.
2. Нарисовать костюм самолета.
3. Присвоить исполнителю костюм самолета.
4. Составить программу:
 - Когда нажмут S - поместить исполнителя в нижний правый угол экрана.
 - Когда нажмут стрелку вверх – самолет перемещается на один шаг вверх.
 - Когда нажмут стрелку вправо – самолет перемещается на один шаг вправо.
 - Когда нажмут стрелку влево – самолет перемещается на один шаг влево.
 - Когда нажмут стрелку вниз – самолет перемещается на один шаг вниз.
5. При нажатии клавиши E – программа останавливается.

Тема «Типы данных. Управляем несколькими исполнителями.»

Практическая работа. «Математик»

В волшебной стране живут ученый кот Математик, Число1, Число2 и Знак математических действий. У каждого Числа есть три костюма в форме чисел, расположенных в строгом порядке - 1, 2, 3. У Знака - 2 костюма в форме знаков арифметических действий: * и +.

Числа и Знак любят играть в Примеры: при каждом нажатии на Зеленый флажок



они случайным образом, как будто кто-то подбрасывает монетку, выбирают свои костюмы.



Как бы не менялись Числа и Знак в примерах, Математик без ошибок их складывает и умножает. Точно не известно, какой волшебник его этому научил, но говорят, что Математик надеется найти мастера, который научит его делить и вычитать.

Секрет Математика в том, что каждый костюм имеет не только имя и рисунок, но и номер. Это обычное число, с которым можно выполнять математические действия. Главное для Математика - узнать, какой номер костюма выпал в примере у каждого Числа и Знака. Мы не раз убеждались, что репортеры знают все. Значит Математику достаточно взять нужные репортеры из категории сенсоры, С-блок полного ветвления если или, логический оператор равенства и математические операторы умножения и сложения.

Сценарий на русском	Сценарий на Скретче
1) Когда игрок щелкает зеленый флаг, 2) Число1 случайно выбирает свой костюм.	
1) Когда игрок щелкает зеленый флаг, 2) Число2 случайно выбирает свой костюм.	
1) Когда игрок щелкает зеленый флаг, 2) Знак случайно выбирает свой костюм.	
1) Когда игрок щелкает по Математику, 2) он определяет, какой костюм сейчас у Знака, если это первый костюм (*), 3) Математик номер костюма Числа1 умножает на номер костюма Числа2, 4) иначе - номер костюма Числа1 прибавляет к номеру костюма Числа2.	

1. Запустите Скретч. Создайте спрайты Математик, Число1, Число2, Знак (рис.1).



Рис. 1. Спрайты проекта Математик

2. Добавьте для спрайтов Число1 и Число2 по три костюма в виде чисел: 1, 2, 3 из папки Letters, а для спрайта Знак нарисуйте два костюма: * (умножение) и + (сложение) (рис. 8.2):

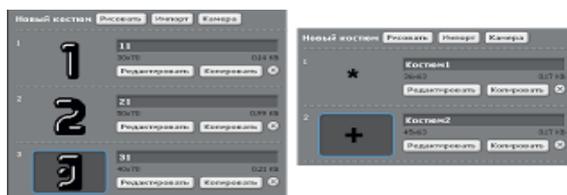


Рис. 2. Костюмы спрайтов число1, число2 и знак

3. Соберите скрипты для каждого спрайта.
4. Нажмите на зеленый флаг, чтобы получить случайные значения костюмов.
5. Нажмите на образ спрайта Математик, чтобы он подумал правильный ответ.
6. Сохраните проект под именем математик.
7. С помощью оператора слияния строковых данных поменяем фразу, которую думает математик. Внесите изменения в блоки подумать по образцу (рис. 3)



Рис. 3. Использование оператора слияния для построения фразы

8. Проверьте работу скрипта.
 9. Сохраните проект.
- Дополнительное задание.* Доработать программу, чтобы научить Математика умножать и складывать другие числа. А заодно - делить и вычитать.

Тема «Ветвления»

Задание 1. Алгоритм нахождения большего из двух заданных чисел

Соберем программу для реализации алгоритма нахождения большего из двух заданных чисел А и В. (Рис. 25).

1. Создайте две переменные А и В.
2. Соберите скрипт по образцу (Рис. 25).

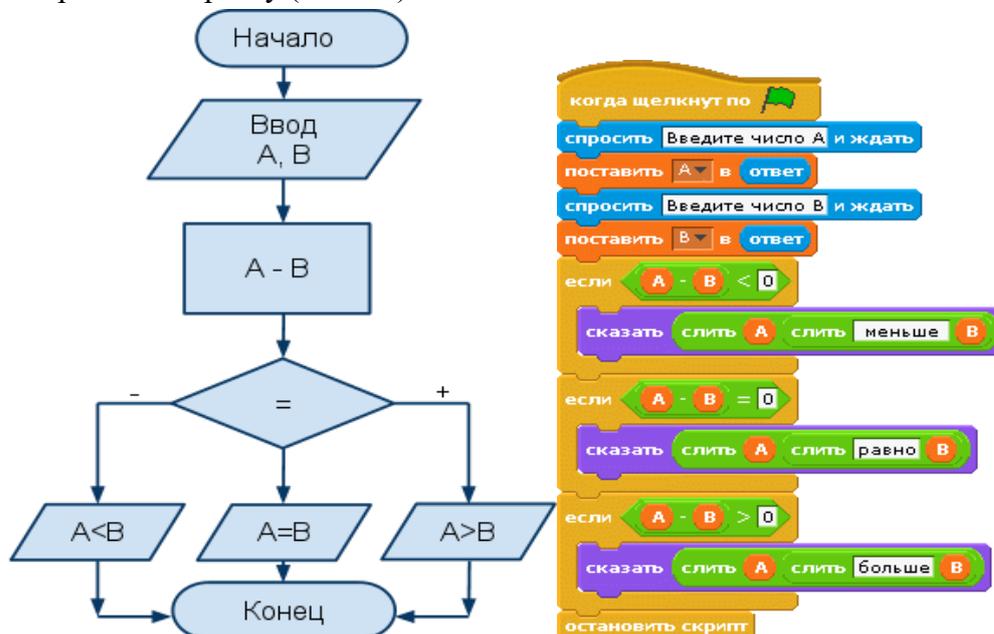


Рис. 25. Реализация алгоритма нахождения большего из двух заданных чисел с использованием неполного ветвления 3. Проверьте его работу.

Практическая работа. Угадай число

Игрок должен угадать случайное число, "задуманное" компьютером, в диапазоне от 0 до 100. Алгоритм приведен на рисунке 26.

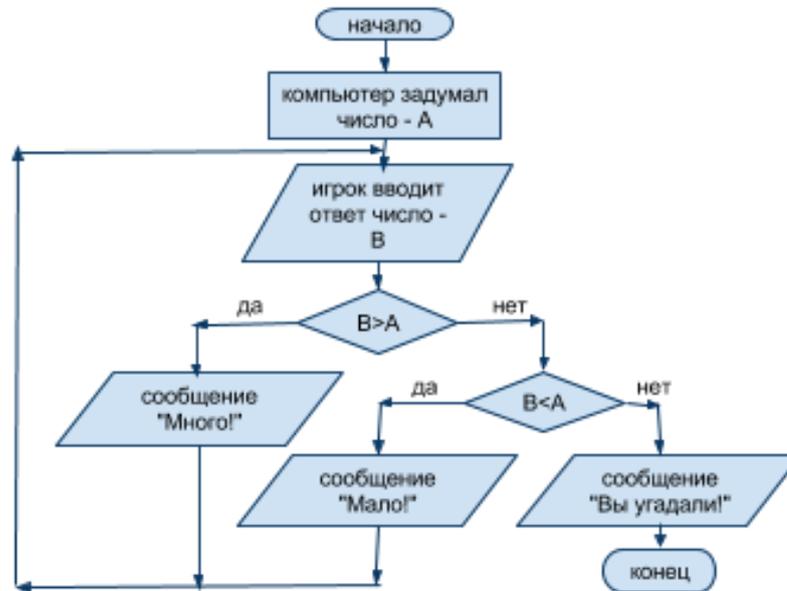


Рис. 26. Алгоритм игры «Угадай число»

1. Начните новый проект.
2. Создайте глобальную переменную число.
3. Соберите скрипт игры (рис. 27.)



Рис. 27. Скрипт игры "Угадай число"

4. Поиграйте.
5. Сохраните проект.
6. Как можно сделать игру интереснее и зрелищнее? Попробуйте внести изменения и сохранить свой вариант проекта.

Вопросы

1. Сколько условных операторов в языке Скретч?
2. Сколько условных операторов можно вставить во внутрь одного условного оператора: а) один; б) не более двух; в) любое количество; г) вообще нельзя.

3. Можно ли использовать условный оператор без логических репортеров? Почему?
4. Почему во всех языках программирования есть условные операторы?
5. Приходилось ли вам когда-нибудь делать выбор? Приведите примеры.

Тема «Списки»

Задание 1. Элементы списка из текстового файла

Научимся добавлять элементы списка из текстового файла.

1. Запустите Блокнот для создания текстового файла
2. Наберите с клавиатуры три строки, начиная с новой строки каждый элемент
3. Сохраните текст с именем уроки.txt, выбрав кодировку UTF-8.
4. Создайте список.
5. Кликните по монитору списка на сцене, нажмите на кнопку импортировать и выберите файл уроки из той папки, где вы его сохранили.
6. Вы должны увидеть на сцене монитор списке уроки, в котором каждому элементу соответствует строке текста:
7. Также можно сделать обратную операцию - сохранить элементы списка в текстовый файл. Нажмите правой кнопкой мыши по монитору списке уроки на сцене. Нажмите на кнопку экспортировать и сохраните данные списка в файл.

Практическая работа. Да или нет?

Списки часто используются для программирования различных викторин. Сделаем компьютерную игру «Да или нет?» В которой Скретч случайным образом будет называть предмет и спрашивать, съедобный ли он? Добавим к проекту подсчет баллов и времени игры с помощью переменных.

Для проекта нам понадобятся 2 списка: съедобное, несъедобное и 5 переменных: баллы, номер, список, начало и время.

1. Создайте новый проект.
2. Создайте два глобальных списка: съедобное, несъедобное (рис.15).



Рис. 15. Списки для игры

3. Создайте пять глобальных переменных:
 - баллы** - для подсчета количества правильных ответов;
 - список** - для случайного выбора в каждом туре игры одного из списков - 1 (съедобное) или 2 (несъедобное);
 - номер** - для случайного выбора в каждом туре одного из трех номеров элементов списка;
 - начало** - для сохранения стартового значения таймера;
 - время** - для сохранения продолжительности игры.
4. Соберите скрипт игры по образцу (рис. 16)

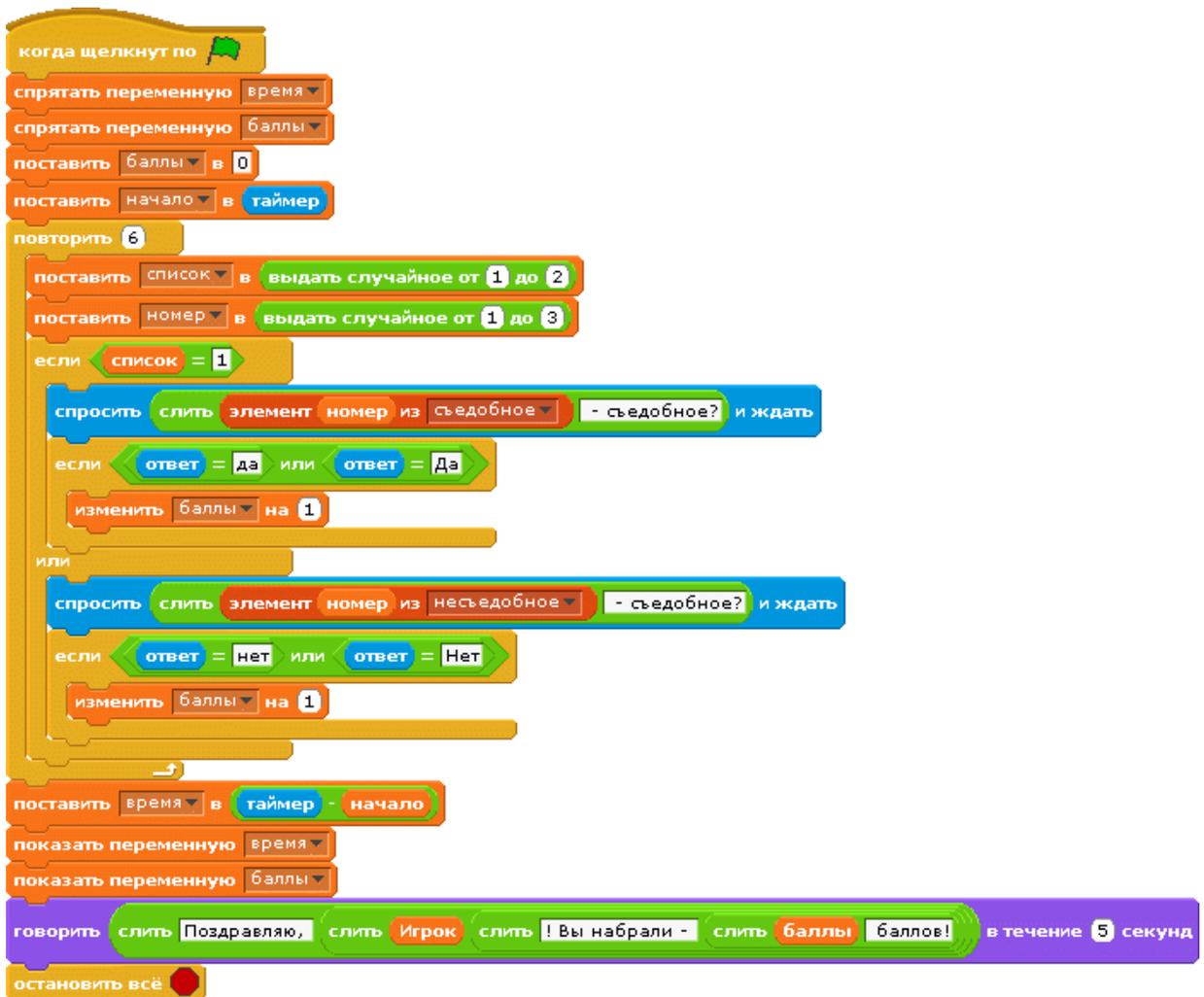


Рис. 16. Скрипт игры «Да - Нет»

5. Проверьте работу скрипта.
6. Сохраните проект.

Задание 1. Улучшим игру, добавив в нее журнал ответов игрока в виде списка - игра. Наша программа должна записывать в журнал заданный вопрос и полученный от игрока ответ. По итогам игры список можно сохранить в текстовом файле, что мы уже умеем делать. Создайте глобальный список игра. Блоки добавления элементов в список игра нужно установить в определенные места скрипта. Самостоятельно решите, где именно они (рис.17.) должны находиться, чтобы правильно записывать ответы игрока?

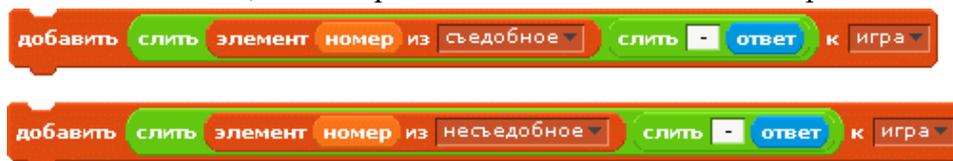


Рис. 17. Блоки записи вопросов и ответов игрока в список игра

В список игра можно также записать продолжительность игры и полученные баллы. Для этого достаточно в нужное место скрипта вставить два блока (рис. 18). Самостоятельно решите, где именно они должны находиться.

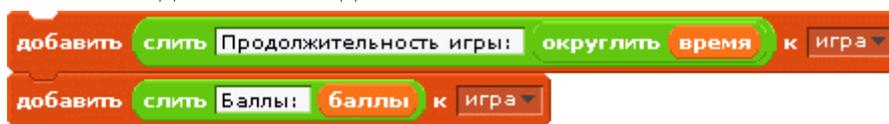


Рис. 18. Блоки записи продолжительности игры и баллов в список игра При желании можно также записать имя игрока. Для этого, как не трудно догадаться, нужно

поставить вопрос об имени в начале игры, до старта подсчета времени игры, и сохранить полученный ответ в списке игра (рис. 19).

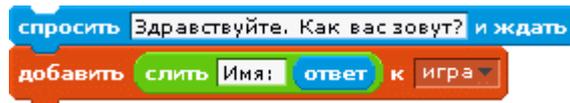


Рис. 19. Сохранения имени игрока в списке игра Сохраните

проект да_нет_2 с внесенными изменениями.

Задание 2. Как можно сделать проект еще интереснее?

Во-первых, можно добавить количество съедобных и несъедобных объектов. Вы это можете сделать самостоятельно.

Во-вторых, нашей игре не хватает зрелищности. Было бы интересно не только называть объекты, но и показывать их изображения. Однако списки не могут сохранять образы. Эту проблему легко решить, создав еще один спрайт, список костюмов которого будет состоять из образов съедобных и несъедобных объектов. Этот спрайт, получив сообщение образ, должен показать костюм, который соответствует предмету. В наших списках содержатся объекты, образы которых можно найти в папке Things, при желании их можно нарисовать самостоятельно. Создайте новый спрайт **образы**.

Сделайте для спрайта список костюмов в той же последовательности, которая изображена на рисунке (рис.20.), с 1 по 3 - съедобные, с 4 по 6 - несъедобные.

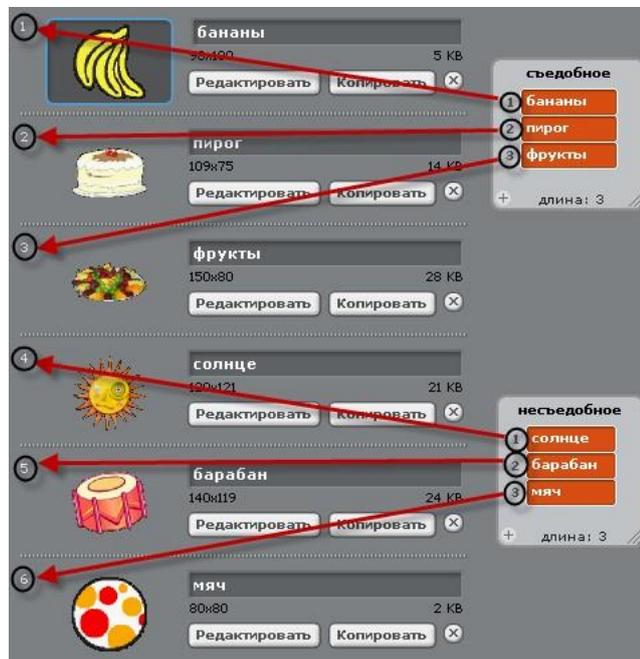


Рис. 20. Список костюмов для вопросов викторины

Добавьте в скрипт викторины блок отправки сообщения образ сразу после того, как будут случайно выбраны списки и объекты (рис. 21).

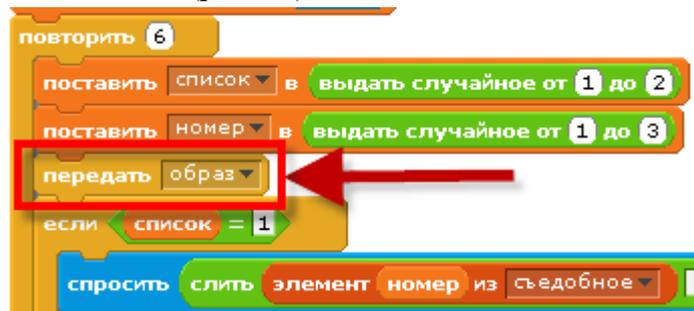


Рис. 21. Отправка сообщения другому спрайту

Создайте для спрайта **образы** скрипт обработки сообщения образ. Если значение переменной список= 1, показывается костюм, номер которого равен номеру элемента в списке **съедобное**. Иначе показывается образ, номер которого равен номеру элемента в списке **несъедобное**. Сделайте так, чтобы в начале игры спрайт **образы** был скрыт и появлялся только после сообщения образ.

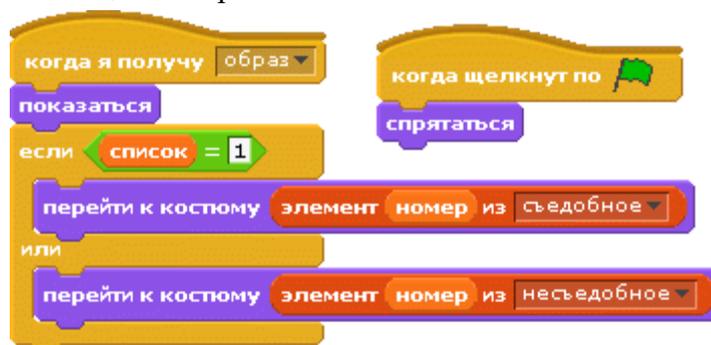


Рис. 22. Скрипты для показа нужных образов Поиграйте.

Сохраните проект.

Задание 3. Можно ли сделать игру еще интереснее? Создавая проект, мы не обращали внимания на сцену, звуки и графические эффекты. Попробуйте внести собственные изменения в проект и сохраните его ремикс под другим именем.

Практическая работа. Диалог

Для этого проекта нам понадобится два спрайта собеседников - Елена и Сергей, глобальная переменная - фраза и локальные списки - слова для каждого спрайта, в них будут храниться реплики собеседников.



Рис. 23 Локальные списки для спрайтов

Переменная фраза выполняет роль счетчика реплик, поэтому ее значение увеличивается на единицу после того, как спрайты обмениваются репликами. Она также служит условием для остановки игры, когда ее значение равно четырём, то есть реплики исчерпаны. Диалог, после нажатия на зеленый флаг, начинает Елена. Скрипт присваивает переменной фраза начальное значение 1, выводит первую реплику из списка слова и посылает сообщение Сергею.



Рис. 24. Скрипты диалога по спискам

Скрипт остановки игры мы отдадим сцене, которая будет ждать, когда фраза = 4, чтобы остановить диалог.

Используя в качестве образца списки слов, предлагаемые скрипты и костюмы, сделайте свой вариант диалога. Если вы увеличите количество фраз в списках, вам придется изменить и константу четыре в скрипте сцены.

Диалоги можно строить на любом языке. Подумайте, как добавить в проект анимацию?

Сохраните проект. **Вопросы**

1. Как в программировании называют порядковый номер элемента в списке?
2. Какие типы данных можно хранить в списке: а) числа; б) строки; в) логические?
3. Как называется список, который связан со всеми объектами проекта?
4. Какой тип кодирования символов нужно использовать при создании списка в Блокноте? Почему так?
5. Приходилось ли вам составлять списки дома или в школе? Если да, то для чего они были нужны?

Тема «Циклы»

Практическая работа. Диалоги -2

В проекте Диалог мы использовали 2 собеседников. Теперь наша задача - сделать такой проект, в который можно легко добавлять любое количество собеседников. Для этого все повторяющиеся данные мы вынесем в переменные и списки. Используя возможности ветвлений и циклов, добавим в проект анимацию.

Объект1 всегда будет начинать диалог, оставаясь в одной точке на сцене. Каждый новый собеседник должен начинать движение от правого края сцены после получения сообщения приходиОбъект#, говорить фразы из списка слова после получения сообщения Объект#, идти назад, получив сообщение уходиОбъект#. Переменная часть этих сообщений - #. В

проекте мы воспользуемся глобальной переменной № и оператором слияния строковых данных.

1. Откройте проект Диалог.
2. Создайте глобальную переменную с именем №.
3. Соберите скрипт сцены. В основе алгоритма - цикл со счетчиком повторений - 2. Это число равно количеству собеседников, которые будут вступать в диалог со спрайтом Объект1. Сцена в проекте играет роль светофора - запускает поведение других объектов, меняя значения переменных и посылая сообщения.

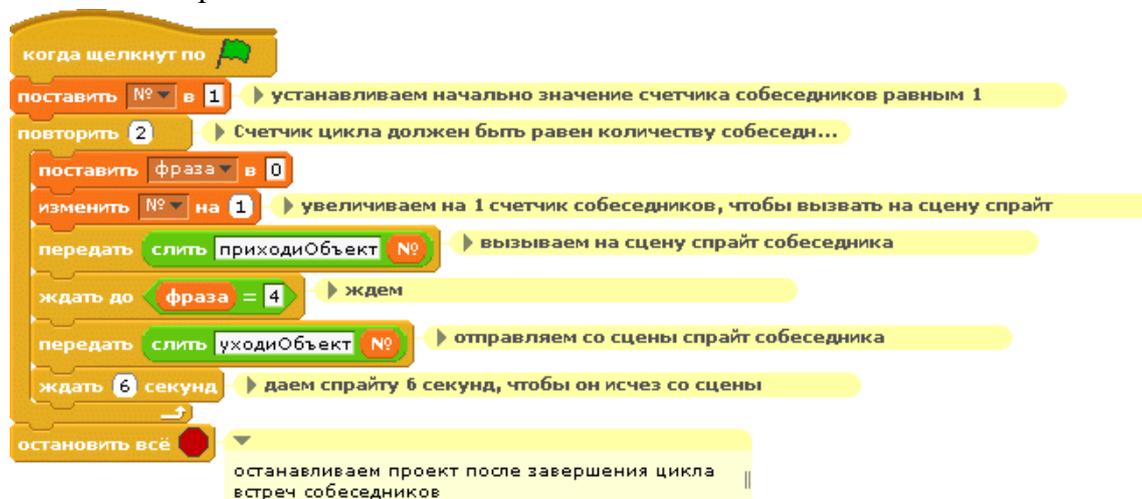


Рис. 28. Алгоритм действий сцены

4. Блок передать может в качестве сообщения использовать и константы, и репортеры:



По своей природе сообщение - это разновидность переменной величины. Только у сообщений нет собственных репортеров. После создания нового сообщения оно появляется в выпадающих списках блоков передать и заголовка когда получу. Причем сообщение исчезнет само, если блок когда получу его не использует для запуска скриптов. Нам нужно создать набор всех сообщений проекта, чтобы появилась возможность использовать соответствующие заголовки:



Рис. 29. Набор сообщений проекта

5. Соберите скрипт появления на сцене Объект1 (рис. 30). Если значение переменной фраза = 0 (это делает сцена в начале своего цикла), цикл всегда запускает выполнение трех действий.

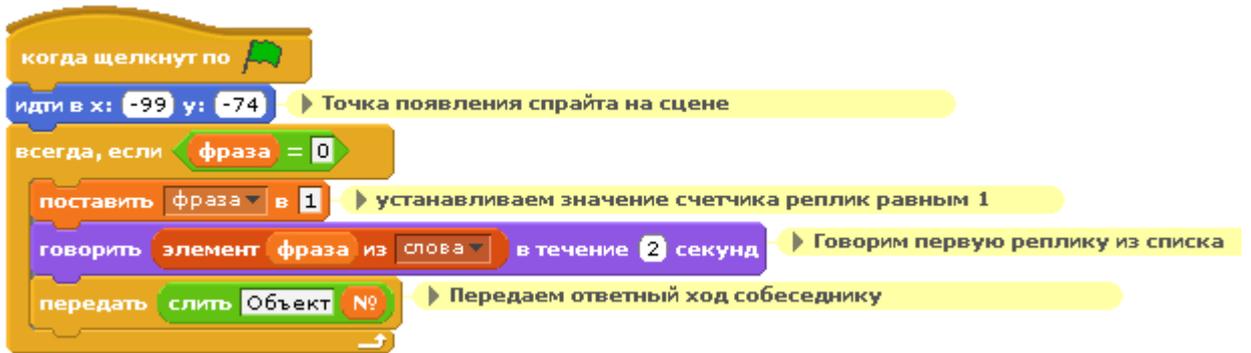


Рис.30. Алгоритм действий Объект1 по нажатию на зеленый флаг

6. Соберите скрипт для обработки сообщения Объект1 (рис. 31)

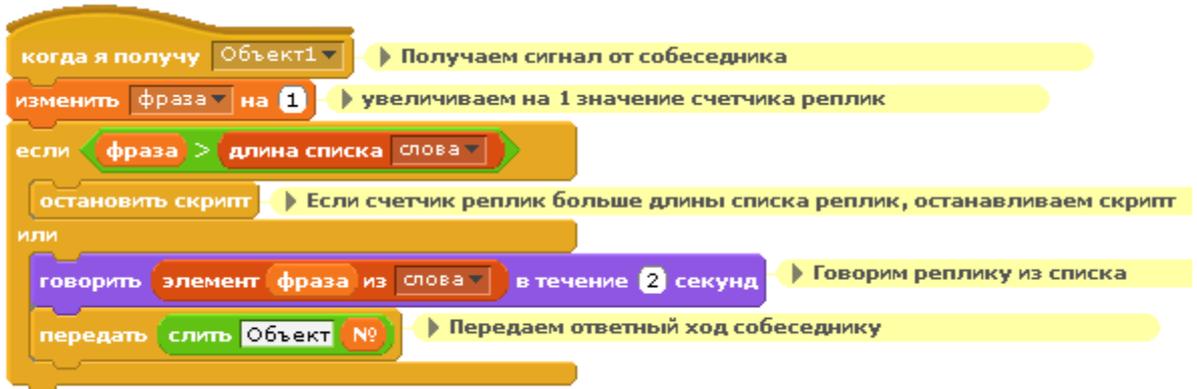


Рис. 31. Алгоритм обработчика сообщения Объект1

7. Соберите скрипт для обработки сообщения приходиОбъект2 (Рис.14.13), который отправляет Сцена (рис. 32).

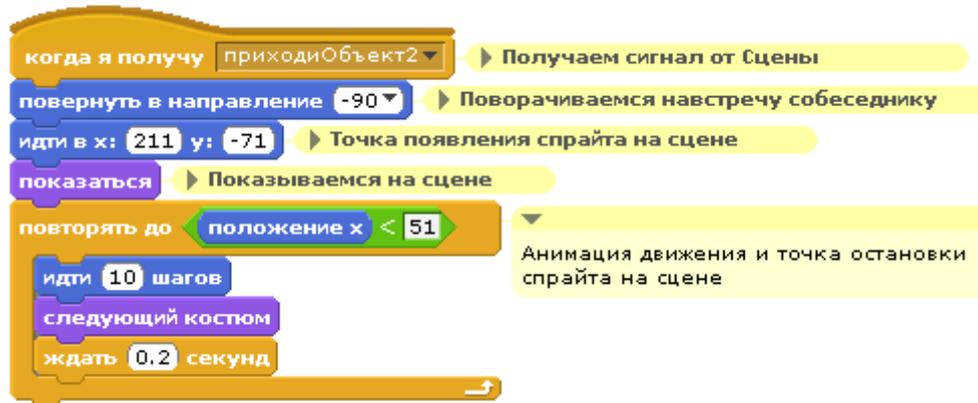


Рис. 32. Алгоритм обработчика сообщения приходиОбъект2

8. Соберите скрипт для обработки сообщения Объект2 (рис. 33), которое отправляет Объект1.



Рис. 33. Алгоритм обработчика сообщения Объект2

9. Соберите скрипт для обработки сообщения уходиОбъект2, которое отправляет Сцена (рис. 35)

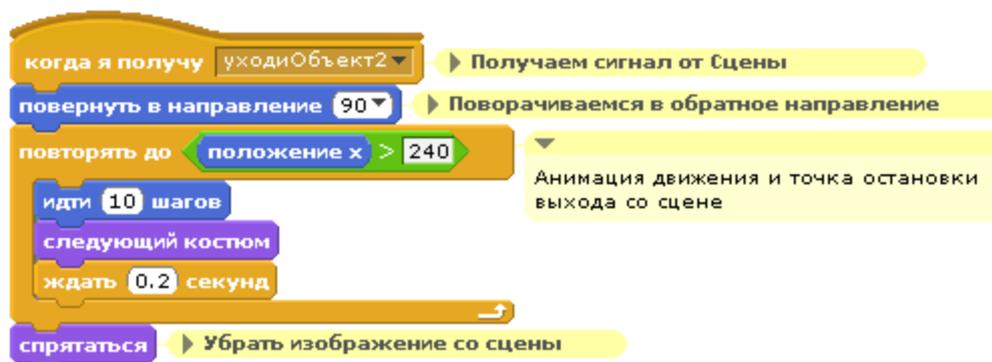


Рис.35. Алгоритм обработчика сообщения уходиОбъект2

10. Дублируйте спрайт Объект2 со всеми его скриптами и списком фраз в Объект3, измените костюмы нового спрайта.
11. Измените имя персонажа Объект3 в элементе №2 локального списка слова: «Мышка. А тебя как зовут?» Внесите изменения в заголовки скриптов спрайта Объект3, выбрав соответствующие уведомления (рис. 36).

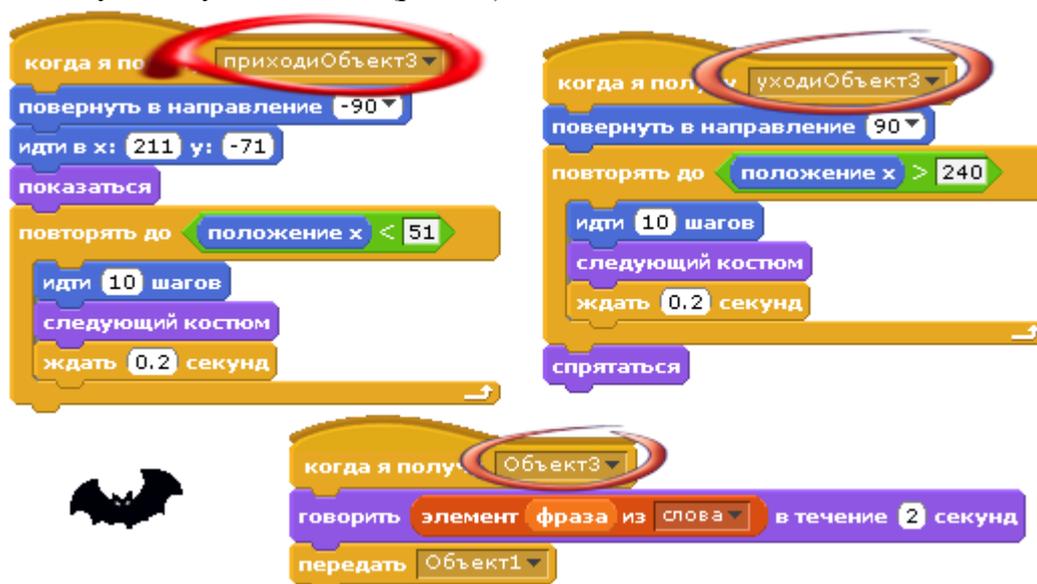


Рис. 36. Скрипты для спрайта Объект3 отличаются только заголовками

12. Запустите проект на выполнение. Поиграйте.
13. Сохраните проект.

Тема «Константы и сенсоры»

Задание 1. Учитель математики

Выполняя проект Математик, мы научили спрайт умножению и сложению. Теперь мы научим его спрашивать у нас правильный ответ и определять, правильно мы ответили или нет. Для этого добавим в скрипт математика блок спросить, а также включим несколько С-блоков для создания ветвлений программы в зависимости от полученных данных.

1. Откройте проект Математик.
2. Выделите спрайт Математик и внесите изменения в его скрипт.

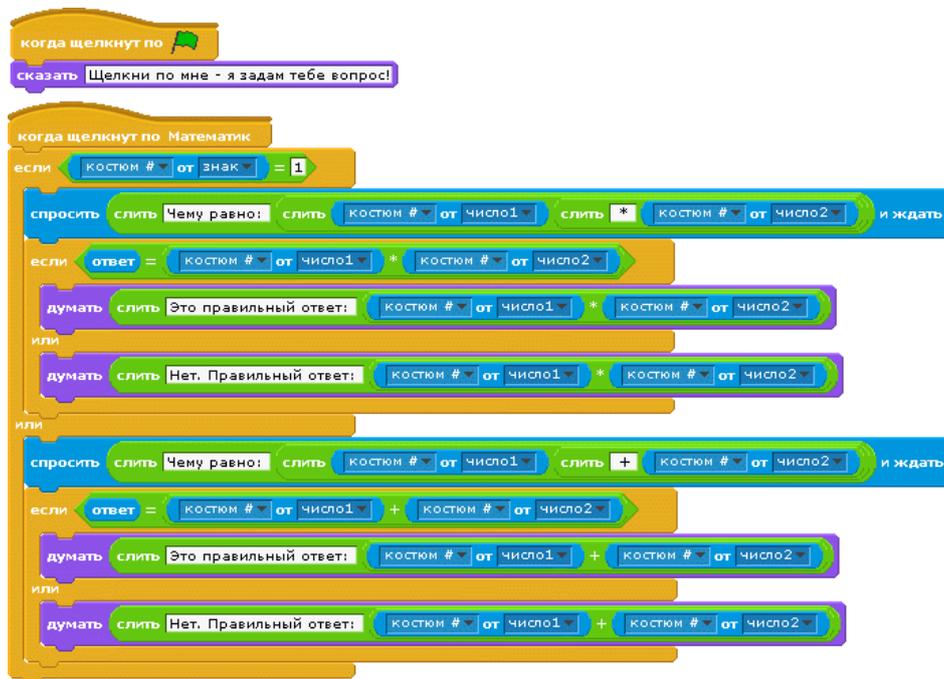


Рис.4. Скрипт викторины

3. Проверьте работу скрипта.
4. Сохраните проект в файле Учитель математики.

Задание 2. Математик с таймером

Добавим в скрипт математика подсчет времени, которое потребуется игроку для ответа на вопрос. Для этого мы перезапустим таймер перед вопросом и выведем его округленное до целого значение, когда получим ответ.

1. Откройте проект Учитель математики.
2. Перейдите на спрайт математик и добавьте в скрипт после заголовка блок перезапустить таймер (рис. 5.).

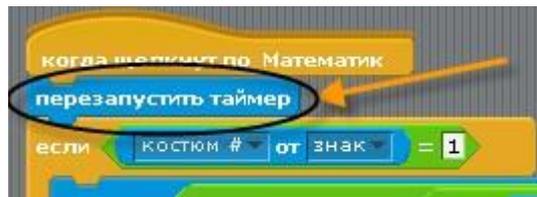


Рис. 5. Перезапуск таймера в начале выполнения скрипта

3. Добавьте в блоки думать дополнительные блоки слияния строк для вывода округленного значения таймера:

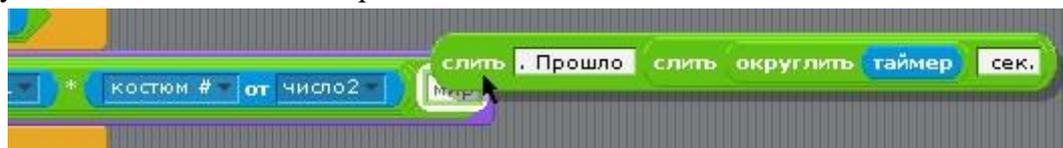


Рис. 6. Вывод округленного значения таймера

4. Запустите скрипт на выполнение. Теперь спрайт должен сообщать не только о правильности ответа, но и о потраченном времени.
5. Сохраните проект Математик с таймером.

Задание 3. Неуловимый спрайт

Запрограммируем действия спрайта в ответ на приближение указателя мыши. Пусть спрайт всегда перемещается на 10 шагов, если расстояние до указателя мыши меньше величины размера спрайта (рис.7).

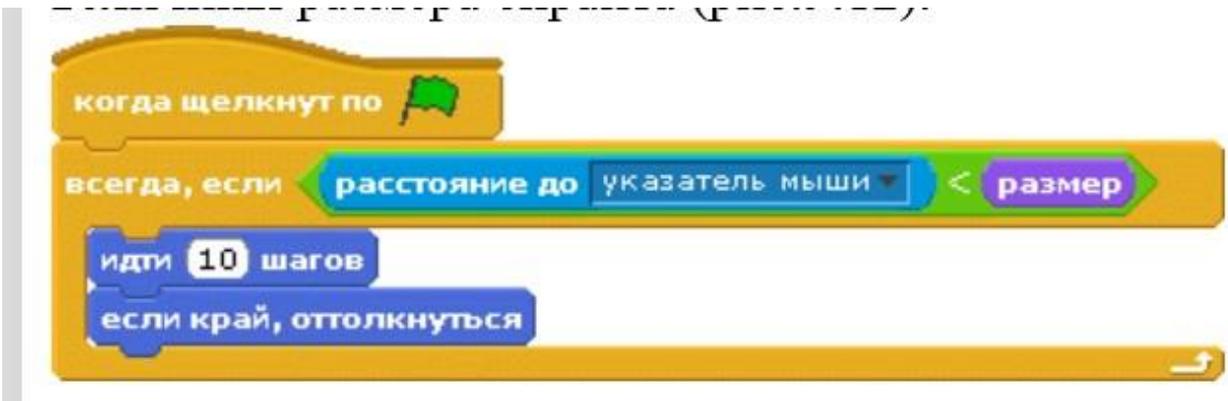


Рис. 7. Использование репортера расстояния для перемещения спрайта

Эту задачу можно решить разными способами. Например, репортер размер из категории Вид можно заменить аналогичным репортером из сенсоров:



Вместо блока идти 10 шагов, можно использовать блоки перемещений в случайные точки сцены. Ведь спрайты могут перемещаться, как ракеты, если вы не забыли.

Задание 4. Тень мяча

Создайте новый проект. Добавьте в него спрайт мяча и нарисуйте спрайт Тень. Соберите для тени скрипт по образцу (рис.9.14). При перемещении мяча его тень должна двигаться параллельно по оси X.

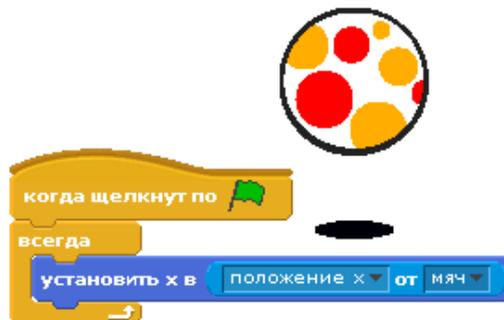
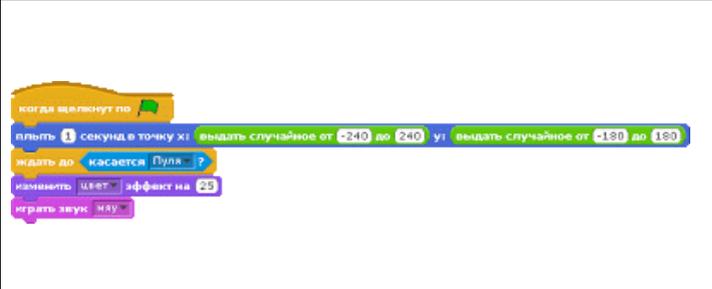


Рис. 8. Программирование тени мяча

Практическая работа . Маг и Кот

В этом проекте мы научимся использовать константы, сенсоры и блоки передачи сообщений.

Описание на русском языке	Описание на Scratch
<p>1) В нижнем левом углу сцены стоит Маг. У него в руке волшебная палочка.</p> <p>2) Маг всегда поворачивается вследза Котом, оставаясь на том же месте.</p> <p>3) Если игрок нажимает на пробел,</p> <p>4) Маг запускает Пулю.</p>	<p>The image shows a Scratch script starting with a 'when clicked' block. It is followed by 'move to x: -161 y: -96', 'always' block with 'turn to face Cat', and an 'if' block with condition 'space key pressed?'. Inside the 'if' block, there are 'send message' and 'stop script' blocks.</p>

<p>1) Получив сигнал, 2) Пуля появляется у Мага, 3) летит в точку размещения Кота, 4) исчезает.</p>	 <p>Scratch script for bullet movement:</p> <ul style="list-style-type: none"> when green flag clicked show go to Mag move 1 seconds in direction: position x from Cat, y: position y from Cat hide
<p>1) Когда игрок нажимает на зеленый флаг, 2) Кот движется в случайном направлении. 3) Когда Пуля касается его, 4) Кот меняет цвет, 5) говорит Мяу.</p>	 <p>Scratch script for cat movement and sound:</p> <ul style="list-style-type: none"> when green flag clicked move 1 seconds in direction: random from -240 to 240, y: random from -180 to 180 wait until touched by Bullet change color effect to 25 play sound: meow

1. Сделайте 3 спрайта: Пулю нарисуйте сами. Для Мага можно взять рисунок wizard1 из папки Fantasy:
2. Если у вас есть возможность подключить микрофон, сделайте так, чтобы Маг запускал Пулю не по нажатию на пробел, а после хлопка в ладоши:
3. Сохраните проект в файле Маг.

Переменные

Задание 1. Как создать и удалить переменную

При выполнении этой команды Скретч выводит диалоговое окно, в котором необходимо определить имя и область действия создаваемой переменной.

1. Щелкните мышкой по категории Переменные в палитре команд (рис.11.1).
2. Нажмите на кнопку Создать переменную.
3. В строке имени введите имя - переменная.
4. Выберите тип переменной Для всех объектов
5. В палитре команд должны появиться пять оранжевых блоков
6. Создайте еще одну переменную - переменная_2.
7. Удалите переменную переменная_2, нажав на 

Задание 2. Мониторы переменных

Научимся управлять внешним видом монитора переменной на сцене.

1. Рядом с блоком переменной переменной поставьте флажок, чтобы показать монитор ее значение на сцене проекта.
2. Нажмите правой кнопкой мыши по монитору переменной, чтобы получить контекстное меню
3. Выберите рычажок, который позволяет изменять значения переменной в самом мониторе, измените значение переменной, перемещая мышью бегунок:

Задание 3. Управление скоростью движения

Создадим скрипт движения спрайта на сцене. Для управления скоростью спрайта используем слайдер монитора переменной.

1. Соберите скрипт (рис.9.). В скрипт мы добавили возможность остановки при нажатии на пробел.



Рис. 9. Скрипт автоматического движения спрайта

2. Создайте переменную.
3. Вставьте в блок движения вместо числовой константы репортер переменной
4. Запустите скрипт на выполнение.
5. Нажмите правой кнопкой мыши на мониторе переменной, чтобы задать возможный диапазон значений переменной.
6. Чтобы упростить регулирование скорости, установите минимальное и максимальное значения переменной.
7. Сохраните проект.

Задание 4. Летучая Мышка

Сделаем проект, в котором нажатием на клавишу пробел мы будем менять значение переменной скорость для летучей мыши. Логические датчики нажатия стрелок клавиатуры используем для изменения направления движения мыши. А также соберем простой скрипт анимации, для которого нам понадобятся два костюма мыши bat1-a и bat1b из папки Animals.

1. Создайте новый проект.
2. Выделите мышью на Объект1 и добавьте для него два костюма bat1-a и bat1-b из папки Animals. Ненужные костюмы удалите.
3. Дайте спрайту имя Мышка.
4. Создайте локальную переменную скорость.
5. Соберите скрипты анимации, управления направлением движения и скоростью (рис.10.)

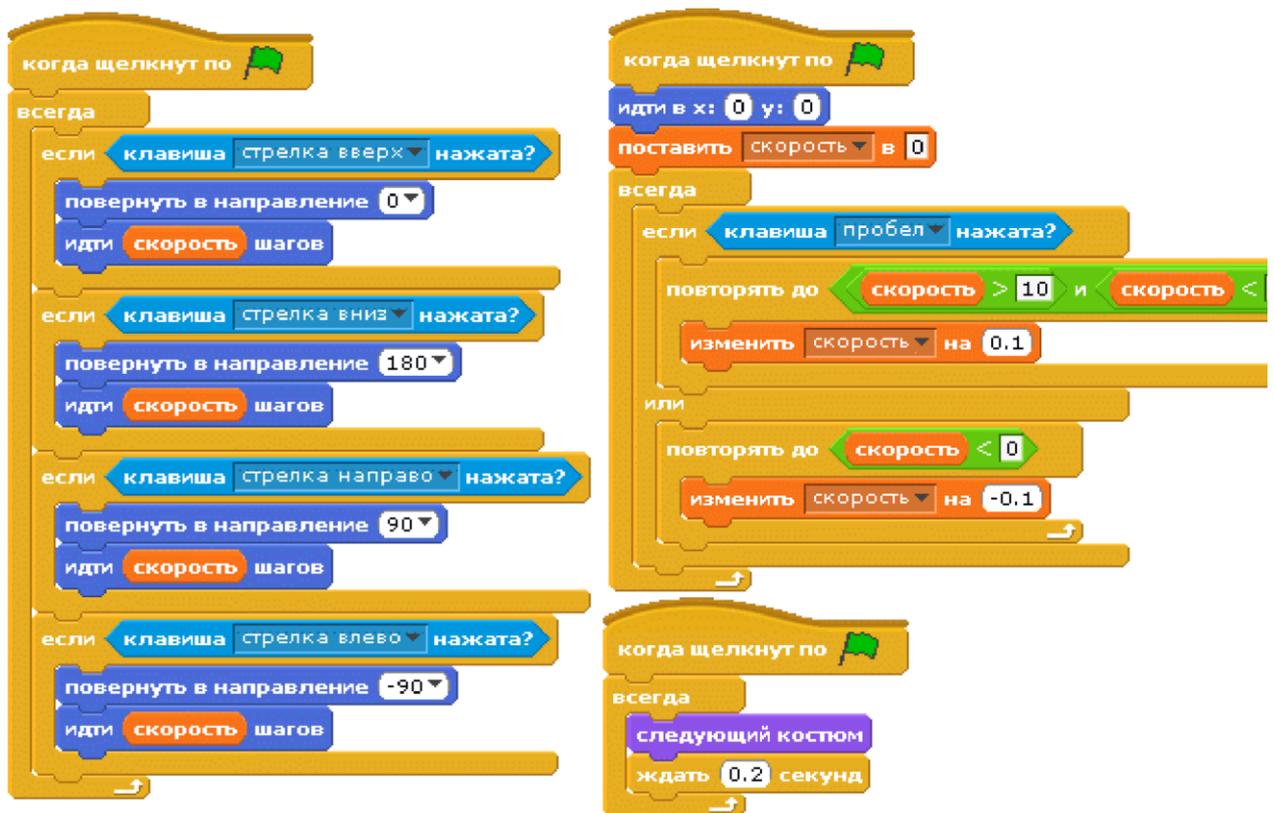


Рис. 10. Скрипты анимации, управления направлением и скоростью движения

6. Проверьте работу скриптов.
7. Сохраните проект.

Практическая работа . Ремикс игры Pong

Компьютерная игра Pong была разработана японской компанией Taito Corporation и выпущена компанией Atari 29 ноября 1972 . Pong является первой компьютерной игрой, получившей широкую известность и на игровых автоматах, и на персональных компьютерах.

Геймплей заключается в том, что игрок горизонтально передвигает ракетку движением мыши, чтобы мяч не упал на землю. В игре есть спрайт мяча - bal, спрайт ракетки - paddle и сцена, у которой один фон red bottom в форме белого прямоугольника с красной линией в нижней части:



Рис. 11. Фон сцены в Pong Прикосновение

мяча к этой линии останавливает игру.

Задание

Создайте проект Pong, добавив в игру переменные.

Как с помощью переменных можно сделать игру интереснее?

Во-первых, можно ввести переменную для регулирования скорости движения мяча.

Во-вторых, создать переменную, которая будет считать время игры до момента пропуска мяча.

В-третьих, создать переменную, которая будет начислять баллы за каждый успешно отбитый мяч.

В-четвертых,... попробуйте придумать сами.

1. Создайте локальную переменную скорость для спрайта ball.
2. Выведите на сцену монитор переменной с рычагом. Теперь игрок сможет мышкой регулировать скорость мяча.
3. Замените в блоках перемещения мяча константу на переменную скорость (рис.12):

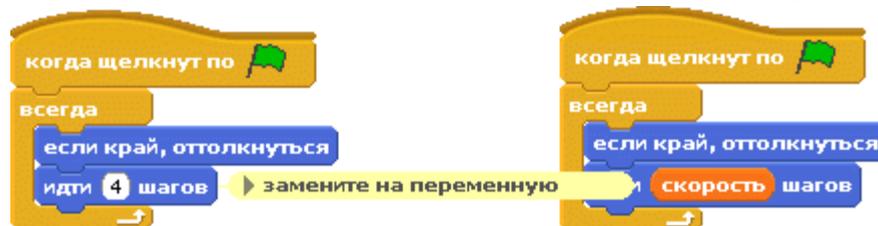


Рис. 12. Изменение скорости движения с помощью переменной

4. Создайте локальную переменную счет, для подсчета количества отбитых мячей, и выведите ее монитор на сцену.
5. Добавьте скрипт, который определяет поведение мяча после прикосновения к ракетке (рис.13) Что изменится, если в этом скрипте после заголовка не поставите блок присвоения переменной нулевого значения?

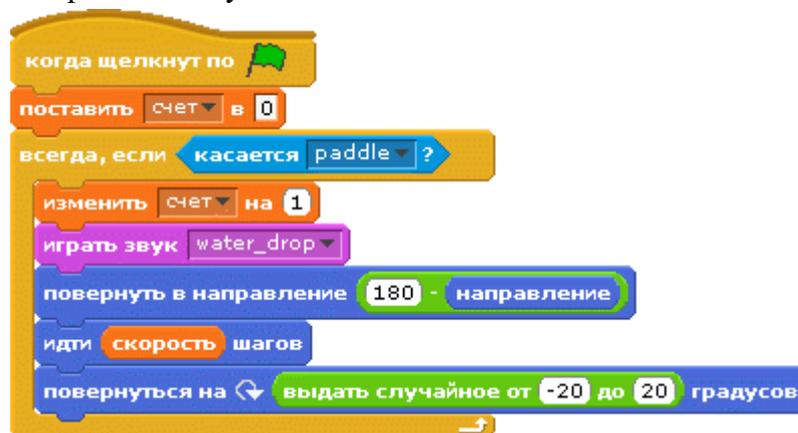


Рис. 13. Начисление очков за отражены мяча

6. Создайте еще две переменные:
 - старт - для сохранения значения таймера на момент старта игры;
 - время - для сохранения значения продолжительности игры как разницы между текущим на момент пропуска мяча значением таймера и его стартовым значением, которое хранится в переменной старт.
7. Добавьте скрипт, который отвечает за остановку игры (рис.14):

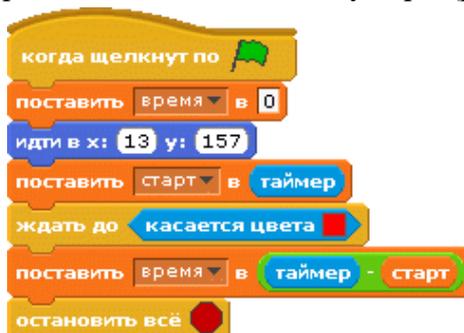


Рис. 14. Подсчет длительности игры 8.

Сохраните созданный ремикс проекта.

Вопросы

1. Сколько здесь дано переменных: скорость, Скорость, быстрота?
2. Можно ли в Скретче отключить таймер?
3. Почему переменным нужны имена?
4. «Не делай другим того, чего не желаешь себе», - говорил Конфуций. Как вы думаете, это моральное правило - константа или переменное значение? Почему так?

Тема «Создание музыки в Scratch»

Задание 1. Влияние темпа на длительность нот. Пронаблюдайте за изменением длительности нот при изменении темпа. Установите максимальную и минимальную величину темпа в Скретч.

1. Соберите скрипт по образцу.

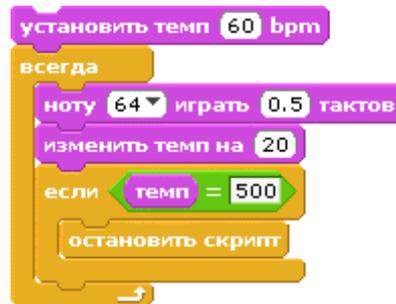


Рис. 37. Программирование темпа

2. Запустите скрипт на выполнение, наблюдая за изменениями длительности звучания нот в каждом шаге цикла при ускорении темпа.
3. Выясните опытным путем, может ли темп быть больше 500 ударов за минуту?
4. Определите минимально возможное значение темпа.
5. Сохраните проект.

Задание 2. Линейный алгоритм гаммы. В музыке гамма - это постепенно восходящее или нисходящее мелодичное движение степеней звукоряда (нот) в пределах какой-либо октавы. Сделаем программу, которая будет проигрывать гамму в пределах первой октавы на основе простого линейного алгоритма:

- 1) установить темп → 2) играть по очереди 12 нот от 60 до 71.
1. Создайте новый проект.
2. Соберите скрипт по образцу.



Рис. 38. Восходящая гамма в темпе 60 ударов на минуту

3. Сохраните проект.

Задание 3. Как по нотам. Исполнитель берет текущую ноту из списка и проигрывает ее. Программа завершается, когда список исчерпан. Чтобы реализовать такой алгоритм, нам понадобится блок цикла с числом повторений, которое равно количеству нот, список нот и список их длины, переменная цикл, которая будет считать уже сыгранные ноты и выбирать новые ноты и их длину (Рис. 39).

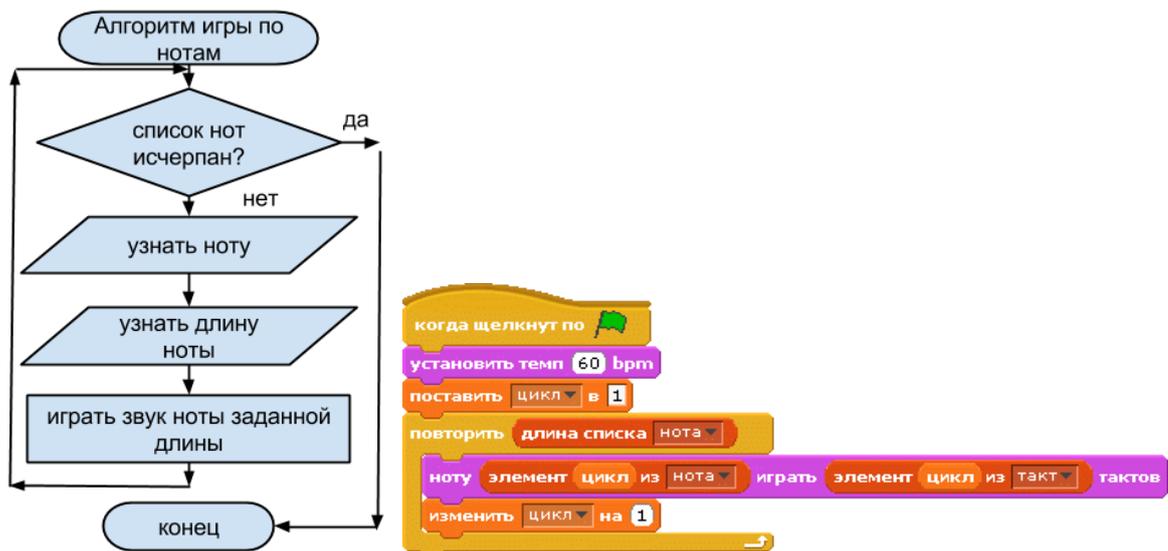


Рис. 39. Скрипт игры по нотам

Поручить выполнение этого алгоритма можно любому объекту, например, сцене.

1. Создайте новый проект.
2. Создайте переменную цикл.
3. Создайте списки нота и такт (Рис. 40), заполните их.

нота		такт	
1	60	1	0.5
2	61	2	0.5
3	62	3	0.5
4	63	4	0.5
5	64	5	0.5
6	65	6	0.5
7	66	7	0.5
8	67	8	0.5
9	68	9	0.5
10	69	10	0.5
11	70	11	0.5
12	71	12	0.5

+ длина: 12 + длина: 12

Рис. 40. Список нот первой октавы и список их длительности

4. Соберите скрипт игры по нотам (Рис. 39) 5. Проверьте работу скрипта.
6. Подумайте, как добавить возможность изменять темп?
7. Сохраните проект.
8. Что случится, если прибавить следующие блоки?



Задание 4. Музыкальный инструмент Пришло время сделать из клавиатуры музыкальный инструмент. Если каждой из 12 нот назначить свою клавишу клавиатуры, мы получим музыкальную клавиатуру, на которой можно играть. Алгоритм очень простой: 1) когда нажата клавиша → 2) звучит ее нота.

Как добавить возможность извлечения звуков разных октав? Можно закрепить звуки других октав за другими клавишами или прибавить переключатели октав, например, при нажатии на числовую клавишу будут звучать ноты закрепленной октавы: 1 - субконтроктавы, 2 - контроктавы, 3 - большой октавы, 4 - малой октавы, 5 - первой, 6 - второй, 7 - третьей, 8 - четвертой и 9 - пятой. Для этого понадобится переменная номера октавы, которую мы назовем #, и список значений октава. Первая октава станет точкой отсчета, потому ее значение равняется 0. Октавам слева присвоим отрицательные значения, а справа - положительные (Рис. 41).

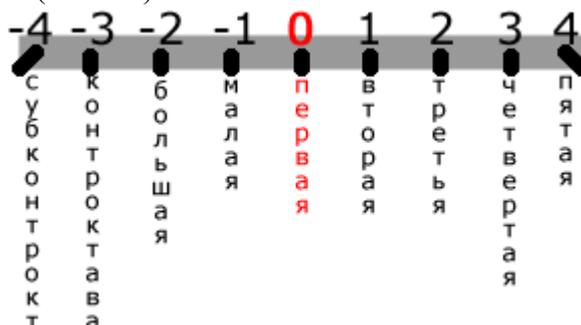


Рис. 41. Числовая ось октав

Так как между одинаковыми нотами соседних октав всегда 12 единиц, используя одну простую формулу, мы сможем вычислить ноту любой октавы: нота X октавы N = нота X 1-

й октавы + (12 * значение октавы N) Например, если мы хотим сыграть ноту ДО большой октавы, нужно к 60 (ее значение в первой октаве) прибавить произведение $12 * 2 = 24$. Так как прибавление к положительному числу (60) отрицательного числа (- 24) приводит к вычитанию, мы получим число 36, это и есть нота ДО большой октаве. В

Скретче эта формула выглядит так:

1. Откройте проект Как по нотам.
2. Создайте список девяти значений октав с именем октава (рис.15.9).



Рис. 42. Список значений октав

3. Создайте переменную с именем # для выбора элементов списка октава.
4. Переключение октав поручим сцене. Соберите для сцены девять скриптов для выбора октав после нажатия на числовую клавишу (Рис. 43)



Рис. 43. Скрипты выбора октавы после нажатия на числовую клавишу клавиатуры

5. Сделайте 12 спрайтов для 12 нот, например, так:



Рис. 44. Образы спрайтов нот на сцене

Изображения нот можно для наглядности расположить на сцене в порядке роста высоты звуков (Рис. 44). 6. Используя схему клавиатуры (Рис. 45), соберите скрипты (рис.46) музыкальных клавиш для каждой ноты.

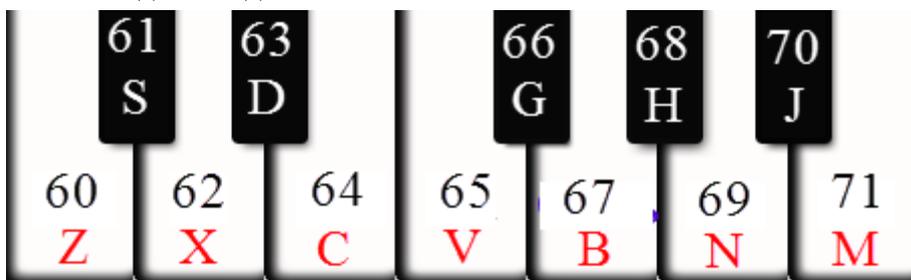


Рис. 45. Модель виртуальной первой октавы

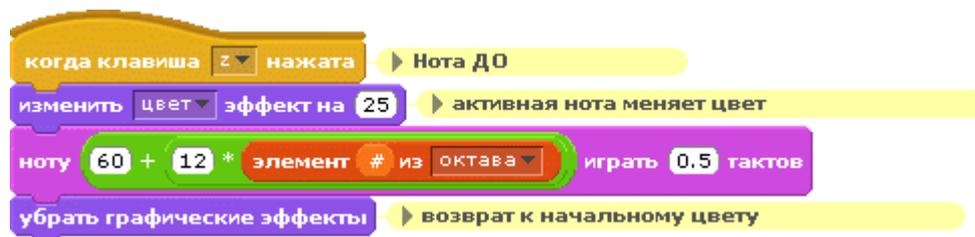


Рис. 46. Скрипт музыкальной клавиши для ноты ДО (60)

7. Проверьте работу скрипта - при нажатии на запрограммированные клавиши должны звучать ноты. Поиграйте на музыкальной клавиатуре в разных октавах.
8. Сохраните проект в файле упражнение_15_4.

Задание 5. Композитор. Наша программа уже умеет проигрывать любую мелодию. Достаточно изменить списки, чтобы изменилась мелодия. Упростим процесс написания музыки, добавив возможность записывать ноты в список после нажатия на клавишу нот в любой октаве. Введем переменную record и кнопку для ее включения: если значение record=0, нажатие на клавишу не будет храниться в списке нот, если record =1 - будет храниться.

Также сделаем кнопки для очистки списков, начала записи, завершения записи и запуска игры мелодии. В музыке используются ноты разной длины. По умолчанию длина нот равняется 0,5 такта - это 1/8 длины целой ноты. Создадим переменную час для определения длины нот и запрограммируем клавишу ↑ на увеличение длительности ноты на 0,2, а клавишу ↓ - на уменьшение на - 0,2.

1. Откройте проект Музыкальный инструмент.
2. Создайте глобальные переменные record и час.
3. Создайте четыре спрайта для кнопок, например, так:



4. Соберите скрипты для кнопок (Рис. 47).



Рис. 47. Скрипты для четырех кнопок

5. Добавьте в скрипты музыкальных клавиш (Рис. 15.13.) блоки команд записи нот, если переменная record = 1 (рис. 48).



Рис. 48. Скрипт игры записи ноты

6. Поиграйте на своем музыкальном инструменте. Что происходит при нажатии на несколько клавиш одновременно?
7. Попробуйте сочинить мелодию.
8. Сохраните проект.
9. Добавьте выбор инструмента 41 - скрипка. Прослушайте исполнение мелодии этим инструментом.
10. Проверьте, как звучит мелодия в исполнении разных инструментов.

Задание 6. Оркестр Чтобы параллельно исполнить мелодию разными инструментами, необходимо скрипты для разных инструментов размещать в разных спрайтах. Сделаем спрайт для выполнения мелодий скрипкой (инструмент 41) и спрайт для мелодий контрабаса (инструмент 44). Они будут играть одну мелодию одновременно, но в разных октавах.

1. Откройте проект Композитор
2. Создайте спрайт скрипка.
3. Копируйте в спрайт скрипка скрипт игры по нотам.
4. Удалите из копии скрипта блоки изменения значения переменной цикл. Что случится, если этого не сделать?
5. Добавьте после заголовка скрипта выбор инструмента 41 (Рис. 49).
6. Чтобы инструмент играл ноты на две октавы выше, добавьте оператор добавления к ноте числа 24 (Рис. 49).

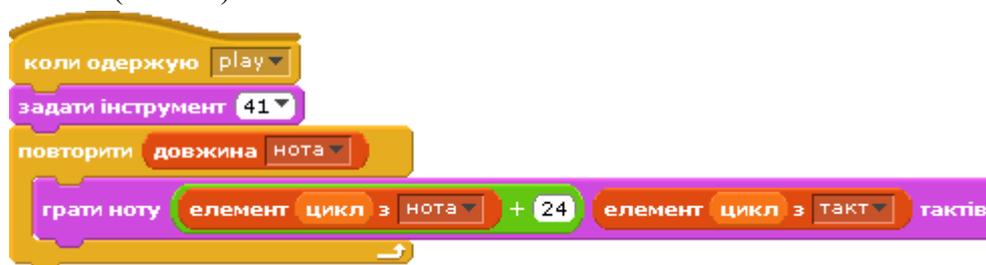


Рис. 49. Скрипт игры нот скрипкой на две октавы выше

7. Создайте спрайт контрабас.
8. Копируйте в спрайт контрабас скрипт игры по нотам спрайта скрипка.
9. Измените инструмент на 44.
10. Чтобы инструмент играл ноты на октаву ниже, используем оператор вычитания из ноты числа 12 (Рис. 50).

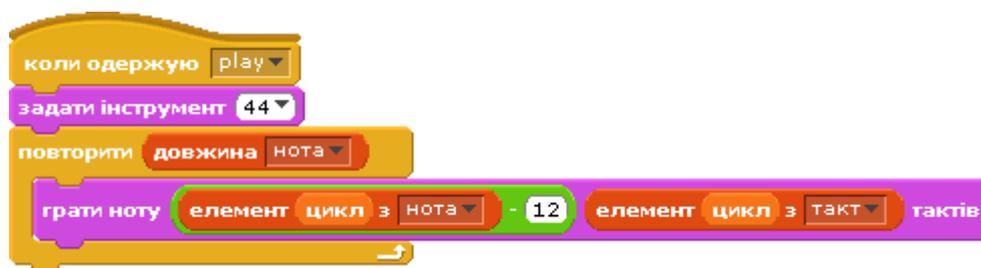


Рис. 50 Скрипт игры нот контрабасом на одну октаву ниже

11. Запустите кнопкой play одновременное выполнение всех мелодий.
12. Используйте блоки громкости звуков для разных инструментов.
13. Сохраните проект. **Практическая работа . Программирование музыки/**

1. "Чижик-пыжик"

Составить программу проигрывания мелодии "Чижик-пыжик".



2. Аккорды

В музыке часто одновременно звучат не только разные инструменты, но и несколько нот, которые играют одним инструментом. Одновременное гармоничное звучание трех или больше нот одним инструментом называют аккордом. Для любой ноты можно подобрать еще как минимум 2 ноты, которые будут вместе хорошо звучать: аккорд = нота1 + нота2 + нота3.

Игру аккордов можно поручить любому объекту, создав три скрипта, которые будут запускаться одновременно в ответ на сообщение. Таблица показывает числовые значения нот аккордов для первой октавы. Установите закономерности, которые лежат в основе размещения нот.

Ступень октавы	Мажорный лад			Минорный лад		
	Нота 1	Нота 2	Нота 3	Нота 1	Нота 2	Нота 3
1	60	64	67	60	63	67
2	61	65	68	61	64	68
3	62	66	69	62	65	69
4	63	67	70	63	66	70
5	64	68	71	64	67	71
6	65	69	72	65	68	72
7	66	70	73	66	69	73
8	67	71	74	67	70	74
9	68	72	75	68	71	75
10	69	73	76	69	72	76
11	70	74	77	70	73	77
12	71	75	78	71	74	78

Чтобы услышать звучание аккордов, сделайте музыкальную клавиатуру для игры аккордами. Используя закономерности размещения нот в аккордах, можно назначить

любой клавише любой аккорд. Вам понадобится одна переменная - `нота1`. Пример показывает, как можно запрограммировать игру аккорда на первой ступени мажорного строя (Рис. 51).



Рис. 51 Скрипты игры аккорда на 1 ступени мажорного строя

Для программирования следующего аккорда достаточно добавить копию скрипта (верхний скрипт рис. 51), ответственного за нажатие клавиши, изменив клавишу клавиатуры и значение ноты.

Также можно добавить игру аккордами любых мелодий по спискам нот. В этом случае понадобится переменная цикл, список нота для хранения записи мелодии, а также скриптдирижер (Рис. 52) :

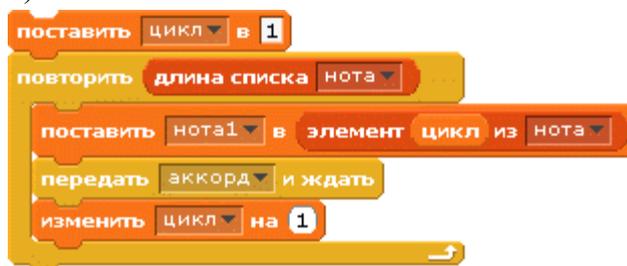


Рис. 52. Игра мелодий аккордами

Вопросы

1. Какое физическое устройство внутри компьютера отвечает за создание звуков?
2. Какая нота установлена в Скретче по умолчанию?
3. Какой темп установлен в Скретче по умолчанию?
4. Чему по умолчанию равна величина такта?
5. Чему по умолчанию равна величина паузы?

Тема «Создание игры по заданному сценарию»

Проект. Разработка игры по заданному сюжету.

Сюжет игры: герой Соник двигается в лабиринте произвольного вида. Управление – с помощью четырех клавиш. По полю случайным образом двигаются несколько монстров. Количество монстров зависит от уровня сложности. Легкий уровень: 2 монстра, средний – 4 монстра, высокий – 6 монстров.

Если Соник сталкивается с перегородкой лабиринта, он замирает и ждет 3 секунды. Если сталкивается с монстром – погибает. Игра завершается если Соник достигает финиша, или сталкивается с монстром.