Муниципальное бюджетное общеобразовательное учреждение
«лицей «Созвездие» № 131» городского округа Самара
443083, г. Самара, ул. Промышленности,319, тел.: 261-14-40, 261-14-90.
E-mail: school131@bk.ru

ОБЛАСТНАЯ НАУЧНО-ФАНТАСТИЧЕСКАЯ КОНФЕРЕНЦИЯ ШКОЛЬНИКОВ И СТУДЕНТОВ «НАМ ЖИТЬ В БУДУЩЕМ»

Тема научной работы:

«Обучение младших школьников основам алгоритмизации и программированию на базе Scratch»

Выполнил: Антонов Андрей Вячеславович,
класс 10 «А»

Руководитель: Макарова Ирина Сергеевна,
доцент кафедры ПМИИС
СамГУПС

Самара, 2019

ВВЕДЕНИЕ

Игра занимает значительное место в процессе обучения школьников начальных классов, т.к. вызывает у детей интерес, хорошее настроение, облегчает усвоение учебного материала. В ходе игры учащиеся стремятся четко выполнять задания, соблюдать правила игры, быть находчивыми и сообразительными. Использование игровых методик при обучении повышает мотивацию, позволяет легче усваивать тот материал, без которого нельзя участвовать в игре.

Основы алгоритмизации и программирования – одна из самых сложных тем информатики. Однако существует тенденция сокращения этого курса в современном содержании предмета «Информатика и ИКТ» в школе [1]. Многие школьники считают, что тема «Алгоритмизация и программирование» является сложной и неинтересной. Сложность заключается в том, что при изучении программирования приходится решать различные математические задачи, результаты которых далеки от наглядности.

Тем не менее, важность раздела «Алгоритмизация и программирование» является очевидной: алгоритмизация помогает при решении задач самого разного характера, где требуется умение анализировать и выбирать оптимальный вариант решения; алгоритмизация способствует развитию абстрактного и логического мышления, которые позволяют выявлять общие закономерности в тех или иных явлениях окружающего мира. Решение любой задачи проходит более успешно при наличии осознанного алгоритма действий, приводящих к результату.

Стимулировать познавательный интерес школьников к разделу «Алгоритмизация и программирование» поможет игровая форма преподнесения учебного материала, которая даёт хороший эффект в изучении навыков программирования. Игры вызывают у школьников устойчивый познавательный интерес. Чтобы наглядно показать приемы

алгоритмизации и программирования можно разработать компьютерную игру, выполняя которую учащиеся будут поэтапно изучать приемы программирования. Использование именно компьютерного варианта имеет свои преимущества, т.к. компьютер для младших школьников вполне привычная вещь.

Цель: разработка компьютерной игры для школьников младших классов, которая обучает в игровой форме основам алгоритмизации.

Задачи:

- Изучить возможности языка Scratch.
- Разработать сценарий игры, позволяющий младшим школьникам познакомиться с основами алгоритмизации и программирования.
- Разработать компьютерную игру.

Гипотеза: Использование компьютерной игры поможет учащимся начальной школы освоить азы алгоритмизации и программирования в увлекательной форме.

§1. SCRATCH – ИГРОВАЯ ОСНОВА ДЛЯ ЗНАКОМСТВА С ОСНОВАМИ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Scratch создан как продолжение идей языка Лого и конструктора Лего небольшой командой программистов в Массачусетском технологическом институте и представляет собойобразовательный язык программирования с графическим интерфейсом, который делает программирование простым и интуитивно понятным [2]. Цель создания языка Scratch заключалась в том, чтобы дети, укоторых нет опыта программирования, могли изучить основные принципы императивного, объектно-ориентированного и многопоточного программирования. Если говорить кратко, то Scratch– это визуальная среда программирования, где в отличие от других языков программирования код пишется не вручную, а собирается из блоков.

Главное окно среды разработки Scratch разделено на несколько частей, сгруппированных в три колонки.

Левая колонка содержит палитру блоков. Блок – это минимальный фрагмент программы в Scratch: переменная, оператор, функция или управляющая структура. Блоки сгруппированы в следующие категории:

- Motion (управление движением спрайтов);
- Looks (графические эффекты и функции вывода think и say);
- Sound (звуковые эффекты);
- Pen (спрайтовый эквивалент черепашьей графики);
- Control (управляющие структуры);
- Sensing (обработка сигналов от мыши, клавиатуры и сенсора);
- Operators (математические, логические и строковые функции и операторы);
- Variables (скалярные и векторные переменные и операции над ними).

Центральная колонка содержит текущий спрайт (персонаж) и его скрипты. Спрайт – это объект в стиле Scratch, ассоциированный с

изображением и набором переменных и скриптов, которые определяют его поведение. Скрипты создаются методом соединения отдельных блоков – либо последовательно, либо размещая блок в определенном месте другого блока (управляющей структуры, функции и т.д.). Один спрайт может обладать несколькими скриптами, которые запускаются независимо — действием пользователя (нажатием клавиши или кнопки мыши), таймером или получением сообщения от другого спрайта.

Правая колонка содержит сцену и список спрайтов. Сцена – область, в которой действуют спрайты при выполнении программы. Она содержит изображения видимых спрайтов, видимые переменные и любые сообщения и графические эффекты, созданные при выполнении программы.

Язык Scratch иллюстрирует несколько важных парадигм:

- объектно-ориентированная: каждый спрайт на самом деле является объектом со своими свойствами (переменными) и поведением (скриптами), и разные объекты могут взаимодействовать.
- структурная (в низкоуровневом понимании): все программы конструируются из ограниченного набора элементов (блоков).
- многопоточная: объекты взаимодействуют посредством обмена сообщениями через блоки broadcast и when I receive.

Кроме того, Scratch предоставляет мощный набор мультимедийных инструментов: стандартные блоки предоставляют множество графических и звуковых функций, а также возможности обработки сигналов с клавиатуры, мыши и сенсора.

У языка Scratch есть несколько аналогов, которые также используются при обучении программированию, например “Черепаха”(LOGO) и “Кукарача”.

Исполнитель “Черепаха” представляет собой среду, в которой ученик должен написать программу, исполняя которую черепаха чертит тот или

иной рисунок при помощи пера.Черепаха может перемещаться по ограниченному полю, при этом она может поднимать и опускать перо, в зависимости от этого она оставляет и не оставляет за собой след. Минусы данного исполнителя в ограниченности сценариев действия черепахи. Также существенным минусом необходимость набора программы вручную.

Кукарача – исполнитель, работающий на клетчатом поле[3]. Кукарача может переползать из клетки в клетку вверх, вниз, влево, вправо, но не по диагонали. Выход исполнителя за пределы поля запрещён. В каждой клетке может находиться кубик с нанесённым на его грань символом. Кукарача, перемещаясь по полю, может толкать один или несколько кубиков перед собой и «сбрасывать» кубики за пределы поля. Некоторые кубики на поле могут иметь «скрытые» символы, обозначаемые знаком вопроса. Считается, что такие кубики расположены в клетке «символом вниз». Толкая скрытый кубик, Кукарача перемещает его в следующую клетку по ходу движения, переворачивает и видит надпись. Исполнитель в состоянии прочитать символ на кубике независимо от того, была ли запись символа обычной или скрытой.Главным преимуществом Scratchперед исполнителем Кукарача является наличие в Scratch возможности менять сцену по ходу работы программы. Тем самым можно создать последовательность сцен, связанную общей идеей, например для обучения. Для создания такой же последовательности сцен в исполнителе Кукарача нужно было бы постоянно менять сцены перезапуском программы, что существенно усложняет обучение в нашем случае.

Также одним из главных преимуществScratch перед другими исполнителями является его интуитивно понятный красочный графический интерфейс, который улучшает восприятие детьми и подростками основ алгоритмизации.

§2. РАЗРАБОТКА КОМПЬЮТЕРНОЙ ИГРЫ

2.1. Сценарий игры

Так как игра по жанру принадлежит к аркадному и логическому видам, то в основу игры положен принцип, согласно которому в каждой сцене ученик изучает один из видов алгоритмов, а сложность каждого уровня будет обусловлена составлением наиболее эффективного алгоритма действия персонажа.

Если рассмотреть алгоритмы, необходимые к изучению школьником(линейный, ветвление, циклический), то можно разделить их по степеням сложности:Линейный -> Ветвление -> Циклический. Таким образом, можно составить последовательность сцен в соответствии со сложностью освоения каждого алгоритма от лёгкого уровня к сложному.

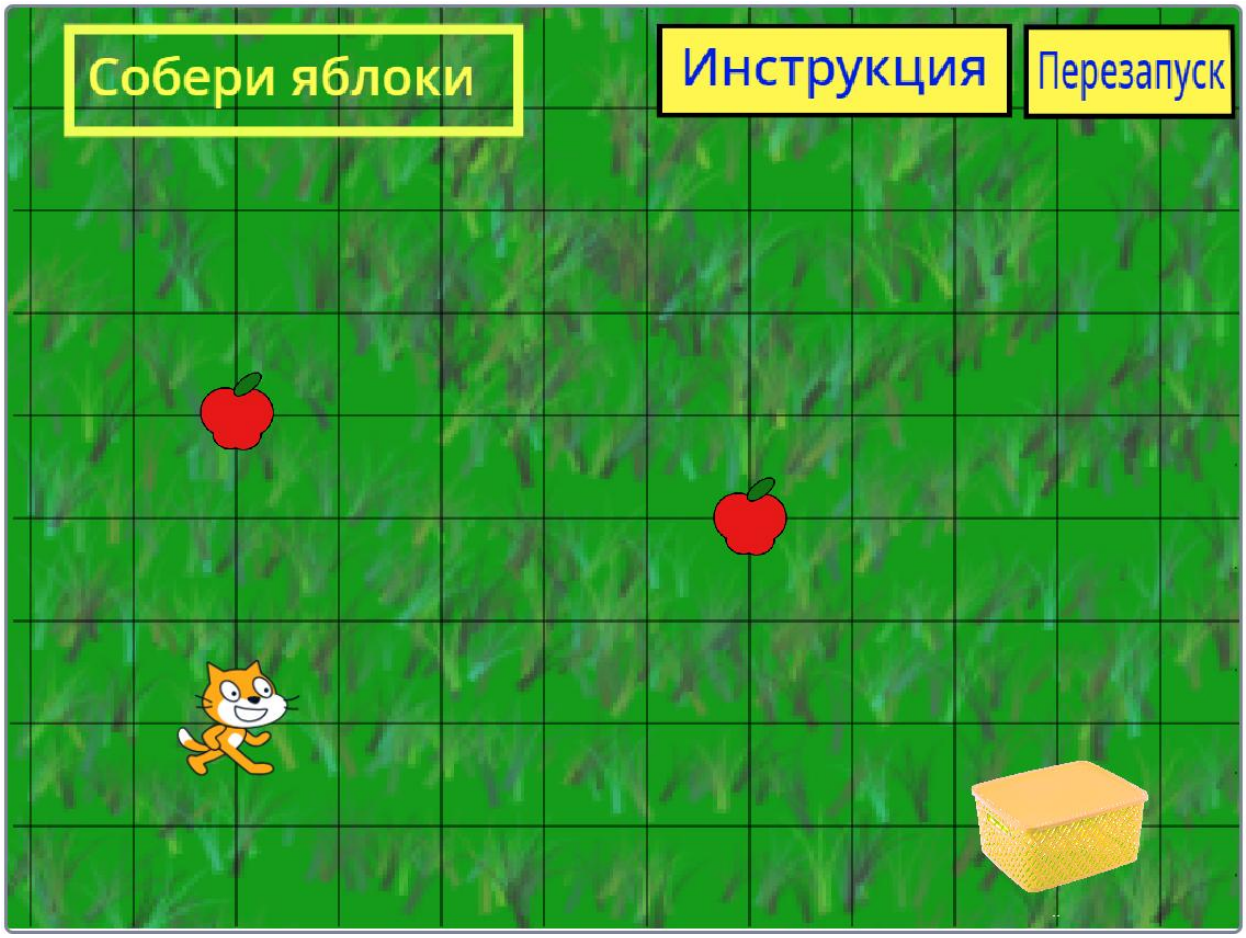
В каждой из сцен игрок будет пользоваться инструментами “Шаг”, “Поворот направо” и “Поворот налево”. Для удобства изучения языка каждая сцена размечена сеткой, в узлах которой размещены предметы, задействованные в сцене, и по узлам которой может перемещаться персонаж. Один шаг персонажа равен расстоянию между соседними узлами сетки по вертикали или по горизонтали. Изначально направление движения персонажа задано вверх по игровому полю. При помощи инструментов “Поворот налево” и “Поворот направо” игрок может изменить направление своего движения, что добавляет некоторой сложности в составления алгоритма.

Также на игровом поле присутствуют две кнопки “Инструкция” и “Перезапуск”. Кнопка “Инструкция” открывает меню, в котором ученик может прочитать информацию об алгоритмических конструкциях для того, чтобы составить алгоритм прохождения сцены самостоятельно. Кнопка “Перезагрузка” перезапускает игру, и игрок начинает проходить заново все сцены.

2.2. Первая сцена

Для изучения линейного алгоритма в форме игры подойдёт простая структура вида “Дойди из пункта А в пункт Б”. Задача усложнена за счет добавления дополнительных пунктов, куда обязательно должен попасть персонаж, и поворотов, которые необходимы, чтобы изменить направление движения. Для этого уровня выбран простой сценарий: игрок должен составить алгоритм, по которому персонаж на экране(Кот) соберёт яблоки и донесёт их до корзины.

В начале сцены Кот говорит игроку: “Помоги мне собрать яблоки. Для этого используй информацию из пунктов "Шаг" и "Поворот"”. После чего ученик должен зайти в раздел инструкция и прочитать параграфы “Шаг” и “Поворот”, в которых изложен порядок создания алгоритма для перемещения персонажа по игровому полю. Перемещение игрока осуществляется при помощи трёх инструментов “Шаг”, “Поворот направо” и “Поворот налево”. “Шаг”- инструмент, перемещающий персонажа в заданном направлении по игровой сетке. Один шаг равен длине одного звена сетки. “Поворот направо” – инструмент, изменяющий направление следующих перемещений персонажа по игровой сетке на 90° по часовой стрелке. “Поворот налево” – инструмент, схожий с “Поворот налево”, но изменяющий направление на 90° против часовой стрелки.

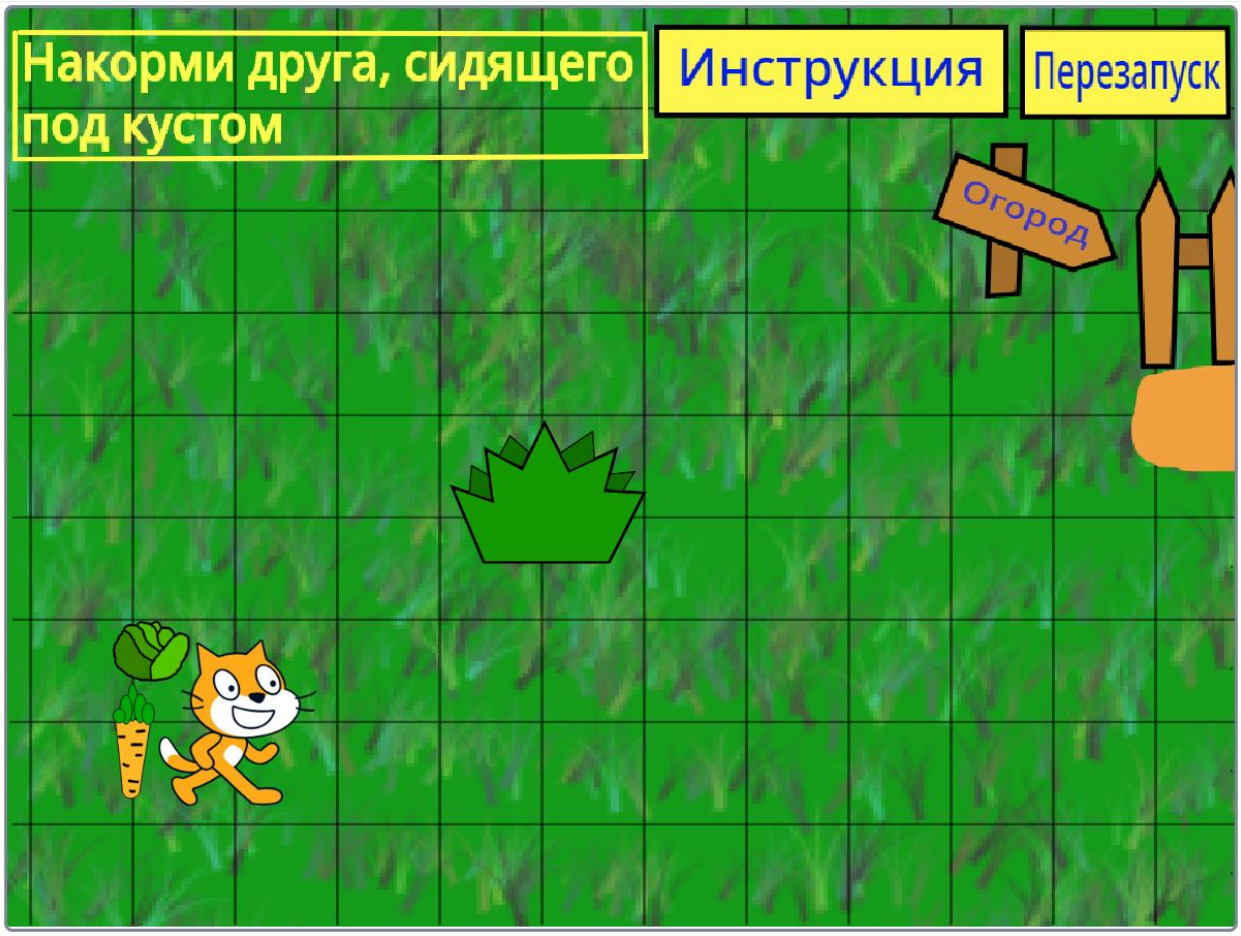


2.3 Вторая сцена

Для изучения разветвляющегося алгоритма была выбрана ситуация, когда ученик должен написать алгоритм, согласно которому персонаж должен дойти до куста и накормить сидящего в ней зверька: зайца – яблоком, овечку - капустой. В результате прохождения сцены игрок должен попасть в огород.

Вначале Кот говорит игроку:”Помоги мне накормить друга, который спрятался под кустом, но я не знаю, кто это. После помоги мне попасть в огород. Для этого тебе нужно изучить пункт "Условия"”. После чего ученик должен изучить параграф “Разветвляющий алгоритм”, в котором изложен порядок составления алгоритма с использованием условного оператора. Ученику предоставляется новый инструмент “Если это (1), то (2), иначе (3)”. Этот инструмент принимает три параметра:(1) – условие, при котором выполняется действие (2); им является название зверька сидящего под

кустом; (2) – действие, которое выполняется при верности условия (1), (3) – действие, которое выполняется при ложности условия (1). Условия: “Заяц”, “Овечка”. Действия: “дать капусту”, “дать яблоко”. В случае если по выполнению алгоритма зверёк будет накормлен не той пищей, сцена считается не пройденной. Программа случайным образом определяет сидящего в кусте зверька.



2.4 Третья сцена

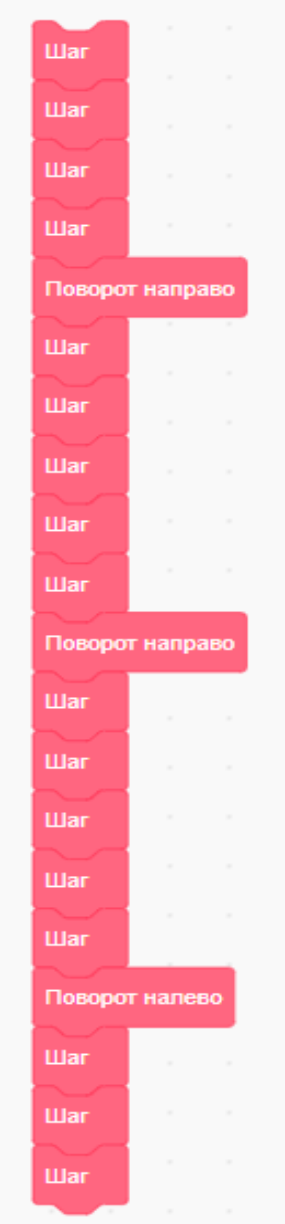
Для изучения циклического алгоритма была выбрана ситуация, когда игроку предлагается собрать 4 кочана капусты и разделить их на две корзины: со свежими кочанами и испорченными кочанами.

В начале сценыКот говорит игроку:”Помоги мне собрать новую капусту. Но попробуй сделать так, чтобы ты использовал как можно меньше команд. Для этого тебе нужно изучить пункт "Циклы" и повторить пункт "Условия"”. После чего ученик должен изучить параграф “Цикл”, в котором

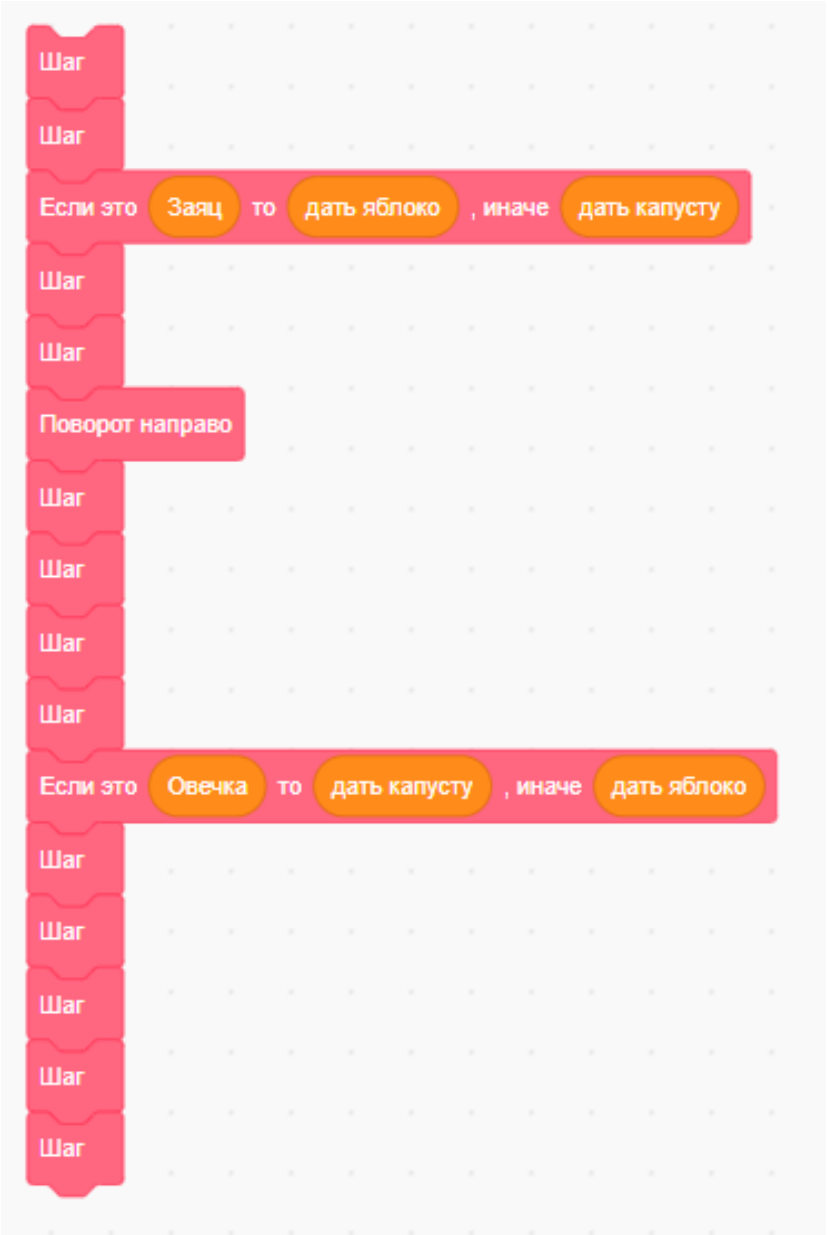
изложен порядок составления циклического алгоритма. Ученику предлагается использовать новый инструмент “Осмотреть кочан, если он (1), то (2)”. Этот инструмент принимает два параметра: (1) – состояние кочана капусты (свежий или испорченный), и (2) – действие, выполняемое при выполнении условия (положить в корзину 1 или положить в корзину 2). Программа случайным образом определяет какой кочан свежий, а какой –нет. В том случае, когда в одной корзине окажутся и свежие, и испорченные кочаны, сцена считается не пройденной.

2.5 Примеры правильно составленных алгоритмов к каждой сцене

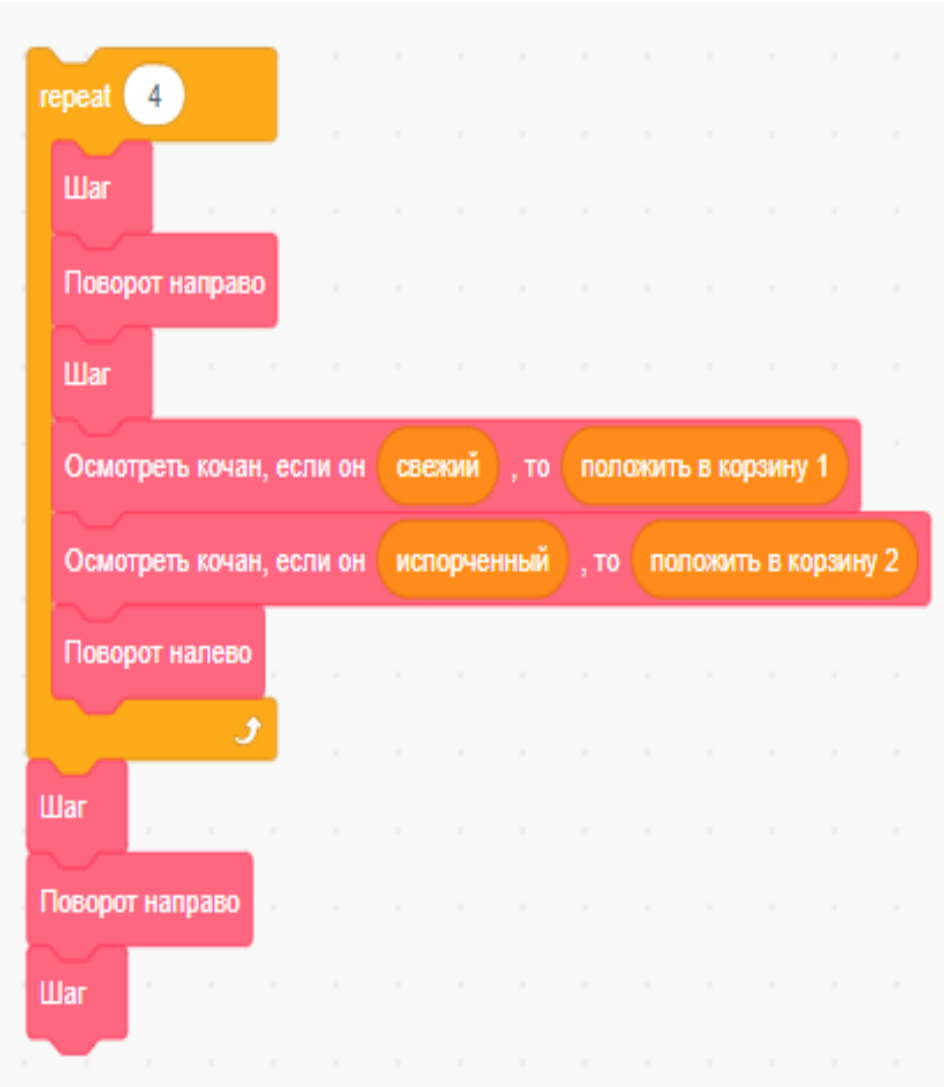
Сцена 1



Сцена 2



Сцена 3


repeat 4
Шаг
Поворот направо
Шаг
Осмотреть кочан, если он свежий , то положить в корзину 1
Осмотреть кочан, если он испорченный , то положить в корзину 2
Поворот налево
Шаг
Поворот направо
Шаг

ЗАКЛЮЧЕНИЕ

Знакомство с языком Scratch показало, что его можно использовать для создания увлекательных сцен и даже целых квестов. Язык обладает широким спектром возможностей по сравнению с другими исполнителями, особенно тем преимуществом, что команды не нужно набирать вручную, программа собирается как конструктор из отдельных блоков.

Разработанная в процессе исследовательской работы компьютерная программа позволит младшим школьникам познакомиться с основными алгоритмическими конструкциями в форме увлекательной игры. Игровая форма изучения материала, наглядность представления стандартных алгоритмов с одной стороны вызывает у детей интерес, а с другой стороны помогает освоить учебный материал. Таким образом, цель исследования достигнута, поставленные задачи выполнены.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. МуртузалиеваА.С., ГаджиевТ.С. О значимости изучения алгоритмизации и программирования в школьном курсе информатики // Вестник социально-педагогического института. №2(14), 2015. С.54-57.
2. Свободная энциклопедия языков программирования. Язык программирования Scratch [Электронный ресурс] //http://progopedia.ru/language/scratch/
3. Роботландский университет, А. А. Дуванов, П. А. Дуванов. Язык программирования Кукарачи [Электронный ресурс] //http://robotlandia.ru/abc4/executor/coc/help/index.htm