

Министерство образования Республики Саха (Якутия)  
Муниципальное бюджетное учреждение дополнительного образования  
«Детский технопарк «КубИТ»  
МО «Намский улус» Республики Саха (Якутия)»

**Использование геймификации  
в кружке «Увлекательное программирование»**

Выполнила: Татаринова Е.А.- педагог ДО  
МБУ ДО ««Детский технопарк «КубИТ»»

2020 г.

## Оглавление

Введение.....	3
1. Использование геймификации при обучении школьников основам программирования.....	6
2. Среды обучения программированию.....	7
2.1. Онлайн среда Code.org.....	8
2.2. Язык программирования Scratch (Скретч).....	14
Заключение.....	23
Использованная литература.....	24

## **Введение**

Информатика – важнейший компонент образования, который формирует целостное мировоззрение системно-информационной картины мира, учебных и коммуникативных навыков детей. В настоящий момент компьютерные технологии широко внедряются в образовательный процесс.

Непосредственное знакомство современного ребенка с компьютером начинается уже с дошкольного возраста. Плюсам и минусам этого раннего взаимодействия посвящено достаточное большое количество статей и работ. В любом случае, на сегодняшний день полностью изолировать ребенка от компьютера - лишено смысла, а вот направить детский интерес в нужное русло - приоритетная задача для родителей и педагогов.

**Актуальность.** В последние десятилетия происходит стремительная информатизация всех сторон жизни общества и всех сфер производственной деятельности. Состояние перехода к информационному обществу, отраженное в законе РФ «Об образовании», и новые ФГОС ставят перед системой обучения информатике новые цели, среди которых:

- наличие социального заказа на формирование личности с высоким уровнем мышления: операционного, алгоритмического, системного;
- пониманием внутреннего устройства сложных информационных процессов и систем и навыками их декомпозиции;
- способностью творческого преобразования реальности;
- получение навыков продуктивного и эффективного использования компьютерной техники;
- получение опыта созидательной деятельности как условия самореализации в жизни.

Согласно Федеральному образовательному стандарту начального общего образования важными целями математического образования являются:

- формирование у обучающихся основ логического и алгоритмического мышления, умений записи и выполнения алгоритмов;

- умений действовать в соответствии с алгоритмом и строить простейшие алгоритмы;
- исследовать, распознавать и изображать геометрические фигуры;
- работать со схемами, представлять, анализировать и интерпретировать данные.

Выпускникам школ необходимо обладать достаточными знаниями и навыками для эффективного использования современных информационных технологий в своей дальнейшей деятельности.

Одним из важных целей обучения программированию является решение новой воспитательной задачи: уменьшение непродуктивного самостоятельного использования компьютера, перенаправление внимания учащихся от задач, не связанных с обучением и развитием, на продуктивное и эффективное использование компьютера, мобильных устройств, информационных и телекоммуникационных технологий.

Практика показывает, что существуют динамические компьютерные игры (то есть игры, насыщенные движением, действием, изменением объектов и их свойств с течением времени), которые школьники могут разрабатывать в процессе обучения программированию. Существует активный интерес школьников к такой деятельности. Разработка динамических игр, несложная с точки зрения программирования, может внести серьезный вклад в повышение мотивации к учению, преодоление когнитивных затруднений, интеллектуальное развитие школьников.

Вопросы обучения детей основам программирования в научной и научно-методической литературе обсуждаются учеными, учителями и специалистами в области информационных технологий. Рассматривая программирование, как общение с компьютером на языке понятном ему, исследователи поддержали идею обучения детей программированию с раннего возраста. На первый взгляд, решение просто: данный вопрос изучается на школьном предмете «Информатика». Однако школьный учитель в рамках темы «Алгоритмы» ограничен количеством учебных часов,

предусмотренных образовательной программой. На помощь приходит дополнительное образование. Не случайно в Федеральном государственном образовательном стандарте начального образования внешкольному образованию детей уделяется большое внимание.

**Цель** – геймификация обучения программированию с помощью современных языков и сред программирования, позволяющих обучающимся освоить программные конструкции, предусмотренные ФГОС, и эффективно использовать полученные знания в ходе дальнейшего обучения и выборе профессии.

**Объект исследования** - процесс обучения школьников программированию.

**Предмет исследования** - геймификация обучения программированию.

**Гипотеза** заключается в том, что геймификация обучения программированию позволит повысить эффективность обучения информатике и будет способствовать применению знаний из области программирования при решении различных учебных задач за счет развития познавательного потенциала и мотивации учащихся, реализации личностно ориентированного и деятельностного подходов в обучении (в соответствии с требованиями ФГОС).\

Цель, предмет и гипотеза определили постановку и необходимость решения следующих **задач**:

1. Изучить и проанализировать теоретические и методические основы обучения школьников программированию;
2. Конкретизировать основные условия и критерии отбора содержания обучения программированию, разработки этапов и средств создания компьютерных игр для обучения программированию;
3. Разработать модель обучения, на основе которой определить содержание и методы геймификации обучения программированию, предусматривающих развитие творческого потенциала и мотивации

обучающихся, способствующих реализации личностно ориентированного и деятельностного подходов к обучению;

## **1. Использование геймификации при обучении школьников основам программирования**

В нашем учреждении МБУ ДО «Детский технопарк «КубИТ» МО «Намский улус» РС(Я)» с. Намцы вот уже 3 год есть кружок «Увлекательное программирование» для детей младшего школьного возраста, в рамках которого я обучаю детей основам программирования.

При обучении раннего программирования возникает проблема: учащиеся начальных классов не могут запоминать сложные команды, длинные коды, написанные, как правило, на иностранном языке (который они только начинают изучать). Для решения данного противоречия необходим язык программирования близкий к образу мышления детей, содержащий команды для работы с интересными и понятными для них объектами, но в то же время, дающий прочную основу для изучения других языков программирования.

В 80-е годы прошлого века одни из основоположников теории искусственного интеллекта, создатели языка «Logo» Сеймур Пейперт и Алан Кей указывали, что средства, которые существенно меняют способы мышления, должны быть доступны ребенку как можно раньше.

С осени 2014 года основы программирования начали изучать в школах Великобритании. Учащиеся начальных классов британских школ с помощью таких программных обеспечений, как MIT's Scratch, Kodu, Logo учатся создавать простые программы по блокам, а в одиннадцать лет учащиеся должны иметь представление о базовых алгоритмических структурах и использовать их при создании учебных программ. Финский проект «Koodi 2016», учебные планы ряда таких стран, как Южная Корея, Эстония, Франция, Австралия также предполагают обучение детей основам программирования с начальных классов.

Следует отметить, что тенденцию раннего обучения программированию в школе поддерживают многочисленные ведущие компании в области информационных технологий, предоставляя доступные инструменты для программирования, но и также повсеместно поддерживают идею обучения программированию в школе. Огромное количество пользователей таких ресурсов, как MIT's Scratch и AppInventor, Codecademy, Code.org и др. показывают растущий интерес современного общества людей к знанию и пониманию искусства программирования.

В таких средах процессе изучения программирования формируются специфические «функциональные мозговые органы». И очень важно, что эти «органы» формируются в общении и предметной деятельности ребенка. Среда программирования – представляет переходные объекты, который служат метафорами, с помощью которых учащиеся превращают опыт телесных манипуляций с вещами (поворот направо, шаг вперед и другие) в понятийные обобщения и абстракции, что важно в младшем школьном возрасте, когда умственная деятельность не отделена от моторной деятельности субъекта.

## **2. Среда обучения программированию**

В настоящее время существует множество онлайн-ресурсов и еще большее количество игр для персональных компьютеров и мобильных устройств, в которые встроены элементы обучения программированию или это обучение является основной целью. Они сильно различаются как по возрастной ориентации, так и по применяемым языкам и технологиям программирования. Множество ресурсов и игр являются англоязычными, но достаточно большое количество уже русифицированы и могут использоваться на занятиях.

В ресурсах, предлагаемым учащимся младших классов, основной используемой технологией является блочное программирование, где программа «собирается» из блоков – команд, соответствующих основным

инструкциям встроенного языка программирования. Как правило, реально существующие языки на этом уровне не используются.

Для выбора подходящей среды разработки были рассмотрены несколько популярных вариантов.

При выборе среды обучения необходимо учитывать следующие аспекты:

- простота изучения и, возможность его изучения маленькими детьми;
- возможность использовать язык не только для составления «игрушечных» программ, но и для решения сложных задач моделирования с использованием динамической графики;
- объектная ориентированность.

## **2.1. Онлайн среда Code.org.**

Рассмотрим некоторые элементы методики обучения программирования обучающихся младшего школьного возраста в среде Code.org.

Можно выделить следующие особенности среды Code.org для формирования алгоритмического мышления обучающихся младшего школьного возраста:

1) Первая и важная особенность – игровая форма обучения. Работа обучающихся в среде представляет собой онлайн-игру, в процессе прохождения которой дети знакомятся с основами программирования. Во время игры дети управляют зомби, пчелкой, художником или фермером, которые перемещаются по игровому полю и выполняют задания. Для этого игрок должен составить цепочку команд, а затем запустить их на выполнение. Перед каждым уровнем даются подсказки, все задания имеют визуальную и звуковую окраску. Данная характеристика среды призвана решить главную проблему при обучении программирования является – мотивация к изучению новой и сложной области.

2) Вторая особенность среды – двойное представление алгоритма – в виде блочного визуального языка и на языке javascript. Программирование исключает написание текста и производится простым перетаскиванием элементов из палитры. Чтобы программа заработала, достаточно написать несколько строчек кода, которые должны быть выполнены после нажатия на кнопку «выполнить». Наглядное отображение и интерактивное исполнение позволяет обучающемуся видеть поэтапное выполнение алгоритма, дает возможность анализировать и корректировать алгоритм.

3) Третья характеристика среды – последовательность и системность в освоении возможностей языка программирования. Данный сайт содержит несколько курсов рассчитанных на разные возрастные группы: от изучения простых команд для детей четырех лет, до работы с циклами, переменными процедурами и функциями, ориентированных на обучающихся 16-18 лет и старше.

Для педагога (после предварительной регистрации) предусмотрены возможности создания классов (групп), назначения им соответствующих курсов и дальнейший мониторинг успеваемости обучающихся (рис.1 и рис.2).

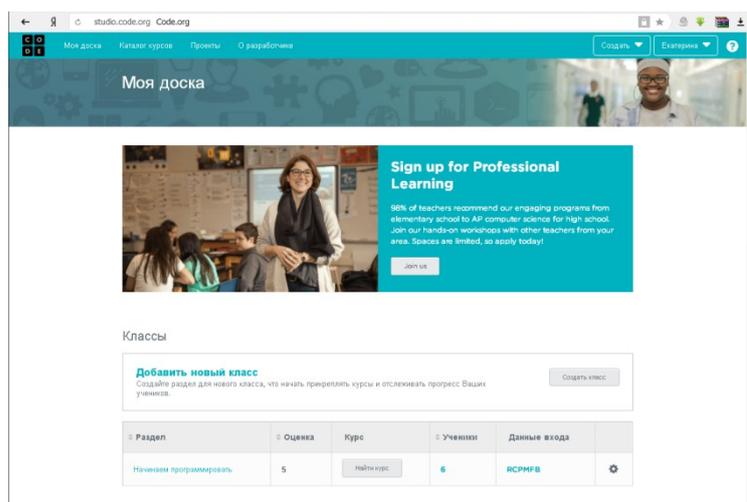


Рис. 1

Группа с  
указанием курса и количества обучающихся

Группа с

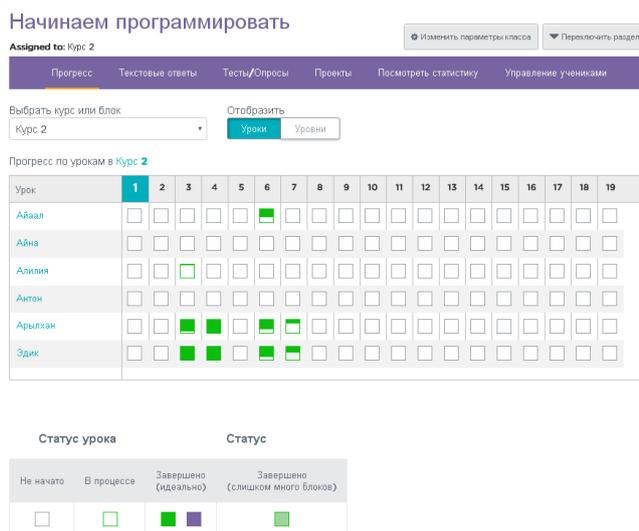


Рис.2 Мониторинг успеваемости обучающихся

В качестве пароля для обучающихся начальной школы можно использовать картинку (рис. 3), для более старшего возраста – кодовое слово.

Начинаем программировать

Assigned to: Курс 2

Изменить параметры класса | Переключить раздел:

Прогресс | Текстовые ответы | Тесты/Опросы | Проекты | Посмотреть статистику | Управление учениками

Добавить несколько учеников | Переместить студента

Имя	Возраст	Пол	Секрет	Действия
Имя ученика (обязательно)			Автоматически сгенерированные	Добавить
Айзаал Логин: code1369767242	10	Мужской	Показать секрет	▼
Айна Логин: code1648146688	9	Женский	Показать секрет	▼
Алия Логин: code1849704953	10	Женский	Показать секрет	▼
Антон Логин: code1218338612	9	Мужской	Показать секрет	▼
Арылхан Логин: code1321736538	11	Мужской	Показать секрет	▼
Эдик Логин: code1945959446	10	Мужской	Показать секрет	▼

Рис. 3

Данные об обучающийсяях группы

Данные об обучающийсяях группы

После формирования группы у педагога имеются ссылки для входа обучающихся на сайт. Перейдя по ссылке, обучающиеся видят фамилии своих одноклассников, находят свою и выбирают кодовую картинку или кодовое слово (рис.4). Кроме того на ресурсе появилась новая возможность – добавить друга при входе в курс. Данная возможность очень удобна при недостаточном количестве компьютеров в классе.



Рис. 4 Карточки с данными для входа

На занятиях для обучающихся 1 года обучения рекомендуется использовать первый курс. Желательно выполнять предлагаемые задания в начале учебного года, когда учащиеся только начинают знакомиться с компьютером. Задания третьего этапа направлены на работу с мышкой и составление картинок (Рис. 5).

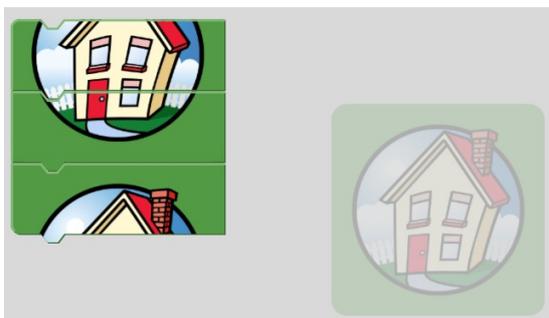


Рис. 5. Пример задания начального курса.

На следующих этапах дети знакомятся с составлением алгоритмов для различных персонажей (Рис. 6).

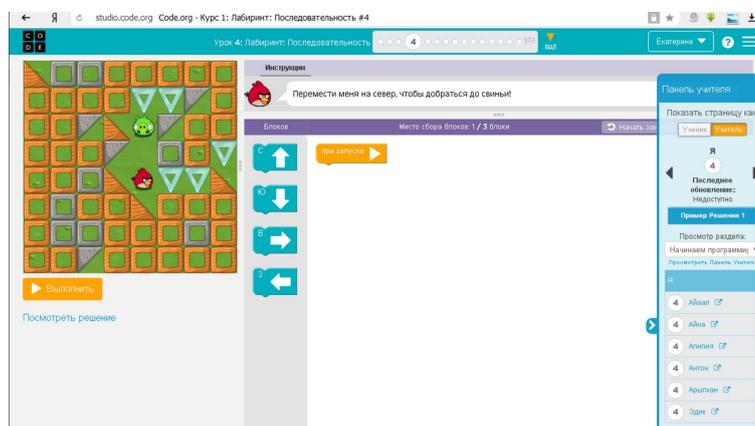


Рис 6. Пример задания 1 курса

В 3 и 4 классе можно переходить ко 2 и 3 курсам. На данном этапе команды в виде стрелок заменяются на команды, написанные на русском языке (Рис. 7). Выполняя задания на данных курсах учащиеся изучают линейные алгоритмические структуры, структуры ветвления и циклические структуры. При освоении этих курсов в средней школе можно рассмотреть функции и процедуры.

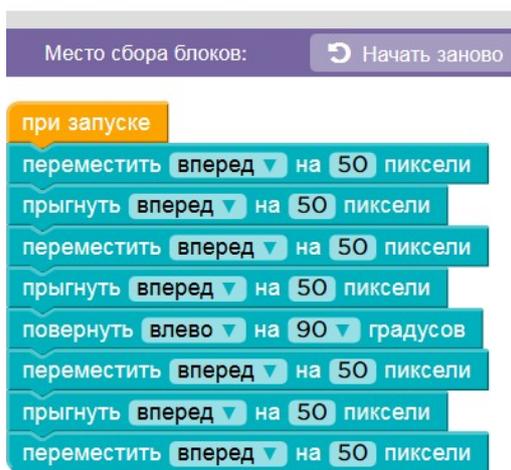


Рис. 7. Пример выполнения линейного алгоритма.

Каждый курс состоит из нескольких этапов. Например, курс 2 состоит из 19 этапов. На каждом этапе задания можно условно разделить на 4 фазы формирования алгоритмического стиля мышления:

1. Введение алгоритма: актуализация знаний, открытие алгоритма учащимися, освоение основных шагов алгоритма (рис. 8).



Рис. 8. Знакомство с командами исполнителя.

2. Усвоение алгоритма: отработка отдельных операций, входящих в алгоритм и усвоение их последовательности (рис. 9).



Рис.9. Отработка операций

3. Применение алгоритма: отработка алгоритма в знакомой и незнакомой ситуациях (рис.10)



Рис.10. Составление алгоритма в новой усложненной ситуации

Одной из веток сайта Code.Org, разработанной в сотрудничестве с Microsoft, стал проект Майнкрафт, выполненный по мотивам популярной игры. Школьники решают задачи по добыче необходимых полезных ископаемых, пользуясь той же блочной технологией программирования (рис.11).

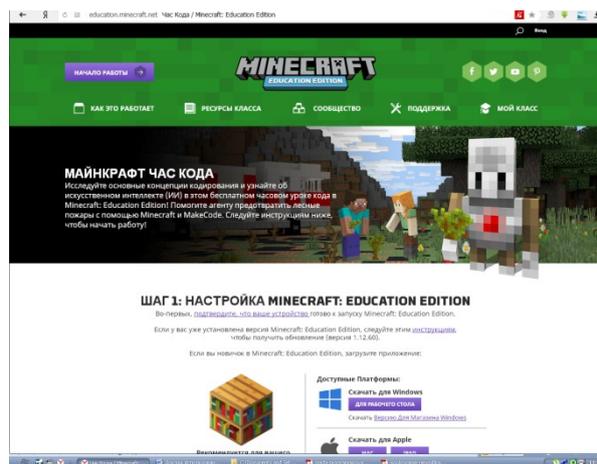


Рис.11. Майнкрафт Час кода

Более серьезный уровень программирования предлагается освоить в проекте Blockly (рис. 12). Здесь для построения программы используются не

только блоки, но и параллельно создается и демонстрируется текст на языке программирования JavaScript, который учащийся может редактировать:



Рис. 12. Код на JavaScript в Blockly

Обучение алгоритмизации и программированию подразумевает не только создание, но и применение алгоритма. Как показывает практика, дети не сразу могут это делать, для многих из них применение алгоритма является серьезной проблемой. В связи с этим мы уделяем достаточное количество времени работе парами с проговариванием каждого шага вслух. Все выполненные задания сохраняются в профиле обучающихся. Наиболее успевающих обучающихся мы выбираем в качестве экспертов-консультантов для помощи отстающим ребятам. Подобная работа повышает не только успеваемость всей группы, но и создает дополнительные стимулы для мотивации обучающихся.

## 2.2. Язык программирования Scratch (Скретч)

Потребность детей в средстве исследования и самовыражения сделало среду программирования Scratch всемирно известной. Среди самых востребованных языков в мире он находится на 24 месте. И с каждым годом активный интерес к нему растет среди взрослых и детей разного возраста. Scratch позволяет сразу видеть обучающимся результат своих действий и учит мыслить особым образом, понимать суть алгоритмов.

Scratch был разработан на языке Squeak. Теоретически это объектно-ориентированный язык с возможностью параллельного программирования. Язык создавался для учащихся 8-16 лет, их более привлекает легкость написания скрипта и использования графики и анимации, аудио и видеоэффектов. Он настолько «лёгок», но, тем не менее, «умён», что доступен и для самостоятельного изучения. Обучающемуся, осваивающему программирование в среде Scratch достаточно уметь читать и иметь элементарные пользовательские навыки работы за компьютером (в ОС Windows).

Основное предназначение данной программы – обучение детей программированию в игровой форме, где вместо сложных синтаксических конструкций языка программирования ребятам приходится работать с графическими объектами, которые называются Спрайтами. Им не надо беспокоиться о создании этих спрайтов при помощи специального класса, использовать атрибуты объектно-ориентированного программирования, чтобы прописать свойства, присущие всем спрайтам. Основная задача этой программы – помочь ребятам понять, что такое алгоритм, какие виды алгоритмов могут встречаться, возможность широко смотреть на поставленную задачу (необходимо учесть и запрограммировать все сценарии развития события), понять, что математика – это не просто предмет, изучаемый в школе, но и хорошее средство, с помощью которого можно заставить бездушные по началу объекты работать именно так, как хочешь ты.

Язык Scratch как средство развития научно-познавательной деятельности подходит по нескольким причинам:

- он легок и понятен в освоении, но имеет все возможности для серьезного программирования;
- позволяет создавать творческие, а не только научные проекты;
- имеет высокую популярность в интернет-сообществе, где обучающийся могут выкладывать свои проекты и общаться на эту тему.

Среда программирования имеет дружелюбный, «не научный», даже «детский» интерфейс, графический редактор для создания визуальных объектов, библиотеки готовых графических объектов, звуковых и аудио-объектов. (рис.1)

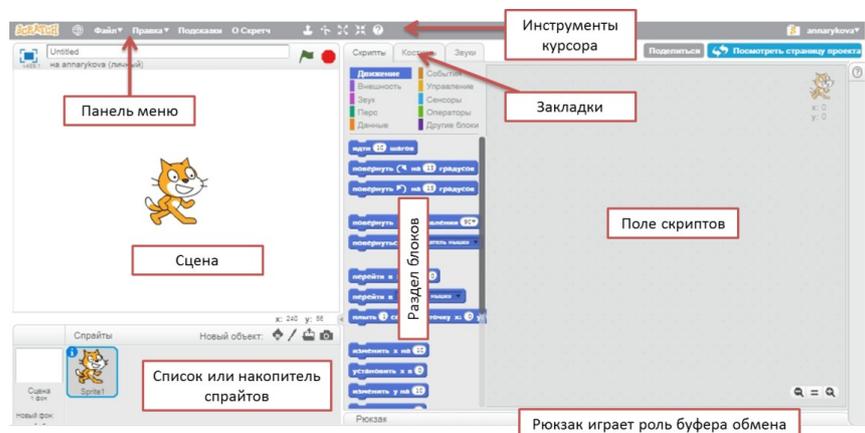


Рис.1. Интерфейс Scratch

Проектирование в Scratch можно выполнять коллективно. Среда идеальна для моделирования, т.к. позволяет не только легко выстроить модель «графически», но и провести тестирование модели и внести исправления в скрипт при неудовлетворяющих результатах или поэкспериментировать с различными входными данными. Одним из достоинств языка является его возможность создавать творческие проекты, с минимумом программирования как такового, постепенно осваивая азы программирования и переходя к более сложным проектам. На первом этапе внеучебной деятельности учащиеся осваивают основы алгоритмизации.

В среде программирования Scratch имеются 8 категорий блоков, с помощью которых реализуются методы и свойства объектов (объект здесь называется «спрайт»). Синий блок содержит операторы движения и направления. С помощью операторов фиолетового блока можно изменить «костюм» спрайта (свойства объекта), задать текстовую выноску объекту (речь или мысль), изменить эффекты, размеры объекта и расположение на определенном слое на сцене. Звуковые эффекты задаются розовыми блоками. Блоки категории перо (зеленые) позволяют «рисовать» в проекте. Команды в



хорошо развито воображение и кто пошёл нестандартным путём, кто просто хорошо и быстро выполнил задание, рассмотрев все случаи, а кому пришлось объяснить заново. После выполнения первого тренажёрного задания, становится понятно приблизительный уровень обучающихся и им предлагается написать на выбор одну из стандартных игр, с правилами которой ребёнок знаком уже давно. Всегда находятся ребята, жаждущие написать свою игру. В таком случае, мы вместе разбираем игровой процесс, рассматриваем все возможные сценарии, взаимодействие объектов внутри игры и так далее. Остальная часть обучения состоит в написании и постепенном улучшении написанной игры.

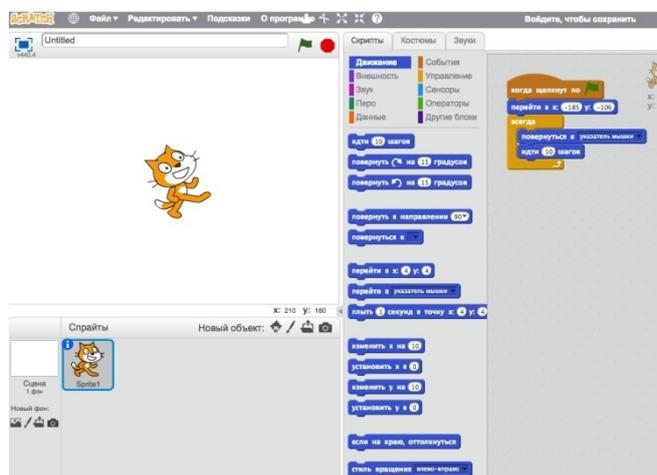


Рис.2. Простая программа для движения спрайта

После изложения игрового процесса, необходимо заострить внимание на взаимодействии объектов между собой, ответить на вопрос «что будет, если...». Язык Scratch устроен таким образом, что отладчик представляет из себя сцену, на которой впоследствии и будет запускаться игра. Тестировать её можно на любом этапе – и это несомненный плюс. В любой момент можно понять, что пошло не так, и как можно быстрее исправить это.

В программе Scratch можно управлять объектами с помощью клавиатуры, но тут есть один изъян: программа не может распознать две и более клавиш, нажатых одновременно, ровно как не может распознать клавишу, которую долго удерживают.

Далее, для игрового процесса желательно использовать переменные, которые в этой программе тоже реализованы. Их значения, при необходимости или отладке, можно тоже выводить на сцену. В любой игре в переменных можно держать текущий счёт игры, лучший результат и всё прочее (рис.3).

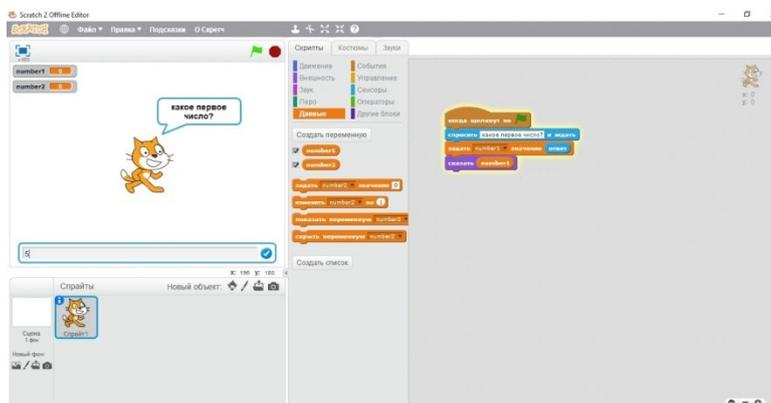


Рис.3. Переменные в Scratch

После реализации игрового процесса, необходимо контролировать начало игры, возможные сценарии для победы и поражения. На этом этапе надо выбрать кнопку, при нажатии на которую текущая игра остановится, а все объекты займут начальное положение, все переменные примут некоторое начальное значение. Для этого необходимо для каждого объекта установить начальное положение, остановить при необходимости. Вызов этих функций должен как раз осуществляться с помощью клавиши начала игры.

Аналогично стоит прописать все сценарии побед и поражений, а для этого нам необходимо:

- остановить все объекты при необходимости, когда игра будет закончена;
- придумать и реализовать критерий для победы (больше количество очков, меньше время для выполнения задания и т.д.);
- продемонстрировать победу одного из игроков, например, сменой фона (рис.4).

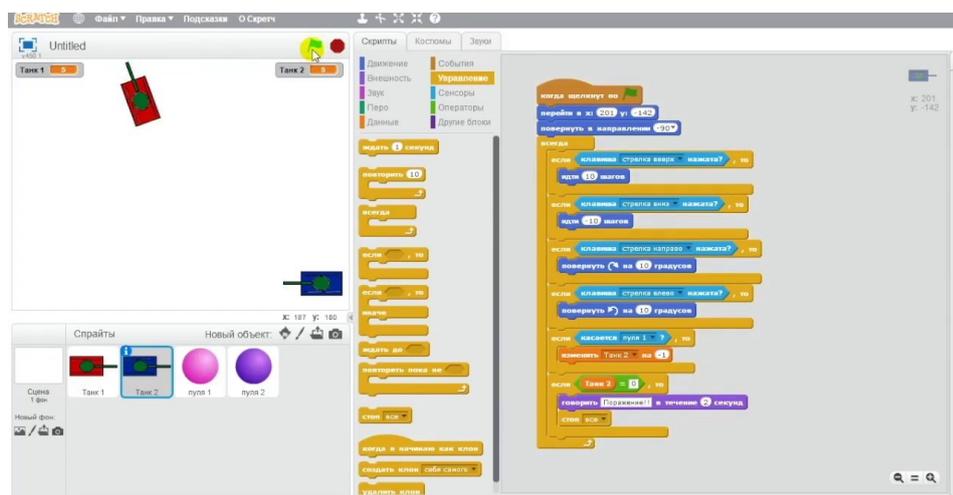


Рис.4. Сценарий победы и поражения для игры «Танчики»

Обычно большая часть времени тратится непосредственно на совершенствование игрового процесса и интерфейса пользователя. Здесь допустимы несколько путей развития вашего проекта:

- возможность мультиплеера (необходимо помнить, что одновременное нажатие клавиш не распознаётся программой);
- создание стратегии игры для компьютера;
- создание уровней сложности.

Первый пункт достигается с точки зрения алгоритмизации очень легко: мы должны поставить ещё один спрайт, прописать на него все те же функции, но запрограммировать их на другие клавиши. В программе Scratch это легко реализовать, так как спрайты можно копировать вместе со всем их кодом. Второй и третий пункты идут рядом друг с другом. Понятно, что во многих играх при желании можно создать такую стратегию, что наш игрок проиграет компьютеру. Аналогично можно создать стратегию, при которой проигрывать будет всегда компьютер. Нам надо найти некоторый баланс, ту золотую середину, при которой интерес к игре будет сохраняться. Достигнуть этого можно, если подключить математический аппарат, а именно, генерацию случайных величин с необходимым законом распределения.

Детям постарше необходимо рассказывать, что в большинстве игр нужно применять физические законы, например, для описания движения спрайтов. Таким образом, физика тоже перестанет быть сухим предметом и найдёт игровое применение (рис.5).

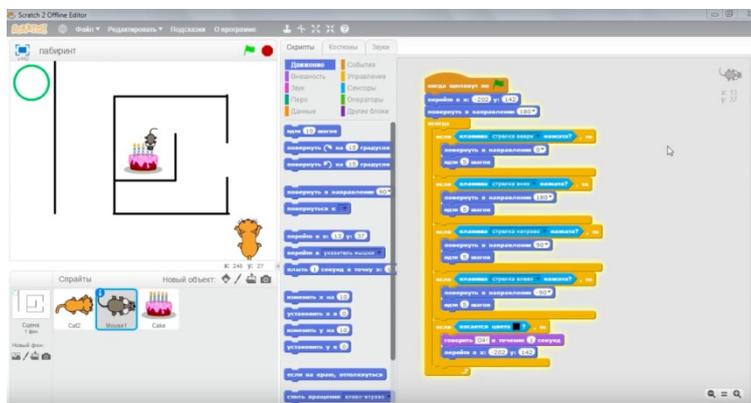
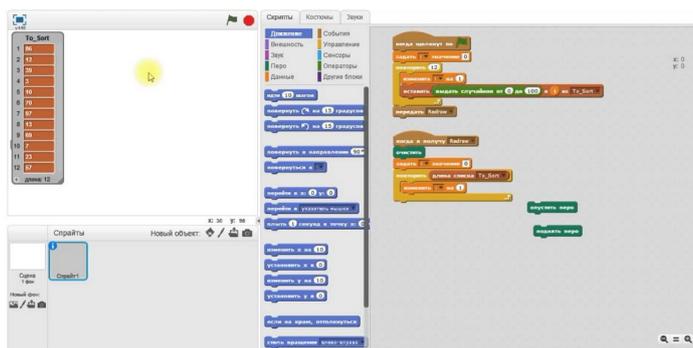


Рис.5. Программа с использованием нескольких спрайтов

Одним из последних этапов написания игрового проекта – его оформление. Здесь необходимо прописать динамику всех сцен, желательно придумать специальную сцену для начала игры, в которой будет небольшое меню, где можно как раз будет выбрать одиночную игру или мультиплеер, выбрать уровень сложности (для одиночной игры). Всё это можно реализовать с помощью управляющих переменных.

Для игр посложнее, (а, следовательно, для детей постарше), в программе Scratch реализовано хранение и обработка множества данных с помощью структуры данных «список». Аналогичные структуры данных применяются уже в вузе, при изучении таких языков программирования, как C++, Python и т.д., т.е. закладывается фундамент для работы со сложными структурами данных (рис.6)



## Рис.6. Список в Scratch

В Scratch имеется возможность создавать свои блоки, которые являются аналогами функций и процедур в стандартных языках программирования. Эти структуры помогают будущим программистам организовать свои мысли таким образом, чтобы код был интуитивно понятным.

Если рассматривать проект написания игр с точки зрения совершенствования навыков программирования, то за время обучения обучающийся знакомится с такими важными понятиями, как алгоритм, его виды (линейный, ветвящийся и циклический), имеет представление об операторе присваивания. Кроме того, ребята получают важные знания о переменных и способах и 4 этапах жизни переменной (декларация, инициализация, динамика переменной, вывод при необходимости) и активно применяют эти знания, например, чтобы отслеживать счёт игры или количество жизней.

## **Заключение**

Среда Code позволяет учащимся младшего школьного возраста знакомиться с основами алгоритмизации и программирования в процессе игры со знакомыми им героями, изучать сложную тему и готовиться к изучению программирования, а Scratch сочетает в себе все достоинства первого языка для обучения программированию: он прост, имеет большой потенциал, знакомство с ним может происходить через игру, что мотивирует и заинтересовывает учащихся.

Раннее обучение программированию позволит с меньшими трудностями перейти к другим учебным языкам в старших классах, а некоторые дети «влюбятся» в процесс программирования.

На мой взгляд, именно с этих сред обучения следует начинать знакомство с программированием. Литературы для знакомства с ними на русском языке не так много, но Интернет-ресурсов достаточно, имеющиеся методики еще нуждаются в доработке.

Использование описанных средств программирования позволяет не только пробудить в школьнике интерес к программированию, но и научить его создавать свои первые программы в виде, понятном и интересном его одноклассникам – в виде компьютерной игры. И если это помогает изучению реальных языков программирования в старших классах, а затем – в высшем учебном заведении, то положительный эффект такого подхода очевиден.

Выявлено, что геймификация при программировании способствует расширению мировоззрения школьников, активизирует творческую деятельность, развивает алгоритмическую и логическую культуру их мышления, реализует межпредметные связи.

### **Использованная литература:**

1. Code.org – Учится в студии кода [Электронный ресурс]. URL:<https://studio.code.org> (дата обращения: 27.11.20157)
2. Minecraft – Education Edition [Электронный ресурс] URL <https://education.minecraft.net/> (дата обращения 03.12.2017)
3. Scratch - Создавайте истории, игры и анимацию [Электронный ресурс] URL: <https://scratch.mit.edu/> (дата обращения 12.12.2017)
4. Дженжер В. О. Место программирования в курсе информатики начальной школы // Вестник ОГУ. 2010. №9 (115), стр. 154-159
5. Захарова И.В., Кузенков О.А. Опыт реализаций требований образовательных и профессиональных стандартов в области ИКТ в российском образовании // Современные информационные технологии и ИТ-образование. 2016. Т. 12. № 3-1. С. 17-31.
6. Пейперт С. Переворот в сознаний: Дети, компьютеры плодотворные идеи: Пер. с англ./под. ред. А.В. Беляевой, В.В. Леонаса. – М.: Педагогика, 1989. – 224с.
7. Программирование на Blocly [Электронный ресурс] URL: <http://blockly.ru/> (дата обращения 12.12.2017)
8. Федеральный государственный образовательный стандарт основного общего образования от 17.12.2010 г. № 1897. 50 с.
9. Час кода для каждого ученика [Электронный ресурс] URL:<https://code.org/> (дата обращения 03.12.2017)
10. Час.Кода [Электронный ресурс] URL: <http://часкода.рф/> / (дата обращения 03.12.2017)