

ОСОБЕННОСТИ ОРГАНИЗАЦИИ КУРСОВ ДЛЯ УЧАЩИХСЯ СРЕДНЕЙ ШКОЛЫ ПО ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ SCRATCH 2.0

Мороз Ольга Викторовна,

кандидат педагогических наук, доцент кафедры информационных образовательных технологий ФГБОУ ВО «Кубанский государственный университет», г. Краснодар,
e-mail: kotkiot@yandex.ru

Косярский Александр Алексеевич,

педагог дополнительного образования МАОУДО ЦТД «Прикубанский», г. Краснодар,
e-mail: 15sasha1998@mail.ru

В СТАТЬЕ ОПИСАНА МОДЕЛЬ ОРГАНИЗАЦИИ КУРСОВ ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ УЧАЩИХСЯ ПО ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ СОБЫТИЙНО ОРИЕНТИРОВАННОЙ СРЕДЫ SCRATCH 2.0. РАССМОТРЕНА ТЕХНОЛОГИЯ ОРГАНИЗАЦИИ ИНДИВИДУАЛЬНОЙ И ГРУППОВОЙ ДЕЯТЕЛЬНОСТИ УЧАЩИХСЯ. ОПРЕДЕЛЕНЫ ОСНОВНЫЕ ТЕМАТИЧЕСКИЕ БЛОКИ И РАЗДЕЛЫ ВНЕУРОЧНОГО КУРСА, РЕКОМЕНДОВАННЫЕ ДЛЯ ИЗУЧЕНИЯ В РАМКАХ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ.

• проектная деятельность • программирование • проектирование • информационные технологии • творческое развитие • алгоритмизация

На сегодняшний день жизнь человека тесно связана с развитием различного рода информационных технологий, что в свою очередь привело к формированию цифрового общества, для которого наибольшую ценность приобрели знания и информация. Практически невозможно представить ни одну из сфер деятельности человека без использования информационных технологий. Основными сферами использования компьютерных технологий по достоинству являются технологическая отрасль, медицина и образование.

Информационно-коммуникационные технологии уже — неотъемлемое звено образовательной системы, так как дают обширные возможности для реализации функций наглядности и научности в процессе обучения, а также позволяют осуществлять контроль знаний с использованием нестандартных форм взаимодействия учащегося с учеником. Но в то же время необходимо заметить, что столь же велика роль информационных технологий преимущественно

в рамках занятий по информатике и информационно-коммуникационным технологиям, а также в кружковой деятельности.

В соответствии со стандартами основной целью изучения информатики является целенаправленное обучение учащихся навыкам работы с информацией.

Другая важная задача и цель изучения информатики — обучение учащегося навыкам логического, алгоритмического и программного мышления. Важность изучения алгоритмизации связана прежде всего с тем, что решение многих задач требует построения плана действий, являясь примером простого алгоритма.

Роль программирования в школьном курсе информатики определяется не только требованиями информационного общества, но и фундаментальными целями, которые предъявляются к школьному курсу информатики в связи с развитием научной отрасли.

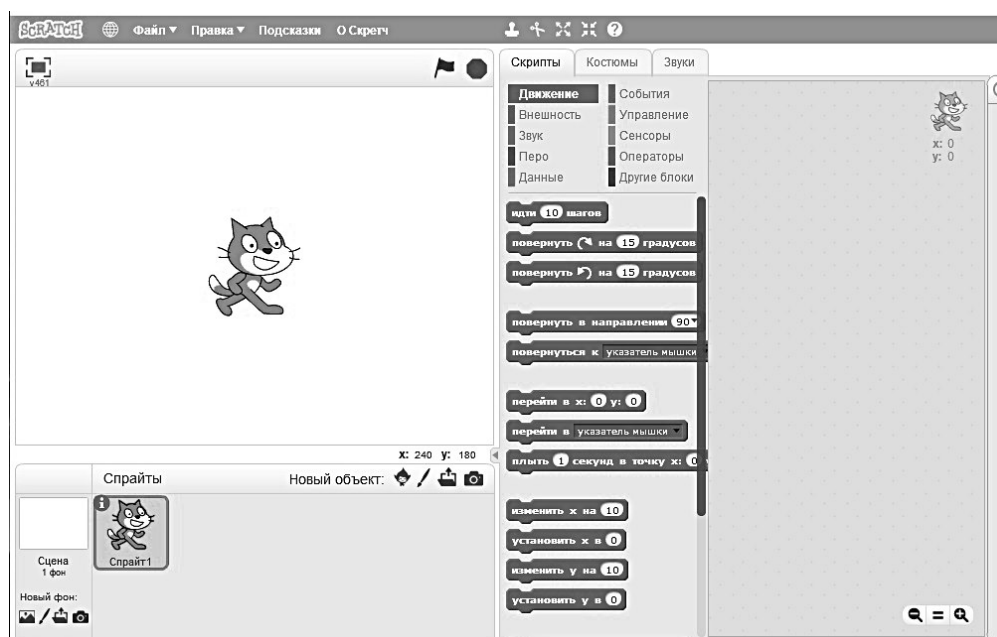


Рис. 1. Интерфейс среды Scratch 2.0

Однако нужно отметить, что очень часто реализовывать данную функцию невозможно, руководствуясь лишь уроками по информатике. В таком случае одно из решений данной проблемы — организация дополнительных курсов по изучению информатики. Но для того, чтобы обеспечить грамотное и наиболее понятное для школьника изучение программирования, необходимо выбрать прежде всего удобную и лёгкую для понимания среду программирования.

На сегодняшний день существует очень много программных средств, которые обеспечивают выполнение поставленной задачи, однако наиболее удобной средой программирования является Scratch 2.0 (рис. 1).

Что такое Scratch? Scratch — это прежде всего объектно ориентированная среда разработки, главной особенностью которой является то, что при разработке программ для построения различных блоков используются разноцветные «кирпичики», подобно детскому конструктору Lego.

В настоящее время в МБОУ СОШ № 89 г. Краснодар совместно с факультетом математики и компьютерных наук ФГБОУ ВО «Кубанский государственный университет» в рамках осуществления федерального инновационного проекта «Роль межшкольного сетевого взаимодействия в развитии уча-

щихся в предметной области «Математика и информатика»», а также на базе клуба по месту жительства «Калейдоскоп» ЦДТ «Прикубанский» Краснодара реализуется программа внеурочной деятельности «Программирование в визуальной событийно ориентированной среде Scratch».

Процесс обучения в рамках курса может быть представлен в виде схемы (рис. 2).

Как видно на рисунке, каждое занятие ориентировано на развитие таких важных навыков и умений как:

- осуществлять анализ проблемной задачи;
- осуществлять поиск решения проблемной задачи;
- умение строить алгоритмическую модель решения поставленной задачи;
- умение работать в коллективе.

В процессе решения поставленных учебных задач учащиеся принимают самое активное участие в осуществлении всего образовательного процесса, что делает его более ориентированным на личностное развитие и обеспечивает наиболее продуктивную творческую атмосферу.

В рамках практических и лекционных занятий школьники изучают основы алгоритмизации и программирования, используя технические возможности среды разработки Scratch.

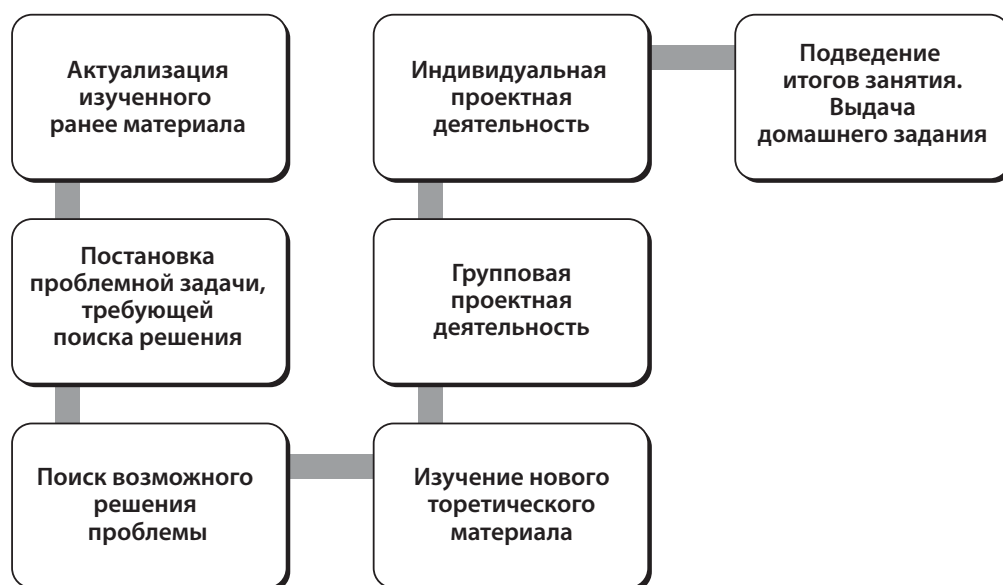


Рис. 2. Структура учебного занятия

Основными формами организации проектной деятельности являются:

- анимированная мультипликация;
- обучающие и развивающие игры;
- приложения, направленные на обучение учащихся и контроль знаний.

Созданные проекты учащиеся размещают на сетевых ресурсах портала **scratch.mit.edu**, а также используют сетевые программные средства адаптации созданного продукта для мобильных операционных систем iOS и Android. Проектная деятельность учащихся состоит из двух важных компонентов:

- 1) индивидуальная проектная деятельность;
- 2) групповая проектная деятельность.

Главной особенностью индивидуальной проектной работы является её оригинальность и неповторимость реализации, что является следствием осуществления творческого подхода к разработке проекта.

При выполнении индивидуальных проектов следует придерживаться определённого алгоритма, для описания которого выделяются этапы (рис. 3).



Рис. 3. Этапы индивидуальной проектной деятельности

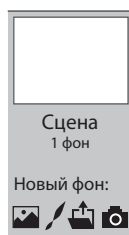


Рис. 4. Вкладка выбора фонов

Приведём алгоритм написания индивидуального проекта «Волшебник и колдунья».

Этап 1. Перед учащимся ставится задача выбрать сцену, на которой происходит действие (рис. 4).

Этап 2. После определения фона сцены учащиеся выполняют задание: нарисовать или выбрать готовые спрайты волшебника и колдуньи. После добавления в проект персонажей получаем следующий набор спрайтов и сцен (рис. 5).

Этап 3. После выбора персонажей и фона для сцены учащиеся получают задание со-

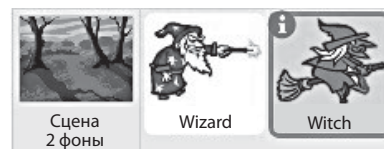


Рис. 5. Забор спрайтов и сцен

здать для волшебника зелья в колбах. Так как колбы изначально не присутствуют в библиотеке спрайтов, учащиеся рисуют спрайт при помощи встроенного графического редактора (рис. 6).

Для создания спрайта волшебной колбы нужно выбрать инструмент «Эллипс», после чего — построить овал, который следует преобразовать в следующей последовательности (рис. 7).

После того, как создана форма волшебной колбы, выполняем заливку объекта и добавляем крышку (рис. 8).

Этап 4. Затем необходимо подобным образом нарисовать спрайты ещё 4 склянок, после чего проект примет следующий вид (рис. 9).



Рис. 6. Создание спрайта и выбор режима графики

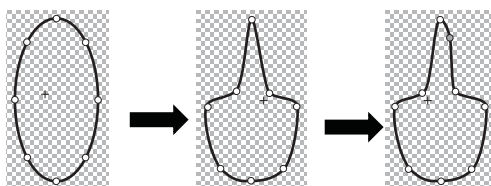


Рис. 7. Создание спрайта волшебной колбы

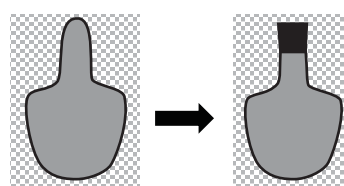


Рис. 8. Готовый спрайт колбы

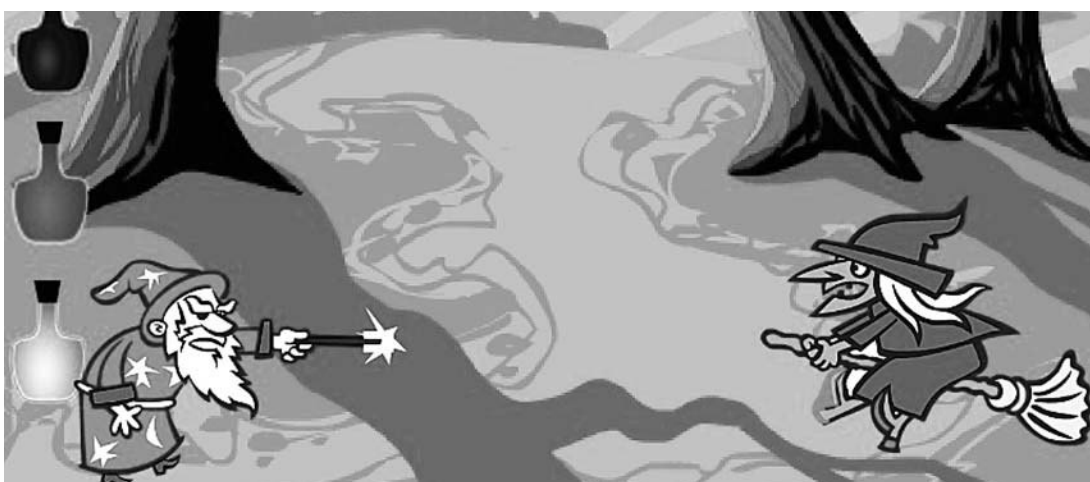


Рис. 9. Сцена проекта «Волшебник и колдунья»

Этап 5. На данном этапе следует разработать скрипты, отвечающие за происходящие на сцене изменения при касании соответствующих спрайтов.

Для каждой колбы нужно составить скрипты, представленные ниже (рис. 10–14).

Этап 6. Програмируем волшебника, для него необходимо составить следующую последовательность скриптов (рис. 15).

Этап 7. Програмируем колдунью, для чего составляем следующую последовательность скриптов (рис. 16).

Этап 8. На данном этапе учащийся осуществляет тестирование проекта и после завершения отладки публично защищает его перед другими учащимися.

В процессе написания индивидуального проекта учащиеся отрабатывают несколько важных навыков:

- построения алгоритмической модели проблемной задачи;
- осуществления поиска ошибок;
- анализа полученных данных и коррекции скриптов;
- одновременной работы с несколькими спрайтами и различными программными блоками.

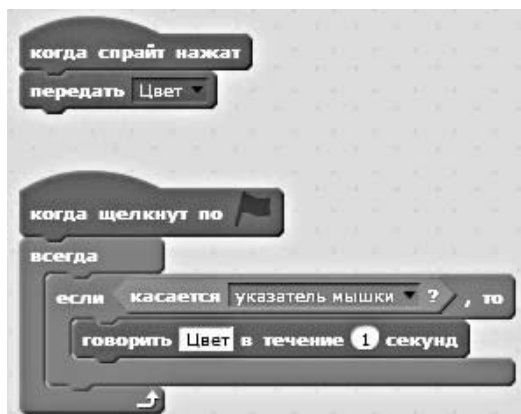


Рис. 10. Скрипт для красной колбы

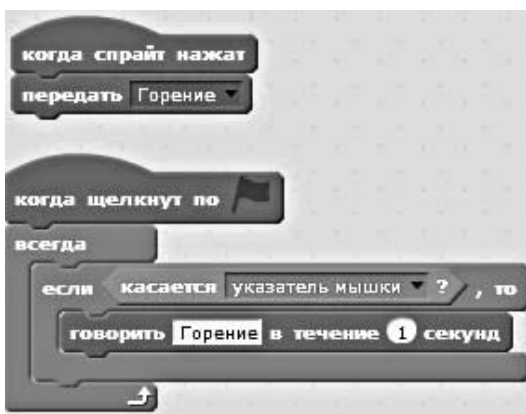


Рис. 11. Скрипт для синей колбы

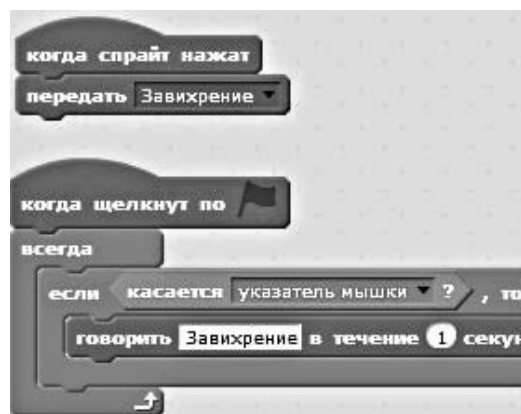


Рис. 12. Скрипт для зелёной колбы

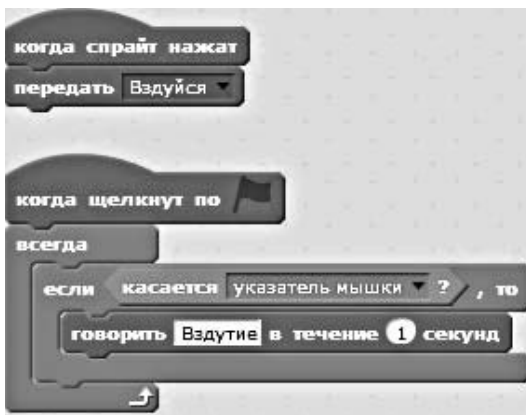


Рис. 13. Скрипт для фиолетовой колбы

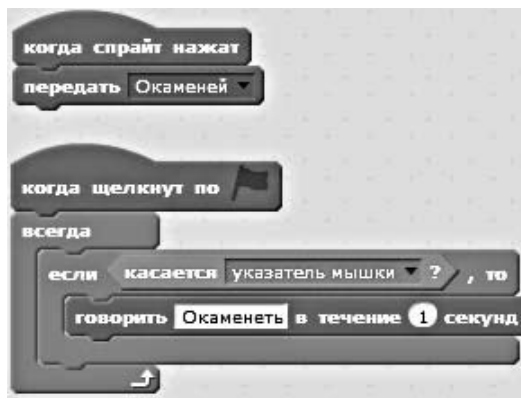


Рис. 14. Скрипт для жёлтой колбы



Рис. 15. Скрипты волшебника

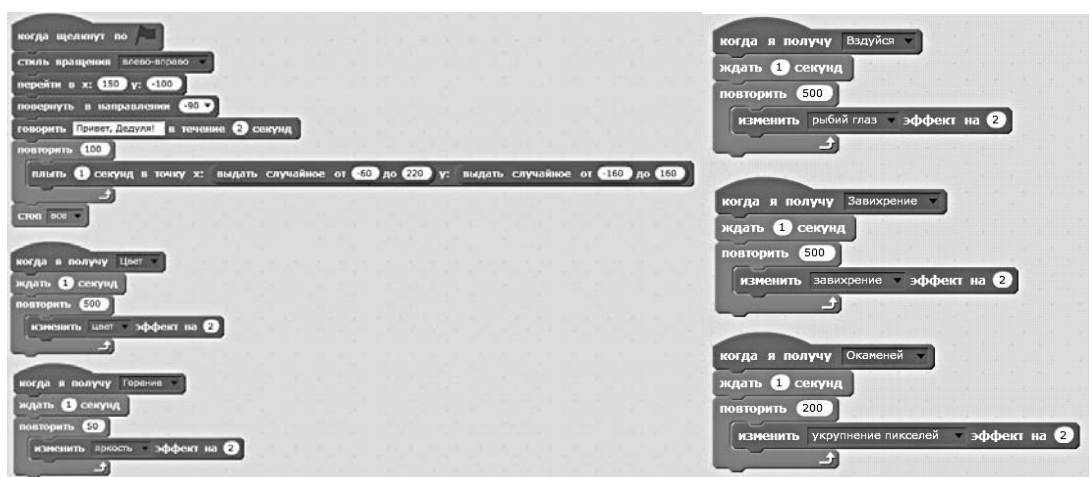


Рис. 16. Скрипты колдуньи

При этом учащиеся закрепляют полученные в рамках теоретических и практических занятий знания, обобщают их и группируют в единую, целостную структуру.

Немалое значение при обучении программированию имеет и групповая работа. Её главной целью является развитие навыков коммуникации и коллективной работы.

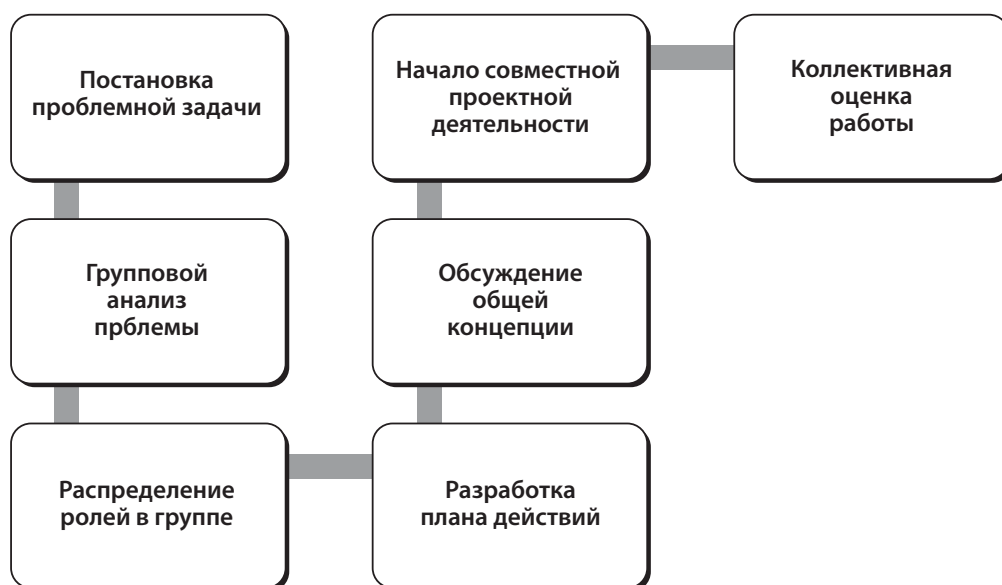


Рис. 17. Структура группового взаимодействия

В процессе групповой проектной деятельности формируются такие навыки и умения как:

- навык коллективной творческой работы;
- навык осуществления анализа поставленной задачи;
- умение распределять функции в группе, разделять общую работу на нескольких человек;
- умение осуществлять творческое взаимодействие.

Общий алгоритм группового взаимодействия может быть описан при помощи следующей схемы (рис. 17).

Аналогично конструированию индивидуального проекта можно выстроить алгоритм выполнения группового проекта, выделив следующие этапы (рис. 18).

Приведём пример создания группового проекта по разработке двухуровневого лабиринта с функциональными клавишами.

Этап 1. Перед учащимися ставится задача распределиться на группы по 5 человек и выбрать участников, которым достанется одна из ролей:

- программист, художник-редактор;
- программист, отладчик.



Рис. 18. Этапы групповой проектной деятельности

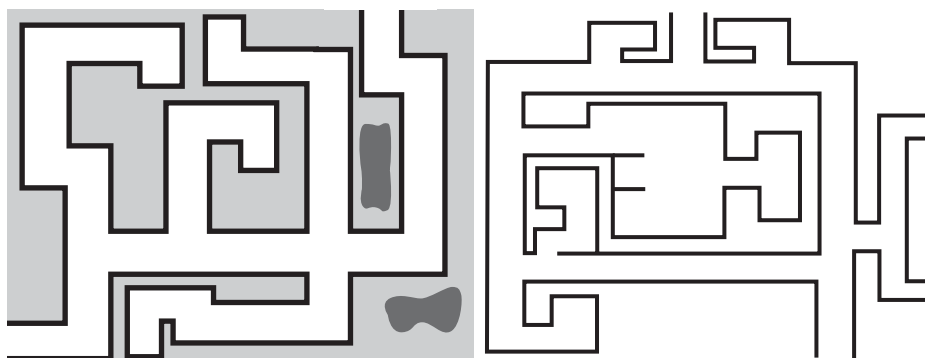


Рис. 19. Макеты уровней игры в среде Scratch 2.0

Этап 2. Учащиеся получают творческое задание разработать макет уровней лабиринта. Участники группы строят макет лабиринта с использованием графического редактора Paint.

Когда модель лабиринта будет построена, участник, выполняющий роль художника-редактора, разрабатывает оформление уровней в среде Scratch в соответствии с созданным макетом (рис. 19). А когда макет готов, проект загружают на сетевой диск или передают посредством USB-носителей остальным участникам группы для дальнейшего программирования лабиринта.

Этап 3. На данном этапе перед участниками группы ставится задача построения сюжетной линии, выбора спрайтов, а также распределения обязанностей по программированию отдельных спрайтов (рис. 20).

Этап 4. Создав первый уровень лабиринта, учащиеся, используя сетевой диск или USB-носители, передают готовые скрипты члену группы, который занимается отладкой программы. В обязанности данного учащегося входит соединение воедино всех разрозненных участков кода, отладка возможных ошибок алгоритмического характера. Затем обработанная программа передаётся остальным участникам группы для дальнейшей разработки. Аналогичные действия проводятся и при завершении создания следующих уровней лабиринта.

Этап 5. На данном этапе осуществляется общий анализ написанной программы всем коллективом программистов, которые вместе с отладчиком осуществляют редактирование кода, доработку скриптов каждого спрайта.

Этап 6. Этот этап предназначен для проведения серии тестов программы с заполнением таблицы тестирования, в которой отражают предполагаемые и наблюдаемые результаты и общую оценку правильности выполнения скрипта (табл. 1).

Этап 7. На данном этапе происходит публичная защита проекта, в процессе которой учащиеся объясняют структуру своего проекта, сюжетную линию и происходящие в программе изменения при взаимодействии спрайтов.

Необходимо отметить, что групповая проектная работа необходима не только для оценки умения учащихся работать в коллективе, но и для творческого сближения школьников с целью профилактики конфликтных ситуаций, так как в процессе такой деятельности ребята учатся достигать компромисса между желаниями всех членов группы и идти на уступки.

Следует подчеркнуть, что проектная деятельность является в большей степени



Рис. 20. Спрайты проекта «Лабиринт»

Таблица 1

Тестирование проекта

Тест	Предполагаемый результат	Наблюдаемый результат	Оценка правильности работы скрипта
Касание зелёных границ	Котёнок переходит в начало уровня	Котёнок переходит в начало уровня	Работает
Касание оранжевых границ	Котёнок переходит в начало уровня	Котёнок переходит в начало уровня	Работает
Касание артефактов	При нажатии клавиши «е» артефакт исчезает	При нажатии клавиши «е» артефакт исчезает	Работает
Касание торта	При нажатии клавиши «е» торт исчезает, котёнок увеличивается	При нажатии клавиши «е» торт исчезает, котёнок увеличивается	Работает
Касание противоядия	При нажатии клавиши «е» противоядие исчезает, котёнок уменьшается	При нажатии клавиши «е» противоядие исчезает, котёнок уменьшается	Работает

методом контроля и коррекции знаний, нежели средством обучения. Для изучения программирования предлагается использование чётко установленной последовательности теоретических блоков и совместных практических занятий, направленных на поиск и решение проблемных задач.

Так, весь теоретический материал курса можно представить в виде следующего списка.

Раздел «Базовые алгоритмические конструкции»:

- создание программной реализации алгоритмической модели проекта;
- обработка звуковой информации;
- автоматизация движения спрайтов по рабочей области, имитация анимации.

Раздел «Графические преобразования»:

- создание упрощённой версии универсального графического редактора цвета;
- изменение прозрачности спрайта с использованием графических возможностей среды;
- автоматизация процесса построения изображений с использованием сенсоров.

Раздел «Многократное повторение или циклические операторы»:

- автоматизация графических эффектов: печать спрайта, построение фигур;
- движение спрайта с изменяющимися параметрами.

Раздел «Логические конструкции. Условный оператор»:

- разработка логического анализатора;
- автоматическое построение фракталов с использованием циклических и условных операторов.

Раздел «Данные и списки»:

- разработка универсального арифметического приложения;
- создание игровых проектов с использованием счётчика;
- разработка универсальной тестовой системы по предметной области «информатика».

Однако приведённый выше список не является обязательным для детального следования и может быть скорректирован преподавателем.

В заключение необходимо сказать, что использование в современном образовательном процессе мощных событийно ориентированных сред, например Scratch, позволяет учащимся участвовать в процессе первичной профориентационной деятельности, формировать умения работы с алгоритмическими конструкциями. Технические возможности Scratch позволяют использовать его для проведения опытов, исследований, для выполнения проектных работ по самым различным дисциплинам, прежде всего — по информатике и математике. □

Литература

1. *Патаркин Е.Д.* Школа SCRATCH // Школьные технологии. — 2010. — № 4. — С. 132–135.
2. *Голиков Д.В.* Scratch для юных программистов. — СПб.: БХВ-Петербург, 2019. — 192 с.: ил.
3. *Голиков Д.В.* 40 проектов на Scratch для юных программистов. — СПб.: БХВ-Петербург, 2019. — 192 с.: ил.
4. *Сахибов А.* Теория применения информационно-коммуникационных технологий в образовании // Наука и мир. — 2014. — Т. 3. — № 4(8). — С. 81–83.
5. *Еремин Е.А.* Среда Scratch — первое знакомство // Информатика. — 2008. — № 20 (573) — С. 17–24.
6. *Яшуев Р.Н.* Работа со школьниками в области информатики. — М., 2007.
7. *Путина А.С.* SCRATCH-олимпиада по креативному программированию // Информатика в школе. — 2018. — № 4 (137). — С. 57–61.
8. *Савченкова М.В.* Scratch-Хакатон «Программируем в среде Scratch» // Информатика в школе. — 2018. — № 4(137). — С. 24–27.

Literatura

1. *Patarkin E.D.* Shkola SCRATCH // SHkol'nye tekhnologii. — 2010. — № 4. — S. 132–135.
2. *Golikov D.V.* Scratch dlya yunyh programmistov. — SPb.: BHV-Peterburg, 2019. — 192 s.: il.
3. *Golikov D.V.* 40 proektov na Scratch dlya yunyh programmistov. — SPb.: BHV-Peterburg, 2019. — 192 s.: il.
4. *Sahibov A.* Teoriya primeneniya informacionno-kommunikacionnyh tekhnologij v obrazovanii // Nauka i mir. — 2014. — T. 3. — № 4(8). — S. 81–83.
5. *Eremin E.A.* Sreda Scratch — pervoe znakomstvo // Informatika. — 2008. — № 20 (573) — S. 17–24.
6. *Yashuev R.N.* Rabota so shkol'nikami v oblasti informatiki. — M., 2007.
7. *Putina A.S.* SCRATCH-olimpiada po kreativnomu programmirovaniyu // Informatika v shkole. — 2018. — № 4 (137). — S. 57–61.
8. *Savchenkova M.V.* Scratch-Hakaton «Programmiruem v srede Scratch» // Informatika v shkole. — 2018. — № 4(137). — S. 24–27.