

Л. В. Денисова, В. О. Дженжер

Оренбургский государственный педагогический университет, Россия

ВОКРУГ ТРЕУГОЛЬНИКА СЕРПИНСКОГО. ИГРА В ХАОС И ПРОГРАММИРОВАНИЕ В СРЕДЕ SCRATCH

Аннотация

В статье рассматривается опыт изучения геометрических фракталов Серпинского на кружковых занятиях по математике и информатике с учениками IV—VI классов. Используется разносторонний подход к построению фрактальных объектов: вырезание из бумаги, игра в хаос на прозрачных пленках и реализация изученного алгоритма на языке программирования Scratch. При этом с детьми обсуждаются некоторые вопросы, связанные с возникновением случайных событий. После того как фрактал получен на компьютере, следует этап исследования построенной модели. В компьютерную «игру в хаос» последовательно добавляются: 1) небольшой поворот на заданное число градусов перед постановкой очередной точки; 2) поворот перед отдельными вершинами фрактала; 3) сжатие фрактала к отдельным вершинам. Делаются выводы о самоподобии получающихся объектов. Предлагаемый метод позволяет изучить некоторые свойства фрактальных множеств, привлекая наглядность, интуицию и компьютерное моделирование.

Ключевые слова: игра в хаос, случайность, треугольник Серпинского, кривая Коха, компьютерное моделирование, программирование, Scratch.

DOI: 10.32517/2221-1993-2020-19-4-20-26

1. Фракталы и программирование

Все, кто увлекается программированием, наверняка хотя бы раз строили какой-нибудь фрактал или изучали изображения, получаемые при помощи фрактальных алгоритмов. Фрактальная геометрия является одним из современных разделов математики, методы и алгоритмы которой находят применение в различных областях, таких, например, как сжатие изображений, геология и геофизика, астрономия и даже медицина. В современных изданиях по занимательной математике для детей также часто встречаются обращения к этой теме [2, 4, 6]. Поэтому введение объектов фрактальной природы в программу физико-математического кружка студентов и школьников вполне оправданно. Однако иногда знакомство с фракталами ограничивается построением и рассмотрением очередной красивой картинкой. Такой подход нам кажется неудачным. За красивыми

картинками часто скрываются не менее красивые математические идеи, многие из которых доступны для понимания даже младшему школьнику.

Мы считаем, что освоение компьютера и программирования не является самоцелью, но снабжает школьника мощным инструментом для изучения многих интересных объектов, которые без компьютера восприниматься гораздо сложнее либо вовсе не поддаются изучению. Следуя этому подходу, мы провели с учениками IV—VI классов серию занятий по применению среды Scratch в изучении некоторых геометрических фракталов.

При проведении занятий нами преследовалось **несколько целей:**

- познакомить детей с фракталами;
- ввести понятие самоподобия;
- поговорить о случайных числах и частоте возникновения некоторых случайных событий;

Контактная информация

Денисова Людмила Викторовна, канд. пед. наук, доцент, доцент кафедры информатики, физики и методики преподавания информатики и физики, Оренбургский государственный педагогический университет, Россия; *адрес:* 460014, Россия, г. Оренбург, ул. Советская, д. 19; *e-mail:* lv-denisova@yandex.ru

Дженжер Вадим Олегович, канд. физ.-мат. наук, доцент, зав. кафедрой информатики, физики и методики преподавания информатики и физики, Оренбургский государственный педагогический университет, Россия; *адрес:* 460014, Россия, г. Оренбург, ул. Советская, д. 19; *e-mail:* vdjenjer@yandex.ru

L. V. Denisova, V. O. Dzhenzher

Orenburg State Pedagogical University, Russia

AROUND THE SIERPINSKI TRIANGLE. THE CHAOS GAME AND THE SCRATCH PROGRAMMING

Abstract

The article discusses the experience of studying Sierpinski geometric fractals in circle classes in mathematics and informatics with students of grades 4–6. A versatile approach to the construction of fractal objects is used: a method of cutting out of paper, Chaos Game on transparent films and implementing the studied algorithm in the Scratch programming language. At the same time, some issues related to the occurrence of random events are discussed with the children. After the fractal is obtained on a computer, the research phase of the constructed model follows. In the computer Chaos Game: (1) a small rotation by a specified number of degrees is sequentially added before setting the next point; (2) a turn in front of individual fractal vertices; (3) compression of a fractal to individual vertices. Conclusions are drawn about the self-similarity of the resulting objects. The proposed method allows us to study some properties of fractal sets, by involving visualization, intuition and computer modeling.

Keywords: Chaos Game, randomness, Sierpinski triangle, Koch curve, computer modeling, programming, Scratch.

- дать представление об исследовательской деятельности, затронув при этом понятие погрешности;
- показать роль компьютера как инструмента исследователя.

В настоящей статье мы бы хотели поделиться некоторыми результатами проведенной работы.

Прежде всего необходимо определиться с тем, в каком возрасте впервые можно начать говорить с детьми о фракталах и какие именно идеи при этом ребята способны воспринять. Для этого приведем здесь **некоторые свойства фрактальных множеств F** по [5], снабдив их собственными комментариями.

- F имеет тонкую структуру, т. е. содержит произвольно малые масштабы. Это означает, что, используя соответствующий микроскоп, мы можем увеличивать фрагмент фрактала до бесконечности, и при этом он будет иметь все ту же структуру. В то же время любой реальный объект, например лист бумаги, таким свойством не обладает. При увеличении листа мы сначала увидим волокна целлюлозы, затем отдельные клетки, молекулы, атомы... А дальше, скорее всего, будет пустота.
- F не описывается традиционным геометрическим языком. Например, из геометрии мы знаем, что окружность — это множество точек, равноудаленных от заданной точки. Описать подобным образом фрактал не представляется возможным.
- F имеет точную или приближенную форму самоподобия. То есть в F можно выделить части, каждая из которых подобна исходному множеству F примерно или точно, при этом правило построения каждой части одно и то же.
- Обычно «фрактальная размерность» множества F не равна его топологической размерности. Например, топологическая размерность точки равна нулю, прямой — единице, плоскости — двум, пространства — трем. Размерность фрактальных множеств может быть дробной. Например, размерность рассматриваемой ниже кривой Коха примерно равна 1,26. Это можно понимать так: хотя кривая Коха на первый взгляд выглядит как линия (т. е. одномерный объект), на самом деле она таковой не является, а представляет собой нечто среднее между линией и плоскостью. А вот размерность «пыли» Кантора равна примерно 0,63, что уже между множеством отдельных точек и линией.
- В большинстве случаев F определяется просто при помощи рекурсии.

Среди указанных свойств *идея самоподобия* может быть вполне воспринята даже *младшим* школьником. Действительно, многие геометрические фракталы, такие как треугольник и салфетка Серпинского, кривая Минковского, кривая дракона, кривая Коха, можно легко построить на бумаге, опираясь только лишь на свойство самоподобия.

Один из самых простых для понимания фракталов — треугольник (решето) Серпинского. Этот фрактал был придуман польским математиком Вацлавом Серпинским в 1915 году.

Существует несколько различных способов построения треугольника Серпинского [10], и очевидно, что

не все они годятся в качестве материала для изучения учащимися начальных классов. Тем не менее некоторые из способов довольно просты, а главное — наглядны.

На занятиях с младшими детьми мы стремились не просто показать очередной способ построения фрактала, а вызвать удивление ребят тем обстоятельством, что, во-первых, повторяющиеся, но, в общем-то, бессмысленные для них шаги привели в итоге к красивой фигуре или узору и, во-вторых, другая последовательность действий с другими объектами привела к тому же самому или очень похожему узору.

Перед тем как строить модель на компьютере, крайне полезно провести так называемый *ручной*, или *подготовительный*, этап.

2. Подготовительный этап

На первом занятии с учениками четвертого класса была построена **бумажная модель** треугольника по следующему алгоритму (описание алгоритма сильно упрощено):

- 1) бумажный лист сложить пополам и повернуть сгибом вниз. Получившийся прямоугольник будем называть «ступенькой»;
- 2) *посередине* сгиба сделать ножницами разрез *до середины* ступеньки;
- 3) часть слева от разреза загнуть вверх, а затем по получившейся новой линии сгиба загнуть половинку ступеньки внутрь большой ступеньки;
- 4) повторять шаги 2, 3 с каждой получившейся ступенькой.

Для лучшего понимания этого способа рекомендуем ознакомиться с видео [3].

Заметим, что на каждой следующей итерации число загибов внутрь будет удваиваться. При этом чем больше итераций будет выполнено, тем более узнаваемым будет треугольник Серпинского (фото 1). Дети, как правило, останавливаются после трех итераций, и только самые терпеливые доходят до четвертой. В любом случае учитель должен выполнить четыре итерации.

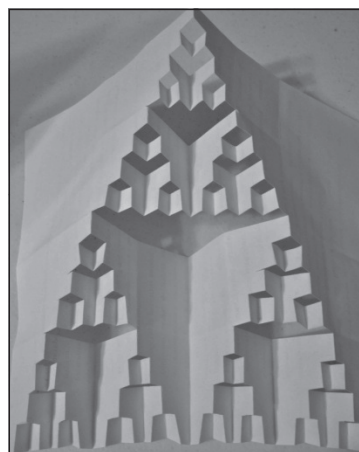


Фото 1. Бумажная модель треугольника Серпинского с четырьмя итерациями

Готовую бумажную модель нужно исследовать.

Сначала, конечно, надо обратить внимание на то, что три треугольника «по краям» являются уменьшенной копией большого (основного) треугольника. Можно

измерить высоту большого треугольника (она равна ширине листа) и заметить, что высоты треугольников по краям в два раза меньше. И то же самое можно сказать про остальные треугольники.

Через полученную модель можно пропустить свет и получить ее проекцию на потолке или стене, тогда фрактальность модели наблюдается очень заметно (фото 2). Именно в этот момент нам представляется уместным начать разговор с детьми о самоподобии.

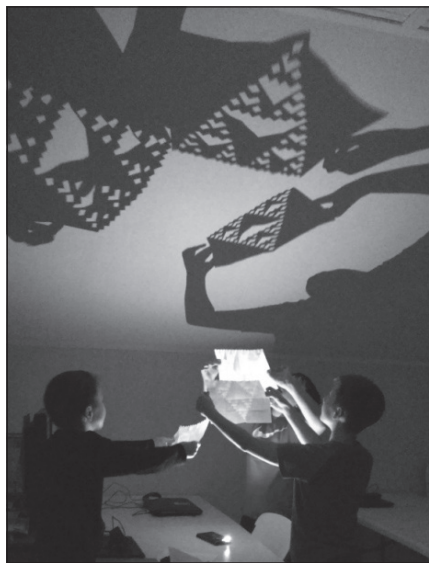


Фото 2. Проекция бумажной модели треугольника Серпинского на потолок

3. Игра в хаос

Идея следующего занятия, а именно использование прозрачных пленок для групповой работы, была позаимствована нами из статьи [8].

Каждому ученику были выданы линейка, ручка, стандартный игровой кубик и прозрачная папка-файл с нарисованными на нем точками 1, 2 и 3 в вершинах равностороннего треугольника (фото 3). На всех файлах точки нарисованы в одних и тех же местах. Задание состояло в том, чтобы подбрасывать игровой кубик и отмечать на файле точки, находящиеся посередине между текущей точкой и точкой с номером $N = (p + 1) \div 2$, где p — число, выпавшее на кубике.

Сначала текущей точкой выбирается первая. Естественно, детям формула получения числа N не приводилась; вместо этого на доске было написано соответствие между выпадавшими на кубике числами 1–6 и номерами точек:

- 1, 2 — 1;
- 3, 4 — 2;
- 5, 6 — 3.

Всего нужно было сделать не менее десяти опытов. В среднем задание выполнялось 10–15 минут.

Когда дети закончили, все их файлы были сложены в одну стопку так, что точки 1, 2 и 3 совпали. В результате мы снова увидели знакомый треугольник. Вообще говоря, хорошо просматривались результаты первой итерации — пустой перевернутый треугольник внутри большого треугольника, а на местах следующих итераций точки стояли довольно хаотично. Под итерацией

здесь понимается один шаг в построении фрактала. Это еще один из простых способов получения треугольника Серпинского.

Явно требовалось большее число точек и их более точная постановка. Сказывался малый возраст экспериментаторов: ученики четвертого класса с большим трудом делят отрезок в сантиметрах и миллиметрах пополам, и часто для простоты мы предлагали округлять результаты. Размеры самих точек также вносили свой вклад в погрешность. Все эти моменты (неточность измерения, неточность вычисления, большие размеры рисуемых точек, малое число опытов и при этом большое время работы) мы обсудили с детьми. Кроме того, мы обратили внимание детей на то, что на самом деле число опытов было больше, чем сделал каждый из них, потому что мы объединили результаты их работы. Это занятие проводилось в разных группах численностью от трех до восьми человек, поэтому и число опытов на общей картинке увеличивалось от трех до восьми раз. Но даже этого оказалось недостаточно, и всем было интересно, что же получится, если провести 1000 опытов. Но, конечно, никто из детей не изъявил желания принять участие в таком эксперименте.

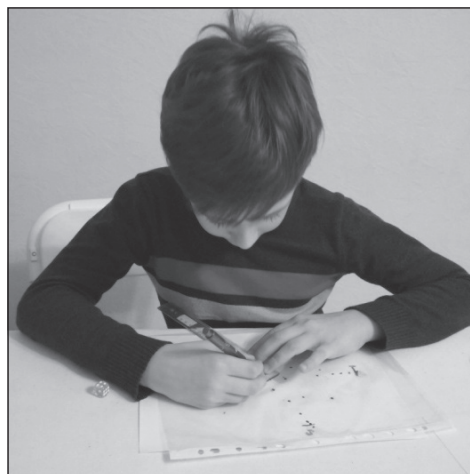


Фото 3. Второе занятие. Играем в хаос

Таким образом, мы подошли к третьему этапу — применению компьютера, который способен выполнить много опытов быстро и точно.

4. Компьютерный этап

Создатель всем известной черепашки LOGO Сеймур Пейперт считал, что, обучая детей программированию, важно, чтобы они составляли программы из команд, которые в принципе могут выполнить сами. Именно поэтому ручной этап предлагаемой серии занятий так важен. По сути дела, нужно было просто помочь детям перенести на компьютер те действия, которые выполнялись ими, когда они «играли в хаос». Для этих целей идеальным инструментом является среда Scratch [9].

Для создания проекта достаточно владеть минимальным набором команд Scratch, которые обычно изучают в самом начале знакомства с этой средой. Дополнительно потребуется только работа с переменными, а также блоки генерации случайного числа и измерения расстояния до объекта.

Прежде чем писать программу, полезно напомнить детям алгоритм «игры в хаос»: случайно выбрать одну из точек 1, 2 или 3 и отметить на середине расстояния до нее новую точку.

Можно представить такой сюжет проекта: паук поймал трех мух и хочет заплести их в паутину. Но всякий раз, как он доходит до середины пути к одной из них, одна из мух хочет улететь, и ему приходится быстрее бежать к ней.

Вначале надо завести два объекта: паука и одну мушку. Важно дать мушке имя «1», а затем продублировать ее дважды. В результате имя очередной мушки будет увеличено на единицу и мы получим три одинаковых объекта с именами от одного до трех. Всю программу выполняет паук, а мушки будут опорными точками с номерами, совпадающими с именем.

Понадобятся две глобальные переменные: *точка* и *счетчик*. В переменную *точка* на каждом шаге будет записываться результат броска кубика, а *счетчик* после выполнения очередного опыта будет увеличиваться на единицу. Таким образом, мы сможем наблюдать за изменением системы при конкретных значениях числа опытов над ней.

Следует обсудить с детьми кое-какие моменты, связанные с генерацией случайных чисел. Например, объяснить, что, хотя на настоящем кубике шесть точек, в программном блоке выгоднее выбрать диапазон 1–3, иначе код станет более сложным. Можно напомнить детям, что такие «кубики» они наверняка встречали и называются они игральными костями. Перед тем как использовать этот блок в программе, полезно просто

понажимать на него, посмотреть на результат работы и заметить, что числа иногда повторяются, а иногда нет. Спросить: когда числа повторяются чаще — при числе граней на кости, равном трем, шести или двенадцати? А при подбрасывании монетки? Отметим, кстати, что во время «игры в хаос» на прозрачных файлах неоднократно раздавались возгласы детей о том, что у них часто повторяется движение к одной и той же точке. Об этом им тоже полезно напомнить.

На рисунке 1 представлены готовая программа и результат ее работы после остановки на значении счетчика, большем 100. Поясим, что в команде *вернуться к <объекту>* обычно имя объекта выбирается из выпадающего списка. Но в данном случае использование переменной *точка*, в которую записывались результаты броска кубика, оказывается очень удобным. То же самое относится и к блоку *расстояние до <объекта>*. Перед использованием этого блока обязательно нужно понажимать на него и посмотреть, что за числа он выдает при изменении положения паука.

Возвращаясь к рисунку 1, отметим, что размер пера здесь равен 10. Примерно такую картину мы наблюдали с детьми на сложенных в стопку файлах.

Оказывается, проведенная детьми ручная работа была не так уж и плоха. Если же проводить опыты с прозрачными файлами в классе численностью 20–30 человек, то общее число опытов возрастет до 300–450 и картина будет выглядеть примерно, как на рисунке 2. Результат более удовлетворительный, но сильно мешает большой размер точек, поэтому все согласились уменьшить размер пера до единицы. Наконец, необходимо

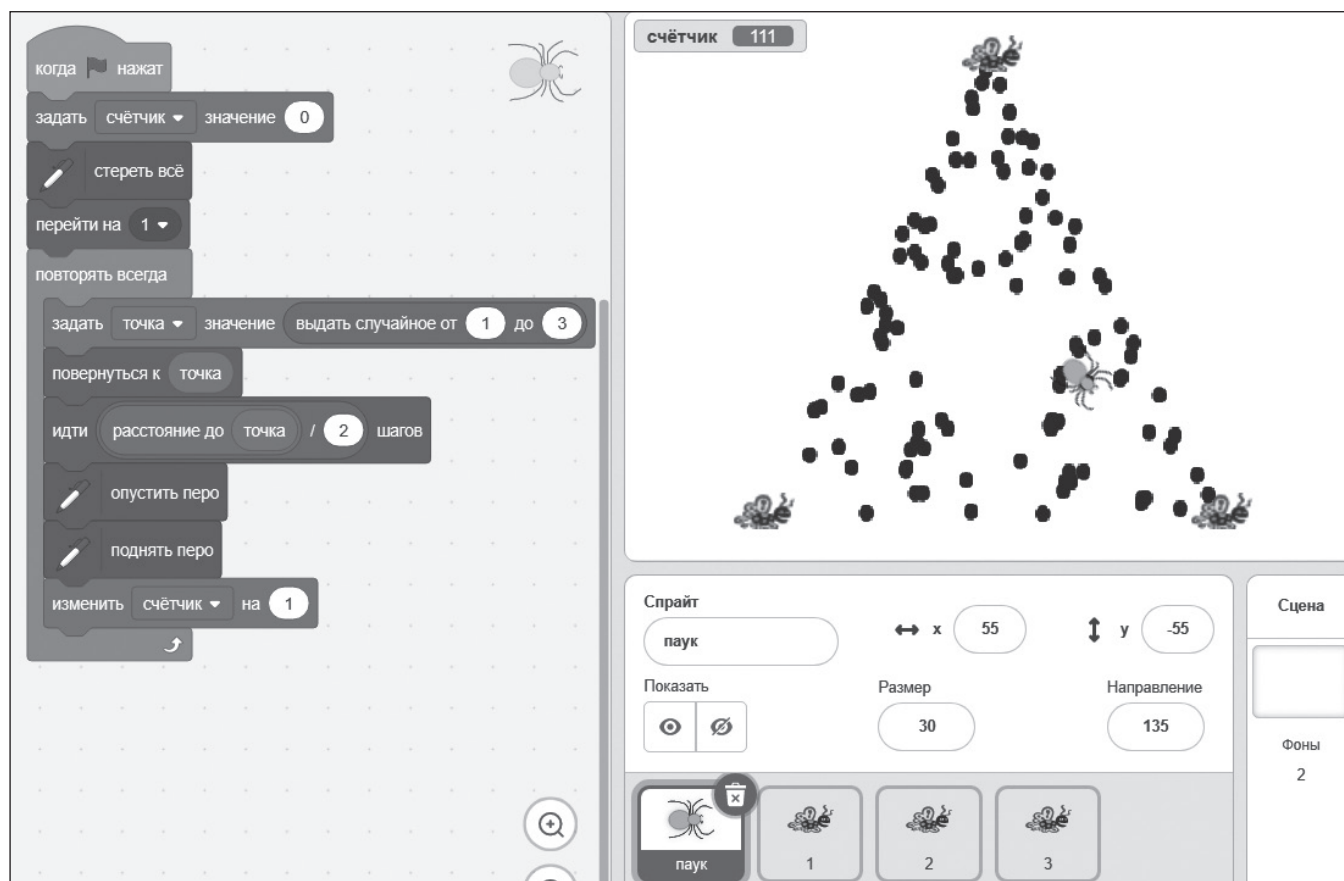


Рис. 1. Программа «игры в хаос» и результат ее работы при 111 опытах

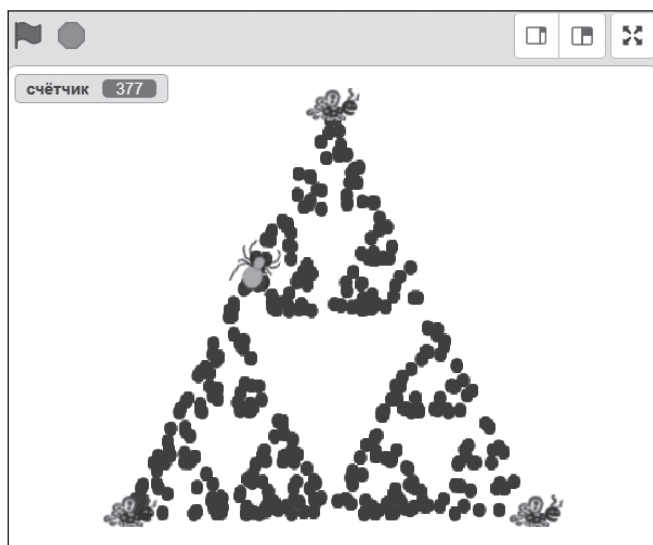


Рис. 2. Результат работы программы при 377 попытках с крупным пером

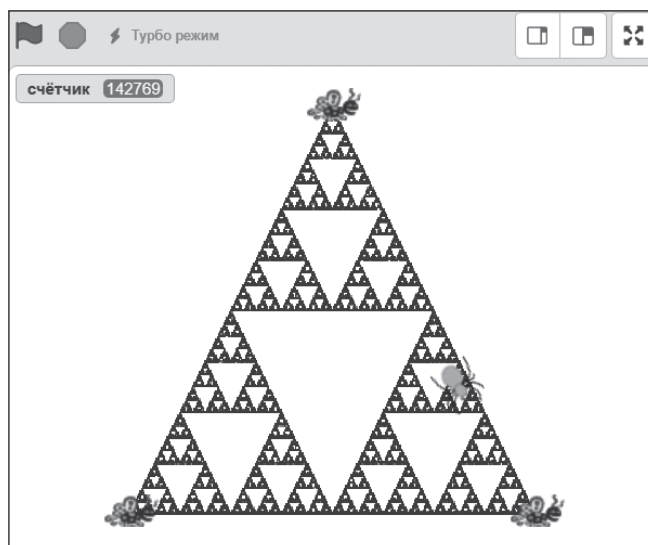


Рис. 3. Результат работы программы в турбо режиме с малым пером

значительно увеличить число опытов, и здесь на помощь приходит *Турбо режим*, который включается через меню *Редактировать* (рис. 3). Восхищение детей трудно передать словами!

Здесь можно достать старую бумажную модель, еще раз исследовать ее и постараться выявить некоторую общую операцию, выполняемую детьми и при работе с бумажной моделью, и при «игре в хаос», и при построении компьютерной модели. Возможно, не сразу, но некоторые ребята все же замечают, что *общим было деление пополам*: каждый раз бумажный лист разрезали посередине и до середины, а точки на файлах отмечались на середине расстояния между точками, и на такое же расстояние перемещался паук.

5. Развитие проекта

Главное — это не построение очередного фрактала, а его дальнейшее исследование. Развитие проекта может идти по следующим направлениям.

Прежде всего отметим, что изменение положения мушек сохраняет треугольник и его самоподобие. Менять положение мушек можно даже во время работы программы, получая интересные картины.

Далее можно добавить в программу команду *повернуть на <число> градусов* после команды *повернуться к <точка>* и исследовать поведение модели

при различных углах поворота. На рисунке 4 показаны результаты при углах в 5, 15 и 25 градусов. Как видно, самоподобие сохраняется.

Если занятие проводится в V—VI классах (или с сильной группой четвероклассников), то можно добавить новое правило: если на кубике выпала точка 1, то сдвигаться на $2/3$ расстояния, а для точек 2 и 3 по-прежнему на половину (рис. 5). То есть треугольник как бы сжимается к верхней вершине. Интересный результат получается, если при движении к первой вершине добавит небольшой поворот (рис. 6). Далее дети могут придумать свои правила для каждой из вершин.

В какой-то момент от детей обязательно поступает предложение увеличить количество точек до четырех. Поставить такой эксперимент очень просто — делаем копию исходной программы, меняем количество точек на кубике на 4, а затем дублируем одну из точек и составляем точки в вершинах квадрата. Запуск программы не оправдывает ожидания — точками заполняется вся внутренность квадрата.

Чтобы построить квадрат (ковер) Серпинского, необходимо еще раз изучить устройство треугольного фрактала и заметить, что можно по-другому объяснить его построение: делим большой треугольник на четыре одинаковых (подобных) треугольника, средний удаляем, а с тремя оставшимися поступаем аналогично. Тогда очевидно, что квадрат надо будет разделить на

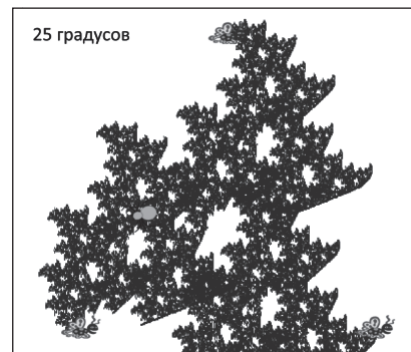
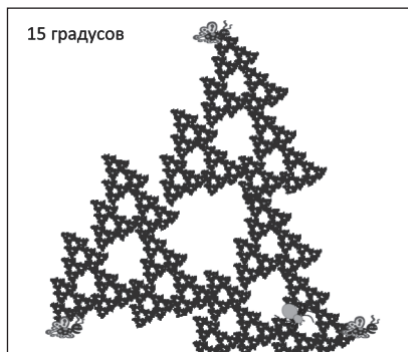


Рис. 4. Самоподобие сохраняется при небольшом повороте

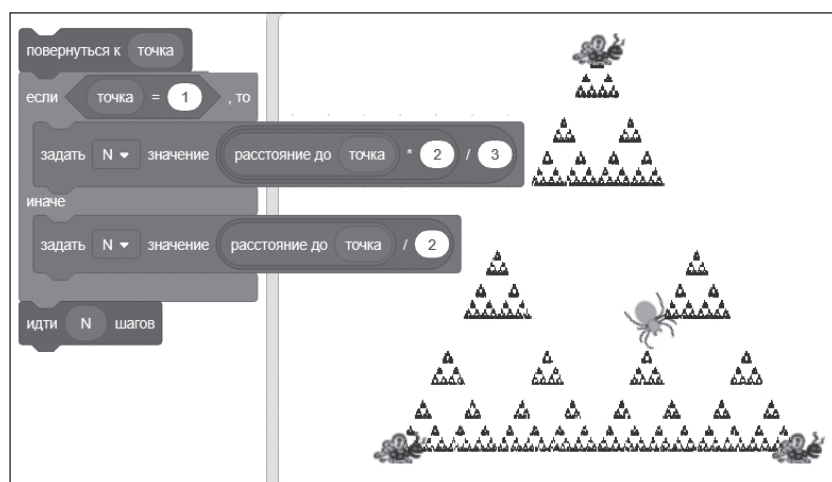


Рис. 5. Самоподобие сохраняется при сжатии к одной из вершин

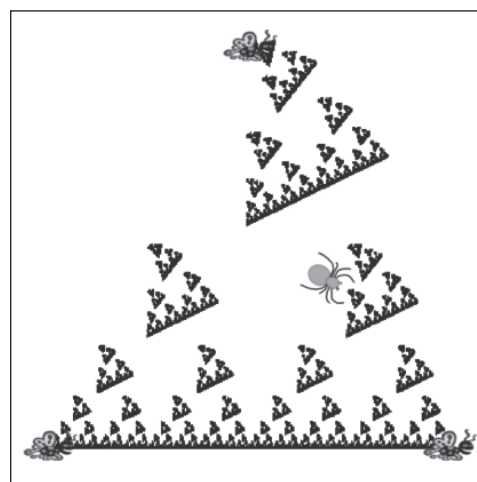


Рис. 6. Самоподобие сохраняется при сжатии и повороте к одной из вершин

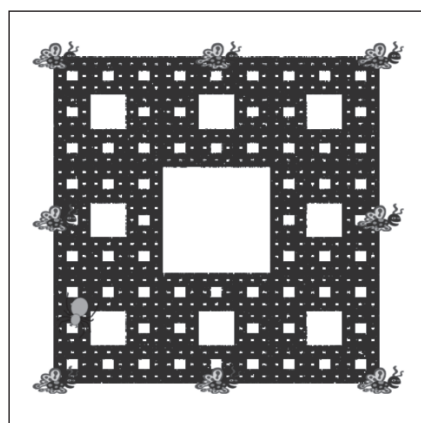


Рис. 7. Ковер Серпинского

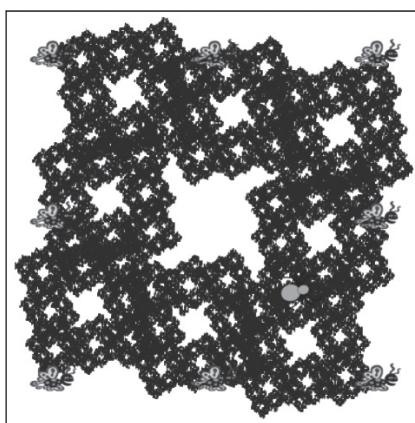


Рис. 8. Ковер Серпинского с поворотом

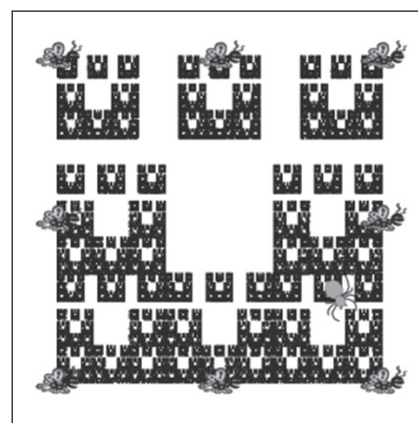


Рис. 9. Ковер Серпинского со сжатием вверх

девять одинаковых квадратов, центральный удалить, а с остальными процедуру повторить. При этом станет видно, что сдвиг на половину расстояния ничего не даст, паучок окажется точно в середине квадрата, а она должна остаться пустой. Тут можно подсказать, что сдвигаться нужно на $2/3$ расстояния до вершины. Проведенный эксперимент снова не дает квадрат Серпинского, хотя получаемая фигура будет фракталом. Дело в том, что необходимы восемь точек, четыре из которых стоят в вершинах квадрата, а еще четыре — на серединах его сторон (рис. 7). В эту игру также можно добавить небольшой поворот (рис. 8) и сжатие: на рисунке 9 показан пример, когда сдвиг к трем верхним вершинам происходил на $3/4$ расстояния, а к остальным — на $2/3$.

Техническое замечание по рисунку 9. Для того чтобы выбрать нужные вершины из восьми, мы использовали полное ветвление *если <условие> то ... иначе...* В качестве условия можно использовать стандартное логиче-

ское выражение, составленное при помощи операторов *или*. Но в Scratch такое составное условие получается очень громоздким и трудным в сборке. Вместо этого удобно использовать строку, в которой будут храниться номера «особенных» вершин (в нашем примере это вершины 1, 2 и 3), и проверку *<123> содержит <точка>?* из меню *Операторы* (рис. 10).

Другие интересные фракталы представлены на рисунках 11, 12. При числе вершин, равном пяти, смещение происходит на $5/8$ расстояния, а при числе вершин, равном шести, — на $2/3$ расстояния. Заметим, что здесь наибольшую трудность у детей вызывает расстановка точек в вершинах правильных фигур. Поэтому эти фракталы желательно строить после занятия на построение правильных многоугольников и узоров из них. Интересно, что внутренняя часть шестиугольника Серпинского представляет собой другой известный фрактал — снежинку Коха.

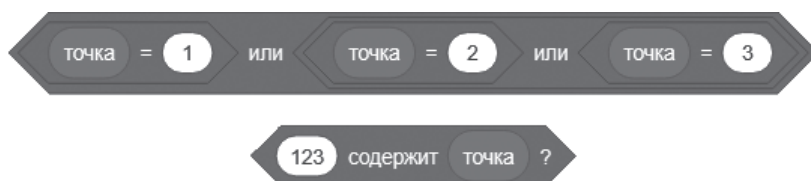


Рис. 10. Логическое условие сверху равносильно условию снизу

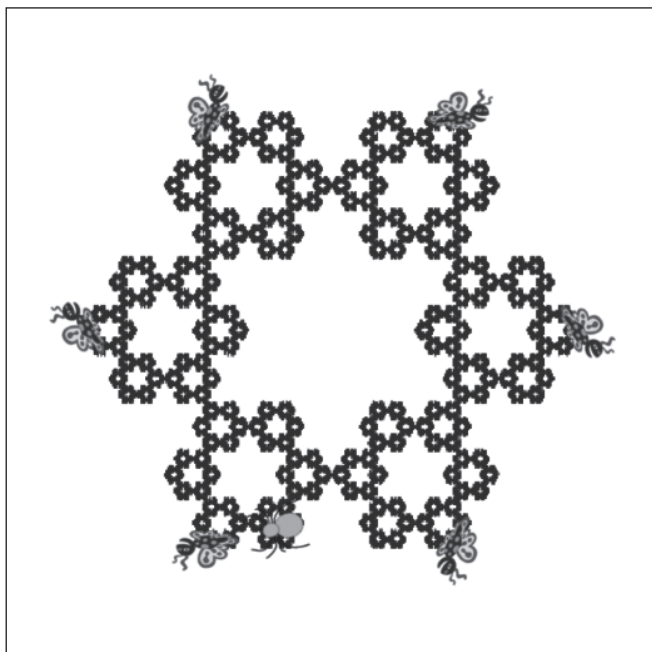


Рис. 11. Пятиугольник Серпинского

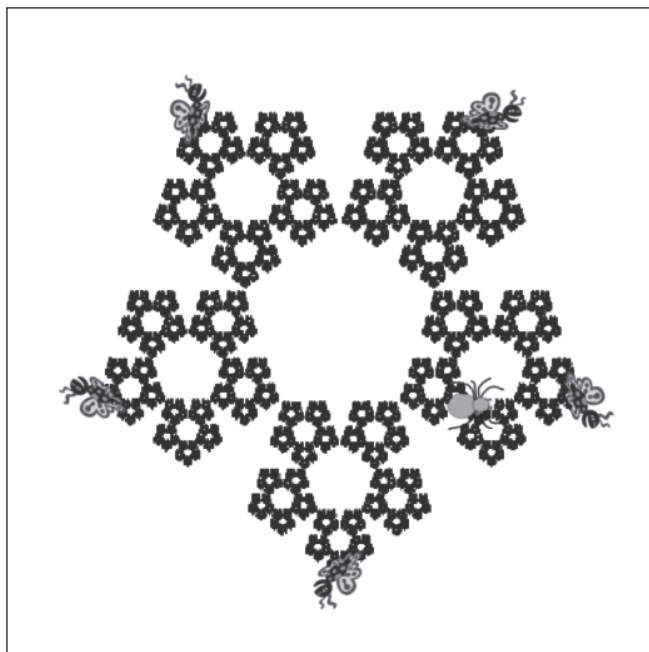


Рис. 12. Шестиугольник Серпинского

6. Заключение

К сожалению, «в наше время большинство учеников заканчивают школу с прочной ненавистью к математике и почти никто из них не понимает, о чем она» [1, с. 38]. Такая прискорбная ситуация требует каких-то действий, направленных на изменение существующего положения. Один из способов сделать это нам видится в поиске тем, демонстрирующих красоту и простоту математических идей, а также позволяющих в итоге создать компьютерную модель изученного. Фрактальные объекты интересны тем, что позволяют прикоснуться к себе не только профессиональным математикам, но и детям.

Использование языка программирования Scratch дает в руки исследователя мощный и удобный инструмент, с помощью которого утомительная механическая процедура превращается в короткую программу. Причем именно после написания кода обычно возникают вопросы «а что, если...», которые и приводят к дальнейшему развитию проекта. Математика не может быть скучной, а математика с компьютером — тем более!

Все проекты, рассмотренные в статье, доступны в галерее проектов [7].

Список использованных источников

1. Берендс Э. Математические пятиминутки. М.: БИНОМ. Лаборатория знаний, 2015. 376 с.
2. Вельтман А. Математика — это красиво! Графическая тетрадь № 2. М.: Манн, Иванов и Фербер, 2018. 96 с.
3. Вырезаем елку-фрактал. <https://www.youtube.com/watch?v=-sWLdWLcw1U>
4. Людовицкая А. Математическая пицца. М.: Манн, Иванов и Фербер, 2018. 80 с.
5. Морозов А. Д. Введение в теорию фракталов. М., Ижевск: Институт компьютерных исследований, 2002. 160 с.
6. Понятная математика с Кристин Даль и Свеном Нурдквистом. Игры, загадки, рисунки. М.: Альбус корvus, 2017. 64 с.
7. Студия Scratch — Фракталы Серпинского. <https://scratch.mit.edu/studios/25808724/>
8. Robert L. Devaney. Chaos in the Classroom. <http://math.bu.edu/DYSYS/chaos-game/chaos-game.html>
9. Scratch — Imagine, Program, Share. <https://scratch.mit.edu>
10. Sierpiński triangle. https://en.wikipedia.org/wiki/Sierpiński_triangle#Analogues_in_higher_dimensions