

Л. В. Денисова, В. О. Дженжер,
Оренбургский государственный педагогический университет

ЛЮБИТЕЛЯМ СРЕДЫ ПРОГРАММИРОВАНИЯ SCRATCH: МАТЕМАТИЧЕСКОЕ ВЫШИВАНИЕ АСТРОИДЫ

Аннотация

В статье показан способ построения кривых как огибающих семейства прямых в среде Scratch. Наглядность способа позволяет применять его на уроках и во внеурочной деятельности по информатике в V–VI классах.

Ключевые слова: алгоритм, исполнитель, синхронизация работы исполнителей, компьютерное моделирование, программирование, Scratch, математическое вышивание.

DOI: 10.32517/2221-1993-2019-18-3-26-31

Введение

Среда Scratch [5] становится все более популярным инструментом для пропедевтического обучения основам программирования детей младшего школьного возраста. В дополнительном образовании курс зачастую строится как серия мастер-классов по программированию игр. В школьном же образовании большее применение находят уроки по рисованию пером, так как это очень напоминает известную черепашку Logo. Использование пера позволяет сосредоточиться именно на алгоритмической сущности заданий, что крайне важно в пропедевтическом плане. Этот подход применяет, например, Т. Е. Со рокина [6] на занятиях с учениками пятого класса.

Рисовать пером можно как с явным обращением к координатам точек, так и без него. Школьники обычно знакомятся с понятием системы координат на уроках математики в шестом классе. Это позволяет им использовать команды Scratch `перейти в x [ ] y [ ]`, `изменить x на [ ]`, `изменить y на [ ]` в своих проектах. Но построение сложных кривых на экране компьютера без знакомства с понятием функции не представляется возможным. Поэтому для V–VI классов более предпочтителен вариант, при котором построение кривой происходит путем смещения объекта относительно предыдущего положения. Причем это не «смещение на вектор», используемое в исполнителе Чертежник, а простые команды типа «идти вперед» и «повернуть на угол».

Существует такой способ понятного и наглядного представления различных кривых, как «математическое вышивание». Подобный термин встречается, например, в учебнике по наглядной геометрии для V–VI классов [4]. В сети Интернет можно встретить большое количество школьных проектов на эту тему по математике и технологии и даже художественных работ в технике «изонить». Авторы помнят, как в то «докомпьютерное» время, когда люди посылали друг другу настоящие, а не виртуальные открытки, школьники вырезали из них «шестеренки» с различным числом зубцов и наматывали на них нитки, последовательно переходя от одного зубца к другому. Меняя расстояние между связанными зубцами, можно было получать различные узоры, очень напоминающие узоры спирографа. Будучи школьниками, мы не знали, что этот **метод основан на математическом определении кривых как огибающих семейства отрезков**.

Например, **астроида является огибающей семейства отрезков постоянной длины, концы которых расположены на двух взаимно перпендикулярных прямых**.

Красота получаемых в такой технике кривых дает учителю дополнительное средство привлечения интереса детей к математике и программированию.

Далее мы **рассмотрим один из вариантов организации занятия по программированию в среде Scratch «Математическое вышивание астроиды»**.

Контактная информация

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

L. V. Denisova, V. O. Dzhenzher,
Orenburg State Pedagogical University

FOR THE FANS OF THE SCRATCH PROGRAMMING LANGUAGE: MATHEMATICAL EMBROIDERY OF ASTROID

Abstract

The article shows a method for constructing curves as envelopes of a family of lines in the Scratch environment. The clarity of the method allows it to be used in the classroom and in extracurricular activities in computer science in grades 5–6.

Keywords: algorithm, performer, synchronization of work of performers, computer simulation, programming, Scratch, mathematical embroidery.

К занятию школьник уже должен уверенно работать в среде Scratch. Необходимо знать и владеть следующими понятиями и командами:

команды движения:

- идти <N> шагов;
- плыть к <объект>;
- повернуться к <объект>;
- установить стиль вращения;

команды управления:

- различные циклы;
- условный оператор;

события:

- передача и обработка сообщений;

сенсоры:

- считывание свойств других объектов;

переменные:

- задание значений переменным;
- увеличение и уменьшение значений числовых переменных;

операторы:

- арифметические;
- сравнение;
- логическое «или»;

перо:

- опустить / поднять перо;
- изменить цвет пера;

другие блоки:

- желательно уметь создавать процедуры с параметрами, но занятие можно построить и без применения процедур. Общий смысл и цель занятия при этом сохраняются.

Построение кривых должно начинаться с их рисования методом математического вышивания на бумаге.

1. Бумажный этап

- На клетчатой бумаге построить две взаимно перпендикулярные прямые — вертикальную и горизонтальную;
- от точки пересечения прямых в каждую из четырех сторон (вверх, вниз, влево, вправо) через равные промежутки отметить по шесть точек, обозначая их порядковыми номерами от одного до шести; точка ноль считается общей;
- в каждой четверти соединить отрезками точки, дающие в сумме число 6.

2. Компьютерный этап

Попытка сразу перенести «бумажный» алгоритм построения в среду Scratch, вероятнее всего, не приведет к успеху. Как правило, ученики V—VI классов еще очень слабо понимают принципы работы в прямоугольной декартовой системе координат, особенно во всех четырех четвертях. Мы стремимся к тому, чтобы при решении нашей задачи приходилось как можно меньше обращаться к этим понятиям.

Поэтому полезно сначала рассмотреть с ребятами только правую верхнюю четверть полученного рисунка и задать им такой вопрос: «Как можно описать алгоритм построения этой фигуры при условии, что точки не имеют подписей?» Как правило, ребята сами

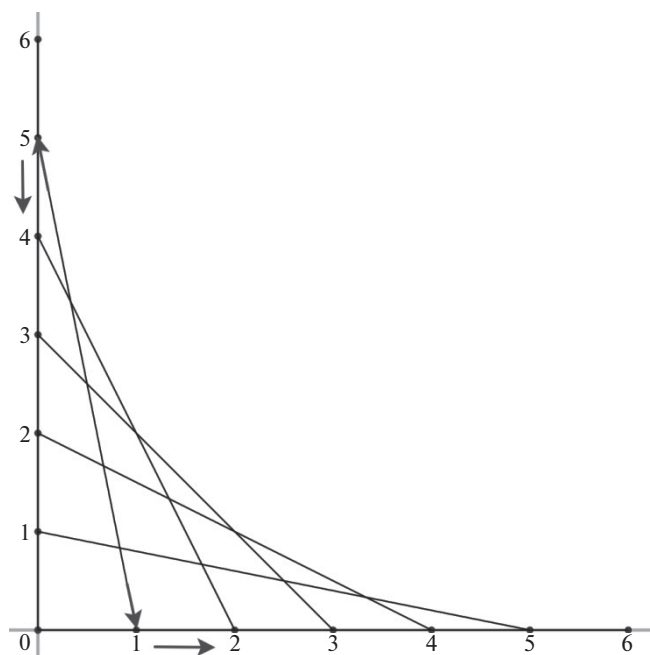


Рис. 1. «Вышивание» астроиды в первой четверти для шести точек

в состоянии сформулировать ответ, пусть и не очень точно (рис. 1):

- а) выбрать самую верхнюю точку на оси Y и самую левую на оси X (для первой четверти — центр системы координат);
- б) соединить выбранные точки отрезком;
- в) сместить точку на оси Y на одно деление вниз, а на оси X — на одно деление вправо;
- г) повторять шаги б и в шесть раз (по числу точек). Именно этот алгоритм мы и реализуем для начала.

2.1. Построение первой четверти астроиды

Итак, нам понадобятся три исполнителя. Два объекта будут двигаться по осям X и Y . Назовем их B и G от слов «вертикаль» и «горизонталь». Это могут быть просто шарики из библиотеки объектов Scratch, размер которых нужно уменьшить до 20 %. Третий объект — *Карандаш*, который будет соединять отрезками точки B и G . Центр костюма *Карандаша* нужно установить на грифеле.

Также нам понадобится фон с системой координат. В библиотеке фонов он называется *xy-grid*. Центр системы координат совпадает с центром экрана. Поместим *вручную* точку B как можно выше на ось Y , а точку G — в начало системы координат. При запуске программы точка G должна повернуть направо, а точка B — вниз, т. е. в направлении своего будущего движения.

Работу нескольких исполнителей нужно согласовывать: *Карандаш* должен соединять точки только тогда, когда они заняли свои позиции; и наоборот, точки могут переходить в новое положение только тогда, когда *Карандаш* закончил свой фрагмент работы. То есть нам нужен «дирижер» или «директор строительства». Роль дирижера можно поручить одному из исполнителей, например *Карандашу*. Каждый раз после того, как *Карандаш* соединит две точки линией, он будет передавать сообщение «Далее» и ждать, пока это сообщение будет обработано всеми получателями. Точки B и G , получив сообщение, перейдут в новые позиции.

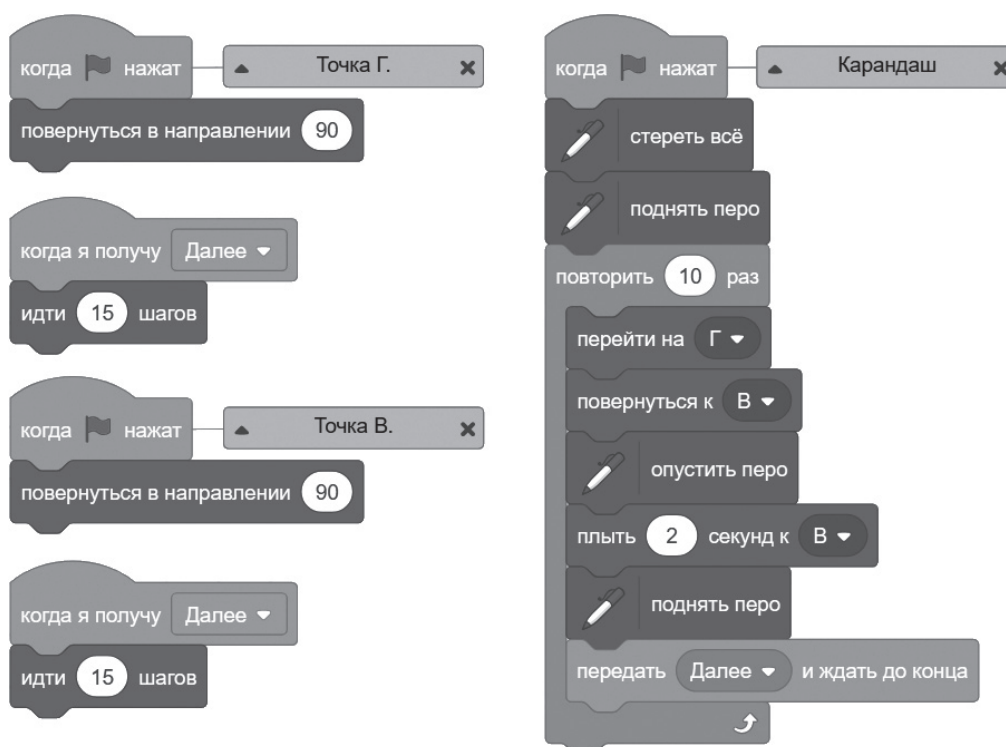


Рис. 2. Первый вариант программы

Это первый, самый простой вариант рисования одной четверти астероида (рис. 2).

Следует обратить особое внимание учеников на существенную разницу в работе команд **передать <сообщение>** и **передать <сообщение> и ждать до конца**. Первая команда передает сообщение и сразу же пере-

ходит к следующей команде скрипта. При этом неважно, обработано сообщение или еще нет. Вторая команда работает несколько иначе: после отправки сообщения выполнение скрипта приостанавливается, и следующая команда будет выполнена только тогда, когда сообщение будет обработано всеми адресатами. В нашем случае

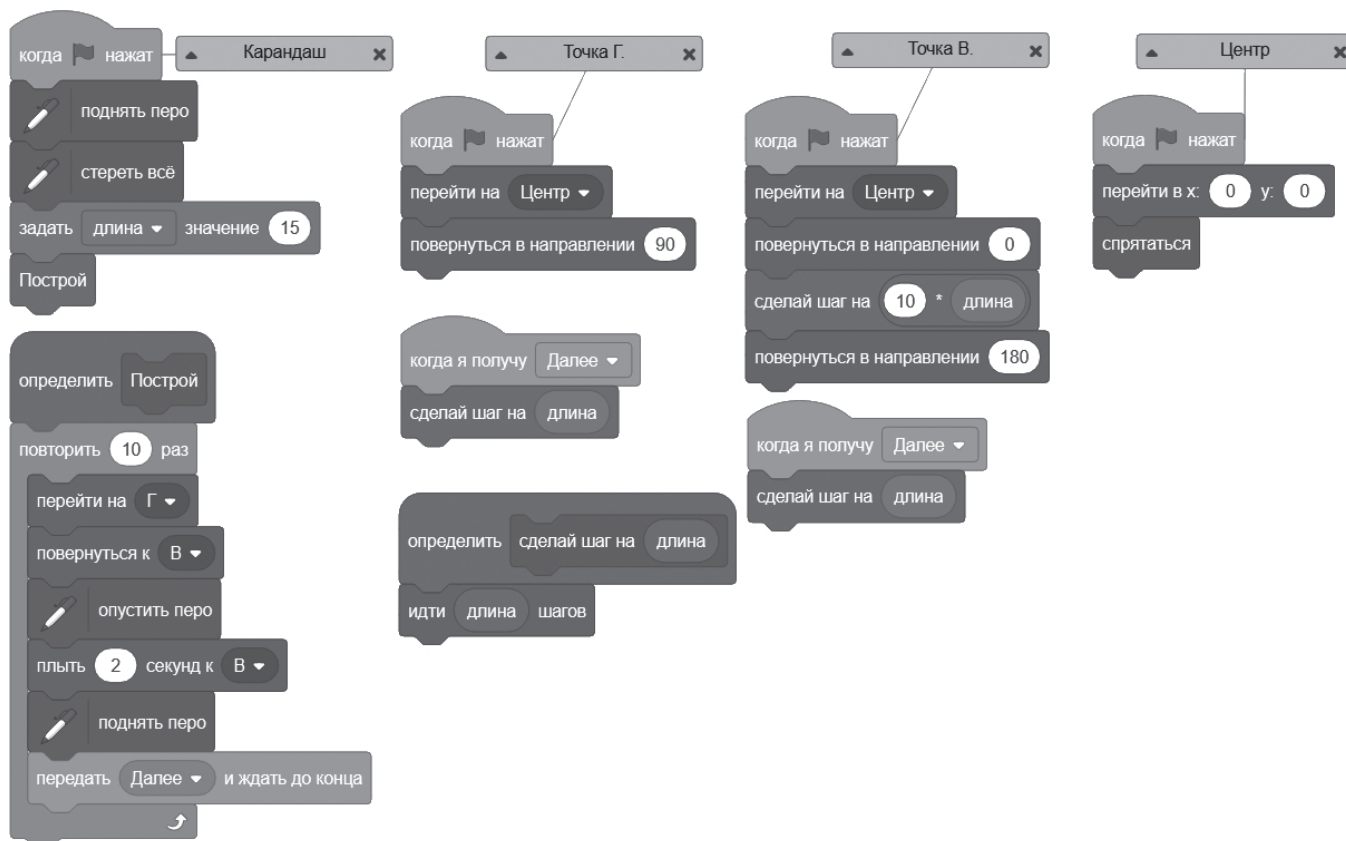


Рис. 3. Автоматическое задание начальных положений

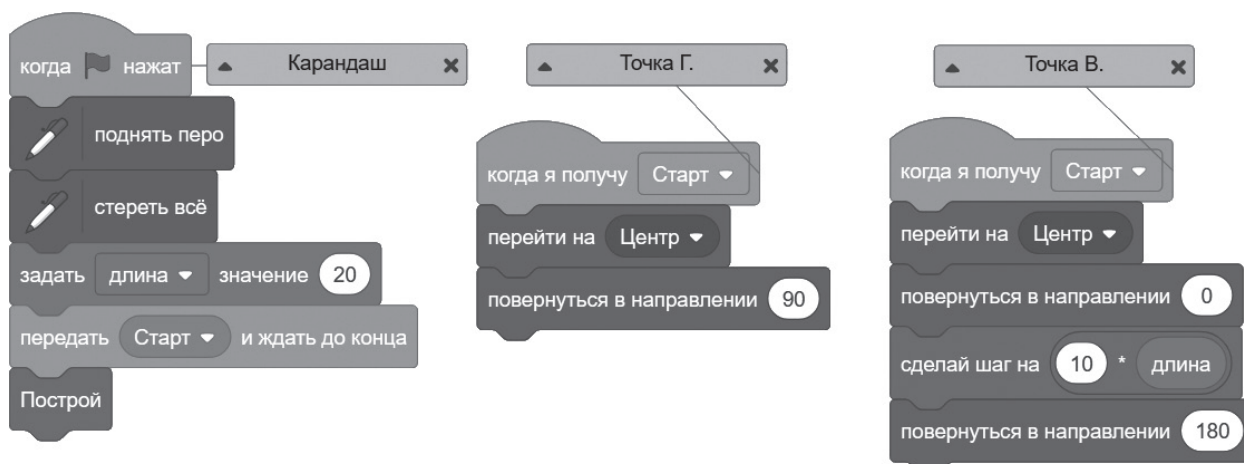


Рис. 4. Измененные фрагменты кода для согласования действий исполнителей

очередная итерация цикла начнется только после того, как точки займут новые позиции.

Для удобства чтения кода оформим процесс построения части кривой в виде процедуры **Построй**. Затем добавим в программу автоматическое задание начального положения точек *В* и *Г*. Нам понадобится новый объект *Центр*, программа которого будет крайне простой: перейти в центр экрана — точку (0; 0). Создадим переменную «длина», задающую смещение точек *В* и *Г* на координатных осях за один шаг. Для более полной аналогии с «бумажным» алгоритмом создадим процедуру **сделай шаг на <длина>**. Процедуры идентичны для обеих точек. При запуске программы обе точки встанут в *Центр*, а затем точка *В* должна сделать 10 шагов вверх. Получим код, как на рисунке 3.

Однако запуск этой программы выявляет **несогласованность действий исполнителей**. Точки еще не успели встать в начальные позиции, а *Карандаш* уже соединяет их. В результате первой итерации цикла линия рисуется не там, где нужно. Особенно это заметно, если программу запустить, через некоторое время досрочно остановить, а затем запустить еще раз.

Решить эту проблему нам снова поможет передача сообщений. Пусть *Карандаш* передает сообщение

«Старт» и ждет, пока это сообщение будет обработано точками *В* и *Г*. У точек следует заменить блоки **Когда флажок нажат** на **Когда я получу <Старт>** (рис. 4).

2.2. Построение полной астроиды

Теперь, когда мы научились строить часть астроиды, подумаем, как достроить ее до конца. Мы рассмотрим здесь два подхода.

«Математический» подход.

Очевидно, *Карандашу* нужно четыре раза сделать почти одно и то же — нарисовать очередную часть астроиды. Но построение должно происходить в разных четвертях. Заметим, что начальное положение точки *Г* всегда одно и то же — центр экрана. Она должна двигаться слева направо, когда построение происходит в первой и четвертой четвертях, и справа налево — во второй и третьей. Точка *В* движется сверху вниз при построении в первой и второй четвертях и снизу вверх — в третьей и четвертой.

Заведем счетчик четвертей. Вначале счетчик должен равняться единице; он должен увеличиваться на 1 каждый раз, когда будет построена очередная четверть кривой. После этого *Карандаш* вновь посылает сообщение «Старт». Начальное положение и направление даль-



Рис. 5. Измененные фрагменты программы для построения астроиды

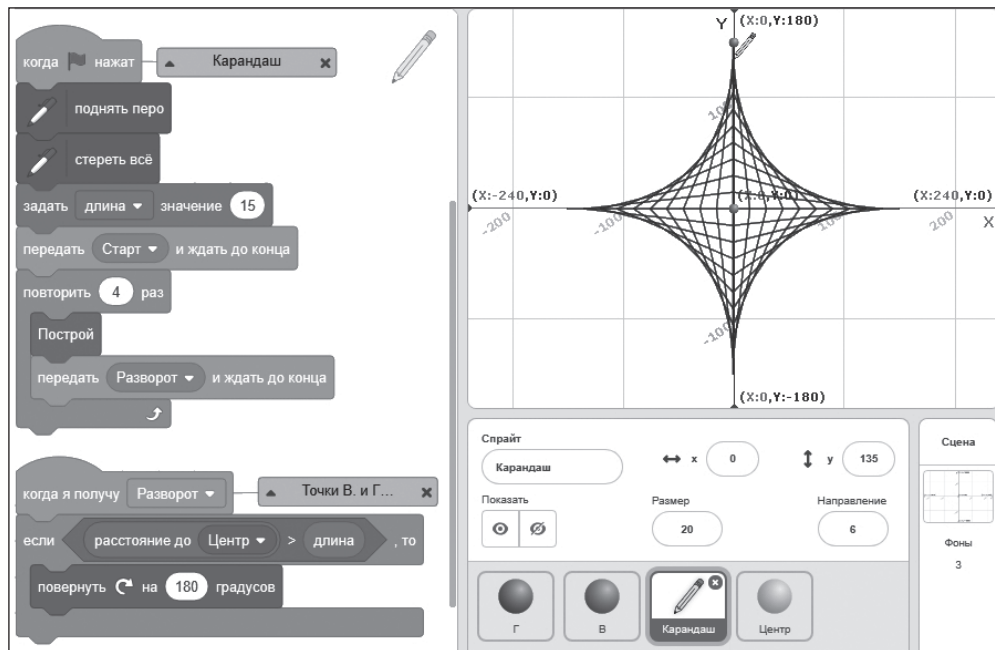


Рис. 6. «Физический» подход к решению задачи и результат работы программы

нейшего движения точек *В* и *Г* выбирается в зависимости от значения счетчика (рис. 5).

В результате астроида полностью построена, но программа стала более громоздкой. Попробуем ее упростить. Для этого вернемся к программе на рисунке 4 и посмотрим на процесс построения с другой точки зрения.

«Физический» подход.

Попробуем запустить процедуру **Построй** два раза. Мы заметим, что после построения первой четверти кривой точки *В* и *Г* будут продолжать движение в тех же направлениях, а *Карандаш* будет по-прежнему рисовать линии от точки *К* к точке. И все было бы хорошо, вот только точку *Г* во время второго захода надо развернуть в противоположную сторону. Можно представить себе, что точка *Г* привязана к пружинке и колеблется вдоль горизонтали около центра системы координат. Такое же, только вертикальное, движение должно быть и у точки *В*.

Будем считать, что, как только пружинка растянулась на определенную длину, которую мы отсчитываем от центра, она начнет сжиматься. Добиться такого поведения можно очень просто. В конце рисования каждой четверти точки получают сообщение «Разворот». Заметим, что точки *В* и *Г* находятся в это время в разных положениях: одна — в центре системы координат, а вторая — на максимальном удалении от центра. Развернуться должна только та точка, которая ушла на максимальное расстояние. Мы добавляем в программу проверку на удаленность от центра: если расстояние до центра больше, чем длина шага, то разворачивайся на 180° и продолжай движение в обратную сторону. Точка, которая расположена в центре, при этом сохранит прежнее направление движения (рис. 6).

Обратим внимание, что передача сообщения «Старт» происходит в этом варианте программы только один раз, так как точки занимают стартовые позиции единожды.



Рис. 7. Изменения в окончательном варианте программы

Завершающий штрих — зададим возможность изменения частоты «стежков». Для этого введем переменную N . Длина шага тогда будет рассчитываться как $150/N$. Число 150 определяется вертикальным размером сцены Scratch — 180 точек от центра экрана вверх и вниз. Мы берем его немного меньше, чтобы не касаться границы экрана. Окончательный вариант программы приведен на рисунке 7.

Замечание.

Построенная кривая, вообще говоря, не является астроидой. Астроида, согласно определению, огибает семейство *отрезков постоянной длины*. В нашем же случае длина соединяющего отрезка меняется на каждом шагу. Мы сознательно пожертвовали точностью изложения для того, чтобы сохранить ясность и понятность кода для учеников V—VI классов. Но познакомить школьников с настоящей астройдой можно на сайте математических этюдов [1, 3].

Школьникам старшего возраста, уже знакомым и с системой координат, и с теоремой Пифагора, можно предложить построить астроиду, а не «псевдоастроиду». При этом можно обойтись всего одним объектом, что снимает все те проблемы, которые мы решали в этой статье (но поднимает другие). Все заинтересованные могут ознакомиться с таким вариантом программы по ссылке: <https://scratch.mit.edu/projects/298663978/>

Заключение

Авторы имеют многолетний опыт преподавания внеурочных занятий по программированию в среде Scratch как в школе, так и в дополнительном образовании. Предложенный способ построения кривых был опробован нами в 2018/2019 учебном году во внеурочной работе со школьниками пятых классов. В течение первого полугодия дети познакомились с основными возможностями среды Scratch. Возможный план занятий представлен в пособии

по организации проектной деятельности [2]. Во втором полугодии начали изучать набор команд «Перо». После того как были освоены базовые принципы работы с таким исполнителем, были введены понятия процедуры и процедуры с параметрами с целью рисования масштабируемых узоров. С этими же ребятами на математическом кружке параллельно занимались исследованием кривых, получаемых в результате различных способов пересечения отрезков. Так возникла идея соотнести полученный ребятам математический опыт с программированием. Проведенные занятия вызвали заметный интерес школьников.

Предложенный способ построения некоторых кривых методом математического вышивания в среде Scratch может быть встроен в пропедевтический курс «Алгоритмика и программирование» для школьников V—VII классов.

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

Список использованных источников

1. Астроида // Сайт «Математические этюды». <http://www.etudes.ru/ru/sketches/astroid/>
2. Дженджер В. О., Денисова Л. В., Рындак В. Г. Проектная деятельность школьника в среде программирования Scratch: методическое пособие. Оренбург: Оренб. гос. ин-т менеджмента, 2009. <http://sites.google.com/site/orensratch/nasi-razrabotki>
3. Котенок на лестнице // Сайт «Математические этюды». <http://www.etudes.ru/ru/sketches/cat-on-ladder/>
4. Математика: наглядная геометрия: учеб. пособие для 5—6 кл. общеобразоват. учреждений / В. А. Панчишина, Э. Г. Гельфман, В. Н. Ксенева и др. М.: Просвещение, 2006.
5. Сайт языка программирования Scratch. <http://scratch.mit.edu>
6. Сорокина Т. Е. Поурочные разработки к программному модулю по информатике для V класса «Пропедевтика программирования со Scratch» // Информатика в школе. 2015. № 10.
7. Студия «Астроида» на сайте Scratch. <https://scratch.mit.edu/studios/6248536/>

НОВОСТИ

Контент МЭШ будет доступен всем желающим в Москве, в России и в мире

Контент «Московской электронной школы» (МЭШ) станет доступен всем желающим: не только ученикам и педагогам столицы, но и жителям других регионов страны. Об этом сообщил мэр Москвы Сергей Собянин.

Проект «Московская электронная школа» стартовал три года назад. «Мы сделали первый шаг, обновив технику и оборудование в школах, — сказал Собянин. — Сейчас в систему загружены десятки тысяч сценариев уроков, массивы данных с каждым днем растут в геометрической прогрессии. Следующий этап — сделать

нашу систему открытой». Мэр подчеркнул: доступными для всех сделают электронные сценарии уроков, интерактивные приложения, пособия и тесты, разработанные учителями. Исключением станут учебные материалы, защищенные авторскими правами. Мобильное приложение с контентом «Московской электронной школы» появится в App Store или Google Play. Скачать его смогут все желающие. Пользователям из других регионов и стран разрешат также загружать в библиотеку проекта свои материалы.

(По материалам «Российской газеты»)