

С. Н. Лосик,

школа № 45 имени Л. И. Мильграма, г. Москва

ПРОПЕДЕВТИКА ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В НАЧАЛЬНОЙ ШКОЛЕ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ KODU GAME LAB

Аннотация

В статье рассматривается опыт преподавания пропедевтического курса параллельных вычислений в третьем классе общеобразовательной школы с использованием среды Kodu Game Lab. Данный курс опирается на УМК «Информатика для всех» для I–IV классов начальной школы Д. И. Павлова под редакцией А. В. Горячева. Приводятся примеры заданий, которые ученики выполняли на уроках.

Ключевые слова: параллельные вычисления, УМК по информатике, программирование, «Информатика для всех», статические объекты, динамические объекты.

DOI: 10.32517/2221-1993-2018-17-7-36-39

Контактная информация

Лосик Сергей Николаевич, учитель информатики школы № 45 имени Л. И. Мильграма, г. Москва; адрес: 117447, г. Москва, Новочеремушкинская ул., д. 11А; телефон: (499) 129-71-55; e-mail: losiksn@ms45.edu.ru

S. N. Losik,
School 45, Moscow

PROPAEDEUTICS OF PARALLEL COMPUTING IN THE PRIMARY SCHOOL USING THE KODU GAME LAB

Abstract

The article discusses the experience of teaching a propaedeutic course of parallel computing in the 3rd grade of primary school using the Kodu Game Lab environment. This course is based on the textbook set "Informatics for All" by D. I. Pavlov for grades 1–4 of primary school (edited by A. V. Goryachev). The examples of the tasks are given that students performed in the classroom.

Keywords: parallel computing, textbook set on informatics, programming, "Informatics for All", static objects, dynamic objects.

Истоки информатики традиционно принято искать в работах западных математиков, таких как Алонзо Черч, Алан Тьюринг и др. И конечно же, в выпущенной в 1948 году книге «Кибернетика» Норберта Винера. Первое время информатика рассматривалась как прикладная область математики и кибернетики. И только с развитием электронных вычислительных систем информатика получила признание как самостоятельная область научного знания и, как следствие, учебная дисциплина. В этой связи появление информатики в учебном процессе было направлено на развитие навыков автоматизации вычислений. Что, в свою очередь, положило начало такому разделу информатики, как программирование.

В СССР вопросы теоретической информатики на государственном уровне долгое время игнорировались, а разработки С. А. Лебедева и В. М. Глушкова в области создания вычислительных машин и автоматизации народного хозяйства, скорее, «терпели», нежели оказывали им существенную поддержку [11]. И только на фоне взрывообразного распространения средств автоматизации вычислений, сформировавшего социальный заказ на специалистов-программистов, учебная дисциплина «Информатика» появилась в школе.

Неудивительно, что информатика пришла в школу под лозунгом «Программирование — вторая грамотность». А содержательная линия программирования, в отличие от сформировавшихся позже линий формализации и моделирования, информационных технологий и иных, по сей день является одной из самых объемных в школьном курсе информатики. А. П. Ершов отмечал, что «грамотность и программирование не только выстраиваются в параллель, соединяясь мостиками аналогий, но и дополняют друг друга, формируя новое представление о гармонии человеческого ума» [3].

Но если относительно значимости линии программирования для курса информатики сомнений обычно не возникает, то возраст начала обучения программированию долгое время был предметом дискуссий. Тот же А. П. Ершов неоднократно ссылаясь по этому вопросу на работы Ж. Пиаже, отмечавшего, что любое знание связано с действием и, как следствие, любая деятельность может вызвать интерес ребенка и иметь для него смысл, если изучение объекта заменить его конструированием.

Соглашаясь с этой мыслью, А. П. Ершов замечал: «Хотелось бы подчеркнуть, что речь идет не о том, чтобы навязать детям новые, не свойственные им навыки и знания, а о том, чтобы проявить и сформулировать те стороны мышления и поведения, которые реально существуют, но формируются стихийно, неосознанно» [3].

Этими же идеями руководствовался и С. Пейперт, разработавший уникальный для своего времени язык LOGO. Этот язык был графичен и состоял из команд, понятных ученикам, команд, которые дети могли выполнить и сами, без компьютера. Кроме того, результатом работ также становилось что-то очень наглядное. Сам С. Пейперт воспринимал LOGO, скорее, как некую моделью языка, реализующего визуально-ориентированный подход, и признавал, что с течением времени появится множество более современных и более продуманных языковых концепций [10].

Время доказало правоту С. Пейперта. Сегодня школам предложен целый ряд различных учебных языков программирования, в основном пиктографического типа. Ю. А. Первин провел большую работу по сравнительному анализу подобных языков и отметил тенденцию к созданию языков для конструирования бестекстовых программ, что реализовано в средах «ПиктоМир», «ПервоЛого», Scratch, «Азбука Роботландии» [5].

Вопросам пропедевтики программирования в начальной школе уделяли внимание множество ведущих отечественных специалистов. Так, А. В. Горячев [2] и А. Л. Семенов [12] много работали над развитием у младших школьников основ алгоритмического мышления, Л. Л. Босова [1] уделила внимание изучению особенностей применения интерактивных сред, а А. В. Могилев [6] — вопросам геймификации при обучении программированию.

Между тем с момента введения ФГОС НОО подходы к обучению информатике на уровне начальной школы существенно пересматриваются [7]. И речь идет не только об обновлении целей и подходов к реализации курса [4], но и об обновлении содержания образования [9].

В вопросах раскрытия содержательной линии «Программирование» на уровне начального образования важно не только следовать положениям ФГОС НОО, но и учитывать социальный заказ общества системе образования. И в этой связи важно обратить внимание на некоторую односторонность реализации линии программирования, а именно на рассмотрение на уровне начального образования темы «Алгоритмы для конкретного исполнителя» в различных вариациях.

Программирование сегодня не укладывается только в рамки алгоритмизации «поведения» конкретного исполнителя. *Одна из главных тенденций современного программирования — параллельные вычисления.*

Под параллельными вычислениями мы будем подразумевать способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно (одновременно).

Тема эта сложна и, разумеется, на уровне начального образования рассматривается упрощенно. Но, учитывая, что период 7–10 лет у ребенка мы уверенно можем считать сенситивным по отношению к обучению основам программирования, вовсе не рассматривать такую важную тему, как «Параллельные вычисления», мы не можем.

Однако на уровне начальной школы проблематика должна рассматриваться скорее как «алгоритмы, управляемые событиями». Основные задачи реализации этого раздела заключаются в формировании у учеников представления об алгоритмах, управляемых событиями, и о взаимодействии нескольких одновременно функционирующих алгоритмов. Также важно раскрыть основные сложности взаимодействия нескольких исполнителей, а именно подходы к обеспечению правильной последовательности взаимодействий между различными вычислительными процессами.

Первое представление об алгоритмах, управляемых событиями, ученики начальной школы, в логике С. Пейперта, получают еще при работе с простыми бытовыми алгоритмами вроде алгоритма перехода через дорогу. Именно события и поведение сторонних исполнителей (светофора и участников дорожного движения) определяют поведение программируемого исполнителя, превращая линейный алгоритм его поведения в условный и даже (при некотором стечении обстоятельств) циклический.

Однако одних бытовых примеров и графического отображения алгоритмических конструкций в тетради для правильного представления об алгоритмах, управляемых событиями, недостаточно.

Опираясь на мнение ведущих специалистов в области пропедевтического обучения программированию,

таких как Ю. А. Первин, Л. Л. Босова, А. В. Могилев, Н. К. Нателаури и ряда других, мы в школе № 45 имени Л. И. Мильграма (г. Москва) при выборе подходов к обучению младших школьников информатике приняли решение иметь обязательную компьютерную часть. Обучение информатике младших школьников на базе нашей школы организовано в форме внеурочных занятий. Дидактическая основа определяется УМК «Информатика для всех» Д. И. Павлова под редакцией А. В. Горячева. В соответствии с имеющейся программой реализация проблематики «Алгоритмы, управляемые событиями» с помощью компьютера с использованием среды Kodu Game Lab начинается в третьем классе.

Разработанная корпорацией Microsoft свободно распространяемая среда **Kodu Game Lab** ориентирована на применение технологии геймификации при обучении программированию. Среда Kodu разворачивается на персональных компьютерах или консолях XBOX.

Kodu Game Lab изначально предлагает обучающимся взглянуть на программирование с нетрадиционной стороны. Каждый пользовательский алгоритм формируется по схеме «условие — действие», что превращает алгоритм в алгоритм, наступлением которого управляет то или иное событие. А это говорит о том, что выбранная среда подходит для реализации заявленной задачи — *пропедевтики параллельных вычислений с помощью компьютера на уровне начальной школы.*

Создание любого проекта в среде Kodu состоит из двух основных частей:

- первая часть — *создание среды* (ландшафта). В этой части среда особо интересна ученикам, так как похожа на любимую ими компьютерную игру Minecraft;
- вторая часть создания проекта — *работа с объектами*.

Объекты в среде Kodu также условно делятся на две группы — объекты и исполнители.

Объекты статичны. Исполнители могут вступать с ними во взаимодействие, сами же объекты могут выполнять какие-то команды, но не могут перемещаться по миру.

Исполнители мобильны. Они также делятся на две группы, которые условно можно назвать аватары и NPC.

Под *аватаром* мы понимаем исполнителя, движениями и иными действиями которого частично или полностью руководит пользователь.

NPC (Non Playable Creature) — это исполнитель, способный взаимодействовать с пользователем или средой, управление им осуществляет компьютер.

Работа с исполнителями на уроке проводится сначала в тетради: ученики формулируют правила действия для исполнителей и записывают их в тетради. После этого ребята реализуют алгоритмы средствами среды Kodu.

Ниже приведены **примеры заданий, в рамках которых обучающиеся осваивают практическую составляющую темы «Алгоритмы, управляемые событиями»**. В этих заданиях параллельность проявляется через взаимодействие двух и более исполнителей, реализующих каждый свой алгоритм и имеющих область пересекающихся задач.

Задание 1. Эмоции и взаимодействия.

Это одно из первых заданий, когда обучающиеся еще не создают полноценных игр, а только осваивают некоторые элементы среды.

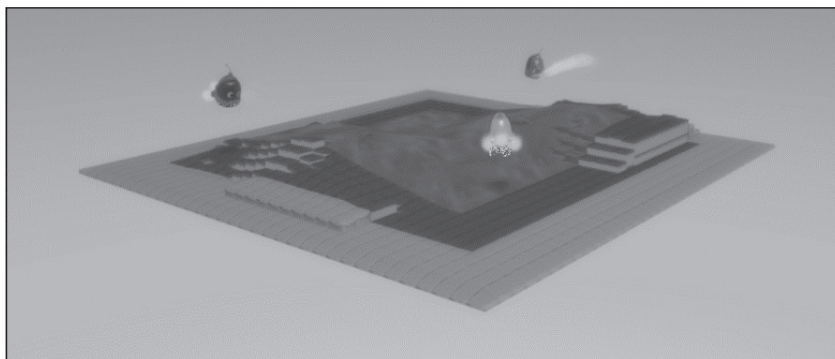


Рис. 1

Для начала ребята готовят простой «мир» и размещают в нем трех исполнителей (рис. 1).

Ученики получают задачу преобразовать двух исполнителей в NPC, один из которых перемещается по некоторому заданному пути, а второй — свободно. Третий исполнитель управляется с клавиатуры (рис. 2). К этому уроку школьники уже знают, как это делать, что превращает данный этап урока в этап актуализации.

Далее каждый ученик получает задание: настроить взаимодействие персонажей. При приближении к другому Kodu любой персонаж должен говорить ту или иную фразу и испытывать эмоцию, при этом фраза и эмоция зависят от того, к какому именно Kodu приближается

персонаж, т. е. от конкретного события. На примере NPC-1 такая программа выглядит, как представлено на рисунке 3.

К концу урока ученики получают небольшой игровой этюд, где поведение трех исполнителей меняется в зависимости от той или иной ситуации. Это позволяет учащимся наглядно проанализировать работу простейших алгоритмов, управляемых событиями, и заложить основу для реализации параллельных вычислений в дальнейшем.

Задание 2. Двое в поисках яблок.

Данный пример представляет собой преобразование изложенной в образовательной программе «Информатика в школе» задачи «Двое в поисках яблок».



Рис. 2



Рис. 3

тика для всех» [8] задачи «Байер и яблоки». Ученикам предлагается изменить игру, создав в ней соревновательный эффект — с компьютерным исполнителем или другим игроком.

В данном случае на карте периодически появляются яблоки красного и синего цветов. Первый исполнитель получает +1 балл при поедании красного яблока и –1 балл при поедании синего яблока. Второй исполнитель, наоборот, получает +1 балл при поедании синего яблока и –1 балл при поедании красного яблока. Кроме того, на карте присутствуют 2 NPC, которые, свободно перемещаясь, поедают любые яблоки в пределах видимости. Но им баллы не начисляются.

«Событие победы» в игре — набранные 20 баллов. Стратегий победы получается несколько, и стратегия зависит как от поведения второго исполнителя, так и от поведения NPC. Игрок может не только поедать «свои» яблоки, но и постараться не допустить к яблокам нужного цвета своего соперника.

При выполнении этого задания ученики получают пример практического применения алгоритмов, управляемых событиями, при создании компьютерных игр. Кроме того, часто занятие посвящается обсуждению возможных победных тактик, в том числе тех, когда рационально съесть «яблоко –1». Также здесь реализуется ряд исследований. Например, перед учениками ставится вопрос: «Если два исполнителя одновременно коснутся объекта “яблоко”, кто именно его съест?» В рамках этого исследования ученики выявляют основные закономерности «принятия решений» в «конфликтной ситуации» между двумя исполнителями.

* * *

Рассмотренные задания — только примеры. На начальном этапе обучения ученики осваивают их немало, и с каждым следующим уроком у них все больше преобладает самостоятельная проектная составляющая деятельности.

Результаты внедрения Kodu в учебный процесс показывают, что ученики уже в третьем классе показывают высокие результаты в рамках акции «Час кода», а в четвертом классе справляются с более сложными проектами в среде Scratch. Что касается далеко идущих оценок влияния использования среды Kodu на освоение

параллельного программирования в школе, то два года — недостаточный срок для подобных оценок. Но совместно с коллегами в школе № 45 имени Л. И. Мильграма мы разработали ряд модулей по освоению параллельного программирования в V—VI классах и в течение ближайших двух лет сможем проанализировать динамику освоения этой темы у учеников, прошедших пропедевтический курс, по сравнению с учениками, которые начали освоение темы «с нуля».

Список использованных источников

1. Босова Л. Л., Сорокина Т. Е. Методика применения интерактивных сред для обучения младших школьников программированию // Информатика и образование. 2014. № 7.
2. Горячев А. В. Методика обучения информатике в начальной школе, реализующая объектно-информационный подход: дис. ... канд. пед. наук. СПб., 2004.
3. Ершов А. П. Программирование — вторая грамотность // Архив А. П. Ершова. http://www.ershov.ras.ru/russian/second_literacy/article.html
4. Кузнецов А. А. Реализация требований нового ФГОС в практике школьного образования // Муниципальное образование: инновации и эксперимент. 2008. № 2.
5. Леонов А. Г., Первин Ю. А. Элементы программирования в непрерывном курсе школьной информатики // Ярославский педагогический вестник. 2013. Т. 3. № 1.
6. Могилов А. В. Что наша жизнь? Геймификация! // Народное образование. 2015. № 6.
7. Павлов Д. И. Новая редакция федерального государственного образовательного стандарта начального общего образования — место информатики в начальной школе // Педагогическая информатика. 2017. № 3.
8. Павлов Д. И. Информатика. 2–4 классы: методическое пособие / под ред. А. В. Горячева. М.: БИНОМ. Лаборатория знаний, 2016.
9. Павлов Д. И. Раскрытие содержательных линий «Представления информации» и «Информационных процессов» на уровне начального общего образования // Открытое и дистанционное образование. 2018. № 1.
10. Пейперт С. Переворот в сознании. Дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989.
11. Ревич Ю. В., Малиновский Б. Н. Информационные технологии в СССР. Создатели советской вычислительной техники. СПб.: БХВ-Петербург, 2014.
12. Семенов А. Л. Концептуальные проблемы информатики, алгоритмы и программирование в школе // Вестник кибернетики. 2016. № 2.

НОВОСТИ

Google сообщила об открытии раздела «Видеоигры» на YouTube

Google представила раздел «Видеоигры» на Youtube. В 2015 году был запущен «YouTube гейминг» — отдельное приложение для геймеров, в котором тестирование проходили новые функции. Среди них: страницы игр, суперчат, спонсорские подписки, ночной режим и даже новая платформа для трансляций на основе отзывов геймеров. Эти функции понравились игровому сообществу, поэтому они были перенесены из приложения на YouTube. Лучшие трансляции и загруженные видео собраны в одном месте.

К тому же добавлены разделы для трансляций и набирающих популярность видео. Теперь можно найти видео о конкретной игре на обновленных игровых страницах. Сюда входят популярные видео, трансляции и другие игры от того же издателя или разработчика.

В марте 2019 года прекратится поддержка приложения «YouTube гейминг», а Google полностью сосредоточится на игровых видео на YouTube, где их аудитория станет намного больше.

(По материалам CNews)