

РАЗРАБОТКА МЕТОДИЧЕСКИХ ПОДХОДОВ К РЕАЛИЗАЦИИ ПРОПЕДЕВТИЧЕСКОГО КУРСА ИНФОРМАТИКИ В НАЧАЛЬНОЙ ШКОЛЕ СРЕДСТВАМИ KODU GAME LAB

А. В. Каплан¹, Д. И. Павлов²

¹ Школа № 2009

117042, Россия, г. Москва, ул. Адмирала Руднева, д. 16, корп. 1

² Московский педагогический государственный университет

119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1

Аннотация

В статье представлены результаты исследовательской работы по разработке методических подходов к пропедевтике программирования в начальной школе с использованием среды Kodu Game Lab. Рассмотрены разнообразные существующие тенденции в начальном обучении программированию, а также связь обучения программированию с развитием вычислительного мышления, формированием «новой грамотности» и достижением результатов начального образования. Выделены семь методических аспектов использования Kodu Game Lab (быстрое вхождение учеников в тему; освоение способа действия вместо запоминания последовательности действий; освоение способа действия вместо листинга; дифференциация результатов при реализации формализованных заданий; реализация проектной деятельности в рамках урока; проведение исследований закономерностей виртуальной среды; развитие навыков групповой работы над цифровым проектом) и приведены примеры их реализации. Учено отсутствие достаточной статистической поддержки анализа эффективности внедрения разработанных подходов, соответственно, сформулированы дальнейшие пути исследования. Также приводятся аргументы в пользу использования изложенных методических подходов.

Ключевые слова: информатика, начальная школа, программирование, пропедевтика программирования, Kodu Game Lab.

DOI: 10.32517/0234-0453-2019-34-8-14-23

Для цитирования:

Каплан А. В., Павлов Д. И. Разработка методических подходов к реализации пропедевтического курса информатики в начальной школе средствами Kodu Game Lab // Информатика и образование. 2019. № 8. С. 14–23.

Статья поступила в редакцию: 1 августа 2019 года.

Статья принята к печати: 17 сентября 2019 года.

Сведения об авторах

Каплан Адели Викторовна, учитель начальных классов, школа № 2009, г. Москва, Россия; adel.caplan@ya.ru; ORCID: 0000-0002-8581-5086

Павлов Дмитрий Игоревич, ст. преподаватель кафедры теории и методики обучения математике и информатике, Институт математики и информатики, Московский педагогический государственный университет, Россия; di.pavlov@mpgu.su; ORCID: 0000-0002-0074-0899

1. Актуальность проблемы пропедевтики программирования с использованием вычислительных сред

Вопросам преподавания информатики на уровне начального образования уделяется сегодня большое внимание. В частности, на проходившей 10–12 октября 2018 года в Санкт-Петербурге XI Международной конференции по преподаванию информатики в школе ISSEP 2018 значительная часть выступлений была посвящена именно пропедевтическому этапу обучения информатике. Специалисты в числе прочего обсуждали современные интерпретации работ Сеймура Пайперта и использование языка Logo при пропедевтике обучения программированию младших школьников. Отмечалось, что навыки чтения и написания программ, усиленные применением математики, являются решающим аспектом развития логического стиля мышления [1].

Высказываются и иные точки зрения. Например, некоторые специалисты отмечают, что традицион-

ные подходы, в частности использование языка Logo, не являются оптимальными на пропедевтическом этапе. Для реализации пропедевтики обучения программированию младших школьников предлагается определить элементарные когнитивные операции в программировании начального уровня и на основе полученных результатов разрабатывать задания различного уровня сложности, которые будут поддерживать плавный процесс обучения навыкам программирования учащихся начальных классов, а также формирование вычислительного мышления, при тщательном соблюдении уместности развития соответствующих видов деятельности [2].

«Вычислительное мышление» — один из самых обсуждаемых сегодня среди специалистов терминов в области преподавания информатики [3]. Изучению этого явления уделяют внимание как зарубежные, так и отечественные специалисты. Так, Е. К. Хеннер в своих работах отмечает, что термин «Computational Thinking» буквально переводится на русский язык как «вычислительное мышление», но перевод этот —

вынужденный вариант, поскольку «английское слово Computational, кроме использования в чисто математическом смысле (производство вычислений), в настоящее время параллельно используется в более широком смысле, родственном термину “Computing” (компьютинг) — собирательному обозначению совокупности компьютерных наук (Computer Science), информационных технологий и информационных систем, компьютерной и программной инженерии» [4].

И. Калаш (I. Kalaš) из Университета Комениуса (Братислава, Словакия) отмечает, что вычислительное мышление «формирует такие качества, как терпимость, уверенность в себе, работоспособность; вырабатывает способность решать неизвестные ранее задачи, справляться с неизвестными проблемами; влияет на способы существования и мышления» [5]. Развитие вычислительного мышления на уроках информатики сегодня все чаще становится определяющей задачей пропедевтического курса информатики как в России, так и за рубежом [6].

Но термин «вычислительное мышление» встречается сегодня не только в научно-методической литературе, связанной с информатикой. Если Ж. М. Винг (J. M. Wing) и ее последователи сегодня говорят о computational thinking как о самодостаточном явлении [7], то в работах И. Д. Фрумина, К. А. Баранникова, М. С. Добряковой, И. М. Реморенко и ряда других отечественных и зарубежных специалистов сделаны предположения о связи элементов вычислительного мышления со всесторонним и гармоничным развитием современной личности в целом. Этот тезис подтверждается и в выводах международного проекта «Ключевые компетентности и новая грамотность», в рамках которой вычислительная грамотность является одним из компонентов «новой грамотности» [8].

В этой связи нами было сделано предположение о необходимости внедрения основ программирования в программу обучения младших школьников. Разделяя мнение о том, что язык Logo не является оптимальным средством пропедевтики программирования, мы не готовы были рассматривать в качестве такого средства и среду Emil, так как она платная [3]. Также, проанализировав результаты исследовательского проекта ScratchMaths, мы решили отказаться от начала обучения с использованием среды Scratch [9].

Среди перспективных и востребованных подходов в преподавании информатики в настоящее время выделяется **технология геймификации**, которая, по мнению специалистов, не только «призвана создать такую информационно-обучающую среду, которая бы способствовала самостоятельному, активному стремлению обучающихся к получению знаний, навыков и умений, таких как критическое мышление, умение принимать решения, работать в команде» [10], но и доказала свою применимость при обучении программированию [11]. Для реализации своего замысла мы выбрали среду Kodu Game Lab, однако, столкнулись с отсутствием разработанных методических подходов к преподаванию основ программирования с использованием этой среды. В этой связи

было принято решение разработать методические подходы к реализации пропедевтического курса программирования в начальной школе с использованием среды Kodu Game Lab.

2. Методы

В ходе работы были использованы следующие методы теоретического исследования: анализ научно-методической литературы, абстрагирование, метод неполной индукции. В опытно-практической части были проведены эксперименты, а также наблюдения за обучающимися, измерение и сравнение результатов обучения.

При изучении учебно-методической литературы была проанализирована трактовка **понятия «вычислительное мышление»** разными педагогами-исследователями. В интерпретации одного из авторов этого термина Сеймура Пейперта, «вычислительное мышление является способом решения проблем людьми, а не попыткой уподобить человеческое мышление компьютерам. Компьютеры — скучные и нудные, а люди умны и обладают воображением. Мы, люди, делаем компьютеры эффективными. Оснащенные вычислительными устройствами, мы используем наш ум, чтобы решать проблемы, которые мы не могли решать до компьютерной эры и создавать системы, обладающие функциональностью, ограниченной только нашим воображением» [12]. Несколько позже одна из последовательниц Пейперта — Жаннетта Винг сформулировала следующий вариант определения: «Вычислительное мышление — это мыслительные процессы, участвующие в постановке проблем и их решения таким образом, чтобы решения были представлены в форме, которая может быть эффективно реализована с помощью средств обработки информации» [13].

Работая над анализом термина «вычислительное мышление», Е. К. Хеннер сделал следующий вывод: «Формирование вычислительного мышления можно рассматривать в связке с формированием информационно-коммуникационной компетентности и информационной культуры. В процессном плане они могут быть неразрывны, но как результат образования вычислительное мышление сохраняет в этой триаде относительную самостоятельность: человек, обладающий вычислительным мышлением, должен быть ориентирован на решение задач с помощью средств инфокоммуникационных технологий, привычно мыслить соответствующими категориями. Возможно, это и есть главная черта вычислительного мышления; его наличие должно стать важным личностным и метапредметным результатом школьного (и не только школьного) образования, а также его инструментом» [4].

Также были проанализированы результаты международного проекта «Ключевые компетентности и новая грамотность» [8], проведенного представителями Института образования Национального исследовательского университета «Высшая школа экономики», Московского городского педагогиче-

ского университета (Россия), Institute of Education, University College of London (Великобритания), Ontario Institute for Studies in Education, University of Toronto (Канада), Graduate School of Education, Peking University (Китай), College of Education, Seoul National University (Южная Корея), Evidence Institute Warsaw University (Польша), Lynch School of Education, Boston College (США), Faculty for Educational Sciences, University of Helsinki (Финляндия) и направленного на концептуальное прояснение идущей трансформации школьного образования.

Одним из результатов работы проекта стало уточнение понятия «грамотность», или, точнее, **формулирование понятия «новая грамотность»**, которое представляет собой развитие традиционного понятия «грамотность» и состоит из пяти компонентов, а именно трех групп компетентностей и двух видов грамотности.

К компетентностям отнесены:

- «компетентность мышления;
- компетентность взаимодействия с людьми;
- компетентность взаимодействия с собой» [8].

К компонентам грамотности отнесены:

- базовая инструментальная грамотность;
- базовые специальные современные знания и умения.

Базовая инструментальная грамотность — неотъемлемый компонент «новой грамотности», она «основана на использовании современных инструментов коммуникации, опирающихся на знаковые системы, подразумевает трансформацию в современных технологических условиях привычной грамотности “читать + писать + считать” с поправкой на форматы взаимодействия и способы передачи информации, в том числе в режиме “человек — человек” и “человек — машина”» [8].

Базовая инструментальная грамотность — составное понятие, она включает:

- читательский компонент;
- математический компонент;
- вычислительный (алгоритмический) компонент.

Анализ работ участников проекта «Ключевые компетентности и новая грамотность» показал, что вычислительный компонент базовой инструментальной грамотности как раз и представляет собой развитие вычислительного мышления, что подтверждает актуальность работы авторов статьи.

Что касается **среды Kodu Game Lab**, то в российской научно-методической литературе представлено мало материалов по использованию данной среды. Отдельными авторами рассматривается использование Kodu Game Lab как инструмента развития алгоритмического стиля мышления и отмечается, что «алгоритмический подход применим не только в компьютерных дисциплинах, но и в общеобразовательных предметах» [14]. Также рассматривается перспективная связка Kodu — Scratch как двух последовательных этапов освоения основ программирования в начальной школе. Этот подход уже реализован в УМК «Информатика для всех», выпущенного

издательством «БИНОМ. Лаборатория знаний» под редакцией А. В. Горячева [15].

Отдельные специалисты предлагают использовать Kodu Game Lab в качестве пропедевтики робототехнической подготовки [16]. Они отмечают, что данная среда как нельзя лучше подходит для использования технологии геймификации при обучении программированию. Мы также уже отмечали в своих работах, что использование среды Kodu Game Lab «позволяет всерьез обновить содержание начальной информатики и адаптировать его к тем планируемым результатам, которые заданы ФГОС НОО» [17].

Однако, несмотря на определенное внимание к использованию среды Kodu Game Lab, нет соответствующих методических разработок, что также подтвердило актуальность проводимой работы.

Опытно-экспериментальная часть исследования проводилась в школах Москвы и Московской области в третьих классах. Содержательную основу опытно-экспериментальной работы составил курс обучения основам программирования для третьего класса, реализованный в УМК «Информатика для всех». Авторы проводили наблюдение за освоением материала, анализировали вовлеченность обучающихся, их результативность, общие тенденции и различия в результатах, показанных учениками. Использовались различные педагогические приемы и технологии. Результаты обобщались и систематизировались.

3. Результаты

В результате обучения были выделены следующие **особенности и приемы проведения занятий по обучению программированию младших школьников с использованием среды Kodu Game Lab**:

- 1) быстрое вхождение учеников в тему;
- 2) освоение способа действия вместо запоминания последовательности действий;
- 3) освоение способа действия вместо листинга;
- 4) дифференциация результатов при реализации формализованных заданий;
- 5) реализация проектной деятельности в рамках урока;
- 6) проведение исследований закономерностей виртуальной среды;
- 7) развитие навыков групповой работы над цифровым проектом.

Рассмотрим подробно каждый полученный результат.

3.1. Быстрое вхождение учеников в тему

На рисунке 1 приведены снимки экрана, сделанные в двух разных программах. Левый снимок — среда Kodu Game Lab, правый — игра Minecraft. Проведенный эксперимент показал, что все ученики с I по XI класс видят существенные сходства между этими двумя программными продуктами.

Более того, впервые увидев на экране Kodu Game Lab, ученики начальной школы восклицают: «О, Minecraft!», а кроме того, обсуждают, что на уроке их ждет игра.

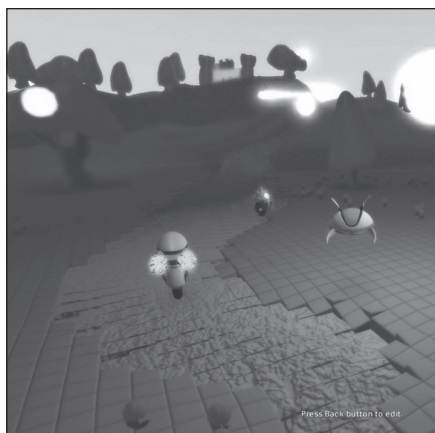


Рис. 1. Сравнение интерфейса двух программ — среды Kodu Game Lab (слева) и игры Minecraft (справа)

Эксплуатация этого не слишком очевидного для взрослого человека сходства — сильный инструмент мотивации в руках учителя. Учеников не стоит убеждать в том, что их ждет игра! Первичный уровень заинтересованности столь высок, а инструментарий среды Kodu Game Lab так разнообразен, что к тому моменту, когда ученики начинают понимать, что Kodu Game Lab и Minecraft — это разные вещи, их заинтересованность в игре уже подменена заинтересованностью в создании игры.

3.2. Освоение способа действия вместо запоминания последовательности действий

Одна из главных проблем при обучении использованию той или иной среды школьниками — алгоритмический подход к обучению. Опирающийся на научно-методические разработки 70–80-х годов XX века, этот подход долгое время считался многообещающим. В частности, отмечалось, что «в ходе работы с применением алгоритмического метода в решении задач у учащихся устанавливаются связи между понятиями, усвоенными в ходе изучения темы, что приводит к осмысленному применению полученных знаний, появляется уверенность в своих силах и способностях. Основными характеристиками особенностями алгоритмического метода являются детерминированность, массовость и результативность» [18]. Однако практика обучения информатике показывает, что при освоении особенностей программной среды алгоритмизация действий при выполнении операционных задач не позволяет ученикам запомнить необходимый порядок действий, что связано с особенностями мышления современных школьников.

Решение этой проблемы было предложено в рамках системы развивающего обучения Д. Б. Эльконина — В. В. Давыдова, где предлагается «строить, организуя переходы от практических задач к поиску общих способов действия, а не методом запоминания частных» [19]. Кроме того, учителя начальной школы часто отмечают, что современные ученики гораздо менее внимательны к словам, произносимым одноклассниками, что часто приводит к тому, что в течение нескольких минут урока несколько учени-

ков могут задать один и тот же вопрос, на который учитель уже ответил.

Kodu Game Lab содержит прекрасный инструментарий для того, чтобы избегать этих сложных моментов и научить детей не просить помощи, под которой зачастую скрывается желание переложить решение задачи на плечи учителя, а освоить способ поиска ответа на вопрос: «Как выполнить конкретное действие?» в среде Kodu Game Lab.

Для качественного освоения особенностей среды ученикам необходимо освоить два способа действия.

Первый из них — проверять активный инструмент. Это поможет избежать многочисленных вопросов типа: «Почему я нажимаю на объект и ничего не происходит?» С первых же занятий ученикам надо сообщить, что, если на карте в режиме редактирования происходит что-то не так, как им бы хотелось, они должны проверить, какой инструмент они выбрали. То есть ученик не использует готовый алгоритм, а создает его, отвечая на вопросы, например, так:

Проблема: при нажатии на объект он не выделяется.

Область поиска: панель инструментов внизу экрана.

Результат поиска: желтая рамка стоит на инструменте «Рука».

Способ исправления: выбрать инструмент «Объект».

Сообщив обучающимся этот способ действия уже на первом занятии, учитель должен организовать его рефлекссию в конце занятия и делать это в начале следующих двух—четырех уроков на этапе актуализации. Если же во время урока учителю поступит вопрос, суть которого — неверно выбранный инструмент, педагогу необходимо привлечь внимание детей и вместе с ними, не сообщая пути исправления, вспомнить, где надо искать проблему.

Второй способ действия призван свести к минимуму вопросы: «Как сделать...?», касающиеся инструментария среды Kodu Game Lab. На рисунке 2 указана область подсказок (аналог контекстного меню), содержание которой меняется в зависимости от выбранного инструмента. По сути, это шпаргалка,

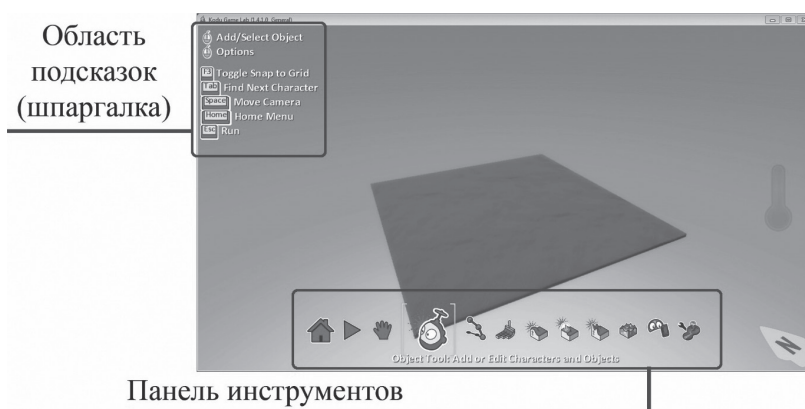


Рис. 2. Основные элементы интерфейса Kodu Game Lab

и научить детей пользоваться ею — одна из перво-степенных задач, если учитель не хочет весь урок отвечать на однотипные вопросы. Ученики не будут запоминать, что и как делать, если учитель постоянно будет думать за них, и вопросы: «Как сменить размер кисти?», «Как поменять цвет объекта?» и т. п. станут основными на каждом уроке, отвлекая учеников от нового материала.

Поэтому учителю необходимо на первом же занятии познакомить учеников со «шпаргалкой», разобрать изменение ее содержания в зависимости от выбранного инструмента и сообщить следующий способ действия: «Если вы забыли, как использовать тот или иной инструмент, ищите ответ в левом верхнем углу экрана». Этот способ действия, так же как и предыдущий, нуждается в рефлексии на соответствующем этапе первого урока и на этапах актуализации второго—четвертого уроков, а также ситуативно, при появлении вопроса: «А как сделать ...?»

Да, с одной стороны, не так сложно сообщить готовый ответ. Но тогда ученики будут копить «неосвоенные» технические навыки, что будет мешать их творчеству, а также созданию программ, и урок превратится в серию однотипных ответов и вопросов. В то же время освоение детьми двух представленных способов действия позволит им всегда находиться «в тонусе» и будет способствовать достижению метапредметных результатов начального образования в части формирования регулятивных УУД.

3.3. Освоение способа действия вместо листинга

Еще одной проблемой при обучении программированию в школе является листинг, т. е. представление готовой программы вместо освоения навыков конструирования алгоритмов.

При использовании среды Kodu Game Lab на уроках в начальной школе эту проблему можно решить с помощью межпредметных связей, а именно с помощью синтаксической работы [20].

Например, если первая программа, которую мы осваиваем с детьми, должна стать программой движения объекта, управляемого клавишами, то, прежде чем разобрать, как создать эту программу,

предложим ученикам порассуждать. Рассуждение, для удобства его интеграции в программу Kodu Game Lab, должно начинаться со слова «когда»:

- Когда что-то произойдет?
- Когда будут нажаты *клавиши*!
- Когда *какие* клавиши будут нажаты?
- Когда будут нажаты клавиши-*стрелки*!
- Что произойдет, *когда будут нажаты клавиши-стрелки*?
- *Объект* будет *двигаться*!

Когда вопросы заданы и выстроены в порядке «от условия к действию», с учениками можно разобрать технологию сбора программы по сформулированному алгоритму. Получится программа, пример которой показан на рисунке 3.

После этого ученикам можно дать ряд заданий на самостоятельное освоение, например, чтобы при нажатии клавиши пробела исполнитель совершал прыжок. Готовой программы учитель в этот момент уже показывать не должен, но необходимо вместе с учениками составить текстовое описание будущей программы, задавая вопросы, как было показано в примере выше. И в дальнейшем надо приучать детей задавать именно вопросы, а не ждать готовой программы. Приведем еще один пример последовательности вопросов и ответов:

- Когда что-то произойдет?
- Когда увижу *дерево*!
- Какое *дерево*?
- *Зеленое*!
- Что произойдет, *когда увижу зеленое дерево*?
- *Объект* испытает *эмоцию*!
- Какую *эмоцию*?
- *Радость*!

Ученики смогут составить программу, как на рисунке 4, самостоятельно.

В дальнейшем соотношение примеров, где разбирается составление конкретной программы (листинга), к примерам, составляемым учениками посредством цепочки вопросов, должно составлять 1:5 или даже 1:7. Листинг-примеры допустимы в сложных и новых случаях, таких как первичный разбор ситуаций, когда одно условие приводит к двум и более действиям за раз, или при объяснении работы с переменными.

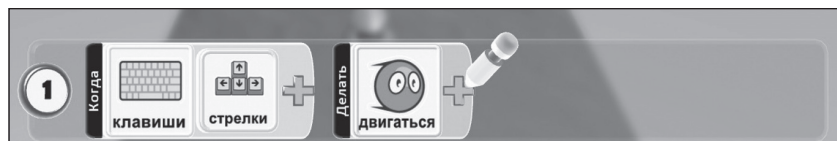


Рис. 3. Пример программы для движения



Рис. 4. Пример программы для отображения эмоции

В результате освоения предложенного способа действия возрастает темп урока, а ученики обретают способность к самостоятельному проектированию программы для конкретного исполнителя, что будет им полезно в основной школе при освоении соответствующего раздела курса информатики.

3.4. Дифференциация результатов при реализации формализованных заданий

Несмотря на все требования, связанные с индивидуализацией обучения, потребностью в индивидуализации результатов и прочими современными психолого-педагогическими тенденциями, мы нередко вынуждены работать с детьми в чрезвычайно формализованных рамках. Так, на уровне основной школы программа, которая вычисляет корни квадратного уравнения или выполняет пузырьковую сортировку массива, будет одинаковой у всех учеников. Разве что отличаться будут имена переменных и особенности форматирования.

На рисунке 5 показаны три работы детей, старших выполнить следующее задание: «Создаваемый мир должен иметь два типа поверхности, два холма, одно дерево и один водоем». И несмотря на то, что задание очень сильно формализовано (такие задания часто встречаются на начальном этапе обучения), работы выглядят *очень* по-разному, они совершенно не похожи одна на другую, хотя каждая из них соответствует установленным критериям.

Подобный эффект повышает вовлеченность учеников в процесс обучения, при этом такая дифференциация не требует от учителя серьезных затрат, позволяя сконцентрироваться на содержании обучения и создании необычных заданий.

3.5. Реализация проектной деятельности в рамках урока

Вопросам проектной деятельности, в частности на уровне начального образования, в настоящее время уделяется большое внимание, так как «проектно-исследовательская деятельность позволяет раскрыть индивидуальные способности детей младшего школьного возраста и дает им возможность приложить свои знания, принести пользу и публично показать достигнутый результат» [21].

Школьные учителя по-разному относятся к методу проектов. Чрезмерная формализация системы образования приводит к тому, что нередко домашние задания или отдельные виды традиционных работ на уроке объявляются проектной работой — только с целью соответствия формальным критериям отчетности. А между тем педагоги нередко отмечают, что реализация проекта в течение 45 минут урока — задача, если не невозможная, то крайне трудная, требующая длительной подготовки.

С этим трудно не согласиться, особенно если помнить, что проект — «это комплекс поисковых, исследовательских, расчетных, графических и дру-



Рис. 5. Примеры детских работ

гих видов работ, выполняемых учащимися самостоятельно (в парах, группах или индивидуально) с целью практического или теоретического решения значимой проблемы» [22].

Однако уже после седьмого-восьмого занятия в среде Kodu Game Lab, т. е. после освоения основных способов действия и особенностей среды, каждое второе занятие начинает соответствовать формальным критериям проектной деятельности (например, изложенным в [23]). Ребенок все больше самостоятельно формулирует задачу, она обладает для ученика пусть субъективной, но новизной и значимостью, для ее реализации учащийся самостоятельно планирует ход работ, осваивает новые навыки и получает на выходе цифровой продукт.

Таким образом, использование среды Kodu Game Lab позволяет не формально, а полноценно реализовать метод проектов, гармонично встроив его в программу обучения и не затрачивая непропорционально серьезных ресурсов на подготовку, а лишь реализуя образовательную программу начального курса информатики.

3.6. Проведение исследований закономерностей виртуальной среды

В отличие от Logo, Emil или Scratch Kodu Game Lab — полноценная виртуальная среда, живущая по определенным законам. Законы эти зачастую сродни привычным нам законам физики, но и они имеют свои особенности, которые нигде заранее не описаны.

Чаще всего мы не замечаем этого, сообщая детям все, что «им нужно знать», заранее и упуская существенный педагогический компонент. А именно то, что среда Kodu Game Lab является прекрасным симулятором исследований. На примере этой среды ученики могут осваивать навыки системного анализа и исследовательской деятельности, реально открывая для себя что-то новое.

Простой пример. Объекты в среде Kodu Game Lab могут передвигаться с разной скоростью. К команде «двигаться» можно добавить модификаторы «быстро» или «медленно». Но нет никаких описаний, насколько быстрее поедет наш исполнитель, если сказать ему «двигаться + быстро». Можно упустить этот момент, а можно предложить детям создать длинный прямой трек, подготовить таймер и засечь, насколько обычное движение отличается от быстрого или медленного, заполнив соответствующую таблицу.

Такое исследование пока сродни простому наблюдению, но оно рождает целый ряд дополнительных вопросов:

- Можно ли сделать два и более одинаковых модификатора? Влияет ли это на скорость?
- Одинаково ли работают модификаторы для разных исполнителей?
- «Двигаться + быстро + медленно» — это то же самое, что просто «двигаться» или нет?
- Влияют ли разные типы поверхности на ускорение/замедление?

- Влияют ли уклоны на ускорение/замедление?

Все это вопросы для самостоятельных исследований, на примере которых ученики могут отработать навыки планирования и проведения исследований, создания моделей и представления результатов.

Кроме упомянутого примера в нашей работе интересы вызвали вопросы:

- Что быстрее падает с высоты — яблоко или мяч?
- Какие из объектов, упавших в воду, будут плавать, а какие утонут?
- Если два персонажа, управляемых компьютером, должны при касании съесть друг друга, то какой из них съест другого первым?

Также ребятам были интересны и многие другие вопросы, которые позволяют отвлечь учеников от игры и дать им почувствовать себя исследователями, не отвлекаясь при этом от курса информатики.

3.7. Развитие навыков групповой работы над цифровым проектом

Освоить все премудрости программирования и просто найти себя в нем может далеко не каждый ученик. Однако современный цифровой продукт не является результатом индивидуального стартапа. Это сложный плод труда целых коллективов, в которых далеко не все являются программистами. Следовательно, ученики, которым программирование дается трудно, все равно могут стать частью творческого коллектива, создающего компьютерную игру. И здесь на выручку нам вновь приходит проектная деятельность. Если ученики уже освоили функционал Kodu Game Lab и в состоянии выполнять индивидуальные открытые задания, некоторые задания следует сделать групповыми, с разделением задач.

Так, интерес представляют следующие роли:

- *автор* — их в коллективе может быть несколько; все участники группы могут предлагать какие-то идеи и участвовать в разработке замысла;
- *дизайнер уровней* — человек, который оформляет уровни, делая их красивыми и удобными;
- *программист* — человек, который «оживляет» мир, отвечая за работу исполнителей в созданном мире;
- *тестировщик* — человек, который опробывает игру в самых разных режимах, вынося свое суждение о качестве игры, предлагая улучшения и указывая на ошибки; такого специалиста можно приглашать из другого коллектива;
- *специалист по продвижению* — человек, который сможет красиво рассказать о проделанной работе, ярко и красноречиво представить проект.

Подобный подход не должен становиться основным при реализации курса, иначе временные трудности могут оттолкнуть от программирования одаренных детей, однако он может позволить найти новые интересы как для учеников, проявивших склонность к программированию, так и для тех, кто не нашел себя в нем.

4. Выводы

Полученные в ходе теоретического поиска и практических исследований методические подходы позволили авторам подготовить методические рекомендации и разработки, которые должны помочь учителям начальной школы и учителям информатики, работающим с младшими школьниками, эффективно и органично реализовывать курс пропедевтики программирования с использованием Kodu Game Lab.

Об эффективности предложенных подходов можно судить по следующим показателям:

- при среднем уровне оттока обучающихся из групп внеурочной деятельности в течение года 25–50 % отток из групп по основам программирования составлял в среднем не более одного-двух учеников на группу в год;
- учитывая предельную пропускную способность одного класса, оборудованного компьютерными рабочими местами, отмечена не только высокая наполняемость групп, но и рост количества групп пропедевтики программирования, открываемых в школе, — в среднем количество таких групп в течение трех лет увеличивается вдвое;
- в течение трех лет объем знаний и навыков, освоенных детьми в рамках курса объемом 34 часа, стал на две трети больше изначального, наполовину выросло количество творческих и проектных заданий, что свидетельствует о более высокой плотности и эффективности занятий при использовании разработанных методических подходов.

Однако в настоящий момент, несмотря на теоретические предпосылки, авторы не берутся судить о влиянии разработанных подходов на формирование вычислительного мышления и на результаты начального общего образования. Для этого собрано недостаточно данных, и этот аспект исследования нуждается в дальнейшей разработке.

Кроме того, авторы исследования к настоящему моменту не проводили обработку результатов обучения с помощью методов математической статистики. В ближайшие два года запланировано существенное расширение исследовательской базы, т. е. количества школ, участвующих в эксперименте. Результаты, полученные из 10–12 различных школ трех-четырех регионов, будут, по мнению авторов, более релевантными.

Таким образом, предложенные методические подходы, безусловно, нуждаются в дальнейшей апробации, обсуждении и развитии, но при этом уже сегодня могут быть полезны учителям начальной школы и учителям информатики, работающим с младшими школьниками в реализации пропедевтического курса программирования с использованием Kodu Game Lab.

Список использованных источников

1. *Loyo A. H.* Effects on the school performance of teaching programming in elementary and secondary schools //

Informatics in Schools. Fundamentals of Computer Science and Software Engineering. Proc. 11th Int. Conf. on Informatics in Schools: Situation, Evolution, and Perspectives. Springer International Publishing, 2018. P. 30–41. DOI: 10.1007/978-3-030-02750-6_3

2. *Gujberova M., Kalas I.* Designing productive gradations of tasks in primary programming education // Proc. 8th Workshop in Primary and Secondary Computing Education (Aarhus, 11–13 November 2013). New York: ACM, 2013. P. 108–117. DOI: 10.1145/2532748.2532750

3. *Босова Л. Л., Каплан А. В.* Международная конференция по школьной информатике ISSEP 2018 // Информатика в школе. 2018. № 9. С. 2–6.

4. *Хеннер Е. К.* Вычислительное мышление // Образование и наука. 2016. № 2. С. 18–33. DOI: 10.17853/1994-5639-2016-2-18-33

5. *Калаш И.* Вычислительное мышление и дошкольное образование // Воспитание и обучение детей младшего возраста. 2013. № 1. С. 50.

6. *Босова Л. Л.* Школьная информатика в России и в мире // Информатизация образования и науки. 2018. № 3. С. 134–145.

7. *Wing J. M.* Computational thinking // Communications of the ACM – Self managed systems. 2006. Vol. 49. Is. 3. P. 33–35. DOI: 10.1145/1118178.1118215

8. *Фрумин И. Д., Добрякова М. С., Баранников К. А., Реморенко И. М.* Универсальные компетентности и новая грамотность: чему учить сегодня для успеха завтра. М.: НИУ ВШЭ, 2018. 28 с. https://ioe.hse.ru/data/2018/07/12/1151646087/2_19.pdf

9. *Kalaš I., Benton L.* Defining procedures in early computing education // Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing. IFIP World Conference on Computers in Education. Cham: Springer, 2017. P. 567–578. DOI: 10.1007/978-3-319-74310-3_57

10. *Варенина Л. П.* Геймификация в образовании // Историческая и социально-образовательная мысль. 2014. Т. 6. № 6-2. С. 314–317.

11. *Павлов Д. И., Бутарев К. В., Балашова Е. В.* О перспективах использования технологий геймификации при раннем обучении объектно-ориентированному программированию // Современные информационные технологии и ИТ-образование. 2018. Т. 14. № 4. С. 977–985. <http://sitito.cs.msu.ru/index.php/SITITO/article/view/465>

12. *Papert S.* Mindstorms: children, computers, and powerful ideas. New York: Basic Books, 1980. 242 p.

13. National Research Council Report of a Workshop on the Scope and Nature of Computational Thinking. Washington, DC: The National Academies Press, 2010. 114 p. DOI: 10.17226/12840

14. *Чебурина О. В.* Формирование алгоритмического мышления в обучении программированию игр // Наука и перспективы. 2017. № 2. С. 75–79. <http://nip.esrae.ru/ru/14-r116>

15. *Павлов Д. И.* Информатика для всех. 2–4 классы: учебно-методический комплект / под ред. А. В. Горячева. М.: БИНОМ. Лаборатория знаний, 2018.

16. *Беленов Н. В., Самсонова О. С.* Робототехника во внеурочной деятельности как фактор развития технических способностей у обучающихся // International Scientific Review. 2015. № 4. С. 11–15.

17. *Каплан А. В.* Применение технологии геймификации в пропедевтике программирования в начальной школе // Информатика в школе. 2018. № 6. С. 65–67.

18. *Ланда Л. Н.* Алгоритмизация в обучении. М.: Просвещение, 1966. 523 с.

19. *Воронцов А. Б.* Практика развивающего обучения по системе Д. Б. Эльконина — В. В. Давыдова. М.: ЦППО «Развитие личности», 1998. 390 с.

20. Эльконин Б. Д., Воронцов А. Б., Чудинова Е. В. Подростковый этап школьного образования в системе Эльконина — Давыдова // Вопросы образования. 2004. № 3. С. 118–142. <https://www.hse.ru/data/2010/12/31/1208182300/07elkonin118-142.pdf>

21. Бледнова Е. В. Проектная деятельность как условие развития творческих способностей учащихся начальной школы // Муниципальное образование: инновации и эксперимент. 2012. № 6. С. 28–31.

22. Поздеева С. И., Кузнецова Т. В. Проектная деятельность в практике учителя начальной школы // Вестник Томского государственного педагогического университета. 2006. № 10. С. 65–68. https://vestnik.tspu.edu.ru/files/vestnik/PDF/articles/pozdeeva_s_i_65_68_10_61_2006.pdf

23. Полат Е. С. Метод проектов: типология и структура // Лицейское и гимназическое образование. 2002. № 9. С. 9–17.

THE DEVELOPMENT OF METHODOLOGICAL APPROACHES TO THE IMPLEMENTATION OF THE PROPEDEUTIC COURSE OF INFORMATICS IN PRIMARY SCHOOL BY MEANS OF KODU GAME LAB

A. V. Kaplan¹, D. I. Pavlov²

¹ School 2009

117042, Russia, Moscow, ul. Admirala Rudneva, 1, building 1

² Moscow Pedagogical State University

119991, Russia, Moscow, ul. Malaya Pirogovskaya, 1, building 1

Abstract

The article describes the results of the study of methodological approaches to teaching programming in primary school using the Kodu Game Lab environment. Existing trends in teaching programming are described, as well as the connection of training with the development of computational thinking and the formation of “new literacy”. Seven methodological aspects of using the Kodu Game Lab are highlighted (quick entry of students into the topic; mastering the method of action instead of remembering the sequence of actions; mastering the method of action instead of listing; differentiation of results in the implementation of formalized tasks; implementation of project activity as part of the lesson; conducting research on the patterns of the virtual environment; development of teamwork skills on a digital project) and examples of the implementation of each of them are provided. The lack of sufficient statistical support for the analysis of the effectiveness of the implementation of the developed approaches has been taken into account, accordingly, further research paths have been formulated. There are also arguments in favor of using the stated methodological approaches.

Keywords: informatics, primary school, programming, propedeutics of programming, Kodu Game Lab.

DOI: 10.32517/0234-0453-2019-34-8-14-23

For citation:

Kaplan A. V., Pavlov D. I. Razrabotka metodicheskikh podkhodov k realizatsii propedevticheskogo kursa informatiki v nachal'noj shkole sredstvami Kodu Game Lab [The development of methodical approaches to the implementation of the propedeutic course of informatics in primary school by means of Kodu Game Lab]. *Informatika i obrazovanie — Informatics and Education*, 2019, no. 8, p. 14–23. (In Russian.)

Received: August 1, 2019.

Accepted: September 17, 2019.

About the authors

Adel V. Kaplan, Primary School Teacher, School 2009, Moscow, Russia; adel.caplan@ya.ru; ORCID: 0000-0002-8581-5086

Dmitry I. Pavlov, Senior Lecturer at the Department of Theory and Methods of Teaching Mathematics and Informatics, Institute of Mathematics and Informatics, Moscow Pedagogical State University, Russia; di.pavlov@mpgu.su; ORCID: 0000-0002-0074-0899

References

1. Loyo A. H. Effects on the school performance of teaching programming in elementary and secondary schools. *Informatics in Schools. Fundamentals of Computer Science and Software Engineering. Proc. 11th Int. Conf. on Informatics in Schools: Situation, Evolution, and Perspectives*. Springer International Publishing, 2018, p. 30–41. DOI: 10.1007/978-3-030-02750-6_3

2. Gujberova M., Kalas I. Designing productive gradations of tasks in primary programming education. *Proc. 8th Workshop in Primary and Secondary Computing Education*. New York, ACM, 2013, p. 108–117. DOI: 10.1145/2532748.2532750

3. Bosova L. L., Kaplan A. V. Mezhdunarodnaya konferentsiya po shkol'noj informatike ISSEP 2018 [The international conference on informatics in schools ISSEP 2018]. *Informatika v shkole — Informatics in School*, 2018, no. 9, p. 2–6. (In Russian.)

4. Khenner E. K. Vychislitel'noe myshlenie [Computational thinking]. *Obrazovanie i nauka — The Education and Science Journal*, 2016, no. 2, p. 18–33. (In Russian.) DOI: 10.17853/1994-5639-2016-2-18-33

5. Kalash I. Vychislitel'noe myshlenie i doshkol'noe obrazovanie [Developing digital literacy of early childhood education teachers]. *Vospitanie i obuchenie detej mladshego vozrasta — Early Childhood Care and Education*, 2013, no. 1, p. 50. (In Russian.)

6. Bosova L. L. Shkol'naya informatika v Rossii i v mire [School informatics in Russia and in the world]. *Informatizatsiya obrazovaniya i nauki — Informatization of Education and Science*, 2018, no. 3, p. 134–145. (In Russian.)

7. Wing J. M. Computational thinking. *Communications of the ACM — Self managed systems*, 2006, vol. 49, is. 3, p. 33–35. DOI: 10.1145/1118178.1118215

8. Frumin I. D., Dobryakova M. S., Barannikov K. A., Remorenko I. M. Universal'nye kompetentnosti i novaya gramotnost': chemu uchit' segodnya dlya uspekha zavtra

[Key competences and new literacy: from slogans to school reality]. Moscow, HSE, 2018. 28 p. (In Russian.) Available at: https://ioe.hse.ru/data/2018/07/12/1151646087/2_19.pdf

9. Kalaš I., Benton L. Defining procedures in early computing education. *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing. IFIP World Conf. on Computers in Education*. Cham, Springer, 2017, p. 567–578. DOI: 10.1007/978-3-319-74310-3_57

10. Varenina L. P. Gejmifikatsiya v obrazovanii [Gamification in education]. *Istoricheskaya i sotsial'no-obrazovatel'naya mysl' — Historical and Social-Educational Idea*, 2014, vol. 6, no. 6-2, p. 314–317. (In Russian.)

11. Pavlov D. I., Butarev K. V., Balashova E. V. O perspektivakh ispol'zovaniya tekhnologiiy gejmifikatsii pri rannem obuchenii ob'ektno-orientirovannomu programmirovaniyu [Gamification technologies for the early education of object-oriented programming]. *Sovremennye informacionnye tekhnologii i IT-obrazovanie — Modern Information Technologies and IT-Education*, 2018, vol. 14, no. 4, p. 977–985. (In Russian.) Available at: <http://sitito.cs.msu.ru/index.php/SITITO/article/view/465>

12. Papert S. *Mindstorms: children, computers, and powerful ideas*. New York, Basic Books, 1980. 242 p.

13. National Research Council Report of a Workshop on the Scope and Nature of Computational Thinking. Washington, DC, The National Academies Press, 2010. 114 p. DOI: 10.17226/12840

14. Cheburina O. V. Formirovanie algoritmicheskogo myshleniya v obuchenii programmirovaniyu igr [Formation of algorithmic thinking in training programming games] *Nauka i perspektivy — Science and Perspectives*, 2017, no. 2, p. 75–79. (In Russian.) Available at: <http://nip.esrae.ru/ru/14-r116>

15. Pavlov D. I. *Informatika dlya vseh* [Informatics for all]. Moscow, Binom, 2018.

16. Belenov N. V., Samsonova O. S. Robototekhnika vo vneurochnoy deyatel'nosti kak faktor razvitiya tekhnicheskikh sposobnostey u obuchayushhikhsya [Robotics in extracurricular activities as a factor in the development

of technical abilities in students]. *International Scientific Review*, 2015, no. 4, p. 11–15. (In Russian.)

17. Kaplan A. V. Primenenie tekhnologiiy gejmifikatsii v propedevtike programmirovaniya v nachal'noj shkole [Gamification for teaching programming in primary school]. *Informatika v shkole — Informatics in School*, 2018, no. 6, p. 65–67. (In Russian.)

18. Landa L. N. Algoritmizatsiya v obuchenii [Algorithmization in learning]. Moscow, Prosveshchenie, 1966. 523 p. (In Russian.)

19. Vorontsov A. B. Praktika razvivayushhego obucheniya po sisteme D. B. Ehl'konina — V. V. Davydova [The practice of developing education in the system of D. B. Elkonin — V. V. Davydov]. Moscow, 1998. 390 p.

20. Elkonin B. D., Vorontsov A. B., Chudinova E. V. Podrostkovyy etap shkol'nogo obrazovaniya v sisteme Ehl'konina—Davydova [The teenage stage of school education in the Elkonin—Davydov system]. *Voprosy obrazovaniya — Educational Studies Moscow*, 2004, no. 3, p. 118–142. (In Russian.) Available at: <https://www.hse.ru/data/2010/12/31/1208182300/07elkonin118-142.pdf>

21. Blednova E. V. Proektnaya deyatel'nost' kak uslovie razvitiya tvorcheskikh sposobnostey uchashhikhsya nachal'noj shkoly [Project activity as a condition for the development of creative abilities of elementary school students]. *Munitsipal'noe obrazovanie: innovatsii i ehksperiment — Municipal Education: Innovation and Experiment*, 2012, no. 6, p. 28–31. (In Russian.)

22. Pozdeeva S. I., Kuznetsova T. V. Proektnaya deyatel'nost' v praktike uchitelya nachal'noj shkoly [Project activities in the practice of an elementary school teacher]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta — Tomsk State Pedagogical University Bulletin*, 2006, no. 10, p. 65–68. (In Russian.) Available at: https://vestnik.tspu.edu.ru/files/vestnik/PDF/articles/pozdeeva_s_i_65_68_10_61_2006.pdf

23. Polat E. S. Metod proektov: tipologiya i struktura [Project method: typology and structure]. *Litsejskoe i gimnazicheskoe obrazovanie — Lyceum and Gymnasium Education*, 2002, no. 9, p. 9–17. (In Russian.)

НОВОСТИ

Заочная школа новых технологических компетенций начинает обучение школьников

23 октября 2019 года в рамках Московского международного форума «Открытые инновации» прошла пресс-конференция, посвященная началу обучения слушателей в Заочной школе новых технологических компетенций (ЗШНТК). В мероприятии приняла участие заместитель министра просвещения Российской Федерации М. Н. Ракова.

Фонд новых форм развития образования при поддержке негосударственного института развития «Иннопрактика» в партнерстве с Рыбаков Фондом в августе текущего года запустили заочную школу новых технологических компетенций для учащихся общеобразовательных организаций. Цель ЗШНТК — разработка новых механизмов образовательной поддержки детей независимо от места их проживания в математической, естественно-научной и технологической областях.

На первом этапе проекта был проведен конкурс для потенциальных наставников, которыми стали молодые научные сотрудники, аспиранты и магистранты из вузов-партнеров. Они прошли отбор и обучение, сформировали свои инновационные образовательные программы. Наиболее эффективные начинающие преподаватели заочной школы по итогам работы получают

денежное вознаграждение в размере 150 тыс. рублей. Оценка результативности наставников пройдет с помощью электронного автоматизированного анализа данных во время и после обучения.

Основные направления ЗШНТК: биотехнологии, робототехнические технологии, нанотехнологии, информационные технологии, космические технологии, фундаментальные науки (математика, физика, химия).

До 10 октября проводился второй этап проекта, когда школьники из разных регионов страны подавали свои заявки на участие в ЗШНТК и проходили тестирование. Набор первой волны слушателей завершился, и заочная школа открывает свои виртуальные двери. В настоящее время уже начинают обучение около 500 школьников, попавших в первую волну слушателей, на специально созданной платформе по девяти программам. К концу года количество обучающихся может достигнуть 3000 человек.

Обучение будет проходить преимущественно в онлайн-формате, а также на базе учреждений дополнительного образования, оснащенных высокотехнологичным оборудованием, в том числе на площадках сети детских технопарков «Кванториум» и сети детских центров «ИТ-куб».

(По материалам федерального портала «Российское образование»)