

ИССЛЕДОВАНИЕ РАЗРАБОТКИ ИГР НА PYTHON

АМАН К.П. , МУСИНА А.А. * 

Аман Кулнар Панабековна – К.т.н., доцент кафедры информатики и информационных технологий, Актюбинский региональный университет имени К.Жубанова, г. Ақтөбе, Қазақстан

E-mail: kaman@zhubanov.edu.kz, <https://orcid.org/0000-0003-4179-4241>

*Мусина Алла Александровна - PhD, старший преподаватель кафедры информатики и информационных технологий, Актюбинский региональный университет имени К.Жубанова, г. Ақтөбе, Қазақстан

E-mail: a.mussina@zhubanov.edu.kz, <https://orcid.org/0000-0002-0643-2280>

Аннотация. Целью исследования являлось выявление влияния разработанного авторами обучающего приложения на развитие навыков программирования 2D игр у учеников 9 класса. Гипотеза исследования заключалась в том, чтобы решить проблему, связанную с трудностями в освоении новых концепций и алгоритмов и нехваткой времени для изучения программирования через разработанное обучающее приложение на Python.

В исследовании авторами были применены такие методы, как опрос учеников, эксперимент, сбор и анализ данных о взаимодействии учеников с приложением и тестирование. Сравнение результатов по ключевым показателям, таким как успеваемость, качество знаний и обученность в двух классах показало, что успеваемость, качество знаний и обученность выше в классе, где проводился эксперимент с использованием обучающего приложения.

Анализ данных о взаимодействии учеников с приложением и результаты тестов продемонстрировали закономерность, что при переходе от одного уровня сложности к другому, у учеников не упал интерес к обучению и изучению программирования, а также, что они успешно справлялись со всеми разноуровневыми заданиями.

Исследование показало, что разделение приложения на три уровня сложности (начальный, средний, высокий) оказалось эффективным подходом. Каждый уровень позволял учащимся систематически углублять свои знания и навыки, что способствовало более глубокому и устойчивому обучению.

Ключевые слова: библиотека Python, обучающее приложение, разработка 2D игр, программа, игра, 2D игра, программирование 2 D игры, программирование игр.

Введение

В современном мире информационных технологий программирование становится все более важным навыком, открывающим двери в мир возможностей и новых решений. Особенно важное внимание стоит уделять обучению программированию в школьной программе.

Рассматриваемая тема актуальна на сегодняшний день, так как наблюдается рост интереса к IT-профессиям, и развитие навыков программирования 2D игр может быть отличным введением в мир программирования. Программирование способствует развитию логического мышления и способности анализировать. Поэтому знания и навыки программирования, полученные на уроках по программированию игр могут пригодиться в будущем ученикам, подготавливая их к различным IT-профессиям.

Цель исследования заключается в изучении влияния обучающих приложений на усвоение программистских концепций школьниками.

Проблемы в обучении программированию учеников 9 класса связаны с несколькими факторами такими, как трудности в освоении новых концепций и алгоритмов, нехватка времени для изучения программирования [1].

Гипотеза исследования заключалась в том, чтобы решить эту проблему через разработанное обучающее приложение на языке программирования Python для школьников 9 класса, которое должно помочь им эффективнее и быстрее освоить темы, связанные с программированием 2D игр.

Анализ литературных источников по методике развития навыков программирования показывает, что данная область также имеет свою историю и эволюцию. Сеймур Паперт

(Seymour Papert) - автор методики «Logo», который разработал ее в 1960-х годах и использовал для обучения детей [2].

Митчел Резник (Mitchel Resnick) - создал программу Scratch, которая позволяет разрабатывать мультимедийные проекты, игры и анимации [3].

Джейн Марголис (Jane Margolis) - автор книги «Unlocking the Clubhouse: Women in Computing», исследовала проблемы разнообразия в области информационных технологий и предложила стратегии по привлечению большего числа женщин в программирование [4].

Методики Пола Граньер (Paul Gruenewald) и Марка Фелдмана (Mark J. Feldman) помогли измерить влияние обучения на развитие навыков программирования [5].

Линда Лиердал (Linda Liukas) - автор книги «Hello Ruby» и методики обучения программированию с использованием рассказов и героев. Её работы ориентированы на обучение детей и стимулирование их интереса к программированию [6].

Материалы и методы исследования

Разработка приложения осуществлялась с использованием языка программирования Python. Приложение состоит из нескольких модулей, каждый из которых ориентирован на конкретный аспект программирования. Задания включают как теоретические вопросы, так и практические задачи. Для оценки навыков пользователей применялись тесты до и после использования приложения, а также сбор обратной связи через анкеты [7, 15].

Перед началом разработки обучающего приложения необходимо предварительно установить определенный набор библиотек и инструментов. В данном случае были выбраны следующие библиотеки:

- Tkinter, это стандартная библиотека для создания графического пользовательского интерфейса (GUI) в Python. Она предоставляет набор виджетов и функционала для создания окон и элементов управления, что позволяет разработчику взаимодействовать с пользователем через графический интерфейс. С помощью этой библиотеки был разработан интерфейс всех вкладок приложения.

- PIL (Python Imaging Library) является библиотекой для работы с изображениями в Python. Она предоставляет множество функций для открытия, редактирования и сохранения изображений различных форматов [8].

- cv2 (OpenCV) является библиотекой компьютерного зрения, и предоставляет широкий спектр инструментов для обработки изображений и видео.

- PyGame, библиотека для разработки 2D игр и мультимедийных приложений на Python. Она обеспечивает инструменты для создания игровой логики, отрисовки графики и воспроизведения звуковых эффектов. В данном случае были все коды программ для школьников написаны с помощью этой библиотеки [9].

Процесс установки этих библиотек может различаться в зависимости от операционной системы. Обычно, они могут быть установлены с помощью менеджера пакетов Python, такого как pip [10].

Данные об эффективности освоения тем, связанных с программированием 2D игр, собирались на этапе рефлексии и этапе оценивания на каждом уроке, что позволяло оценивать прогресс изучения темы и выявлять проблемные области, требующие доработки.

В исследовании авторами были применены следующие методы:

1. Опрос учеников. Вопросы были количественными, оценки от 1 до 10.
2. Экспериментальное исследование. Был проведен эксперимент, где один из параллельных классов обучался с использованием обучающего приложения, а другой класс - без него. Сравнение результатов по ключевым показателям, таким как успеваемость и усвоение материала.

3. Анализ данных. Был проведен сбор и анализ данных о взаимодействии учеников с приложением, то есть количество выполненных заданий, время, проведенное в приложении, результаты тестов с целью выявления закономерностей.

4. Тестирование. Было проведено предварительное и контрольное тестирование для определения уровня знаний до и после использования обучающего приложения.

В эксперименте участвовали два параллельных 9-х класса из Мичуринского комплекса «Школа-ясли-детский сад» ЗКО, район Байтерек В экспериментальном классе обучались 24 ученика, в контрольном - 22. Критерием отбора выступало отсутствие у школьников предварительного опыта разработки игр и базовые знания по информатике, полученные в рамках школьной программы.

Для обеспечения корректности статистических выводов были приняты меры по контролю внешних факторов:

- занятия вел один и тот же учитель;
- оба класса изучали одинаковый теоретический материал;
- количество часов, выделенных на изучение темы, было идентичным;
- обучение проводилось в одно и то же время суток, чтобы исключить влияние утомляемости.

Такой подход обеспечил сопоставимость условий и позволил объективно оценить влияние обучающего приложения на результаты обучения.

Этапы исследования влияния использования обучающего приложения на развитие навыков программирования 2D игр включали следующие шаги:

1. На этапе постановки проблемы авторами была определена проблема и поставлена цель ее решения. Была выдвинута гипотеза исследования.
2. На этапе литературного обзора были изучены существующие методики и подходы к обучению программированию 2D игр, проведен анализ научных статей, публикаций и книг по данной теме для выявления лучших практик и подходов.
3. Далее были выбраны методы исследования, которые наилучшим образом подошли для достижения цели исследования.
4. Были выдвинуты задачи исследования, которые включали в себя создание заданий и упражнений, направленных на развитие навыков программирования 2D игр у учеников.
5. На этапе проведения эксперимента был реализован обучающий курс, проведены занятия, и оценка результатов и эффективности выбранных методов и подходов.
6. Далее авторами был проведен анализ результатов и обработка полученных данных, сравнение результатов до и после обучения.
7. Были сформулированы выводы и рекомендации.

Технические ограничения и особенности реализации приложения:

Разработанное обучающее приложение функционирует в операционной системе Windows и корректно работает на компьютерах с минимальной конфигурацией: процессор не ниже Intel Core i3, 4 ГБ оперативной памяти и установленный интерпретатор Python версии 3.10 и выше.

Для полноценной работы требуются дополнительные библиотеки - Tkinter, PIL, PyGame и OpenCV. Их установка осуществляется через менеджер пакетов *pip*. Архитектура приложения модульная, что позволяет при необходимости расширять функционал, добавлять новые задания и уровни сложности.

Проверка корректности выполнения заданий реализована через встроенный механизм анализа кода: приложение автоматически отслеживает выполнение ключевых операций, корректность синтаксиса и обработку игровых событий. В случаях, требующих творческого подхода, предусматривалась дополнительная ручная проверка преподавателем.

Таким образом, приложение обладает потенциалом масштабирования, однако в текущей версии не поддерживает мобильные операционные системы и требует базовых навыков работы с Python.

Результаты и их обсуждение

В разработанном обучающем приложении для программирования игр на Python были реализованы три уровня сложности, каждый из которых охватывает определенные навыки и концепции программирования игр. На главной странице приложения имеется возможность выбора уровня сложности (рисунок 1).

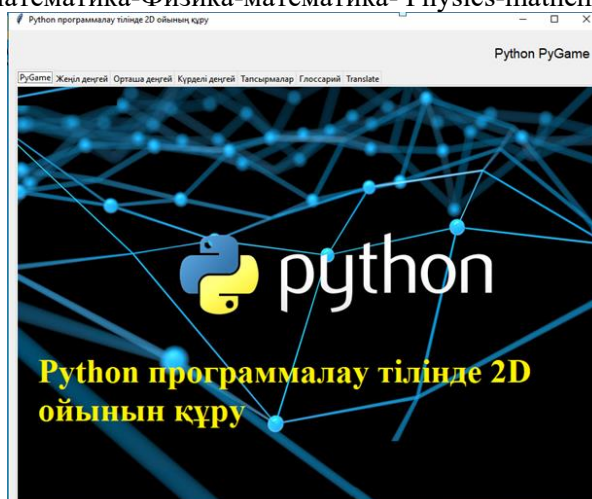


Рисунок 1. Главная страница приложения

Ниже представлены детальные описания каждого уровня и полученные результаты:
1. Начальный уровень (рисунок 2).

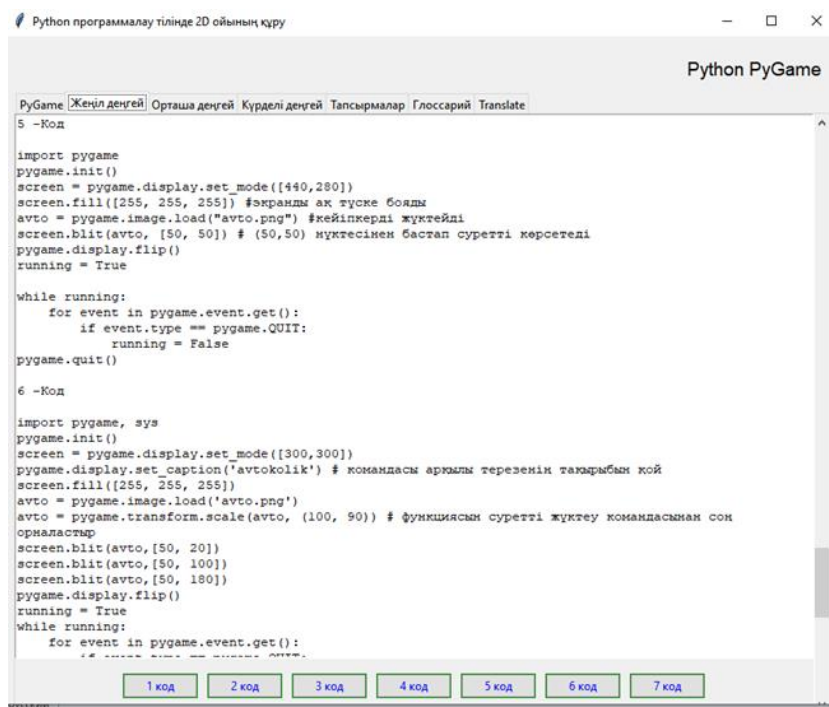


Рисунок 2 Начальный уровень

Навыки и концепции:

1. Основы синтаксиса Python.
2. Создание игрового окна и отображение графики.
3. Обработка событий и управление игрой (рисунок 3).

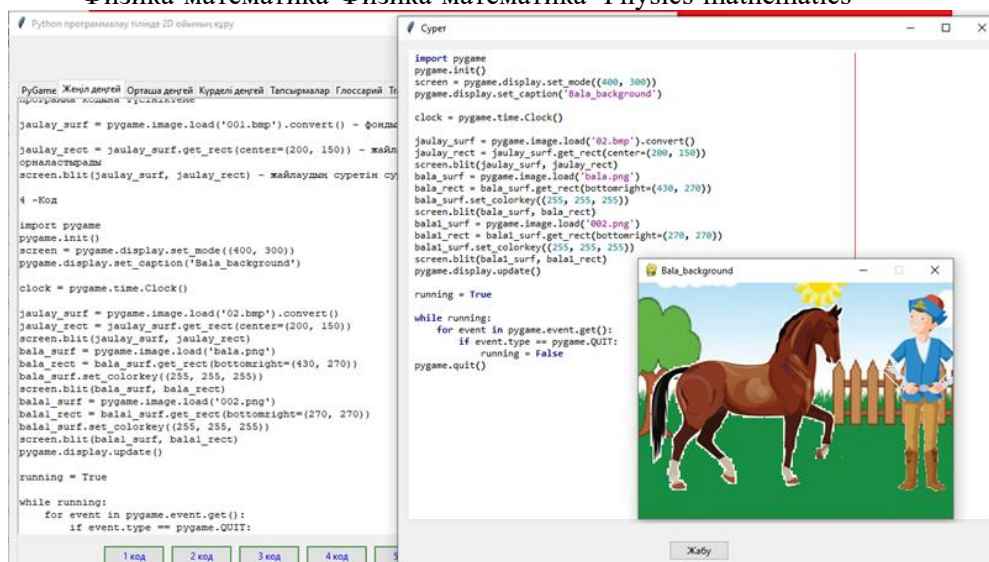


Рисунок 3. Всплывающее окно с исполненным кодом

2. Средний уровень (рисунок 4).

Навыки и концепции:

1. Операции с изображениями и звуками, движущиеся изображения.
2. Создание более сложной игровой логики.
3. Работа с коллизиями и физикой игры.

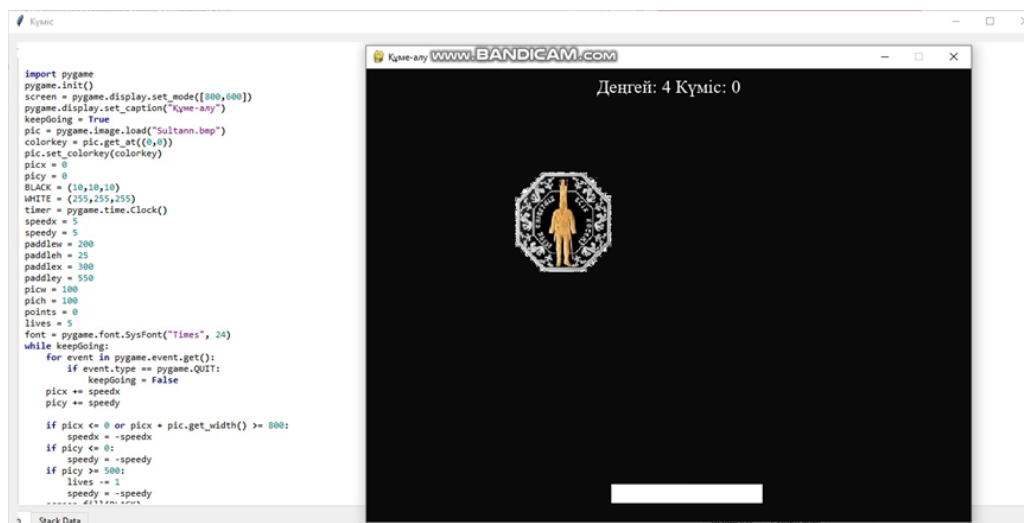


Рисунок 4. Средний уровень

3. Высокий уровень (рисунок 5).

Навыки и концепции:

1. Продвинутая работа с анимацией и многозадачностью.
2. Использование внешних библиотек и модулей.
3. Создание игр с уровнями и сохранением состояния.

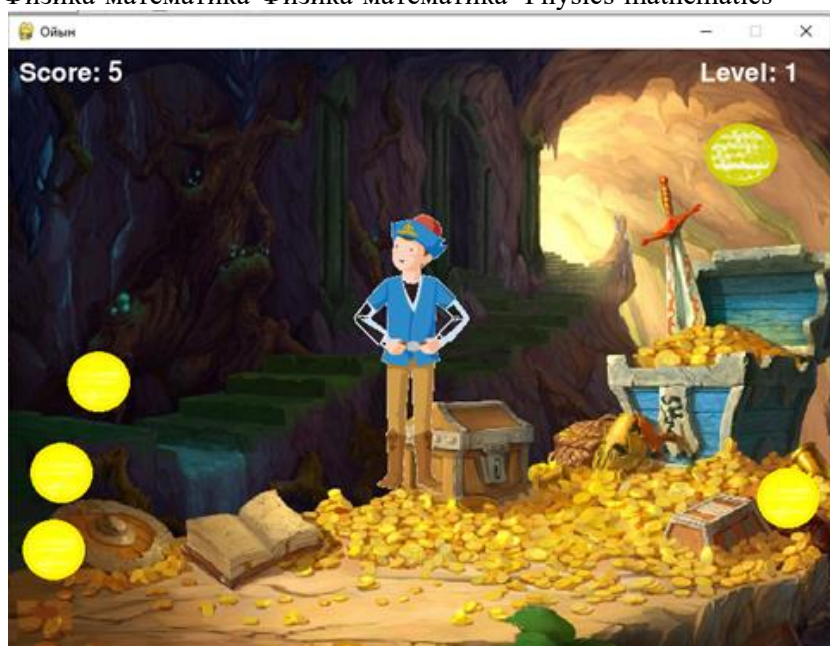


Рисунок 5. Создание сложной игры

Был проведен эксперимент, где один из параллельных классов обучался с использованием обучающего приложения, а другой класс - без него. Сравнение результатов по ключевым показателям, таким как успеваемость, качество знаний и обученность приведено в таблице 1 и на рисунке 6.

Таблица 1. Сравнение ключевых характеристик в двух классах.

	класс, не использующий приложение	класс, использующий приложение
Успеваемость	95	100
Качество знаний	66	87
Обученность	54	90

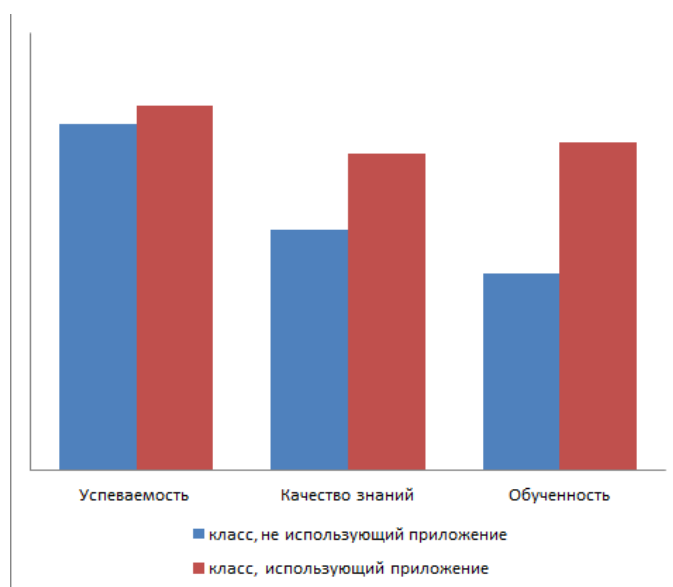


Рисунок 6. Сравнение ключевых параметров.

Ключевые параметры измерялись по следующим формулам:

Успеваемость = (количество «5» + количество «4» + «количество «3»)/ общее количество учащихся.

Качество знаний = (количество «5» + количество «4»)/ общее количество учащихся.

Обученность = (количество «5» + количество «4» * 0,64 + количество «3» * 0,36 + количество «2» * 0,16 + количество «неатестованных» * 0,08)/ общее количество учащихся.

Был проведен сбор и анализ данных о взаимодействии учеников с приложением, то есть количество выполненных заданий, время, проведенное в приложении, результаты тестов с целью выявления закономерностей, который отражен в таблице 2.

Таблица 2. Выявление закономерностей

	1 уровень сложности (всего 50 заданий)	2 уровень сложности (всего 25 заданий)	3 уровень сложности (всего 20 заданий)
Среднее количество выполненных заданий на одного ученика (штук)	45	22	15
Среднее время проведенное в приложении одним учеником (час)	10	12	15
Средний результат тестов по классу (балл)	9	8	8

Было проведено предварительное и контрольное тестирование для определения уровня знаний до и после использования обучающего приложения, результаты которого отражены на рисунке 7.

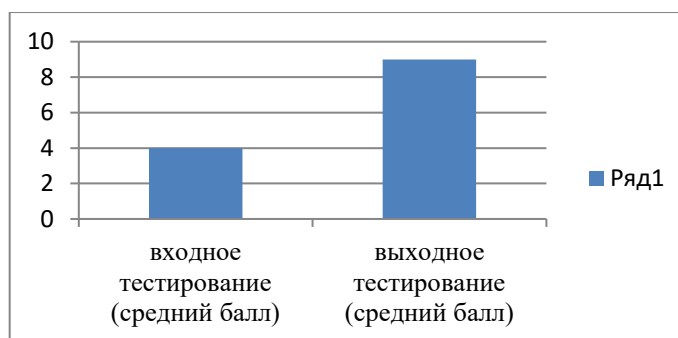


Рисунок 7. Результаты входного и выходного тестирования.

На начальном уровне школьники овладели основами и научились создавать игровое окно. Значительно увеличился интерес к программированию и пришло понимание основных концепций игровой разработки.

На среднем уровне школьники научились создавать более сложные игры с разнообразными графическими и звуковыми эффектами.

На высоком уровне школьники способны были создавать игры с продвинутыми возможностями и сложными уровнями.

В итоге, разделение приложения на уровни сложности позволило школьникам постепенно углублять свои знания и навыки в программировании игр, начиная с базовых концепций и постепенно переходя к более сложным аспектам разработки. Этот подход

способствует более эффективному и глубокому обучению и позволяет школьникам достигать значительных результатов в программировании игр.

Сравнение результатов по ключевым показателям, таким как успеваемость, качество знаний и обученность в двух классах показало, что успеваемость, качество знаний и обученность выше в классе, где проводился эксперимент с использованием обучающего приложения.

Анализ данных о взаимодействии учеников с приложением, то есть количество выполненных заданий, время, проведенное в приложении, результаты тестов продемонстрировали закономерность, что при переходе от одного уровня сложности к другому, у учеников не упал интерес к изучению программирования. Проектирование и разработка цифровых игр в образовательных целях требуют сочетания педагогических, технологических и психологических подходов [11].

Аналитическое обсуждение результатов

Полученные данные свидетельствуют о наличии выраженной зависимости между временем работы учащихся в приложении и результатами контрольного тестирования: ученики, проводившие в приложении более 45 минут на каждом занятии, показали средний рост баллов на 27% по сравнению с учениками, у которых время работы составляло менее 20 минут. Это подтверждает, что практическое взаимодействие с игровыми механизмами усиливает понимание алгоритмических конструкций.

Различия между уровнями сложности также оказались закономерными. На начальном уровне мотивация учеников поддерживалась за счёт быстрого получения результата - возможность сразу увидеть созданный объект на экране формировала эффект успеха. На среднем уровне интерес сохранялся благодаря появлению новых элементов (звук, движение, коллизии), что стимулировало поисковую активность. На высоком уровне наблюдалось существенное усложнение логики, однако ученики, прошедшие предыдущие этапы, демонстрировали устойчивые навыки и уверенность, что снижало риск отказа от деятельности.

Таким образом, постепенное усложнение заданий выступило ключевым фактором успешного обучения, а игровой формат позволил поддерживать высокий уровень вовлечённости на всех этапах эксперимента.

Заключение

Результаты нашего исследования свидетельствуют о важной роли обучающих приложений в контексте образования и развития программистских навыков у школьников. Проведенное исследование позволило авторам оценить влияние приложения на знания, навыки и мотивацию учащихся в области программирования игр.

Исследование показало, что разделение приложения на три уровня сложности оказалось эффективным подходом. Каждый уровень позволял учащимся систематически углублять свои знания и навыки, что способствовало более глубокому и устойчивому обучению.

Исходя из полученных результатов, авторы рекомендуют дальше развивать обучающее приложение. Это может включать в себя расширение функциональности, добавление дополнительных уровней сложности, а также интеграцию дополнительных инструментов и ресурсов для обучения.

Результаты предварительного и контрольного тестирования для определения уровня знаний до и после использования обучающего приложения, показали, что после изучения темы значительно вырос показатель знаний учеников.

Для совершенствования обучающих приложений в образовательном процессе, были разработаны следующие рекомендации:

1. Использование алгоритмов машинного обучения и искусственного интеллекта для адаптации контента и уровня сложности к индивидуальным потребностям и уровню знаний каждого ученика.
2. Обеспечение обучения через практическую деятельность и задания. Интерактивные элементы, включая задачи, упражнения и проекты, могут значительно повысить

эффективность обучения.

3. Использование встроенных инструментов аналитики могут помочь учителям и ученикам отслеживать прогресс, определять слабые места и адаптировать обучение соответственно.

Список литературы

1. Якименко О. В. Применение обучающих программ-тренажеров в обучении программированию // Вестник ТГПУ. – Томск, 2019.
2. The Children's Machine: Rethinking School in the Age of the Computer // Choice Reviews Online. – 1993. – Vol. 31, No. 03. – P. 31–1648. – DOI: [10.5860/choice.31-1648](https://doi.org/10.5860/choice.31-1648).
3. Kafai Yasmin Bettina, Resnick Mitchel. Constructionism in Practice: Designing, Thinking, and Learning in a Digital World. – 1996. – Режим доступа: <http://ci.nii.ac.jp/ncid/BA28658289>.
4. Byrne Jake Rowan, Fisher Lorraine, Tangney Brendan. A 21st Century Teaching and Learning Approach to Computer Science Education: Teacher Reactions // Communications in Computer and Information Science. – 2016. – P. 523–540. – DOI: [10.1007/978-3-319-29585-5_30](https://doi.org/10.1007/978-3-319-29585-5_30).
5. Feldman J. The structure of perceptual categories // Journal of Mathematical Psychology. – 2020. – Vol. 41, No. 2. – P. 145–170.
6. Pérez-Marín D. Teaching programming in early childhood education with stories // ICERI2019 Proceedings. – IATED, 2019. – P. 9206–9212.
7. Эрик М. Изучаем Python: программирование игр, визуализация данных, веб-приложения. – 3-е изд. – Санкт-Петербург: Издательский дом «Питер», 2019.
8. Чарушин Ф. Д., Бужинская Н. В. Этапы проектирования 2D-игры платформера // Наука и перспективы. – 2023. – № 1. – С. 75–82.
9. Белолубский М. М., Сорочинский М. А. Изучение модуля «Pygame» для создания игр на языке программирования Python // Современное образование: традиции и инновации. – 2020. – № 2. – С. 220–222.
10. Quintero Carlos Andrés Caldas, Cifuentes Álvarez Gary Alberto. Serious Games and Computer Programming Competencies Development in Educational Contexts // IEEE Revista Iberoamericana de Tecnologías del Aprendizaje. – 2023. – Vol. 18, No. 1. – P. 48–53. – DOI: [10.1109/rita.2023.3250504](https://doi.org/10.1109/rita.2023.3250504).

References

1. YAkimenko O. V. Primenenie obuchayushchih programm-trenazhyorov v obuchenii programmirovaniyu // Vestnik TGPU. – Tomsk, 2019.
2. The Children's Machine: Rethinking School in the Age of the Computer // Choice Reviews Online. – 1993. – Vol. 31, No. 03. – P. 31–1648. – DOI: [10.5860/choice.31-1648](https://doi.org/10.5860/choice.31-1648).
3. Kafai Yasmin Bettina, Resnick Mitchel. Constructionism in Practice: Designing, Thinking, and Learning in a Digital World. – 1996. – Режим доступа: <http://ci.nii.ac.jp/ncid/BA28658289>.
4. Byrne Jake Rowan, Fisher Lorraine, Tangney Brendan. A 21st Century Teaching and Learning Approach to Computer Science Education: Teacher Reactions // Communications in Computer and Information Science. – 2016. – P. 523–540. – DOI: [10.1007/978-3-319-29585-5_30](https://doi.org/10.1007/978-3-319-29585-5_30).
5. Feldman J. The structure of perceptual categories // Journal of Mathematical Psychology. – 2020. – Vol. 41, No. 2. – P. 145–170.
6. Pérez-Marín D. Teaching programming in early childhood education with stories // ICERI2019 Proceedings. – IATED, 2019. – P. 9206–9212.
7. Erik M. Izuchaem Python: programmirovaniye igr, vizualizatsiya dannyh, veb-prilozheniya. – 3-e izd. – Sankt-Peterburg: Izdatel'skiy dom «Piter», 2019.
8. CHarushin F. D., Buzhinskaya N. V. Etapy proektirovaniya 2D-igry platformera // Nauka i perspektivy. – 2023. – № 1. – S. 75–82.
9. Belolyubskiy M. M., Sorochinskij M. A. Izuchenie modulya «Pygame» dlya sozdaniya igr na yazyke programmirovaniya Python // Sovremennoe obrazovanie: tradicii i innovacii. – 2020. – № 2. – S. 220–222.
10. Quintero Carlos Andrés Caldas, Cifuentes Álvarez Gary Alberto. Serious Games and

PYTHON-ДА ОЙЫНДАРДЫ ӘЗІРЛЕУДІ ЗЕРТТЕУ

АМАН К.П. , МУСИНА А.А. * 

Аман Кулнар Панабекқызы – Т.ғ.к., информатика және ақпараттық технологиялар кафедрасының доценті, Қ. Жұбанов атындағы Ақтөбе өңірлік университеті, Ақтөбе қ., Қазақстан
E-mail: kaman@zhubanov.edu.kz, <https://orcid.org/0000-0003-4179-4241>

***Мусина Алла Александровна** - PhD, информатика және ақпараттық технологиялар кафедрасының аға оқытушысы, Қ. Жұбанов атындағы Ақтөбе өңірлік университеті, Ақтөбе қ., Қазақстан
E-mail: a.mussina@zhubanov.edu.kz, <https://orcid.org/0000-0002-0643-2280>

Анатпа. Зерттеудің мақсаты 9-сынып оқушыларында 2D ойындарын бағдарламалау дағдыларын дамытуға авторлар әзірлеген оқыту қосымшасының әсерін анықтау болды. Зерттеудің гипотезасы жаңа тұжырымдамалар мен алгоритмдерді меңгерудегі қиындықтарға және Python-да әзірленген оқу қолданбасы арқылы бағдарламалауды үйренуге уақыттың жетіспеушілігіне байланысты мәселені шешу болды.

Зерттеуде авторлар студенттерге сауалнама жүргізу, эксперимент жүргізу, оқушылардың қолданбалы өзара әрекеттесу деректерін жинау және талдау, тестілеу сияқты әдістерді қолданды. Екі сыныпта оқу үлгерімі, білім сапасы және оқыту сияқты негізгі көрсеткіштер бойынша нәтижелерді салыстыру оқу қолданбасын пайдалана отырып эксперимент жүргізілген сыныпта оқу үлгерімі, білім сапасы және оқыту жоғары екенін көрсетті.

Оқушылардың қосымшамен өзара әрекеттесуі туралы деректерді талдау және тест нәтижелері бір қиындық деңгейінен екіншісіне ауысқан кезде оқушылардың бағдарламалауды үйренуге және үйретуге деген қызығушылығы төмендегендігін, сондай-ақ барлық көп деңгейлі тапсырмаларды сәтті орындағанын көрсетті.

Зерттеу қолданбаны үш қиындық деңгейіне (бастапқы, орта, жоғары) бөлу тиімді тәсіл екенін көрсетті. Әр деңгей студенттерге өздерінің білімдері мен дағдыларын жүйелі түрде тереңдетуге мүмкіндік берді, бұл нәтижелі және тұрақты оқуға ықпал етті.

Бұл жұмыстың ғылыми жаңалығы жаңадан бастаушылар үшін бағдарламалауға және оқушыларды ынталандыруға, жаңадан үйренушілерге кедергіні азайту сияқты өзекті білім беру және технологиялық мәселелерді шешуде жатыр.

Түйін сөздер: Python кітапханасы, оқыту қолданбасы, 2D ойын әзірлеу, бағдарлама, ойын, 2D ойын, 2D ойынды бағдарламалау, ойындарды бағдарламалау.

RESEARCH ON GAME DEVELOPMENT IN PYTHON

AMAN K.P. , MUSSINA A.A. * 

Aman Kulnar Panabekovna – Candidate of technical sciences, docent, department of computer science and information technologies, Zhubanov Aktobe regional university, Aktobe, Kazakhstan.

E-mail: kaman@zhubanov.edu.kz, <https://orcid.org/0000-0003-4179-4241>

***Mussina Alla Alexandrovna** - PhD, senior lecturer, department of computer science and information technologies, Zhubanov Aktobe regional university, Aktobe, Kazakhstan.

E-mail: a.mussina@zhubanov.edu.kz, <https://orcid.org/0000-0002-0643-2280>

Abstract. The purpose of the study was to create a specially developed educational application for the development of 2D game programming skills among 9th grade teachers. The hypothesis of the study was to solve the problem of the difficulty in creating new concepts and algorithms and the lack of time for software development through a developed learning application in Python.

In the research, the authors used methods such as a teacher survey, an experiment, and the collection and analysis of data on teacher interaction with the app and testing. A comparison of the results on key indicators such as success, quality of knowledge, and learnability in the two classes showed that success, quality of knowledge, and learnability were higher in the classroom where the experiment was conducted using a learning application.

An analysis of teachers' profit and loss data with the application and test results demonstrated the legitimacy that, when moving from one level of difficulty to another, teachers did not lose interest in teaching and learning programming, and also that they successfully spread with all multi-level tasks.

The study showed that dividing the application into three levels of difficulty (initial, medium, and high) proved to be an effective approach. Each level allowed students to systematically deepen their knowledge and skills, which allowed

for deeper and more consistent learning.

The scientific novelty of this work lies in solving current educational and technological problems, such as reducing barriers for beginners in programming and motivating students.

Key words: Python library, educational application, 2D game development, program, game, 2D game, programming 2 D games, game programming.