

МЕТОДИ НАВЧАННЯ ШКОЛЯРІВ ПРОГРАМУВАННЮ НА ОСНОВІ СТВОРЕННЯ КОМП'ЮТЕРНОЇ ГРИ

Павло Квак,
Запорізький національний університет
Людмила Чернікова,
КЗ «ЗОІППО» ЗОР

У статті аналізуються методи навчання програмуванню школярів на основі створення комп'ютерних ігор, висвітлюючи теоретичні та практичні аспекти цього процесу. У дослідженні розглядаються основні інструменти та мови програмування, що використовуються у процесі створення ігор, включаючи Scratch, Unity та Python з Pygame. Автори наводять покроковий алгоритм розробки простої комп'ютерної гри, від планування до тестування. Описано виклики та стратегії подолання труднощів, як-от обмежений час, бюджет або доступ до технологій. Проаналізовано перспективи застосування цього методу в освітніх ініціативах, з акцентом на технологічні інновації, які можуть змінити підходи до навчання програмуванню.

Ключові слова: навчання програмуванню в школі, створення ігор, Scratch, Unity, Python, Pygame, ігрова педагогіка, покроковий алгоритм розробки ігор, освітні технології, інновації в освіті.

Постановка проблеми. У сучасному світі, де технології швидко розвиваються та перетворюють майже кожен аспект нашого повсякденного життя, вміння програмувати стає однією з найцінніших навичок, які може освоїти молоде покоління. Програмування не лише відкриває двері до численних професійних можливостей у галузі технологій, але й сприяє розвитку логічного й обчислювального мислення, уміння розв'язувати проблеми. Через свою актуальність, вивчення програмування у школах стає все більш популярним, і використання практичних методів навчання, а саме створення комп'ютерних ігор, вважається одним із найефективніших підходів [1].

Підхід до навчання через проєктування та розробку ігор дає змогу учням не тільки оволодіти технічними аспектами програмування, але й розвивати критичне мислення, творчість і командну взаємодію [2]. Цей метод підвищує мотивацію учнів, оскільки вони бачать безпосередній результат своєї роботи у вигляді функціональної гри, яка є цікавою та залучає здобувачів освіти до навчання. Також це дає учням можливість стикнутися з реальними технічними викликами та творчо знайти шляхи їх вирішення, готує їх до майбутніх професій в індустрії цифрових технологій.

Аналіз останніх досліджень і публікацій. Гра завжди викликала інтерес у мислителів різних епох від Платона та Арістотеля до Гегеля та Шиллера, її вивчали вчені, як-от Д. Ушинський і С. Рубінштейн. У контексті освітніх закладів, дослідження використання ігор зосереджені переважно на ділових

(І. Макаренко, М. Касьяненко, О. Шерман [16]) і рольових (А. Думінш, Л. Зайцева) іграх [4]. Теоретичне осмислення дидактичних ігор здійснювали А. Капська, І. Носаченко й інші дослідники. Вивчення використання комп'ютерних ігор в освітніх цілях є важливим напрямком у дослідженні комп'ютерної освіти, де вчені (Р. Горохова, П. Нікітін) вважають, що ігри максимально розкривають дидактичні можливості комп'ютерів, значно підвищуючи ефективність освітнього процесу та визнаючи гру однією з найбільш ефективних методик у комп'ютерному навчанні [3].

Метою статті є аналіз і прикладний аспект ефективності методів навчання програмуванню школярів через створення комп'ютерних ігор, а також оцінка потенціалу ігрового підходу як засобу підвищення мотивації та зацікавленості учнів.

Виклад основного матеріалу. Основні теоретичні аспекти. Аналіз методів навчання програмуванню школярів дозволяє виділити різноманітні підходи та платформи, які застосовуються в сучасних освітніх закладах по всьому світу. Зокрема, стає все більш популярним використання інтерактивних платформ і ігрових додатків, які не тільки допомагають учням засвоїти основи мов програмування, але й розвивають логічне мислення та алгоритмічні навички [5], також можна самостійно опанувати певні теми та курси [11]. Прикладами таких платформ можуть бути Scratch, Code.org, а також спеціалізовані ігрові середовища, де учні можуть створювати власні проекти й ігри.

Для створення таблиці, яка відображає аналіз методів навчання програмуванню школярів, ми обрали такі критерії: метод навчання, цільова аудиторія, основні переваги, ключові технології та інструменти, а також основні виклики.

Таблиця 1

Основні методи навчання програмуванню школярів

Метод навчання	Цільова аудиторія	Основні переваги	Ключові технології та інструменти	Основні виклики
Ігрові платформи	Молодші класи	Інтерактивність, залученість	Scratch, Blockly	Обмежена складність завдань
Онлайн-курси	Середня школа	Доступність, гнучкість навчання	Codecademy, Khan Academy	Потреба в самодисципліні
Проекти з програмування	Старші класи	Практичний досвід, розвиток критичного мислення	GitHub, Arduino	Висока складність проєктів
Класичні лекції та лабораторні роботи	Всі вікові групи	Структуроване та систематичне навчання	Текстові редактори, компілятори	Недостатність практичного досвіду
Гейміфікація	Молодші та середні класи	Мотивація через гру, веселощі	Minecraft Education Edition	Можливість відволікань від навчання

У таблиці наведено структуровані методи навчання програмуванню, забезпечуючи широкий огляд того, як можна застосовувати ці методи в середній освіті.

Однією з ключових тенденцій у сфері освіти є гейміфікація, яка включає використання елементів ігрового дизайну в освітньому процесі. Гейміфікація сприяє залученню учнів через механіки винагороди, прогресії рівнів і змагальності, що мотивує учнів досліджувати та глибше занурюватися в матеріал [10]. Зокрема, у контексті навчання програмуванню, гейміфіковані завдання дозволяють учням переживати радість від успіхів у рішенні складних задач, розвиваючи при цьому витривалість і наполегливість у досягненні поставлених цілей [9]. Використання гейміфікації може значно покращити досвід навчання та водночас підвищити зацікавленість учнів у вивченні програмування, що є особливо важливим у вік цифрових технологій [7, 12, 13].

Розглянемо практичні аспекти створення ігор. Розробка комп'ютерних ігор як метод навчання програмуванню для школярів включає застосування різних інструментів і мов програмування, які дозволяють ефективно впроваджувати проектно-орієнтований підхід в освітній процес [14]. Один із найпопулярніших інструментів серед початківців – це Scratch, який дозволяє створювати ігри за допомогою візуального програмування, що ідеально підходить для молодших школярів. Scratch спрощує процес програмування завдяки інтуїтивно зрозумілому інтерфейсу і робить навчання цікавим і доступним.

Для старших учнів і більш складних проектів можна використовувати Unity, який дозволяє створювати більш розширені 2D та 3D ігри. Unity використовує C# для програмування, що вимагає більш глибокого розуміння концепцій програмування, але водночас надає учням важливий досвід роботи з професійними інструментами, які застосовуються в ІТ-індустрії [6].

Також у навчальних цілях може застосовуватися Python з бібліотекою Pygame, що є чудовим вибором для тих, хто вже має базові знання у програмуванні та хоче створювати власні ігри на Python. Pygame сприяє розумінню, як можна маніпулювати об'єктами в ігровому просторі, обробляти події та створювати візуальні ефекти, що робить її ідеальною для освітніх проектів, де акцент робиться на практичне застосування набутих навичок [8].

Наведемо алгоритм розробки простої комп'ютерної гри.

Розробка комп'ютерної гри, навіть простої, може бути захоплюючим і водночас складним процесом. Нижче описано п'ять основних етапів, які включають планування, дизайн, кодування, тестування та ітерації на прикладі розробки невеликої гри «Лабіринт» у Scratch.

1. Планування.

- Опис завдання. Визначення мети гри та її основної механіки. Наприклад, у грі «Лабіринт» мета полягає в тому, щоб допомогти персонажу знайти шлях від початкової точки до виходу з лабіринту.

- Сценарій гри. Описуємо, як гравець буде взаємодіяти з грою та які перешкоди він може зустріти.

2. Дизайн.

- Створення макету. Розробка простого лабіринту в графічному редакторі Scratch, визначення місць розташування перешкод і бонусів.
- Дизайн персонажа. Створення або вибір персонажа, який буде переміщатися по лабіринту.

3. Кодування.

- Рух персонажа. Програмування основних команд для переміщення персонажа за допомогою стрілок на клавіатурі.
- Логіка гри. Кодування умов перемоги та поразки, наприклад, гра закінчується перемогою, коли персонаж досягає виходу.

4. Тестування.

- Виявлення помилок. Перевірка гри на наявність помилок у логіці руху та виконання заданих умов.
- Збір відгуків. Запрошення однокласників або друзів протестувати гру та дати відгуки щодо ігрового процесу й інтерфейсу.

5. Ітерації.

- Удосконалення гри. Внесення змін на основі відгуків і повторне тестування.
- Додавання нових функцій. Інтеграція нових ідей, а саме додавання нових рівнів або складніших загадок.

Приклад.

У грі «Лабіринт» на Scratch ми визначили кілька простих рівнів складності. Один із рівнів включає збір ключів для відкриття дверей на шляху до фінішу. Персонаж керується за допомогою стрілок на клавіатурі, а програмний код в Scratch відповідає за перевірку наявності ключа перед дверима. Якщо ключ є, двері «відкриваються» (зникають), і персонаж може продовжити свій шлях до фінішу. Такий підхід допомагає здобувачам освіти практично застосовувати свої навички програмування та розвивати логічне мислення.

Тепер напишемо простий приклад коду в Scratch для гри «Лабіринт», де персонаж має зібрати ключі, щоб відкрити двері та досягти фінішу. Оскільки Scratch використовує візуальне програмування, опишемо блоки коду та їх взаємозв'язок.

1. Створення спрайтів.

- Спрайт «Персонаж».
- Спрайт «Ключ».
- Спрайт «Двері».
- Спрайт «Фініш».

2. Рух персонажа.

Додати блоки управління з клавіатури для переміщення персонажа:

- Коли клавіша «стрілка вліво» натиснута, переміщуйте спрайт «Персонаж» на -10 по осі x.

- Коли клавіша «стрілка вправо» натиснута, переміщуйте спрайт «Персонаж» на 10 по осі x.
- Коли клавіша «стрілка вгору» натиснута, переміщуйте спрайт «Персонаж» на 10 по осі y.
- Коли клавіша «стрілка вниз» натиснута, переміщуйте спрайт «Персонаж» на -10 по осі y.

3. Збір ключів.

Додати скрипт до спрайту «Ключ»:

- Використовуйте блок «при зіткненні з [Персонаж]», ховати спрайт «Ключ» і змінювати змінну «має ключ» на «так».
- Взаємодія з дверима.
- Додати скрипт до спрайту «Двері»:
- Використовуйте блок «якщо» для перевірки, чи змінна «має ключ» дорівнює «так».
- Коли «Персонаж» зіткнеться з «Двері» і «має ключ» дорівнює «так», ховати спрайт «Двері».

4. Перемога:

Додати скрипт до спрайту «Фініш»:

- Використовуйте блок «при зіткненні з [Персонаж]», виведіть повідомлення «Ви виграли!» та зупиніть гру.

Як базовий приклад, цей алгоритм можна розширити, додаючи додаткові елементи геймплея (часові обмеження, різні рівні складності тощо). Scratch дозволяє легко втілювати ці ідеї за допомогою інтуїтивно зрозумілого інтерфейсу та широкого набору програмних блоків.

Отже, проектування ігор в освітньому процесі стає все більш популярним підходом у сучасній освіті, особливо у викладанні програмування. Цей метод надає учням можливість не лише засвоїти технічні навички, а й розвинути креативне мислення, командну роботу й аналітичні здібності. Розробка навчальних планів, що інтегрують створення ігор в освітній процес, вимагає детального підходу [15]. Важливо створити навчальні курси, що включають вивчення основних концепцій програмування, графічного дизайну й інтерактивної анімації. Наприклад, уроки можуть охоплювати основи роботи в середовищах, як-от Scratch або Unity, де учні мають змогу в реальному часі спостерігати за результатами своєї роботи. Типові завдання та проекти, які можна використовувати для оцінювання прогресу учнів, включають створення ігор із певними вимогами. Наприклад, учитель може задати завдання створити гру на тему екології, де гравці вчаться сортувати сміття або очищувати океан від пластику. Такі проекти не тільки покращують технічні навички учнів, а й залучають їх до вирішення реальних проблем, стимулюючи критичне мислення та соціальну відповідальність. У цьому контексті, оцінювання може здійснюватися на основі кількох критеріїв: технічна складність проекту, креативність у вирішенні задачі, якість

і логічність коду, а також здатність роботи в команді та використання зворотного зв'язку для вдосконалення проєкту. Врахування цих факторів допомагає вчителям точніше оцінювати здобутки учнів і надавати їм конкретні поради для подальшого розвитку.

Висновки. Підсумком можна сказати, що у процесі впровадження методу навчання через створення ігор існують значні виклики. Зокрема, школи часто стикаються з обмеженим часом, бюджетом або доступом до необхідних технологій. Наприклад, не кожна школа може дозволити собі передове програмне забезпечення або комп'ютерну техніку, яка необхідна для викладання сучасного програмування. Також труднощі можуть виникати через недостатній час на поглиблене вивчення теми через напружений учбовий план.

Для подолання зазначених викликів можуть бути використані кілька стратегій. Наприклад, школи можуть шукати додаткове фінансування через гранти або спонсорства для придбання необхідного обладнання. Також можна використовувати вільно доступні ресурси, як-от Scratch або інші безкоштовні платформи, які дозволяють створювати ігри з мінімальними витратами. Окрім того, заклади освіти можуть організовувати майстер-класи та тренінги для вчителів, щоб підвищити їхню компетентність у сфері цифрових технологій.

Потенціал розширення програми навчання через ігри на інші предмети та діяльності є значним. Наприклад, можна адаптувати ігрові методики для викладання математики, фізики, літератури й інших дисциплін. Ігрові стратегії можуть допомогти учням краще засвоїти складні концепції через візуалізацію та інтерактивність, що робить процес навчання більш захоплюючим і ефективним.

Також технологічні інновації можуть істотно змінити підхід до навчання програмуванню. Упровадження штучного інтелекту, машинного навчання та віртуальної реальності в освітні процеси відкриває нові можливості для створення більш динамічних та інтерактивних навчальних середовищ. Ці технології дозволяють створювати симуляції та віртуальні експерименти, які можуть забезпечити глибше розуміння програмування та його застосування в реальному світі.

Список використаних джерел:

1. Бабенко В. П. Методика викладання програмування в закладах загальної середньої освіти. 2021.
2. Васько О. О. Створення інтерактивних дидактичних ігор засобами Microsoft Office Power Point. Фізико-математична освіта, 2019. Випуск 3 (21). С. 25–31.
3. Горохова Р. І., Нікітін П. В. Можливості сучасних інформаційних технологій у проведенні психолого-педагогічних досліджень. Міжнародний електронний журнал «Освітні технології та суспільство (Educational Technology & Society)», 2012. V.15. № 2. С. 390–411.
4. Думінш А. А., Зайцева Л. В. Комп'ютерні ігри у навчанні та технології їх розробки. Міжнародний електронний журнал «Освітні технології та суспільство (Educational Technology & Society)», 2013. V.15. № 3. С. 534–545.

5. Жемчужников Д. Г. Створення комп'ютерних ігор як навчання школярів програмуванню. Інформатика і освіта, 2012. № 8 (237). С. 49–51.
6. Create with Code. URL: <https://learn.unity.com/course/create-with-code> (дата звернення: 15.04.2024).
7. Комп'ютерні ігри: користь і шкода для дитини. URL: <https://t1p.de/0v0tl> (дата звернення: 15.04.2024).
8. Офіційний сайт мови програмування Python. URL: <https://www.python.org/> (дата звернення: 15.04.2024).
9. Олійник Ю. П. Ігрофікація в освіті: до питання про визначення поняття. Сучасні проблеми науки і освіти, 2015. № 3. С. 476.
10. Онлайн-копії планів уроків. URL: <https://www.stencyl.com/teach/> (дата звернення: 15.04.2024).
11. Освітня платформа та конструктор безкоштовних відкритих онлайн-курсів Stepik. Опис використання класів. URL: <https://welcome.stepik.org/ru/class> (дата звернення: 15.04.2024).
12. Посібник для вчителя WoofJS. URL: <https://t1p.de/8rkz6> (дата звернення: 15.04.2024).
13. Пушкарьова Т. О., Рибалко О. О. Засоби створення електронних освітніх ресурсів для початкової школи. Фізико-математична освіта, 2017. Випуск 4 (14). С. 271–275.
14. Хворостіна Ю. В., Удовиченко О. М., Юрченко А. О. Особливості використання дидактичних ігор на уроках математики. Інноваційна педагогіка, 2019. Випуск 19. Том 3. С. 141–146.
15. Чурок С., Шамо́ня В. Використання комп'ютерних ігор в навчанні інформатики учнів основної школи. Освіта. Інноватика. Практика, 2022, 10 (1), С. 60–70. URL: <https://doi.org/10.31110/2616-650X-vol10i1-007> (дата звернення: 15.04.2024).
16. Шерман О. Комп'ютерні ігри як засіб впровадження політичних стереотипів. Укр. нац. ідея: реалії та перспективи розвитку, 2008. Вип. 20. С. 150–154.

KVAK P., CHERNIKOVA L. METHODS OF TEACHING PROGRAMMING TO STUDENTS THROUGH CREATING A COMPUTER GAME

The article analyzes methods of teaching programming to school students through the creation of computer games, highlighting both theoretical and practical aspects of this process. It examines the main tools and programming languages used in the game development process, including Scratch, Unity, and Python with Pygame. The authors provide a step-by-step algorithm for developing a simple computer game, from planning to iteration and testing. The challenges and strategies for overcoming difficulties such as limited time, budget, or access to technology are described. Future prospects for the application of this method in educational initiatives are analyzed, with a focus on technological innovations that could change approaches to teaching programming.

Key words: school programming education, game creation, Scratch, Unity, Python, Pygame, game pedagogy, step-by-step game development algorithm, educational technology, innovations in education.

Надійшла до редакції 28.04.2024 р.