ТЕОРЕТИЧНІ ТА МЕТОДОЛОГІЧНІ ЗАСАДИ ПРОФЕСІЙНОЇ ОСВІТИ

УДК 378.147:004
DOI: 10.32342/3041-2196-2024-2-28-11

Г.І. АЛЄКА,
доктор філософії,
старший викладач кафедри інформатики та прикладної математики, Криворізький державний педагогічний університет (м. Кривий Ріг)

МЕТОДИЧНІ ПІДХОДИ ДО НАВЧАННЯ ПРОГРАМУВАННЯ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ У СЕРЕДОВИЩІ SCRATCH

Метою дослідження є пошук, обґрунтування і висвітлення сутності методичних підходів до навчання програмування в середовищі Scratch.

У дослідженні використовуються методи аналізу і синтезу, систематизації та конкретизації наукових даних про методи і стратегії навчання алгоритмізації та програмування майбутніх фахівців, а також методи розробки методологічних підходів до організації роботи в середовищі Scratch.

У статті виокремлено п'ять методичних підходів до навчання програмування в середовищі Scratch, зокрема кодування за зразком, аналіз коду, пошук помилок у коді, доповнення коду, створення коду проєкту за вимогами. Розглянуто особливості впровадження у освітній процес кожного із запропонованих підходів у підготовці майбутніх вчителів інформатики.

Зазначено, що навчання програмування має розпочинатися із засвоєння прийому «кодування за зразком». Цей підхід доречно застосовувати на лекційних заняттях. Наведено приклади проєктів, які доцільно використовувати на перших заняттях, які присвячені ознайомленню з програмним середовищем Scratch. Викладено основні переваги «кодування за зразком».

З'ясовано, що прийом «аналіз коду» подібний до читання перед тим, як навчитися писати. Цей підхід дозволяє студентам ознайомитися зі способом поєднання компонентів і конструкцій мови. Представлення коду може бути різним: на друкованих картках, на онлайн-дошці для спільної роботи, у середовищі Scratch тощо.

Визначено, що наступним підходом, не менш важливим у навчанні програмування, є «пошук помилок у коді». Його доцільно використовувати на практичних заняттях для роботи у групах. Запропоновано реалізацію групової роботи під час онлайн-навчання засобами платформи Zoom. Описано приклади проєктів, які використовуються для реалізації підходу «пошук помилок у коді» із зазначенням мети гри, багів та представленням можливого дизайну гри.

Презентований у статті четвертий методичний підхід полягає у опануванні навичок програмування шляхом виконання завдань, які передбачають доповнення коду проєкту. Цей метод доцільно використовувати у тих випадках, коли складність проєкту висока, і неможливо відтворити код проєкту в межах академічної години. Задля реалізації методичного підходу «доповнення коду» викладач має надавати студентам посилання на проєкт, з якого вони потім будуть створювати «Ремікс».

П'ятим методичним підходом є «створення коду проєкту за вимогами». Авторка пропонує його реалізацію у вигляді покрокової інструкції створення проєкту. Завдання із покроковим алгоритмом створення інтерактивного проєкту, анімації чи гри можуть бути адаптовані до рівня підготовки кожного студента, що забезпечує оптимальний темп засвоєння матеріалу.

Зроблено висновок про те, що кожен із запропонованих підходів є невід'ємною складовою змісту дисциплін з алгоритмізації та програмування. Виконуючи завдання в межах кожного із підходів, студенти набувають професійно важливого досвіду.

Ключові слова: *методика викладання інформатики, середовище Scratch, візуальне середовище програмування, ігрові методи навчання, методи навчання програмування, майбутні вчителі інформатики.*

Постановка проблеми. Нині відбувається стрімкий розвиток процесу інформатизації й цифрової трансформації суспільства. Інформаційні технології стали невід'ємною частиною життя людини. Тому підготовка студентів в закладах вищої освіти має бути орієнтована на розвиток у них здатності сприймати нові знання й умінні застосовувати набуті компетентності не тільки під час навчання у ЗВО, а й у майбутній професійній діяльності. Динамічні зміни в суспільстві спонукають до професійного самовизначення, здатності адаптуватися до нових умов, визначати мету діяльності й можливі шляхи її досягнення. Це особливо актуально для студентів педагогічних університетів, оскільки система освіти постійно реформується для учнів, батьків і вчителів [Вдовичин, Лазурчак, 2019, с. 54; M. Moiseienko, N. Moiseienko, Lavrentieva, 2023].

Важливою складовою процесу навчання майбутніх учителів інформатики є засвоєння студентами фундаментальних понять і формування в них практичних навичок роботи з програмними засобами. Ключовим завданням є формування у студентів методологічного стилю мисленню, надання знань із предметної галузі й розвиток умінь застосовувати ці знання на практиці із застосуванням сучасних ІКТ [Вдовичин, Лазурчак, 2019, с. 55].

Підготовка фахівців першого (бакалаврського) рівня вищої освіти спеціальності 013 Початкова освіта з додатковою спеціалізацією «Програмування» передбачає вивчення нормативних та варіативних дисциплін. Що стосується нормативних дисциплін, то саме на другому році навчання викладається дисципліна «Основи алгоритмізації та програмування». У вивченні цієї дисципліни у студентів повинна сформуватись база знань, умінь і навичок, що їх будуть супроводжувати в подальшій професійній діяльності.

У результаті вивчення навчальної дисципліни майбутньому вчителю інформатики початкових класів пропонується ознайомитися з середовищем програмування Scratch – ефективним інструментом для навчання основам алгоритмізації та програмування у молодших школярів та учнів 5-6 класів. Він може бути використаний не лише на уроках інформатики в школі, а й у процесі підготовки вчителів початкових класів до їх професійної діяльності. Тож визначення ефективних методичних підходів до навчання програмування в середовищі Scratch виявляється доцільним та актуальним.

Аналіз останніх досліджень. Питання застосування середовища Scratch у навчанні програмування студентів закладів вищої освіти є об'єктом уваги науковців. Наприклад, О. Яценко розглядає переваги середовища програмування Scratch та його особливості [Яценко, 2017, с. 278]. До того ж науковиця у дослідженні здійснює аналіз найбільш популярних навчальних середовищ блокового візуального програмування. За результатами порівняння, здійсненого О. Яценко, Scratch визначено найбільш зручним та адаптованим для навчання майбутніх учителів початкових класів [Яценко, 2023]. О. Дущенко приділяє увагу аналізу можливостей середовища Scratch і пропонує авторські завдання для вивчення Scratch майбутніми вчителями інформатики [Друщенко, 2019, с. 36]. О. Дудка, О. Власій та Г. Магомета дослідили можливості використання середовища Scratch для формування ключових міжпредметних і предметних компетентностей учнів; запропонували шляхи реалізації компетентнісного підходу на прикладі створення навчальних Scratch-проєктів ігрового характеру [Дудка, Власій, Магомета, 2018, с. 88]. М. Мар'єнко, І. Борисюк розглядають Scratch як один із можливих веб-ресурсів для реалізації гейміфікації в освітньому процесі [Мар'єнко, Борисюк, 2023, с. 74]. Проте, незважаючи на досить значну кількість досліджень, що стосуються використання середовища програмування Scratch у підготовці майбутніх учителів, відсутні праці, присвячені методичним підходам до навчання програмування, що й зумовило необхідність розробки методичних підходів до навчання програмування майбутніх учителів інформатики у середовищі Scratch.

Формулювання цілей статті. Метою дослідження є пошук, обґрунтування і висвітлення сутності методичних підходів до навчання програмування в середовищі Scratch.

У дослідженні використовуються **методи** аналізу і синтезу, систематизації та конкретизації наукових даних про методи і стратегії навчання алгоритмізації та програмування майбутніх фахівців, а також методи розробки методологічних підходів до організації роботи в середовищі Scratch.

Виклад основного матеріалу. Scratch – мультиплатформне візуальне середовище програмування з відкритим вихідним кодом. Середовище було створено Массачусетським тех-

нологічним інститутом у 2007 році для навчання школярів основам програмування. Scratch перекладено на 70 мов світу, у т. ч. і українською [Яценко, 2023]. На рис. 1. наведено переваги середовища програмування Scratch.

За результатами вивчення першоджерел виокремлюємо такі методичні підходи до навчання програмування у середовищі Scratch, як-от: кодування за зразком; аналіз коду; пошук помилок у коді; доповнення коду; створення коду проєкту за вимогами. Розглянемо особливості впровадження кожного із перелічених підходів у освітньому процесі при підготовці майбутніх учителів інформатики початкової школі.

Рис. 1. Переваги середовища Scratch для навчання алгоритмізації та програмування

Навчання програмування майбутніх учителів має розпочинатися з написання коду за зразком. Цей підхід доречно застосовувати на лекційних заняттях. У цьому випадку студенти стають не пасивними споживачами інформації, а активними учасниками освітнього процесу: викладач поступово пише код програми, пояснюючи функціональне призначення кожного із блоків, які використовуються для реалізації проєкту, а студенти повторюють усі дії на своїх персональних комп'ютерах. Якщо при реалізації проєкту ті чи ті блоки повторюються, то викладач може ставити до аудиторії запитання на усвідомлення та розуміння структури коду. Наприкінці заняття студенти разом із викладачем тестують отримані результати та виправляють помилки за їх наявності [Selby, 2011].

Розглянемо приклади проєктів, які доцільно використовувати в підготовці майбутніх учителів інформатики початкової школи. На нашу думку, одним із перших проєктів, який студенти запрограмують за зразком одночасно з викладачем, є коротка історія, яка передбачає діалог між спрайтами й рух кожного зі спрайтів за допомогою стрілок на клавіатурі. Програмування руху спрайта є базовою навичкою, яка потрібна для програмування майже будь-якої гри. Не менш важливою навичкою є програмування діалогу між спрайтами, оскільки команда «Говорити «Привіт!» 2 сек.» використовується на початку більшості проєктів для повідомлення гравцеві про початок гри. Також цей блок доцільно використовувати для пояснення правил гри, висвітлення сюжету тощо. Інший проєкт – це гра «Ping pong», завдяки якій студенти ознайомлюються з програмуванням переміщення спрайта за допомогою мишки, умовним оператором, командою генерації випадкового числа й однією з команд розділу «Датчики».

Метод написання коду за зразком підходить для початкового етапу навчання програмування, оскільки дозволяє навчитися на конкретних прикладах і поступово опанувати базові поняття мови програмування, розвиваючи впевненість у своїх силах і навички самостійного розв'язання практичних завдань. Головними перевагами прийому «кодування за зразком» є:

– зниження рівня складності, адже метод дозволяє уникати перевантаження інформацією, оскільки викладач фокусується під час заняття на конкретному проєкті;

– підтримка мотивації аудиторії, оскільки такий підхід надає можливість швидко отримати результат у вигляді робочих ігрових проєктів, що підвищує впевненість студентів, зокрема тих, які не мають попереднього досвіду програмування;

– закріплення теоретичних знань на практиці: повторення та відтворення проєктів сприяє формуванню базових навичок програмування, які стають основою для написання коду самостійно.

Інший методичний підхід у навчанні програмування – це аналіз коду, що подібний за своєю структурою до навчання читанню перед тим, як навчитися писати. Аналіз коду дозволяє студентам ознайомитися зі способом поєднання компонентів і конструкцій мови програмування. Код може бути представлений у різних формах – на друкованих картках, на онлайн-дошці для спільної роботи, середовищі Scratch тощо [Selby, 2011].

Головна мета підходу «аналіз коду» – це читання готового коду й розуміння його логіки, перш ніж переходити до написання власного. Особливо цей підхід доречний для більш слабких студентів. Здатність пояснити логіку програмування й код є необхідною умовою задля оволодіння навичками написання коду, хоча це не означає здатності писати код [Selby, 2011].

Наступним підходом, не менш важливим у навчанні програмування, є «пошук помилок у коді». Його доцільно використовувати на практичних заняттях для роботи у групах. Розглянемо особливості реалізації групової роботи під час онлайн-навчання. Усіх студентів слід об'єднати у групи по 2-3 людини. Перед об'єднанням у групи пропонується провести тестування, що передбачає оцінювання знань, отриманих на попередньому занятті. Кожен студент після проходження тестування має поставити «+» у чат Zoom. Наприклад, перші два студенти, що поставили «+» будуть утворювати першу групу, два наступних – другу групу тощо. Після формування груп та розподілу їх по вебірнарних кімнатах у Zoom, викладач відправляє посилання на проєкти, що містять баги. Задача студентів – знайти всі баги та виправити їх. Рівень складності багів залежить від кількості матеріалу, який встигли опанувати студенти.

Розглянемо окремі приклади проєктів, які використовують для реалізації зазначеного підходу. Перший проєкт передбачає керування феєю за допомогою стрілок на клавіатурі. При цьому слід уникати всіх хмар, оскільки при дотику до них спрайт переміщується на початок. Мета гри: дістатися до кристала. Як тільки гравець дістанеться кристала, фея скаже «Ура, перемога» і гра почнеться спочатку. Баги, що трапляються у завданні, – спрайт відслідковує дотик лише до однієї хмари; не реалізована перевірка на дотик до кристалу – не запрограмовано перемогу (рис. 2).

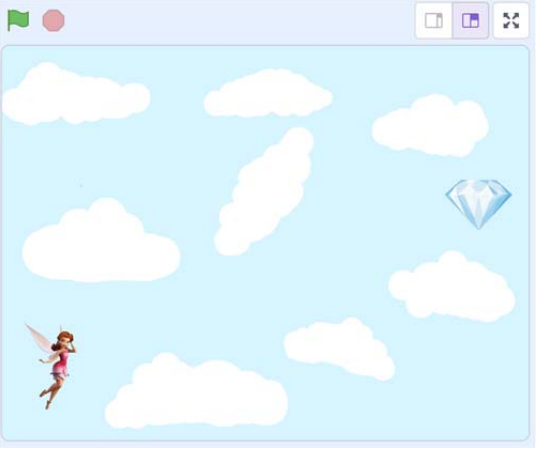

Рис. 2. Візуалізація проєкту «Керування феєю»

Мета другого проєкту: перестрибувати всі знаки питання протягом однієї хвилини задля отримання головного призу – магічного кристалу. Для керування спрайту використовують стрілки ліворуч і праворуч і пробіл. За умови дотику гравця до знаку питання гра зупиняється. Якщо гравець перемагає, то кристал миготить різними кольорами. Баги, що трапляються у завданні, – при дотикові до знаків питання гра не зупиняється; стрибок не працює (замість стрибка спрайт просто переміщується вгору); кристал не миготить різними кольорами (Рис. 3).

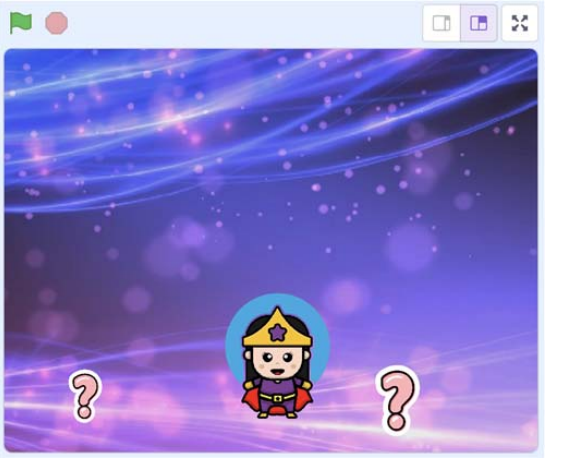

Рис. 3. Візуалізація завдання «Перестрибування знаків питання»

Третій проєкт для пошуку багів – це «Космічний шутер», у якому гравець керує космічним кораблем за допомогою стрілок. Мета гри: уникати астероїдів, які рухаються в напрямку космічного корабля. Якщо астероїд торкається корабля, гравець програє. Час виживання вимірюється, а результат відображається наприкінці гри. Баги, що трапляються у завданні, – астероїди рухаються надто швидко, що робить гру неможливою для проходження; при дотику астероїдів до космічного корабля гра не зупиняється; астероїди при дотику до протилежної межі сцени не зникають (рис. 4).

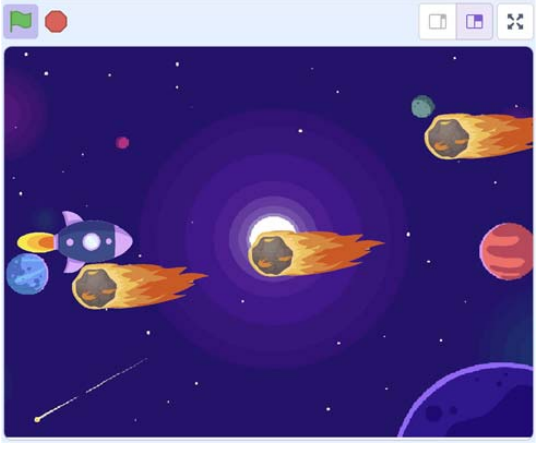

Рис. 4. Візуалізація завдання «Космічний шутер»

«Морські пригоди під водою» – четвертий проєкт для пошуку багів. Гравець керує морським коником двома клавішами: вгору та вниз. Завдання – зібрати всі золоті монети протягом 30 секунд, уникаючи акул. Якщо морський коник торкається акули, гра зупиняється і гравець програє. Після того, як морський коник збере всі скарби за 30 секунд, з'явиться повідомлення «Перемога!». Баги, що трапляються в завданні, – акули не рухаються, що робить гру занадто легкою; після 30 секунд гри не з'являється повідомлення «Перемога!» (рис. 5).

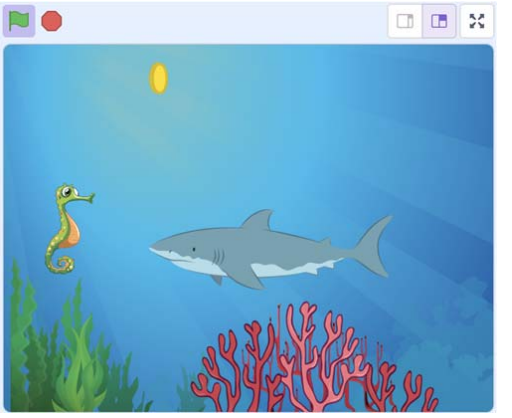

Рис. 5. Гра «Морський коник під водою»

У п’ятому проєкті головний герой – чарівник, який має зібрати всі пляшки із зіллям. З кожною зібраною пляшкою рахунок збільшується на 1. Потім пляшка зникає. Баги, що трапляються у завданні, – зілля не зникає; рахунок не збільшується; після перезапуску гри рахунок не обнуляється (рис. 6).

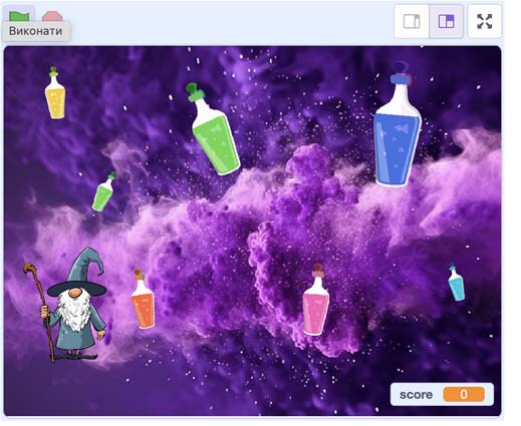

Рис. 6. Гра «Чарівник»

У наступному проєкті у чарівниці є 5 життів. При дотику до зірочки втрачається одне життя. Як тільки всі життя закінчуються, гра зупиняється. Баги, що трапляються у завданні, – життя не зменшується при дотику зірки до чарівниці; при дотику чарівниці до зірочки значення змінної змінюється кілька разів (Рис. 7).

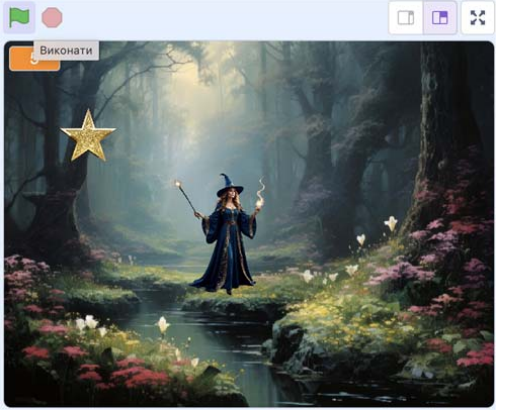

Рис. 7. Гра «Чарівниця»

Сьомий проєкт – скролінг-анімація, яка починається при натисненні на кнопку «Старт», а зупиняється – при натисненні на кнопку «Стоп». Баги, що трапляються у завданні, – анімація не працює, оскільки клони створюються, але не використовуються; анімація запускається при натисненні на зелений прапорець, а не на кнопку «Старт»; кнопка «Стоп» не працює (Рис. 8).

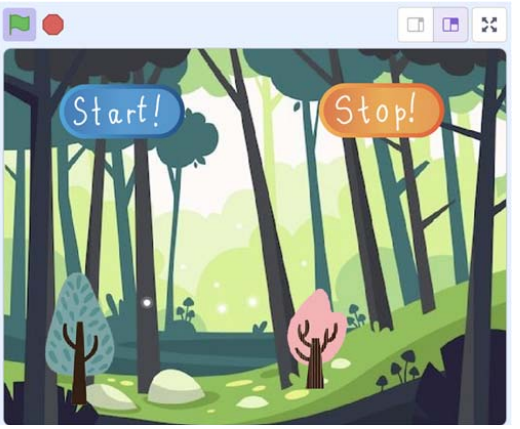

Рис. 8. Візуалізація скролінг-анімації

Отже, можна пересвідчитися в тому, що «пошук помилок у коді» є ефективним методичним підходом у навчанні програмування в середовищі Scratch, оскільки:

– помилки в програмі легко зрозуміти завдяки тому, що середовище програмування є візуальним;

– пошук помилок студенти реалізують через аналіз виконання команд, що допомагає зрозуміти логіку алгоритму і побудувати правильний порядок блоків;

– існує миттєвий зворотній зв'язок завдяки тому, що програму можна перевірити одразу, зокрема, якщо спрайти взаємодіють неправильно, то студенти можуть швидко ідентифікувати проблему та виправити її;

– процес пошуку помилок не має серйозних наслідків, що дозволяє студентам вільно експериментувати без остраху зіпсувати програмний код.

Четвертий методичний підхід полягає в опануванні навичок програмування шляхом виконання завдань, які передбачають доповнення коду проєкту. Цей метод доцільно використовувати у тих випадках, коли складність проєкту висока, і неможливо відтворити код проєкту в межах академічної години. Запропонований підхід доцільно застосовувати також із метою заощадження часу на виконанні таких рутинних завдань як завантаження спрайтів, тла та музичних треків до шаблону проєкту [Деревянчук, 2023].

Для реалізації методичного підходу «доповнення коду» викладач має надавати студентам посилання на проєкт, з якого вони потім будуть створювати «Ремікс». Для створення реміксу студенти мають бути авторизовані на платформі Scratch (https://scratch.mit.edu/). Варто зазначити, що створювати проєкти в Scratch можна і в десктопній версії, але без прив'язки до акаунту, що робить неможливим використання запропонованого вище функціоналу. До того ж, десктопна версія має ряд інших недоліків, які варто враховувати при організації навчального процесу. Серед них: відсутність автозбереження, незручний формат звітування за умови онлайн навчання [Алєка, 2024, с. 83].

Розглянемо три приклади проєктів для реалізації четвертого підходу. Перший з них – це створення меню до готової гри. Це може бути проєкт, який здобувачі освіти попередньо створили на занятті, наприклад гра «Лабіринт» [Алєка, 2024, с. 83]. При його реалізації студенти опановують використання таких команд, як «оповістити повідомлення» та «коли я отримую повідомлення». Другий – завершення скролінгової гри (scrollers), а саме реалізації рухомого фону (scrolling background). Скролінг – це ігровий термін, що означає переміщення вмісту вікна. Використання скролінгового фону дозволяє створити ілюзію руху в гоночному симуляторі [Нікітін, Нікітіна, 2024, с. 45]. Третім проєктом може бути гра «Платфор-

мер». У проєкті-шаблоні слід підготувати 5-7 рівнів, що міститимуть перепони, при дотику до яких гравець втрачатиме одне життя і переміщатиметься на початок рівня. До того ж на одному з рівнів стане в нагоді створення перешкоди типу «батут».

Нижче пропонуємо приклад практичної роботи, яку доцільно використовувати для реалізації останнього методичного підходу – «створення коду проєкту за вимогами».

Завдання 1

1. Підберіть тло з бібліотеки Scratch або завантажте його з мережі інтернет.

2. Додайте 1 спрайт персонажа та 7-10 спрайтів-аксесуарів із вкладки «Мода».

3. Для кожного аксесуара завантажте з мережі інтернет короткий звук тривалістю від 5 до 8 секунд.

Завдання 2

1. Змініть розміри спрайтів-аксесуарів, якщо вони завеликі або замалі для персонажа.

2. Напишіть код, щоб персонаж з'являвся у центрі сцени на початку гри, а решта спрайтів зверху і знизу нього, при цьому не торкаючись персонажа. Для цього використовуйте команду «перемістити в x… y…», що міститься в групі «Рух».

Завдання 3

1. Оберіть перший спрайт-аксесуар.

2. Продовжуйте писати скрипт, додавши команду, яка дозволяє перетягувати спрайт, коли гра запущена.

3. Допишіть код, щоб відтворювався раніше доданий звук, коли цей аксесуар дотикається до персонажа. Для відтворення звука при дотику аксесуара до персонажа використовуйте команду «Відтворити звук … до кінця».

4. Скопіюйте написаний код на інші спрайти-аксесуари. Оберіть «увімкнення правильного звуку» на кожному спрайті.

Завдання 4

1. Оберіть спрайт-персонаж.

2. Напишіть скрипт, який буде реалізовувати анімацію персонажа, коли на нього наведено вказівник миші. Для цього використайте умовний оператор неповної форми та команду, яка реалізує зміну образу на наступний.

3. Оберіть тло. Запрограмуйте постійну плавну зміну ефекту «Колір», коли гра почалася.

Завдання 5

1. Оберіть один із спрайтів-аксесуарів або додайте новий.

2. Використовуючи розширення «Музика», запрограмуйте мелодію та її включення, коли аксесуар перебуває на персонажі.

3. Підберіть інструмент та ноти для створення цікавої музики. Якщо написана музика звучить надто голосно, використовуй команду «Встановити гучність __ %» у вкладці «Звук».

Виконання практичних робіт, що передбачають розробку інтерактивних проєктів, анімацій та ігор, формують у студентів вміння вирішувати прикладні завдання, що відповідають сучасним потребам освіти. Завдання можуть бути адаптовані до рівня підготовки кожного студента, що забезпечить оптимальний темп засвоєння матеріалу.

Висновки. Середовище візуального програмування Scratch є універсальним інструментом створення навчальних проєктів різного рівня складності. За результатами дослідження запропоновано п'ять методичних підходів до навчання програмування майбутніх вчителів інформатики: кодування за зразком; аналіз коду; пошук помилок у коді; доповнення коду; створення коду проєкту за вимогами. Кожен із запропонованих підходів є невід'ємною складовою змісту дисципліни, що передбачає вивчення основ алгоритмізації та програмування. Виконуючи завдання в межах кожного із підходів, студенти набувають досвіду, який зможуть використовувати у власній професійній діяльності, зокрема під час викладання інформатики в початковій та основній школі.

Список використаних джерел

Алєка, Г. І. (2024). Особливості використання середовища Scratch при підготовці майбутніх вчителів інформатики початкової школи. *Наукові записки Серія: Педагогічні науки*, 216, 82–87. doi: 10.36550/2415-7988-2024-1-216-82-87

Вдовичин, Т. Я., Лазурчак, Л. В. (2019). Особливості вивчення програмування майбутніми вчителями інформатики. *Information Technologies in Education,* 2 (39), 54–66. doi: 10.14308/ite000696

Друщенко, О. С. (2019). Необхідність вивчення середовища програмування Scratch майбутніми вчителями інформатики. *Інноваційна педагогіка*, 18 (3), 35–40. doi: 10.32843/2663-6085-2019-18-3-7

Деревянчук, О. В. (2023). Розробка моделі нечіткої когнітивної карти для створення STEM-проєктів у професійній підготовці майбутніх фахівців інженерно-педагогічних спеціальностей. *Вісник Університету імені Альфреда Нобеля. Серія «Педагогіка і психологія». Педагогічні науки*, 2 (26), С. 160–169. doi: 10.32342/2522-4115-2023-2-26-16

Дудка, О. М., Власій, О. О., Магомета, Н. М. (2018). Реалізація компетентнісного підходу до вивчення програмування на Scratch. *Відкрите освітнє е-середовище сучасного університету,* 5, 88–96. doi: 10.28925/2414-0325.2018.5.8896

Мар'єнко, М. В., Борисюк, І. Ю. (2020). Гейміфікація освітнього процесу під час вивчення дисциплін природничо-математичного циклу учнями ЗЗСО. *Фізико-математична освіта*, 4 (26), 72–78. doi: 10.31110/2413-1571-2020-026-4-013

Нікітін, С. О., Нікітіна, Л. О. (2018). *Основи комп'ютерних ігор та ігрових програм: довідник модуля*. Харків: «Друкарня Мадрид».

Яценко, О. І. (2017). Середовище програмування «Scratch»: аналіз можливостей використання з метою формування інформативних компетентностей вчителя початкової школи. *Актуальні питання сучасної інформатики*, 5, 276–278. Відновлено з http://eprints.zu.edu.ua/25788

Яценко, О. І. (2023). Огляд середовищ програмування з метою їх добору для формування інформаційно-цифрової компетентності майбутніх вчителів початкових класів. *Академічні візії*, 22. Відновлено з https://academy-vision.org/index.php/av/article/view/525/478

Moiseienko, M. V., Moiseienko, N. V., Lavrentieva, O. O. (2023). Developing pre-service teachers' digital competence through informatics disciplines in teacher education programs. In *Proceedings of the 6th International Workshop on Augmented Reality in Education (AREdu 2023)* (pp. 45-52). Kryvyi Rih. Retrieved from https://ceur-ws.org/Vol-3844/paper11.pdf

Selby, C. (2011). Four approaches to teaching programming. In *Learning, Media and Technology: a doctoral research conference.* 04 July 2011, London. Retrieved from https://eprints.soton.ac.uk/346935/1/LMT%25202011.pdf

References

Alieka, H. I. Peculiarities of using scratch environment in the process of training future primary school IT teachers. *Academic Notes. Series: Pedagogical Sciences*, 2024, no. 216, pp. 82–87. doi: 10.36550/2415-7988-2024-1-216-82-87 (In Ukrainian).

Derevyanchuk, O. Development of a fuzzy cognitive map model for creating STEM projects in professional training of future specialists in engineering and pedagogical specialties. *Alfred Nobel University Journal of Pedagogy and Psychology*, 2023, no. 2 (26), pp. 160–169. doi: 10.32342/2522-4115-2023-2-26-16 (In Ukrainian).

Drushchenko, O. S. The need to study the Scratch programming environment by future computer science teachers. *Innovative Pedagogy*, 2019, issue 18, part 3, pp. 35–40. doi: 10.32843/2663-6085-2019-18-3-7 (In Ukrainian).

Dudka, O. M., Vlasii, O. O., Mahometa, N. M. Implementation of the competence-based approach to learning programming on Scratch. *Open Educational E-environment of Modern University*, 2018, no. 5, pp. 88-96. doi: 10.28925/2414-0325.2018.5.8896 (In Ukrainian).

Marienko M., Borysiuk I. Gamification of the educational process during the study of disciplines of the natural and mathematical cycle by pupils of IGSE. *Physical and Mathematical Education*, 2020, issue 4(26), pp. 72–78. doi: 10.31110/2413-1571-2020-026-4-013 (In Ukrainian).

Moiseienko, M. V., Moiseienko, N. V., Lavrentieva, O. O. Developing pre-service teachers' digital competence through informatics disciplines in teacher education programs. *Proceedings of the 6th International Workshop on Augmented Reality in Education (AREdu 2023)*. Kryvyi Rih, 2023, pp. 45–52. Available at: https://ceur-ws.org/Vol-3844/paper11.pdf (Accessed 20 September 2024).

Nikitin, S.O., Nikitina, L.O. (2018). *Osnovy kompiuternykh ihor ta ihrovykh prohram: dovidnyk modulia* [Fundamentals of computer games and game programs: Module reference-guide]. Kharkiv, «Drukarnia Madryd» Publ., 138 p. (In Ukrainian).

Selby, C. Four approaches to teaching programming. *Learning, Media and Technology: a doctoral research conference.* London, 2011. Available at: https://eprints.soton.ac.uk/346935/1/LMT%25202011.pdf (Accessed 20 September 2024).

Vdovychyn, T., Lazurchak, L. (2019). *Osoblyvosti vyvchennia prohramuvannia maibutnimy vchyteliamy informatyky.* [Features of studying programs by future teachers of informatics]. *Information Technologies in Education,* 2019, no. (39), pp. 54-66. doi: 10.14308/ite000696 (In Ukrainian).

Yatsenko, O. I. An overview of the programming environment for the purpose of their selection for the future primary school teachers' information and digital competence formation. *Academic visions*, 2023, no. 22. Available at: https://academy-vision.org/index.php/av/article/view/525/478 (Accessed 20 September 2024). (In Ukrainian).

Yatsenko, O. I. *Seredovyshche prohramuvannia «Scratch»: analiz mozhlyvostei vykorystannia z metoiu formuvannia informatyvnykh kompetentnostei vchytelia pochatkovoi shkoly* [Scratch programming environment: analysis of its possible appliance to form primary school teachers' informative competences]. *Aktualni pytannia suchasnoi informatyky* [Topical Issues of Modern Computer Science], 2017, no. 5, pp. 276–278. Available at: http://eprints.zu.edu.ua/25788 (Accessed 20 September 2024). (In Ukrainian).

METHODOLOGICAL APPROACHES TO TRAINING FUTURE COMPUTER SCIENCE TEACHERS IN PROGRAMMING WITH SCRATCH ENVIRONMENT

Alieka Halyna, Doctor of Philosophy in Educational and Pedagogical Sciences, Senior Lecturer of the Department of Computer Science and Applied Mathematics, Kryvyi Rih State Pedagogical University, Kryvyi Rih.

e-mail: galina.ivanova.2308@gmail.com.

ORCID: 0000-0001-6432-2154

DOI: 10.32342/3041-2196-2024-2-28-11

Keywords: *methodology of teaching computer science, Scratch environment, visual programming environment, game-based learning methods, methods of teaching programming, future computer science teachers.*

The article addresses the issue of updating the content of training future computer science teachers through the use of visual environments for teaching programming and algorithmisation.

The ***purpose of the study*** *is to explore, substantiate, and elucidate the essence of methodological approaches to teaching programming in the Scratch environment. The research* ***objectives*** *include identifying the features and advantages of Scratch in teaching programming to future computer science teachers and applying the acquired professional knowledge in teaching computer science at primary and secondary school levels. Additionally, the study seeks to outline approaches to teaching programming in the Scratch environment and to develop educational tasks for implementing each approach at various stages of teaching programming to future teachers.*

The study employs ***methods*** *of analysis and synthesis, systematisation, and specification of scientific data on techniques and strategies for teaching algorithmisation and programming to prospective specialists, as well as methods for designing methodological approaches to organising work in the Scratch environment.*

Based on the analysis of primary sources, five methodological approaches to teaching programming in the Scratch environment have been identified: coding by example, code analysis, debugging code, code augmentation, and creating project code based on requirements. The article examines the specifics of implementing each of these approaches in the educational process for training future computer science teachers.

It has been noted that programming instruction should begin with mastering the "coding by example" technique. This approach is appropriate for use in lecture sessions. Examples of projects suitable for initial sessions dedicated to familiarising students with the Scratch programming environment are provided, along with the primary advantages of the "coding by example" approach.

The study reveals that the "code analysis" technique is akin to reading before learning to write. This approach enables students to understand how components and constructs of the language are combined. Code presentation formats may vary, including printed cards, online collaborative whiteboards, or directly within the Scratch environment.

The next crucial approach to programming instruction is "debugging code". This technique is best suited for group work during practical classes. A proposal is made for implementing group work in an online learning environment using the Zoom platform. Examples of projects utilised for the "debugging code" approach are described, detailing the objectives of the game, the bugs to be identified, and possible game design solutions.

The fourth methodological approach presented in the article involves acquiring programming skills by completing tasks that require project code augmentation. This method is recommended for cases where the project complexity is high, making it impossible to recreate the project code within a single academic session. To implement the "code augmentation" approach, instructors should provide students with links to projects from which they will create "Remixes".

The fifth methodological approach is "creating project code based on requirements". The author proposes implementing this approach through step-by-step instructions for project creation. It has been substantiated that tasks with a step-by-step algorithm for creating interactive projects, animations, or games can be adapted to the preparedness level of each student, ensuring an optimal learning pace.

*It has been **concluded** that each of the proposed approaches is an integral component of the content of algorithmisation and programming courses. By completing tasks within the framework of each approach, students acquire professionally valuable experience.*

Одержано 09.09.2024.