

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Фізико-математичний факультет
Кафедра інформатики та прикладної математики

«Допущено до захисту»

Завідувач кафедри

_____ Моїсеєнко Н.В.

Реєстраційний № _____

«___» _____ 2023 р.

«___» _____ 2023 р.

**РОЗРОБКА ФАКУЛЬТАТИВНОГО КУРСУ З
ПРОГРАМУВАННЯ МОВОЮ SCRATCH ДЛЯ
ПРОФІЛЬНОГО НАВЧАННЯ ІНФОРМАТИКИ**

Кваліфікаційна робота
студентки групи Ім-23
ступінь вищої освіти «магістр»
спеціальності
014.09 Середня освіта (Інформатика)
Рижаніної Каріни Олексіївни

Керівник: кандидат фізико-математичних
наук, доцент
Тарасова Олена Юріївна

Оцінка:

Національна шкала _____

Шкала ECTS ___ Кількість балів ___

Голова ЕК _____

Члени ЕК _____

ЗАПЕВНЕННЯ

Я, Рижаніна Каріна Олексіївна, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавала і не одержувала недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело. Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомена. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

ЗМІСТ

ВСТУП	4
1. SCRATCH У ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ	7
1.1. Аналіз предметної області.....	7
1.2. Використання Scratch у системах освіти різних країн світу.....	14
1.3. Аналіз варіантів можливого застосування Scratch у навчальних програмах старшої профільної школи	26
Висновки до розділу 1	32
2. МЕТОДИЧНА РОЗРОБКА ФАКУЛЬТАТИВНОГО КУРСУ з ПРОГРАМУВАННЯ МОВОЮ SCRATCH ДЛЯ ПРОФІЛЬНОГО НАВЧАННЯ ІНФОРМАТИКИ УЧНІВ ЛІЦЕЇВ	34
2.1. Методичні особливості вивчення Scratch у ліцеї.....	34
2.2. Опис методичного комплексу для підтримки факультативного курсу з Scratch для профільного навчання.....	41
2.3. Аналіз результатів педагогічного експерименту	53
2.4. Рекомендації щодо використання навчально методичного комплексу в навчальному процесі.....	67
Висновки до розділу 2	68
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТКИ.....	77
Додаток А.....	77
Додаток Б	79
Додаток В.....	80

ВСТУП

Актуальність теми. Розробка навчально-методичного комплексу для організації та проведення факультативу з програмування мовою Scratch для профільного навчання інформатики має важливу актуальність у зв'язку з наступними факторами:

Розвиток цифрової грамотності. У сучасному світі важливо мати не лише базові знання інформатики, але й навички програмування. Scratch, завдяки своєму візуальному інтерфейсу, надає можливість вивчати основи програмування навіть без попереднього досвіду.

Підготовка до STEM-освіти. Scratch може слугувати відмінною основою для подальшої підготовки учнів у галузях STEM (наука, техніка, інженерія, математика). Він допомагає формувати навички, які можуть стати основою для подальшого вивчення більш складних програмних мов.

Зацікавлення учнів. Scratch має ігровий характер та зацікавлює учнів, що полегшує процес навчання. Вивчення програмування через створення анімацій, ігор чи інших проєктів може бути цікавим та мотивуючим для учнів.

Підготовка вчителів. Розробка навчально-методичного комплексу передбачає також і методичні рекомендації для вчителя до викладання програмування на Scratch. Це важливо, оскільки вчителі повинні мати необхідні навички та ресурси для успішного проведення занять.

Стимулювання творчості. Scratch дозволяє учням не лише вивчати програмування, але й реалізовувати свої творчі ідеї. Робота над власними проєктами може сприяти розвитку творчості та самовираження.

Сприяння різноманітності у навчанні. Scratch може бути використаний для навчання різних тем і предметів, включаючи математику, фізику, географію, що робить його універсальним інструментом для розвитку різних аспектів освіти.

Отже, розробка навчально-методичного комплексу для факультативу з програмування на Scratch може вносити важливий внесок у підготовку учнів до цифрового суспільства та розвиток їх компетентностей у галузі інформатики.

Мета роботи: розробка навчально-методичного комплексу для організації та проведення занять факультативу з програмування мовою Scratch для профільного навчання інформатики.

Для досягнення мети поставлені *завдання:*

- провести теоретичний аналіз наукових та методичних джерел з предметної області дослідження;
- дослідити практику використання Scratch в освітніх системах різних країн світу;
- проаналізувати можливі варіанти використання Scratch на заняттях інформатики у ліцєях;
- розробити навчальну програму та навчально-методичний комплекс для організації та проведення факультативу з програмування мовою Scratch для профільного навчання інформатики;
- провести педагогічний експеримент (констатувальний, формувальний та контрольний етапи) для дослідження ефективності використання навчально-методичного комплексу.

Об'єкт дослідження: курси за вибором та факультативні заняття з інформатики для ліцєїстів.

Предмет дослідження: планування та методичні рекомендації для проведення факультативу з програмування мовою Scratch для профільного навчання інформатики учнів ліцєїв.

Практичне значення. Розробка навчально-методичного комплексу для факультативу з програмування мовою Scratch для профільного навчання інформатики має практичне значення в декількох аспектах:

Розвиток цифрової грамотності. Допомагає учням розвивати навички цифрової грамотності, оскільки вони вчаться створювати програми, розуміти логіку програмування та працювати з інтерфейсом програмування.

Підготовка до професійного розвитку. Навички, отримані в процесі вивчення програмування на Scratch, можуть слугувати відмінною основою для подальшого вивчення інших мов програмування. Це може виявитися

корисним для тих, хто має амбіції працювати в галузі інформаційних технологій.

Розвиток творчих навичок. Сприяє розвитку творчих навичок учнів, оскільки вони можуть творити свої власні проекти, анімації та ігри. Це стимулює творчий підхід до розв'язання завдань.

Впровадження інтерактивного навчання. Допомагає вчителям впроваджувати інтерактивні методи навчання, що може поліпшити якість освіти та збільшити зацікавленість учнів у вивченні інформатики.

Поєднання теорії та практики. Дозволяє поєднати теоретичні знання з практичними навичками. Учні не лише вивчають теорію, але і застосовують її в реальних проєктах, що покращує їхнє розуміння та навички.

Залучення учнів до навчання. Використання Scratch може зробити процес навчання більш цікавим та привабливим для учнів. Вони можуть бути більш зацікавленими та мотивованими вивчати інформатику через створення власних проєктів.

Гнучкість та адаптабельність. Навчально-методичний комплекс може бути адаптований до різних потреб індивідуальних груп учнів. Це сприяє гнучкості у процесі навчання та врахуванню різних рівнів підготовки.

Підтримка вчителів. Допомагає підтримувати вчителів, надаючи їм засоби та ресурси для ефективного викладання програмування на Scratch.

Загалом, розробка такого навчально-методичного комплексу сприяє розвитку важливих навичок учнів, готує їх до викликів сучасного інформаційного суспільства та сприяє підвищенню якості навчання в галузі інформатики.

Структура роботи. Робота складається зі вступу, двох розділів, висновків, списку використаних джерел та додатків. Основний текст роботи містить 28 рисунків, 5 таблиць та викладений на 70 сторінках. Під час написання роботи використано 48 джерел.

РОЗДІЛ 1

SCRATCH У ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ

1.1. Аналіз предметної області

Інформатика є найважливішим компонентом навчального плану середньої освіти. Вона відіграє величезну роль при вирішенні одного з головних завдань освіти – формуванні цілісного світогляду, системно-інформаційної картини світу, навчальних і комунікативних навичок та основних психічних якостей учнів [32, с.54].

Практично всі концепції реформування середньої освіти ґрунтуються на широкому впровадженні комп'ютерних технологій у навчальний процес.

В останні десятиліття здійснювалась інформатизація всіх аспектів життя суспільства і всіх сфер виробничої діяльності. Стан переходу до інформаційного суспільства ставить нові цілі перед навчанням інформатики:

- соціальне замовлення на формування особистості з високим рівнем мислення: операційного, алгоритмічного, системного;
- розумінням внутрішнього устрою складних інформаційних процесів і систем та навичками їх декомпозиції;
- здатністю до творчої трансформації реальності;
- набуття навичок продуктивного та ефективного використання інформаційних технологій;
- набуття творчого досвіду як умови самореалізації в житті.

Інформатика стає однією з основних галузей наукових знань, що вивчають інформаційні процеси, методи та інструменти для опрацювання інформації. Це дисципліна, що швидко розвивається, пов'язана з використанням інформаційних технологій, сфера її застосування в житті постійно зростає [30, с.12].

У процесі навчання школярів реалізуються ідеї розвитку творчих здібностей. Серед шкільних предметів інформатика посідає особливе місце. Під час навчання інформатики у школярів формується світогляд,

розвиваються логічне та алгоритмічне мислення, інформаційна культура, набуваються навички використання засобів інформатизації, осмислюється феномен інформації, що є основою для формування фундаментальних знань.

Нині методична система навчання інформатики школярів інтенсивно розвивається – конкретизуються цілі та принципи, розвиваються методи й засоби навчання, які враховують не тільки сучасні умови інформатизації, фундаменталізації та гуманітаризації освіти, а й проблему школярів, пов'язану з труднощами в засвоєнні знань, що виникають у них через ті чи інші причини.

Одним зі складних розділів шкільного курсу інформатики є розділ алгоритмізації та програмування.

У процесі навчання школярам необхідно набути знання про форми представлення алгоритмів, складання блок-схем, засвоїти алгоритмічні структури, зрозуміти синтаксис мови програмування, конструкції операторів, набути навичок програмування алгоритмів тощо.

Один із підходів до навчання школярів розділу алгоритмізації та програмування може бути заснований на підборі спеціального програмного й апаратного забезпечення.

До їх числа можна віднести візуальне об'єктно-орієнтоване середовище програмування Scratch (Рис. 1.1), яке базується на традиціях мови Лого і Лего-Лого.

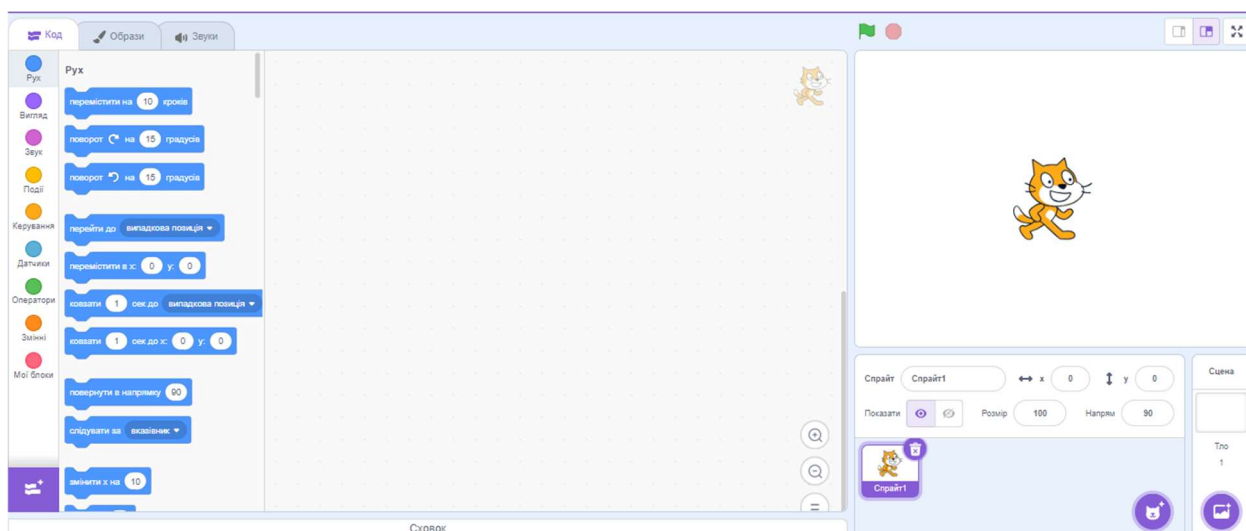


Рис. 1.1. Інтерфейс середовища програмування Scratch

Це середовище має зручний і нескладний інтерфейс, дає змогу не тільки програмувати, а й реалізовувати графіку та моделювання [18, с.81].

Основні особливості Scratch:

– блочне програмування: для розробки програм у середовищі Scratch просто потрібно комбінувати графічні блоки. Блоки створені таким чином, щоб їх можна було об'єднувати лише в синтаксично вірних конструкціях, що мінімізує ймовірність виникнення помилок. Різні типи даних мають відмінні форми, що відзначає їхню взаємну сумісність. Також можливо вносити зміни в програму під час її виконання, що дозволяє експериментувати з новими ідеями і вдосконалювати код навіть під час роботи програми;

– маніпуляція даними: у Scratch можна створювати програми, які контролюють та комбінують графіку, анімацію, музику та звуки. Це розширює можливості управління візуальними елементами даних, що стає все більш популярним у сучасній інформаційній культурі;

– спільна робота і обмін: сайт проєкту Scratch надає можливість переглядати та використовувати проєкти інших користувачів, модифікувати їх зображення та скрипти, а також додавати свої власні проєкти. Найбільш значущим досягненням є загальне середовище та культура, які сформувалися навколо використання Scratch.

Однією з основних концепцій мови програмування Scratch є сприяння розвитку ідей від початкової концепції до завершеного програмного продукту. У Scratch існують всі необхідні інструменти для цього, такі як стандартні конструкції для мови процедурного типу (послідовності, розгалуження, цикли, змінні, типи даних), об'єктно-орієнтовані можливості (об'єкти, поля, методи, передача повідомлень і обробка подій), інтерактивні можливості (взаємодія об'єктів, обробка подій від користувача та поза комп'ютером), паралельне виконання (одночасне виконання методів об'єктів), та можливість створення інтерфейсу користувача.

Ці особливості Scratch сприяють розвитку особистісних якостей учнів, таких як відповідальність, адаптивність, комунікативні вміння, творчість, допитливість, критичне та системне мислення, робота з інформацією та медіа, міжособистісна взаємодія, спрямованість на саморозвиток та соціальна відповідальність.

Педагогічний та дидактичний потенціал Scratch робить його перспективним інструментом для організації міждисциплінарної проєктної науково-пізнавальної діяльності школярів, спрямованої на їх особистісний та творчий розвиток [23].

На рис. 1.2 подана схема [29], на якій перераховані передумови вибору Scratch в навчальному процесі.



Рис. 1.2. Вибір Scratch у навчальному процесі

Основними компонентами програми є об'єкти-спрайти. Для них задається графічне представлення, яке може бути імпортоване з будь-якого джерела зображення, і скрипт дій, який складається з блоків за принципом drag-and-drop (перенесення). Ці блоки бувають декількох видів: рух, вигляд, звук, події, керування, датчики, оператори, змінні, мої блоки.

Вікно програми розділене на кілька частин (Див. Рис. 1.1). Справа вгорі велике вікно з білим фоном і об'єктом Кіт в центрі. У цьому вікні відбуватимуться події, запрограмовані нами. Об'єкт Кіт відразу з'являється за замовчуванням. Ми можемо використовувати його в нашій програмі, а можемо й не використовувати. Клік мишкою на зеленому прапорці вгорі запускає програму, а на червоному кружечку – зупиняє [24, с.5].

Спрайти можна не тільки вибрати з наявних, а й намалювати самому.

У 2014 році вийшла версія Scratch для дітей молодшого віку під назвою ScratchJr. Це мобільний застосунок для Android та iOS, у якому діти також керують якому діти також керують спрайтами, тільки в більш спрощеній формі.

У блоках не використовується текст, тому діти можуть вчитися програмувати до того, як навчаться читати, і їм доступний обмежений набір дій: прості рухи спрайтів і робота зі звуками та зображеннями.

Програмування в середовищі Scratch є візуальним і швидким, що важливо для тих, хто навчається [5, с.42].

Наведемо приклади завдань з використанням середовища програмування Scratch на уроках засвоєння нових знань та уроках комплексного застосування знань і вмінь.

Розроблені завдання орієнтовані на учнів 10-11-х класів і вік учнів – чотирнадцять-шістнадцять років. Уроки з ліцеїстами проходять у групах по 12-15 осіб.

Тривалість одного уроку 45 хв. Під час уроку відбувається часта зміна діяльності: сприйняття матеріалу на великому екрані та на слух, участь в обговоренні поставленого завдання, робота з паперовими інструкціями, робота на комп'ютері в середовищі Scratch, усне представлення свого проєкту.

Кожен ліцеїст, який бере участь у роботі з виконання запропонованого завдання, має змогу не лише розв'язати поставлену задачу, а й творчо доопрацювати свій мініпроєкт, запропонувати свій план виконання запропонованого завдання.

На уроці засвоєння нових знань занурення в мову програмування починається з вивчення теми «Знайомство із середовищем Scratch» [27].

Розглянемо завдання, які використовуються під час вивчення теми.

Завдання 1. Запуск програми Scratch і знайомство із середовищем Scratch

Слід зазначити, що всі дії, який вчитель пояснює, мають підтверджуватися візуально за допомогою демонстрації на екрані інтерфейсу

програми або зображеннями іконок інтерфейсу програми або зображеннями іконок команд.

Отже, учитель пропонує запустити програму та виконати деякі дії:

- обери команду Створюй, щоб відкрити вікно редактора програми;
- переглянь відео-фрагмент про можливості програми Scratch;
- якщо мова інтерфейсу програми англійська, то зміни її, натиснувши на значок «Глобус»;
- розгорни сцену на весь екран і знайди кнопки Пуск і Стоп;
- розглянь інтерфейс вікна редактора;
- запиши в зошит назви об'єктів інтерфейсу програми та їхні функції: спрайт – персонаж, об'єкт, герой; сцена – місце, де спрайти рухаються, малюють, взаємодіють; фон – задній план сцени;

Далі учитель знайомить учнів із властивостями спрайта. Пояснює, що для того, щоб спрайт робив цікаві речі, його слід запрограмувати, перемістивши блоки команд з панелі блоків у поле скриптів і з'єднати їх між собою. При цьому вчитель спочатку показує сам ці дії, потім пропонує виконати подібні дії учням [34, с.11].

Завдання 2. Створення першої програми. Учитель пояснює, що кожен скрипт починається з одного з блоків із закругленою «шапкою» з групи Події.

Пропонується виконати алгоритм дій:

- перетягни та «приклей» блок із групи Події;
- нижче першого блоку розташуй блок із групи Рух;
- перевір, чи так у тебе вийшло;
- виконай свою першу програму – перевір як працює скрипт. Для запуску скрипта натисни Зелений прапорець;
- перетягни та «приклей» блок із групи Події;
- нижче цього блоку розташуй блок із групи Зовнішній вигляд і вистав кількість секунд, що дорівнює 5;
- для запуску скрипта натисни клавішу Пробіл і виконай програму;

- перетягни і «приклей» блок із групи Події;
- нижче цього блоку розташуй блок з групи Рух і вистав кількість секунд – 5 с;
- перевір програму – для запуску скрипта клацни по ньому.

Завдання 3. Збереження та відкриття програми

Учитель пояснює, що для подальшої роботи з програмою, її слід зберігати. Пояснює алгоритм збереження, показує, куди може зберегтися проєкт, якщо не вибрати свій шлях, і пропонує виконати такі дії:

- для збереження програми виконай команду «Файл» – «Зберегти» та обери шлях збереження на свій комп'ютер.

Далі вчитель пропонує перевірити проєкт після збереження, запустивши його знову. Учням дається нова команда:

- для відкриття збереженого проєкту з комп'ютера слід виконати команду «Файл» – «Завантажити» з комп'ютера;
- перейти в потрібну теку та клацнути по потрібному проєкті.

На уроці комплексного застосування знань і умінь відбувається закріплення отриманих знань про середовище Scratch і команди роботи з блоками. Школярі вчаться знаходити помилки у програмі і виправляти їх.

Таким чином, простий візуальний інтерфейс середовища програмування Scratch дасть змогу учням розвинути алгоритмічне мислення та набути навичок складання програм. За допомогою блоків – основних компонентів середовища – учні самостійно можуть створювати кадри руху персонажа, роблячи тим самим перші кроки у світі програмування [33, с.87].

У процесі використання середовища програмування Scratch школярі реалізують творчий підхід, використовують інформаційні та телекомунікаційні технології, набувають уміння навчатися тощо. Школярі можуть конструювати свої програми-процедури з блоків аналогічно тому, як вони конструюються в Лего.

Використовуючи в процесі навчання середовища програмування Scratch, школярі вчаться самостійно складати алгоритми та усвідомлювати

функціональність роботи циклів і умовних операторів, вчать приймати самостійні рішення, можуть бачити практичне призначення алгоритмів і програм.

Крім того, скрипти програмування в середовищі Scratch схожі із синтаксисом мови програмування Pascal, що може полегшити засвоїти цю мову програмування.

Використання візуального об'єктно-орієнтованого середовища програмування Scratch під час навчання інформатики школярів, сприяє розвитку наочно-образного мислення і дає змогу задовольняти пізнавальні інтереси школярів.

Scratch використовується для робототехніки та STEM-освіти завдяки своєму інтуїтивно зрозумілому графічному інтерфейсу, який дозволяє навчати програмуванню без використання складних кодів. Ось як він застосовується в цих контекстах:

Програмування роботів. Підтримка різних робототехнічних платформ: Scratch може взаємодіяти з різними роботами та мікроконтролерами, такими як LEGO Mindstorms, Makey Makey, micro:bit тощо. Керування рухом: Учні можуть програмувати рухи роботів, визначати їхню поведінку та реагування на зовнішні події.

STEM-освіта. Вивчення наукових концепцій: Scratch може бути використаний для створення інтерактивних симуляцій та експериментів, що допомагає вивчати фізику, хімію, біологію тощо.

Математичне моделювання. Учні можуть створювати проекти, які включають математичні концепції, такі як геометрія, алгебра чи статистика.

Інтерактивні уроки з програмування. Scratch використовується для навчання основ програмування в рамках STEM-предметів.

1.2. Використання Scratch у системах освіти різних країн світу

Ефективність навчання, на нашу думку, залежить від методик і підходів викладання комп'ютерних наук у школах. Грамотно побудований курс

програмування допоможе школярам зрозуміти основні концепції, такі як алгоритми, мови програмування, архітектура обчислювальних систем тощо.

Програмування дає школярам можливість проявити себе в ролі творця або проєктувальника: володіючи знаннями мови програмування, можна написати практично будь-яку програму, а цікаві та актуальні для розробки теми допоможуть стимулювати розвиток у дітей творчих здібностей. При цьому навички, одержувані школярами, сприяють розвитку критичного мислення, освоєнню прийомів розв'язання проблем в інших галузях.

Однак школи України та багатьох розвинених країн Заходу потребують реформування навчальних планів на державному рівні [17].

Одним із підходів у цьому напрямі вважається можливо раннє навчання школярів алгоритмізації та програмування. У Великобританії, Франції, Ірландії розроблено курси з програмування для учнів початкових класів, а також проводяться додаткові заняття для школярів, де вони можуть познайомитися з основами алгоритмів і створенням найпростіших комп'ютерних програм. У школах Австрії, Фінляндії школярам викладають основи програмування, де застосовують середовище Scratch.

Південна Корея

У Південній Кореї учні в середній школі вивчають програмування з 2015 р., воно представлене кількома предметами циклу «Комп'ютерні науки»: «Інформаційне суспільство і комп'ютер», «Програмування», «Програмне забезпечення», «Комп'ютерне моделювання», «Комп'ютерний дизайн». Причому цей цикл предметів починається в молодшій школі (з 6 років) [10, с.18]. Учням старшої школи пропонуються курси за вибором.

Сполучені штати Америки

Слід зазначити, що сучасна Computer Science походить зі США, але тільки в 10 % шкіл США є курси з вивчення основ програмування, а вивчення комп'ютерних наук починається з університетів. Для зміни цієї ситуації у США було запущено агітаційні та мотиваційні відеоролики за участю відомих програмістів, засновників ІТ-компаній, спортсменів і зірок з метою викликати

високу зацікавленість до вивчення програмування у школярів і широкою аудиторії країни.

Але, незважаючи на такі величезні успіхи в цій науці, у масовій американській школі не передбачено навчання основ програмування. Утім, у країні все ж є освітні програми, що дають змогу вивчати програмування учням старшої школи, які надалі бачать своє майбутнє поруч з ІТ.

Здебільшого, вся американська Computer Science починається з університетської освіти [38, с. 21].

Scratch є популярним інструментом для навчання програмування в школах США. Мова програмування Scratch є частиною Computer Science для всіх (CSforAll), ініціативи, яка має на меті надати можливість учням навчатися комп'ютерній науці в усіх школах.

У Сполучених Штатах Америки Scratch відіграє важливу роль у навчанні програмуванню та розвитку комп'ютерної грамотності серед учнів. Ось кілька аспектів досвіду використання Scratch у США [40]:

- *Навчальні програми в школах.* У багатьох школах США Scratch включений до навчальних програм з комп'ютерних наук та технологій. Він використовується для навчання учнів основам програмування та стимулювання їхньої креативності.

- *Ініціатива «Code.org».* Організація «Code.org» пропагує навчання програмуванню у школах США та по всьому світу. Вони надають навчальні матеріали та курси, включаючи використання Scratch, для вчителів та учнів [37, с.4].

- *Гурткова робота.* У багатьох школах та бібліотеках США створюються клуби та гуртки, де учні можуть навчатися програмуванню з використанням Scratch. Це дозволяє дітям вдосконалювати свої навички у невимушеній атмосфері.

- *Ініціатива «CSforAll».* Ця програма має на меті забезпечити можливість учням навчатися комп'ютерним наукам в усіх школах США. У

цьому контексті використання Scratch є однією з ключових складових навчання.

- *Конкурси та змагання.* У США проводяться різні конкурси та змагання для учнів, де вони можуть демонструвати свої навички у створенні проєктів з використанням Scratch.

- *Інтеграція в STEM-навчання.* Scratch використовується для навчання предметів STEM (наука, технології, інженерія, математика), де учні можуть створювати симуляції та інтерактивні додатки.

- *Ініціатива «Girls Who Code».* Організація «Girls Who Code» сприяє навчанню програмуванню дівчат. Вони використовують інструменти, включаючи Scratch, для навчання комп'ютерних навичок.

- *Спільноти вчителів та учнів.* Вчителі та учні у США активно спілкуються через форуми, соціальні мережі та інші платформи, де діляться досвідом використання Scratch та навчальними матеріалами [25, с.12].

Великобританія

Як успішний приклад впровадження нових навчальних стандартів можна вказати Великобританію. Восени 2015 р. у всіх школах Великобританії було впроваджено навчальні плани з вивчення основ програмування – «Національна навчальна програма Англії: комп'ютерні програми навчання» [4].

Так, учням від 5 років за курсом програмування пропонується вивчати створення елементарних програм, а з 11 років учні опановують різноманітні алгоритми та, принаймні, дві мови програмування. Учням пропонується вивчати просту логіку, роботу з двійковими системами та спільну роботу комп'ютерного обладнання та програмного забезпечення, відповідним чином використовувати структури даних; розробляти модульні програми, які використовують процедури або функції. Часто використовувані мови програмування – це Java або Python.

Усі ці нові підходи та програми викладання комп'ютерних наук викликали чималий негатив з боку організацій, чий ІКТ стандарти були довгі

роки впроваджені в школах Великобританії. Нова навчальна програма передбачає викладати дитячу комп'ютерну науку, інформаційні технології та цифрову грамотність, вона спрямована на навчання написання коду і створення власних програм.

Передбачається вивчення роботи з комп'ютером, а також вивчення принципів роботи комп'ютера і можливостей змусити його працювати на себе. Введення нового навчального стандарту в шкільні курси було пов'язане з постійними скаргами технічних компаній на брак кваліфікованих кадрів. До розробки курсу були залучені Microsoft, Google і Королівська інженерна академія.

У Великобританії Scratch є популярним інструментом для навчання програмуванню та розвитку комп'ютерної грамотності в школах.

Ось деякі аспекти досвіду використання Scratch у Великій Британії:

- *Code Club*. Code Club – це ініціатива, яка надає навчальні ресурси та матеріали для вчителів та гурткових керівників. Вони розробили набір курсів, включаючи використання Scratch, для навчання програмуванню дітей у віці від 9 до 13 років.

- *BBC Micro:bit*. Scratch можна поєднати з мінікомп'ютером BBC Micro:bit є популярним у Великобританії. Учні можуть використовувати Scratch для програмування Micro:bit та створення різноманітних проєктів.

- *Digital Making at Home*. Raspberry Pi Foundation пропонує програму «Digital Making at Home», яка включає в себе навчальні посібники та відеоуроки, у тому числі використання Scratch, для навчання програмуванню та творчому дизайну.

- *BBC Teach*. BBC Teach надає навчальні ресурси для вчителів, включаючи уроки, пов'язані з Scratch, для навчання комп'ютерним наукам та програмуванню.

- *STEM Learning*. Організація STEM Learning надає навчальні ресурси та підтримку для вчителів, які використовують Scratch для навчання предметів STEM.

– *Інновації у навчанні.* Великобританія є лідером у впровадженні інноваційних методів навчання. Використання Scratch в контексті креативного програмування є однією з сучасних педагогічних практик.

– *Університети та додаткові навчальні ресурси.* Великобританія має численні університети та організації, які пропонують навчальні ресурси для вчителів та учнів, щоб допомогти їм в освоєнні Scratch та програмуванні загалом [18, с.9].

Польща

На офіційному сайті Міністерства освіти та науки Польщі можна знайти навчальні програми з інформатики – «Інформатика для учнів початкової школи» авторства Sylwia Maciuk та «Твій світ завтра» авторства Agnieszka Krawińska [6].

У навчальному змісті програми навчання інформатики під назвою «Інформатика для учнів початкової школи» для четвертого класу теми «Алгоритми навколо нас» та «Розв’язування задач за допомогою історій та алгоритмів» вивчаються 6 годин, що складає 17% від загального обсягу годин інформатики. У п’ятому класі виділяється 18 годин на теми з програмування, а саме «Конструювання алгоритмів. Операції над числами, властивості множин», «Вступ до розвитку комп’ютерного мислення. Розв’язування задач за допомогою алгоритмів» та «Освітні платформи з візуальною мовою програмування», а це складає вже 51,4. У шостому класі теми «Освітні платформи з візуальною мовою програмування», «Вправи для розвитку обчислювального мислення. Розв’язування задач Використання алгоритмів. Реалізація та тестування програм» та «Розробка алгоритмів. Робототехніка та штучний інтелект» вивчаються 16 годин – 40% від загального обсягу годин інформатики. В сьомому класі також є місце для програмування – 12 годин на теми «Вправи для розвитку обчислювального мислення. Розв’язування задач Використання алгоритмів. Розробка способів візуалізації алгоритмів», «проектування, розробка та тестування програм у процесі вирішення проблем» та «Робототехніка та штучний інтелект. Керуючі програми» – 34,2%.

У восьмому класі вивчаються теми «Розвиток знань про алгоритми. Демонстрація роботи алгоритмів», «Оптимізація програмного коду в процесі розв'язування задач» та «Робототехніка та штучний інтелект. Проектування, розробка та тестування програм керування роботами» 11 годин, а це 31,4% від загального обсягу годин інформатики.

У пункті, що описує організацію умов та проведення навчання зазначено пропонувані програми і веб-сайти, що стосуються навчального контенту. До вивчення програмування віднесли Scratch та Scratch Jr, а також веб-портали, що підтримують навчання програмування – CodeAcademy, Unmarco.Allcancode, Blockly.Games, Tynker, Toxicode, Codecombat, Studio.Code.

Китай

У Китаї, з огляду на зростаючу потребу в ІТ-фахівцях, які можуть створювати складні додатки та керувати ними, в освітній системі відбувається навчання програмуванню в самому ранньому віці [9]. Багато китайських дітей зараз отримують доступ до програмування вже перед школою. Зазвичай вони працюють над освоєнням математики та китайської мови, але зараз намагаються здобути нові навички і в галузі технологій, комп'ютерних наук і програмування.

Австралія

У новому навчальному році, програма викладання інформатики в австралійських школах була капітально оновлена. Тепер основи програмування з'являться у школярів з третього класу. Починаючи з сьомого класу, школярі вивчатимуть одну зі звичайних мов програмування, а в старших класах планується дати основи об'єктно-орієнтованого програмування.

Однак освітня програма не затверджена, рішення ухвалюватимуть індивідуально в кожному штаті. Можливо, що навичками програмування володітимуть не всі австралійські випускники [32, с.32].

У Австралії Scratch є популярним інструментом для навчання програмуванню та комп'ютерній грамотності.

Ось деякі аспекти досвіду використання Scratch у цій країні:

– *ACARA (Австралійська рада з оцінювання та навчання)*. ACARA визначає освітні стандарти для Австралії, і Scratch включено в ці стандарти як один із інструментів для навчання цифровим технологіям та програмуванню.

– *Digital Technologies Curriculum*. В 2016 році був введений новий курс «Digital Technologies» в освітніх програмах Австралії для учнів з 5 по 10 клас. Scratch використовується як частина цього курсу для вивчення програмування та цифрових навичок.

– *CoderDojo*. CoderDojo – це глобальна організація, яка пропагує навчання програмуванню для дітей. Вони мають філії у багатьох містах Австралії, де діти можуть навчатися програмувати, включаючи використання Scratch.

– *ICTENSW*. ICT Educators NSW є професійною організацією, яка надає підтримку вчителям інформаційних технологій у Новому Південному Уельсі. Вони організують події, воркшопи та навчальні ресурси для вчителів, включаючи використання Scratch.

– *Обласні освітні офіси*. Різні обласні освітні офіси та організації пропонують підтримку та навчальні ресурси для вчителів з впровадження Scratch в навчальний процес.

– *Australian Computing Academy*. Австралійська Академія Комп'ютерних Наук (АСА) надає навчальні ресурси та курси для вчителів та учнів для навчання цифрових технологій та програмування, включаючи використання Scratch.

– *Додаткові ресурси для вчителів*. Вчителі можуть користуватися онлайн-платформами, такими як «Education Queensland's Learning Place» та «Victorian Department of Education's FUSE», де є навчальні матеріали та підтримка для викладання Scratch.

Фінляндія

У Фінляндії, яка відома своєю передовою системою освіти, використання Scratch є важливим компонентом навчання цифровим навичкам та програмуванню. Зупинимось на декілька аспектах досвіду використання Scratch у Фінляндії:

– *Включення в навчальні програми.* Scratch включено до навчальних програм у Фінляндії. Він використовується для навчання учнів основам програмування та комп'ютерної грамотності в рамках уроків інформатики та технологій.

– *Розвиток креативності та логічного мислення.* Учні використовують Scratch для створення власних проєктів, включаючи ігри, анімації та інтерактивні історії. Це сприяє розвитку їхньої креативності, логічного мислення та проблемного вирішення завдань.

– *Інтеграція в STEM-навчання.* Scratch використовується для навчання предметів STEM (наука, технології, інженерія, математика), де учні можуть створювати симуляції та інтерактивні додатки.

– *Відкриті ресурси та спільноти.* У Фінляндії існують різноманітні відкриті ресурси та спільноти, де вчителі та учні можуть обмінюватися навчальними матеріалами та ідеями для використання Scratch.

– *Робототехніка.* Scratch може бути поєднаний з робототехнічними платформами, що дозволяє учням програмувати та керувати роботами.

– *Відкриті конкурси та змагання.* Учні можуть брати участь у відкритих конкурсах та змаганнях, що сприяє мотивації та розвитку їхніх навичок.

– *Ініціатива «Me & MyCity».* Проєкт «Me & MyCity» надає учням можливість створити власну віртуальну містечко, використовуючи Scratch, де вони можуть відтворювати ситуації з реального життя та навчатися фінансовій грамотності.

– *Ресурси для вчителів.* Вчителі можуть скористатися різноманітними навчальними ресурсами, включаючи онлайн-курси та матеріали, які сприяють ефективному використанню Scratch в навчанні.

Також варто зазначити і наявність конкурсів з програмування в Scratch які проходять у різних куточках світу.

First Lego League (FLL) [1] – міжнародна програма, спрямована на стимулювання інтересу до науки, техніки та інженерії серед дітей та підлітків віком від 4 до 16 років. В програмі задіяний конструктор Lego Mindstorms для створення та візуального програмування роботів, включаючи використання Scratch. FLL охоплює багато аспектів STEM (наука, технологія, інженерія та математика) та включає в себе також елементи співпраці, командної роботи та дослідницьких проєктів.

Основні дивізіони First Lego League:

Discover, для учасників 4-6 років – вступна програма STEM, що розпалює зацікавленість молодших учнів та сприяє формуванню їхньої навчальної активності через використання практичних занять у класі та вдома за допомогою конструкторів LEGO DUPLO.

Explore, для учасників 6-10 років – у команді Explore діти зосереджуються на вивченні основ інженерії через вирішення реальних проблем, вивчають навички проєктування та програмування, а також розробляють унікальні рішення, використовуючи конструктори LEGO та платформу LEGO® Education SPIKE Essential.

Challenge, для учасників 9-16 років – суть змагання полягає в дружбі, оскільки команди учнів беруть участь у дослідженні, вирішенні проблем, програмуванні та розробці. Вони будують та програмують LEGO-робота, який виконує місії роботизованої гри, яка може бути створена за допомогою Scratch.

FLL визначає нову тему кожного сезону, яка визначає завдання для роботичного турніру та може слугувати інспірацією для дослідницьких

проектів. Програма FLL надає дітям можливість розвивати навички програмування, робототехніки, аналізу та розв'язання реальних проблем.

CoderDojo Coolest Projects: CoderDojo. CoderDojo [2] – це глобальна мережа безкоштовних клубів, призначених для навчання програмування дітей та підлітків. Ідея полягає в тому, щоб створити відкрите, дружнє середовище, де діти можуть навчатися програмуванню, створювати проекти та ділитися досвідом з іншими учасниками.

Одною з важливих подій у світі CoderDojo є «Coolest Projects» (Найкрутіші Проекти). Це щорічний захід, де учасники CoderDojo представляють свої програмні розробки та проекти під час виставки. Захід призначений для того, щоб надихнути, підтримати та визнати творчість та зусилля молодих програмістів.

Деякі ключові аспекти CoderDojo Coolest Projects включають:

- виставка проектів: учасники мають можливість представити свої проекти та продемонструвати їхню роботу іншим учасникам та відвідувачам;
- категорії проектів: проекти можуть включати різноманітні області програмування, такі як веб-розробка, ігри, мобільні додатки, робототехніка та інше;
- взаємодія: Coolest Projects надає можливість молодим програмістам зустрічатися, обмінюватися ідеями, навичками та досвідом з різних частин світу;
- нагороди та визнання: учасники можуть отримати визнання за свої досягнення та можливо отримати нагороди в різних категоріях.

CoderDojo Coolest Projects сприяє розвитку творчого мислення, командної роботи та навичок програмування серед молоді та створює платформу для виявлення та визнання їхніх зусиль у сфері технологій.

Scratch Day [8] – це щорічна глобальна ініціатива, яка святкується у спільноті користувачів Scratch. Scratch Day проводиться як свято, під час якого спільнота Scratch об'єднується, щоб святкувати та ділитися своїми творчими

проєктами, ідеями та навичками програмування. Основні аспекти Scratch Day включають:

- заходи на локаціях: люди з усього світу організують різні заходи – зустрічі, воркшопи, турніри та інші активності, де учасники можуть обмінюватися досвідом та демонструвати свої проєкти;
- онлайн активності: крім заходів на локаціях, Scratch Day також включає в себе онлайн ініціативи, такі як вебінари, конкурси, челенджі та інші можливості для участі через Інтернет;
- тематика та проєкти: щороку Scratch Day має нову тематику або фокус, який надається головним організатором. Учасники можуть створювати та демонструвати проєкти, які відповідають цій темі;
- спільнота та взаємодія: Scratch Day сприяє взаємодії та обміну ідеями серед користувачів Scratch, створюючи позитивне та мотивуюче середовище для творчості та вивчення програмування.

Scratch Day відзначається своєю відкритістю та доступністю для всіх, незалежно від рівня досвіду. Це свято розширює можливості для вивчення програмування та розвитку технологічних навичок серед дітей, підлітків та дорослих.

В Україні також проводяться конкурси та фестивалі з програмування в Scratch.

Наприклад інтернет-ресурс «Інформатика» з 2016 по 2021 рік провів 4 фестивалі з Scratch-проєктів, участь в якому приймали учні 1-11 класів у різних номінаціях – «Навчаємось разом з Рудим Котом!», «Цифрові історії», «Ігри» [21].

Green Coding [3] – це онлайн конкурс-хакатон з програмування на платформі Scratch, спрямований на привертання уваги дітей та молоді до актуальних екологічних проблем. Учасники формують команди з 4 учнів та 1 ментора, розробляють свій власний проєкт на екологічну тематику та представляють його на онлайн конференції. Подія відбувається у межах ініціативи Meet and Code, яку фінансують корпорація SAP та Федеральне

міністерство внутрішніх справ Німеччини. Український конкурс з програмування для дітей також стартував у 2019 році та проводиться за підтримки різних організацій, таких як Фонд BrainBasket та ІТ-компанія EPAM.

SCRATCH CUP [20] – це міжнародні змагання з програмування, які збирають разом юних програмістів 5-7 класів загальноосвітніх закладів Закарпатської області України. Конкурс проводиться в рамках проєкту «Віртуальна реальність єднає Карпати: інтерактивні шкільні лабораторії» спільної операційної програми Румунія-Україна 2014-2020.

Конкурс SCRATCH CUP забезпечує можливість для учасників розвивати свої навички програмування, креативне мислення та командну роботу, а також стимулює інтерес до технологій та ІТ-сфери.

У висновку хотілося б зазначити, що в багатьох країнах світу всерйоз задумались про те, що сучасній людині необхідно знати основи програмування нітрохи не менше, ніж уміти читати і писати. Інформатика як освітня дисципліна швидко розвивається.

Варто відзначити, що кожна країна може впроваджувати Scratch зі своїми особливостями та адаптаціями до власної освітньої системи. Усі ці приклади демонструють широкий спектр можливостей, які надає використання Scratch у системі освіти для навчання програмуванню та розвитку комп'ютерної грамотності учнів.

Середовище Scratch є потужним інструментом для навчання програмуванню та розвитку комп'ютерної грамотності в учнів різних вікових категорій. Scratch використовується в системах освіти різних країн світу як інструмент для навчання основ програмування та алгоритмізації.

1.3. Аналіз варіантів можливого застосування Scratch у навчальних програмах старшої профільної школи

Профільне навчання дає змогу школяреві старшого ступеня обрати свій майбутній професійний шлях.

Цілі вивчення інформатики в старших класах:

- формування інформаційної та алгоритмічної культури;
- формування уявлення про комп'ютер як універсальний пристрій опрацювання інформації;
- розвиток основних навичок і вмінь використання комп'ютерних пристроїв;
- формування уявлення про комп'ютерні пристрої;
- формування уявлення про основні досліджувані поняття: інформація, алгоритм, модель та їх властивості;
- розвиток алгоритмічного мислення, необхідного для професійної діяльності в сучасному суспільстві;
- розвиток умінь складати та записувати алгоритм для конкретного виконавця; формування знань про алгоритмічні конструкції, логічні значення та операції;
- знайомство з однією з мов програмування та основними алгоритмічними структурами (лінійною, умовною та циклічною);
- формування вмінь формалізації та структурування інформації, вміння обирати спосіб представлення даних відповідно до поставленого завдання (таблиці, схеми, графіки, діаграми), з використанням відповідних програмних засобів обробки даних;
- формування навичок і вмінь безпечної та доцільної поведінки під час роботи з комп'ютерними програмами та в Інтернеті, вміння дотримуватися норм інформаційної етики та права [31, с.54].

Дослідницька та проєктна діяльність є невід'ємною частиною роботи школяра впродовж вивчення всього безперервного курсу інформатики.

У старшій школі проєктна та дослідницька робота зі школярами відіграє особливо важливу роль у профільному навчанні інформатики, що пов'язано і з поглибленим вивченням предмета, і з набуттям навичок, які використовуються в практичній діяльності.

Наступні компетенції [5], необхідні для професійної діяльності, пов'язаної з інформаційними технологіями, тією чи іншою мірою формуються під час роботи над дослідницькими проектами:

– *Системне мислення*: робота над ІТ проектом передбачає розгляд поставленої задачі та її розв'язання з погляду системного підходу, що є основою наукового світогляду [28]. Це формує здатність розглядати проект, що реалізовується, як систему взаємопов'язаних модулів, що має системний ефект, властивість, яка характерна для системи в цілому, і не характерна для її частин окремо.

– *Програмування*: реалізація проекту виконується однією з мов програмування. Уміння програмувати дає змогу грамотно використовувати сучасні інформаційні технології та системи в будь-якій предметній галузі, можливо, безпосередньо не пов'язані з ІТ, завдяки логічному, алгоритмічному мисленню, яке формується під час освоєння програмування [9].

– *Міжгалузєва комунікація*: здатність пов'язувати між собою в одному проекті інформацію з різних предметних галузей, застосувати знання з однієї предметної галузі для перетворення інформації з іншої предметної галузі, здатність отримати інформацію від експерта в тій чи іншій предметній галузі та представити її однією з формальних мов кодування інформації, здатність працювати з фахівцями різних предметних областей

– *Творчі навички*: робота над дослідницьким проектом дає змогу реалізувати творчий потенціал того, хто навчається, вона виконується в такий спосіб, що виникає потреба генерувати ідеї, мислити конструктивно, створювати алгоритми, а також вимагає ефектного і грамотного оформлення результатів роботи.

– *Уміння працювати в групі*: деякі проекти виконуються в групі (2-4 особи), і в кожного проекту є 2-3 керівники. Обов'язки розподіляються не тільки між учнями, а й між керівниками, кожен з яких виконує свою функцію. У цій ситуації в учня формується модель, за якою розподіляються обов'язки в роботі над груповим проектом і способи взаємодії між усіма учасниками,

зокрема набувається навичка дистанційного спілкування з використанням сучасних комунікаційних технологій.

Крім перерахованого, під час роботи над проектом у учнів формуються такі метапредметні навички [11, с.45], як:

- постановка мети, планування шляхів її досягнення (необхідний етап будь-якого дослідження або проекту);
- здатність обирати ефективний шлях досягнення мети (наприклад, використання ефективного алгоритму розв'язання задачі);
- оцінка отриманого результату (відповідно до критеріїв, складених на етапі моделювання, на етапі отримання математичної моделі);
- підбиття підсумків проекту або його етапу;
- коригування шляху досягнення результату за умов, що змінюються (наприклад, у разі додавання додаткових характеристик у математичний опис, у разі зміни початкових умов);
- уміння працювати з різними знаковими системами, моделями, виконувати перекодування інформації з однієї системи в іншу (досягається під час програмування, проектування за допомогою схем).

Проектна та дослідницька діяльність за своїм змістом незалежно від тематики проектів націлена на здобуття цілої низки метапредметних результатів навчання і є одним із найефективніших способів їх досягнення.

Розробка ігрових Scratch-проектів на основі компетентнісного підходу. Основна спрямованість компетентнісних завдань полягає в усвідомленому набутті знань і вмінь формулювати рішення, планувати процеси для отримання результатів, оцінювати їх правильність і оптимальність, виявляти і виправляти помилки.

Пропонується розробити факультатив для впровадження компетентнісного підходу до навчання програмування та впровадити його в навчальний процес з інформатики [14, с.4].

Варто зазначити, що бажання дітей реалізовувати свої ідеї в Scratch виходить набагато далі, ніж шкільна програма. Тому розробка системи

факультативу може розвивати навички та компетенції, мотивувати вчителів та учнів, а також забезпечити науково-методичний супровід навчального процесу.

Факультатив – це відома сучасна форма навчання, під час якої відпрацьовуються практичні навички з використанням різноманітних методів і прийомів для поглиблення знань, розширення кругозору, виховання самостійності та наполегливості, розвитку навичок узагальнення та міркування.

Метою факультативів є заохочення взаємодії між учасниками, розвиток ключових та предметних компетентностей, підвищення рівня інформаційної культури, підвищення інтересу до навчання та розширення кругозору.

Використання інформаційно-комунікаційних технологій розвиває ключові компетентності учнів. Коли діти вивчають скретч-програмування, у них розвивається не лише логічне та алгоритмічне мислення, а й навички роботи з мультимедіа, що створює умови для активного дослідницького навчання. Учні вчаться розуміти і створювати комп'ютерні програми та використовувати комп'ютер для вирішення різних завдань. Таким чином, на зміну застарілій концепції «користувача комп'ютера» приходять концепція «творця і співавтора інформаційних просторів».

На рис. 1.3 наведено приклад формування складових інформаційної компетентності учнів [18] при проходженні майстер-класів.



Рис. 1.3. Формування складових інформаційної компетентності учнів

Пропонований факультатив складається з п'яти блоків з різним категоріями проєктів: «Знайомство з Scratch», «Створення ігор», «Scratch+робототехніка», «Креативне програмування» та «STEM».

Унікальною особливістю організації та проведення факультативів є те, що вони можуть проходити в різних форматах, як в аудиторіях, так і на віддалених локаціях. За відсутності високошвидкісного інтернету особливістю викладання в аудиторії є наявність програми Scratch на комп'ютері, друкованих матеріалів, демонстрацій проєктів та живих дискусій [36, с.89].

За наявності вільного доступу до глобальної мережі Інтернет, всі учасники мають доступ до інструкцій, розташованих на сайті «Факультатив з Scratch» [35], працюють в онлайн редакторі Scratch, публікують проєкти в спільноті скретчерів [7], готують презентацію своєї роботи, переглядають та оцінюють результати, використовуючи сервіси хмарних технологій (google-презентації, google- таблиці, google-форми) [15, с.15].

Метою факультативу є розвиток компетентностей у сфері навчання впродовж життя, інформаційно-цифрової, обізнаності та самовираження, екологічної грамотності та здорового способу життя. Тому завдання факультативу є наступними:

- розвиток алгоритмічного та логічного мислення;
- розвиток самостійності та наполегливості у досягненні цілей;
- розвиток культури взаємоповаги та дискурсу;
- розвиток культури самовираження та здорового способу життя;
- виховання культури взаємоповаги та ведення дискусії;
- формування навичок узагальнення;
- підвищення рівня інформаційної культури.

Виходячи з поставлених цілей і завдань, будується процес організації факультативу.

Орієнтовні етапи проведення факультативу такі:

- організаційний момент, психологічна підготовка до факультативу;

- аналіз етапу створення ігрового проєкту;
- розробка сценаріїв;
- розробка алгоритмів реалізації сценаріїв;
- створення та аналіз схеми гри;
- пошук необхідної довідкової інформації;
- розробка необхідних спрайтів та фонів;
- програмування алгоритмів;
- валідація та тестування гри;
- опис та презентація проєкту;
- підготовка презентації проєкту;
- аналіз та оцінка проєкту;
- підбиття підсумків [39, с.81].

Таким чином, вивчення програмування мовою Scratch надає широкі можливості для ефективного формування ключових міжпредметних компетентностей учнів. Проєктування та реалізація ігрових Scratch-проєктів дає можливість активізувати навчальний процес, створювати інтерактивні анімаційні матеріали, співпрацювати над проєктами через Scratch-спільноту, обмінюватися результатами та мотивувати до самостійної роботи.

Висновки до розділу 1

У процесі навчання та вдосконалення навичок програмування із використанням середовища Scratch ліцеїсти реалізують творчий підхід, використовують інформаційні та телекомунікаційні технології, набувають уміння: навчатися, самостійно складати алгоритмічні структури та усвідомлювати функціональність їх роботи, виокремлюють практичне значення алгоритмів і програм. Крім того, використання Scratch під час навчання інформатики сприяє розвитку наочно-образного мислення і дає змогу задовольняти пізнавальні та творчі інтереси та потреби ліцеїстів.

Згідно проведеного літературного огляду, у багатьох країнах світу серйозно задумалися про те, що сучасній людині необхідно знати основи програмування нітрохи не менше, ніж уміти читати і писати.

Варто відзначити, що кожна країна впроваджує Scratch зі своїми особливостями та адаптаціями до власної освітньої системи. Усі ці приклади демонструють широкий спектр можливостей, які надає використання Scratch у системі освіти для навчання програмуванню та розвитку комп'ютерної грамотності учнів.

Середовище Scratch є потужним інструментом для навчання програмуванню та розвитку комп'ютерної грамотності учнів різних вікових категорій. Він використовується в системах освіти різних країн світу як інструмент для навчання основ програмування та алгоритмізації.

Таким чином, вивчення програмування на Scratch дає широкі можливості для ефективного формування ключових, міжпредметних та предметних компетентностей учнів. Проектування та реалізація Scratch-проектів ігрового характеру дає можливості для активізації процесу навчання, створення інтерактивних анімованих матеріалів, колективної співпраці над проектами та обмін результатами через Scratch-спільноту, а також мотивує до самостійної діяльності.

РОЗДІЛ 2

МЕТОДИЧНА РОЗРОБКА ФАКУЛЬТАТИВНОГО КУРСУ З ПРОГРАМУВАННЯ МОВОЮ SCRATCH ДЛЯ ПРОФІЛЬНОГО НАВЧАННЯ ІНФОРМАТИКИ УЧНІВ ЛІЦЕЇ

2.1. Методичні особливості вивчення Scratch у ліцеї

Вивчення Scratch у ліцеї відкриває учням захоплюючий та пізнавальний світ програмування через гру, що робить процес не лише корисним, але й цікавим. Неможливо заперечити що ця платформа є надзвичайно корисною для розвитку алгоритмічного мислення та цифрової компетентності серед школярів.

Одна з основних переваг використання Scratch полягає у його інтерактивному характері, завдяки якому навчання стає схожим на захоплюючу гру. Учні можуть не тільки отримувати теоретичні знання, але й використовувати їх на практиці шляхом створення власних проєктів та програм.

Педагогічна методика вивчення Scratch у ліцеї передбачає застосування ефективних підходів, якими максимально спирається засвоєння матеріалу учнями:

1. Орієнтованість на проєктну діяльність. Такий підхід передбачає використання практичних проєктів для вивчення програмування. Для впровадження цього підходу можна використовувати:

а. створення ігор: можна запропонувати учням створити свою власну гру у Scratch в якій можна додавати різних персонажів, ворогів та рівнів складності;

б. створення інтерактивних історій: для створення історії учні можуть додавати різні гілки сюжету, елементи вибору та анімації;

с. симуляція та експерименти: це дозволить вивчити певні концепції або явища, можна створити симуляцію фізичного експерименту або імітацію екосистеми;

d. створення власних блоків: це сприятиме розвитку розуміння функцій та абстракцій в програмуванні;

e. презентація проєктів: учні можуть демонструвати свої роботи перед класом, що розвивати навички публічного виступу та сприятиме обміну ідеями та враженнями серед учнів;

f. використання розширень та зовнішніх ресурсів: може включати в себе використання зображень, звуків, підключення до зовнішніх сенсорів та пристроїв.

Завдання, орієнтовні на проєктну діяльність створюють таке навчання, що сприяє кращому розумінню та застосування програмування у реальних ситуаціях.

2. Інтеграція з іншими предметами:

a. мова та література: заохочення учнів створювати інтерактивні казки чи ігри на основі літературних творів. Це може полегшити розуміння сюжетів та розвивати навички написання;

b. математика: створення гри, в якій гравець вирішує математичні завдання, може допомогти закріплювати знання математики та розвивати логічне мислення;

c. мистецтво: учні можуть створювати власні анімації або використовувати Scratch для моделювання художніх концепцій. Це може допомогти їм розвивати творчі навички та виражати свою уяву через цифрове мистецтво;

d. наука: можна пропонувати учням створювати інтерактивні симуляції для пояснення наукових концепцій. Наприклад, можна створити анімації для пояснення законів фізики або біологічних процесів;

e. історія: учні можуть створювати історичні інтерактивні ігри або анімації, щоб вивчати події минулого. Це дозволяє їм зануритися в історичні контексти та розуміти події краще;

f. географія: можна використовувати Scratch для створення інтерактивних карт, де учні можуть вивчати географічні особливості різних регіонів та країн;

g. технології та ІТ: розробка програм та ігор у Scratch може допомогти учням краще розуміти базові концепції програмування, що може бути корисним для подальшого вивчення технічних предметів;

h. громадянська освіта: створення проєктів, які акцентують на глобальних проблемах або питаннях громадянської відповідальності, може сприяти формуванню в учнів глибокого розуміння соціальних питань та активного громадянства.

Інтеграція Scratch з іншими предметами дозволяє створити зв'язок між різними навчальними дисциплінами, розвиваючи при цьому різнобічні навички учнів.

3. Градація складності завдань:

a. початок з простих завдань: починайте з базових концепцій та простих завдань, щоб учні змогли ознайомитися із засобами Scratch. Наприклад, дозвольте їм створити прості анімації або рухомі об'єкти за допомогою блоків;

b. поступовий ріст складності: варто плавно переходити до складніших завдань по мірі зростання рівня навичок учнів. Наприклад, почніть з простого керування одним об'єктом і плавно переходьте до багатьох об'єктів, умов та циклів;

c. індивідуалізація навчання: слід враховувати індивідуальні рівні навичок та інтереси учнів. Для тих, хто швидко освоює базові концепції, можна пропонувати складніші завдання або проєкти;

d. диференціація завдань: дозвольте учням вибирати проєкти або завдання, які вони знаходять цікавими. Це допоможе створити індивідуалізований підхід до вивчення та розвитку;

e. підтримка з боку вчителя: вчителі можуть визначити оптимальний темп вивчення, враховуючи реакції учнів. Якщо виявляється, що група

стикається з труднощами, вчитель може надавати додаткову підтримку та пояснення;

f. постійний аналіз прогресу: здійснення постійного моніторингу прогресу учнів та адаптація методики вивчення відповідно до їхніх потреб. Це може включати коригування темпу навчання, надання додаткових завдань або розширення матеріалів.

Загальна ідея полягає в тому, щоб забезпечити гладкий та зрозумілий перехід від простих концепцій до складніших, розвиваючи при цьому навички програмування учнів та сприяючи їхній творчості.

4. Застосування реальних завдань. Важливо включати у навчальний процес проекти, які мають реальний або практичний зв'язок з реальним світом. Такі проекти дозволяють учням застосовувати свої знання в практичних ситуаціях і відчувати конкретний результат своєї праці. Декілька ідей для реальних проектів у вивченні Scratch в ліцеї:

a. створення ігор: учні можуть розробляти власні ігри на основі Scratch. Це може бути не лише цікавим завданням, але і можливістю вдосконалити навички програмування через створення інтерактивних та цікавих ігор;

b. інтерактивні оповідання: завданням може бути створення інтерактивних оповідань, де учні використовують Scratch для створення цікавих сценаріїв з різними персонажами та варіантами розвитку подій;

c. екологічні проекти: учні можуть розробляти проекти, пов'язані з екологією та природоохороною. Наприклад, створення ігор або анімацій, які навчають ефективному використанню ресурсів або пропагують екологічно чисті практики;

d. соціальні проекти: учні можуть працювати над проектами, що спрямовані на розв'язання соціальних проблем. Наприклад, створення ігор або мультфільмів, які підкреслюють важливість толерантності, рівності та інших соціальних цінностей;

е. мультимедійні презентації: учні можуть використовувати Scratch для створення мультимедійних презентацій на різноманітні теми. Це може допомогти розвинути навички не лише програмування, а й візуального дизайну та комунікації;

ф. співпраця зі спільнотою: учні можуть працювати над проектами, які мають взаємодію зі спільністю або розв'язують конкретні проблеми у школі чи місцевій громаді.

Залучення учнів до реальних проєктів допомагає їм бачити цінність своєї роботи та створювати щось корисне та значуще. Такий підхід також сприяє розвитку критичного мислення, творчості та навичок співпраці.

5. Робота в команді:

а. групові проєкти: учні можуть працювати над груповими проєктами, де кожен член групи відповідає за певний аспект проєкту. Наприклад, один учень може відповідати за створення персонажів, інший - за логіку гри, третій - за анімацію і тд;

б. обговорення та обмін ідеями: можна створити форму чи дошку для обговорення проєктів. Учні можуть ділитися своїми ідеями, надавати поради один одному та обговорювати важливі питання;

с. спільна проблема та її вирішення: можна задавати учням завдання, що потребують спільного розв'язання. Наприклад, створення гри, де декілька персонажів взаємодіють між собою. Це стимулює командну роботу та вирішення спільних завдань;

д. взаємна оцінка та фідбек: варто заохочувати учнів залишати коментарі та оцінювати роботу одне одного. Це допомагає виробити критичне мислення та розвиває навички конструктивного відгуку;

е. спільні події та презентації: варто організовувати такі події, на яких групи можуть презентувати свої проєкти. Це надає можливість поділитися досвідом, вивчити від інших та встановити конструктивний діалог;

f. менторство та підтримка: створення системи менторства, де досвідчені учні можуть допомагати менш досвідченим. Це стимулює взаємодопомогу та обмін знаннями.

Робота в команді дозволяє учням взаємодіяти, навчатися одне від одного та розвивати навички спільної роботи, що є важливими як у сфері програмування, так і в інших аспектах життя.

б. Формування творчого мислення. Цей підхід є ключовим у навчанні програмування в Scratch, оскільки важливо стимулювати учнів експериментувати та розширювати їхні творчі можливості, що можна впровадити за допомогою:

a. створення власних проєктів: можна заохочувати учнів створювати власні проєкти з нуля. Наприклад, вони можуть розпочати створення своєї гри, інтерактивної історії або анімації. Це дозволяє їм відчувати власну творчість та власний внесок;

b. застосування власних ідей: учні мають можливість висловити свої власні ідеї та концепції у своїх проєктах. Це може бути стосовно героїв, сюжету, стилістики чи будь-яких інших елементів;

c. експерименти з блоками: варто надавати учням можливість експериментувати з різними блоками та їхніми параметрами. Вони можуть змінювати швидкість руху об'єктів, розмір, кольори тощо. Це дозволяє їм розігрувати свої ідеї та бачити, як вони впливають на результат;

d. використання розширень та власних блоків: якщо навчати учнів використовувати розширення та створювати власні блоки, то це може дати їм можливість розширити функціонал Scratch та створювати складніші програми;

e. заохочення до різноманітності: важливо підкреслювати важливість різноманітності у творчому процесі. Учні можуть використовувати різні графічні, звукові та інші ресурси для створення унікальних та цікавих проєктів;

f. стимулювання самовираження: доцільним є заохочення учнів виражати свої ідеї та інтереси через свої проєкти. Це може бути засобом для них виразити свою унікальність та індивідуальність.

Розвиток творчості учнів під час вивчення Scratch важливий для того, щоб навчання програмуванню було цікавим та стимулюючим. Такий підхід сприяє розвитку креативних мислення та навичок проблемного вирішення.

7. Звітність та оцінювання результатів. Такий підхід може грати важливу роль у стимулюванні учнів та визначенні їхнього прогресу. Для забезпечення ефективної звітності та оцінювання можна застосовувати:

a. чіткі критерії оцінювання: може включати такі аспекти, як креативність, функціональність, використання різних програмних конструкцій тощо. Варто зробити ці критерії доступними учням заздалегідь;

b. самооцінка: залучення учнів до самооцінки їхніх проєктів дозволяє їм порівнювати свою роботу з визначеними критеріями та визначати свої сильні та слабкі сторони;

c. взаємна оцінка: учні можуть оцінювати проєкти своїх однокласників, це може створити стимул для взаємопокращення та взаємного вивчення;

d. використання портфоліо: учні можуть створювати портфоліо своїх проєктів. Це не лише дасть можливість відстежувати їхній прогрес, але і буде корисним при представленні своїх навичок у майбутньому;

e. стимулювання креативності: надання додаткових балів або «відзнак» за оригінальність та креативність може підтримати учнів у виявленні власного стилю та підходу до програмування.

Забезпечення ефективної звітності та оцінювання допоможе не лише визначити рівень засвоєння матеріалу учнями, але й створить стимули для їхнього подальшого розвитку в області програмування.

8. Підтримка зовнішніх ресурсів. Це може значно розширити можливості вивчення Scratch у ліцеї та зробити процес навчання цікавішим та різноманітним. Для цього можна використати відеоуроки та онлайн-курси:

залучення відеоуроків та онлайн-курсів для вивчення Scratch на платформах, таких як YouTube або Khan Academy, що може бути корисним для учнів.

Враховуючи ці методичні аспекти, вивчення Scratch у ліцеї може бути цікавим та ефективним для учнів, розвиваючи їхні навички програмування та стимулюючи творчий підхід до вирішення завдань.

2.2. Опис методичного комплексу для підтримки факультативного курсу з Scratch для профільного навчання

Для оформлення навчально методичного комплексу факультативу для ліцеїстів зі Scratch нами було обрано формат сайту, оскільки такий варіант дає можливість використовувати матеріали факультативу будь-якій людині з будь-яким пристроєм, звісно за умови підключення до глобальної мережі Інтернет. Було розглянуто безліч варіантів створення сайту – за допомогою конструкторів сайтів (Wix, UKit, UMI, GoogleSites), за допомогою CMS-систем (WordPress, Joomla, Drupal, OpenCart, phpBB) та шляхом самостійного написання вихідного коду сайту [41]. Свій вибір віддали сервісу конструкторів сайтів GoogleSites і на це є декілька причин:

- використання додатків Google для організації освітнього процесу переважної більшості гімназій та ліцеїв;
- вже наявний достатній досвід створення сайтів за допомогою цього конструктора, тому жодних труднощів виникати не може;
- Google Sites синхронізуються з акаунтом Google і всі дії, що виконуються для заповнення сайту зберігаються автоматично і нічого не загубиться;
- можна надати доступ до редагування сайту довіреним особам та своїм іншим акаунтам, це забезпечить можливість роботи навіть при втраті доступу до акаунту Google;
- також важливою причиною є безкоштовність цього конструктора – можна безкоштовно створювати необмежену кількість сайтів на 1000 сторінок [42].

Таким чином створили сайт за готовим шаблоном і підлаштували дизайн під потреби факультативу, а не писали самостійно код сайту за допомогою того ж самого HTML та CSS, що забрало б дуже багато часу.

Подальша робота полягала у створенні завдань факультативу та опису їх рішень. Інструкції створили в окремому текстовому документі, з якого вже матеріали переносили на сайт. Такий варіант дає можливість подивитися обсяг завдань та вже мати уявлення про те скільки сторінок на сайті потрібно додати, як розмістити текст та картинки, в якому порядку давати завдання – тут краще використовувати метод градації складності завдань – чим далі проходить завдання, тим складніше вони стають.

Для зручності розділили завдання на різні категорії – «Знайомство з Scratch», «Створення ігор», «Scratch+робототехніка», «Креативне програмування» та «STEM-напрямок». Для педагогічного експерименту було створено та описано блок «Знайомство з Scratch» з теоретичним матеріалом про Scratch для тих, хто не працював раніше в цьому середовищі. Також створено та описано 6 завдань блоку «Створення ігор», і поки проходив експеримент корегували завдання на основі сприйняття відповідної інформації експериментальною групою, також поки проходив експеримент були дописані інші завдання блоку з іграми та всі інші блоки завдань.

Для візуалізації можна додати приклад як завдання факультативу описані в текстовому документі й на сайті, розглянемо завдання «Створення меню гри» блоку «Створення ігор».

У текстовому документі описуємо інструкцію до заняття:

Створимо меню з 3 кнопками – «грати», «вибір персонажу» та «назад».

Створюємо новий проєкт, називаємо його «меню», прибираємо зайві спрайти та обираємо тло для головного меню (рис.2.1).

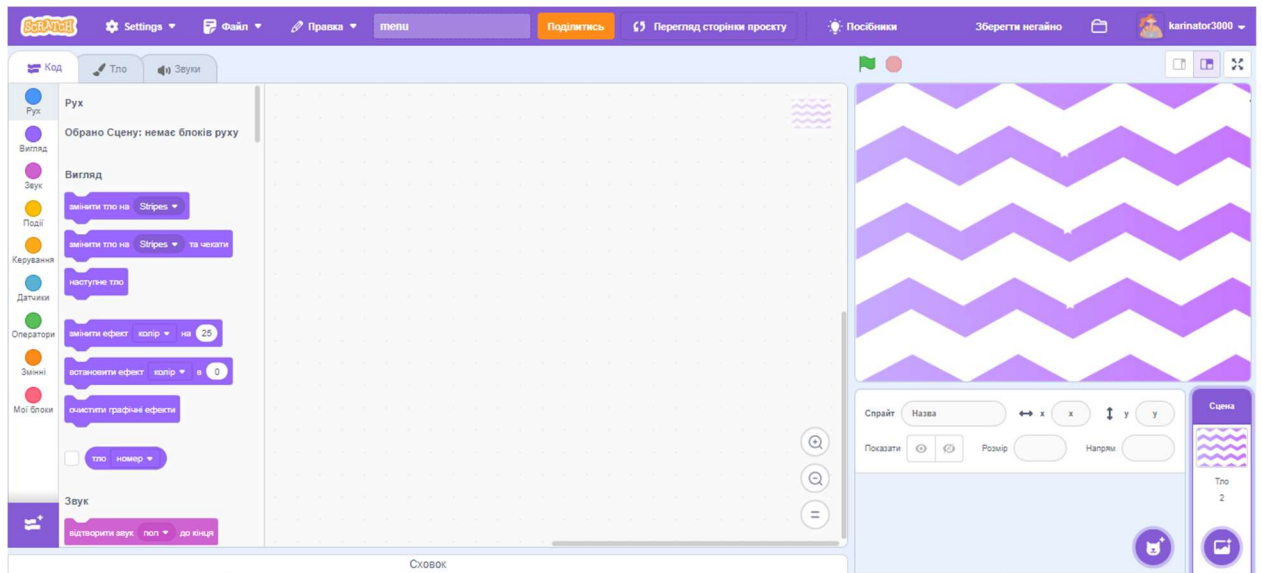


Рис. 2.1. Створення нового проєкту

Додаємо 2 спрайти Button2, називаємо їх «Грати» та «Вибір персонажа», змінюємо колір та розмір кнопок, додаємо на них відповідний текст, розташовуємо на сцені (рис. 2.2).

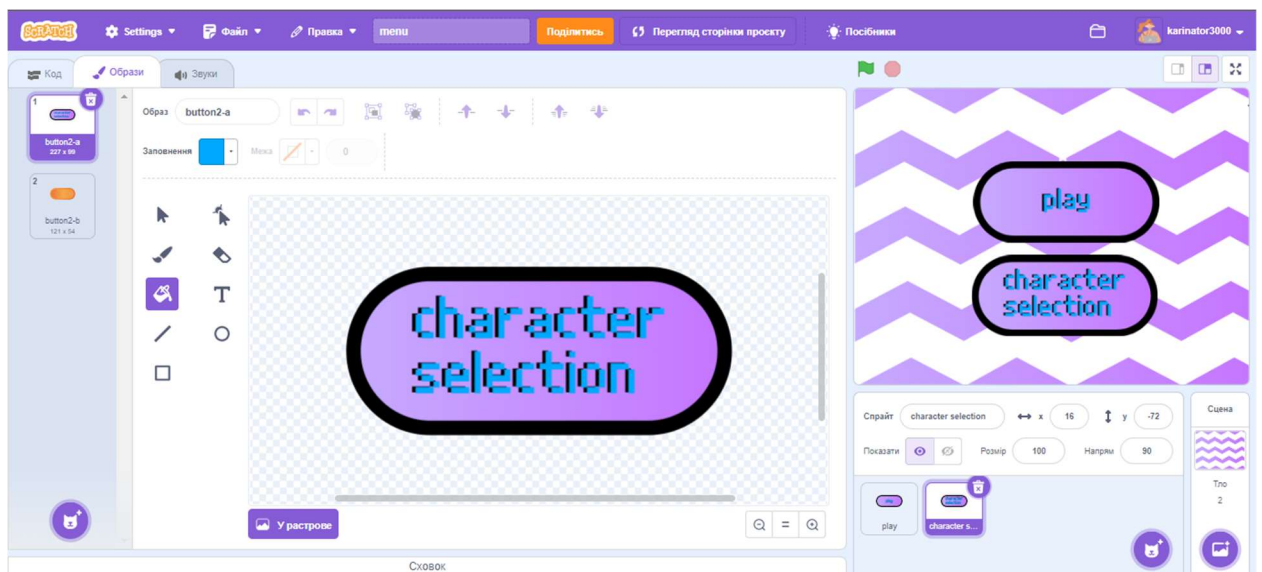


Рис. 2.2. Створення кнопок «Грати» та «Вибір персонажу»

Починаємо писати код для кнопок. Основний функціонал – при натисканні на кнопку перенести в іншу частину гри. Кнопка «грати» буде кнопкою для запуску самої гри, а кнопка з вибором персонажа – дає можливість обрати персонажа. При переході в інше меню кнопки мають зникнути, а при поверненні назад – з’явитися.

Вигляд коду для кнопки «грати» зображений на рис. 2.3.

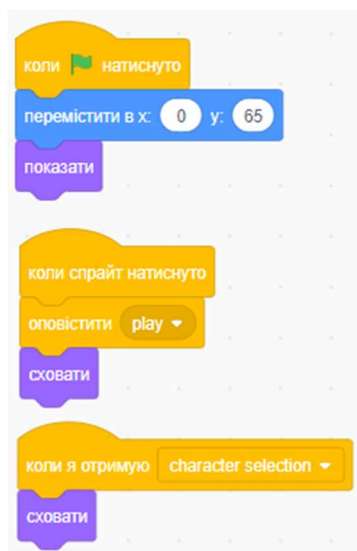


Рис. 2.3. Код для кнопки «Грати»

Код для кнопки «вибір персонажа» продемонстрований на рис. 2.4.

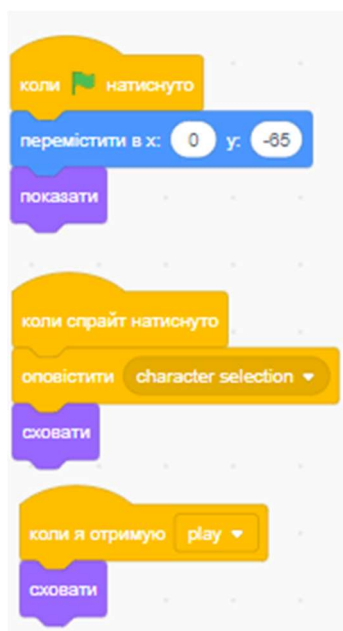


Рис. 2.4. Код для кнопки «Вибір персонажу»

Також можна додати плавну появу (emergence) та зникання (disappearance) кнопок (рис. 2.5).

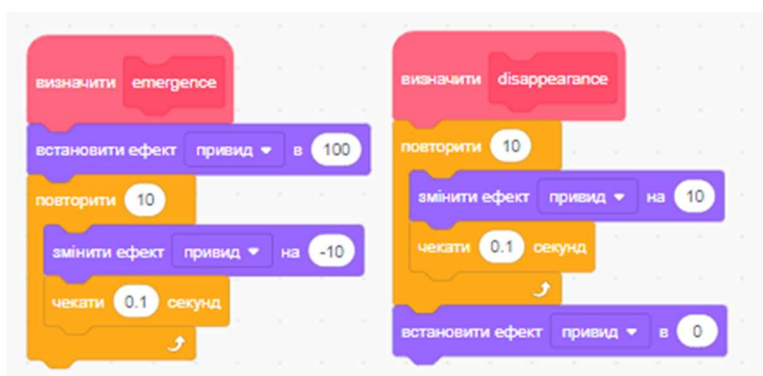


Рис. 2.5. Код для плавної появи та зникнення кнопок

Ці блоки додаємо перед «показати» та «сховати» в обох кнопках.

Перейдемо до опрацювання кнопки «вибір персонажа».

Додаємо 2 нових тла (для гри та для меню вибору персонажа) та трьох спрайтів, які і будуть персонажами. Рекомендація: розмір персонажів має бути приблизно однаковий.

Також додаємо кнопку «назад» (рис. 2.6).

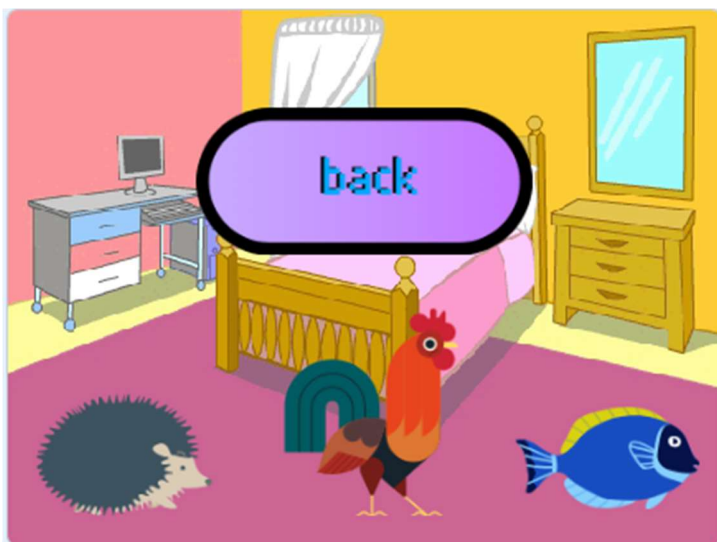


Рис. 2.6. Додавання кнопки «Назад»

Функціонал спрайтів-персонажів: при натисканні кнопки «вибір персонажа» з'явитися, при наведенні на спрайт – збільшитись, при натисканні кнопки «назад» – зникнути.

Код для всіх персонажів (рис. 2.7):

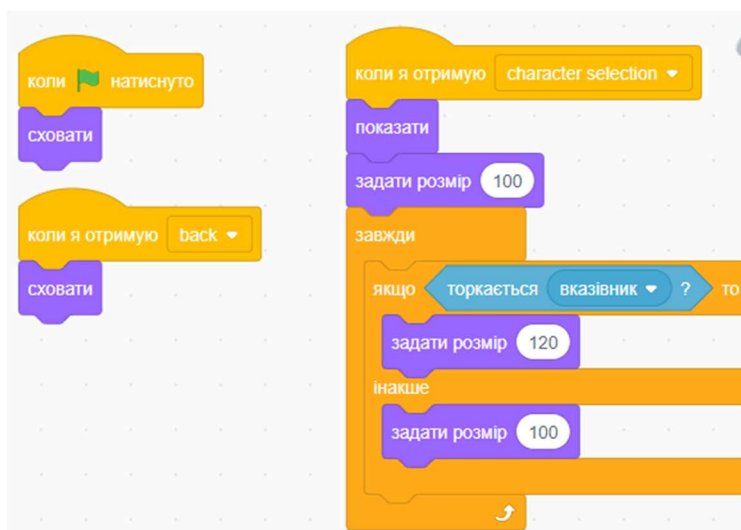


Рис. 2.7. Код для спрайтів-персонажів

Також додаємо кожному персонажу код, який працює таким чином: якщо натиснуто на спрайт, то оповістити що натиснуто саме на цей спрайт.

Код для Їжачка показано на рис. 2.8.

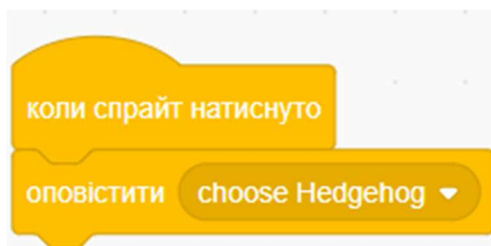


Рис. 2.8. Код для спрайту «Їжак»

Код для Півня показано на рис. 2.9.

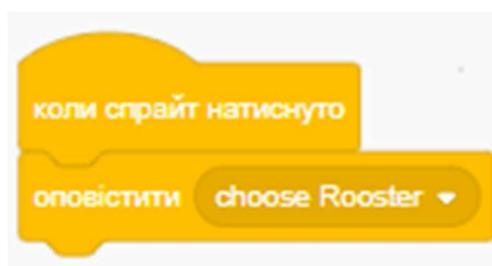


Рис. 2.9. Код для спрайту «Півень»

Код для Риби показано на рис. 2.10.

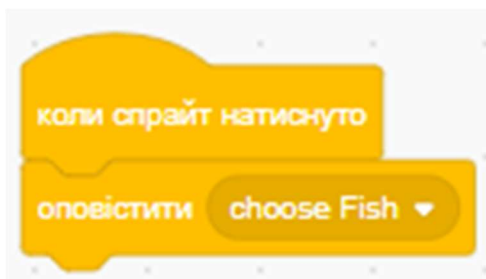


Рис. 2.10. Код для спрайту «Риба»

Функціонал кнопки «назад» майже не відрізняється від кнопки «меню» – тільки замість появи ставимо зникання та навпаки (рис. 2.11).



Рис. 2.11. Код для кнопки «Назад»

Оскільки ми додали кнопку «назад», яка має повертати попереднє меню, то варто дописати код для інших кнопок – при отриманні повідомлення «назад» – з’явитись (рис. 2.12).

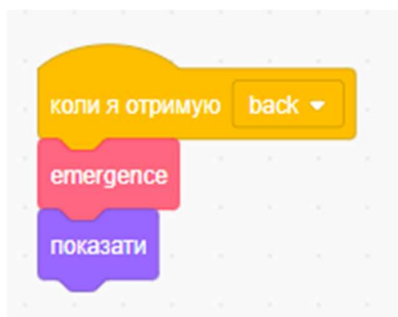


Рис. 2.12. Код для кнопок з поверненням їх у меню

Залишилось додати код на тло. При натисканні на прапорець та кнопку «назад» має з’являтися тло для меню, а при натисканні на кнопку «вибір персонажа» – інше тло. Для зручності кожне тло краще підписати його функцією (рис. 2.13).

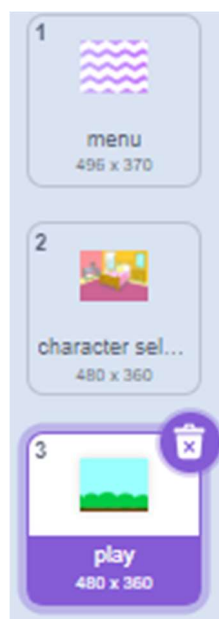


Рис. 2.13. Приклад назв фонів

Вигляд коду зображений на рис.2.14.

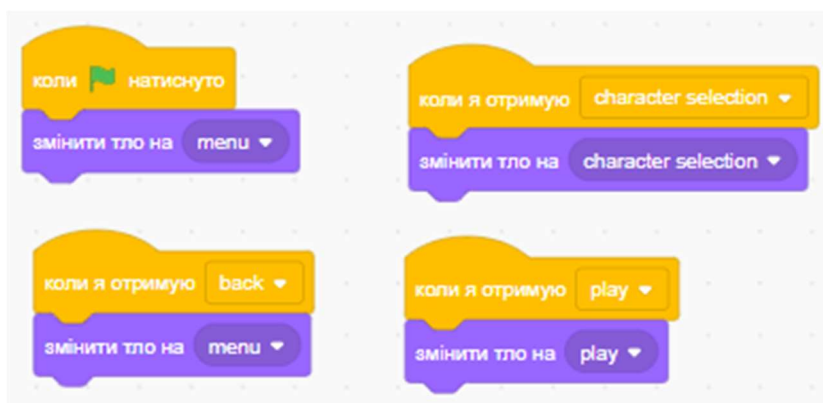
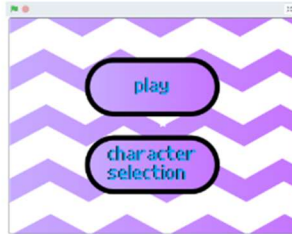


Рис. 2.14. Код для тла

На цьому етапі перевіряємо працездатність кнопок та зберігаємо проєкт.

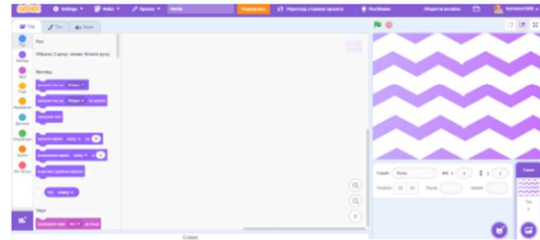
Вигляд цього ж заняття, розміщеного на сайті (рис. 2.15–2.19):

Меню гри



Переглядаємо приклад та створюємо власне меню.

Створимо меню з 3 кнопками – «Грати», «Вибір персонажу» та «Назад».
Створимо новий проєкт, назвемо його «меню», приберемо зайві спрайти та обираємо тло для головного меню.



Додаємо 2 спрайти Button2, назвемо їх «Грати» та «Вибір персонажа», змінюємо колір та розмір кнопок, додаємо на них відповідний текст, розташовуємо на сцені.

Почнемо писати код для кнопок. Основний функціонал – при натисканні на кнопку перейти в іншу частину гри. Кнопка «Грати» буде кнопкою для запуску самої гри, а кнопка з вибором персонажа – дає можливість обрати персонажа. При переході в інше меню кнопки мають зникнути, а при поверненні назад – з'явитися.
Код для кнопок матиме такий вигляд:

Рис. 2.15. Верхня частина сайту з початком створення проєкту

Головна сторінка Зміст

Код кнопки "грати"

```

колі натиснуто
  перемістити в х: 0 у: 65
  показати

колі спрайт натиснуто
  оповістити play
  сховати

колі я отримую character selection
  сховати

```

Код кнопки "вибір персонажа"

```

колі натиснуто
  перемістити в х: 0 у: -65
  показати

колі спрайт натиснуто
  оповістити character selection
  сховати

колі я отримую play
  сховати

```

Також можна додати плаву позу (emergence) та зникання (disappearance) злюпок:

```

колі натиснуто
  оповістити emergence
  оповістити disappearance
  перемістити в х: 0 у: 65
  перемістити в х: 0 у: -65
  зникнути
  зникнути

```

Ці блоки додаємо перед «показати» та «сховати» в обох кнопках.

Перейдемо до опрацювання кнопки «вибір персонажа».

Додаємо 2 нових тла (для гри та для меню вибору персонажа) та трьох спрайтів, які і будуть персонажами. Рекомендація: розмір персонажів має бути приблизно однаковий. Також додаємо кнопку «назад».

Функціонал спрайтів-персонажів: при натисканні кнопки «вибір персонажа» з'являється, при наведенні на спрайт – збільшується, при натисканні кнопки «назад» – зникає.

Код для всіх персонажів:

```

колі натиснуто
  сховати

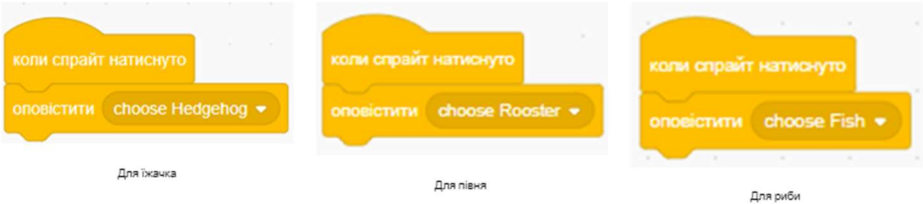
колі я отримую back
  сховати

колі я отримую character selection
  показати
  задати розмір 100
  завжди
    якщо торкається вказівник ? то
      задати розмір 120
    інакше
      задати розмір 100

```

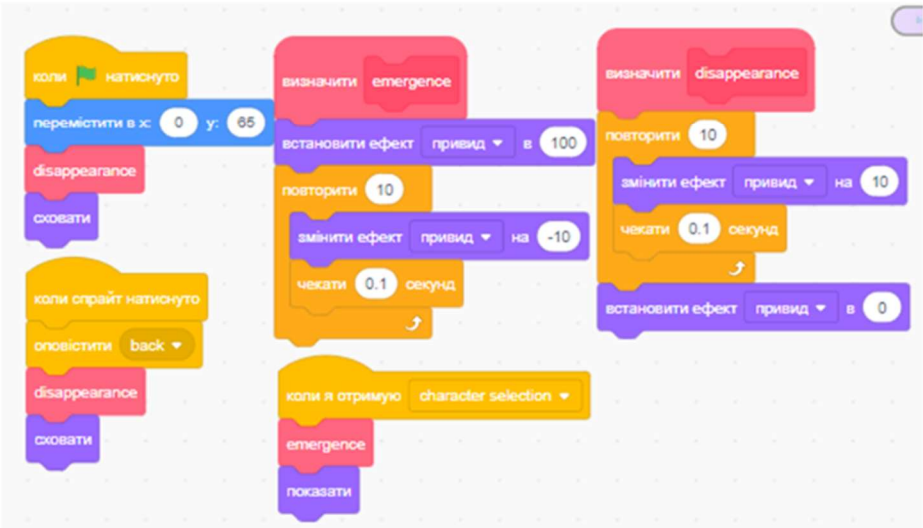
Рис. 2.16. Продовження сторінки сайту з створенням проєкту

Також додаємо кожному персонажу код, який працює таким чином: якщо натиснуто на спрайт, то оповістити що натиснуто саме на цей спрайт.



Для їжачка
Для півня
Для риби

Функціонал кнопки «назад» майже не відрізняється від кнопки «меню» - тільки замість появи сталею зникання та назваки.



Оскільки ми додали кнопку «назад», яка має повертати попереднє меню, то варто дописати код для інших кнопок - при отриманні повідомлення «назад» - з'явитися:

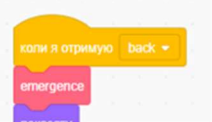


Рис. 2.17. Продовження сторінки сайту з створенням проєкту

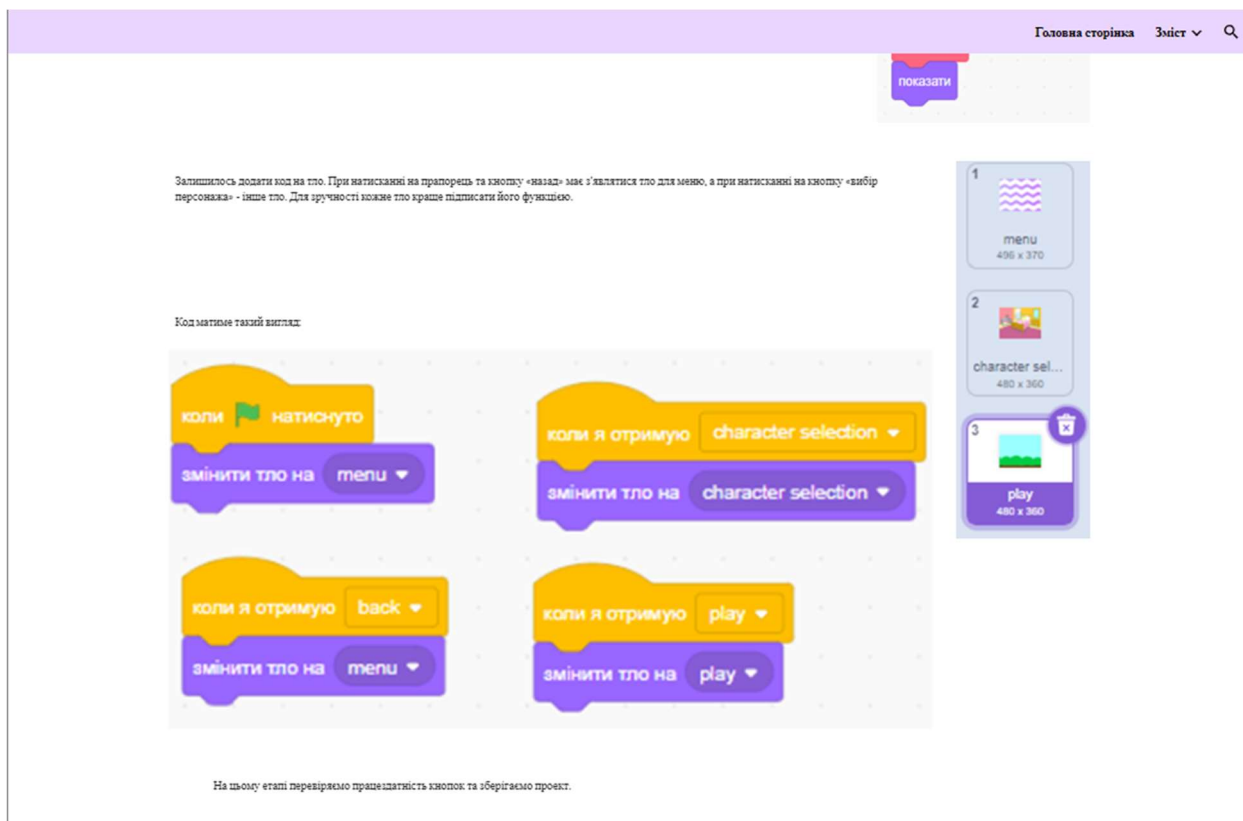


Рис. 2.18. Продовження сторінки сайту з завершенням проекту

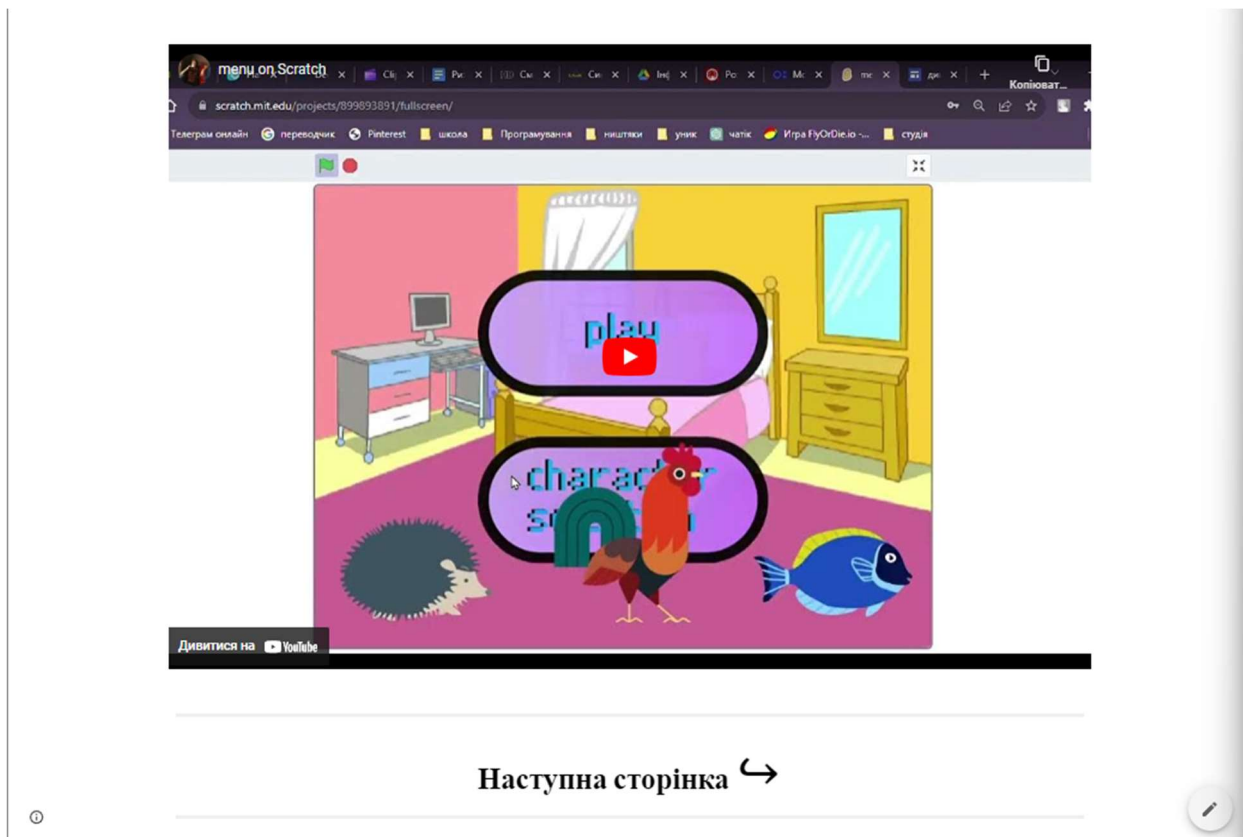


Рис. 2.19. Кінець сторінки сайту

2.3. Аналіз результатів педагогічного експерименту

Експеримент почався ще навесні 2023 року – учням 9-А та 9-Б класу Криворізької гімназії №114 Криворізької міської ради [22] було запропоновано взяти участь у фестивалі «STEAM FEST» від відділу STEM-освіти Київського Палацу дітей та юнацтва у конкурсі «Scratch Game» (створення ігор в Scratch), для цього було сформовано групу та проведено перші пробні факультативні заняття на яких команда опрацювала в розділі факультативу «Створення ігор» перший та другий урок – «Створення меню гри та створення опції вибору персонажу в гри». Також діти самостійно придумали гру «Ловець квітів» та написали для цієї гри код. Результатом роботи команди став проєкт «Летюча фея» що зайняв перше місце, діти отримали сертифікати, подарунки (рис. 2.20–2.21) та корисний досвід. Ці ж учні й увійшли у склад експериментальної групи як ліцеїсти.



Рис. 2.20. Призи за перемогу у фестивалі «STEAM FEST»



Рис. 2.21. Сертифікати за перемогу у фестивалі «STEAM FEST»

З першого вересня 2023 року, коли учні перейшли до ліцею, ми почали активне проведення експерименту – сформували 2 групи по 10 учнів – група 10 класу та група 11 класу. Більшість дітей навчаються в Криворізькому ліцеї №127 Криворізької міської ради, але частина учнів виїхала в інші країни, тому було вирішено проводити заняття факультативу дистанційно.

Експеримент проходив у 3 етапи – констатувальний, формувальний та контрольний.

Констатувальний етап спрямований на визначення початкового рівня навичок програмування в Scratch та зацікавленості в самому експерименті, для визначення яких учні пройшли опитування, результати якого зазначено в табл. 2.1.

Таблиця. 2.1

Результати констатувального етапу експерименту

	Учень	Клас	Чи вивчали Scratch у школі?	Чи використовували Scratch для створення ігор/проєктів?	Чи брали участь в конкурсах з програмування Scratch?	Чи хотіли б вивчати Scratch в 10/11 класі?	Чи знаходили ресурси для самостійного вивчення Scratch?	Оцініть свій рівень програмування в Scratch від 0 до 10
1	Алхімов Данііл	10	Так	так	так	так	ні	7
2	Укібаєв Кирило	10	Так	ні	ні	так	ні	8
3	Маринець Євгенія	10	Так	так	так	так	так	6
4	Лисенко Ілля	10	Так	ні	ні	так	ні	5
5	Борисюк Кароліна	10	Так	ні	ні	так	ні	5
6	Балясна Вікторія	10	Так	так	так	так	ні	7
7	Карась Дар'я	10	Так	так	так	так	так	8
8	Джагарян Кирило	10	Так	ні	ні	так	ні	10
9	Зима Даниїл	10	Так	ні	ні	так	ні	6
10	Данилова Лана	10	Так	так	так	так	ні	7

	Учень	Клас	Чи вивчали Scratch у школі?	Чи використовували Scratch для створення ігор/проектів?	Чи брали участь в конкурсах з програмування Scratch?	Чи хотіли б вивчати Scratch в 10/11 класі?	Чи знаходили ресурси для самостійного вивчення Scratch?	Оцініть свій рівень програмування в Scratch від 0 до 10
11	Чупрій Єлизавета	11	Так	так	ні	так	так	6
12	Манжос Катерина	11	Так	ні	ні	так	ні	7
13	Коржук Анастасія	11	Так	ні	ні	так	так	5
14	Федусов Єгор	11	Так	ні	ні	так	ні	5
15	Рибалка Павло	11	Так	ні	ні	так	ні	4
16	Бойко Єва	11	Ні	ні	ні	так	ні	5
17	Міхеев Андрій	11	Так	так	так	так	так	8
18	Кравцова Поліна	11	Ні	ні	ні	так	ні	4
19	Романюк Данііл	11	Так	так	так	так	ні	7
20	Жмакіна Марія	11	Так	ні	ні	так	ні	5

Дані цієї таблиці візуалізовані за допомогою стовпчастої діаграми (рис. 2.22).

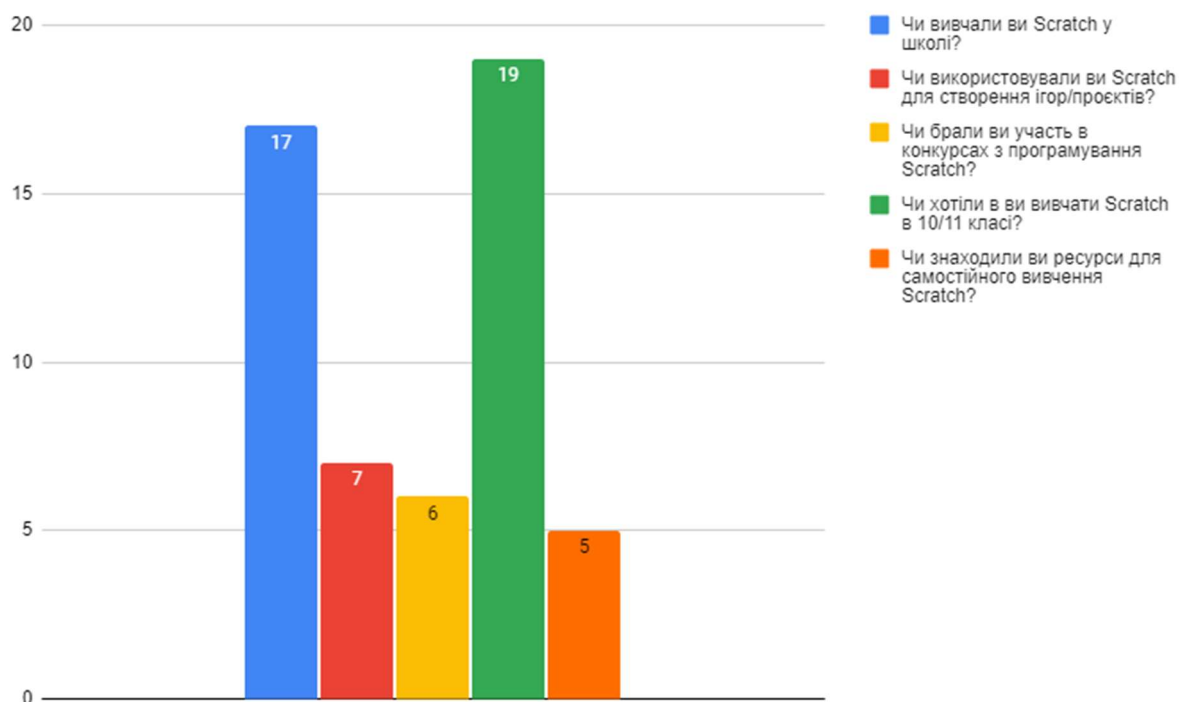


Рис. 2.22. Результати опитування учасників експерименту

Також діти оцінювали свої наявні навички програмування в Scratch за шкалою від 0 до 10 (рис. 2.23).

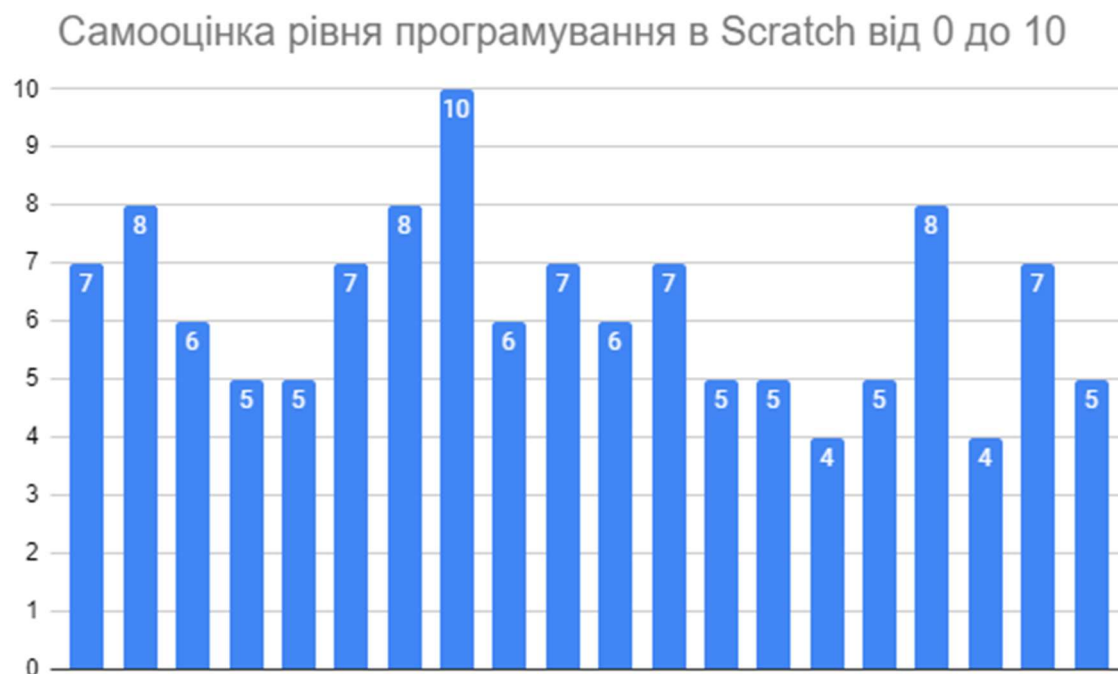


Рис. 2.23. Результат оцінювання власних навичок програмування в Scratch учасниками експерименту

Після констатувального етапу ми перейшли до формувального – цей етап спрямований на розвиток і вдосконалення навичок програмування в Scratch. Зустрічалися раз на тиждень, у п'ятницю, заняття тривали 60 хвилин – з 16:30 до 17:30, звісно, іноді ми відставали від запланованого часу на 5-10 хвилин. Також на тривалість заняття впливала і повітряна тривога в місті Кривий Ріг, адже під час повітряної тривоги ми переходили в укриття. Заняття 22 вересня ми перервали о 17:13, 27 жовтня – о 17:18, а 10 листопада заняття було повністю скасовано, 17 листопада ми закінчили заняття о 17:12. Протягом цих 2,5 місяці ми провели 11 занять на яких опрацювали 6 завдань факультативу та останнє заняття було присвячено самостійному створенню власної гри з використанням набутих знань та навичок. Звісно, якби заняття тривали не 60 хвилин, а 90, то ми б одне завдання могли опрацювати за одне заняття, але учні не мали багато вільного часу, тому проводили заняття саме з такою тривалістю, яка є – 60-70 хвилин.

По завершенню експерименту учні як результат своєї роботи мали власний проєкт-гру, основою якої є меню з вибором персонажу. Всі проєкти різні – в них залучені різноманітні спрайти, сцени та звуки, також саму гру учень обирав та писав код до неї сам. Контрольним етапом, що спрямований на оцінку результатів експерименту, було опитування, в якому діти оцінювали завдання факультативу та свої набуті навички (табл. 2.2).

Завдяки цьому опитуванню можна зробити висновок про те, наскільки збільшився рівень володіння Scratch в учнів, це від 0 до 5 рівню (рис. 2.24).

Оцінка проєктних реалізацій учнів (згідно розроблених критерії оцінки проєкту (табл. 2.3)) представлена у таблиці 2.4.

На рисунку 2.25 можна побачити різницю між самооцінкою набутих навичок програмування та фактичною оцінкою їх проєктів за критеріями оцінювання.

Також деякі учні залишили відгук на цей факультатив з враженнями та побажаннями (табл. 2.5) (орфографія та пунктуація авторів збережена).

Таблиця 2.2

Результати завершального опитування учасників експерименту

	Учень	Клас	Оцініть завдання факультативу з Scratch за шкалою від 0 до 10	Чи порекомендували б Ви цей факультатив своїм друзям або однокласникам як ресурс для вивчення програмування?	Чи плануєте Ви використовувати факультатив надалі? (опрацювати інші розділи)	Чи плануєте Ви брати участь в конкурсах з програмування Scratch використовуючи знання, отримані при вивченні факультативу?	Оцініть свій теперішній рівень володіння Scratch (від 0 до 10)
1	Алхімов Данііл	10	8	вагаюсь з відповіддю	так	вагаюсь з відповіддю	10
2	Укібаєв Кирило	10	9	так	так	вагаюсь з відповіддю	10
3	Маринець Євгенія	10	7	вагаюсь з відповіддю	так	так	10
4	Лисенко Ілля	10	7	так	вагаюсь з відповіддю	вагаюсь з відповіддю	7
5	Борисюк Кароліна	10	8	так	вагаюсь з відповіддю	ні	8
6	Балясна Вікторія	10	10	так	так	так	10
7	Карась Дар'я	10	9	так	вагаюсь з відповіддю	так	9
8	Джагарян Кирило	10	8	вагаюсь з відповіддю	ні	ні	10
9	Зима Даниїл	10	8	ні	ні	ні	7

	Учень	Клас	Оцініть завдання факультативу з Scratch за шкалою від 0 до 10	Чи порекомендували б Ви цей факультатив своїм друзям або однокласникам як ресурс для вивчення програмування?	Чи плануєте Ви використовувати факультатив надалі? (опрацювати інші розділи)	Чи плануєте Ви брати участь в конкурсах з програмування Scratch використовуючи знання, отримані при вивченні факультативу?	Оцініть свій теперішній рівень володіння Scratch (від 0 до 10)
10	Данилова Лана	10	9	вагаюсь з відповіддю	так	так	10
11	Чупрій Єлизавета	11	8	ні	вагаюсь з відповіддю	так	9
12	Манжос Катерина	11	10	так	вагаюсь з відповіддю	вагаюсь з відповіддю	9
13	Коржук Анастасія	11	10	так	так	так	10
14	Федусов Єгор	11	8	вагаюсь з відповіддю	вагаюсь з відповіддю	вагаюсь з відповіддю	7
15	Рибалка Павло	11	8	так	так	ні	8
16	Бойко Єва	11	8	вагаюсь з відповіддю	вагаюсь з відповіддю	вагаюсь з відповіддю	8
17	Міхеєв Андрій	11	10	так	так	так	9
18	Кравцова Поліна	11	10	так	так	вагаюсь з відповіддю	7
19	Романюк Данііл	11	8	вагаюсь з відповіддю	так	ні	10
20	Жмакіна Марія	11	8	так	вагаюсь з відповіддю	вагаюсь з відповіддю	7

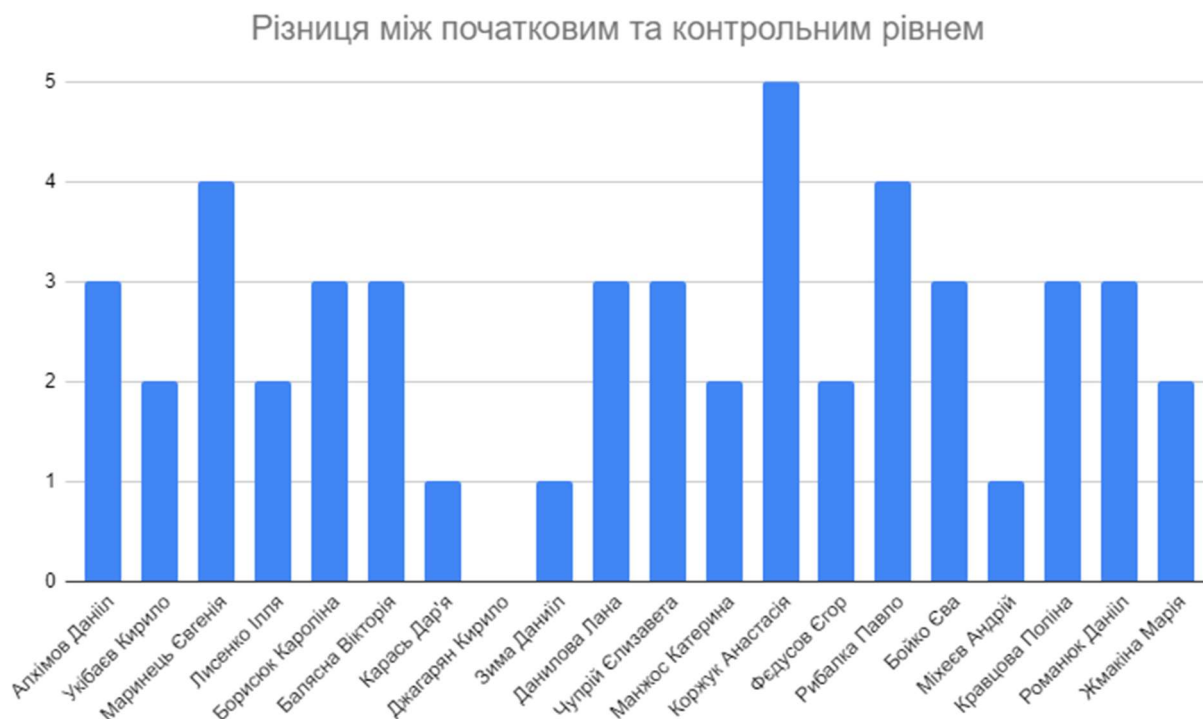


Рис. 2.24. Результат збільшення рівня володіння Scratch в учасників експерименту

Таблиця 2.3

Критерії оцінювання проєктів

Критерій	Опис критерію
Відповідність завданню	проєкт отримує бал якщо основна ідея проєкту відповідає мінімум на 80% завданню
Завершеність	бал отримує проєкт, якщо його можна переглянути і програти не менше, ніж за 1 хвилину, і не більше, ніж за 5-7 хвилин, якщо є зрозумілий початок і зрозуміле завершення
Майстерність	проєкт отримує бал якщо в ньому є всі потрібні кнопки та персонажі, продуманий дизайн
Творчий підхід	бал отримує проєкт, якщо використано нові та унікальні спрайти, фони та звуки

Оптимальність коду	проект отримує бал за виправдане, оптимальне та доречне використання програмних конструкцій, відсутність штучних ускладнень коду та виконуваних команд і скриптів
Культура коду	бал отримує проект за відсутність невикористовуваних блоків або незакінчених скриптів, іменування змінних, яке "говорить" і тд

Таблиця 2.4

Оцінювання проєктних реалізацій учнів

	Учень	Критерій оцінювання проєкту						Сума балів
		Відповідність завданню	Завершеність	Майстерність	Творчий підхід	Оптимальність коду	Культура коду	
1	Алхімов Даниїл	1	1	1	1	1	1	6
2	Укібаєв Кирило	1	1	1	1	1	1	6
3	Маринець Євгенія	1	1	1	1	1	1	6
4	Лисенко Ілля	1	0	0	1	1	1	4
5	Борисюк Кароліна	1	0	1	1	1	1	5
6	Балясна Вікторія	1	1	1	1	1	1	6
7	Карась Дар'я	1	1	1	1	1	0	5
8	Джагарян Кирило	1	1	1	1	1	1	6
9	Зима Даниїл	1	0	0	1	1	0	3

	Учень	Критерій оцінювання проєкту						Сума балів
		Відповідність завданню	Завершеність	Майстерність	Творчий підхід	Оптимальність коду	Культура коду	
10	Данилова Лана	1	1	1	1	1	1	6
11	Чупрій Єлизавета	1	1	1	1	1	0	5
12	Манжос Катерина	1	0	1	1	1	1	5
13	Коржук Анастасія	1	1	1	1	1	1	6
14	Федусов Єгор	1	0	0	1	1	1	4
15	Рибалка Павло	1	1	1	1	1	0	5
16	Бойко Єва	1	1	1	1	0	1	5
17	Міхєєв Андрій	1	1	1	1	1	1	6
18	Кравцова Поліна	1	0	0	1	1	0	3
19	Романюк Данііл	1	1	1	1	1	0	5
20	Жмакіна Марія	1	0	0	1	1	0	3

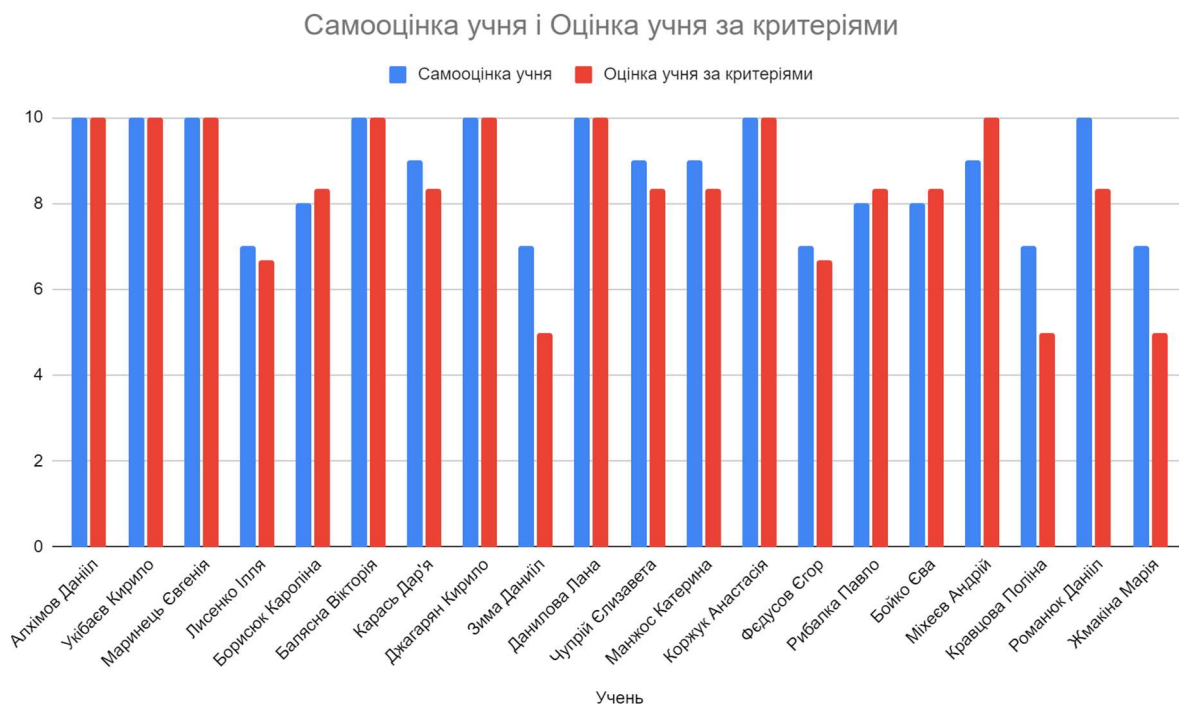


Рис. 2.25. Різниця між самооцінкою учнями набутих навичків та фактичною оцінкою їх проєктів

Таблиця 2.5

Відгуки про факультатив учасників експерименту

	Учень	Відгук
1	Алхімов Даниїл	Цікаві завдання, зрозуміло описані. Не вистачає окремих уроків по алгоритмам та подіям.
2	Укібаєв Кирило	Каріно Олексіївно, Ваша робота велика, прям видно що вкладено багато сил. Я б порекомендував додати більше розробок по культовим іграм – марію, галага, якісь файтінги. Дякую
3	Маринець Євгенія	Учениця не надала свій відгук в опитуванні.

	Учень	Відгук
4	Лисенко Ілля	Шкода що не все зміг зробити (та є ще проєкти з іграми). Каріна Олексіївна, не кидайте цю роботу, вона заслуговує на існування.
5	Борисюк Кароліна	Факультатив гарно оформлений, завдання зрозумілі
6	Балясна Вікторія	Чудовий ресурс для саморозвитку, дякую
7	Карась Дар'я	Є багато ідей для наступних ігор, є куди розвиватися)
8	Джагарян Кирило	В мене великий досвід в програмуванні, тому завдання для мене не були важкими або незрозумілими.
9	Зима Даніїл	Учень не надав свій відгук в опитуванні.
10	Данилова Лана	Курс хороший, інструкція – бомба, хотілось би побільше якогось візуалу робити в іграх.
11	Чупрій Єлизавета	Я не знала що скретч настільки простий в використанні, коли раніше його вивчала в школі він казався нудним, але створення ігор досить непоганий та приємний досвід

	Учень	Відгук
12	Манжос Катерина	Прикольні вправи, зрозуміло описано, є посилання на готові проекти і можна подивитись одразу що вийде в ітоге. Бажаю подальшого розвитку, цей курс підійде як старшим, так і молодшим учням.
13	Коржук Анастасія	Неперевершено! Меншого і не очікувала!
14	Федусов Єгор	Ну, наче норм. Не дуже люблю скретч, але ігри цікаві та прикольні. не знаю що можна додати
15	Рибалка Павло	Топ, замість уроків інформатики було б класно
16	Бойко Єва	Учениця не надала свій відгук в опитуванні.
17	Міхєєв Андрій	Чекаю розширення. зробіть побільше завдань
18	Кравцова Поліна	Я ще з 5 чи 6 класу не використовувала скретч, навіть не думала що можна створювати там ігри. ДЯК за таку можливість побачити програмування з іншого боку!
19	Романюк Данііл	норм, зашло
20	Жмакіна Марія	Мало часу працювали, завдання цікаві, але хотілось щоб прям замість уроків інформатики були

Отже, можна зробити висновки, що робота зі Scratch є цікавою та простою, розвиває в дітях логічне мислення, здатність творити, та формує нові навички. Як можна побачити з результатів педагогічного експерименту, інтерес до середовища серед молоді дуже значний, але через недостатню підготовку вчителів та невелику кількість занять з програмування у ліцеях не всі можуть користуватись ним та реалізовувати власні творчі ідеї та проєкти.

У подальшому є доцільним працювати в даному напрямку, а саме у напрямку вдосконалення якості викладання інформатики, впровадження нових форм подання матеріалу та створення нових цікавих блоків з програмування в середовищі Scratch.

2.4. Рекомендації щодо використання навчально-методичного комплексу у навчальному процесі

На основі педагогічного експерименту можна сформулювати методичні рекомендації щодо використання методичного комплексу у навчальному процесі:

- Форма занять – майже всі блоки факультативу підходять для будь-якої форми проведення занять – очної, дистанційної або змішаної. Виключенням є блок «Scratch+робототехніка» – його можна використовувати за наявності конструктора Lego Wedo 2.0. Такі заняття зазвичай проходять в спеціалізованих школах або приватних студіях, адже робототехніка стрімко вривається в шкільну систему, тому цей блок можна використовувати там, де є курси робототехніки;

- Тривалість занять – для опрацювання одного завдання в середньому потрібно 80 хвилин, тому буде доцільно встановлювати тривалість заняття півтори години. За відведений час діти встигнуть виконати й складне завдання, а якщо завдання було легшим і на нього витратили менше часу – можна дати додаткове завдання для зберігання тривалості заняття та розвитку креативності ті логічного мислення, навичок самостійно вирішувати завдання в учнів;

– При наявності вільного часу не давати дітям байдикувати, а давати додаткові завдання. Це може бути додавання фонові музики для проєкту, анімації для кнопок та плавні переходи і тд;

– Не стримувати дитячу уяву – якщо учень має своє уявлення про реалізацію певного проєкту – дайте йому можливість це зробити. ВИ нічого не втратите, а дитина набереться досвіду, навіть якщо нічого не вдасться.

Окремо можна зазначити рекомендації з оцінки проєктних реалізацій учнів, зручно оформлювати у форматі бальної системи (Див. табл. 2.3).

Також рекомендуємо використовувати додаткові ресурси для вивчення Scratch, якими можна надихатися та користуватися, такі як:

– Канал YouTube «Дистанційне навчання» [16]. Цей ресурс має 32 уроки з програмування алгоритмів в Scratch і може знадобитися для вивчення алгоритмічного програмування та створення алгоритмів як і в рамках шкільної програми, так і в позаурочній діяльності.

– Сайт «Вивчаємо Scratch разом» [13]. Розроблений кафедрою математики та інформатики і методики навчання ДВНЗ «Прикарпатський національний університет імені Василя Стефаника», має 6 блоків уроків з програмування та розробки ігор в Scratch.

– Ресурс SCRATCH. ПРОГРАМУЄМО ІГРИ [26] . Автор цього збірника – Сергій Курінний, описав 25 уроків з програмування Scratch для дітей.

– Збірник практичних робіт у середовищі програмування Скретч з інформатики для 5–7 класів (збірник практичних завдань) [19]. У цьому ресурсі представлені практичні роботи з програмування в Scratch 2.0, але можна адаптувати завдання і для нової версії Scratch.

Висновки до розділу 2

Вивчення Scratch у ліцеї є важливим для розвитку алгоритмічного мислення та цифрової компетентності серед учнів оскільки Scratch надає інтерактивний підхід до навчання програмування через створення власних

проєктів. Педагогічна методика використання Scratch включає: орієнтованість на проєктну діяльність (створення ігор, інтерактивних історій, симуляцій та експериментів), інтеграцію з іншими предметами (мова та література, математика, мистецтво, наука, історія, географія, технології та громадянська освіта), градацію складності завдань, застосування реальних завдань (створення ігор, інтерактивних оповідань, екологічних та соціальних проєктів, мультимедійних презентацій), роботу в команді, формування творчого мислення, звітності та оцінювання результатів, підтримку зовнішніх ресурсів.

Завдання та опис факультативу спочатку створювались в окремому текстовому документі, сортувались за категоріями та впорядковувались за методом градації складності завдань – від легших до складніших. Після цього завдання розміщувались та публікувались на сайті, що створювався за допомогою конструктора сайтів GoogleSites.

Отже, можна підсумувати, що робота зі Scratch є цікавою та водночас простою, розвиває в дітях логічне мислення, здатність творити, та формує нові навички. Як можна побачити з результатів педагогічного експерименту, інтерес до середовища серед молоді дуже значний, але через недостатню підготовку вчителів та невелику кількість занять з програмування у ліцях не всі можуть користуватись ним та реалізовувати власні творчі ідеї та проєкти.

ВИСНОВКИ

Результатом кваліфікаційної роботи є спроектований та розроблений навчально-методичний комплекс для організації та проведення занять факультативу з програмування мовою Scratch для профільного навчання інформатики.

У процесі виконання роботи для досягнення мети були розв'язані усі поставлені завдання:

– проведений теоретичний аналіз наукових та методичних джерел з предметної області дослідження. Під час використання середовища програмування Scratch у школярів розвивається творчий підхід, вони використовують інформаційні та телекомунікаційні технології, а також набувають навичок самостійного навчання. Використовуючи в процесі навчання середовище програмування Scratch, школярі вчаться самостійно складати алгоритми та усвідомлювати функціональність роботи циклів і умовних операторів, вчаться приймати самостійні рішення, можуть бачити практичне призначення алгоритмів і програм. Використання візуального об'єктно-орієнтованого середовища програмування Scratch під час вивчення інформатики сприяє розвитку наочно-образного мислення і задовольняє пізнавальні інтереси школярів;

– досліджено практику використання Scratch в освітніх системах різних країн світу. У багатьох країнах світу розглядають можливість вивчення основ програмування як необхідної навички, аналогічно читанню та письму. Важливо відзначити, що кожна країна може впроваджувати Scratch, враховуючи свої особливості та адаптації до власної освітньої системи. Ці приклади демонструють різноманітні можливості використання Scratch у системі освіти для навчання програмуванню та розвитку комп'ютерної грамотності учнів;

– проаналізовані варіанти використання Scratch на заняттях інформатики у ліцеях. Вивчення програмування на Scratch відкриває широкі можливості для ефективного формування ключових, міжпредметних та предметних

компетентностей учнів. Розробка та реалізація Scratch-проектів ігрового характеру дозволяє активізувати навчальний процес, створювати інтерактивні анімовані матеріали, спільно працювати над проектами та обмінюватися результатами через Scratch-спільноту. Крім того, це стимулює самостійну активність учнів;

– розроблено навчальну програму та навчально-методичний комплекс для організації та проведення факультативу з програмування мовою Scratch для профільного навчання інформатики. Результатом роботи є сайт, створений за допомогою конструктора GoogleSites, на сторінках якого розміщено та описано завдання факультативу, що містять: посилання на демонстраційну програму, покрокову інструкцію з створення програми з описом та скріншотами програм;

– проведений педагогічний експеримент (констатувальний, формувальний та контрольний етапи) для дослідження ефективності використання навчально-методичного комплексу. На констатувальному етапі експерименту було визначено що учні зацікавлені в даній роботі та мають середній рівень навичок програмування в Scratch. В ході формувального етапу експерименту було проведено 11 занять з розробки ігор в середовищі Scratch, середня тривалість яких була 60-70 хвилин. На контрольному етапі експерименту було проведено опитування учасників, у результаті якого отримані відгуки про сам факультатив новий рівень навичок програмування в Scratch – в більшості учасників він піднявся до високого рівня.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. About | FIRST LEGO League. 2023. URL: <https://www.firstlegoleague.org/about#>
2. Coolest Projects | The Technology Showcase For Young People. 2023. URL: <https://online.coolestprojects.org/>
3. Green Coding: онлайн конкур-хакатон з програмування на Scratch – RoboClubIF. 2023. URL: <http://roboclub.if.ua/post/green-coding-onlain-konkur-khakaton-z-programuvannya-na-scratch>
4. Malan, David J. and Henry H. Leitner. Scratch for Budding Computer Scientists. ACM SIGCSE Bulletin. 2007. 39(1). С. 223–227. URL: <http://nrs.harvard.edu/urn-3:HUL.InstRepos:2819231>
5. Maloney J., Burd L., Kafai Y., Rusk N., Silverman, B., and Resnick M. Scratch: A Sneak Preview, Proceedings of IEEE Conference on Creating, Connecting, and Collaborating through Computing (Kyoto, Japan), 2004, p. 84
6. Materiały do pobrania – programy nauczania – programy do szkoły podstawowej – Ośrodek Rozwoju Edukacji. 2023. URL: <https://www.ore.edu.pl/2019/12/programy-nauczania-programy-do-szkoly-podstawowej/>
7. Scratch – Imagine, Program, Share. 2023. URL: <https://scratch.mit.edu/>
8. Scratch Day 2023 — Scratch Foundation. 2023. URL: <https://www.scratchfoundation.org/scratch-day>
9. Semenikhina O., Rudenko Yu. Problems of educating to programming of students and way of their overcoming. Information technologies and learning tools, 2018. 4(66). PP. 54–64.
10. Апостол Д. Формування та розвиток предметних компетентностей на уроках інформатики. 2014. URL: <https://bit.ly/2zkvtp1>
11. Васенко О. В. Реалізація можливостей інтегрованого середовища розробки Lazarus у вивченні інформатики в школі. Комп'ютер у школі та сім'ї. 2016. № 7. С. 32–35.

12. Василенко Я. П., Кирстюк І. П. Про педагогічні та дидактичні особливості середовища Scratch як інструменту навчання основам алгоритмізації та програмування. Матеріали III Міжнародної науково-практичної інтернет-конференції «Сучасні інформаційні технології та інноваційні методики навчання: досвід, тенденції, перспективи». Тернопіль, ТНПУ, 2019. С. 21–24.

13. Вивчаємо Scratch разом. 2023. URL: <https://sites.google.com/comp-sc.if.ua/scratch-together/%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%B0-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0?authuser=0>

14. Глинський Я.М., Палюшок Л.В. Яку мову програмування вивчати у школі (матеріали для дискусії). Комп'ютер у школі та сім'ї. 2013. № 8. С. 9–18

15. Горошко Ю., Костюченко А., Шкардибарда М. Використання ВПЗ у процесі вивчення основ програмування. Інформатика та інформаційні технології. 2012. №1. С. 22–25.

16. Дистанційне навчання. 2023. URL: <https://www.youtube.com/@user-dw6wy3nf8n>.

17. Дмитренко К. А., Коновалова М. В., Семиволос О. П., Бекетова С. В. Звичайні форми роботи – новий підхід: розвиваємо ключові компетентності: метод. посіб. Х.: ВГ «Основа», 2018. 119 с. URL: <https://bit.ly/2O1O7Xn>

18. Дудка О. М., Власій О. О. Особливості вивчення програмування на Scratch. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. Науковий журнал. 2017. №26. С. 81–87. URL: <http://ki.lutsk-ntu.com.ua/node/134/section/17>

19. Збірник практичних робіт у середовищі програмування Скретч з інформатики для 5–7 класів (збірник практичних завдань). 2023. URL:

<https://naurok.com.ua/zbirnik-praktichnih-robit-u-seredovischi-programuvannya-skretch-282612.html>

20. Знайшов. 2023. URL: https://znayshov.com/News/Details/mizhnarodni_zmahannia_u_seredovyshchi_scratch

21. Інформатика. 2023. URL: <https://it-science.com.ua/poststag/3?page=1>

22. КГ №114 КМР. 2023. URL: <https://sites.google.com/view/school114>

23. Кирстич І.П., Василенко Я.П. Про педагогічні та дидактичні особливості середовища Scratch як інструменту навчання основам алгоритмізації та програмування. URL: <http://dspace.tnpu.edu.ua/bitstream/123456789/14524/1/Kyrstych.pdf>

24. Костерна Л. В. Програмування в середовищі SCRATCH. Комп'ютер у шк. та сім'ї. 2015. №4. С. 17–19.

25. Кремень В. Г. У змісті шкільної освіти мають відбутися зміни. URL: <https://bit.ly/2OVp4pc>

26. Курінний С. Scratch. Програмуємо ігри. 2023. URL: https://shron1.chtyvo.org.ua/Kurinnyi_Serhii/Scratch_Prohramuiemo_ihry_25_urokiv_dlia_ditei.pdf?PHPSESSID=82b4vm85li8ilj7rjona5u8a11

27. Лещук С. О. Scratch: ази формування програмістського мислення. FOSS Lviv – 2018: матеріали міжнар. наук.-практ. конф., м. Львів, 26-29.04 2018 р.

28. Лубіна Є. А. Мобільне навчання у дидактиці вищої школи. Вісник Львівського ун-ту: Педагогіка. 2009.

29. Машталір О.В., Лещук С.О., Дільна Н.З. Пропедевтика алгоритмізації засобами Scratch. URL: http://dspace.tnpu.edu.ua/bitstream/123456789/15343/1/3_Mashtalir_Leshchuk_Dilna.pdf

30. Морзе Н. В., Барна О. В. Я досліджую світ. Підручник для 3 класу закладів загальної середньої освіти (у 2-х частинах): Ч. 2. К.: УОВЦ «Оріон», 2020. 172 с.
31. Морзе Н. В., Гладун М.А., Дзюба С.М. Формування ключових і предметних компетентностей учнів робототехнічними засобами STEM-освіти. Інформаційні технології і засоби навчання. Науковий журнал. 2018. Том 65. №3. С.37–52. URL: <https://journal.iitta.gov.ua/index.php/itlt/article/view/2041/1348>
32. Морзе Н., Барна О., Вембер В., Кузьмінська О. Система компетентнісних завдань як засіб формування компетентностей на уроках інформатики. Інформатика та інформаційні технології в навчальних закладах. 2015. № 4. С. 17–27. URL: <https://bit.ly/2R1n9Vm>.
33. Нова українська школа: poradnik dla vchytelja. Під заг. ред. Бібік Н. М. К.: ТОВ «Видавничий дім «Плеяди». 2017. 206 с. URL: <https://bit.ly/2EMebGN>
34. Патаракин Е.Д. Школа Scratch. Шкільні технології. 2010. С. 132–135.
35. Рижаніна К.О. Факультатив з Scratch. [Електронний ресурс]. Режим доступу: <https://sites.google.com/view/scratch-facultative>.
36. Роль програми «Скретч» у освіті середніх шкіл України URL: <http://www.refsua.com/referat-8508-1.html>.
37. Самойленко Н. І., Семко Л. П. Формування інформаційних компетенцій на уроках інформатики в основній школі. URL: <https://bit.ly/2DPAt8Z>.
38. Семко Л. П. Компетентнісний підхід до навчання інформатики в основній школі. 2013. URL: <https://goo.gl/Ln9ici>.
39. Скасків Г. М. Методичні аспекти використання проєктних технологій у системі дистанційного навчання у процесі вивчення інформатики. Наук. зап. Терноп. нац. пед. ун-ту ім. В. Гнатюка. Сер. Педагогіка. 2011. Вип. 1. С. 108–114.

40. Скретч (мова програмування). 2023. URL: <https://www.wikidata.uk-ua.nina.az/Scratch.html>.

41. Як створити сайт самостійно і безкоштовно. 2023. URL: <https://pbb.lviv.ua/statti-i-novyny/statti-shchodo-stvorennia-saitu/jak-stvoryty-sayt-samostiyno-i-bezkoshtovno/>.

42. Google Sites що це? Пісочниця для студентських проєктів чи зручний конструктор сайтів?. 2023. URL: <https://apix-drive.com/ua/blog/reviews/schto-take-google-sites>.

ДОДАТКИ

Додаток А

План факультативу

1 – Знайомство з Scratch

1. Запуск Scratch
2. Автономний редактор
3. Редактор Scratch та спрайти
4. Графічний редактор
5. Робота з блоками коду
6. Демонстрація ваших програм

Перший розділ є ознайомчим, він створений для тих користувачів, які не працювали з Scratch.

2 – Створення ігор

1. Створення меню гри
2. Створення опції вибору персонажу в грі
3. Створення гри «Піаніно»
4. Створення гри «Стрибучий дінозавр»
5. Створення гри «Flappy Bird»
6. Створення гри «Математична вікторина»
7. Створення гри «Арканоїд»
8. Створення гри «Галага»
9. Створення гри «Змійка»
10. Створення гри «Дудл-джамп»

3 – Scratch+робототехніка

1. Анімація кліпання
2. Малювання в повітрі
3. Лабіринт
4. Полювання
5. Галага
6. Втеча від перешкод (горизонтальна/вертикальна)

7. Математична вікторина

4 – Креативне програмування

1. Створення милої жабки
2. Плавучий острів
3. Захисник планети
4. Банка компліментів
5. Печиво з передбаченням

5 – STEM

1. Марсохід
2. Екологічний моніторинг
3. Симулятор космічного польоту
4. Розв'язання математичних головоломок
5. Симуляція роботи сонячної електростанції
6. Імітація науки про вогонь та хімічних реакцій
7. Історична симуляція
8. Програмування музики та звуку

Питання анкети**(констатувальний етап педагогічного експерименту)**

1. Вкажіть ваше ім'я та прізвище.
2. Вкажіть ваш клас.
3. Чи вивчали ви Scratch у школі?
4. Чи використовували ви Scratch для створення ігор/проектів?
5. Чи брали ви участь в конкурсах з програмування Scratch?
6. Чи хотіли в ви вивчати Scratch в 10/11 класі?
7. Чи знаходили ви ресурси для самостійного вивчення Scratch?
8. Оцініть свій рівень програмування в Scratch від 0 до 10.

Питання анкети**(контрольний етап педагогічного експерименту)**

1. Вкажіть ваше ім'я та прізвище.
2. Вкажіть ваш клас.
3. Оцініть завдання факультативу з Scratch від 0 до 10.
4. Чи порекомендували б Ви цей факультатив своїм друзям або однокласникам як ресурс для вивчення програмування?
5. Чи плануєте Ви використовувати факультатив надалі?
(опрацьовувати інші розділи)
6. Чи плануєте Ви брати участь в конкурсах з програмування Scratch використовуючи знання, отримані при вивченні факультативу?
7. Оцініть свій теперішній рівень володіння Scratch від 0 до 10.
8. Залиште відгук про факультатив.