

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДОЛОГІЇ РОЗРОБЛЕННЯ НАВЧАЛЬНИХ ІГРОВИХ ЗАВДАНЬ З ІНФОРМАТИКИ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

(тема)

Виконав:
студент 2 курсу, групи ІНФМ-21-1

Проценко А. В.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Сакало Є. С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)
2022 р.

Кобилін О.А.
(прізвище, ініціали)

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«___» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Проценку Артему Валерійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження та реалізація методології розроблення навчальних ігрових завдань з інформатики для мобільних пристроїв

затверджена наказом по університету від 9 листопада 2022 року № 1469Ст

2. Термін подання студентом роботи до екзаменаційної комісії 26 листопада 2022 р.3. Вихідні дані до роботи теоретичні відомості про методи розроблення навчальних ігрових завдань з інформатики, науково-технічна й науково-методична література, матеріали конференцій, дані інтернет-мережі, мова програмування JavaScript, середовище розроблення PhpStorm.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд теоретичного матеріалу щодо використання ігрових завдань під час навчання.

2. Аналіз існуючих ігрових застосунків для навчання з інформатики.

3. Програмна реалізація вебзастосунку з навчальними ігровими завданнями.

4. Тестування розробленого застосунку і аналіз результатів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми навчальних ігрових завдань, постановка задачі, аналіз існуючих навчальних ігрових завдань, екранні форми програмного продукту.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	09.11.2022	
2	Аналіз завдання, підбір літератури	09.11.22-10.11.22	
3	Аналіз літератури з досліджуваної проблеми	10.11.22-12.11.22	
4	Дослідження проблеми ігрових завдань	12.11.22-15.11.22	
5	Аналіз існуючих застосунків для навчання	15.11.22-18.11.22	
6	Програмна реалізація	18.11.22-20.11.22	
7	Оформлення пояснювальної записки	20.11.22-23.11.22	
8	Перевірка на плагіат	26.11.22	
9	Рецензування	29.11.22	
10	Підготовка презентації та доповіді	30.11.22	
11	Занесення роботи в електронний архів	04.12.22	
12	Попередній захист кваліфікаційної роботи	05.12.22	

Дата видачі завдання 9 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Сакало Є. С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 59 с., 4 табл., 36 рис., 1 дод., 40 джерел.

ВЕБЗАСТОСУНОК, ІГРОВЕ ЗАВДАННЯ, НАВЧАННЯ ІНФОРМАТИЦІ, НАВЧАННЯ ПРОГРАМУВАННЮ, МОБІЛЬНИЙ ПРИСТРІЙ, JAVASCRIPT.

Об'єктом дослідження є методологія розроблення навчальних ігрових завдань з інформатики для мобільних пристроїв.

Метою дослідження є розробка ПЗ з навчальними ігровими завданнями, яке можна буде застосовувати задля навчання інформатики з використанням мобільних пристроїв.

Використано методи аналітичного обґрунтування. Проведено дослідження методів використання навчальних ігрових завдань як продуктивного способу навчання інформатиці. Досліджені існуючі навчальні ігрові завдання для мобільних пристроїв. Розроблено вебзастосунок, який сприяє вивченню інформатики учнями, а саме принципів роботи з пам'ятю, стеками.

У результаті дослідження здійснена програмна реалізація навчальних ігрових завдань з інформатики.

WEB APPLICATION, GAMING, LEARNING COMPUTER SCIENCE, LEARNING PROGRAMMING, MOBILE DEVICE, JAVASCRIPT.

The object of the research is the methodology of developing educational game tasks in informatics for mobile devices.

The purpose of the research is to develop a software with educational game tasks that can be used to teach computer science using mobile devices.

Analytical reasoning methods were used. A study of the methods of using educational game tasks as a productive way of learning computer science was conducted. Existing educational game tasks for mobile devices were studied. A web application has been developed that helps students learn computer science, namely the principles of working with memory and stacks.

As a result of the research, the software implementation of educational game tasks in informatics was carried out.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Теоретико-методологічний аспект дослідження проблеми ігрових завдань	8
1.1 Сутність навчальних ігрових завдань	8
1.2 Загальний огляд технологій електронного навчання.....	10
1.3 Огляд стандартів електронного навчання	11
1.4 Огляд мобільного навчання в системі сучасних освітніх технологій	12
1.5 Загальний огляд процесу навчання комп'ютерному програмуванню.....	15
1.6 Постановка задачі дослідження.....	15
2 Аналіз існуючих навчальних ігрових завдань	17
2.1 «LightBot»	17
2.2 «Code Adventures».....	19
2.3 «DinoCoding».....	20
2.4 «Algorun Free»	22
2.5 Порівняння реалізацій керуючих структур, функцій, змінних та подій в іграх з елементами мобільного кодування.....	25
3 Розроблення навчальних ігрових завдань з інформатики	27
3.1 Обґрунтування вибору середовища програмної реалізації	27
3.2 Програмна реалізація.....	29
3.3 Інструкція користувача	31
3.4 Тестування розробленої моделі.....	34
3.3.1 Загальний огляд тестування методом чорної скриньки.....	34
3.3.2 Результати проведення тестування ПЗ	35
Висновки	47
Перелік джерел посилання	49
Додаток А Екранні форми програмного продукту	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕОМ – електронно-обчислювальна машина

ЗВО – заклад вищою освіти

ІКТ – інформаційно-комунікаційні технології

ПК – персональний комп'ютер

ОС – операційна система

CSS – Cascading Style Sheets (каскадні таблиці стилів)

HTML – HyperText Markup Language (мова розмітки гіпертексту)

IDE – Integrated Development Environment (інтегроване середовище розробки)

PDA – Personal Digital Assistant (персональний цифровий помічник)

SPA – Single Page Application (односторінковий застосунок)

ВСТУП

Інформатика є однією з дисциплін, які стрімко розвиваються. Базова інформатика (основи та базові поняття) як правило вивчається в школах. Передова або поглиблена інформатика вивчається в ЗВО і пропонує можливість подальшого розвитку навичок вирішення проблем шляхом вивчення різноманітних мов і парадигм програмування, а також отримання навичок і знань з широкого кола тем в області інформатики.

Сьогоднішній рівень розвитку інформаційно-комунікаційних технологій дозволяє використовувати в навчальному процесі інноваційні методики навчання: дистанційні освітні технології, ulearning (всепроникаюче навчання), m-learning (мобільне навчання), e-learning (електронне навчання).

Термін «m-learning», що з'явився в англійській літературі близько 10 років тому [1], останнім часом став все частіше використовуватися в Україні. В нашій країні за останній час кількість користувачів смартфонів збільшилася в три рази (до 34%), згідно з результатами дослідження компанії TNS Infatest на замовлення Google. Тоді як в 2012 році ці показники досягали лише 7%.

Навчальні ігрові завдання можуть сприяти кращому засвоєнню матеріалу учнями з інформатики, ніж інші види навчання. Такі завдання можуть базуватися на складанні алгоритмів, розв'язанні задачі й проходженню тестів по інформатиці.

Актуальність дослідження полягає у перевагах, які мають навчальні ігрові завдання перед традиційними способами вивчення інформатики, а також в поширенні «мобільного навчання» в Україні.

1 ТЕОРЕТИКО-МЕТОДОЛОГІЧНИЙ АСПЕКТ ДОСЛІДЖЕННЯ ПРОБЛЕМИ ІГРОВИХ ЗАВДАНЬ

1.1 Сутність навчальних ігрових завдань

Процес інформатизації суспільства стає все більш стрімким. Щоб на заняттях кожен учень працював зацікавлено та енергійно, учителі розшукують нові продуктивні методи навчання і такі методичні прийоми, які спонукали б розумову активність учнів, заохочували б їх до самостійного здобуття знань. Це завдання не тільки і не стільки змісту освіти, скільки технологій, які застосовуються під час навчання. Серед різних напрямів сучасних педагогічних технологій найбільш плідним є використання ігрових завдань.

До необхідності перегляду засобів, методів, форм, цілей і змісту навчання учнів спонукають нові вимоги, які висуває перед освітніми системами суспільство, психолого-педагогічні особливості учнів, які вирости в світі, в якому переважають інтернет, мобільні технології, комунікація і взаємодія [1–7].

Використання ігрових завдань спонукає учнів до роботи, піднімає мотиваційну сферу освітньої діяльності, що визначається цілою низкою факторів: характером навчальної діяльності учнів, сформованістю її структурних складників (навчальних дій, навчальних завдань, самоаналізу й самоконтролю), змістом навчання для кожного учня; взаємодії в ході гри з іншими членами ігрового проекту; зрілістю цілей, характером мотивів, що акомпанують процес виконання навчального ігрового проекту; формою організації освітнього процесу й способом взаємодії усіх співучасників кожної залученої до гри групи [8–14].

Перед сучасним навчальним закладом постає завдання розвинути ініціативу, творчий потенціал, самостійність учнів. Орієнтоване на

самореалізацію і самовдосконалення особистості навчання стає пріоритетом сучасної освіти.

Гра, як особливе організоване заняття, потребує напруги розумових і емоційних сил. Вона завжди передбачає вживання рішення, бажання виграти та загострює інтелектуальну діяльність граючих.

Через навчальні ігри людина зможе експериментувати, відкривати та пізнати навчальний матеріал, який її оточує, оскільки матеріальні ресурси сприяють її фізичному, соціальному та емоційному тренуванню, яке можна відчувати безпосередньо, і як висновок, навчальний матеріал за допомогою навчальних ігор засвоюється краще [15–20]. За допомогою ігрового методу навчання людина навчатиметься через відкриття, експерименти, зроблені з усіма предметами, які вона знаходить у своєму оточенні [21, 22]. Завдяки грі людина також отримує навички природним шляхом, і після трьох років усе, що є у дитини, матиме безпосередній зв'язок із грою, і якщо вона продовжуватиметься, користувач зможе дати краще значення для світу, в якому він росте.

Більшість досліджень, які базуються на мобільних іграх для навчання студентів, зосереджуються на перевагах викладання і навчання в середовищі школи чи кампусу, деякі дослідження показують, що використання ігор у викладанні та навчанні може мати позитивний вплив на навчання студентів [5, 7–9], хоча це трапляється не в усіх випадках, тому що процес створення та проектування навчальних ігор є важливим атрибутом для розробки хороших навчальних ігор.

Спираючись на попередні дослідження, можна зробити висновок про використання етапу розвитку дитини шляхом формування бази знань за допомогою ігрової методики з привабливим і гарним ігровим дизайном, так що через навчання гра зможе сприяти засвоєнню інформації.

1.2 Загальний огляд технологій електронного навчання

E-learning належить до великої науково-практичної галузі, яка носить загальну назву автоматизованого навчання.

Її розвиток можна поділити на 3 етапи:

– перший етап охоплює період 1920–1950-ті рр. з моменту появи електромеханічних ПК до широкого впровадження електронних ПК. Перший етап характеризується застосуванням різних електронних, механічних й електромеханічних індивідуалізованих пристроїв, за допомогою яких подавався навчальний матеріал і виконувався контроль знань за технологією програмованого навчання;

– другий етап (1950–1980-х рр.) та охоплює період пов'язаний з широким впровадженням ЕОМ у практику, що не могло залишити осторонь фахівців у галузі освіти, тому спочатку з'являються ідеї навчання кібернетики в школі, впровадження елементів прикладної математики в навчальний процес, з'являються комп'ютерно-орієнтовані середовища навчання, автоматизовані системи контролю знань та управління навчальним процесом. Ключовими термінами цього періоду стали комп'ютерні системи контролю знань, комп'ютерна підтримка навчального процесу, комп'ютерно-орієнтовані системи навчання й інтелектуальні навчаючі системи. Під час цього періоду була створена велика кількість спеціалізованого програмного забезпечення – автоматизованих навчальних систем Tutor, Coursewriter, PLATO та ін. Цьому сприяли очевидні переваги електронних комп'ютерів над електромеханічними, а саме більш широкі засоби для перегляду навчальних матеріалів, висока швидкість опрацювання та розрахунків, наявність пам'яті для зберігання навчальних матеріалів тощо;

– третій етап розпочався у 1980-х рр. і продовжується донині. Він розпочався з появою ПК і комп'ютерних мереж. Глобальна мережа Інтернет надала потужний імпульс у розвитку навчальних технологій. Віддалений доступ до навчального контенту, Web-технології, використання спільних і

розподілених ресурсів забезпечив кардинальний зріст продуктивності професійної підготовки, її масовості й доступності, уможливив створення фахових середовищ і систем для надання навчальних послуг і реалізації різноманітних видів формальної і неформальної освіти. Ключовими термінами цього періоду є мобільне навчання, електронне навчання, дистанційне навчання, навчання протягом всього життя, неперервна освіта, віртуальний університет, віртуальне навчання, гіпертекст, Web-курси, Інтернет [23, 24].

1.3 Огляд стандартів електронного навчання

Прийняття стандартів і специфікацій сприяє домінуванню незалежних від платформи відкритих технологій і сприяє створенню орієнтованих на користувача систем електронного навчання. Стандартизовані технології мають ряд переваг, які захищають і підтримують інвестиції в електронне навчання. Це загалом:

- сумісність: вміст від багатьох постачальників можна легко поширювати серед споживачів і багатьох систем. Легко вирішуються проблеми перекладу, комунікації, обміну інформацією та досягається прозора взаємодія систем;

- можливість повторного використання: вміст і код можна швидко та легко збирати, розбирати та повторно використовувати. Крім того, об'єкти вмісту можна адаптувати та використовувати в контексті, відмінному від початково розробленого;

- керованість: системи можуть відстежувати відповідну інформацію про учня та контент. Профілі учнів, освітня ціль та контент «розмовляють однією і тою ж мовою», тому легше знайти, керувати та зібрати «правильний матеріал» для кожного конкретного випадку;

– доступність: учень може отримати доступ до відповідного вмісту в потрібний час на відповідному пристрої. Сховища контенту можуть бути розроблені та доступні для любителів або професіоналів, які використовують будь-яку програму на основі загальних стандартів;

– довговічність: вміст створюється один раз і багато разів трансплантується на різні платформи та системи з мінімальними зусиллями. Покупці не потрапляють у «пастку» власної технології навчання конкретного постачальника, і їхні інвестиції стають постійними та несприятливими для ризику;

– масштабованість: функціональність технологій навчання може бути розширена, щоб обслуговувати більшу кількість населення та організаційні цілі. Прибуток організації від інвестицій у продукти електронного навчання може зрости, якщо їх можна використати за межі початкового обсягу.

Стандартизація процедур електронного навчання може бути досягнута через співпрацю всіх учасників спільноти електронного навчання: розробники, постачальники та користувачі повинні працювати разом, щоб створювати, перевіряти, встановлювати та поширювати стандарти для кожного завдання електронного навчання [25].

1.4 Огляд мобільного навчання в системі сучасних освітніх технологій

Термін «мобільне навчання» (m-learning) означає використання мобільних і кишенькових ІТ-пристроїв, таких як PDA, мобільні телефони, ноутбуки та технології планшетних ПК, у викладанні та навчанні.

Оскільки ПК та Інтернет стають основними освітніми інструментами, технології стають більш портативними, доступними, ефективними та простими у використанні. Це надає багато можливостей для розширення участі та доступу до ІКТ, зокрема Інтернету. Мобільні пристрої, такі як

телефони та кишенькові комп'ютери, коштують набагато дешевше, ніж настільні комп'ютери, і тому є менш дорогим способом доступу до Інтернету (хоча вартість підключення може бути вищою). Поява планшетних комп'ютерів тепер дозволяє мобільний доступ до Інтернету з рівною, якщо не більшою, функціональністю, ніж настільні комп'ютери [26–28].

M-learning, або «мобільне навчання», має різне значення для різних спільнот. Незважаючи на те, що воно пов'язане з електронним навчанням і дистанційною освітою, воно відрізняється своєю спрямованістю на навчання в різних контекстах і навчання за допомогою мобільних пристроїв. Одне з визначень мобільного навчання таке: навчання, яке відбувається в різних місцях або яке використовує переваги можливостей навчання, які пропонують портативні технології. Іншими словами, мобільне навчання зменшує обмеження місця навчання завдяки мобільності звичайних портативних пристроїв.

Термін охоплює: навчання за допомогою портативних технологій, де фокус зосереджений на технології (яка може бути у фіксованому місці, наприклад, у класі); навчання в різних контекстах, де основна увага приділяється мобільності учня, взаємодії з портативною або стаціонарною технологією; і навчання в мобільному суспільстві, зосереджуючись на тому, як суспільство та його інституції можуть пристосувати та підтримати навчання все більш «мобільного» населення, яке не задоволене існуючими методологіями навчання.

M-learning є зручним у тому сенсі, що воно доступне практично з будь-якого місця (клас, таксі, пральня, ванна кімната тощо), що забезпечує доступ до всіх різноманітних навчальних матеріалів. Крім того, це співпраця; тобто обмін відбувається майже миттєво між усіма, хто використовує той самий вміст, що, у свою чергу, також призведе до отримання миттєвих відгуків і порад. M-learning також забезпечує потужну портативність, замінюючи книги та нотатки невеликими ОЗУ, наповненими адаптованим навчальним вмістом. Крім того, такий вид навчання є захоплюючим і веселим. За

допомогою такого типу навчання набагато легше поєднувати гру та навчання для більш ефективного та цікавого досвіду [29].

Мобільні навчальні завдання повинні відповідати на такі запитання:

– доступ: наскільки широко доступна бездротова мережа, яка розповсюджуватиме мобільний контент?

– насиченість: чи швидко завантажуються сторінки? Чи анімація відтворюється плавно та безперебійно? Чи потокове медіа (медіа, яке споживається – читається, слухається, переглядається) передається з достатньою швидкістю?

– ефективність: наскільки великий клієнт, якому знадобиться використовувати певний медіаплеєр? Як швидко програма завантажуватиметься та відтворюватиметься?

– гнучкість: чи можна буде переглядати програму на різних пристроях? Чи можна відтворювати вміст, призначений для використання з одним типом пристроїв або операційною системою, на інших пристроях із деякими очікуваннями порівнянної якості?

– безпека: чи захищено інтерактивний мобільний пристрій від хробаків і вірусів? Чи захищено спільний вміст від перехоплення ненавмисними одержувачами?

– надійність: чи вміст відобразатиметься узгоджено, незалежно від браузера, пристрою та розміру екрана?

– інтерактивність: чи дозволяє програма користувачам вільно взаємодіяти з дисплеєм і вмістом? [30, 31].

Завдяки використанню мобільного навчання користувач може отримати доступ до навчального контенту без будь-яких кордонів, тож до нього можна отримати доступ у будь-який час з цікавими ілюстраціями.

1.5 Загальний огляд процесу навчання комп'ютерному програмуванню

Щоб отримати навички комп'ютерного програмування, потрібні чотири види знань: базові математичні знання, розуміння використання IDE, вивчення синтаксису обраної мови програмування та перетворення рішення в логіку програми.

Найбільше труднощів у програмістів-початківців виникає з четвертим типом знань, тобто зі створенням своїх алгоритмів для різного роду завдань. Під час розробки логіки алгоритмів студентам доступні три типи керуючих структур: послідовності, вибірки та ітерації. Їм доводиться поєднувати ці структури керування, щоб створити логіку програми, що іноді може бути складним завданням. Коли учні починають вивчати основи програмування, мобільні ігри з кодуванням можуть допомогти їм зрозуміти, як використовувати керуючі структури. У цьому дослідженні розглядається кілька мобільних ігор («Lightbot», «Code Adventures», «DinoCoding», «AlgoRun Free»), які можна використовувати для впровадження алгоритмізації та програмування. У цих мобільних іграх дітям належить вирішувати захоплюючі завдання по управлінню рухами персонажа. В іграх використовуються значки інструкцій, щоб швидко розробляти код, розташовуючи різні типи значків у правильному порядку. У цьому дослідженні порівняно реалізацію керуючих структур у цих іграх.

1.6 Постановка задачі дослідження

Таким чином, навчальні ігрові завдання можуть сприяти кращому засвоєнню матеріалу учнями з інформатики, ніж інші види навчання. Такі завдання можуть базуватися на складанні алгоритмів, розв'язанні задачі та проходженню тестів по інформатиці. Зважаючи на усі переваги, які надає у

навчанні подібні реалізації завдань з інформатики, дослідження та реалізація методології з їх розробки для мобільних пристроїв залишається актуальною задачею і вимагає подальшого вивчення.

Об'єктом дослідження є методологія розроблення навчальних ігрових завдань з інформатики для мобільних пристроїв.

Метою дослідження є розробка ПЗ з навчальними ігровими завданнями, яке можна буде застосовувати задля навчання інформатики з використанням мобільних пристроїв.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих навчальних ігрових завдань для мобільних пристроїв і методологій їх розроблення;
- розробити навчальне ігрове завдання для мобільних пристроїв.

2 АНАЛІЗ ІСНУЮЧИХ НАВЧАЛЬНИХ ІГРОВИХ ЗАВДАНЬ

У цьому розділі буде коротко описано деякі мобільні ігри з елементами кодування для ОС Android, які можна ефективно використовувати для знайомства з комп'ютерним програмуванням і алгоритмізацією. Незважаючи на те, що ці ігри були в першу чергу розроблені для учнів молодшої школи, їх можна успішно використовувати в освіті навіть у середніх школах чи університетах, особливо коли учні вперше вивчають алгоритмізацію та комп'ютерне програмування.

Будуть розглянуті наступні навчальні ігри для мобільних пристроїв:

- «LightBot»;
- «Code Adventures»;
- «DinoCoding»;
- «Algorun Free».

2.1 «LightBot»

Перша гра, яка буде розглянута – це «Lightbot». Ця гра є найстарішою серед усіх навчальних ігор для програмування для мобільних пристроїв, описаних у цьому дослідженні. Ця гра використовувалася як інструмент у багатьох попередніх наукових роботах [32–35], більшість із яких довели ефективність такого інструменту в освіті інформатики.

Гра містить кілька плиток у 3D-просторі та робота (рис. 2.1). На початку деякі плитки сині. Мета гри полягає в тому, щоб керувати роботом і підсвітити всі сині плитки. Гра містить 6 категорій (основи, процедури, перевантаження, цикли, умови та складні рівні) із 50 головоломками. Щоб вирішити головоломку, користувачі повинні створити невелику програму, за якою буде слідувати робот.

Також на рисунку 2.1 зображені червоні та зелені плитки. Ці плитки використовуються для умовних операторів «if». Коли робот намагається підсвітити ці плитки, колір робота змінюється на колір плитки, на якій він стоїть. Після цього можна виконувати будь-яку інструкцію залежно від кольору робота. Коли робот намагається знову запалити плитку червоного або зеленого кольору, колір робота повертається до нормального (сірого) кольору.

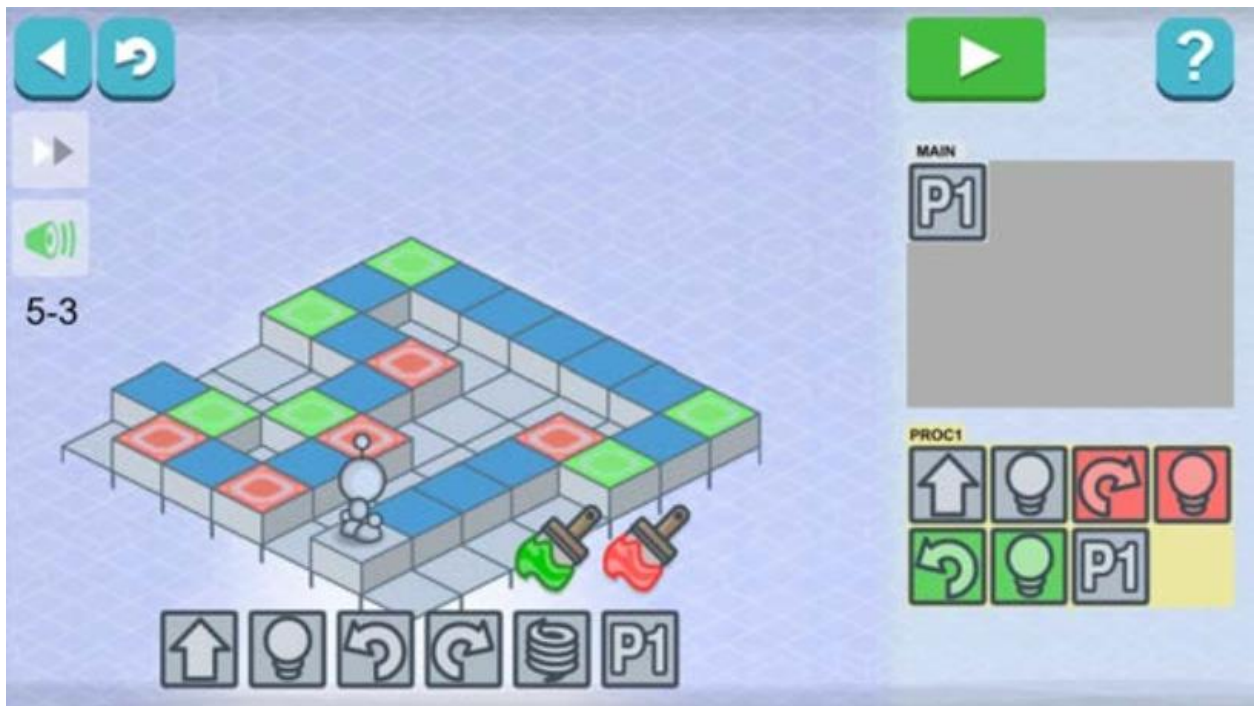


Рисунок 2.1 – Скріншот «LightBot» із розв’язком головоломки, що містить два умовні оператори «if»

На рисунку 2.2 показано деякі інструкції, які можна використовувати в грі (рухатися вперед, підсвічувати тайл або щось активувати, повертати ліворуч/праворуч, стрибати, викликати процедуру P1, повернутися з фактичної процедури), і зелений пензель. Використовуючи зелений пензлик, гравці також можуть вибрати будь-яку інструкцію із зеленим фоном. Такі інструкції будуть виконані, якщо лише поточний колір робота збігається з кольором інструкції.



Рисунок 2.2 – Приклади інструкцій «LightBot»

2.2 «Code Adventures»

Гра «Code Adventure» [36] – це двовимірна гра, де рожевій істоті потрібно знайти шлях до барвистого райдужного кола телепорту (рис. 2.3). Гра містить 6 категорій (основи, процедури, цикли, списки, умови та складні завдань) із 34 головоломками. Щоб вирішити головоломку, користувачі повинні створити невелику програму, за якою слідує рожева істота.

У цій грі гравці також можуть використовувати змінні. Рожева істота може збирати фрукти, які зберігатимуться в списку з 3 елементів, реалізованому у вигляді стека (LIFO, Last In First Out). Пізніше ці фрукти можна віддати птахам, які перекривають шлях до телепорту. Однак кожен птах приймає лише певний вид фруктів, тому гравцям потрібно написати програмний код, який дає кожному птаху бажаний фрукт (яблуко, апельсин або виноград).

На рисунку 2.3 також зображено, як оператори «if» реалізовані в грі. У той час як гравці можуть написати свої основні функції F1, F2 на 1-й вкладці вікна програмного коду, кожен умовний оператор має підготовлену область інструкцій на 2-й вкладці вікна програмного коду. Умовні оператори виконуються на основі верхнього (останнього) елемента списку збережених фруктів.



Рисунок 2.3 – Скріншоти «Code Adventure» із рішенням головоломки, яка містить три умовні оператори «if»

На рисунку 2.4 показано деякі з інструкцій, які можна використовувати в грі «Code Adventure» для створення програмних кодів розв’язків головоломок, а саме: рухатися ліворуч/праворуч, підніматися або спускатися по драбині, повернути руку або дати фрукт пташці, зачекати, поки рухомий острів досягне місця призначення, викликати функцію F1, викликати функцію F2, виконати оператори IF.

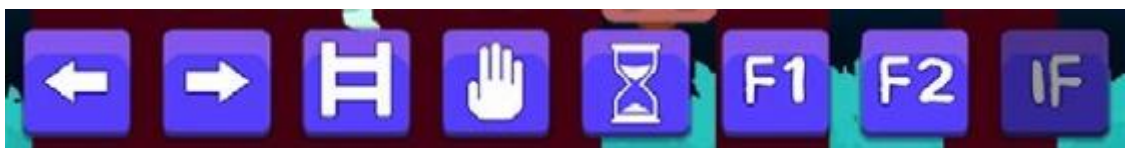


Рисунок 2.4 – Приклади інструкцій «Code Adventure»

2.3 «DinoCoding»

DinoCoding [37] також є двовимірною грою, де гравцеві потрібно написати однорядковий програмний код, який буде виконуватися динозавром, який керує роботом (рис. 2.5). Гра містить 6 груп головоломок із

загальною кількістю 108 головоломок. У кожній групі головоломок спочатку гравець повинен вибрати колір робота з 3 доступних кольорів. Далі гравцеві потрібно написати невеликий програмний код, щоб досягти мети.

Цілі та типи роботів різні для кожної групи головоломок:

– у першій групі головоломок мета гри – зібрати всі монети та запалити рослину;

– у другій групі головоломок робот повинен підвести робота до телепорту. У цій групі головоломок доступні лічильні цикли («за»), щоб діти могли виконати деякі інструкції більше разів;

– мета третьої групи головоломок – допомогти іншим роботам сісти на ракету, розбудивши їх або зарядивши батареї. Щоб вирішити ці головоломки, зазвичай, потрібен підхід, керований подіями (подія відбувається, коли наш робот натикається на іншого робота), а рішення здебільшого містять умовні («if») твердження;

– у наступній групі головоломок мета гри – відремонтувати зламаного робота. Для вирішення цих головоломок гравцям зазвичай доводиться використовувати цикли підрахунку («за»);

– мета п'ятої групи головоломок – відкрити скриню зі скарбами. Для вирішення цих головоломок необхідний підхід, керований подіями, подібно до головоломок у третій групі. Тут подія відбувається, коли наш робот натикається на стіну;

– в останній групі головоломок мета гри – зібрати всі монети та лопнути бульбашки, які захопили інші роботи. У цих головоломках учні повинні використовувати функції для виконання однієї послідовності інструкцій кілька разів.

На рисунку 2.6 показано деякі інструкції, які можна використовувати в грі «DinoCoding» (переміщення праворуч/ліворуч, стрибок праворуч/ліворуч, ремонт робота) і лічильний цикл «for» для виконання деяких інструкцій декілька разів. Останній фіолетовий значок на малюнку призначений для

створення рахункових («за») циклів. Будь-який цикл може містити одну або декілька інструкцій, які можна виконати від 1 до 5 разів.



Рисунок 2.5 – Скріншот «DinoCoding» з рішенням головоломки, що містить функцію, яка викликається 2 рази в основній програмі



Рисунок 2.6 – Приклади інструкцій «DinoCoding»

2.4 «Algorun Free»

У грі «Algorun Free» [38] робот виконує команди гравця в 3D-просторі. Рисунок 2.7 показує деякі інструкції, які можна використовувати в грі (рух вперед, поворот ліворуч/праворуч, виклик функції F1/F2/F3).

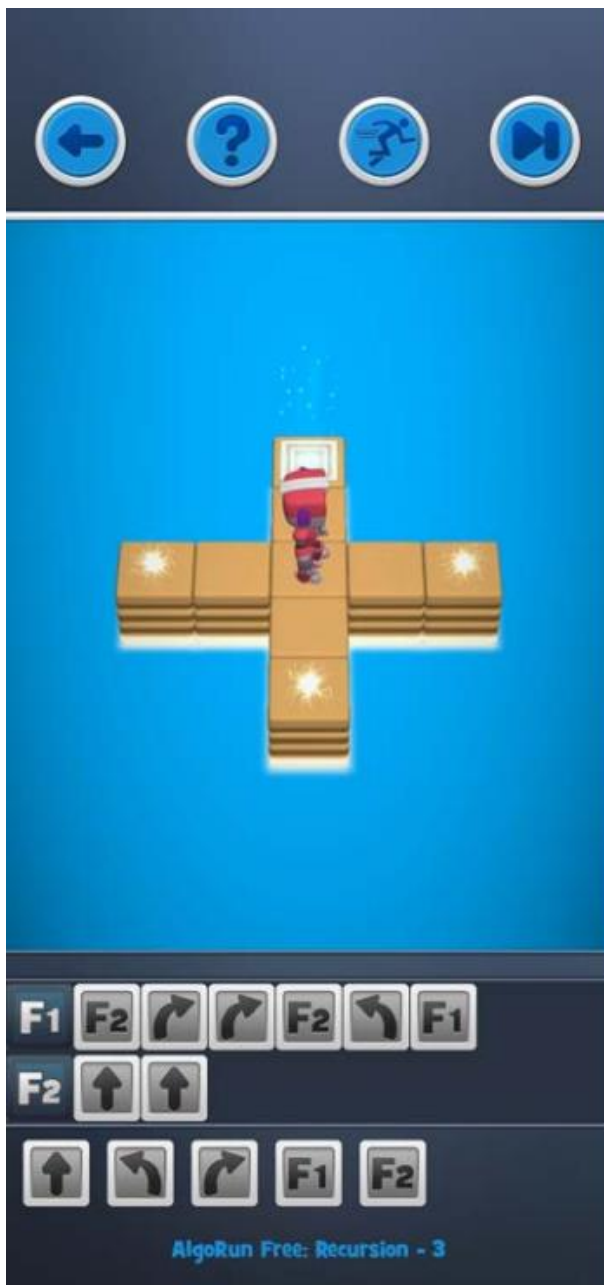


Рисунок 2.7 – Приклади інструкцій «Algorun Free»

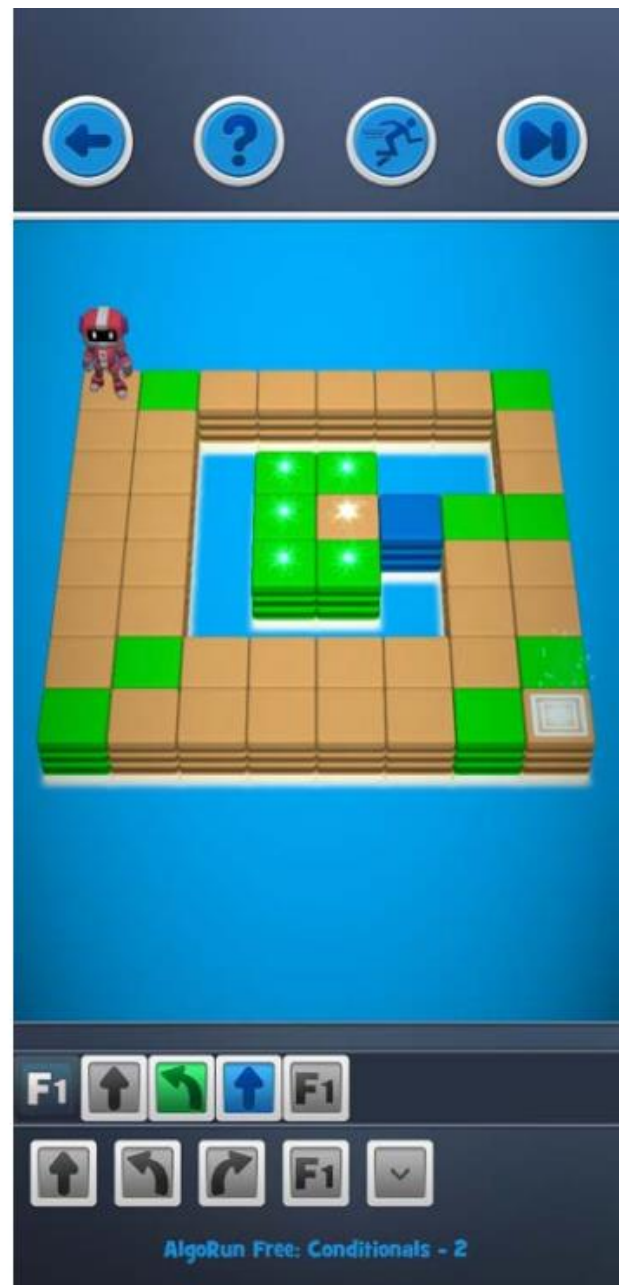
Мета гри – зібрати всі білі вогні та наступити на плитку за допомогою телепорту (рис. 2.8 а)). У грі є 4 категорії (основи, функції, рекурсія та умовні оператори) із 30 головоломками. Щоб вирішити головоломку, користувачі повинні створити невелику програму.

На рисунку 2.8 зображено дві головоломки. Рішення для обох з них містять в собі рекурсивні функції. На обох скриншотах видно, що рекурсивні функції (F1) виконують деякі інструкції та, нарешті, викликають самі себе. У реальній мові програмування написання рекурсивної функції таким чином неприйнятно, оскільки це призведе до нескінченного циклу. Однак у цих мобільних іграх із кодуванням використання цих типів рекурсивних функцій є прийнятним. Виконання програмного коду автоматично припиниться, коли робот досягне своєї мети (наступить на плитку з телепортом). В подібних головоломках кодування зустрічається проблема незавершених циклів.

У програмному кодї рішення правильної головоломки також можна побачити, як у цій грі використовуються умовні оператори («if»). Подібно до гри «LightBot», будь-яка інструкція може мати зелений або синій фон. Ці інструкції будуть виконуватися в грі «Algorun Free», тільки якщо робот стоїть на плитці того ж кольору.



а)



б)

Рисунок 2.8 – Скріншоти «Algorun Free»:

- а) із рішенням головоломки, що містить рекурсивну функцію;
- б) із рішенням головоломки, що містить два умовні оператори «if»

2.5 Порівняння реалізацій керуючих структур, функцій, змінних та подій в іграх з елементами мобільного кодування

Таблиця 2.1 показує, що створити послідовність інструкцій можна в усіх наведених іграх. Однак кількість інструкцій у послідовності дуже обмежена в усіх з них. «DinoCoding» має найбільш обмежений простір для інструкцій; зазвичай він може містити лише 4 або 5 інструкцій. Цей невеликий простір може призвести до дуже обмежених і схожих рішень для головоломок.

Таблиця 2.1 – Результати порівнянь

Особливості	«LightBot»	«Code Adventures»	«DinoCoding»	«Algorun Free»
1	2	3	4	5
Послідовність інструкцій	Так	Так	Так	Так
Умовний оператор «if»	Так	Так	Так	Так
Умовний оператор «if-else»	Ні	Ні	Ні	Ні
Умовний оператор «switch»	Ні	Ні	Ні	Ні
Цикл «for»	Ні	Так	Так	Так
Цикл «while»	Ні	Ні	Ні	Ні
Цикл «do-while»	Ні	Ні	Ні	Ні

Продовження таблиці 2.1

1	2	3	4	5
Рекурсія	Так	Так	Ні	Так
Функції	Так	Так	Так	Так
Змінні	Ні	Так	Ні	Ні
Події	Ні	Ні	Так	Ні

Переглянувши та проаналізувавши мобільні ігри з елементами програмування, можна зробити висновок, що всі наведені ігри можуть бути корисними в навчанні. Вони можуть сприяти розумінню основних концепцій комп'ютерного програмування. Порівнюючи програмні коди розв'язків головоломок із програмним кодом, написаним на реальній мові програмування, наприклад С, можна побачити багато схожостей і деякі відмінності. Важливо підкреслити деякі відмінності для студентів, що переходять від ігор до програмування реальною мовою програмування. Учні повинні розуміти, що можливість використання циклів без завершення існує тільки в іграх (і не має місця в реальних мовах програмування), а також, як правильно переписати рекурсивні функції реальною мовою програмування.

3 РОЗРОБЛЕННЯ НАВЧАЛЬНИХ ІГРОВИХ ЗАВДАНЬ З ІНФОРМАТИКИ

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи було розроблене ПЗ, яке містить у собі навчальні ігрові завдання, що несуть у собі мету покращити в учнів розуміння о роботі алгоритмів. Для реалізації було обрано IDE PhpStorm. PhpStorm засновано мовою програмування Java.

PhpStorm – це PHP IDE. Воно забезпечує запобігання помилкам на льоту, автозаповнення та рефакторинг коду, налагодження нульової конфігурації та розширений редактор HTML, CSS і JavaScript. PhpStorm також надає потужні вбудовані інструменти для налагодження, тестування та профілювання ваших програм [39].

Рефакторинг перейменування працює для файлів, функцій, констант, класів, властивостей, методів, параметрів, локальних і глобальних змінних. Вносити глобальні зміни в проект легко та безпечно. Локальні зміни вносяться миттєво на місці.

PhpStorm IDE дозволяє використовувати механізм шаблонів Smarty і пропонує наступні функції продуктивності для роботи з файлами .tpl:

- підсвічування синтаксису для ключових слів, рядків, змінних і дужок (рис. 3.1);
- підсвічування синтаксичних помилок;
- розумне доповнення функцій і атрибутів;
- автоматична вставка парних дужок, лапок і закриваючих тегів;
- коментування виділеного блоку за допомогою однієї гарячої клавіші.

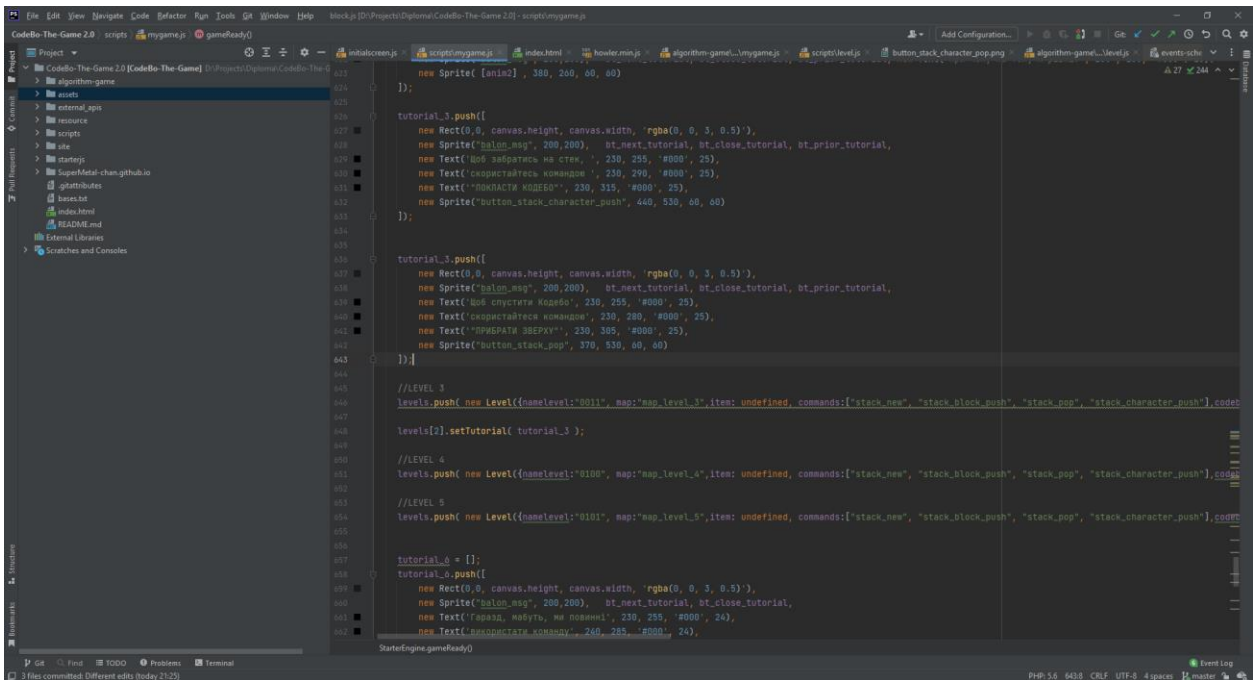


Рисунок 3.1 – Приклад інтерфейсу PhpStorm

Покроковий код і оцінка за допомогою вбудованого відладчика PhpStorm надає численні параметри для налагодження коду PHP, що дозволяє:

- встановити умовні контрольні точки за допомогою інтелектуального кроку, який дає змогу вибрати певний метод для переходу з ланцюжка викликів, перевіряти контекстно-релевантні локальні змінні та визначені користувачем спостереження, включаючи масиви та складні об'єкти, і редагувати значення на льоту;

- обчислити вираз під час виконання;
- налагодження сторінки в кількох сеансах одночасно;
- підтримувати сеанс налагодження під час переходу між сторінками.

За допомогою PhpStorm можна розробляти сучасні веб, мобільні та настільні програми за допомогою JavaScript і Node.js.

PhpStorm також підтримує React, Angular, Vue.js та інші фреймворки та забезпечує тісну інтеграцію з різними інструментами для веброботи.

Допомога в кодуванні з підтримкою JavaScript включає доповнення для ключових слів, міток, змінних, параметрів і функцій, підсвічування помилок і синтаксису, форматування, перевірку коду та швидкі виправлення, а також загальний і специфічний для JavaScript рефакторинг. PhpStorm також інтегрується з лінтерами JavaScript і перевіркою типу Flow.

За допомогою вбудованого налагоджувача можна налагоджувати як клієнтський, так і серверний код і навіть запускати фрагменти коду JavaScript в інтерактивній консолі налагоджувача. Треба зауважити, що PhpStorm підтримує налагодження JavaScript лише в Chrome або будь-якому іншому браузері родини Chrome.

PhpStorm інтегрується з платформами тестування Jest, Karma, Protractor, Cucumber і Mocha. PhpStorm підтримує запуск і налагодження тестів, а також навігацію між тестами та предметами або між невдалим тестом і фрагментом коду, який спричинив проблему.

Єдине, що повинен зробити користувач для роботи з JavaScript (або TypeScript) в IDE PhpStorm – це перевірити, чи увімкнутий відповідний плагін для роботи у налаштуваннях.

3.2 Програмна реалізація

Для програмної реалізації навчальних ігрових завдань було розроблено вебзастосунок «CodoBo».

Вебзастосунок – це ПЗ, яке можна відкрити за допомогою будь-якого браузера. Зовнішній інтерфейс вебпрограми розробляється за допомогою таких мов програмування як: HTML, CSS, JavaScript, які підтримуються на будь-якому браузері (Opera, Chrome, Mozilla тощо). У той час як для написання серверної частини (back-end) може використовуватися будь-яка інша мова програмування або фреймворк, наприклад, Python, PHP, Ruby, Java.

Основні переваги вебзастосунків:

- на відміну від мобільних додатків, для вебзастосунків не потрібно схвалення жодних платформ, щоб випустити свою програму;
- вебзастосунки це більш економний варіант для будь-якого підприємства;
- вебзастосунки можуть застосовуватися на будь-якій операційній системі (Linux, Mac, Windows), оскільки всі вони підтримують сучасні браузерери;
- у зв'язку з тим, що у вебзастосунку використовується той самий код порівняно з desktop-застосунками їх набагато легше підтримувати;
- застосунок простіше програмувати оскільки він не включає багато роботи з такими елементами, як ядро, процесор, відеокарта.

Для реалізація ПЗ з навчальними ігровими завданням доречно розробка SPA.

SPA – односторінковий інтерактивний застосунок. Важливо, що він не тільки знаходиться на одній сторінці, а й, подібно до повноцінної програми, є інтерактивним. Так інформаційний вебсайт може складатися з однієї сторінки, але, по суті, не бути SPA. В односторінковому вебзастосунку користувач, перемикаючись між вкладками, залишається на одній сторінці. Причому підвантажуються та оновлюються лише необхідні частини контенту, що грає на користь швидкості SPA.

Приклад односторінкової програми – gmail. Коли користувач перемикається між списками повідомлень, адреса сторінки не змінюється. Це відмінна ознака SPA.

Основна мова створення SPA – JavaScript. Невеликий односторінковий застосунок можна зробити за допомогою бібліотеки jQuery. Однак цей варіант не найоптимальніший для великих проектів. Найкраще використовувати фреймворки Vue, React або Angular.

Для реалізації навчальних ігрових завдань було розроблено ПЗ, що складається з декількох файлів:

- `level.js` – відповідає за відображення посібника, елементів інтерфейсу користувача (предмет, команди, блоки), консоль і відображення помилок та попереджень;
- `mygame.js` – відповідає за відображення позначок команд і інших ресурсів, анімації, підказки на сценах, ліміти команд у ланцюжках на сценах;
- `block.js` – відповідає за створення і анімацію блоків;
- `codebo.js` – відповідає за переміщення робота Кодебо, пуск, паузу, перезавантаження сцени, виконання команд, пов'язаних з стеками і предметом, обробку різних рівнів поверхі;
- `item.js` – відповідає за відображення предмету;
- `map.js` – розміщення блоків і предметів на карті;
- `initialscreen.js` – початкові анімації;
- `mainmenu.js` – інтерфейс головного меню;
- `mapmenu.js` – інтерфейс меню з вибором мап.

3.3 Інструкція користувача

Відкривши вебзастосунок, користувач після завантаження бачить головне меню (рис. А.1). Гравець може обирати серед трьох функцій: розпочати гру, подивитись довідку або змінити конфігурацію.

Якщо гравець натискає «ГРАТИ», то потрапляє до меню обирання мапи (рис. А.2). У випадку, якщо користувач починає гру в перше, для нього буде доступна лише перша сцена. Для доступу до кожної наступної мапи, треба пройти минулу. Сцени пронумеровані за бінарною системою числення. Гравець може повертатися до мап, які він вже проходив.

Під час проходження першої сцени (рис. А.3) користувач познайомиться з роботом і дізнається про основну задачу гри. Мета завдання полягає у тому, щоб допомогти роботу дістатися червоного прапора. Для

цього гравець повинен побудувати ланцюжок з команд руху і запустити його. Команди будуть виконуватись по чергово. На першій мапі користувачу доступно лише три команди: вперед, праворуч, ліворуч. Після побудови ланцюга користувач повинен натиснути на кнопку «ПУСК», щоб активувати роботу. У випадку, якщо гравець правильно проклав маршрут, сцену буде пройдено. Відповідно від кількості задіяних команд, користувач отримає від 1 до 3 зірок за проходження мапи. Після цього наступна сцена стане доступною для учня. Якщо під час складання ланцюга команд користувач допустить помилку, то він зможе побачити її подробиці в ігровій консолі (рис. 3.2).

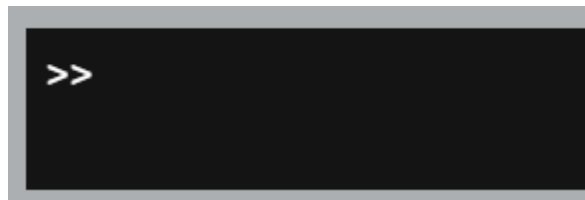


Рисунок 3.2 – Ігрова консоль, що відображає помилки

На другій мапі (рис. А.4) користувач зустріне перешкоду, так як робот не може підніматися і опускатися на різні рівні.

Третя мапа (рис. А.5) схожа на другу, але тепер користувачу доступні нові команди, а саме: створення нового стеку, додавання і прибирання блоку до стека та додавання робота Кодебо по стеку. Тепер учень повинен скористатися стеком, щоб змінити рівень, на якому знаходиться робот. Спочатку сцени гравець має 1 блок для будування стеку. Кількість можливих команд не дозволяє учню пройти рівень якимось інакше, окрім будування стеку.

На четвертій мапі (рис. А.6) гравець має одразу 4 блоки для будування напочатку.

На п'ятій мапі (рис. А.7) гравцю потрібно буде розвернути робота під час знаходження у стеку.

На шостій мапі (рис. А.8) користувач дізнається про ще одну перевагу користування командою «ПРИБРАТИ ЗВЕРХУ». Вона може як знімати

Кодебо зі стеку, так і знімати блоки зі стеку й додавати до загальної кількості блоків, які в своєму розпорядженні має учень.

На сьомій і восьмій мапах (рис. А.9, А.10) учень має лише 2 блоки на початку, тому йому треба буде ретельно скористуватися знаннями, які він отримав під час проходження минулої сцени.

Під час проходження дев'ятої сцени (рис. А.11) учень дізнається про нову команду: «ДОДАТИ ЕЛЕМЕНТ ДО СЦЕНИ». За допомогою неї він може використати міст, щоб здолати водяну перешкоду. Проте для того, щоб використати предмет, треба спочатку його підібрати, ставши на ту саму клітину, на якій знаходиться стілець.

Десята сцена (рис. А.12) є найважчою у грі й учень повинен скористатися усіма придбаними з минулих сцен вміннями, щоб здолати її.

Під час проходження навчальних ігрових завдань користувачу треба буде скористатися різноманітними командами.

Команди руху, що присутні в грі (рис. 3.3):

- рух вперед;
- поворот ліворуч;
- поворот праворуч;
- «НОВИЙ СТЕК» (створює новий стовпчик, в який можна додавати блоки і самого Кодебо);
- «ПОКЛАСТИ БЛОК» (додає блок до стеку);
- «ПОКЛАСТИ КОДЕБО» (додає робота до стеку);
- «ПРИБРАТИ ЗВЕРХУ» (прибирає останній блок або робота зі стеку);
- «ДОДАТИ ЕЛЕМЕНТ ДО СЦЕНИ» (якщо у користувача є предмет, то додає його до сцени).



Рисунок 3.3 – Команди, завдяки яким користувач вибудовує ланцюги дій

На верхній плашці інтерфейсу гри (рис. 3.4) користувач може скористуватися наступними командами: отримати довідку про сцену, повернутися до меню вибору мап, розпочати або зупинити виконання команд, перезапустити сцену, а також перевірити предмет.

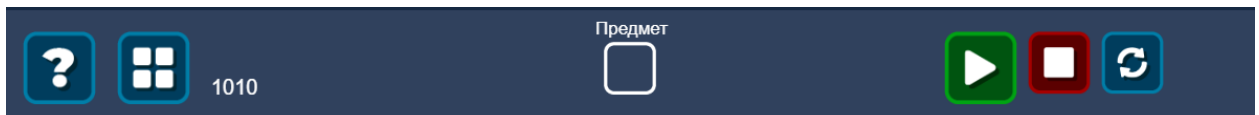


Рисунок 3.4 – Верхня частина інтерфейсу вебзастосунку

3.4 Тестування розробленої моделі

3.3.1 Загальний огляд тестування методом чорної скриньки

Для перевірки правильності роботи ПЗ було проведено тестування методом чорної скриньки.

Тестування – це один із найважливіших етапів розроблення ПЗ. Його необхідність полягає в аналізі ПЗ на відповідність стандартним вимогам та у виявленні помилок у роботі продукту.

Від тестування залежить якість роботи ПЗ, наскільки належно воно працюватиме, його особливості й зручність користування. Завдяки тестуванню можна вчасно виявити помилки у роботі програми й виправити їх.

Тестування «Black Box» (також відоме як поведінкове, непрозоре, закрите, специфікаційне тестування, тестування на очі або тестування

методом чорної скриньки) – це метод тестування програмного забезпечення, який аналізує функціональність ПЗ, не знаючи багато про внутрішню структуру/дизайн елемента, що тестується, і порівнює вхідне значення з вихідним значенням.

Основна увага при тестуванні «Black Box» робиться на функціональність системи в цілому. Термін «тестування поведінки» також використовується для тестування чорної скриньки. Дизайн поведінкових тестів дещо відрізняється від дизайну тестів із чорною скринькою, оскільки використання внутрішніх знань не є суворо забороненим, але все одно не рекомендується. Більшість застосунків перевіряються методом «Black Box».

Це тестування відбувається протягом усього періоду розробки програмного забезпечення та життєвого циклу тестування, тобто на етапах тестування модулів, інтеграції, системи, приймання та регресії.

3.3.2 Результати проведення тестування ПЗ

Було проведено декілька тест-кейсів задля перевірки належної роботи застосунку.

Перший екран, що бачить користувач при вході до вебзастосунку – це головний екран (рис. А.1). Результати тестування головного екрану показані в таблиці 3.1.

Таблиця 3.1 – Тест-кейс «Головний екран»

№	Крок	Очікуваний результат	Pass/Fail
1	2	3	4
1	Натиснути на кнопку «КОНФІГУРАЦІЯ»	Відкривається меню конфігурації параметрів гри	Fail

Продовження таблиці 3.1

1	2	3	4
2	Натиснути на кнопку «ДОВІДКА»	Відкриється довідка до проходження навчальних ігрових завдань	Fail
3	Натиснути на кнопку «ГРАТИ»	Відкриється меню вибору мапи	Pass

У результаті проведення тест-кейсу було виявлено, що такі функції головного меню, як конфігурація і довідка, не реалізовані, не дивлячись на наявність кнопок в інтерфейсі користувача. Для виправлення помилок системи треба розробити відповідні функції головного меню.

Було проведення тест-кейс запуску першого рівня з точки зору користувача, що вперше увійшов у систему. Тестування було проведено у браузерях «Google Chrome», «Opera» і «Mozilla Firefox», а його результати відображені у таблиці 3.2.

Таблиця 3.2 – Тест-кейс «Запуск першого рівня»

№	Крок	Очікуваний результат	Pass/Fail
1	2	3	4
1	Перейти на головну сторінку гри за посиланням: https://supermetal-chan.github.io/	Головна сторінка гри відкрита	Pass
2	Натиснути на кнопку «ГРАТИ»	Відкриється меню вибору мапи	Pass

Продовження таблиці 3.2

1	2	3	4
3	Натиснути на позначку мапи «А-0001»	Відкриється перша сцена з порадою	Pass
4	Натиснути на кнопку «>>»	Відображення наступної поради	Pass
5	Натиснути на кнопку «Х»	Закриття вікна з порадами і перехід до першого навчального завдання	Pass

Тест-кейс з запуску першого рівня не показав помилок, ПЗ належно відповіло на команди користувача на усіх трьох браузерях, на яких перевірялася робота ПЗ.

Перший екран, який користувач бачить після натискання кнопки «ГРАТИ» – це екран вибору мап. Вебзастосунок зберігає дані о кількості пройдених гравцем мап. Відповідно, під час першого запуску вебзастосунку користувачу буде доступна лише перша мапа («А-0001»), що показано на рисунку 3.5.

Результати проведення тест-кейсу роботи меню вибору мап наведені у таблиці 3.3.

У грі реалізовано такий елемент інтерфейсу користувача, як консоль, що показує помилку або попередження і номер елемента ланцюга, що є причиною сповіщення. Цей елемент інтерфейсу має на меті полегшити проходження сцени для користувача. Помилки й попередження відмічаються різними кольорами – червоним і жовтим відповідно. Ключовою відмінністю між цими двома видами сповіщень є те, що попередження на відміну від помилки не зупиняє процес виконання команд у грі.

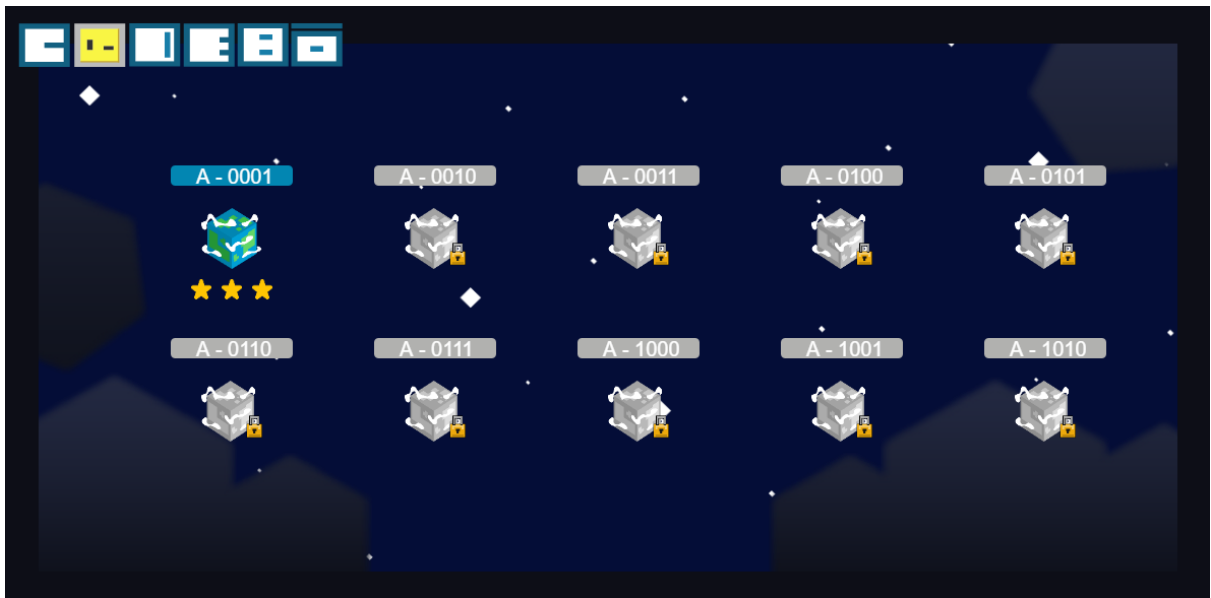


Рисунок 3.5 – Початкове меню вибору мап

Таблиця 3.3 – Тест-кейс «Меню вибору мап для нового користувача»

№	Крок	Очікуваний результат	Pass/Fail
1	Натиснути на позначки мап від «А-0010» до «А-1010»	Сцени не відкриваються, відображається анімація на позначках мап	Pass
2	Натиснути на позначку мапи «А-0001»	Відкриється перша сцена з порадою	Pass

Було перевірено відображення попереджень у консолі під час проходження першої сцени. На рисунку 3.6 показано, що робот має тільки 2 блока попереду, а через 3 блоки від робота знаходиться вода. При заповненні усіх можливих команд у ланцюжку командами «вперед», було проведено пробну активацію ланцюга команд. Починаючи з виконання третьої команди у консоль почали надходити попередження. Сцена зреагувала на команди користувача належним чином: робот пересунувся на 2 клітини вперед, було показано 20 сповіщень, сцена не була завершена. Тепер команди з ланцюга можна прибрати двома способами: натиснути на кнопку «перезавантажити

сцену» або прибрати їх поштучно, натискаючи на відповідні позначки у лівій частині екрана.

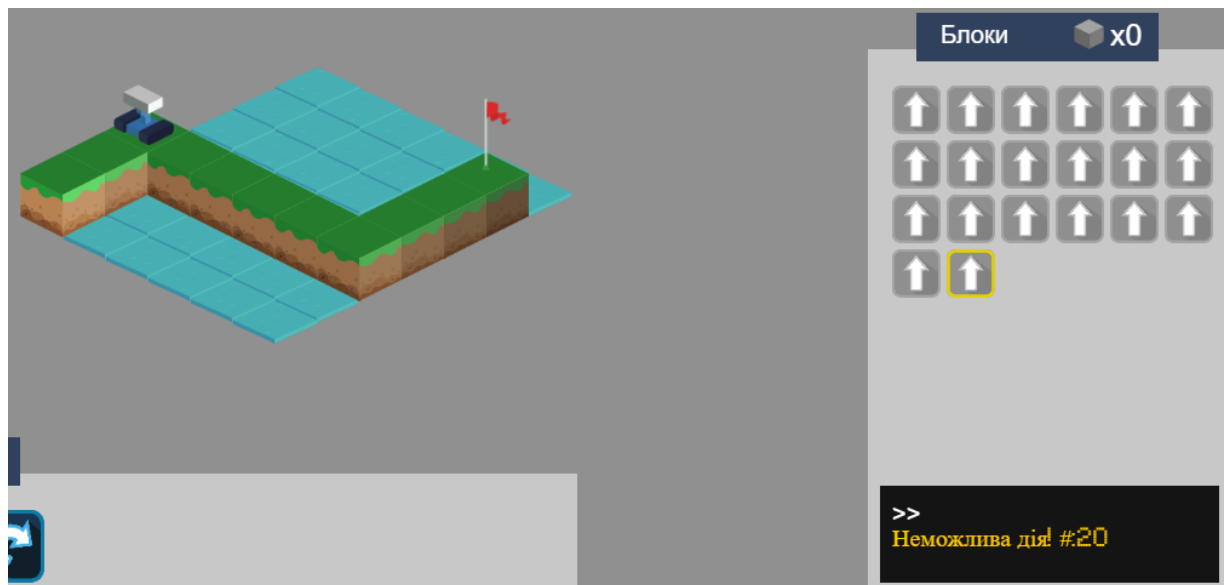


Рисунок 3.6 – Скріншот роботи програми з відображеним попередженням

Однією з найголовніших механік у грі є будування стеків за допомогою блоків. Цей ігровий механізм має широке розмаїття можливостей і тому потребує детального тестування.

Якщо користувач вирішить використати команду «ПОКЛАСТИ БЛОК» декілька разів, маючи тільки один блок, то нова команда не буде додана до ланцюга. Теж саме трапляється, якщо користувач спробує прибрати блок зі стеку за допомогою команди «ПРИБРАТИ ЗВЕРХУ», якщо перед роботом нема стеку.

Було перевірено можливість користувача прибрати блоки зі стеку за допомогою команди «ПРИБРАТИ ЗВЕРХУ» на третьому рівні, що має тільки 1 блок за замовчуванням, за такою схемою:

Крок 1. За допомогою команди «НОВИЙ СТЕК» створити новий стек.

Крок 2. За допомогою команди «ПОКЛАСТИ БЛОК» додати блок до стеку.

Крок 3. За допомогою команди «ПРИБРАТИ ЗВЕРХУ» прибрати блок зі стеку.

Крок 4. Повторити Крок 2.

При спробі помістити робота на вищій рівень землі за допомогою команди «ПОКЛАСТИ КОДЕБО» при відсутності попереду стеку, консоль видасть помилку «Неможливо залізти на стек» (рис. 3.7). Можна звернути увагу, що після виконання третьої команди, яка видала помилку, процес виконання ланцюжка припинився, а код в помилці вказує саме на третю команду, що відповідає дійсності.

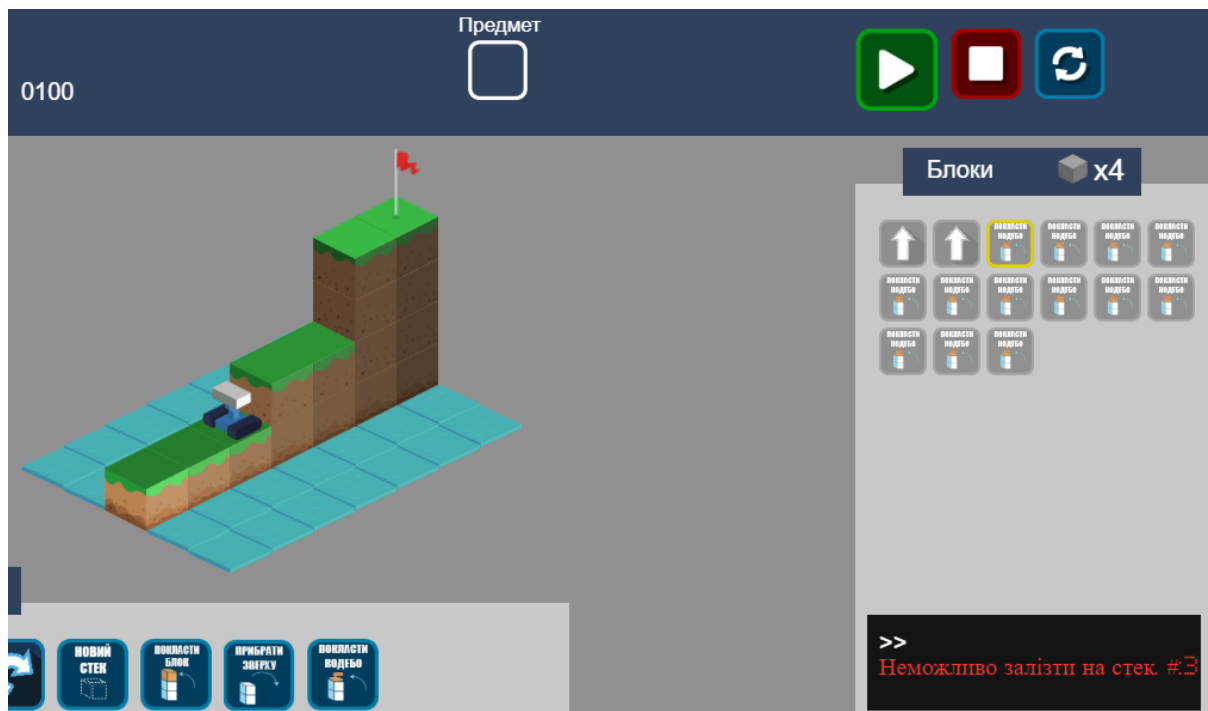


Рисунок 3.7 – Скріншот ланцюжка з консоллю з помилкою «Неможливо залізти на стек»

Якщо користувач спробує збудувати декілька стеків на одній і тій самій клітинці або збудувати стек на водній поверхні, то гра видасть попередження показане на рисунку 3.8.

Якщо спробувати покласти блок на стек, перед цим не створюючи новий стек, то програма видасть помилку «Неможливо додати блок!» (рис. 3.9). Така помилка буде зображена не дивлячись на те, яка перед роботом знаходиться поверхня, якщо на ній нема створеного стеку.

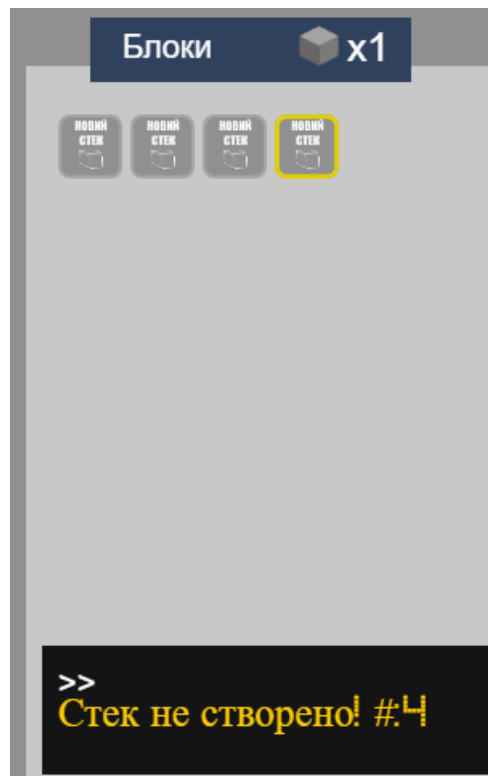


Рисунок 3.8 – Скріншот ланцюжка з консоллю з попередженням

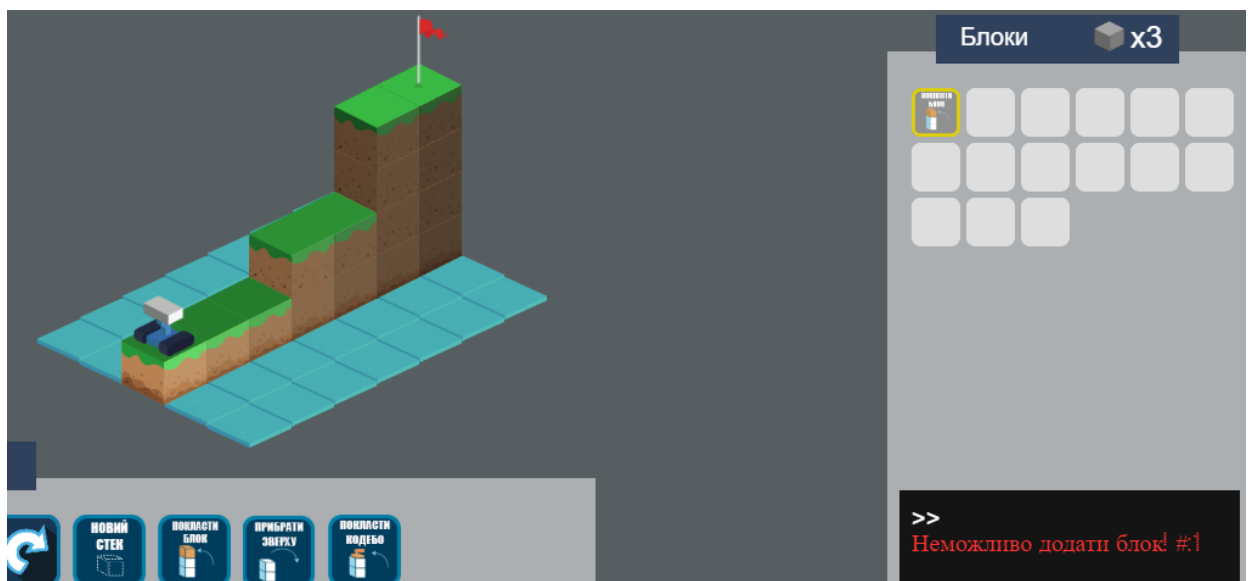


Рисунок 3.9 – Приклад помилки «Неможливо додати блок!»

Якщо спробувати покинути стек, коли попереду робота знаходиться поверхня іншого рівня або водна поверхня, то консоль видасть наступну помилку: «Неможливо покинути стек!» (рис. 3.10).

Якщо спробувати розмістити предмет не на клітинці з водою, то це не видасть помилки, хоча й зупинить виконання ланцюжка дій (рис. 3.11). Це неналежне виконання програми. Для виправлення ситуація треба додати нову помилку, яка буде відображатися тоді, коли користувач буде намагатися розмістити предмет не на водній поверхні.

Якщо під час гри користувач спробує поставити предмет у той час, коли робот знаходиться нагорі стеку, тоді програма продовжить виконання без видачі попередження (рис. 3.12), що протиречить ігровому дизайну ПЗ. Але, якщо перед Кодебо знаходиться водяна поверхня (рис. 3.13), тоді блок буде встановлено, що також не відповідає логічним умовам сцени. Рішенням даних проблем є додання помилки при спробі користувача встановити предмет, знаходячись на стеку.

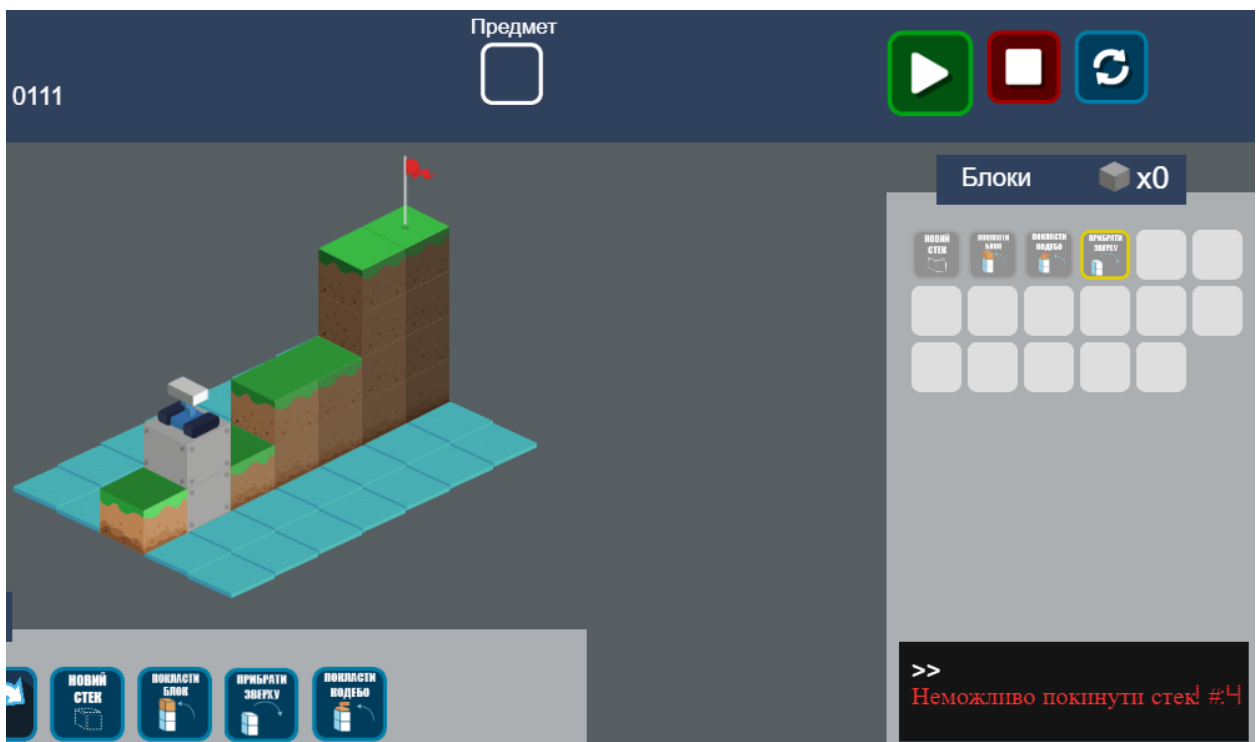


Рисунок 3.10 – Приклад помилки «Неможливо покинути стек!»



Рисунок 3.11 – Скріншот, що відображає неправильне виконання програми (відсутність помилки у консолі)



Рисунок 3.12 – Скріншот, що відображає неправильне виконання програми (некритична помилка)

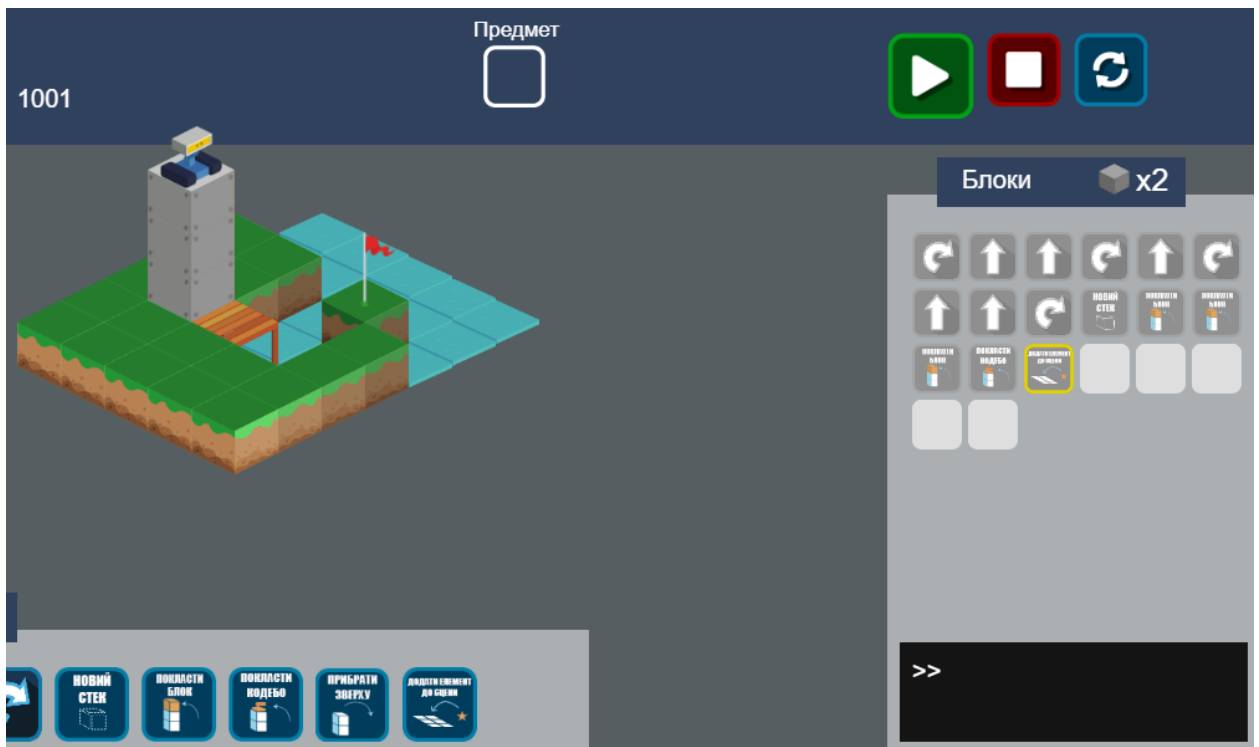


Рисунок 3.13 – Скріншот, що відображає неправильне виконання програми (критична помилка)

На останній, десятій, мапі, у ході тестування було знайдено декілька помилок у роботі програми. Якщо користувач збудує стек з блоків чотирьох блоків і спробує створити новий стек перед собою, то увесь стовпчик земляних блоків перетвориться на стек з чотирьох блоків (рис. 3.14).

У випадку, якщо користувач піде далі і перетворить третій земляний стовпчик на стек, після чого спробує взяти предмет, перемістившись на його клітину (рис. 3.15), тоді це викличе неправильну роботу програми і предмет не буде узятий, так як він знаходиться на стеку, а робот перемістився на клітинку з предметом тільки зійшовши зі стеку.

Наостанок було розглянуто ситуацію, у якій користувач спробує взяти предмет і розташувати його над водяною клітинкою, а потім виконати команду «вперед». У результаті виконання подібного ланцюжку команд програма допустить одразу 2 помилки: предмет буде розміщений на водяній клітинці не дивлячись на те, що робот знаходиться на три рівня вище за

потрібний, а сам робот, після виконання програми «вперед», опиниться «у повітрі» (рис. 3.16).

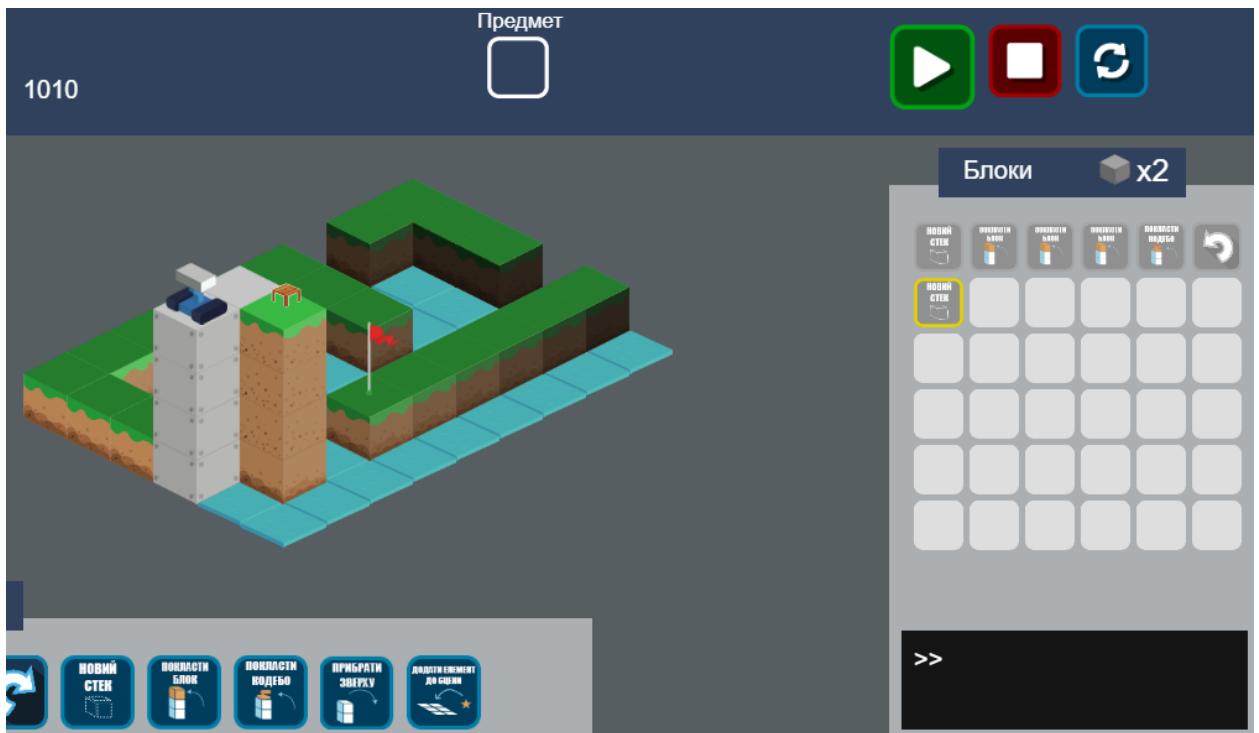


Рисунок 3.14 – Скріншот гри, що зображає неправильну роботу програми



Рисунок 3.15 – Робот не може взяти предмет

Для виправлення описаних проблем треба ввести помилку в разі, якщо користувач намагається розмістити предмет у неналежному місці, а також, якщо користувач намагається стати на предмет, знаходячись на рівнях, вище за міст.

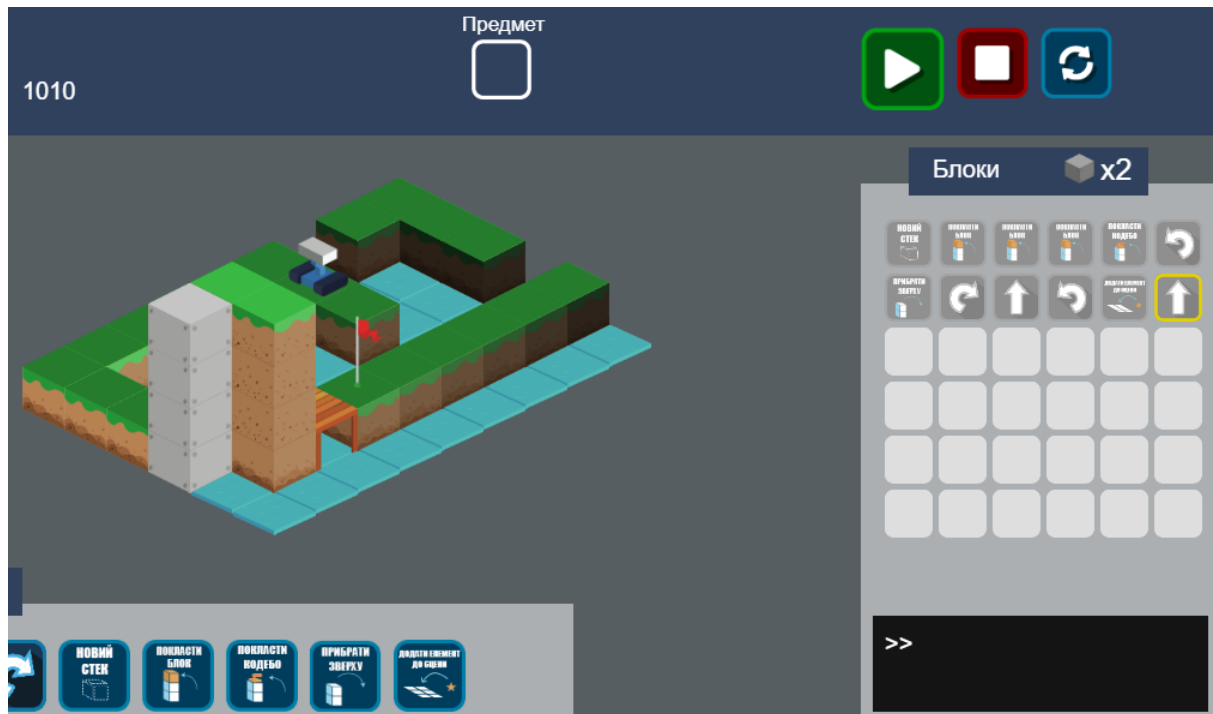


Рисунок 3.16 – Робот знаходиться «у повітрі»

У результаті проведення тестування було виявлено наступні помилки роботи ПЗ:

- відсутній функціонал для кнопок «КОНФІГУРАЦІЯ» і «ДОВІДКА»;
- відсутність помилок і попереджень у консолі при деяких ситуаціях у грі;
- хибна взаємодія робота з предметом і стеками в указаних сценаріях;
- місцями інтерфейс гри замалий, щоб розгледіти написи.

Усі знайдені при проведенні тестування помилки повинні бути виправлені перед вводом застосунка до навчальної програми з інформатики.

ВИСНОВКИ

У рамках кваліфікаційної роботи була запропонована і реалізована методологія розроблення навчальних ігрових завдань з інформатики для мобільних пристроїв.

Інтерес до вивчення інформатики багато в чому залежить від того, як саме проходять заняття. Навіть на найкращих уроках елемент інтерактивності збереже розвиток захопленості предметом. Тому на інформатиці потрібно якнайширше застосовувати нетрадиційні освітні технології.

У ході дослідження існуючих навчальних ігрових завдань було виявлено аспекти ігор, завдяки яким навчання інформатиці учнів стане більш ефективним і оригінальним. Можна зробити висновки, що використання ігрового ПЗ під час занять може покращити результати учнів. Враховуючи те, що електронне і мобільне навчання як у світі, так і в Україні кожен рік набирає більшої популярності, задача розробки для ПЗ є актуальною.

Було розроблено вебзастосунок, який можна запускати як на мобільних пристроях, так і на будь-яких інших пристроях, які надають можливість користуватися браузером. Реалізовано гру з 10 навчальними ігровими завданнями різної складності, в яких користувач ознайомиться з принципами функціонування стеку, а також почергового виконання команд програмою. Для проходження цих завдань гравцю треба розробити алгоритми проходження головоломок, що також є необхідною навичкою. Якщо користувача виникнуть складнощі з проходженням завдань, він зможе звернутися до довідки й отримати поради, щодо можливості складання алгоритмів у грі.

Перспективою дослідження є розробка повноцінного мобільного додатку і розширення функцій, які допоможуть користувачам засвоїти більше знань з інформатики (умовні оператори, цикли).

Результати дослідження апробовано у вигляді тез доповідей під час «VII Міжнародної науково-практичної конференції «Innovative areas of solving problems of science and practice» [40].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дичківська, І. М. (2004). Інноваційні педагогічні технології: навч. посібник.
2. Кукушкин, В. (2006). Ігрові технології на уроках, Відкритий урок: розробки, технології, досвід: Науково-методичний журнал, 11(12), С. 3–9.
3. Ahmad M.A., Gorokhovatskyi V., Tvoroshenko I., Vlasenko N., Mustafa S.K. (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, *International Journal of Engineering Trends and Technology*, 69(10), pp. 186-192.
4. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40–48.
5. Творошенко І.С., Табашник В.А. (2018) Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків, *Збірник наукових праць Харківського національного університету Повітряних Сил*, 1(55), С. 122–128.
6. Tvoroshenko I. (2019) Development of models of spatial analysis of status of interactive processes of complex systems.
7. Tvoroshenko I.S., and Kramarenko O.O. (2019) Software determination of the optimal route by geoinformation technologies, *Radio Electronics Computer Science Control*, 3, pp. 131-142.
8. Dicheva, D., Hodge, A. (2018). Active Learning through Game Play in a Data Structures Course, SIGCSE '18: Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 18, pp. 834–839.
9. Plass, J. L., Mayer, R. E., Homer, B. D. (2020). Handbook of Game-Based Learning. Boston: The MIT Press.

10. Lucas, F. M. (2017). The Game as an Early Childhood Learning Resource for Intercultural Education, *Procedia – Social and Behavioral Sciences*, 237, pp. 908–913. 10.

11. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069–6084.

12. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785–1797.

13. Tvoroshenko I., and Tkachenko D. (2020) Mechanisms of image classification based on descriptors of local features, *Abstracts of IV International Scientific and Practical Conference «Integration of scientific bases into practice» (October 12-16, 2020). Stockholm, Sweden*, pp. 443–448.

14. Tvoroshenko I., and Zarivchatskyi R. (2020) Analysis of existing methods for searching object in the video stream, *Abstracts of VI International Scientific and Practical Conference «About the problems of science and practice, tasks and ways to solve them» (October 26-30, 2020). Milan, Italy*, pp. 500–505.

15. Sun-Lin, H. Z., Chiou, G. F. (2019). Effects of gamified comparison on sixth graders' algebra word problem solving and learning attitude, *Educ. Technol.*, 22, pp. 120–130.

16. Van, R. R., Zaman, B. (2017). Why gamification fails in education and how to make it successful: introducing nine gamification heuristics based on self-determination theory, *Serious games and edutainment applications*, pp. 485–509.

17. Творошенко І.С. (2018) Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем, *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*, pp. 118–121.

18. Кучеренко Є.І., Творошенко І.С., Анопрієнко Т.В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки, *Системи обробки інформації*, № 2, С. 76–82.

19. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.
20. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Сучасні інформаційні системи*, 6 (3), С. 5–12.
21. Liu, E. Z., Chen, P. K. (2013). The Effect of Game-Based Learning on Students' Learning Performance in Science Learning – A Case of 'Conveyance Go,' *Procedia - Social and Behavioral Sciences*, 103, pp. 1044–1051.
22. Кучеренко Е.И., Творошенко И.С. (2010) Прикладные аспекты моделирования нечетких процессов в сложных системах, *Збірник наукових праць Харківського університету Повітряних сил*, 1(123), С. 127–131.
23. Istiono, W. (2019). Education Games To Learn Basic Algorithm With Near Isometric Projection Method, *INTERNATIONAL JOURNAL OF ADVANCED STUDIES IN COMPUTER SCIENCE & ENGINEERING*, 8(7), pp. 6–10.
24. Творошенко, И. С. (2010). Анализ процессов принятия решений в интеллектуальных системах. *Системы обработки информации*, (2), 248-253.
25. Жебровський, Б. М., Литовченко, І. В. (2007). Формування культури користування мобільним зв'язком. Методичні рекомендації, Київ.
26. Peters, K. (2007). M learning: Positioning Educators for a Mobile, Connected Future, *The International Review of Research in open and Distance Learning*, 8(2), pp. 1–18.
27. Tvoroshenko I., and Dziubenko M. (2020) Modern methods of analysis of the movement scheme using video detection of vehicles, *Abstracts of V International Scientific and Practical Conference «Study of modern problems of civilization» (October 19-23, 2020). Oslo, Norway*, pp. 422–428.
28. Brodsky, M (2003). E-learning Trends Today and Beyond, *LTi Newslines*. Retrieved from URL:

<http://www.ltimagazine.com/ltimagazine/article/articleDetail.jsp?id=56219> (дата звернення 09.11.2022).

29. Mentor, D. (2019). Advancing Mobile Learning in Contemporary Educational Spaces.

30. Zheng, R. (2020). Cognitive and Affective Perspectives on Immersive Technology in Education, *Advances in Educational Technologies and Instructional Design*, 1(1), pp. 292–329.

31. Baruch, A. F. (2019). Mobile Technologies in Educational Organizations, *Advances in Educational Technologies and Instructional Design*, 1, pp. 255–324.

32. Giannakoulas, A, Xinogalos, S. (2019). Current Trends in One-line Games for Teaching Programming Concepts to Primary School Students.

33. Duarte, E. V., Gutierrez, E. C., Aedo, M. (2017). When the Robot meets the Turtle: a Gentle Introduction to algorithms and functions.

34. Akcaoglu, M., Jensen, L. J., Gonzalez, D. (2021) Understanding Children's Problem-solving Strategies in Solving Game-based Logic Problems, *International Journal of Technology in Education and Science*, 5(2), pp. 245–257.

35. Seelow D. (2021) *Teaching in the Game-Based Classroom: Practical Strategies for Grades 6-12*. London: Routledge.

36. Code Adventures. URL: <https://play.google.com/store/apps/details?id=com.cyborc.codeadventures> (дата звернення: 09.11.2022).

37. Dinosaur Coding - Coding Games for kids. URL: <https://play.google.com/store/apps/details?id=com.imayi.dinocode> (дата звернення: 09.11.2022).

38. Algorun Free: Coding game. URL: <https://play.google.com/store/apps/details?id=com.bitcrumbs.AlgoRunFree> (дата звернення: 09.11.2022).

39. PhpStorm: The Lightning-Smart IDE for PHP Programming by JetBrains. URL: <https://www.jetbrains.com/phpstorm> (дата звернення: 09.11.2022).

40. Проценко А. (2022) Дослідження та реалізація методології розроблення навчальних ігрових завдань з інформатики для мобільних пристроїв, *Abstracts of VII International Scientific and Practical Conference «Innovative areas of solving problems of science and practice» (November 8-11, 2022). Oslo, Norway*, pp. 660-663.