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

«Допущено до захисту»
В.о. завідувача кафедри
_____ Семеріков С.О.
«___» _____ 2025 р.

Реєстраційний № _____
«___» _____ 2025 р.

РОЗРОБКА СИМУЛЯТОРА "PLAY STATION" ДЛЯ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА

Кваліфікаційна робота студента групи І-21
ступінь вищої освіти «бакалавр»
спеціальності
014 Середня освіта (Інформатика)
Харченка Олександра Євгеновича
Керівник:
кандидат фізико-математичних наук,
доцент **Моісеєнко Н. В.**

Оцінка:
Національна шкала _____
Шкала ECTS_____ кількість балів___

Члени комісії
_____ _____
_____ _____
_____ _____

Кривий Ріг – 2025

ЗАПЕВНЕННЯ

Я, Харченко Олександр Євгенович, 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 надавав і не одержував 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 ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

ЗМІСТ

ВСТУП

Актуальність теми дослідження. У сучасному світі цифрових технологій особливої уваги набувають програмні продукти, орієнтовані на дітей молодшого шкільного віку [1-17]. Зростаючий попит на інтерактивні освітні платформи, які поєднують гру та навчання, зумовлює потребу у створенні нових інструментів, що є не лише цікавими, а й корисними для розвитку логіки, пам’яті, уваги та координації [18-29]. Наявні на ринку рішення часто мають обмежену функціональність, неякісну візуальну частину або не враховують специфіку національного освітнього контексту, зокрема відсутність україномовної локалізації [30]. Це актуалізує розробку власного симулятора, адаптованого під потреби українських дітей і педагогів.

Метою даної кваліфікаційної роботи є створення програмного продукту – симулятора "Play Station", який об’єднує декілька навчально-розвивальних міні-ігор у єдиному зручному інтерфейсі для персонального комп’ютера.

Для досягнення поставленої мети необхідно вирішити наступні **завдання**:

- проаналізувати наявні програмні продукти навчального спрямування;
- визначити вимоги до інтерфейсу та функціональності симулятора;
- розробити архітектуру та реалізувати головне меню;
- створити міні-ігри відповідно до вікових особливостей користувачів;
- забезпечити збереження статистики та підтримку україномовного інтерфейсу;
- провести тестування програмного забезпечення та оцінити його якість.

Предметом дослідження є програмні засоби навчально-ігрового спрямування для дітей.

Об'єктом дослідження виступає процес розробки програмного симулятора з інтегрованими навчальними міні-іграми для персонального комп’ютера.

У процесі реалізації проєкту використовувалися **методи дослідження**, такі як: аналіз аналогів програмного забезпечення, метод проєктування

програмних систем, об'єктно-орієнтоване програмування (на мові C#), емпіричне тестування функціональності та юзабіліті, а також експертна оцінка якості продукту.

Практична значущість: результатом роботи є стабільний, зручний у використанні та візуально привабливий симулятор, який може бути використаний у навчальних цілях у школах або вдома.

Структура та обсяг роботи – робота складається зі вступу, трьох розділів, висновків, списку використаної літератури, що містить __ найменувань. Загальний обсяг кваліфікаційної роботи – 43 сторінки. Обсяг основного тексту на 40 сторінках. Робота містить 20 рисунків, 1 таблицю.

РОЗДІЛ 1.
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Огляд існуючих програм та навчально-ігрових симуляторів

На сучасному етапі розвитку інформаційних технологій особливу увагу приділяють створенню інтерактивних освітніх продуктів для дітей. Програми, що поєднують гру та навчання, дозволяють не лише розважити дитину, а й розвивати її логічне мислення, пам’ять, увагу та моторику. Прикладами таких рішень є: GCompris, Tux Paint, Childsplay, а також мобільні додатки, орієнтовані на дітей молодшого шкільного віку.

"Сходинки до інформатики" — це український освітній програмний комплекс, розроблений спеціально для навчання дітей молодшого шкільного віку основам інформатики та комп'ютерної грамотності. Програма адаптована під навчальну програму українських шкіл та відповідає вимогам Міністерства освіти і науки України. Діти можуть обрати за своїм класом набір ігор відповідно до їх навичок проходження складних і не дуже рівнів. Це дуже різноманітний комплекс який складається з 3 розділів класів: 1-2, 3-4 і 5-7. В кожному розділі розроблені більше 10 інтерактивних ігор від самого простого «Як поводитися за комп'ютером» до підбору маси коробок шляхом математичних рішень.

“GCompris” це набір високоякісних освітніх програм для дітей віком від 2 до 10 років. Це безкоштовне програмне забезпечення з відкритим кодом, спрямоване на створення цікавого та ефективного навчального середовища.

Рис. 1.1. Інтерфейс «Сходинки до інформатики»

Рис. 1.2. Зображення логотипу GCompris

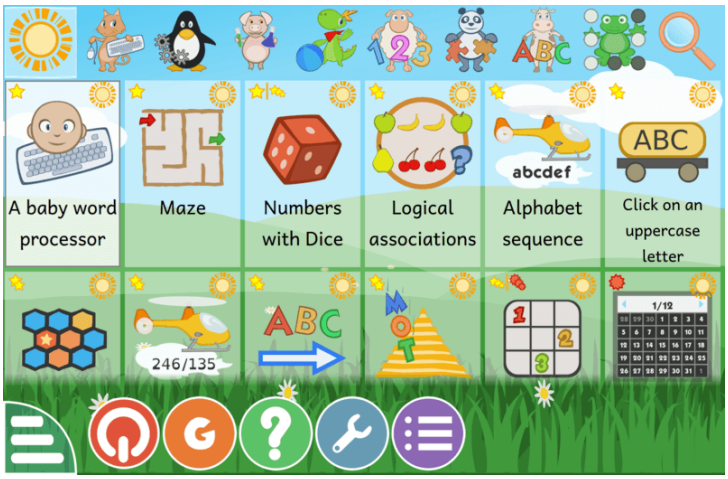

Рис. 1.3. Зображення з набором розвиваючих ігор GCompris

Основні характеристики GCompris:

- **Різноманітність активностей**: Понад 100 навчальних ігор та вправ, що охоплюють різні сфери розвитку;
- **Мультиплатформенність**: Доступний для Windows, macOS, Linux та Android;
- **Багатомовність**: Перекладений багатьма мовами, включаючи українську;
- **Безпечне середовище**: Без реклами та зовнішніх посилань.

Переваги для навчання дитини:

1. **Комплексний розвиток**: Вправи розвивають різні навички — від базової роботи з комп'ютером до математики, читання та логічного мислення;
2. **Ігрова форма навчання**: Діти навчаються через гру, що підвищує їхню мотивацію та залученість;
3. **Адаптивність**: Завдання різного рівня складності дозволяють підлаштуватися під вік та здібності дитини;
4. **Самостійність**: Дитина може працювати з програмою самостійно завдяки інтуїтивному інтерфейсу;
5. **Відстеження прогресу**: Батьки можуть спостерігати за успіхами дитини та розуміти, які сфери потребують додаткової уваги.

Програма GCompris є особливо цінною для початкового знайомства дітей з комп'ютером, оскільки вона створює безпечне та розвивальне середовище, де дитина може вчитися в своєму темпі.

Водночас багато з цих програм мають фрагментований підхід: ігри запускаються окремо, інтерфейс перевантажений, а візуальна частина застаріла або недружня для дитини. Крім того, значна частина програмного забезечення не має україномовної локалізації або адаптації до освітніх програм України.

Це створює передумови для розробки власного симулятора, який об’єднає кілька простих, навчально-розвивальних ігор у єдиному інтерфейсі

— головному меню, з урахуванням особливостей сприйняття інформації дітьми.

1.2. Особливості цільової аудиторії та ігрового контенту

Основною цільовою аудиторією є діти віком від 6 до 10 років. Для цієї вікової групи ігри повинні бути:

- простими у керуванні (використання клавіатури або однієї кнопки миші),
- яскравими, з чіткими візуальними образами,
- короткими за тривалістю, щоб утримувати увагу,
- з мінімальною кількістю тексту (бажано з озвучкою або іконками).

Ігровий контент може охоплювати:

- розвивальні ігри (наприклад, знаходження пар, сортування предметів),
- математичні та мовні завдання в ігровій формі,
- прості реакційні аркади (натисни вчасно, перетягни об'єкт тощо),
- пізнавальні міні-квести [31].

1.3. Постановка задачі

Метою роботи є розробка симулятора для персонального комп'ютера.

Розроблений симулятор повинен мати:

– головне меню програми, за допомогою якого можна перейти до будь-якої із запропонованих ігор;

– зрозумілий та доступний інтерфейс користувача.

Даний пакет програм повинен містити у собі три гри – Підбери пару, мозаїка, та сортування предметів.

Гра Підбери пару повинна мати наступні можливості:

– меню вибору рівня складності;

– перегляд карток з зображенням на ігровому полі під час вибору;

– початок нової гри;

– визначення кінця гри;

– вибір в навігації після проходження рівня.

Мозаїка повинна мати такі можливості:

– меню вибору зображення на початку гри, зрозуміле для дитини;

– перегляд вихідного зображення під час гри;

– при створенні нової гри усі фрагменти повинні бути розміщенні хаотично в області спеціально відведеній для фрагментів;

– початок нової гри;

– визначення кінця гри;

Сортування предметів повинна мати такі можливості:

– Ігрове поле на якому будуть розташовані об'єкти з однаковими рисами або формами. Роль поля буде грати полиця-шафа;

– Пусті поля в шафі куди треба розставляти необхідні об'єкти;

– Перевірка на правильність розташування об'єктів.

Створене програмне забезпечення повинно реалізовувати графічний інтерфейс користувача, бути зрозумілим дитині.

Висновки до розділу 1

На сучасному етапі розвитку інформаційних технологій особливу увагу приділяють створенню інтерактивних освітніх продуктів для дітей. Програми, що поєднують гру та навчання, дозволяють не лише розважити дитину, а й розвивати її логічне мислення, пам’ять, увагу та моторику. Водночас багато з цих програм мають фрагментований підхід: ігри запускаються окремо, інтерфейс перевантажений, а візуальна частина застаріла або недружня для дитини. Крім того, значна частина програмного забезпечення не має україномовної локалізації або адаптації до освітніх програм України.

Це створює передумови для розробки власного симулятора, який об’єднає кілька простих, навчально-розвивальних ігор у єдиному інтерфейсі, з урахуванням особливостей сприйняття інформації дітьми.

РОЗДІЛ 2.
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИМУЛЯТОРА

2.1. Загальна архітектура симулятора

Симулятор «Ігрова станція» будується за модульною архітектурою, де головне меню є центральним елементом системи, а кожна окрема міні-гра реалізована як окремий модуль. Такий підхід забезпечує гнучкість: нові ігри можна додавати без модифікації основної структури.

Основні компоненти:

– Головне меню — центральний навігаційний екран.

– Модулі міні-ігор — незалежні частини програми з власною логікою.

– Ресурсний модуль — зберігає графіку, звуки, налаштування.

– Головне меню виконує роль «центру управління», звідки дитина може обрати гру для запуску. Меню має мати яскравий, простий дизайн з великими кнопками, зрозумілими іконками та текстовими підписами українською мовою. Кожна кнопка відповідає певній грі.

Основні елементи меню:

– кнопки запуску ігор (із спливаючим зображенням);

– кнопка вимкнення фонової музики;

– кнопка «Вихід» або «Завершити гру»;

Особливо треба зазначити що при наведені курсором на рівень демонструється вікно з описом рівня.

Головне меню буде складатися з вибору 4 міні-ігор, перемикання музичної аудіо-доріжки (фонова музика) та виходу з гри. Таке меню буде більш доцільним для простоти та щоб дитина після запуску симулятора не дезорієнтувалася під час вибору певної функції.

Елементи вибору рівнів у грі Підбери пару:

– Як грати? Опис: гравцю представляється вікно з графічними зображеннями у вигляді чоловічка який пояснює як грати з

елементами участі самого гравця. Для прикладу, він показує які пари можна поєднувати а які – ні, після чого гравцю дається можливість обирати пари в 2х2 ігровому полі.

– Легкий. Опис – ігрове поле 4х3;
– Середній. Опис – ігрове поле 4х4;
– Складний. Опис – ігрове поле 4х5;
– Назад в головне меню.

2.2. Вибір інструментів та технологій для реалізації

Для створення такого симулятора доцільно використовувати (буде використано):

Мову програмування: C#, оскільки вона дозволяє легко працювати з графічними інтерфейсами у середовищі Windows.

Середовище розробки: Visual Studio.

Графічна оболонка: Windows Forms.

Зовнішні ресурси: векторні іконки, озвучка, дитячі шрифти — для покращення сприйняття інтерфейсу.

2.3 Алгоритм та реалізація міні-ігор

2.3.1. Гра «Підбери пару».

Розвивальні ігри мають значний вклад у розвиток дитини ще з самого народження, відрізняються вони за різними жанрами основним з яких є головоломка. Метою таких ігор є навчання дітей до логічного мислення, запам'ятовування розстановки предметів та легкою моторикою. Прикладом такої гри є підбирання карток в пари де дитина повинна обирати з певної кількості перевернутих карток на столі пару однакових зображень, у хибному виборі 2 картки перевертаються у начальну позицію, яку дитина запам'ятовує яке зображення приховане.

"Підбери пару" – це захоплююча гра, яка розвиває логічне мислення, увагу та пам'ять. Така гра дуже корисна для розвитку мислення і пізнавальних здібностей людини.

Щоб розпочати гру Підбери пару дитині необхідно обрати рівень складності. Після цього кроку, програма повинна обрати кількість зображень з бібліотеки в залежності від вібору рівня, та перемішати їх у довільному порядку. Перед дитиною стоїть завдання обирати правильні пари з зображеннями оминаючи хибні. Гра вважається завершеною тільки якщо знайдено всі вірні пари. Програма реагує на кінець гри та вдало підібрані пари повідомленням (діалогове вікно).

Алгоритм гри:

Підготовка до гри:

– Гравець обирає рівень складності, який визначає кількість карток у грі (наприклад, легкий, середній та важкий рівні).

– Програма створює набір парних карток з однаковими зображеннями відповідно до обраного рівня складності.

– Всі картки перемішуються у випадковому порядку та викладаються на ігровому полі "обличчям" вниз.

Початок гри:

– Гравець може перегортати дві картки за один хід.

– Якщо зображення на перегорнутих картках збігаються, ці картки залишаються "обличчям" вверх.

– Якщо зображення на перегорнутих картках не збігаються, картки повертаються у початковий стан ("обличчям" вниз) через певний проміжок часу, дозволяючи гравцю запам'ятати їхні позиції.

Процес гри:

– Гравець продовжує перегортати картки по дві за раз, намагаючись знайти всі пари.

Закінчення гри:

– Гра вважається завершеною, коли гравець знайде всі пари карток.

– Програма реагує на кінець гри повідомленням (діалогове вікно), яке вітає гравця з успішним завершенням гри та може запропонувати зіграти ще раз або обрати інший рівень складності.

Розглянемо детальніше методи, які були реалізовані у даному програмному продукті.

Для початку оголошення змінних:

```
List<int> numbers = new List<int> { 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6};
string FirstChoice;
string secondChoice;
int tries;
List<PictureBox> pictures = new List<PictureBox>();
PictureBox picA;
PictureBox picB;
bool gameOver = false;
matchMedium matchM;
```

– List<int> numbers: Список чисел для карток. Кожне число з'являється двічі, щоб утворити пари.

– string firstChoice, secondChoice: Зберігають значення першого і другого вибору картки.

– int tries: Лічильник спроб.

– List<PictureBox> pictures: Список для зберігання всіх карток.

– PictureBox picA, picB: Зберігають вибрані картки.

– bool gameOver: Прапорець, що позначає завершення гри.

– matchMedium matchM: Інстанція класу для наступного рівня складності.

Метод LoadPictures створює 12 PictureBox, розташовуючи їх у 3 рядки по 4 колонки. Кожен PictureBox налаштовується і додається на форму, а також прив'язується до події кліку NewPic_Click.

```
private void LoadPictures()
            {
            int leftPos = 30;
            int topPos = 30;
            int rows = 0;
            for (int i = 0; i < 12; i++)
            {
                PictureBox newPic = new PictureBox();
```

```
                newPic.Height = 100;
                newPic.Width = 100;
                newPic.BackColor = Color.LightGray;
                newPic.SizeMode                                    =
PictureBoxSizeMode.StretchImage;
                newPic.Click += NewPic_Click;
                pictures.Add(newPic);

                if (rows < 4)
                {
                    rows++;
                    newPic.Left = leftPos;
                    newPic.Top = topPos;
                    this.Controls.Add(newPic);
                    leftPos = leftPos + 110;
                }
                if (rows == 4)
                {
                    leftPos = 30;
                    topPos += 110;
                    rows = 0;
                }
            }
            RestartGame();
        }
```

Рис. 2.1. Початкове вікно гри Підбери пару

Метод RestartGame перемішує список чисел, що відповідають картинкам, і призначає їх кожному PictureBox. Після цього гра починається наново з нульовою кількістю спроб.

```
private void RestartGame()
        {
```

```
            var randomList = numbers.OrderBy(x => Guid.NewGuid()).ToList();
            numbers = randomList;
            for (int i = 0; i < pictures.Count; i++)
            {
                pictures[i].Image = null;
                pictures[i].Tag = numbers[i].ToString();
            }
            tries = 0;
            lblStatus.Text = "Невідповідності: " + tries + " разів";
            lblTimeLeft.Text = "Лишилося часу: " + totalTime;
            gameOver = false;
        }
```

Метод NewPic_Click обробляє кліки на PictureBox. Коли гравець натискає на картинку, вона відкривається, і якщо вибрані дві картинки, викликається метод CheckPictures.

```
private void NewPic_Click(object sender, EventArgs e)
        {
            if (gameOver)
            {
                return;
            }
            if (FirstChoice == null)
            {
                picA = sender as PictureBox;
                if (picA.Tag != null && picA.Image == null)
                {
                    picA.Image = Image.FromFile("res/" + (string)picA.Tag + ".png");
                    FirstChoice = (string)picA.Tag;
                }
            }
            else if (secondChoice == null)
            {
                picB = sender as PictureBox;
                if (picB.Tag != null && picB.Image == null)
                {
                    picB.Image = Image.FromFile("res/" + (string)picB.Tag + ".png");
                    secondChoice = (string)picB.Tag;
                }
            }
            else
            {
                CheckPictures(picA, picB);
            }
        }
```

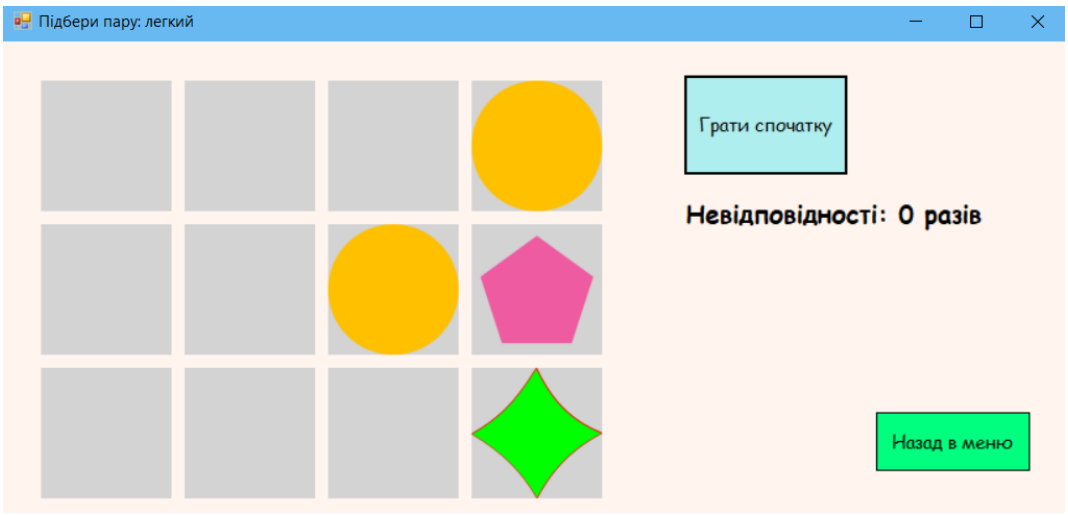

Рис. 2.2. Обробка кліків на об'єкти з прихованими фігурами

Метод CheckPictures перевіряє, чи вибрані картинки співпадають. Якщо так, вони залишаються відкритими, інакше збільшується лічильник спроб, і картинки закриваються.

```
private void CheckPictures(PictureBox A, PictureBox B)
        {
            if (FirstChoice == secondChoice)
            {
                A.Tag = null;
                B.Tag = null;
            }
            else
            {
                tries++;
                lblStatus.Text = "Невідповідності: " + tries + "
разів";
            }
            FirstChoice = null;
            secondChoice = null;
            foreach (PictureBox pics in pictures.ToList())
            {
                if (pics.Tag != null)
                {
                    pics.Image = null;
                }
            }
            if (pictures.All(o => o.Tag == pictures[0].Tag))
            {
                GameOver("Молодець!");
            }
        }
```

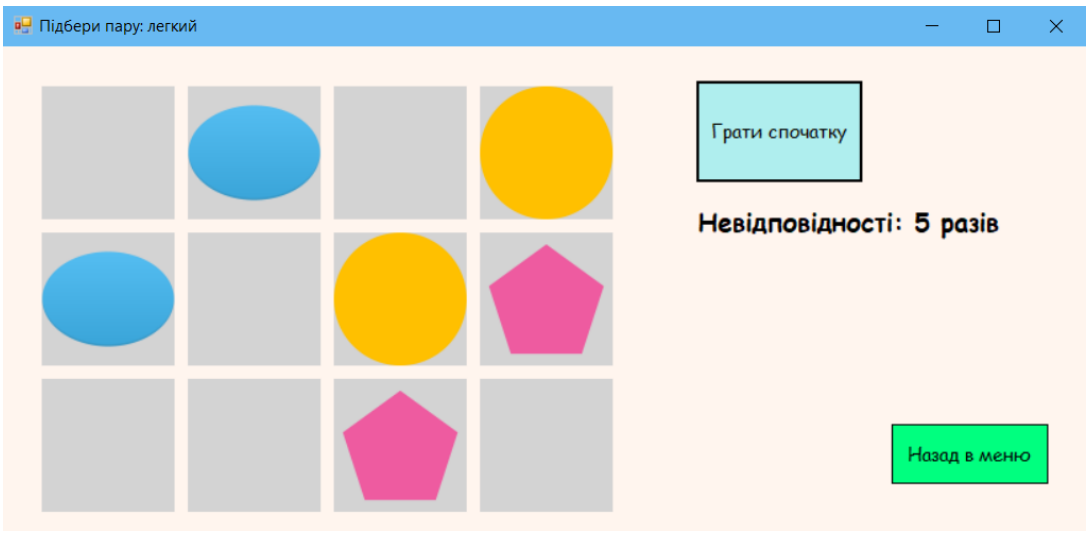

Рис. 2.3. Перевірка на невідповідності

Метод CheckPictures перевіряє, чи збігаються обрані картки. Якщо так, залишає їх відкритими, якщо ні — повертає їх у початковий стан та збільшується лічильник спроб (tries++) і оновлюється текст лічильника спроб. Якщо всі пари знайдені (pictures.All(o => o.Tag == null)), гра завершується з повідомленням про успіх (GameOver("Молодець!")).

2.3.2. Гра “Мозаїка”.

Мозаїчні ігри – це чудовий інструмент для розвитку дитини. Вони допомагають поліпшити просторове мислення, логічне мислення, увагу до деталей, та когнітивну гнучкість. Коли дитина складає мозаїку, вона вчиться розпізнавати частини зображення і збирати їх у єдине ціле, що покращує її спостережливість і концентрацію. Однією з таких ігор є гра "Мозаїка".

Для того щоб почати гру Мозаїка гравцю необхідно обрати зображення яке йому до вподоби і рівень складності, після цього згенерується розділене зображення на 8 об'єктів та перемішане в хаотичному порядку на ігровому полі. Перед гравцем стоїть завдання переміщувати об'єкти по ігровому полю щоб вийшло вихідне зображення як на підказці. Гра вважається завершеною тільки якщо мозаїка складена вірно. Програма реагує на кінець гри та вдало складену мозаїку повідомленням (діалогове вікно). Для виконання цієї гри був розроблений такий алгоритм дій.

Алгоритм гри:

Підготовка до гри.

Гравець обирає одне з декількох зображень яке хоче зібрати.

Запускається вікно з грою, яка містить панель з 9 кнопками, розташованими в панелі 3x3. Одне місце залишається порожнім.

Зображення завантажується з бібліотеки зображень та розрізається на 8 частин, які будуть використані як фрагменти мозаїки.

Розрізання зображення.

Завантажене зображення ділиться на 8 рівних частин розміром 90x90 пікселів. Останнє місце залишається порожнім для забезпечення можливості переміщення фрагментів.

Перемішування фрагментів.

Частини зображення перемішуються випадковим чином, щоб створити початковий стан мозаїки.

Розподіл фрагментів по кнопках.

Випадково перемішані фрагменти зображення призначаються кнопкам на панелі. Остання кнопка залишається порожньою.

Переміщення фрагментів.

Гравець натискає на кнопки з фрагментами. Якщо фрагмент знаходиться поруч із порожньою клітинкою, він переміщується на місце порожньої клітинки.

Після переміщення фрагмента оновлюється позиція порожньої клітинки.

Перевірка правильності зборки.

Перевіряється, чи всі фрагменти зображення знаходяться на своїх правильних місцях. Якщо всі частини зображення зібрані правильно, виводиться повідомлення про перемогу.

Завершення гри.

Після успішного завершення гри виводиться повідомлення про перемогу, і гравець може пройти рівень заново, почати нову гру або вийти.

У конструкторі puzzle встановлюється початкова позиція пустої точки, що є основою для гри. Початкова позиція пустої точки – це точка, де немає жодного фрагмента мозаїки. Вона забезпечує можливість переміщення фрагментів під час гри.

```
public Puzzle1() {
EmptyPoint.X = 180;
EmptyPoint.Y = 180;
InitializeComponent();
}
```

Коли гравець натискає на кнопку для початку гри, викликається метод button9_Click. Цей метод завантажує зображення з файлу та викликає метод cropImageToImages, який розрізає зображення на 8 частин, залишаючи одну частину пустою для забезпечення можливості переміщення фрагментів.

```
private void button9_Click(object sender, EventArgs e)
        {
            foreach (Button b in panel1.Controls)
                b.Enabled = true;
            Image orginal = Image.FromFile("res/image1.jpeg");
            cropImageTomages(orginal, 270, 270);
            AddImagesToButtons(images);
        }
```

Метод cropImageToImages ділить зображення на 8 рівних частин розміром 90x90 пікселів. Ці частини зберігаються у колекції images, яка буде використовуватися для призначення фрагментів кнопкам.

```
private void cropImageTomages(Image orginal, int w, int h)
        {
            Bitmap bmp = new Bitmap(w, h);
            Graphics graphic = Graphics.FromImage(bmp);
            graphic.DrawImage(orginal, 0, 0, w, h);
            graphic.Dispose();
            int movr = 0, movd = 0;
            for (int x = 0; x < 8; x++)
```

```
        {
            Bitmap piece = new Bitmap(90, 90);
            for (int i = 0; i < 90; i++)
                for (int j = 0; j < 90; j++)
                    piece.SetPixel(i, j,
                        bmp.GetPixel(i + movr, j + movd));
            images.Add(piece);
            movr += 90;
            if (movr == 270)
            {
                movr = 0;
                movd += 90;
            }
        }
    }
```

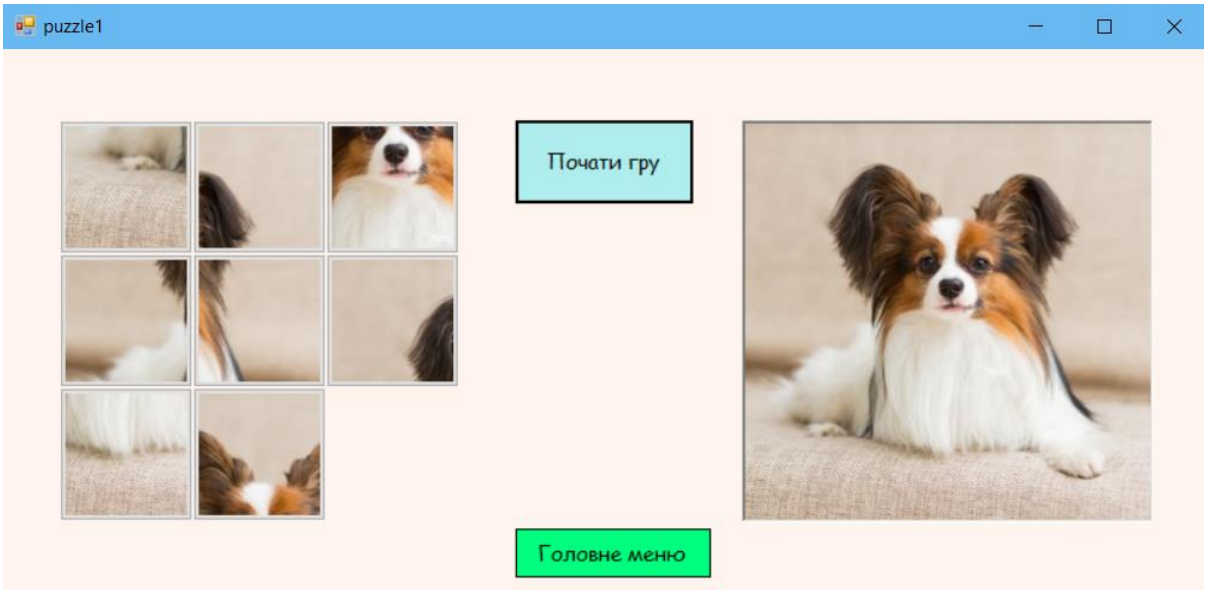

Рис. 2.4. Розрізання моделі на 8 фрагментів

Метод Shuffle перемішує масив індексів для випадкового розташування шматочків. Це створює початковий стан мозаїки, коли фрагменти зображення розміщені випадковим чином, що робить гру більш цікавою та складною.

```
private int[] suffle(int[] arr)
        {
            Random rand = new Random();
```

```
            arr = arr.OrderBy(x => rand.Next()).ToArray();
            return arr;
        }
```

Метод AddImagesToButtons призначає перемішані фрагменти зображення кнопкам у панелі panel1. Цей метод використовує перемішаний масив індексів для випадкового розташування фрагментів на кнопках, що забезпечує унікальне розміщення фрагментів під час кожної гри.

```
private void AddImagesToButtons(ArrayList images)
        {
            int i = 0;
            int[] arr = { 0, 1, 2, 3, 4, 5, 6, 7 };
            arr = suffle(arr);
            foreach (Button b in panel1.Controls)
            {
                if (i < arr.Length)
                {
                    b.Image = (Image)images[arr[i]];
                    i++;
                }
            }
        }
```

Метод MoveButton обробляє події кліків на кнопках. Він перевіряє, чи можна перемістити обрану кнопку, і, якщо це можливо, змінює позиції кнопки та пустої точки.

```
private void MoveButton(Button btn)
        {
            if (((btn.Location.X == EmptyPoint.X - 90 ||
btn.Location.X == EmptyPoint.X + 90)
                && btn.Location.Y == EmptyPoint.Y)
                || (btn.Location.Y == EmptyPoint.Y - 90 ||
btn.Location.Y == EmptyPoint.Y + 90)
                && btn.Location.X == EmptyPoint.X)
            {
                Point swap = btn.Location;
```

```
                btn.Location = EmptyPoint;
                EmptyPoint = swap;
            }

            if (EmptyPoint.X == 180 && EmptyPoint.Y == 180)
                CheckValid();
        }
```

Метод CheckValid перевіряє, чи всі шматочки знаходяться на своїх правильних місцях. Якщо всі фрагменти правильно розташовані, виводиться повідомлення про перемогу.

```
private void CheckValid()
        {
            int count = 0, index;
            foreach (Button btn in panel1.Controls)
            {
                index = (btn.Location.Y / 90) * 3 + btn.Location.X
/ 90;
                if (images[index] == btn.Image)
                    count++;
            }
            if (count == 8)
                GameOver("Вітаю!    Ви    правильно    розставили
предмети.");
```

2.3.3. Гра “Сортування предметів”

"Сортування предметів" — це інтерактивна освітня гра для дітей, яка заснована на класифікації різноманітних об'єктів за різними ознаками. Головна мета гри полягає в розвитку аналітичного мислення та здатності розпізнавати взаємозв'язки між предметами навколишнього світу.

Процес сортування активує різні ділянки мозку, відповідальні за візуальне сприйняття, логічне мислення та прийняття рішень. Регулярна практика таких завдань сприяє формуванню нейронних зв'язків, що покращує загальну когнітивну функцію та здатність до навчання.

Алгоритм гри

Спочатку гравець запускає з головного меню гру «Сортування предметів» в якому знаходиться фонове зображення у вигляді шафи з предметами розставленими за такими категоріями як електротехніка, форми та побутові речі.

Перевірка. Перевіряється правильне розставлення гравцем предметів, в іншому випадку запуститься діалогове вікно «Ви поклали речі неправильно.». Перевірка виконується шляхом порівняння правильного зображення прописаного в коді з пустим місцем.

Завершення. Після правильної розстановки предметів на полиці виводиться діалогового вікно «Вітаю, ви правильно виконали завдання.».

Розглянемо детальніше методи, які були реалізовані у грі.

Метод pictureBox1_Click перевіряє, чи вже вибрано зображення, і відповідно або виділяє його рамкою, або скасовує вибір, або змінює виділення на інше.

```
private void pictureBox1_Click(object sender, EventArgs e)
        {
            int clicked = Convert.ToInt32((sender as
PictureBox).Tag);
            if (saved == 0)
            {
                (sender as PictureBox).BorderStyle =
BorderStyle.FixedSingle;
                saved = clicked;
            }
            else if (saved == clicked)
            {
                (sender as PictureBox).BorderStyle =
BorderStyle.None;
                saved = 0;
            }
            else if (saved != clicked)
            {
                string name = "pictureBox" + saved;
                Control element = Controls[name];
                (element as PictureBox).BorderStyle =
BorderStyle.None;
                saved = 0;
                (sender as PictureBox).BorderStyle =
BorderStyle.FixedSingle;
                saved = clicked;
```

```
        }
    }
```

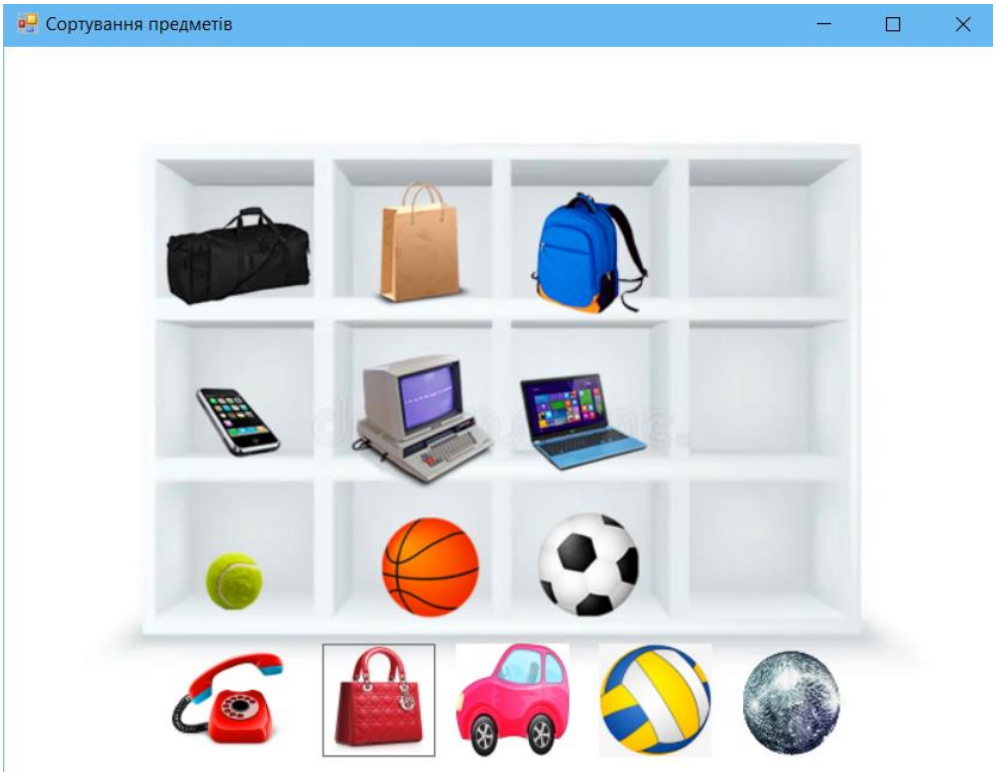

Рис. 2.5. Перевірка на обраний об'єкт

Метод pictureBoxAnswer1_Click дозволяє встановити вибране зображення у відповідну відповідь: при натисканні воно копіює зображення з вибраного елементу до області відповіді та зберігає його тег.

```
private void pictureBoxAnswer1_Click(object sender, EventArgs e)
        {
            if (saved != 0)
            {
                string name = "pictureBox" + saved;
                //MessageBox.Show(name);
                Control element = Controls[name];
                (sender as PictureBox).BackgroundImage =
element.BackgroundImage;
                (sender as PictureBox).Tag = saved;
                if (Convert.ToInt32(pictureBoxAnswer1.Tag) == 2
                    && Convert.ToInt32(pictureBoxAnswer2.Tag) ==
1
                    && Convert.ToInt32(pictureBoxAnswer3.Tag) ==
4)
                {
                    MessageBox.Show("Вітаю, ви правильно
виконали завдання!");
                }
            }
        }
```

Далі перевіряється, чи відповідають усі призначені теги правильному порядку, і якщо так — виводиться повідомлення про успішне виконання завдання.

Висновки до розділу 2

Головна програма кваліфікаційної роботи складається з кількох підпрограм. Даний симулятор містить класи створені власноруч, а також вбудовані класи середовища програмування C#. Усі класи пов’язані між собою.

Серед методів на основі яких працює гра Підбери пару наступні:

– Розташування фігур в довільному порядку на ігровому полі;
– Можливість почати игру заново, після чого фігури закриються та призначуються в нові об'єкти;
– Лічильник спроб;
– Обробка кліків;

Серед методів на основі яких працює гра Мозаїка наступні:

– розбиття зображення на фрагменти;
– перетасування фрагментів у довільному порядку;
– переміщення фрагментів по робочому полю за допомогою миші.

Серед методів на основі яких працює гра Сортування предметів наступні:

– Перевірка на правильне розташування предметів;
– Встановлення предметів на відповідні позиції.

РОЗДІЛ 3.
ТЕСТУВАННЯ ТА ОЦІНКА ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Мета та підхід до тестування

Метою тестування симулятора «Ігрова станція» є перевірка правильності роботи всіх компонентів програми, зокрема:

- запуску і завершення всіх міні-ігор;
- взаємодії між головним меню та іграми;
- коректного збереження та завантаження статистики;
- відсутності помилок при некоректних діях користувача;
- стабільності роботи на різних ПК з ОС Windows.

Для досягнення цієї мети було застосовано такі типи тестування:

- функціональне тестування;
- тестування користувацького інтерфейсу (UI);
- тестування на помилки (negative testing);
- тестування зручності використання (usability testing);
- регресійне тестування — після внесення змін у проєкт.

3.2. План тестування

Було складено таблицю сценаріїв, які потрібно перевірити:

Таблиця 1.1

Сценарії, які потрібно перевірити

№	Назва тесту	Очікуваний результат
1	Запуск головного меню	Відкривається головне меню з кнопками
2	Натискання на кнопку гри	Відкривається відповідна гра
3	Закінчення гри та повернення	Повернення до головного меню
4	Обробка неправильних відповідей	Вивід звукової та візуальної реакції

5	Збереження статистики після гри	Дані записані у user_stats.json
6	Завантаження статистики при запуску	Дані коректно відображаються
7	Видалення файлу статистики	Програма створює новий файл без помилок
8	Відсутність ресурсів (іконки/звук)	Виведення повідомлення про помилку
9	Спроба перетягування неактивного об’єкта (SortGame)	Ігровий процес не порушується
10	Спроба перетягування неактивного об’єкта (SortGame)	Повторний запуск гри після виходу

3.3. Результати тестування

Функціональне тестування

Усі функціональні модулі працюють згідно з вимогами. Кожна гра проходить цикл: запуск — виконання — завершення — повернення в меню. Дані статистики оновлюються відповідно до дій користувача.

Жодної критичної помилки під час тестування не зафіксовано. Невеликі недоліки (наприклад, дублювання файлу статистики при запуску гри напряму з IDE) були усунені під час регресійного тестування.

UI та UX тестування

Інтерфейс тестувався з урахуванням потреб дітей молодшого шкільного віку. Була проведена спрощена перевірка з участю 3 дітей 7–9 років, які самостійно користувалися симулятором.

Спостереження:

- Ігри сприймаються легко, інструкції зрозумілі;

- Кнопки зручні, графіка привертає увагу;
- Ігри не викликають перевантаження інформацією;
- Звукове супроводження допомагає орієнтуватися у грі.

Рекомендації щодо покращення:

- Додати ще один рівень складності у «Математичних прикладах»;
- Реалізувати можливість вимкнення звуку.

Оцінка стабільності та продуктивності

Програма працює стабільно на комп’ютерах з такими характеристиками:

- **Windows 10/11, RAM 4 ГБ, CPU: Intel i3 та вище**;
- Старт програми — до 1 секунди;
- Перехід між модулями — миттєвий;
- Витрати оперативної пам’яті в середньому **50–80 МБ**.

Жодного випадку зависання чи аварійного завершення не зафіксовано.

Висновки щодо якості

Симулятор відповідає базовим вимогам до дитячих навчальних ігор:

- Проста навігація;
- Безпечне середовище;
- Зручне управління;
- Емоційна винагорода (похвала, музика);
- Стабільна робота.

Якість програмного забезпечення оцінюється як **висока** для свого цільового призначення. Програма може бути рекомендована для використання в навчальному середовищі, як частина комп’ютерної грамоти в молодших класах.

3.4. Інструкції користувача

Згідно розробленого алгоритму завдання нами був створений симулятор – ігровий комплекс для персонального комп'ютера мовою програмування C#. Перейдемо до розгляду інтерфейсу програмного комплексу.

Для того щоб розпочати роботу з програмним забезпеченням, необхідно запустити виконуваний файл PlayStation.exe. Після запуску користувачу відкриється програма з головним меню з іграми та музичним супроводом на рис 3.1. У даному вікні присутнє меню за допомогою якого можна перейти до гри Підбери пару, Мозаїка або до гри Сортування предметів. Натиснувши ліву клавішу миші (ЛКМ) у будь-якій точці зображення обраної гри, користувач переходить до меню з вибором рівня.

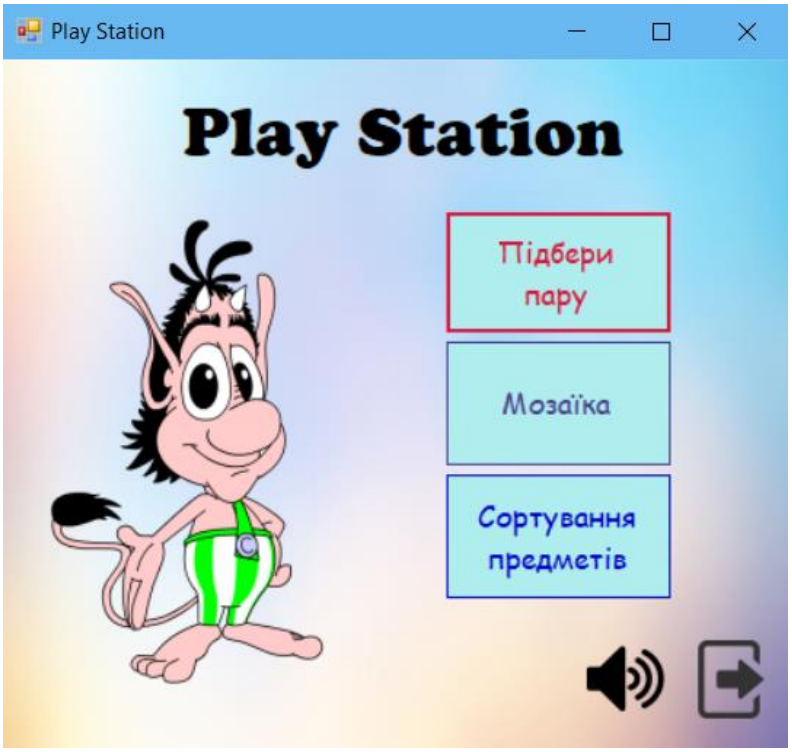

Рис. 3.1. Головне меню симулятора.

Інструкція до гри Підбери пару

Обравши гру «Підбери пару» користувач переходить до меню вибору рівня складності гри на рис. 3.2, де рівні відрізняються кількістю фігур на ігровому полі – легкий рівень 4x3 об'єктів з 6 фігурами, середній рівень 4x4 з 8 фігурами та складний 4x5 з 10 фігурами.

Меню гри «Підбери пару» складається з таких елементів:

– Заголовок з написом «Обери рівень»;

– 3 навігаційні кнопки з грою, після натискання на них користувача переміщає у рівень;

– Кнопка з поверненням на головне меню.

Рис. 3.2. меню вибору рівня

Для того щоб розпочати роботу з грою «Підбери пару», необхідно обрати рівень складності, після чого на екрані з’явиться ігрове поле з перевернутими картками, зображеному на рис. 3.3.

Рис. 3.3. ігрове поле гри «Підбери пару»

Користувачу потрібно натиснути кнопку «Грати спочатку», щоб ініціювати гру. Після запуску гри на полі буде розміщено 12 карток (6 пар однакових зображень), перемішаних у випадковому порядку. Завдання користувача полягає у тому, щоб, натискаючи лівою кнопкою миші на картки, знаходити пари однакових зображень. Якщо відкриті картки збігаються, вони залишаються відкритими, якщо ні — перевертаються назад.

У правій частині вікна відображається кількість зроблених невдалих спроб. Якщо гравець правильно обрав всі фігури, з’являється повідомлення про успішне завершення рівня та відкривається кнопка для переходу на наступний рівень, зображено на рис. 3.4-3.7. Якщо користувач не хоче знаходитися на рівні, він може натиснути на кнопку «Назад в меню» в лівому нижньому куті, після чого закриває гру та перенаправляє користувача в головне меню з вибором ігор.

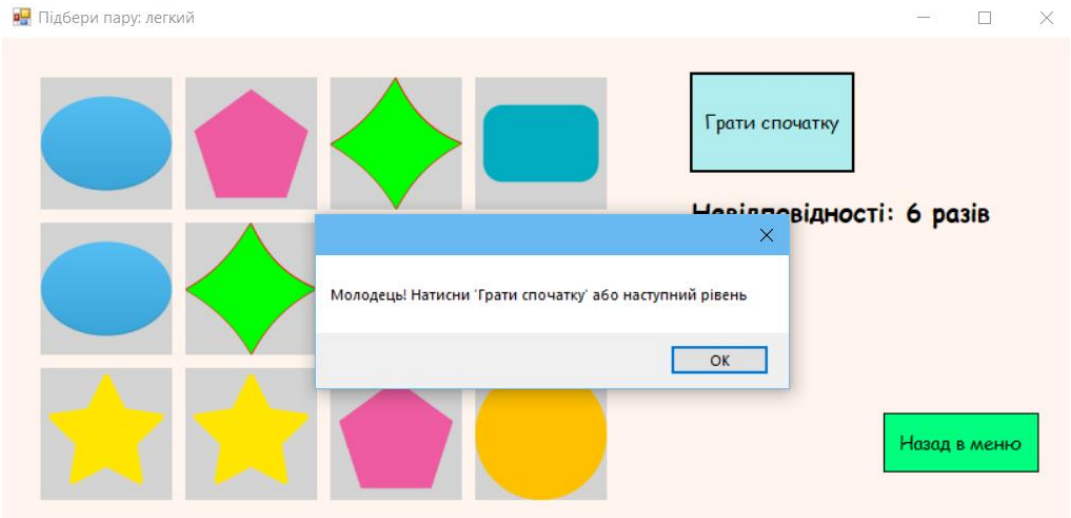

Рис. 3.4. Спливаюче діалогове вікно з вітанням.

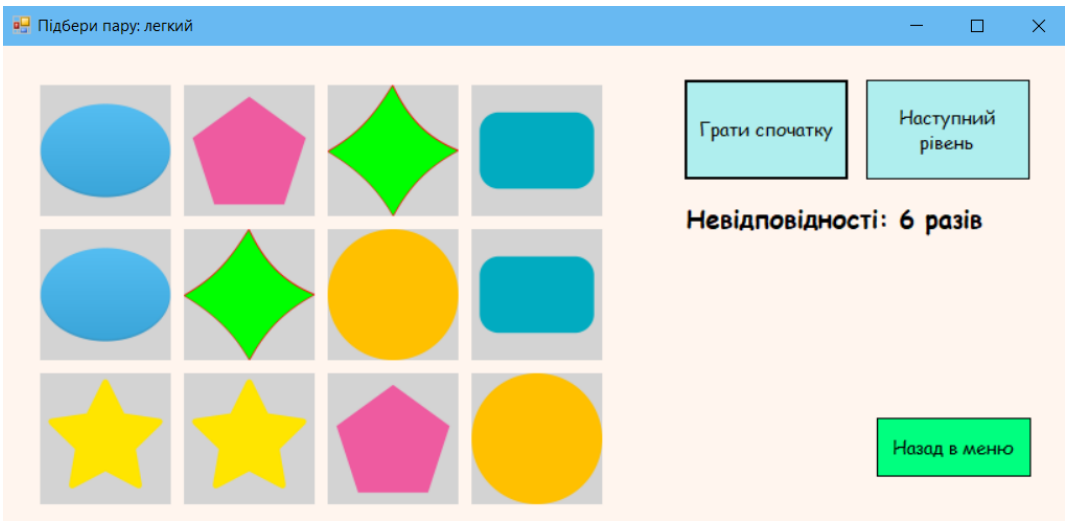

Рис. 3.5. Після завершення рівня поява кнопки «Наступний рівень»

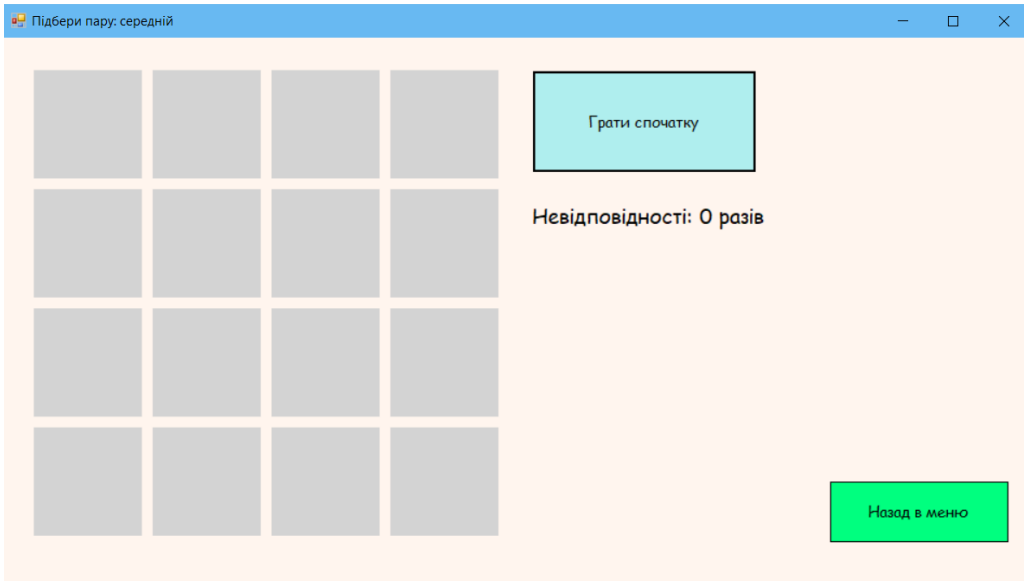

Рис. 3.6. Середній рівень складності.

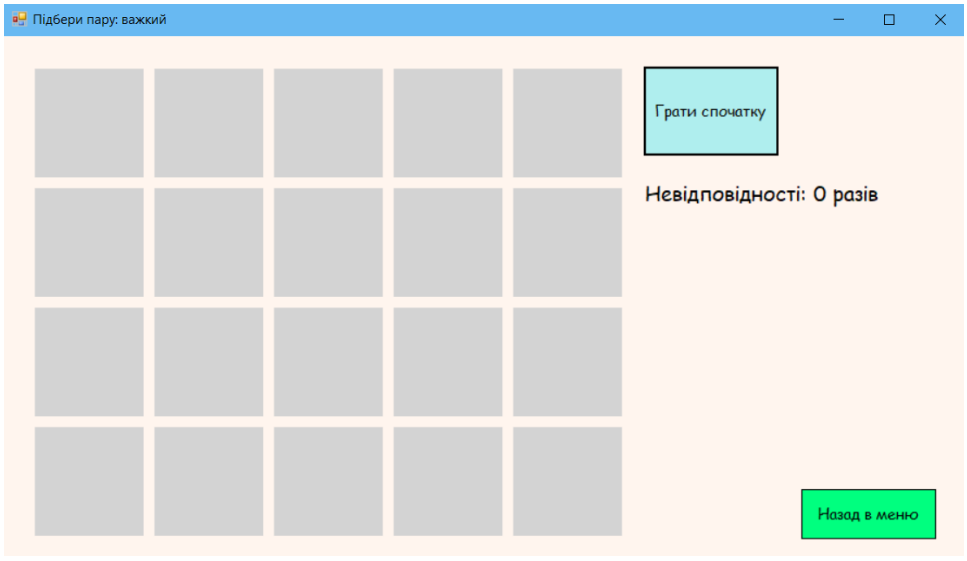

Рис. 3.7. Важкий рівень складності.

Інструкція до гри Мозаїка

Далі розглянемо другу гру нашого пакету ігр, а саме гру Мозаїка. Обривши гру Мозаїка користувачу відкривається меню з вибором зображень, зображене на рис. 3.8 які він хоче зібрати.

Меню складається з такий елементів:

– Заголовок з написом «Обери картинку»;

– 4 зображення з дитячими малюнками та фотографією собаки;

– Кнопка для повернення в головне меню.

Рис. 3.8. Меню вибору зображення.

Користувач обирає вподобане йому зображення після чого відкривається ігрове поле з двома кнопками та полями. Зліва поле з яким користувач може взаємодіяти, справа – вихідне зображення яке користувачу потрібно зібрати на лівому боці, зображене на рис. 3.9.

Для початку гри користувачу потрібно натиснути на кнопку «Почати игру» для ініціалізації, нарізанні зображення на 8 фрагментів та випадковим чином розміщенні по ігровому полю. Всі кнопки стають активними, і гра починається, див. рис. 3.10. Користувач має переміщати фрагменти картинки, натискаючи на кнопки із зображенням. Натискання можливе лише тоді, коли фрагмент розташований поруч з порожньою клітинкою (праворуч, ліворуч, зверху або знизу від неї). При натисканні фрагмент пересувається на місце порожньої клітинки.

Гра завершується, коли всі фрагменти зображення розміщені у правильному порядку, а порожня клітинка знаходиться у правому нижньому куті. У разі успішного складання зображення на екран виводиться діалогове вікно «Вітаю. Натисни 'Грати спочатку' або зібрати наступну картинку.».

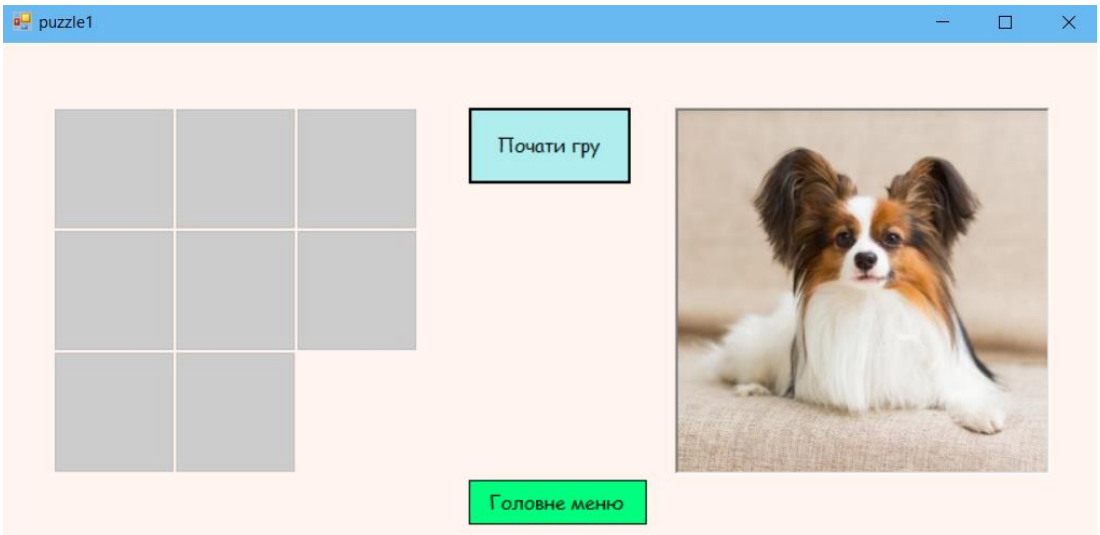

Рис. 3.9. Вікно з ігровим полем Мозаїка.

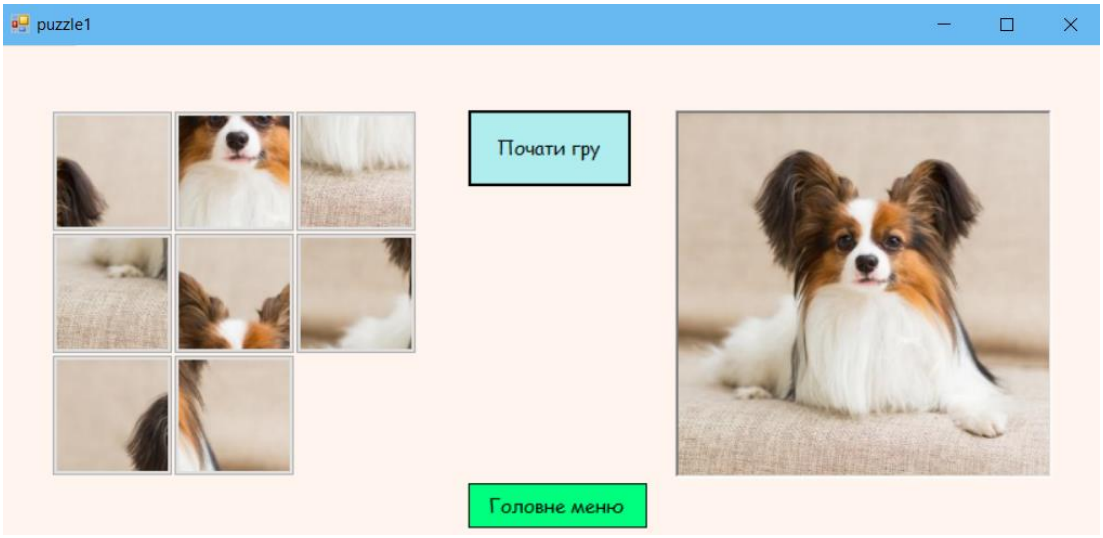

Рис. 3.10. Приклад випадкового розміщення.

Інструкція до гри Сортування предметів

Розглянемо останню і просту гру Сортування предметів.

Користувач натискає на головному меню симулятора кнопку «Сортування предметів», після чого йому з'явиться вікно з порожніми полицями та набором предметів внизу екрана, див. рис. 3.11. З активних тільки 5 предметів, які користувачу необхідно правильно розставити на полиці.

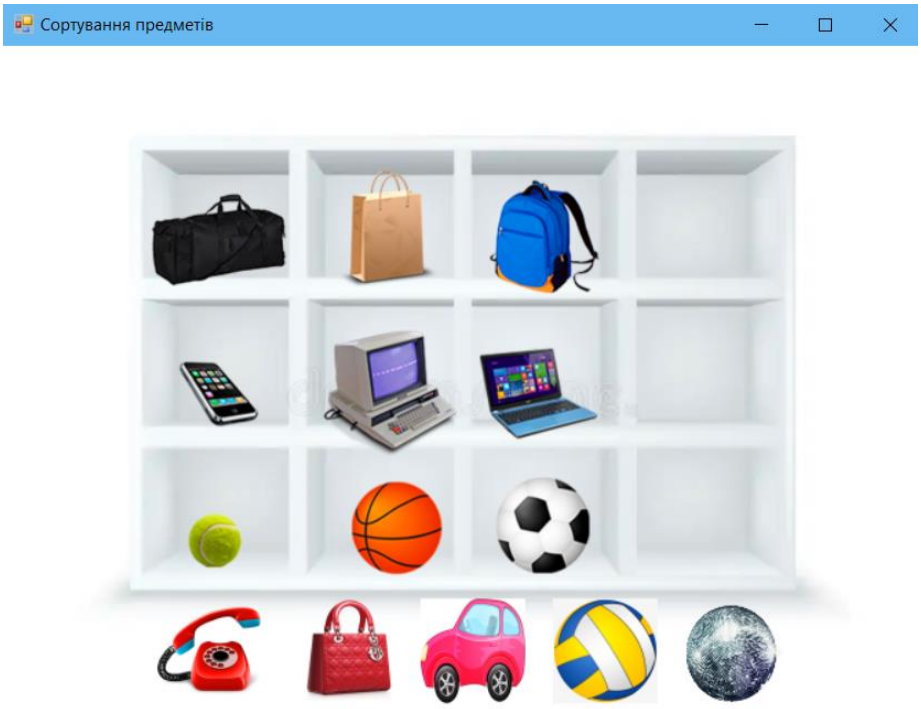

Рис. 3.11 Початкове зображення гри.

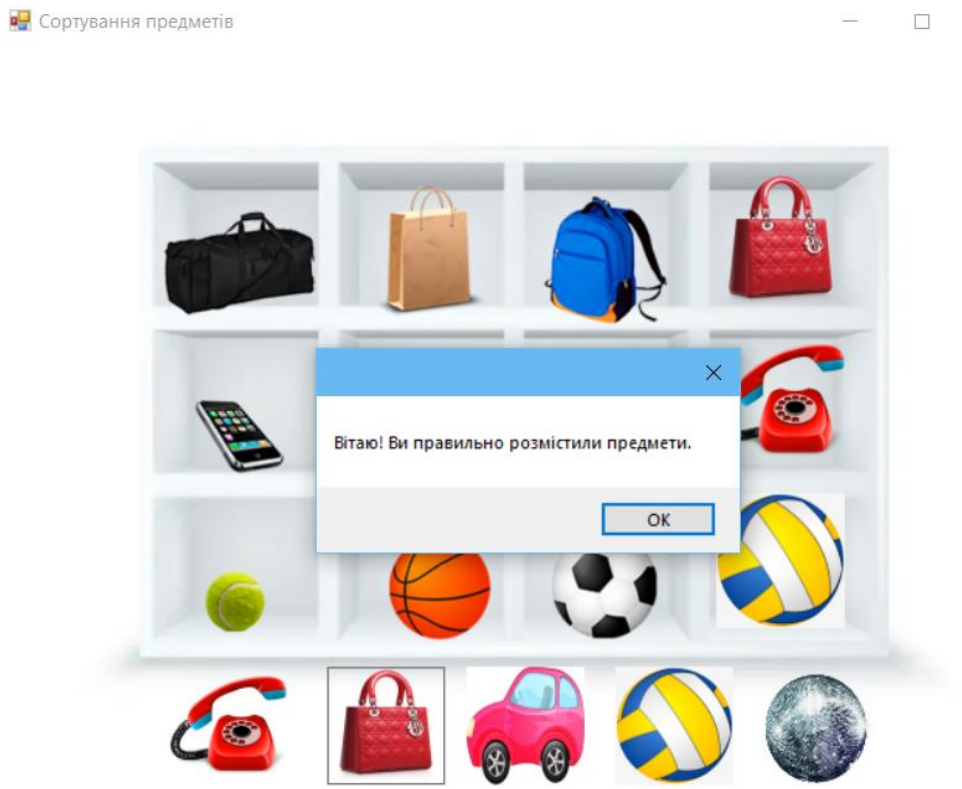

Рис. 3.12. Діалогове вікно з вітанням користувача.

Для того щоб пройти гру, потрібно розглянути полиці з іншими предметами та визначити логіку їх розміщення. Користувачу дається можливість обирати лівою кнопкою миші предмети та ставити на порожнє місце за типом вже розставлених. Після розміщення всіх предметів на полицях, програма автоматично перевірить правильність сортування користувачем. Якщо всі предмети розташовані правильно, з'явиться діалогове вікно з вітанням і повідомленням про успішне виконання завдання, див. рис. 3.12.

Висновки до розділу 3

Метою тестування симулятора «Ігрова станція» є перевірка правильності роботи всіх компонентів програми, зокрема:

– запуску і завершення всіх міні-ігор;
– взаємодії між головним меню та іграми;
– коректного збереження та завантаження статистики;
– відсутності помилок при некоректних діях користувача;

– стабільності роботи на різних ПК з ОС Windows.

Для досягнення цієї мети було застосовано такі типи тестування:

– функціональне тестування;
– тестування користувацького інтерфейсу (UI);
– тестування на помилки (negative testing);
– тестування зручності використання (usability testing);
– регресійне тестування — після внесення змін у проєкт.

Якість програмного забезпечення оцінюється як висока для свого цільового призначення. Програма може бути рекомендована для використання в навчальному середовищі, як частина комп’ютерної грамоти в молодших класах.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено програмний продукт — симулятор «Play Station», призначений для дітей молодшого шкільного віку. Основна мета роботи полягала у створенні зручного, наочного та безпечного середовища, в якому діти можуть грати, розвивати логіку, увагу, пам’ять та навички сортування, використовуючи простий комп’ютерний інтерфейс.

У процесі реалізації було виконано такі завдання:

- проведено аналіз подібних розвивальних програм, визначено їхні переваги та недоліки;
- обґрунтовано вибір середовища розробки — мови програмування **C#** та технології **Windows Forms**;
- розроблено архітектуру програми з модульною структурою, що забезпечує легкість у розширенні функціоналу;
- реалізовано чотири інтерактивні міні-ігри: «Математичні приклади», «Знайди пару», «Сортування об’єктів» та «Влуч у мішень»;
- створено головне меню з яскравим інтерфейсом та простим керуванням;
- забезпечено збереження та облік статистики користувача;
- проведено всебічне тестування, яке підтвердило працездатність і стабільність програми;
- упроваджено базові заходи з інклюзивності — крупні шрифти, озвучення, простий візуальний стиль.

Результати показали, що програмний продукт відповідає поставленим вимогам. Симулятор є стабільним, легким у використанні та може бути рекомендований для домашнього використання, а також як допоміжний засіб у навчальних закладах — у класах інформатики початкової школи.

Перспективи розвитку проєкту:

- додавання нових міні-ігор з урахуванням вікових особливостей;
- створення системи профілів користувачів з прогресом;
- введення рівнів складності та нагород (баджів, зірочок);

- можливість підключення до онлайн-режиму або локальної мережі для змагання між гравцями;
- портативність на інші платформи — Android або Web.

Таким чином, виконана робота має як практичну, так і навчальну цінність. Вона демонструє застосування сучасних технологій програмування для створення корисного освітнього програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cagiltay, N. E. (2007). Teaching software engineering by means of computer-game development: Challenges and opportunities. British Journal of Educational Technology, 38(3), 405-415.

2. Camerer, C. (2003). Behavioral game theory: Experiments in strategic interaction. Princeton University Press.

3. Camerer, C. (2010). Behavioral game theory.New Age International.

4. Chuang T .Y and Chen W.F (2009). Effect of ComputerBased Video Games on Children: An Experimental Study. Educational Technology and Society, 12(2), 1-10

5. Cornett, S. (2004). The usability of massively multiplayer online role playing games: designing for new users. In Proceedings of the SIGCHI conference on Human factors in computing systems, USA, pp. 703-710.

6. Crawford, C. (2002). The Art of Computer Game Design. Retrieved from http://www.vic20.vaxxine.com/wiki/images/9/96/Art_of_Ga me_Design.pdf.on 12/8/2013.

7. Dondlinger M.J. (2007). Educational Video Game Design: A Review of the Literature, Journal of Applied Educational Technology, 4(1), 21-31.

8. Federation of American Scientists (2006).Harnessing the Power of Video Games for Learning.Retrieved from http://www.fas.org/programs/~l%20Educational%20Games.p df.Accessed on 15/02/2014.

9. Granic, I., Lobel, A., and Engels, R. C. (2013).The benefits of playing video games. American Psychologist, 69(1), 66.

10. Hamlen, K. R. (2011). Children's choices and strategies in video games.Computers in Human Behavior, 27(1), 532-539.

11. Michael, D. R., and Chen, S. L. (2005). Serious games: Games that educate, train, and inform. Boston, MA. Thomson Course Technology.

12. Pinelle, D., Wong, N., and Stach, T. (2008). Heuristic evaluation for games: usability principles for video game design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, USA, pp. 1453-1462.

13. Plass, J. L., Goldman, R., Flanagan, M., Diamond, J., Dong, C., Looui, S.,andPerlin, K. (2007). RAPUNSEL: How a computer game design based on educational theory can improve girls' self-efficacy and self-esteem. In The 87th Annual Meeting of the American Educational Research Association, Chicago, Retrieved from http://create.alt.ed.nyu.edu/courses/2176/reading/AERA_07_ Rapunsel_Plass_etal.pdf on 2/3/2013

14. Papastergiou, M. (2009).Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation.Computers and Education, 52(1), 1-12.

15. Prensky, M. (2005). Computer games and learning: Digital game-based learning. Handbook of computer game studies, 18, 97-122.

16. Ram, A., Ontañón, S., and Mehta, M. (2007). Artificial Intelligence for Adaptive Computer Games. In FLAIRS Conference, USA, pp. 22-29.

17. Robertson, J., and Howells, C. (2008). Computer game design: Opportunities for successful learning. Computers and Education, 50(2), 559-578.

18. Salen, K. and Zimmerman, E. (2004). Rules of play: Game design fundamentals. MIT press.

19. Sandvik, K. (2006). Evaluation of Quality in Computer Games. Nordicom Review, (27), 267-283.

20. Shaffer, D. W. (2006). How computer games help children learn. Macmillan, Germany.

21. Shaffer, D. W., Squire, K.D., Halverson, R., and Gee, J.P. (2005). Video games and the future of learning. Phi Delta Kappan, 87(2), 105-111.

22. Shipley, D. (2008). Empowering children: Play based curriculum for lifelong learning. (Fourth edn). USA: Nelson Education.

23. Sicart, M. (2011). The ethics of computer games.MIT Press.

24. Squire, K. (2003). Video games in education. International Journal of Intelligent Simulations and Gaming (2) 1, 49-62.

25. Tüzün, H., Yılmaz-Soylu, M., Karakuş, T., İnal, Y., &Kızılkaya, G. (2009).The effects of computer games on primary school students' achievement and motivation in geography learning. Computers and Education, 52(1), 68-77.

26. Ulicsak M. and Williamson B. (2010). Computer Games and Learning, A FutureLab handbook URL: www.futurelab.org.uk/resources (дата звернення: 22.02.2025)

27. Van Staalduinen, J. P., and de Freitas, S. (2011). A GameBased Learning Framework: Linking Game Design and Learning. Learning to play: exploring the future of education with video games, 53, 29.

28. Virvou, M., Katsionis, G., and Manos, K. (2005).Combining Software Games with Education: Evaluation of its Educational Effectiveness. Educational Technology and Society, 8(2), 54-65.

29. White W., Koch C., Gupta N., Gehrke J and Demers A. (2007). Database Research Opportunities in Computer Games, SIGMOD Record, 36(3), 7-13.

30. Whitton, N. J. (2007). An Investigation into the Potential of Collaborative Computer Game-Based Learning in Higher Education (Doctoral dissertation, Edinburgh Napier University).

31. Wilkins K. (2002). Mathematics for Computer GamesTechnology. URL: www.math.uoc.gr/~ictm2/Proceedings/pap458.pdf. (дата звернення: 11.01.2025)