

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ДЕННОЇ ОСВІТИ

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Допускається до захисту
Завідувач кафедри _____ Ольховська О.В.
(підпис)
« _____ » _____ 2022 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОЇ РОБОТИ
на тему**

**«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З
ТЕМИ «РЕКУРСИВНІ ФУНКЦІЇ» АНГЛОМОВНОГО
ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «ТЕОРІЯ
АЛГОРИТМІВ»**

**зі спеціальності 122 «Комп'ютерні науки»
освітня програма «Комп'ютерні науки»
ступеня магістра**

Виконавець роботи Бурко Андрій Олександрович
_____ « _____ » _____ 2022р.
(підпис)

Науковий керівник к. ф.-м. н., Черненко Оксана Олексіївна
_____ « _____ » _____ 2022р.
(підпис)

ПОЛТАВА – 2022 р.

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Ольховська О.В.

« 8 » вересня 2022р.

**ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК
ВИКОНАННЯ ДИПЛОМНОЇ РОБОТИ**

Здобувач вищої освіти зі спеціальності 122 «Комп'ютерні науки»

Освітня програма «Комп'ютерні науки»

Прізвище, ім'я, по батькові Бурко Андрій Олександрович

1. Тема «Розробка програмного забезпечення тренажеру з теми «Рекурсивні функції» англomовного дистанційного навчального курсу «Теорія алгоритмів»

затверджена наказом ректора № 121-Н від «_» _____ 2022 р.

Термін подання студентом дипломної роботи «_» _____ 2022 р.

2. Вихідні дані до магістерської роботи: публікації з теми навчальні тренажери в дистанційних курсах з комп'ютерних наук.

3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1. ПОСТАНОВКА ЗАДАЧІ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд робіт з аналогічними завданням та реалізацією

2.2. Переваги та труднощі в використанні оглянутих програм.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Походження рекурсивних функцій.

3.2. Основні поняття теорії рекурсивних функцій.

3.3. Алгоритм роботи тренажера

3.4. Блок-схема програми-тренажера

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис розробки програмного забезпечення тренажеру

4.2. Інструкція для роботи з тренажером

ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

4. Перелік графічного матеріалу: 3-4 аркуші блок-схем, інші необхідні ілюстрації.

5. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Постанова задачі	<u>Черненко О.О.</u>		
2. Інформаційний огляд	<u>Черненко О.О.</u>		
3. Теоретична частина	<u>Черненко О.О.</u>		
4. Практична реалізація	<u>Черненко О.О.</u>		

6. Календарний графік виконання магістерської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доробка (за необхідністю), рецензування		

Дата видачі завдання « 8 » вересня 2022 р.

Здобувач вищої освіти

Бурко Андрій Олександрович

Науковий керівник

к. ф.-м. н., Черненко О.О.

Результати захисту дипломної роботи

Дипломна робота оцінена на

(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від « _____ » _____ 2022 р.

Секретар ЕК _____

(підпис)

(ініціали та прізвище)

Затверджую

Зав. кафедрою _____

к.ф.-м.н. О. ОЛЬХОВСЬКА

Погоджено

Науковий керівник _____

доцент, к.ф.-м.н. О. ЧЕРНЕНКО

«_____» _____ 2022 р.

«_____» _____ 2022 р.

План

Дипломної роботи здобувача вищої освіти ступеня магістра

спеціальності 122 Комп'ютерні науки

освітня програма 122 Комп'ютерні науки

Прізвище, ім'я, по батькові _____ Бурко Андрій Олександрович _____

**На тему «Розробка програмного забезпечення тренажеру з теми
«Рекурсивні функції» англomовного дистанційного навчального курсу
«Теорія алгоритмів»**

ВСТУП

1. ПОСТАНОВКА ЗАДАЧІ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд робіт з аналогічними завданням та реалізацією

2.2. Переваги та труднощі в використанні оглянутих програм.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Походження рекурсивних функцій.

3.2. Основні поняття теорії рекурсивних функцій.

3.3. Алгоритм роботи тренажера

3.4. Блок-схема програми-тренажера

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис розробки програмного забезпечення тренажеру

4.2. Інструкція для роботи з тренажером

ВИСНОВОК

Здобувач вищої освіти _____ А.БУРКО

«_____» _____ 2022 р.

РЕФЕРАТ

Записка: 58 стор., в т.ч. основна частина 54 стор., джерел - 14.

Предмет розробки – предметом даної магістерської роботи є програма-вчитель для навчання чи удосконалення знань з теми «Рекурсивні функції».

Мета роботи – Алгоритмізувати та запрограмувати англomовний тренажер з теми «Рекурсивні функції».

Методи, які були використані для розв’язування задачі – програмна реалізація виконана на платформі Unity 2022 і мові програмування C#.

ЗМІСТ

ВСТУП	8
1. ПОСТАНОВКА ЗАДАЧІ	10
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	11
2.1. Огляд робіт з аналогічними завданням та реалізацією.....	11
2.2. Переваги та труднощі в використанні оглянутих програм.	11
3. ТЕОРЕТИЧНА ЧАСТИНА	22
3.1. Походження рекурсивних функцій.	22
3.2. Основні поняття теорії рекурсивних функцій.	23
3.3. Алгоритм роботи тренажера.....	25
3.4. Блок-схема програми-тренажера.....	30
4. ПРАКТИЧНА ЧАСТИНА	31
4.1. Опис розробки програмного забезпечення тренажеру	31
4.2. Інструкція для роботи з тренажером	35
ВИСНОВОК.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТОК А.....	48

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовного позначення, скорочень, символів
Тренажер	Комп'ютерна програма, призначена для вивчення і закріплення різноманітних практичних навичок
<i>ДК</i>	Дистанційний курс.
<i>ДН</i>	Дистанційне навчання.
<i>С#</i>	Мова програмування.
<i>Unity 2022</i>	Платформа для програмної реалізації.
<i>IDE MS Visual Studio</i>	Середовище для написання програмного коду.
<i>Префаб</i>	Виступає в ролі шаблону для створення екземплярів об'єкта, що зберігається в сцені Unity.

ВСТУП

Слідуючи модернізації людського життя покращується тип та методи як онлайн- так і офлайн-навчання. Одним з прикладів модернізації є програми тренажери, що можуть за допомогою свого функціоналу навчити користувача однієї теми або цілої низки тем. Такі програми дуже зручні в користуванні адже замінюють собою повноцінне офлайн-навчання. Маючи в собі як теоретичні так і практичні завдання ці програми-вчителі забезпечують користувачеві повне опанування чи покращення знань майже з будь-якої сфери.

Аналогічних до обраного тренажеру програм в мережі інтернет та не було знайдено. Тож, тема даної курсової роботи та створення тренажеру є і необхідним для покращення знань чи навчання студентів.

Мета роботи – Алгоритмізувати та запрограмувати англomовний тренажер з теми «Рекурсивні функції».

Об'єкт роботи – програмування тренажеру для платформи MOODLE з теми «Рекурсивні функції» дисципліни «Теорія алгоритмів».

Предмет роботи – предметом даної магістерської роботи є програмування для навчання чи удосконалення знань з теми «Рекурсивні функції».

Методи роботи – роботу проведено з застосуванням платформи Unity 2022, мови програмування C#, а також середовища для написання програмного коду IDE MS Visual Studio.

Складові пояснювальної записки даної магістерської роботи:

- Титульний аркуш;
- Зміст;
- Вступ;
- Теоретична та практична частини роботи;
- Висновки;

- Рекомендації;
- Список використаних джерел;

Обсяг пояснювальної записки: 51 стор., в т.ч. основна частина 47 стор.,
джерел - 14.

1. ПОСТАНОВКА ЗАДАЧІ

Головним завданням роботи є програмування англomовного тренажеру за розробленим попередньо алгоритмом, для засвоєння теми «Рекурсивні функції» для дистанційного проходження курсу «Теорія алгоритмів».

Також для повної реалізації даного завдання потрібно забезпечити можливість інтеграції тренажеру в систему MOODLE.

Необхідні складові проведеної роботи:

- створення постановки задачі;
- розробка алгоритму та блок-схеми тренажеру;
- тестування та перевірку елементів тренажеру на різних етапах з описом процесу тестування;

Головні вимоги до програмного забезпечення:

- Тренажер повинен запускатися на комп'ютерах з будь якою встановленою ОС WINDOWS та на 32 чи 64 розрядних системах.
- При виборі неправильної відповіді тренажер повинен видавати повідомлення про помилку.
- Навігація в тренажері повинна бути зручна та зрозуміла.
- Тренажер повинен відповідати вимогам системи дистанційної освіти MOODLE та переданий для впровадження до курсу «Теорія алгоритмів».
- Різноманітність навчальних матеріалів.
- Зручний дизайн, що не відволікає від роботи з тренажером.
- Після вибору правильної відповіді користувач повинен отримати доступ до продовження роботи з тренажером.
- Після завершення роботи з тренажером користувач повинен отримати доступ до вікна тренажеру у якому може або завершити роботу через кнопку «Вихід» або повторити роботу за допомогою кнопки «Повтор».

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд робіт з аналогічними завданням та реалізацією

Для інформаційного огляду було обрано два схожих за програмною реалізацією та поставленими завданнями тренажери, що були попередньо розроблені студентами і впроваджені в заочний курс «Системного аналізу та теорії прийняття рішень» дистанційної системи навчання ПУЕТ MOODLE.

Обрані тренажери були проаналізовані на рахунок переваг та труднощів що можуть виникнути в здобувачів освіти під час тренувань. Виявлені плюси та мінуси будуть враховані при розробці тренувального засобу навчання з обраної теми.

Вибраними оглядовими роботами стали:

1. Тренажер з теми «Метод гілок та меж в задачі про найкоротший шлях» за розробки Ставкового М.Ю. [2]
2. «Знаходження вектору пріоритетів та відношення узгодженості в методі аналізу ієрархій» за авторства Гусар М.В для дистанційного курсу «Системний аналіз та теорія прийняття рішень» [3].

2.2. Переваги та труднощі в використанні оглянутих програм.

На початку роботи з тренажером «Метод гілок та меж в задачі про найкоротший шлях» перед користувачем з'являється стартове вікно з основною інформацією та кнопкою «Розпочати» (див. Рисунок 2.1)

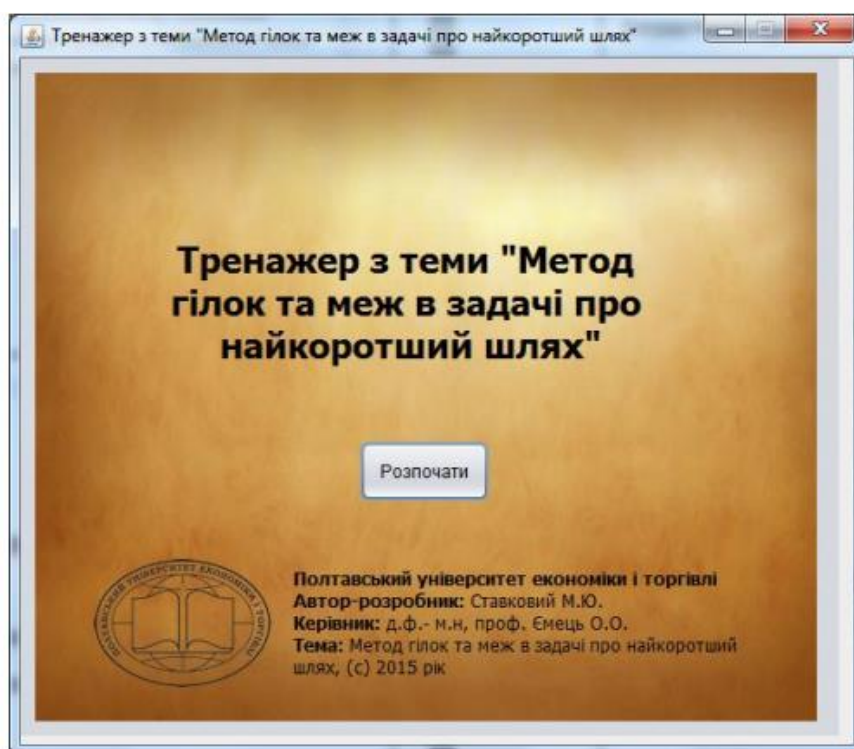


Рисунок 2.1 - Стартове вікно тренажеру

Далі користувачу необхідно здійснити вибір прикладу з кількох ймовірних (див. Рисунок 2.2), після чого буде відображено умову задачі (див. Рисунок 2.3).

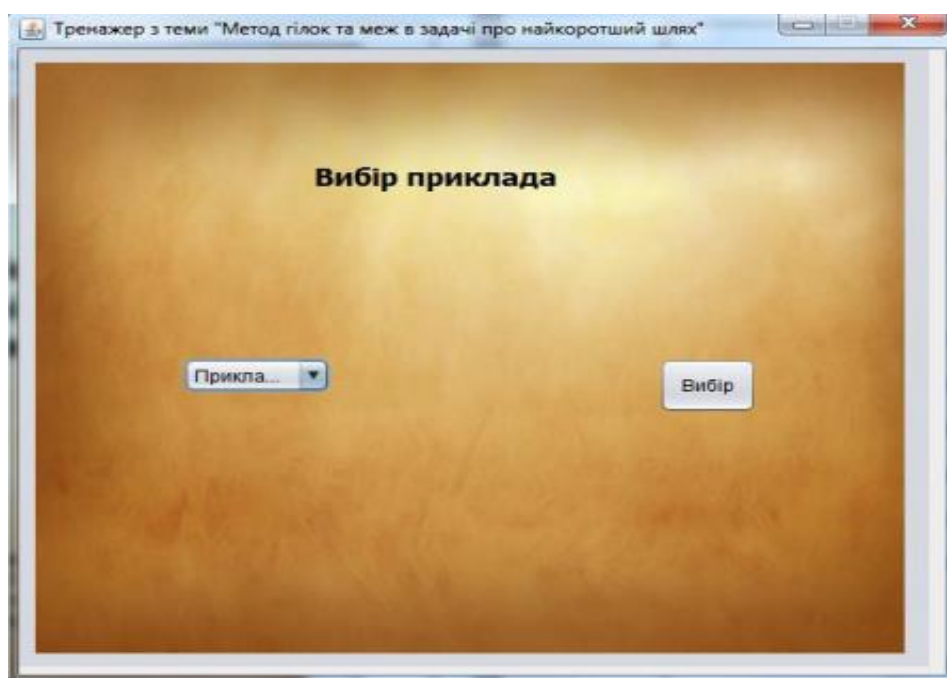


Рисунок 2.2 - Панель вибору прикладу

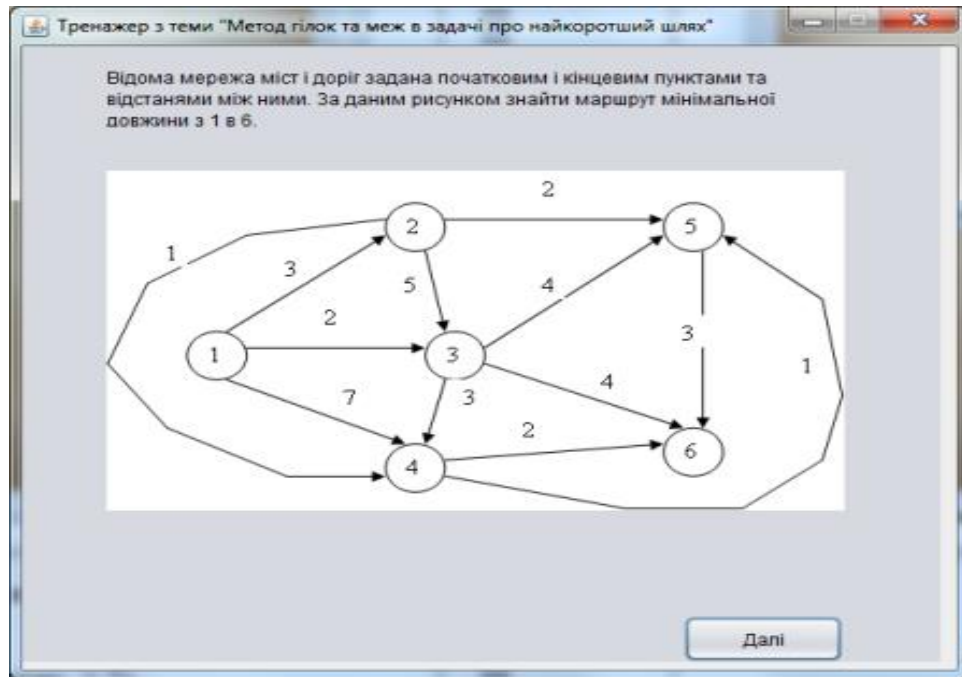


Рисунок 2.3 - Вікно з умовою задачі

При натисненні кнопки «Далі» увазі користувача надається завдання з можливими варіантами відповіді (див. Рисунок 2.4)

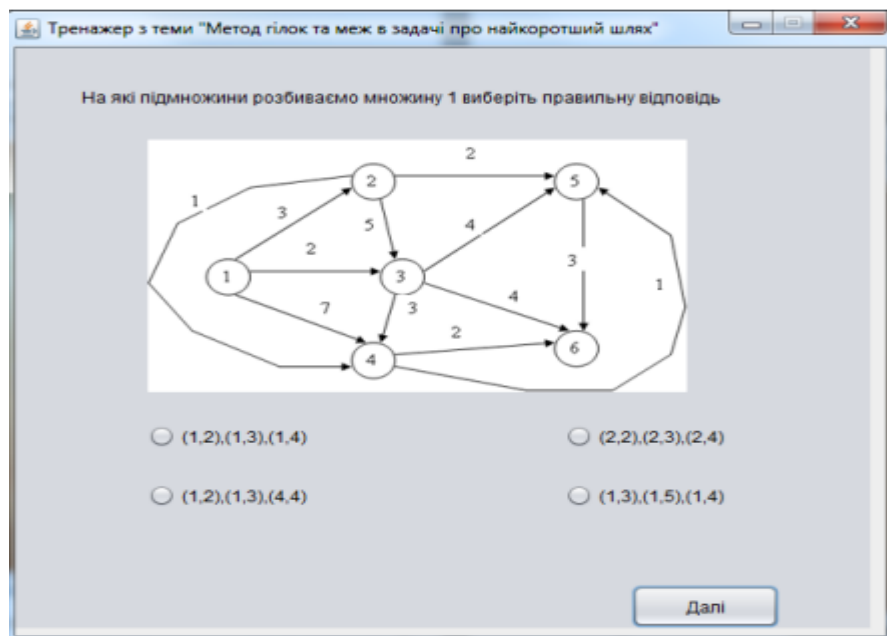


Рисунок 2.4 - Перше завдання з варіантами відповіді

При невірній обраній відповіді- відображається довідка (див. Рисунок 2.5), при вірній- перехід до наступного завдання.

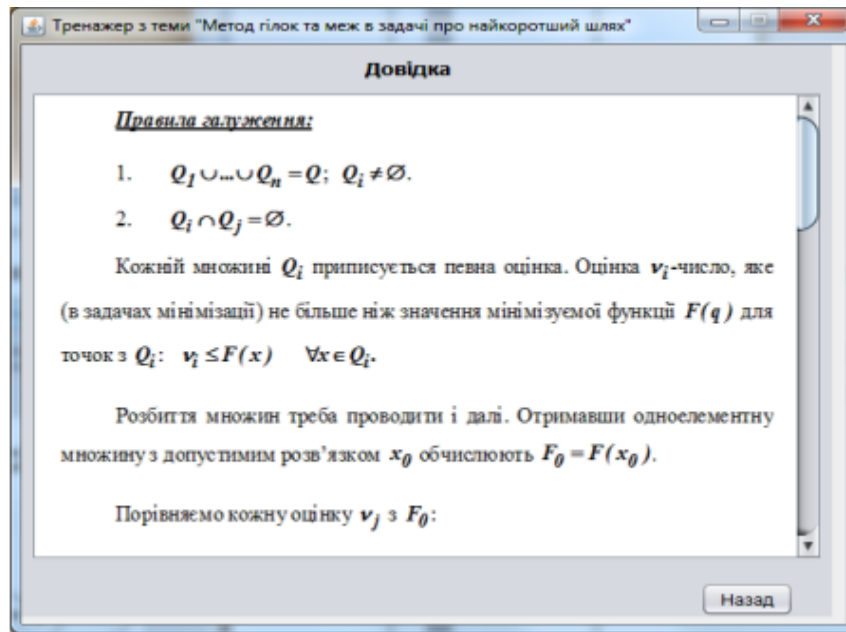


Рисунок 2.5 - Вікно з довідкою

Розв'язавши завдання, отримуємо відповідне повідомлення (див. Рисунок 2.6)

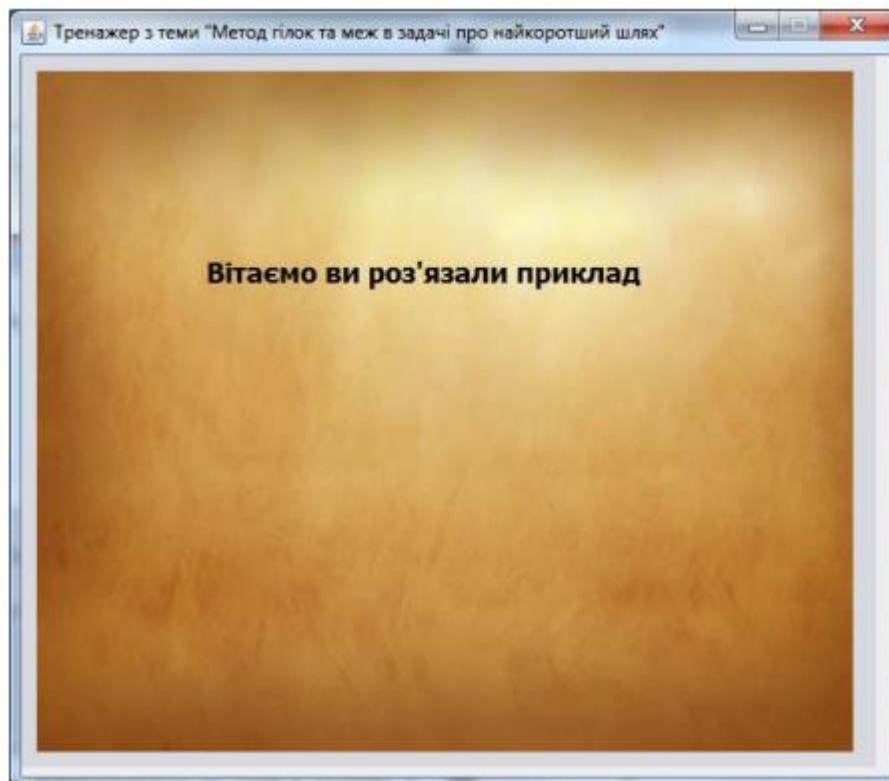


Рисунок 2.6 - Повідомлення про виконання завдання

Перевагами даного тренажеру є:

- 1.Зручний та цікавий дизайн
- 2.Завдання різних типів складності
- 3.Можливість повернутися до попереднього завдання
- 4.Забезпечення довідковою інформацією

Недоліки тренажеру Ставкового М.Ю:

1. Відсутність кнопки для повторної роботи.
2. Відсутність завдань з полями для вводу.
3. Не достатньо добре розроблена система перевірки правильності виконання завдань.
4. Відсутність повідомлення в разі помилки.

На початковій сторінці тренажеру Гусар М.В. «Знаходження вектору пріоритетів та відношення узгодженості в методі аналізу ієрархій» виводиться тема, дані про розробника та пропонується вибрати мову (українську чи англійську) (див. Рисунок 2.7; 2.8) та умову задачі (див. Рисунок 2.9; 2.10; 2.11).

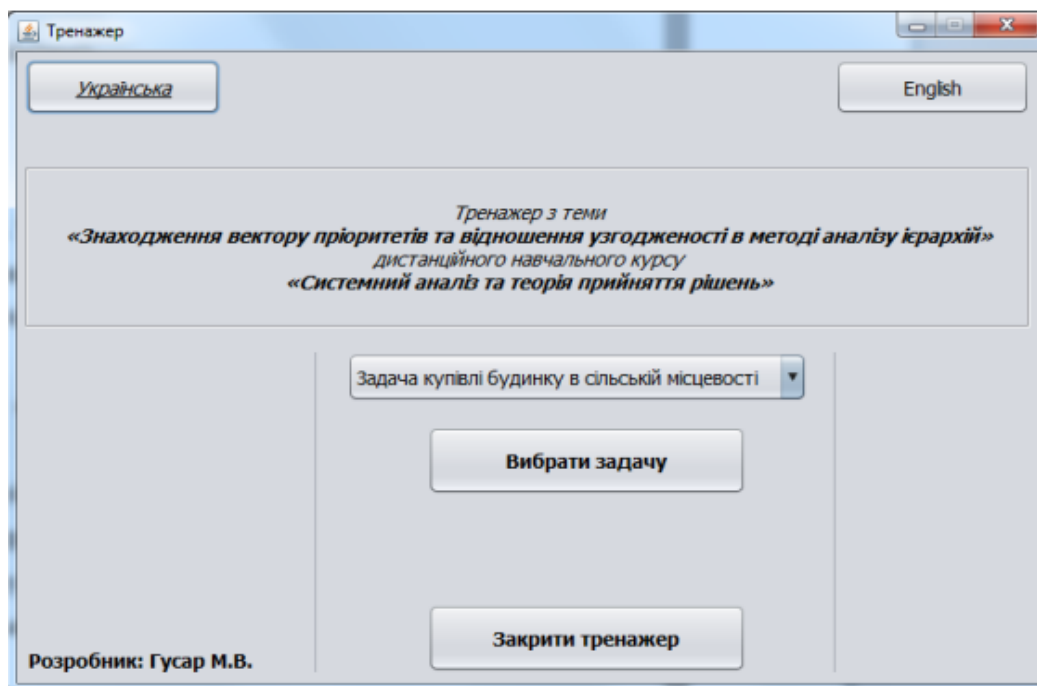


Рисунок 2.7- Стартова панель тренажеру

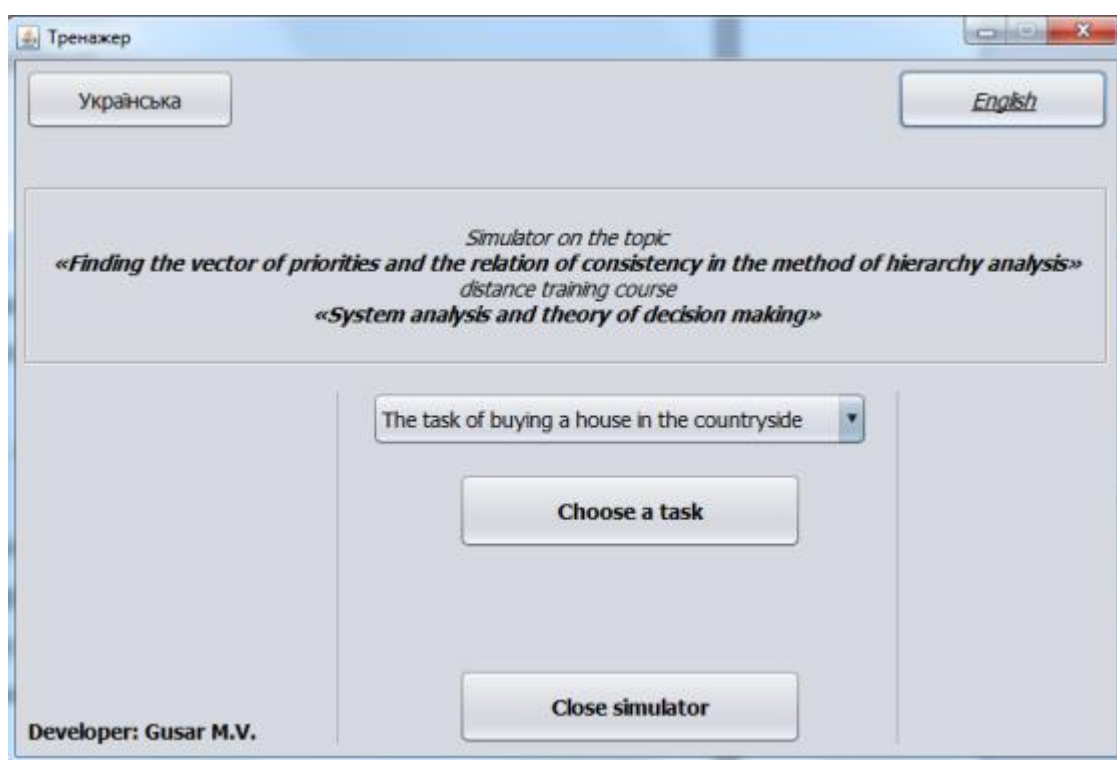


Рисунок 2.8 - Стартова панель англійською мовою

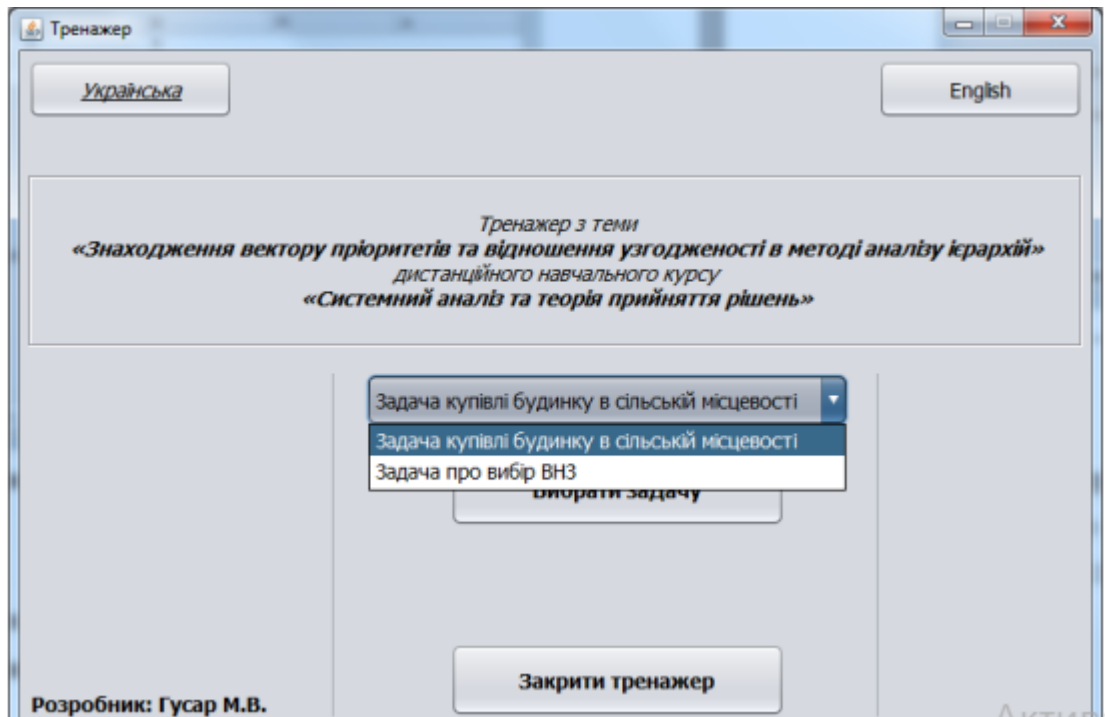


Рисунок 2.9 - Вибір задачі

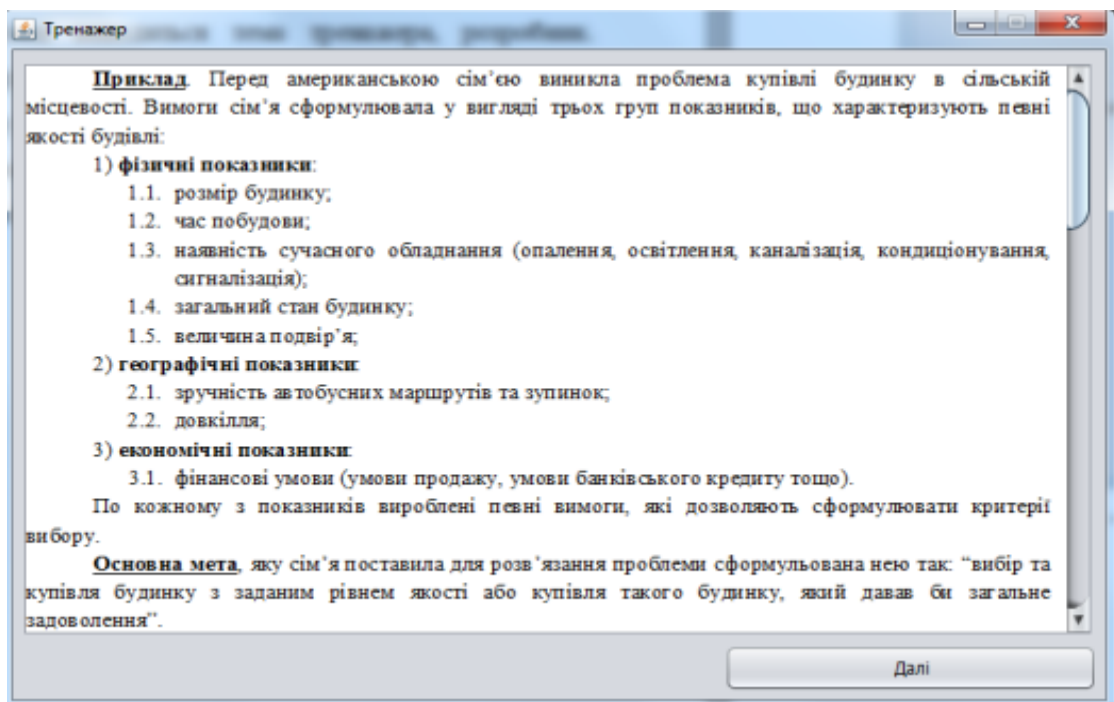


Рисунок 2.10 - Умова першої задачі

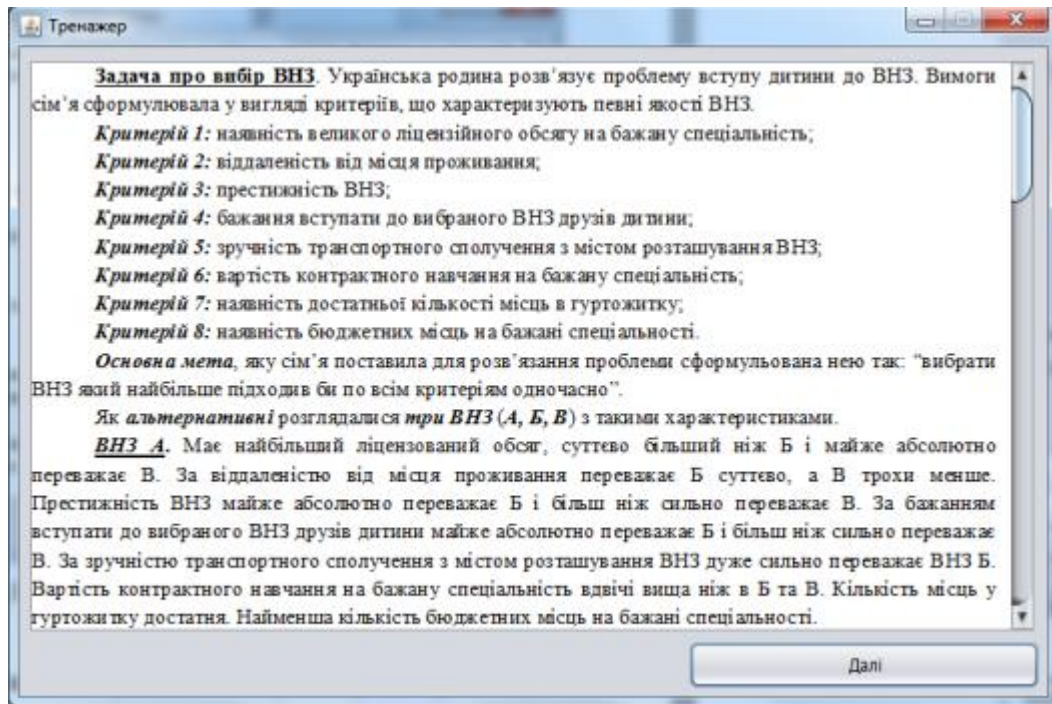


Рисунок 2.11 - Умова другої задачі

Далі студенту необхідно заповнити таблицю з вказанням вектору пріоритетів за параметрами узгодженості оцінок (див. Рисунок 2.12; 2.13)

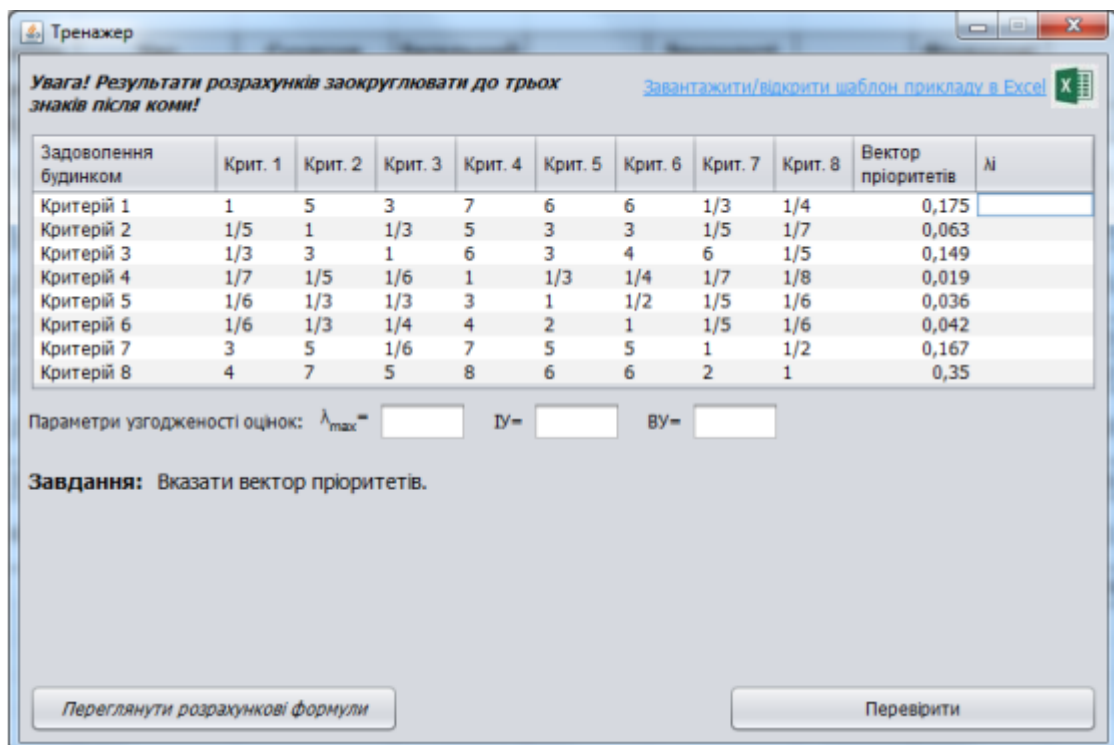


Рисунок 2.12 - Таблиця попарних порівнянь

Тренажер

Увага! Результати розрахунків заокруглювати до трьох знаків після коми! [Завантажити/відкрити шаблон прикладу в Excel](#)

Задоволення будинком	Крит. 1	Крит. 2	Крит. 3	Крит. 4	Крит. 5	Крит. 6	Крит. 7	Крит. 8	Вектор пріоритетів	λ_i
Критерій 1	1	5	3	7	6	6	1/3	1/4	0,175	9,61
Критерій 2	1/5	1	1/3	5	3	3	1/5	1/7	0,063	8,953
Критерій 3	1/3	3	1	6	3	4	6	1/5	0,149	12,501
Критерій 4	1/7	1/5	1/6	1	1/3	1/4	1/7	1/8	0,019	8,872
Критерій 5	1/6	1/3	1/3	3	1	1/2	1/5	1/6	0,036	8,603
Критерій 6	1/6	1/3	1/4	4	2	1	1/5	1/6	0,042	8,74
Критерій 7	3	5	1/6	7	5	5	1	1/2	0,167	10,352
Критерій 8	4	7	5	8	6	6	2	1	0,35	9,116

Параметри узгодженості оцінок: $\lambda_{\max} =$ $I_U =$ $I_V =$

Завдання: Вказати параметри узгодженості оцінок: визначити $\lambda_{\max}$

Рисунок 2.13 - Параметри узгодження оцінок

Після розв'язання завдання та натиснення кнопки «Далі» студенту пропонується заповнити таблицю узгодження впливу факторів третього рівня на фактори другого рівня (див. Рисунок 2.14)

Тренажер

Увага! Результати розрахунків заокруглювати до трьох знаків після коми! [Завантажити/відкрити шаблон прикладу в Excel](#)

A1	A	Б	В	PI	N
A	1	6	8		
Б	1/6	1	4		
В	1/8	1/4	1		

Параметри узгодженості оцінок: $\lambda_{\max} =$ $I_U =$ $I_V =$

Завдання: Вказати вектор пріоритетів.

Рисунок 2.14 - Таблиця впливу факторів третього на фактори другого рівня

У разі виникнення труднощів з розв'язанням, можна скористатися кнопкою «Переглянути розрахункові формули», що виконує функцію підказки на правильний хід вирішення задачі та відображає необхідні для розв'язку формули (див. Рисунок 2.14)

Тренажер

Увага! Результати розрахунків заокруглювати до трьох знаків після коми! [Завантажити/відкрити шаблон прикладу в Excel](#)

Задоволення будинком	Крит. 1	Крит. 2	Крит. 3	Крит. 4	Крит. 5	Крит. 6	Крит. 7	Крит. 8	Вектор пріоритетів	N
Критерій 1	1	5	3	7	6	6	1/3	1/4		
Критерій 2	1/5	1	1/3	5	3	3	1/5	1/7		
Критерій 3	1/3	3	1	6	3	4	6	1/5		
Критерій 4	1/7	1/5	1/6	1	1/3	1/4	1/7	1/8		
Критерій 5	1/6	1/3	1/3	3	1	1/2	1/5	1/6		
Критерій 6	1/6	1/3	1/4	4	2	1	1/5	1/6		
Критерій 7	3	5	1/6	7	5	5	1	1/2		
Критерій 8	4	7	5	8	6	6	2	1		

Параметри узгодженості оцінок: $\lambda_{\max} =$ $IV =$ $VU =$

Завдання: Вказати вектор пріоритетів.

$$V_i \approx \sqrt[n]{\prod_{j=1}^n a_{ij}}, \quad i = 1, 2, \dots, n, \quad P_i = \frac{V_i}{\sum_{i=1}^n V_i}, \quad i = 1, 2, \dots, n$$

Рисунок 2.14 - Розрахункові формули для розв'язку

Наступним кроком відобразиться умова третього завдання (див. Рисунок 2.15)

Тренажер

Розглянемо вплив факторів 3-го рівня на фактори 2-го рівня, тобто визначимо перевагу (значущість, "вагу") кожного з будинків *А, Б, В* по відношенню до кожного з факторів 2-го рівня. Це робиться на основі даних задачі побудовою 8-ми (по кількості факторів 2-го рівня) матриць парних порівнянь для трьох (будинки *А, Б, В*) факторів третього рівня. Результати – матриці та результати їх аналізу (вектори пріоритетів та міри узгодженості).

Рисунок 2.15 - Умова третього завдання

Якщо при виконанні розрахунків було допущено помилку, на екрані з'явиться відповідне вікно (див. Рисунок 2.16)

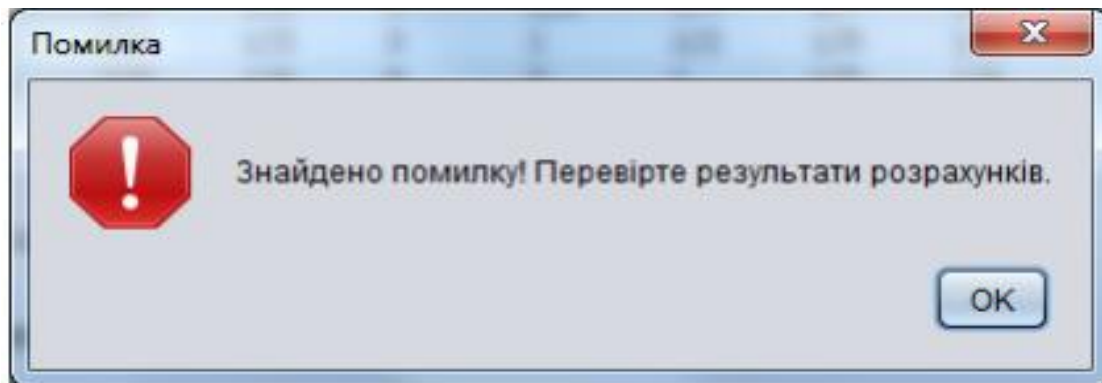


Рисунок 2.16 - Вікно з помилкою

По завершенню роботи відображається результат розв'язання задачі з можливістю переходу на стартову сторінку.

Плюси перевіреної програми-тренажера:

- наявність кількох варіантів завдань з можливістю вибору;
- забезпечення вибору мови користування;
- функціонування підказки з розрахунковими формулами;
- перевірка правильності розв'язання;

Недоліки тренажера, розробленого Гусар М.В:

- відсутнє забезпечення теоретичним матеріалом;
- однотипність завдань;
- не розроблене оцінювання тренування за балами.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Походження рекурсивних функцій.

Одним із способів уточнення інтуїтивного поняття «алгоритму» є рекурсивні функції.

З кожним алгоритмом одночасно зв'язана функція, яку цей алгоритм ставить у відповідність (встановлює взаємно однозначну відповідність) допустимим початковим даним деякий результат.

Раніше були розглянуті приклади функцій обчислювані за Тьюрінгом. Теза Тьюрінга стверджує: Функція обчислювана за допомогою довільного алгоритму тоді і тільки тоді, коли вона обчислювана за допомогою МТ.

Виникає запитання: Чи для будь-якої функції можна побудувати МТ, що буде її обчислювати?

Якщо ні, то для яких функцій можна побудувати МТ(обчислювальний алгоритм) , як описати ці алгоритмічно обчислювальні функції?

Дослідження цих питань стало основою для створення в 30-х рр ХХ ст. теорії рекурсивних функцій. Сама теорія була описана по аналогії з аксіоматичною теорією на базі заданої системи аксіом. Спочатку біли вибрані найпростіші функції, ефективна обчислюваність яких буда очевидна (так звані «аксіоми»). Далі сформульовані правила (названі операторами), на основі уже існуючих (так звані «правила виведення»). Тоді класом функцій буде сукупність всіх функцій, що отримуються з найпростіших з допомогою вибраних операторів.

Наша мета – довести, що цей клас функцій співпадає з класом функцій, обчислюваних з допомогою машин Тьюрінга.

Ідея доведення в один бік проста: спочатку довести обчислювальність по Тьюрінгу найпростіших функцій, а потім обчислювальність функцій, отриманих з обчислюваних за Тьюрінгом з допомогою вибраних операторів.

Основні поняття, необхідні для розуміння теми:

Функція – це відповідність між елементами двох множин, коли кожному елементу однієї множини ставиться у відповідність елементи іншої множини.

Рекурсія – часткове означення об'єкта через себе; спосіб зведення завдання до того ж самого але змінними початковими даними.

Рекурсивна функція – функція, означена через саму себе.

3.2. Основні поняття теорії рекурсивних функцій.

Розглянемо, функції, задані на множині натуральних чисел, і значення результату функції теж натуральне, тобто, якщо задана f , то $D(f)=E(f)=N$. Функції пропонується вибрати частково-визначені на N , тобто визначені не для всіх значень аргументів.

В якості вихідних найпростіших функцій виберемо натуральні:

$S(x)=x+1$ (функція слідування);

$O(x)=0$ (нуль – функція);

$Inc_m(x_1, x_2, \dots, x_n) = x_m$ (функції проектори, $1 \leq m \leq n$)

Ці функції є обчислювані (точніше, правильно обчислювані) за Тьюрінгом (доведення цього факту розглянемо на практичних заняттях).

В якості операторів, з допомогою яких будемо будувати нові функції, виберемо наступні три оператора:

- 1) оператор суперпозиції;
- 2) оператор примітивної рекурсії;
- 3) оператор мінімізації.

Озн. 1. n – місна функція f отримана з m – місної функції f і n – містних функцій $g_1, g_2, g_3, \dots, g_m$ з допомогою оператора суперпозиції, якщо для всіх $x_1, \dots, x_n$ справедлива рівність:

$$f(x_1, x_2, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)).$$

Означення використовує поняття суперпозиції функції і складної функції.

Суперпозиція – підстановка функції у функцію.

Складна функція – функція від функції.

Теорема. Якщо функції $f(x_1, \dots, x_m)$, $g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)$ правильно обчислювані за Тьюрінгом, то правильно обчислювана і складна функція (суперпозиція функцій)

$$\varphi(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n))$$

Озн. 2. $(n+1)$ – місна функція φ отримана з n – місної функції f і $(n+2)$ – місної функції g з допомогою оператора примітивної рекурсії, якщо для довільних $x_1, \dots, x_n, y$ справедливі рівності :

$$\varphi(x_1, \dots, x_n, 0) = f(x_1, \dots, x_n) \quad (1)$$

$$\varphi(x_1, \dots, x_n, y+1) = g(x_1, \dots, x_n, y, \varphi(x_1, \dots, x_n, y)). \quad (2)$$

Рівності (1) – (2) називаються схемою примітивної рекурсії.

Зауваження 1. Рекурсія виконується тільки по одній змінній y ; інші n змінних $x_1, x_2, \dots, x_n$ на момент використання схеми примітивної рекурсії зафіксовані і виконують роль параметрів.

Зауваження 2. Схема примітивної рекурсії виражає кожне значення функції φ не тільки через функції f і g , але й через попереднє значення функції φ . Тобто, щоб отримати значення $\varphi(x_1, x_2, \dots, x_n, k)$ необхідно виконати $k+1$ обчислення по схемі – для $y=0, 1, \dots, k$.

Озн. 3. Функція називається примітивно-рекурсивною, якщо вона може бути отримана з найпростіших функцій O, S, I_n, m з використанням скінченного числа операторів суперпозиції і примітивної рекурсії.

Озн. 4. n – місна функція φ виводиться з $(n+1)$ – місної функції f_1 і f_2 з допомогою оператора мінімізації, або оператора найменшого числа, якщо для довільних $x_1, x_2, \dots, x_n, y$ рівність

$$\varphi(x_1, \dots, x_n) = y$$

виконується тоді і тільки тоді, коли значення $f_1(x_1, x_2, \dots, x_n, 0)$ та

$f_i(x_1, x_2, \dots, x_n, y-1)$, де $i = 1, 2$, визначені і попарно нерівні, тобто

$f_1(x_1, \dots, x_n, 0) \neq f_2(x_1, \dots, x_n, 0), \dots,$

$f_1(x_1, \dots, x_n, y-1) \neq f_2(x_1, \dots, x_n, y-1),$

але

$f_1(x_1, \dots, x_n, y) = f_2(x_1, \dots, x_n, y)$

Іншими словами: величина $\phi(x_1, \dots, x_n)$ дорівнює найменшому значенню аргумента y , для якого виконується остання нерівність.

Для оператора мінімізації використовують позначення:

$\phi(x_1, \dots, x_n) = \mu\gamma[f_1(x_1, \dots, x_n, y) = f_2(x_1, \dots, x_n, y)].$

Якщо $f_2=0$, то

$\phi(x_1, \dots, x_n) = \mu\gamma[f(x_1, \dots, x_n, y)=0]$

Оператор мінімізації часто називають μ -оператором.

Оператор мінімізації використовується для породження важливого класу рекурсивних функцій.

3.3. Алгоритм роботи тренажера

Крок 1. Запуск застосунку.

Крок 2. Видача початкового меню з темою та виконавцем роботи.

Крок 3. Перехід до навчальних матеріалів шляхом натиснення на кнопку «Старт».

Крок 4. Видача практичного завдання у вигляді тесту:

1. Функція називається частково рекурсивною, якщо

Виберіть одну відповідь:

А. її можна отримати з найпростіших функцій O , S , In_m за допомогою кінцевої кількості операторів суперпозиції, примітивної рекурсії та мінімізації

В. її можна отримати з найпростіших функцій O , S , In_m за допомогою кінцевої кількості операторів суперпозиції та примітивної рекурсії

С. її можна отримати з найпростіших функцій O , S , In_m за допомогою кінцевого числа операторів суперпозиції

Правильна відповідь – А

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 5. Видача практичного завдання у вигляді тесту:

2. Вам буде запропоновано вибрати функції

Виберіть одну відповідь:

А. всюди визначені на Z

В. частково визначені на Z

С. скрізь визначені на N

Правильна відповідь - В

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 6. Видача практичного завдання у вигляді тесту:

3. Функція називається частково рекурсивною якщо

Виберіть одну відповідь:

А. її можна отримати з найпростіших функцій O , S , Inm за допомогою кінцевої кількості операторів суперпозиції, примітивної рекурсії та мінімізації

В. її можна отримати з найпростіших функцій O , S , Inm за допомогою кінцевої кількості операторів суперпозиції та примітивної рекурсії

С. її можна отримати з найпростіших функцій O , S , Inm за допомогою кінцевого числа операторів суперпозиції

Правильна відповідь - А

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 7. Видача практичного завдання у вигляді тесту:

4. Рекурсія це

Виберіть одну відповідь:

А. відповідність між елементами двох множин, коли одному елементу однієї множини відповідає один елемент іншої множини

Б. відповідність між елементами двох множин, коли всі елементи однієї множини відповідають елементам іншої множини

С. часткове визначення предмета через нього самого; спосіб зведення завдання до такого ж, але зі змінними вихідними даними

Правильна відповідь - С

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 8. Видача практичного завдання у вигляді тесту:

5. Суперпозиція функцій – це

Виберіть одну або декілька відповідей:

А. підстановка функції у функцію

Б. найкраща позиція аргументу у функції

С. комплексна функція

Правильна відповідь - А

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 9. Видача практичного завдання у вигляді тесту:

6. Наступна функція має вигляд

Введіть відповідь у полі для вводу нижче:

Правильна відповідь - $S(x)=x+1$

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 10. Видача практичного завдання у вигляді тесту:

7. Продовжте теорему. Якщо функції $f(x_1, \dots, x_m)$, $g_1(x_1, \dots, x_n)$, $g_2(x_1, \dots, x_n)$, $g_3(x_1, \dots, x_n)$, $g_4(x_1, \dots, x_n)$, $g_5(x_1, \dots, x_n)$ правильно обчислені Тьюрингом, тоді

Виберіть одну відповідь:

А. правильно обчислюється і сума цих функцій

В. правильно розрахована і різниця між цими функціями

С. правильно розрахована складна функція $\varphi(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n))$

Правильна відповідь - С

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 10. Видача практичного завдання у вигляді тесту:

8. Рекурсія у формулах

$$\varphi(x_1, \dots, x_n, 0) = f(x_1, \dots, x_n) \quad \varphi(x_1, \dots, x_n, y+1) = f(x_1, \dots, x_n, \varphi(x_1, \dots, x_n, y))$$

(1)

$$\varphi(x_1, \dots, x_n, y+1) = g(x_1, \dots, x_n, y, \varphi(x_1, \dots, x_n, y)) \quad \varphi(x_1, \dots, x_n, y+1) = g(x_1, \dots, x_n, y, \varphi(x_1, \dots, x_n, y))$$

(2) виконується

Виберіть одну відповідь:

А. n змінних $x_1, x_2, \dots, x_n$ і змінна y

В. лише одна змінна y ; інші n змінних $x_1, x_2, \dots, x_n$ на момент використання схеми примітивної рекурсії є фіксованими і виступають параметрами

С. для змінних $x_1, x_2, \dots, x_n$

Правильна відповідь - В

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 11. Видача практичного завдання у вигляді тесту:

9. Функція називається примітивно-рекурсивною if

Виберіть одну відповідь:

А. можна отримати з найпростіших функцій $O, S, I_m O, S, I_m n$ з кінцевим числом операторів суперпозиції, примітивної рекурсії та мінімізації.

В. її можна отримати з найпростіших функцій $O, S, I_m O, S, I_m n$ за допомогою кінцевої кількості операторів суперпозиції

С. її можна отримати з найпростіших функцій $O, S, I_m O, S, I_m n$ за допомогою кінцевого числа операторів суперпозиції та примітивної рекурсії

Правильна відповідь - С

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 12. Видача практичного завдання у вигляді тесту:

10. Функція це

Виберіть одну відповідь:

А. часткове визначення предмета через нього самого; спосіб зведення завдання до такого ж, але зі змінними вихідними даними

Б. відповідність між елементами двох множин, коли всі елементи однієї множини відповідають елементам іншої множини

С. відповідність між елементами двох множин, коли одному елементу однієї множини відповідає один елемент іншої множини

Правильна відповідь – С

Після вибору правильного варіанту відповіді відбувається перехід до наступного практичного завдання.

Крок 13. Перехід до фінального вікна тренажеру з повідомлення про завершення роботи.

Крок 14. Повтор роботи тренажеру через відповідну кнопку чи завершення роботи.

3.4. Блок-схема програми-тренажера

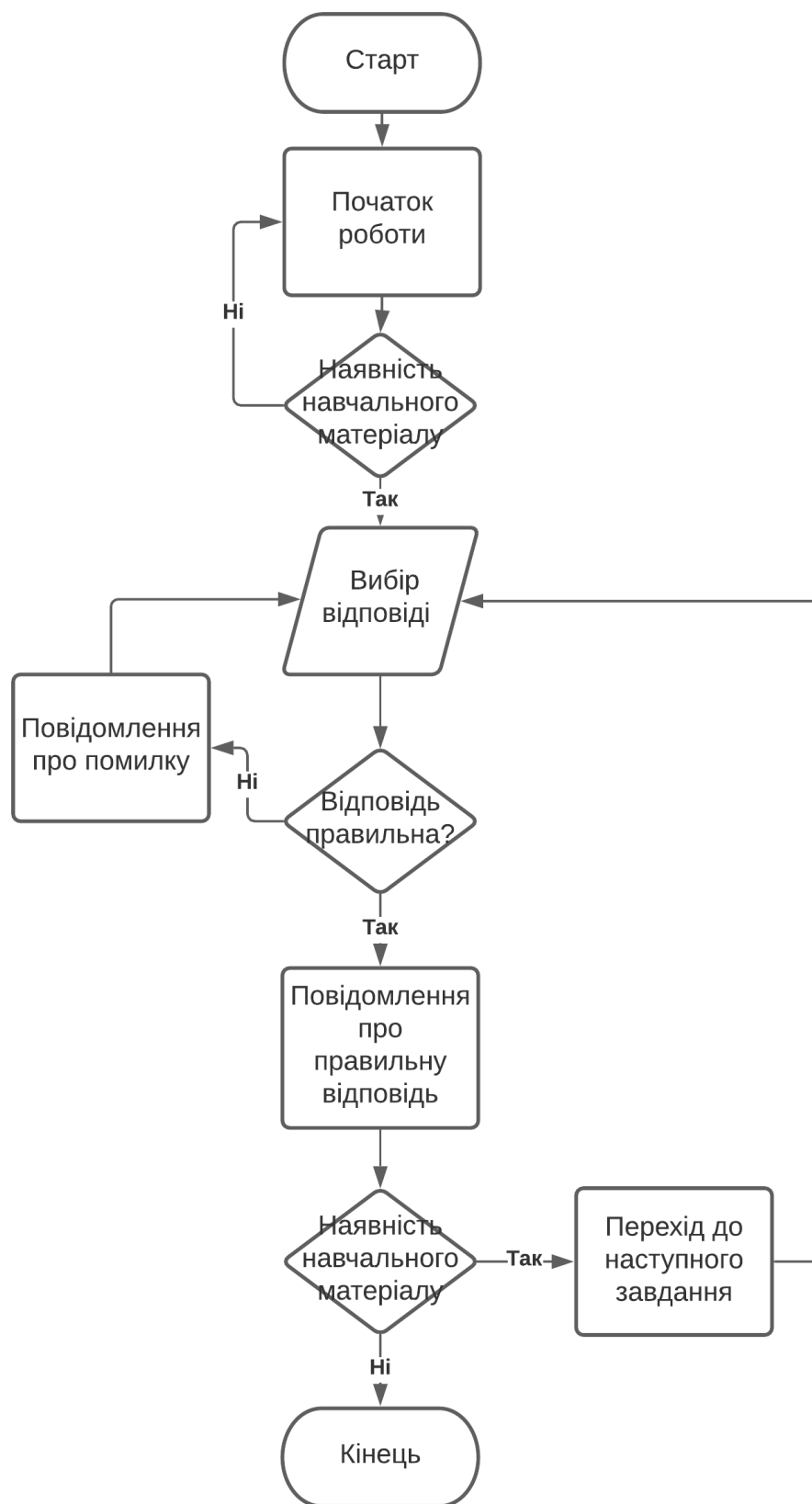


Рисунок 3.1 – Блок-схема алгоритму роботи тренажера

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис розробки програмного забезпечення тренажеру

Після запуску тренажеру користувач повинен перейти до початкового меню тренажеру в якому може бачити тему та виконавця роботи.

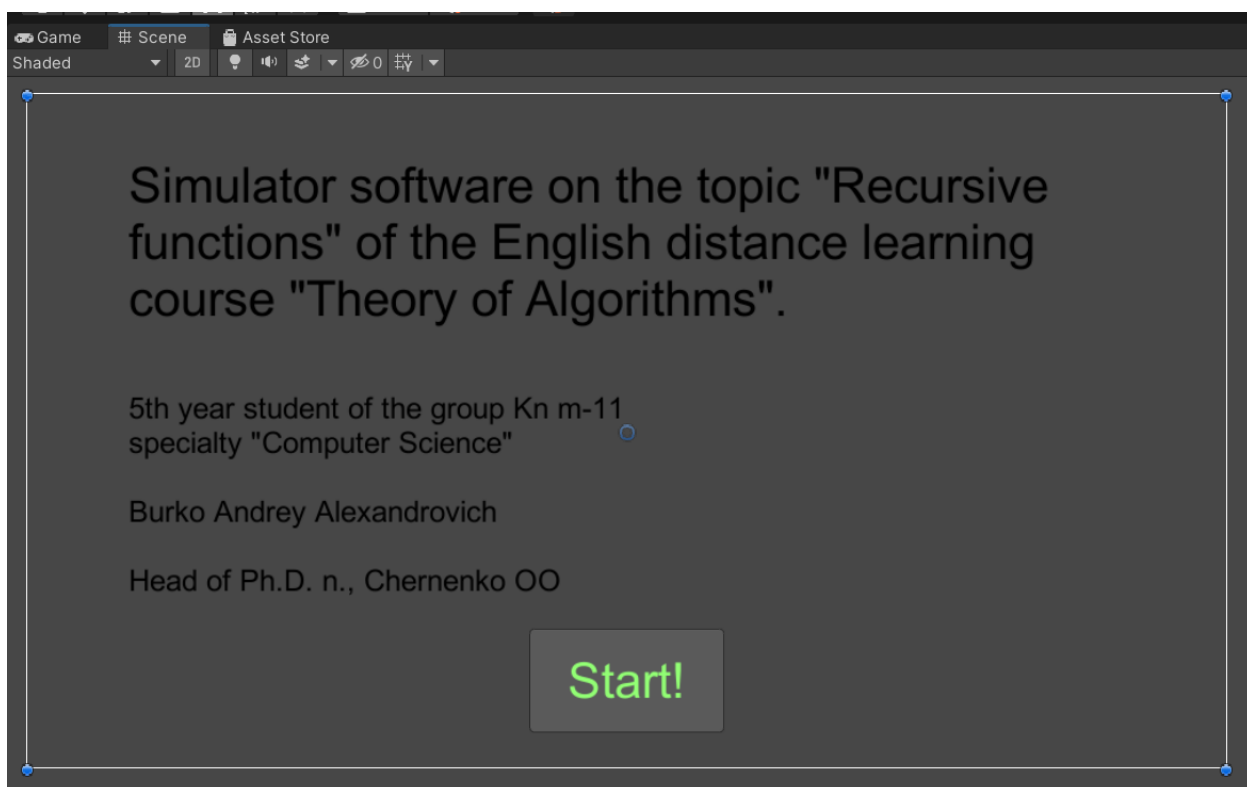


Рисунок 4.1 – Початкове меню тренажеру

Кнопка «Start!» перенаправляє користувача до першого слайду з практичними завданнями. Кнопки в тренажері працюють за допомогою скриптів логічних функцій, що вмикають один елемент та вимикають інший.

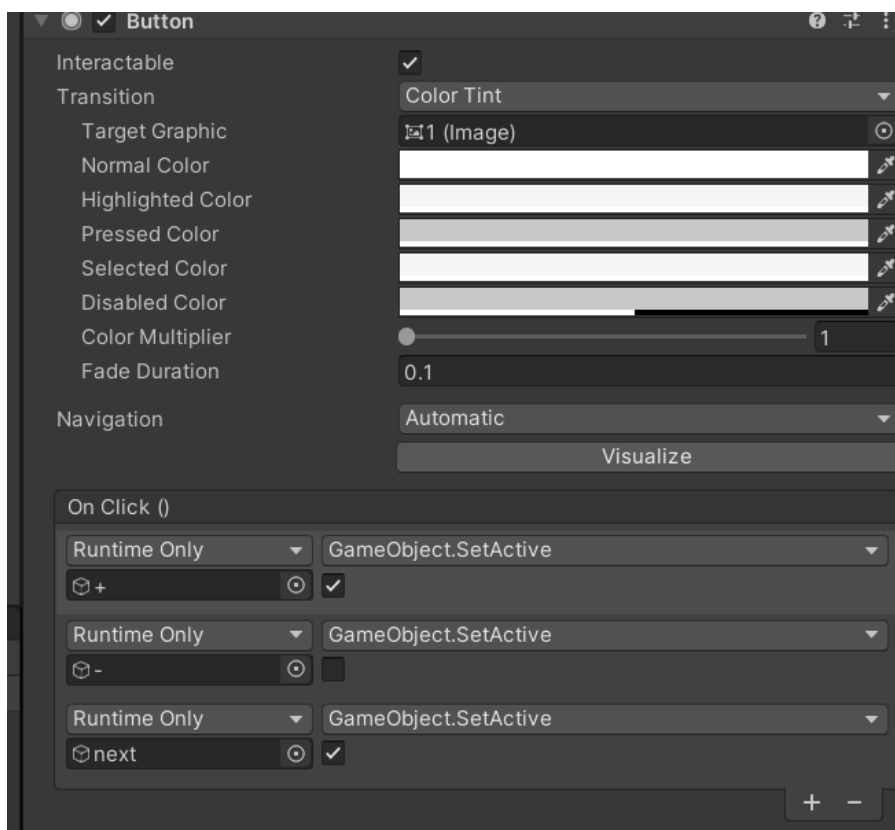


Рисунок 4.2 – Скрипти для роботи з кнопками

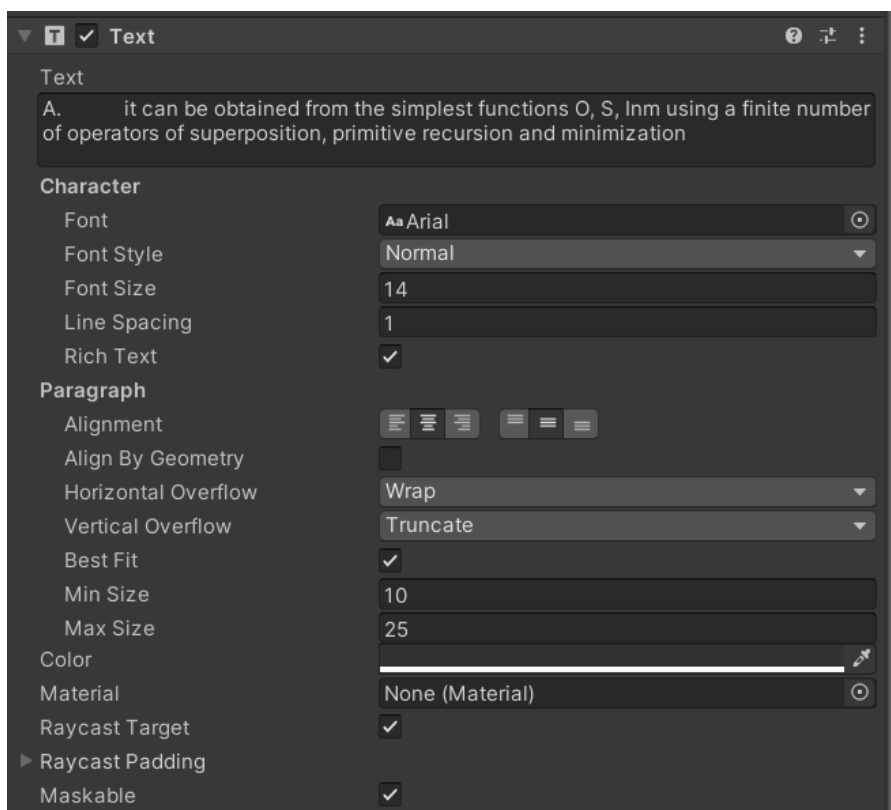


Рисунок 4.3 – Скрипти для роботи з текстом

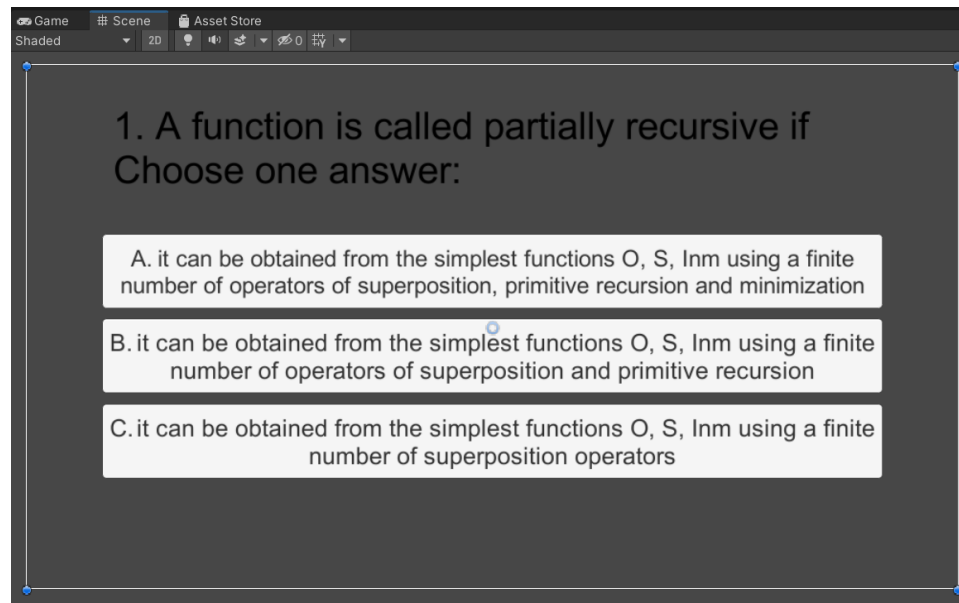


Рисунок 4.4 – Зовнішній вигляд практичних завдань

Кожен варіант відповіді реалізовано у вигляді кнопки, що у разі вибору правильної відповіді вмикає текстовий елемент з повідомленням про вибір правильної відповіді та вмикає кнопку «Continue», що дозволяє продовжити роботу з тренажером.

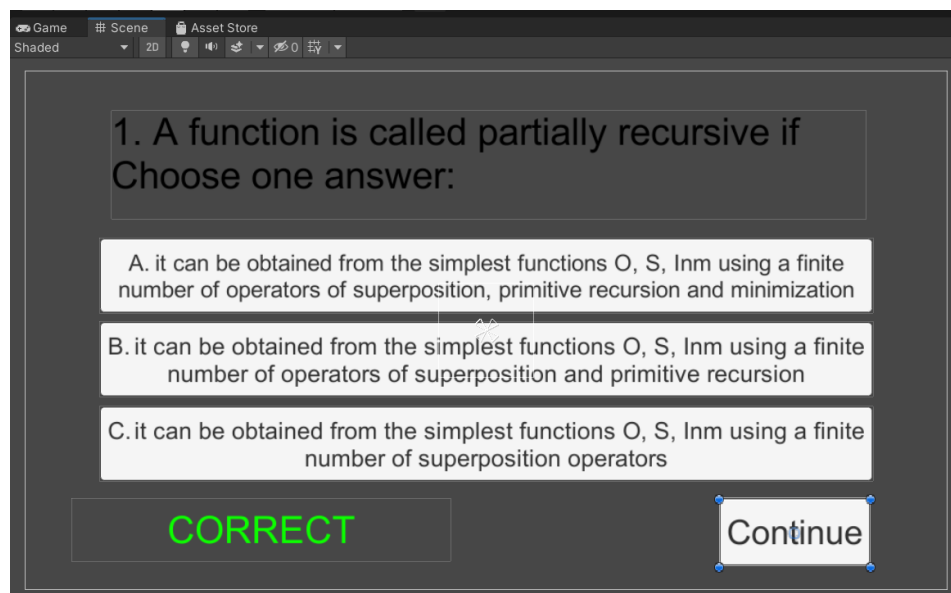


Рисунок 4.5 – Вибір правильного варіанту відповіді

У разі вибору неправильної відповіді вмикається текстовий елемент з повідомленням про помилку та вмикається кнопка «Continue» (вимкнена за замовчуванням).

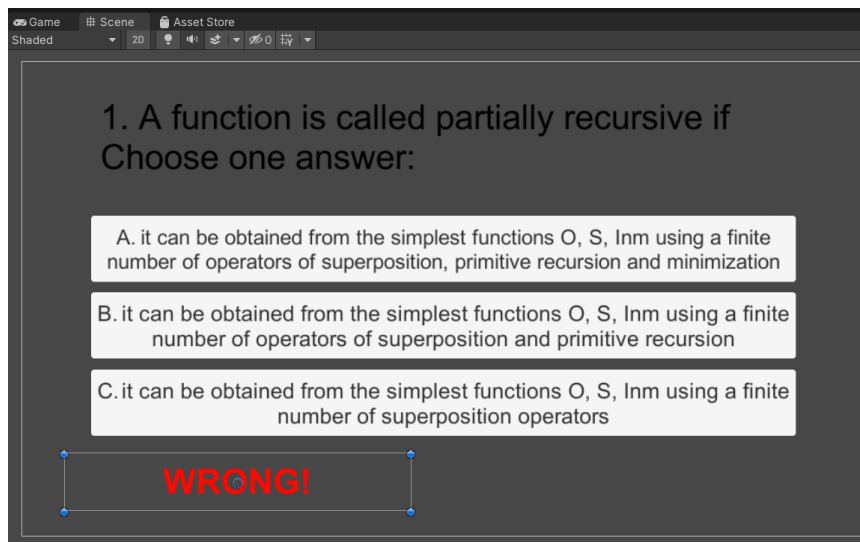


Рисунок 4.6 – Вибір неправильного варіанту відповіді

Після завершення роботи з завданнями в елементах тренажера користувач переходить до фінального вікна у якому бачить повідомлення про завершення роботи та кнопку «Retry», що перенаправить його до початкового меню тренажера.

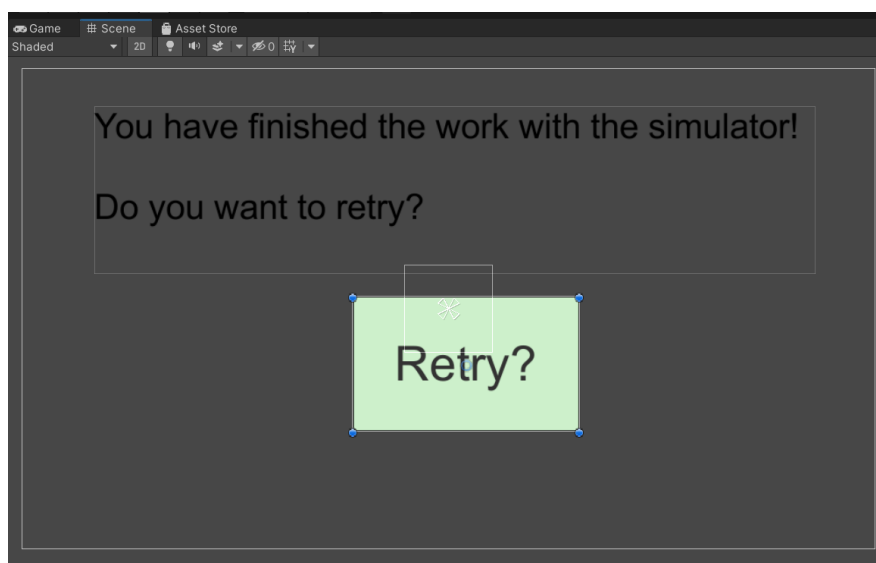


Рисунок 4.7 – Фінальне вікно тренажера

4.2. Інструкція для роботи з тренажером

Після запуску програмного застосунку користувач переходить до меню в якому може вибрати мову для проходження тренажеру. Після вибору відбувається перехід до практичних завдань.

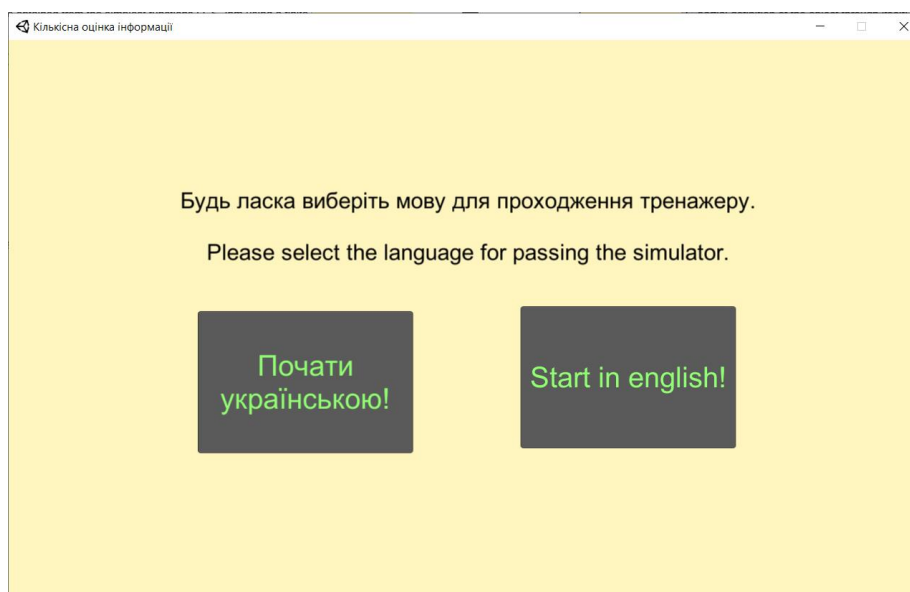


Рисунок 4.8 – Меню вибору мови

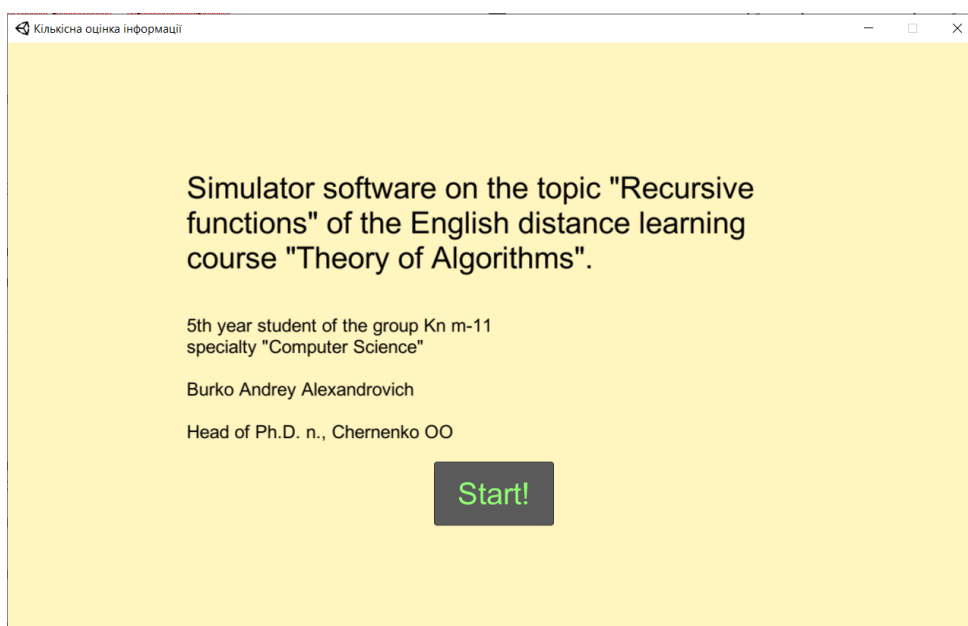


Рисунок 4.9 – Початкове меню тренажеру

Після натиснення на кнопку «Start!» на початковому меню тренажеру користувач переходить до практичних завдань.

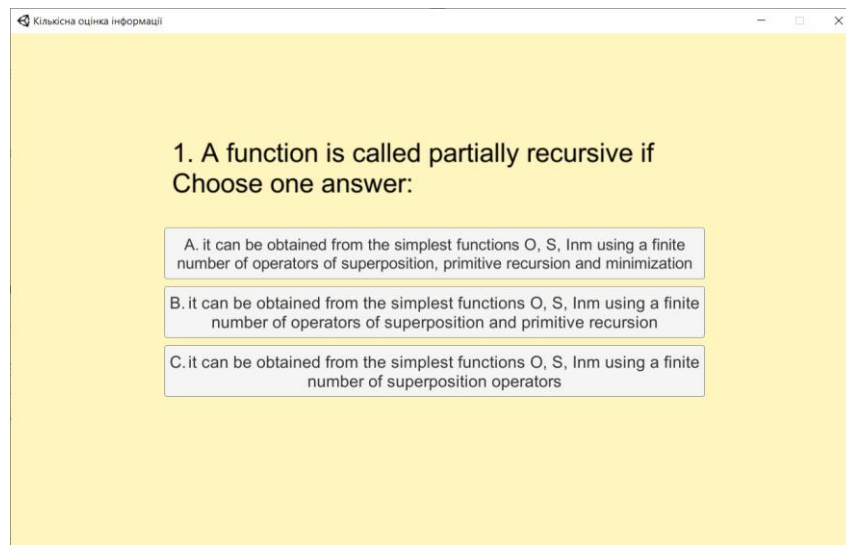


Рисунок 4.10 – Практичне завдання в тренажері

Після вибору неправильного варіанту відповіді користувач отримує повідомлення про помилку.

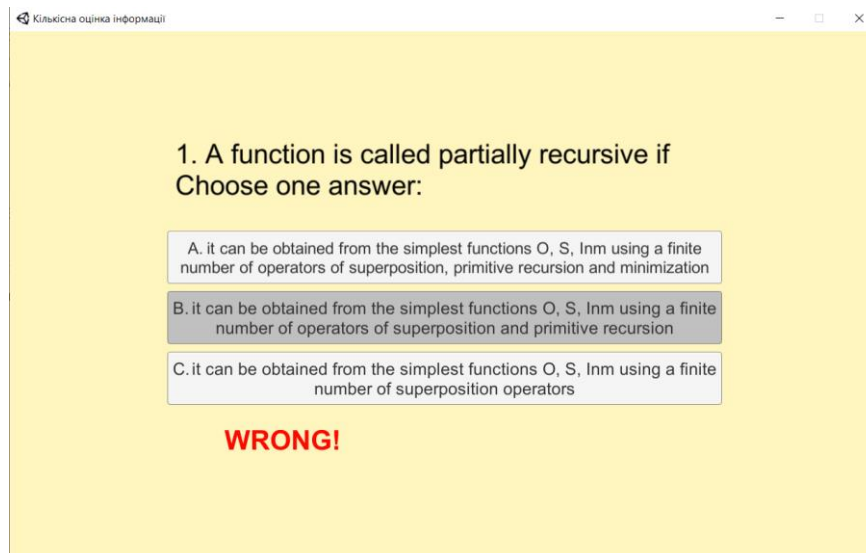


Рисунок 4.11 – Вибір неправильної відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

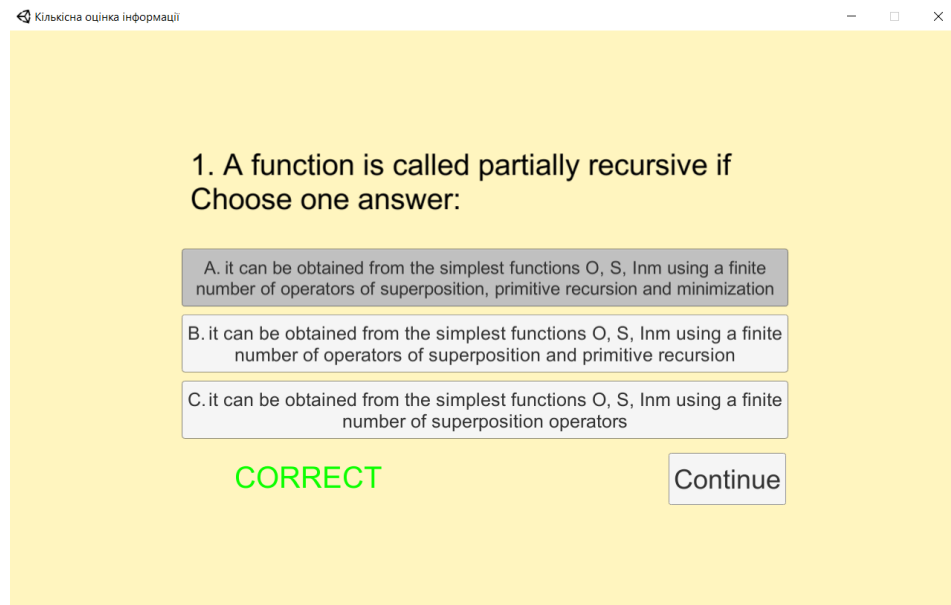


Рисунок 4.12 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

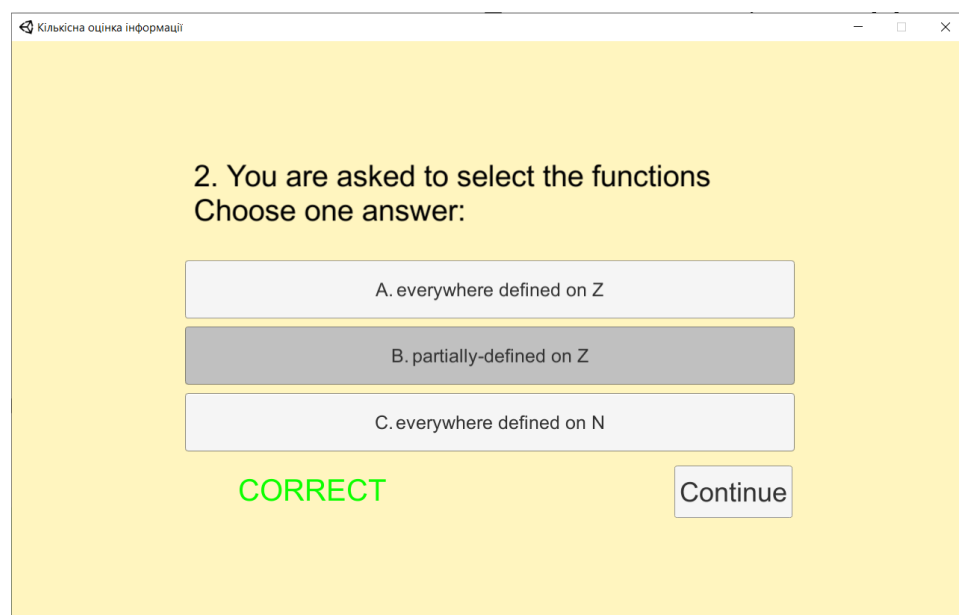


Рисунок 4.13 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

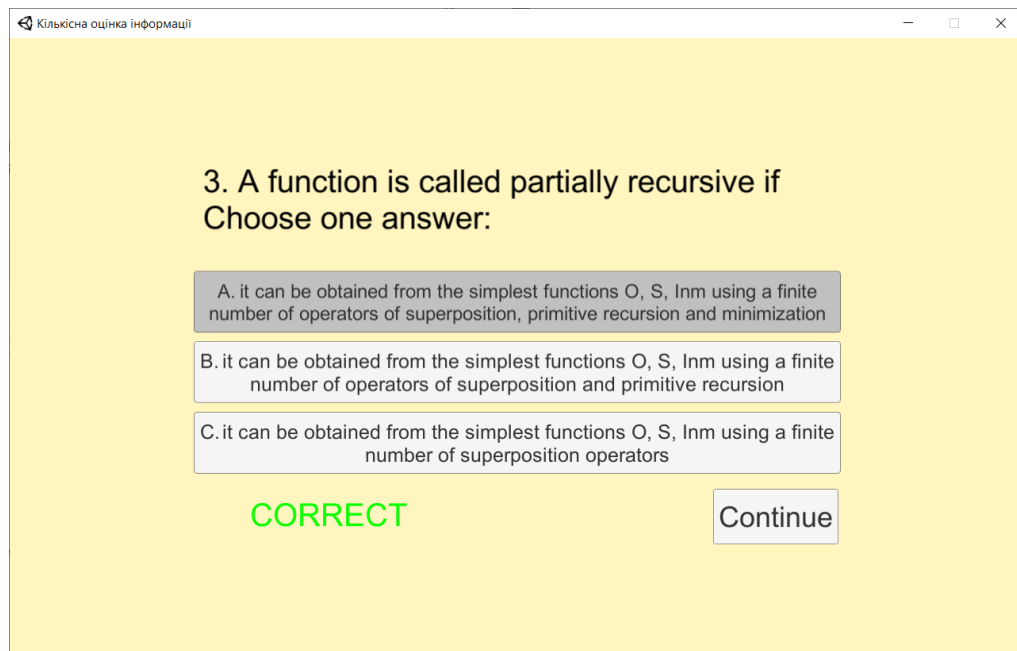


Рисунок 4.14 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

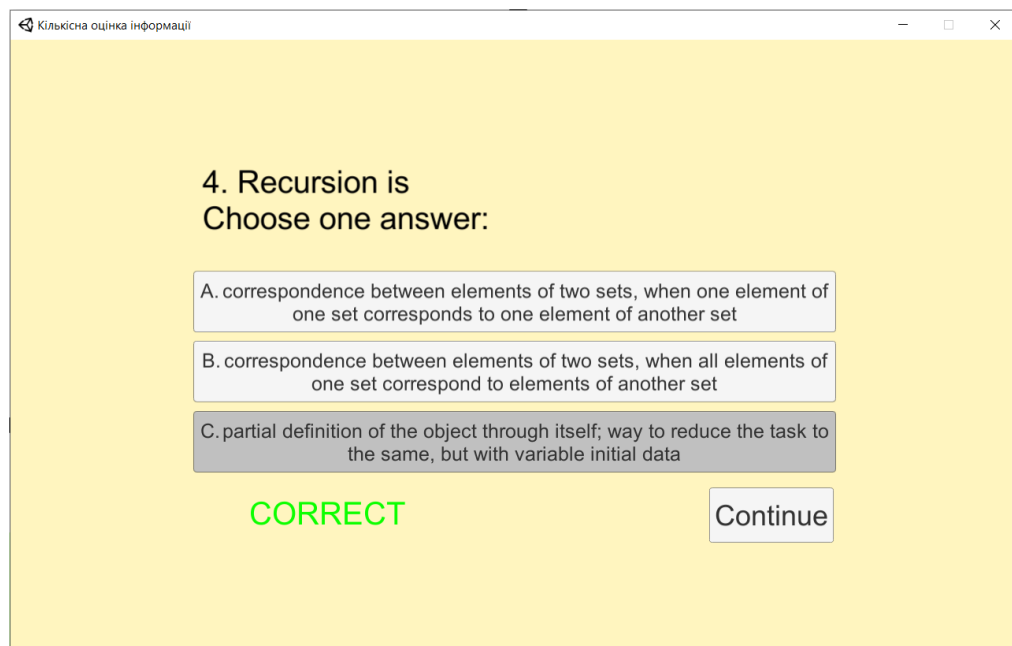


Рисунок 4.15 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue». У даному завданні присутня можливість обрати оду або кілька варіантів відповіді.

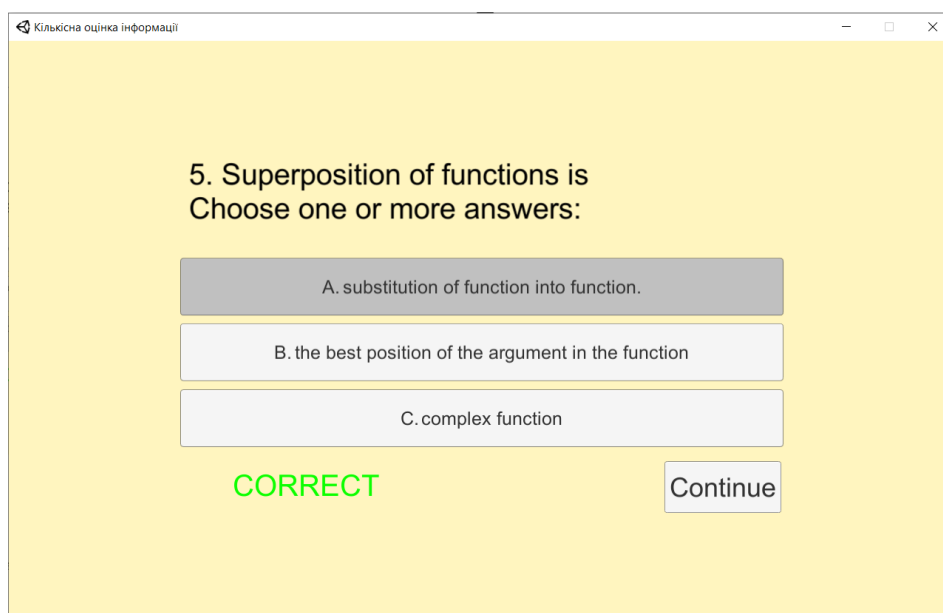
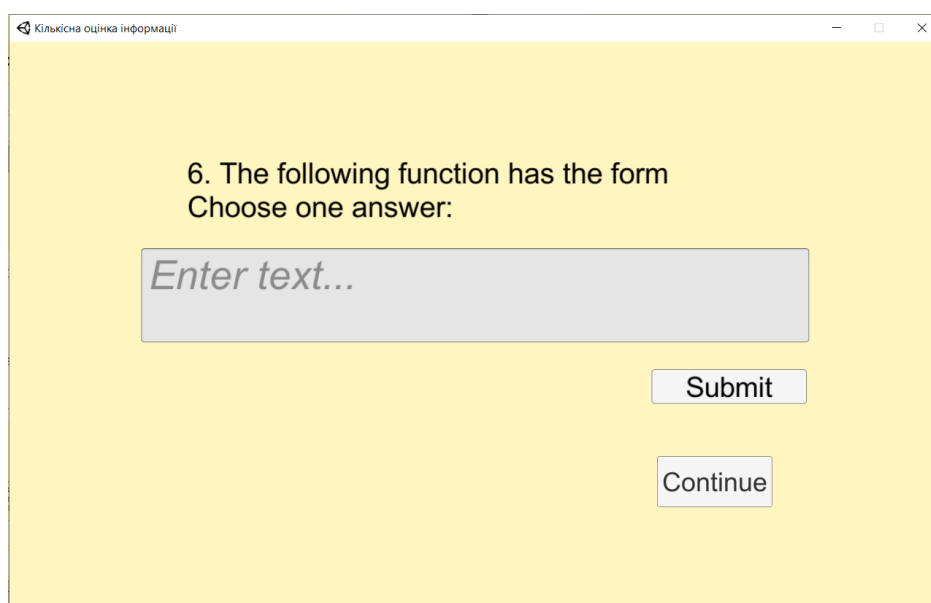


Рисунок 4.16 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

Наступне завдання оформлене як завдання з полем для вводу де користувач може ввести свою відповідь.



Кількісна оцінка інформації

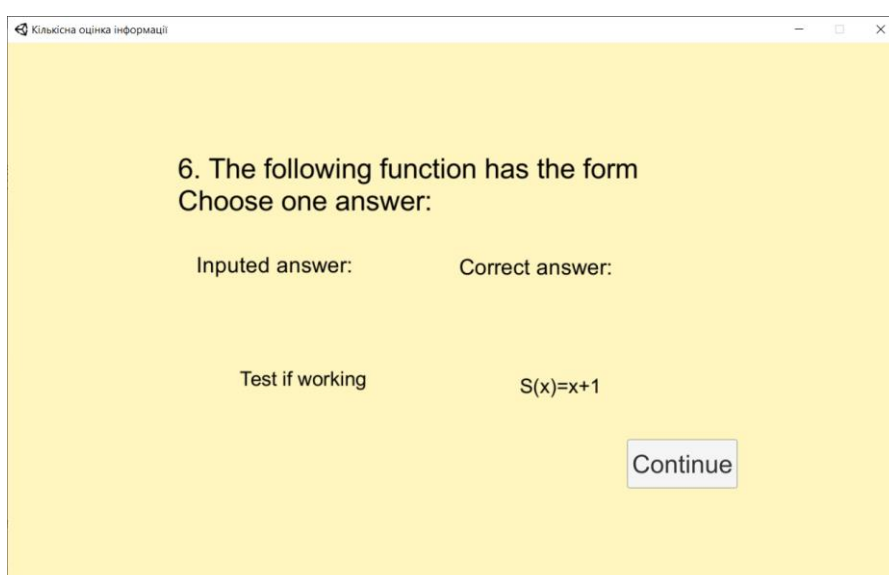
6. The following function has the form
Choose one answer:

Enter text...

Submit

Continue

Рисунок 4.17 – Завдання перед вводом інформації



Кількісна оцінка інформації

6. The following function has the form
Choose one answer:

Inputed answer:	Correct answer:
Test if working	$S(x)=x+1$

Continue

Рисунок 4.18 – Завдання після вводу інформації

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

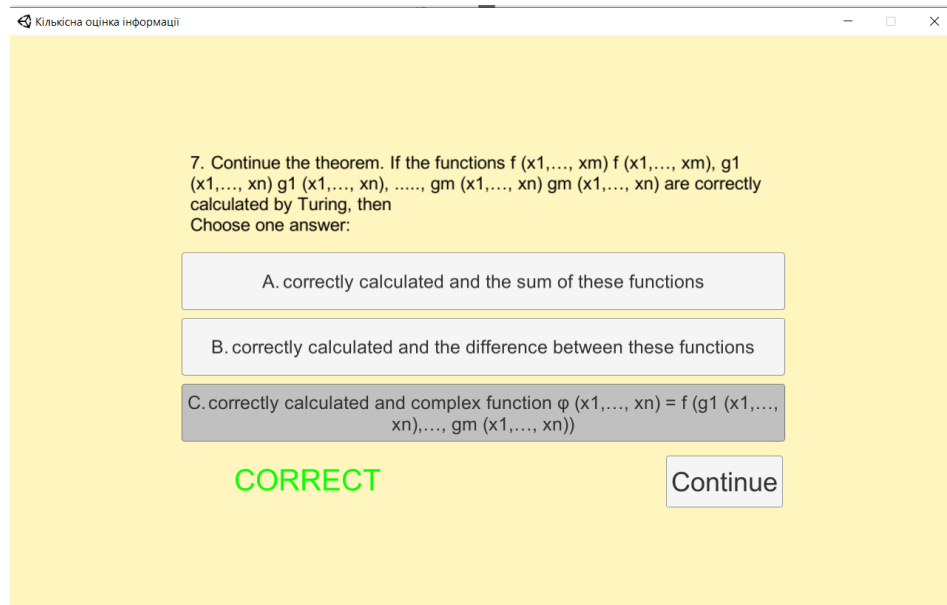


Рисунок 4.19 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

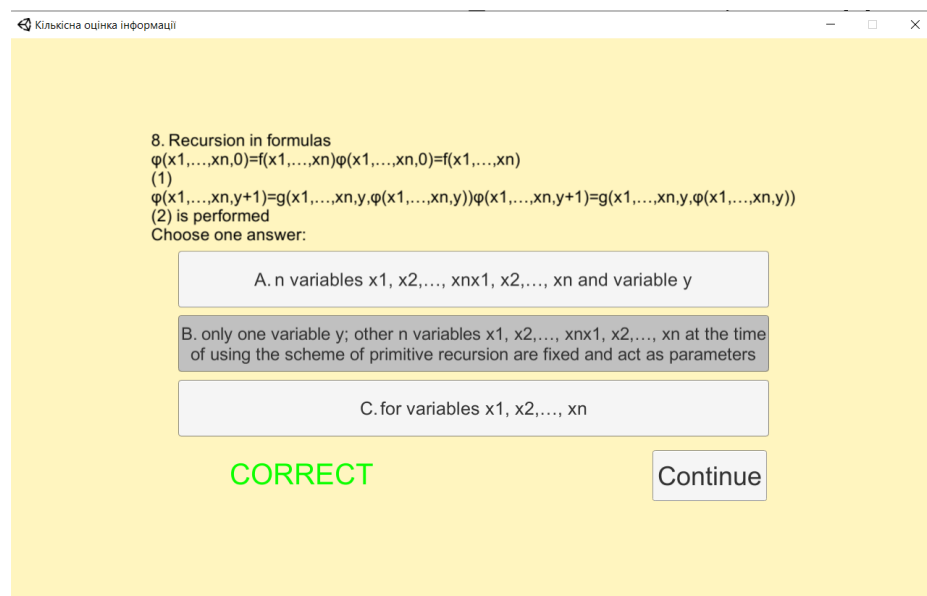


Рисунок 4.20 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

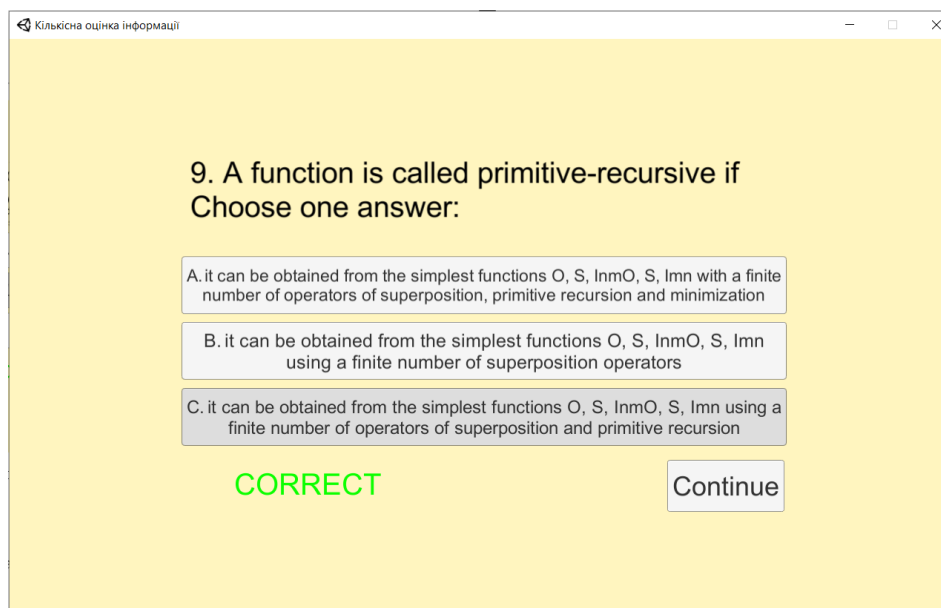


Рисунок 4.21 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue».

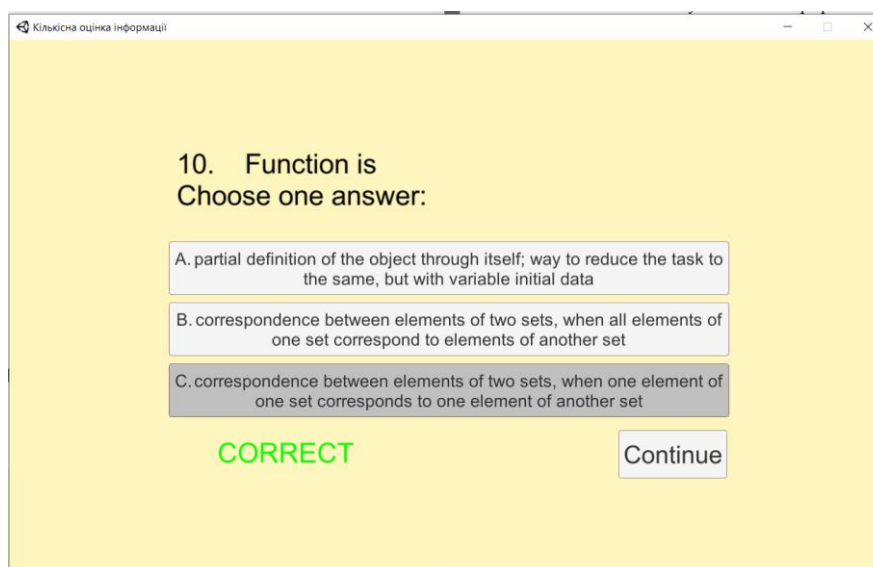


Рисунок 4.22 – Вибір правильного варіанту відповіді

Після вибору правильного варіанту відповіді користувач отримує повідомлення про правильну відповідь та доступ до кнопки «Continue», що перенаправить його до фінального вікна тренажеру. На фінальному вікні

тренажеру користувач отримує повідомлення про успішне завершення тренажеру та доступ до кнопки «Retry».

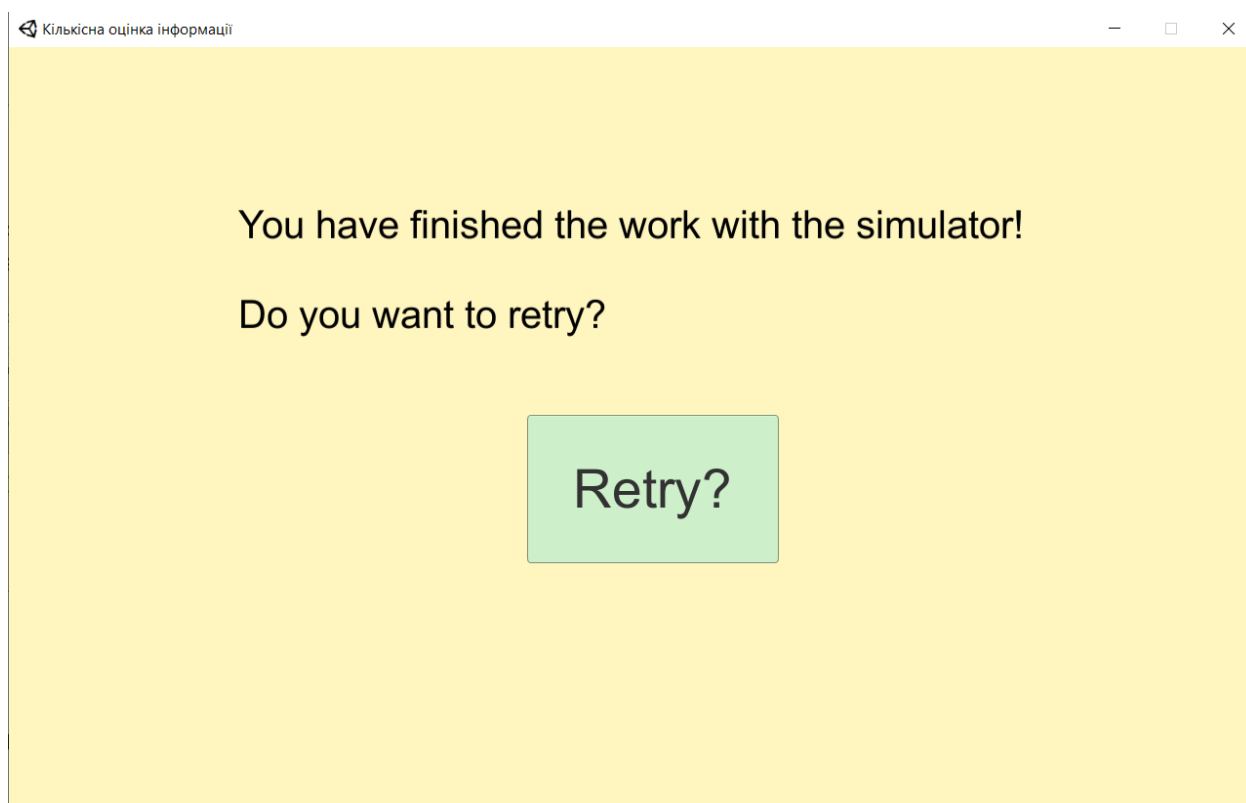


Рисунок 4.23 – Фінальне вікно тренажеру, кнопка «Повтор»

ВИСНОВОК

Складові проведеної роботи:

- Створено постановку задачі;
- Обрано методи для розробки програми;
- Створено алгоритм роботи тренажеру;
- Створено блок-схеми роботи тренажеру;
- Програмно реалізовано елементи програми-тренажеру на платформі Unity 2022;
- Практичні завдання в тренажері реалізовано у вигляді тестів;
- В тренажері реалізовано зручний та інтуїтивно зрозумілий дизайн;
- В тренажері реалізовано перевірку валідності введеної відповіді та вивід відповідного повідомлення;
- Продовження роботи з практичним матеріалом неможливе без вибору правильної відповіді;
- Повтор роботи з тренажером реалізовано через кнопку «Повтор?» в кінці роботи з тренажером;
- Виконано тестування елементів програмного забезпечення, помилок в роботі програми не виявлено.
- Навчальні матеріали тренажеру складаються з практичних тестових завдань з декількома варіантами. При виборі неправильного варіанту відповіді користувач отримує відповідне повідомлення, при виборі правильного варіанту відповіді – повідомлення про правильність відповіді та доступ до кнопки «Далі». Після завершення роботи з тренажером користувач переходить до вікна тренажеру у якому може або завершити роботу через кнопку «Вихід» або повторити роботу за допомогою кнопки «Повтор».

Позитивні аспекти розробленого програмного забезпечення:

- Тренажер запускається на комп'ютерах з будь якою встановленою ОС WINDOWS та на 32 чи 64 розрядних системах.
- При виборі неправильної відповіді тренажер видає повідомлення про помилку.
- Навігація в тренажері зручна та зрозуміла.
- Тренажер відповідає вимогам системи дистанційної освіти MOODLE та переданий для впровадження до курсу «Теорія алгоритмів».
- Повтор роботи тренажеру через відповідну кнопку чи завершення роботи.
- Мету та завдання роботи виконано, створено тренажер для навчання темі «Кількісна оцінка інформації» дистанційного навчального курсу "Теорія інформації та кодування".

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології» галузь знань - 12 «Інформаційні технології» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2017. – 71 с.
2. Ставковий М.Ю. Розробка тренажеру з теми "Метод аналізу ієрархій" для дистанційного навчання / М.Ю. Ставковий // Інформатика та системні науки (ІСН-2014) : матеріали V Всеукр.наук.-практ. конф., (м. Полтава, 13–15 березня 2014 р.). – Полтава: ПУЕТ, 2014. – С. 296- 298. - Режим доступу:<http://dspace.puet.edu.ua/handle/123456789/2843>
3. Гусар М.В. Розробка тренажеру з теми «Знаходження вектору пріоритетів та відношення узгодженості в методі аналізу ієрархій» для дистанційного навчання / М.В.Гусар // Системний аналіз та теорія прийняття рішень : ПУЕТ, 2020. – С. 71 -76. - Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/8993>
4. Мова програмування C# [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – Режим доступу: <http://uk.wikipedia.org/wiki/C#>.
5. Платформа Юніті [Електронний ресурс] . – Режим доступу: <https://unity.com/ru>
6. Unity (ігрова платформа) [Електронний ресурс] . – Режим доступу: [https://ru.wikipedia.org/wiki/Unity_\(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA\)](https://ru.wikipedia.org/wiki/Unity_(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA))
7. ДСТУ 1.5:2015 - Національна стандартизація [Електронний ресурс] . – Режим доступу: <https://udhtu.edu.ua>

Гермейер Ю. Б. Введение в теорию исследования операций. — М.: Наука, 1971. — 384 с. — (Оптимизация и исследование операций). — 22 500 экз.

8. Гермейер Ю. Б., Морозов В. В., Сухарев А. Г., Фёдоров В. В. Задачник по исследованию операций. — М.: Издательство МГУ, 1975.

9. Дегтярёв Ю. И. Исследование операций: учебник для вузов по специальности АСУ. — М.: Высшая школа, 1986.

10. Хемди А. Таха. Введение в исследование операций = Operations Research: An Introduction. — М.: Вильямс, 2007. — 912 с. — ISBN 0-13-032374-8.

11. Грешилов А. А. Математические методы принятия решений. — М.: МГТУ им. Н.Э. Баумана, 2006. — 584 с. — ISBN 5-7038-2893-7.

12. Акоф Р., Сасиени М. Основы исследования операций. — М.: Мир, 1971. — 533 с.

13. Билл Фрэнкс. Революция в аналитике: Как в эпоху Big Data улучшить ваш бизнес с помощью операционной аналитики = The Analytics Revolution: How to Improve Your Business By Making Analytics Operational In The Big Data Era. — М.: Альпина Паблишер, 2016. — ISBN ISBN 978-5-9614-4132-1.

14. Префабы (Prefabs) [Электронный ресурс].— Режим доступа — <https://docs.unity3d.com/ru/530/Manual/Prefabs.html>

ДОДАТОК А

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Inputs : MonoBehaviour
{
    public GameObject Input;
    public GameObject AfterInput;
    public InputField inputedTxt;
    public Text ShowInpTxt;

    public class QuitButton : MonoBehaviour
    {
        public void QuitGame()
        {
            Debug.Log ("QUIT!!");
            Application.Quit();
        }

        void Start()
        {
            Button btn = nxtButton.GetComponent<Button>();
            btn.onClick.AddListener(TaskOnClick);
        }

        void TaskOnClick()
        {
            Theme1.SetActive(false);
            Theme2.SetActive(true);
            Debug.Log("You touched this button.");
        }
    }
}

```

```
void Start()
{
    Button btn = menuButton.GetComponent<Button>();
    btn.onClick.AddListener(TaskOnClick);
}

void TaskOnClick()
{
    Theme6.SetActive(false);
    MainMenu.SetActive(true);
    Debug.Log("You touched this button.");
}

public void Next()
{
    Input.SetActive(false);
    AfterInput.SetActive(true);
    ShowInpTxt.text = inputdTxt.text;
}
}
```