

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

_____ Олена ОЛЬХОВСЬКА

(підпис)

«___»_____2023 р.

ДИПЛОМНА РОБОТА

на тему

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЕЛЕКТРОННОГО
НАВЧАЛЬНОГО СИМУЛЯТОРА З ТЕМИ «МІНІМІЗАЦІЯ
БУЛЕВИХ ФУНКЦІЙ. МЕТОД КВАЙНА-МАК-КЛАСКИ»
ДИСТАНЦІЙНОГО КУРСУ «ДИСКРЕТНА МАТЕМАТИКА»**

зі спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

ступеня магістра

Виконавець роботи Балан Ярослав Дмитрович

_____«___»_____2023р.

(підпис)

Науковий керівник к.ф.-м.н., доц., Парфьонова Тетяна Олександрівна

_____«___»_____2023р.

(підпис)

ПОЛТАВА 2023

АНОТАЦІЯ

Записка: 48 стор., в т.ч. основна частина 46 стор., джерел - 11.

Предмет розробки – електронний навчальний симулятор з теми «Мінімізація булевих функцій. Метод Квайна-Мак-Класки»

Мета роботи – здійснити реалізацію програмного забезпечення електронного навчального симулятора для дистанційного оволодіння знань з теми «Мінімізація булевих функцій. Метод Квайна-Мак-Класки» для навчального курсу «Дискретна математика».

Методи, які були використані для розв’язування задачі – для розроблення програми було обрано середовище IDE MS Visual Studio та платформу Unity 2022 та мову програмування C#.

ЗМІСТ

ВСТУП.....	5
1. ПОСТАНОВКА ЗАДАЧІ.....	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Огляд тренажерів з подібною тематикою та задачами	8
2.2. Позитивні сторони оглянутих робіт.....	16
2.3. Негативні сторони оглянутих робіт	16
2.4. Необхідність та актуальність теми.....	16
3. ТЕОРЕТИЧНА ЧАСТИНА.....	17
3.1 Основні методи мінімації логічних функцій	17
3.2 Мінімізація булевих функцій за методом Квайна-Мак-Класки.....	21
3.3 Алгоритм роботи тренажеру.....	27
3.4. Блок-схема програми тренажера	30
4. ПРАКТИЧНА ЧАСТИНА	31
4.1 Опис розробки тренажера	31
4.2 Інструкція до роботи з тренажером	38
ВИСНОВОК.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовного позначення, скорочень, символів
Тренажер	Комп'ютерна програма-вчитель, призначена для вивчення і закріплення знань з різноманітних тем.
<i>ДК</i>	Дистанційний курс.
<i>ДН</i>	Дистанційне навчання.
<i>C#</i>	Мова програмування.
<i>Unity 2021</i>	Платформа для програмної реалізації.
<i>IDE MS Visual Studio</i>	Середовище для написання програмного коду.

ВСТУП

В наш час впровадження інформаційних та телекомунікаційних засобів навчання є особливо актуальним. Це відкриває можливості для користувачів навчатися самостійно, власним темпом та в будь-якому зручному місці за допомогою електронних пристроїв, таких як телефон, комп'ютер чи планшет.

Новітні технології навчання мають численні переваги, такі як доступність, різноманітність подачі матеріалу, ефективність та зручність.

Магістерська робота, над якою ведеться розробка, стосується створення тренажера для дистанційного курсу «Дискретна математика». Основна мета цього тренажера - надання засобу для вивчення та удосконалення знань студентів в галузі «Мінімізація булевих функцій. Метод Квайна-Мак-Класки». Розроблений тренажер визначається як важливий інструмент для вдосконалення навичок визначення меж системи, структурування завдань, співставлення рішень з зовнішніми умовами, а також агрегації критеріїв системного аналізу та вибору формалізації для вирішення проблеми.

Мета цієї роботи полягає в програмній реалізації симулятора для дистанційного навчання, об'єкт роботи - створення програми для забезпечення дистанційного навчання, а предмет роботи - програма-симулятор на тему «Мінімізація булевих функцій. Метод Квайна-Мак-Класки». В процесі розробки програми використовується середовище IDE MS Visual Studio та платформа Unity 2022, а також мова програмування C#.

Отже, створення імітаційних засобів для вивчення та оптимізації булевих функцій є актуальним завданням, а розроблений тренажер відповідає цим вимогам, допомагаючи студентам системно оволодіти матеріалом та навичками в даній галузі.

1. ПОСТАНОВКА ЗАДАЧІ

У епоху сучасного інформаційного розвитку набуває все більшого значення вдосконалення методів навчання, зокрема в області дискретної математики та оптимізації булевих функцій. З метою досягнення цієї мети планується створення програмно-імітаційного інструменту, що базується на методі Квайна-Мак-Класки, спрямованого на оптимізацію булевих функцій.

Завдання та етапи реалізації:

1. Створення алгоритму роботи програми: Розробка алгоритму, що базується на методі Квайна-Мак-Класки, включаючи основні етапи обробки та оптимізації булевих функцій.

2. Розробка практичних та теоретичних завдань: Створення практичних завдань для закріплення отриманих знань та теоретичних матеріалів для підвищення ефективності навчання.

3. Складання блок-схеми алгоритму роботи тренажеру: Графічне відображення алгоритму програми, сприяючи кращому розумінню та уточненню його деталей.

4. Вибір мови програмування та програмна реалізація: Вибір мови програмування C# для створення електронного начального симулятора.

5. Розробка зручного інтерфейсу для навігації: Створення інтуїтивно зрозумілого інтерфейсу для зручності користувачів та полегшення навігації в програмі.

6. Тестування та перевірка програми на валідність: Проведення тестів для перевірки коректності роботи програми та відповідності поставленим завданням.

7. Передбачення та усунення можливих помилок: Аналіз можливих помилок у роботі програмного забезпечення та розробка механізмів для їх усунення.

Очікувані результати: Очікується, що розроблений тренажер стане ефективним інструментом для самостійного вивчення та оптимізації булевих

функцій студентами, а також забезпечить можливість докладного вивчення методу Квайна-Мак-Класки.

Заключення: Робота спрямована на поліпшення процесу навчання та розвитку студентів у галузі дискретної математики через впровадження інноваційного програмного тренажера на основі методу Квайна-Мак-Класки для мінімізації булевих функцій.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд тренажерів з подібною тематикою та задачами

Було обрано роботи з електронного архіву університету, які мають подібну тематику, схожі завдання та програмну реалізацію, для подальшого використання в майбутньому проекті. На основі цих критеріїв було виявлено два тренажери:

1. «Побудова блок-схем алгоритмів лінійної структури», створений Шакуро Вадимом для освітнього курсу «Програмування П» [2]
2. «Побудова блок-схем алгоритмів циклічної структури» дистанційного навчального курсу «Інформатика», розробник Бибка Богдан [3]

Тренажер побудови блок-схем алгоритмів лінійної структури включає три умови – алгоритми, які представлені кодами мови Object Pascal і вимагають створення блок-схем.

Початкова умова виводиться на етапі роботи (див. Рисунок 2.1.1). На кожному етапі користувачеві потрібно обрати правильну відповідь із трьох варіантів. Якщо відповідь вибрана неправильно, вона підсвічується червоним кольором (див. Рисунок 2.1.2); в іншому випадку вона позначається зеленим (див. Рисунок 2.1.3). Система не надає пояснень щодо помилок, і студент має самостійно виправляти свої помилки.

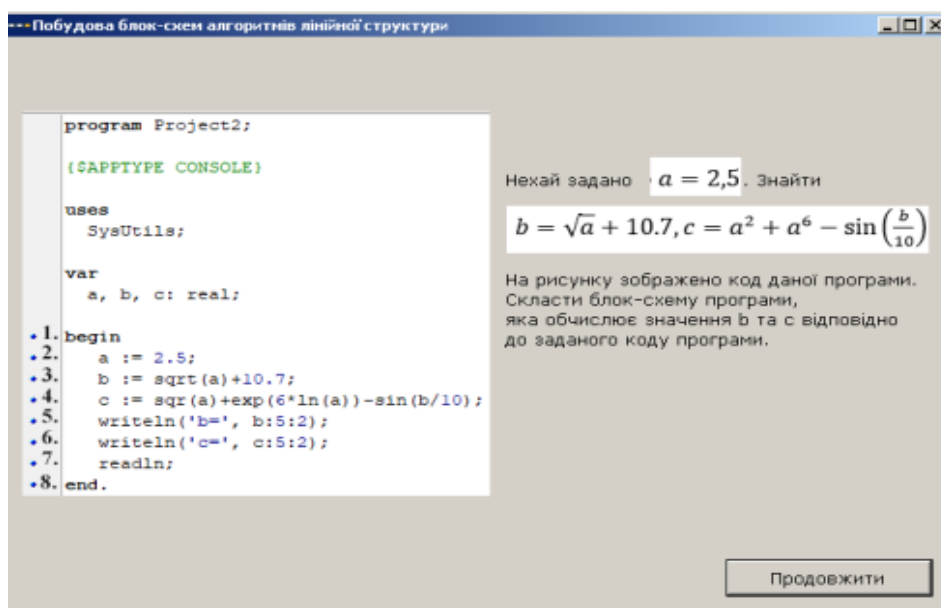


Рисунок 2.1.1- Умова першого завдання

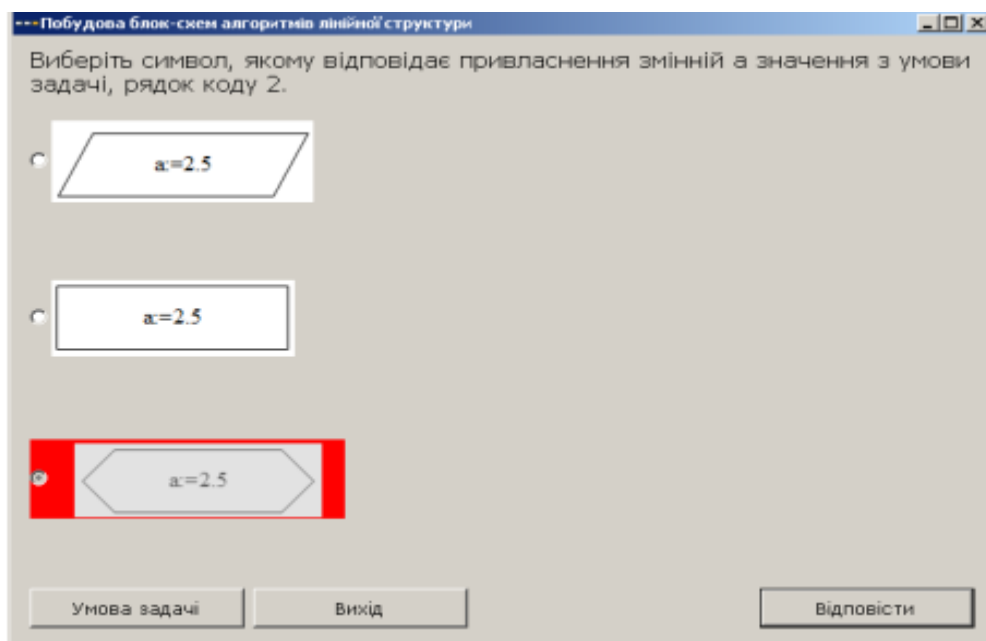


Рисунок 2.1.2- Неправильна відповідь на другому кроці

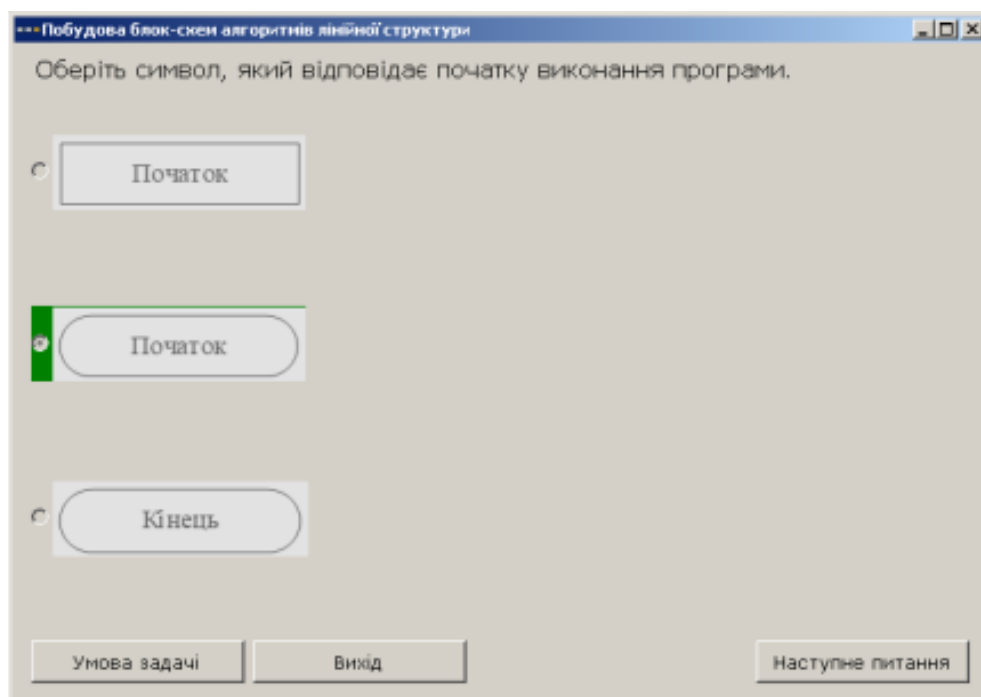


Рисунок 2.1.3- Правильна відповідь на першому кроці

Під час кожного етапу користувач може повернутися до початкової умови завдання, скориставшись відповідною кнопкою. Для завершення роботи тренажеру

із збереженням результатів служить кнопка "Вихід". Перехід до наступної умови виконується через відповідну кнопку, розташовану в нижньому правому куті екрану. Цей перехід здійснюється, незалежно від того, чи було правильно виконано попереднє завдання.

На завершення проходження студенту пропонується огляд підсумкового малюнка, що відображає правильно побудовану блок-схему за заданим алгоритмом (див. Рисунок 2.1.4).

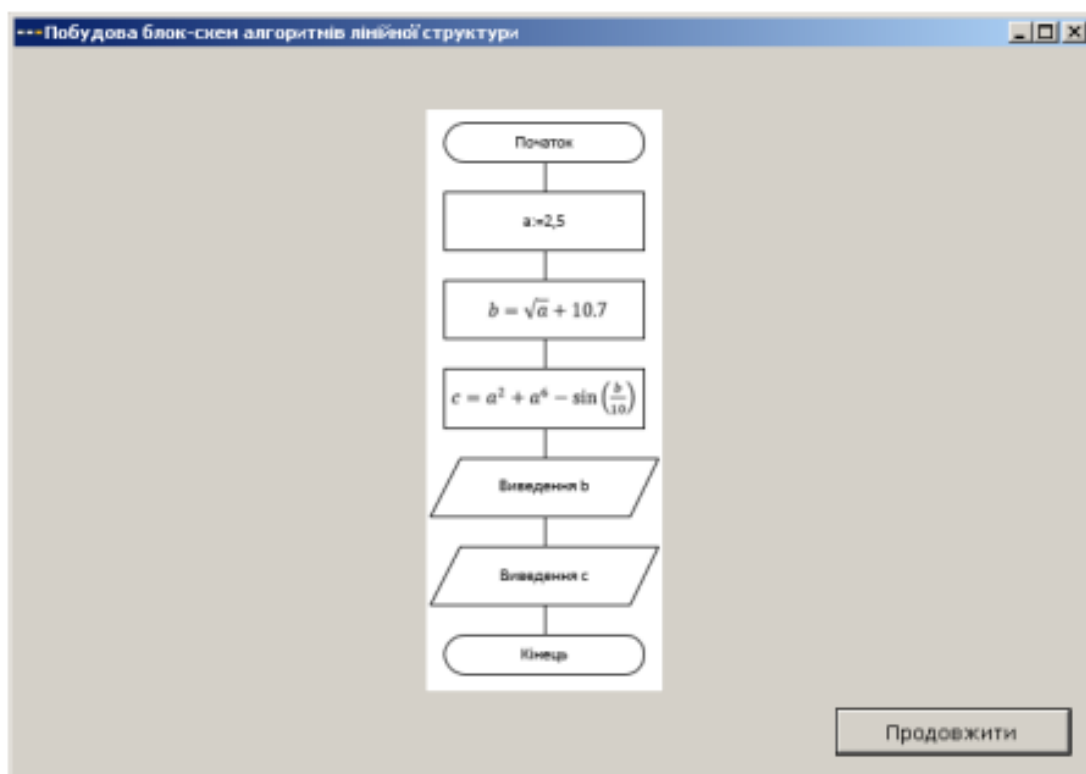


Рисунок 2.1.4- Підсумкова блок-схема (крок 8)

Результати функціонування тренажеру для "Побудова блок-схем алгоритмів циклічної структури" можна оглянути на рисунку 2.1.5. У вихідному вікні користувачеві надається можливість обрати один із двох прикладів та почати тренінг (див. Рисунок 2.1.5). Після цього виводиться умова, представлена кодом програми з циклічним оператором (див. Рисунок 2.1.6).

Задачею користувача є створення блок-схеми. Йому доступно повернення до умови на кожному етапі тренажеру, використовуючи відповідну кнопку.



Рисунок 2.1.5- Стартове вікно тренажеру

Побудувати блок-схему алгоритму, заданого алгоритмічною мовою Object Pascal:

```

const
  N = 5;
var
  a, Sum, i: integer;
BEGIN
  Sum := 0;
  i := 1;
  while (i <= N) do
  begin
    write('Введіть число: ');
    readln(a);
    Sum := Sum + a;
    i := i + 1;
  end;
  writeln('Сума =', Sum);
  readln;
END.
  
```

Тренінг

Рисунок 2.1.6- Умова завдання

На кожному етапі необхідно вибрати правильну відповідь, переносячи обраний малюнок на прямокутник жовтого кольору, після чого слід натискати кнопку «Далі» (див. Рисунок 2.1.7).

Якщо відповідь не вибрана, а кнопка «Далі» вже натиснута, з'являється попередження про некоректність дій користувача (див. Рисунок 2.1.8). У випадку правильної відповіді (див. Рисунок 2.1.9), виводиться відповідне повідомлення (див. Рисунок 2.1.10).

При виникненні помилки з'являється повідомлення з тлумаченням (див. Рисунок 2.1.11). У цьому випадку користувачу необхідно виправити помилку і знову натискати кнопку «Далі». Після кожної правильної відповіді відображується блок-схема з обраним символом (див. Рисунок 2.1.12).

Після завершення останнього етапу виводиться повна блок-схема відповідно до заданої умови в правильному вигляді (див. Рисунок 2.1.13).

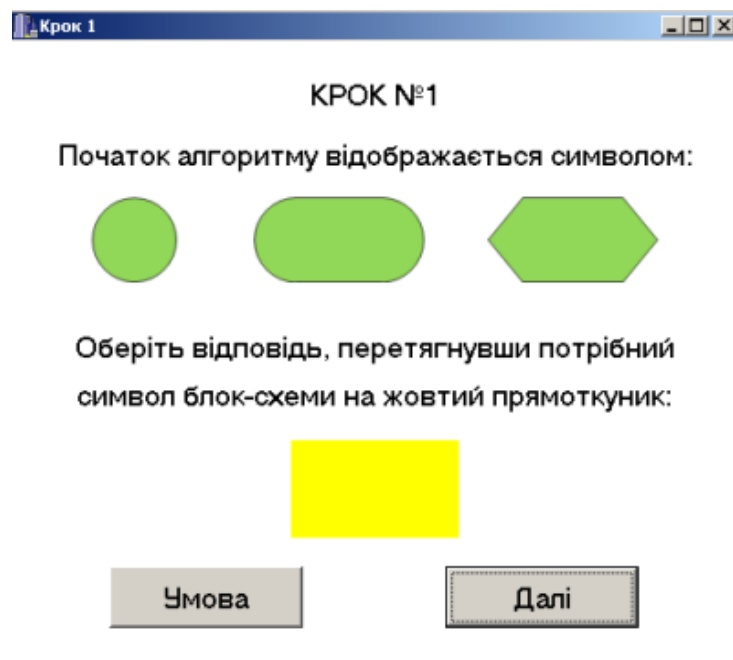


Рисунок. 2.1.7- Приклад завдання (крок 1)

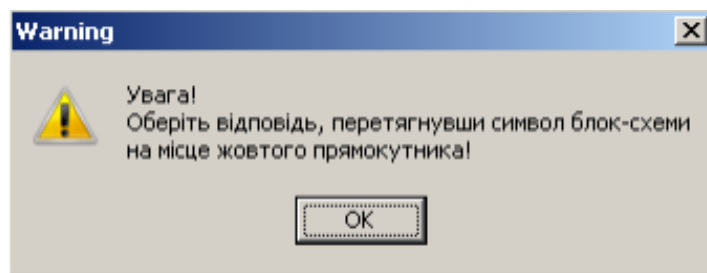


Рисунок 2.2.8- Попередження про некоректні дії користувача

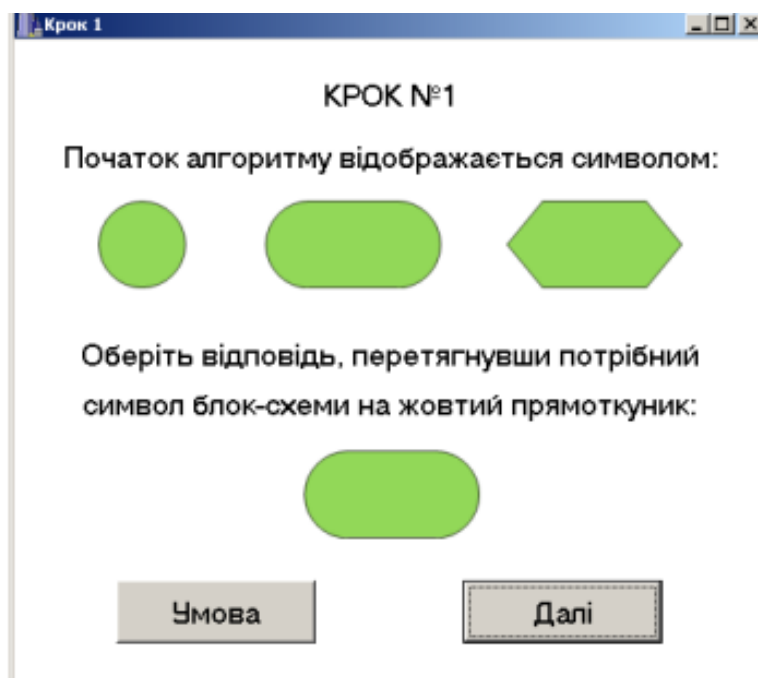


Рисунок 2.1.9- Правильна відповідь

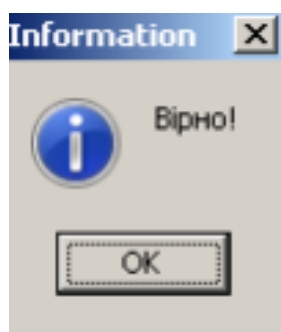


Рисунок 2.1.10- Повідомлення про правильність відповіді

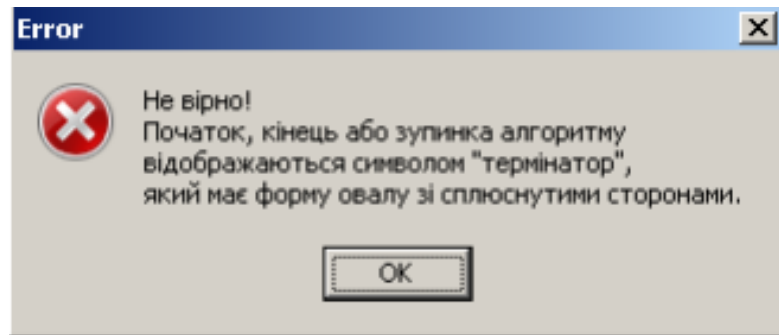


Рисунок 2.1.11- Тлумачення помилки



Рисунок 2.1.12- Блок-схема після проходження другого завдання

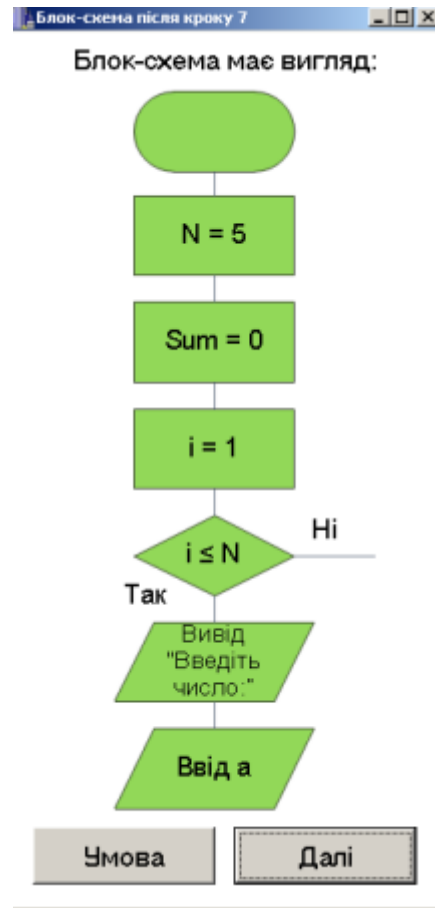


Рисунок 2.1.13- Блок-схема після завершення тренінгу

2.2. Позитивні сторони оглянутих робіт

Позитивні аспекти вивчених робіт включають:

- забезпечення доступу до умови на кожному етапі тренування;
- використання колірної дизайну програми, що значно підвищує зацікавленість користувача;
- наявність різноманітних прикладів, що сприяє глибшому засвоєнню теми;
- відображення правильної підсумкової блок-схеми;
- належне функціонування повідомлень про помилкові та правильні відповіді;
- в програмі №2 наявність повідомлень із тлумаченням помилок.

2.3. Негативні сторони оглянутих робіт

Негативні аспекти при використанні тренінгів включають:

- Програма №1 містить помилки у коді програми, і вона не має функціоналу для відображення повідомлень з поясненням помилок.
- В обох програмах відсутні блоки з теоретичним матеріалом по темі.
- Завдання в тренінгах є одноманітними.

2.4. Необхідність та актуальність теми

Схожих програм-симуляторів в мережі Інтернет та репозитарії DSpace, що відповідають обраній темі, під час огляду не виявлено. Тренажери, які мають схожу тематику, мають вагомні недоліки. Таким чином навчальний симулятор «Мінімізація булевих функцій. Метод Квайна-Мак-Класки" відзначиться підвищенням рівня засвоєння цієї теми студентами університету, зробивши його необхідним.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Основні методи мінімації логічних функцій

Під мінімізацією логічної функції розуміється виконання перетворень з метою отримання найбільш простого уявлення функцій алгебри логіки. В інженерній практиці використовуються такі основні методи мінімізації:

- послідовного перебору
- послідовного спрощення аналітичного виразу
- карт Карно
- Квайна
- Квайна - Мак-Класки
- Л.Т. Мавренкова

Метод послідовного перебору заснований на простому переборі всіх можливих варіантів алгебраїчного виразу функції булевої алгебри. Використання методу не потребує спеціальних знань, проте, він застосовується лише для дуже простих функцій алгебри логіки, утворених від однієї або двох змінних. Даний метод через невиправдану кількість ресурсів для реалізації відразу відкинемо.

Метод послідовного спрощення аналітичного вираження базується на перетворенні функції алгебри логіки з використанням основних законів і тотожностей булевої алгебри. Перевагою методу є можливість його застосування для мінімізації будь-яких функцій алгебри логіки, представлених у вигляді аналітичного виразу. Однак розглянутий метод досить трудомісткий для виконання вручну, мало наочний та вимагає великого досвіду, уваги і інтуїції і, отже, легко призводить до виникнення помилок.

Карти Карно - це метод спрощення булевих виразів алгебри логіки.

Карта Карно зменшує потребу в великих обчисленнях, оскільки використовує можливості аналітичних функцій людини та їх здібності до розпізнавання образів через що є дуже наочним. Метод, заснований на застосуванні карт

Карно, передбачає завдання функцій алгебри логіки у вигляді координатних карт станів. Після запису функції в карту Карно відразу можна записати мінімальну форму функції, що істотно зменшує ймовірність появи помилки.

Метод Квайна заснований на послідовному застосуванні до ДСНФ операцій неповного склеювання і елементарного поглинання. Метод чудово підходить в основному для функцій алгебри логіки від невеликої кількості аргументів.

Метод Квайна-Мак-Класкі в своїй основі використовує запропонований Мак-Класкі зручний підход до практичної реалізації алгоритму Квайна. Під час використання даного методу функція алгебри логіки представляється однією з досконалих нормальних форм. Проте відмінність від метода Квайна полягає в тому, що конституенти одиниці та конституенти нуля заданої функції записуються у вигляді умовних чисел, які називаються номерами відповідних конституент, а не в натуральному вигляді. Такий підхід суттєво спрощує обробку великої кількості аргументів і значно зменшує ймовірність появи помилки на етапі виконання перетворень.

Як і метод Квайна-Мак-Класкі – метод Л.Т. Мавренкова базується на використанні умовних чисел, які відповідають різним наборам значень аргументів для уявлення конституент одиниць. Наочне відображення методу полягає у поданні зв'язків між двійковими сусідніми з використанням спеціальної решітки. За допомогою операції склеювання робочих наборів, розташованих вздовж горизонтальних та вертикальних ліній решітки, відбувається мінімізація даної функції.

Як один із варіантів можна розглянути метод невизначених коефіцієнтів. Метод невизначених коефіцієнтів є найбільш трудомістким з усіх представлених. Для мінімізації функції за допомогою цього методу бажано мати повну таблицю наборів чисел в ДСС. Знаходження мінімальної ДНФ зводиться до вирішення системи лінійних рівнянь і складається з наступних етапів:

- Складаються рівняння з усіх наборів коефіцієнтів від 1 до N
- змінних.

- Викидаються рівняння, в яких значення функції дорівнює нулю, а
- також нульові коефіцієнти
- Виконується пошук найбільш часто зустрічаються терм
- мінімального рангу.
- Видаляються повторювані коефіцієнти.

Розглянутий метод невизначених коефіцієнтів ефективний, якщо число аргументів функції не більше, ніж 5 - 6. Це пов'язано з тим, що число рівнянь дорівнює 2^N в ступені N. Більш ефективним є виписування не всіх можливих кон'юнкцій для функції, а тільки тих, які можуть бути присутніми в ДНФ даної функції. На цьому заснований метод Квайна. При цьому передбачається, що функція задана у вигляді СДНФ. В даному методі елементарні кон'юнкції рангу N, що входять до ДНФ, називаються мінітермами рангу N. Оскільки методи послідовного аналітичного спрощення та послідовного перебору дуже ресурсомісткі і важкореалізуємі в вигляді програмного коду розглянемо детальніше реалізацію карт Карно та методу Квайна-Мак-Класкі.

Використання карт Карно дозволяє виконувати мінімізацію логічної функції в процесі її складання. Для мінімізації логічного виразу в карті Карно виділяються всі контури містять по 2 в ступені N одиниць. Якщо який-небудь контур перетинає зміни значення однієї або декількох змінних, то ці змінні не вносяться в логічну функцію для даного контуру. Карта заповнюється одиницями, кількість яких дорівнює кількості доданків в СДНФ, кожний доданок - 1 одиниця.

Для карт Карно існують правила мінімізації:

- Необхідно покрити контурами усі групи одиниць (ДНФ) у карті
- Карно. В рамках кожного контуру повинні знаходитися лише одиниці – це відповідає операції склеювання - знаходження
- імплікант даної функції.
- Кількість одиниць контуру має бути ступенем двійки (1, 2, 4, 8, ...).
- Кожен контур буде відповідати простий імпліканті лише якщо він

- буде покривати максимально можливу кількість клітинок.
- Контури повині включати всі одиниці в карті (навіть поодинокі).
- Будь-яка одиниця може включатися безмежну кількість разів в контури.
- Безліч контурів, що покривають всі одиниці функції, утворюють кінцеву ДНФ.
- Етапи мінімізації:
- Перехід від ДНФ до досконалої ДНФ. Членами будуть несклеювані елементарні кон'юнкції.
- Перехід до кінцевої форми.
- Перехід до мінімальної ДНФ.

На карті Карно мінімізація в своїй основі тримає ідею вибору контурів таким чином, щоб контури не перетинали кордон зміни якомога більшої кількості змінних. При дотриманні цієї умови, кінцева функція вийде максимально мінімальною.

Як ми бачимо метод карт Карно є зручним методом мінімізації булевих функцій до 5 змінних. Але важко спростити булеві функції, що мають більше 5 змінних, використовуючи цей метод.

Метод Квайна-Мак-Класкі є табличним методом, заснованим на концепції простих імплікантів. Ми знаємо, що простий імплікант - не може бути додатково зменшений шляхом об'єднання з будь-яким іншим з членів даної булевої функції. Цей табличний метод корисний для отримання простих імплікантів шляхом повторного використання булевої операції поглинення.

Для спрощення булевих функцій за допомогою табличного методу Квайна-Мак-Класкі необхідно виконати наступні кроки: Впорядковати задані терми в порядку зростання і створити групи на основі кількості присутніх одиниць у їх двійкових поданнях. Таким чином, отримуємо, щонайменше $N+1$ груп, при умові що є N булевих змінних у булевій функції або, як у випадку програмування алгоритму, N біт у двійковому еквіваленті мінімальних термінів. Порівняти

мінімальні терми, присутні в наступних групах. Якщо відбувається зміна тільки в одній бітовій позиції, то візьмемо пару з цих двох мінімальних терм. На отриманій позиції вионати поглинення, а інші біти залишити в тому ж вигляді. Далі повторюємо попередній крок з новоутвореними термами, доки не буде отримано всіх простих імплікантів. Сформулювати таблицю простих імплікант. Вона буде складатися з набору рядків і стовпців. Основні імпліканти можуть бути розміщені в рядку, а терми можуть бути поміщені в колонку. Помістити «1» у клітинки, що відповідають мінімальним термам, які покриваються кожним простим імплікатом. Знайти прості імпліканти, переглянувши кожний стовбчик. Якщо мінімальний терм покривається тільки одним простим імплікантом, то він є кінцевим простим імплікантом. Ці незамінні імпліканти будуть частиною спрощеної булевої функції. Зменшити таблицю основних імплікантів, видаливши рядок кожного основного первинного імпліканта і стовпці, що відповідають мінімальним термінам, які охоплюються цим основним імплікатором. Повторити попередній крок для таблиці зменшених основних імплікант. Зупинити цей процес, коли всі терми виконання цієї булевої функції закінчаться.

3.2 Мінімізація булевих функцій за методом Квайна-Мак-Класки

Недолік методу Квайна полягає в необхідності попарного порівняння всіх термів на першому етапі при знаходженні первинних імплікант. Із збільшенням кількості вихідних термів збільшується кількість попарних порівнянь, що ускладнює розв'язок.

Метод Квайна-Мак-Класки є модифікацією методу Квайна. Він заснований на кубічному поданні термів ДНФ із попередньою розбивкою кубів на підгрупи, обумовлені однаковим числом одиниць. Розбивка дає можливість порівнювати куби тільки за сусідніми за числом одиниць групами для зменшення кількості переборів.

В ітеративній процедурі мінімізації попарне порівняння можна виконувати тільки між сусідніми групами.

Вихідне завдання функції визначається для зручності десятковими кодами двійкових кубів, що відповідають ДНФ.

Знаходження первинних імплікант на першому етапі можна спростити за допомогою числового зображення булевих функцій, а саме:

1. Всі вихідні терми записуються у вигляді їхніх двійкових номерів.
2. Всі номери розбиваються на непересічні групи за числом одиниць. Умовою утворення r -куба є наявність розбіжності в $(r-1)$ -кубах тільки за однією координатою в одному двійковому розряді й наявність спільних незалежних координат.
3. В i -групу включають всі номери наборів, що мають у своєму записі i одиниць.
4. Попарне порівняння виконується тільки між сусідніми за номером групами. Групи, які розрізняються за двома розрядами і більше, не має сенсу порівнювати.

Приклад 1. Потрібно мінімізувати логічну функцію за методом Квайна-Мак-Класки:

$$f(x_1, x_2, x_3, x_4) = \sum_1(3,4,5,7,9,11,12,13) \quad (3.1)$$

Розв'язок. Функція зображена числовим способом. На підставі таблиці істинності можна подати функцію в кубічній формі. Для цього слід виписати всі 0-куби, що утворюють комплекс K^0 :

$$K^0 = \begin{array}{c} \boxed{0011} \\ \boxed{0100} \\ \boxed{0101} \\ \boxed{0111} \\ \boxed{1001} \\ \boxed{1011} \\ \boxed{1100} \\ \boxed{1101} \end{array}$$

Рисунок 3.1 - Всі 0-куби, що утворюють комплекс K^0 :

Слід розбити комплекс K^0 на групи за кількістю одиниць у кожному двійковому наборі. Таких груп буде три:

$$K_1^0 = \{0100\}, \quad K_2^0 = \begin{bmatrix} 0011 \\ 0101 \\ 1001 \\ 1100 \end{bmatrix}, \quad K_3^0 = \begin{bmatrix} 0111 \\ 1011 \\ 1101 \end{bmatrix}$$

Рисунок 3.2 - Розбиття комплексу K^0 на групи

1) Визначення первинних імплікант.

а) Порівняння сусідніх за номером груп кубів – склеювання кубів (поглинена координата замінюється на X):

$$K_1^0 \vee K_2^0 = K_{12}^1 = \begin{bmatrix} 010X \\ X100 \end{bmatrix}$$

$$K_2^0 \vee K_3^0 = K_{23}^1 = \begin{bmatrix} 0X11 \\ X011 \\ 01X1 \\ X101 \\ 10X1 \\ 1X01 \\ 110X \end{bmatrix}$$

Рисунок 3.3 – Склеювання кубів

б) Всі 1-куби розбиваються на групи за принципом розташування незалежної координати X (нижній індекс відповідає порядковому номеру розташування незалежної координати X):

$$K_1^1 = \begin{bmatrix} X011 \\ X100 \\ X101 \end{bmatrix}, \quad K_2^1 = \begin{bmatrix} 0X11 \\ 1X01 \end{bmatrix}, \quad K_3^1 = \begin{bmatrix} 01X1 \\ 10X1 \end{bmatrix}, \quad K_4^1 = \begin{bmatrix} 0010X \\ 110X \end{bmatrix}$$

Рисунок 3.4 – Розбиття кубів на групи

Куби всередині кожної групи порівнюються з метою одержання K^2 -кубів.

Порівняння кубів усередині комплексу K_1^1 дає $K_{11}^1 = \begin{bmatrix} X10X \\ X011 \end{bmatrix}$, а порівняння усередині K_4^1 дає $K^2 = \{ X10X \}$, що вже міститься в K_{11}^1 , тобто отримана первинна імпліканта другого рангу $x_2\bar{x}_3$.

2) Складання таблиці й розміщення міток. Складається таблиця вихідних термів і тих імплікант, які не брали участь у склеюванні. Якщо у вихідний терм входить яка-небудь первинна імпліканта, то на перетинанні відповідного рядка й стовпця вказується мітка «*» (табл. 3.1).

Таблиця 3.1 – Складання таблиці й розміщення міток

Номери стовпців	1	2	3	4	5	6	7	8
Набори	0011	0100	0101	0111	1001	1011	1100	1101
0X11	*			*				
X011	*					*		
01X1			*	*				
10X1					*	*		
1X01					*			*
X10X		*	*				*	*

3) Визначення істотних імплікант. Істотна імпліканта визначається єдиною міткою в якому-небудь стовпці таблиці. Істотною імплікантою 2-го рангу є

терм $x_2\bar{x}_3$, що відповідає 2-кубу $X10X$. Стовпці, що відповідають істотній імпліканті, відокремлюються (табл. 3.2).

Таблиця 3.2 – Визначення істотних імплікант і видалення зайвих стовпців

	Номери стовпців	1	2	3	4	5	6	7	8
	Набори	0011	0100	0101	0111	1001	1011	1100	1101
A	0X11	*			*				
B	X011	*					*		
C	01X1			*	*				
D	10X1					*	*		
E	1X01					*			*
	X10X		*	*				*	*

4-й і 5-й етапи, які відповідають викреслюванню зайвих стовпців і зайвих первинних імплікант, відсутні.

б) Вибір мінімального покриття. Виділяються стовпці відповідні істотним імплікантам (у цьому випадку тільки однієї істотної імпліканти). Складається таблиця термів та імплікант, які залишилися невиділеними. Вирішується завдання мінімального покриття рядків стовпцями (табл.3.3).

Таблиця 3.3 – Покриття рядків стовпцями

	Номери стовпців	1	4	5	6
	Набори	0011	0111	1001	1011
A	0X11	*	*		
B	X011	*			*
C	01X1		*		
D	10X1			*	*
E	1X01			*	

Таким чином, обране мінімальне покриття рядків стовпцями: $\{0X11\}$, $\{10X1\}$. Воно відповідає термам $\bar{x}_1x_3x_4$, $x_1\bar{x}_2x_4$. З урахуванням отриманої раніше істотної імпліканти $x_2\bar{x}_3$ остаточне мінімальне зображення функції має вигляд: $f(x_1, x_2, x_3, x_4) = x_2\bar{x}_3 \vee \bar{x}_1x_3x_4 \vee x_1\bar{x}_2x_4$.

Отже, методи мінімізації булевих функцій використовуються у всіх програмних додатках, пов'язаних із синтезом обчислювальних пристроїв. Вони дозволяють у середньому на 20 – 30% одержати більш економічний проект із позиції апаратних витрат.

Найбільш практично орієнтованим є метод Квайна-Мак-Класки, що оперує кубічним зображенням булевих функцій. Недоліком обох методів є застосування імплікантної таблиці для розв'язання завдання знаходження мінімального покриття, що вимагає великого обсягу пам'яті для реальних об'єктів.

3.3 Алгоритм роботи тренажеру

Крок 1. Запуск тренажеру.

Крок 2. Перехід до навчальних матеріалів.

Крок 3. Видача навчального матеріалу.

Завдання 1. Яка основна мета методу Квайна-Мак-Класки у мінімізації булевих функцій?

- а) Збільшення кількості термінів у функції.
- б) Зменшення кількості термінів у функції.
- с) Збільшення розміру булевої таблиці.

Відповідь: б

Завдання 2. Які з перерахованих етапів не входять у процес мінімізації методом Квайна-Мак-Класки?

- а) Визначення множин ваг булевої функції.
- б) Побудова вагованого графа Квайна.
- с) Знаходження локальних максимумів ваг функції.

Відповідь: с

Завдання 3. Що знаходиться на ребрах графа Квайна?

- а) Терміни булевої функції.
- б) Множини ваг.
- с) Глобальні максимуми ваг функції.

Відповідь: а

Завдання 4. Що таке локальний максимум ваг функції в методі Квайна-Мак-Класки?

- а) Термін, який максимізує вагу всієї функції.
- б) Термін, який максимізує вагу в конкретному контексті.
- с) Термін, який завжди має найбільшу вагу.

Відповідь: б

Завдання 5. Як визначається критерій закінчення процесу мінімізації методом Квайна-Мак-Класки?

- а) Зупинка, коли всі терміни функції стають глобальними максимумами.
- б) Зупинка, коли ваги не змінюються протягом певного кількості ітерацій.
- с) Зупинка, коли функція стає константою.

Відповідь: б

Завдання 6. Які переваги має метод Квайна-Мак-Класки порівняно з іншими методами мінімізації булевих функцій?

- а) Зменшення кількості використовуваних термінів.
- б) Зручність у визначенні ваг та використання графічного представлення.
- с) Відсутність обмежень на розмір булевої функції.

Відповідь: б

Завдання 7. Визначте мінімальну кількість термінів, яку може мати мінімізована булева функція.

- а) 1
- б) 2
- с) 0

Відповідь: а

Завдання 8. Які умови повинні бути виконані для того, щоб термін став глобальним максимумом в методі Квайна-Мак-Класки?

- а) Вага терміна повинна бути найбільшою серед всіх термінів.
- б) Вага терміна повинна бути найменшою серед всіх термінів.

с) Вага терміна повинна бути найбільшою в межах конкретного контексту.

Відповідь: а

Завдання 9. Для чого використовують поля для вводу відповіді при мінімізації булевих функцій? Введіть варіант відповіді в полі знизу.

а) Введення ваг термінів.

б) Введення самої булевої функції.

с) Введення числових значень ваг для аналізу.

Відповідь: б

Завдання 10. Знайдіть мінімізований вигляд для функції: $F(A, B, C) = \Sigma(0, 1, 2, 3, 5, 7)$. Введіть варіант відповіді в полі знизу.

а) $F(A, B, C) = A' + B' + C'$

б) $F(A, B, C) = A'BC' + AB'C' + ABC$

с) $F(A, B, C) = A'B'C' + AB + AC'$

Відповідь: а

Крок 4. Після завершення роботи користувач може повторити роботу з тренажером.

3.4. Блок-схема програми тренажера



Рисунок 3.5 – Блок-схема алгоритму роботи тренажера

4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки тренажера

Початок процесу розробки тренажеру передбачає створення проекту в середовищі Unity та його початкове налаштування.

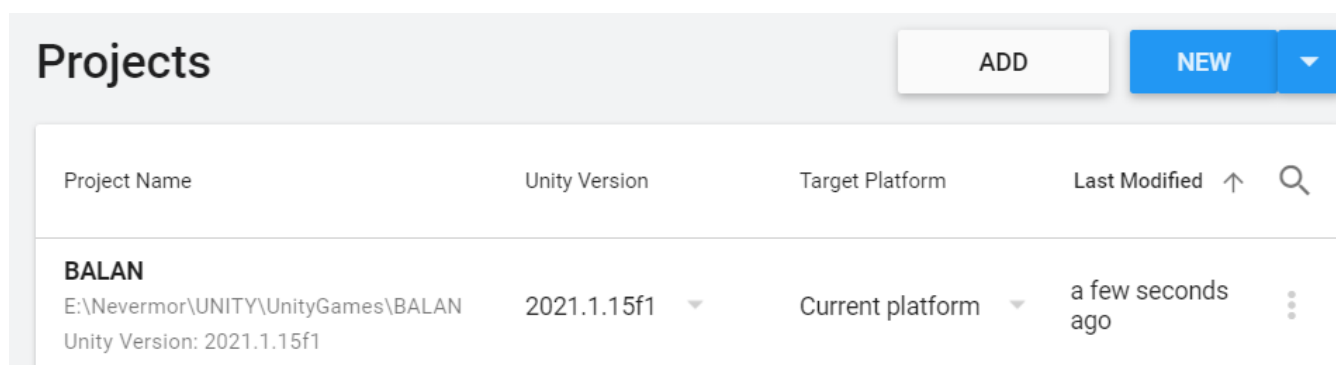


Рисунок 4.1 – Створення проекту в Юніті

Після створення проекту та активації компонентів, а також додавання тексту, кнопок і скриптів, вихідне меню тренажеру в області робочого простору програми виглядає наступним чином:

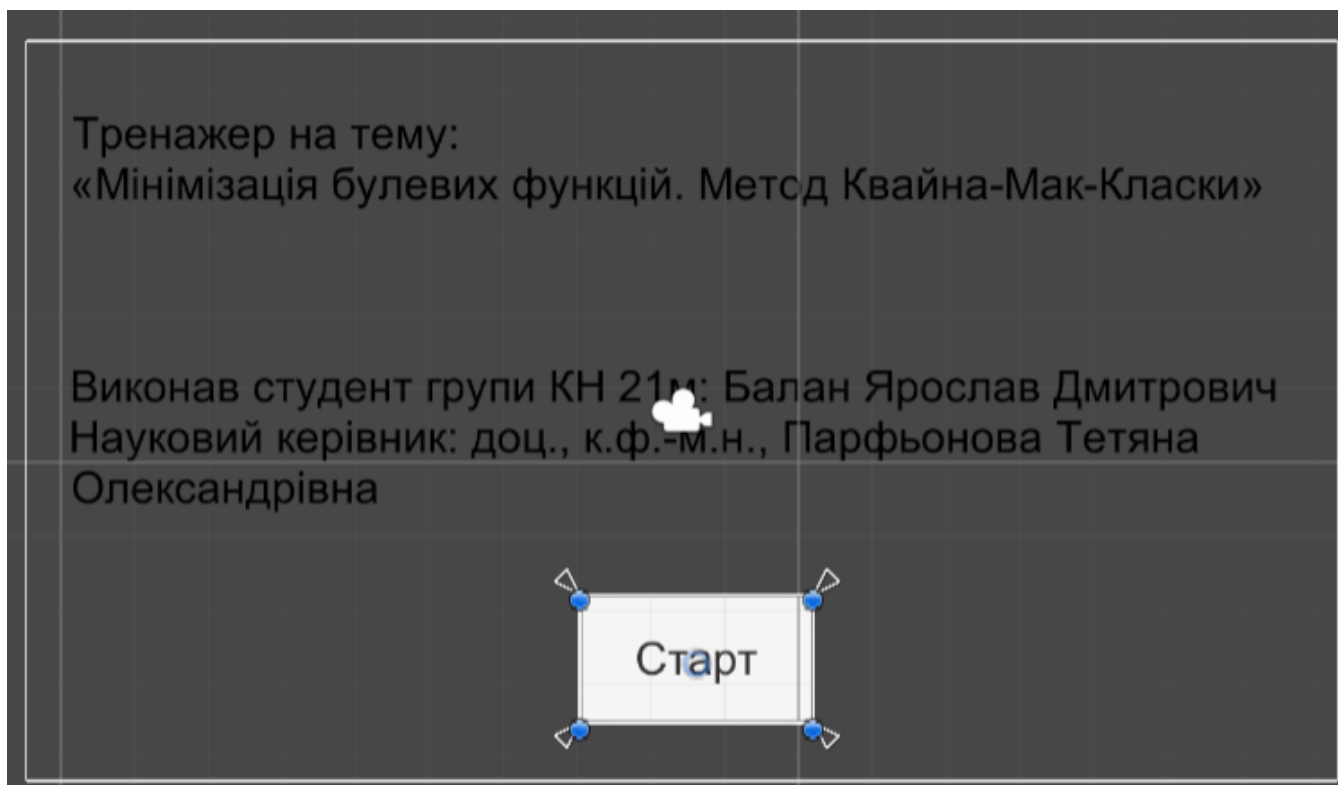


Рисунок 4.2 – Стартове меню тренажеру в меню сцени

Для забезпечення правильного відображення тренажера на різних моніторах, за винятком дуже малих екранів, потрібно налаштувати параметри камери в середовищі Unity. Параметри камери для цього проекту мають такий вигляд:

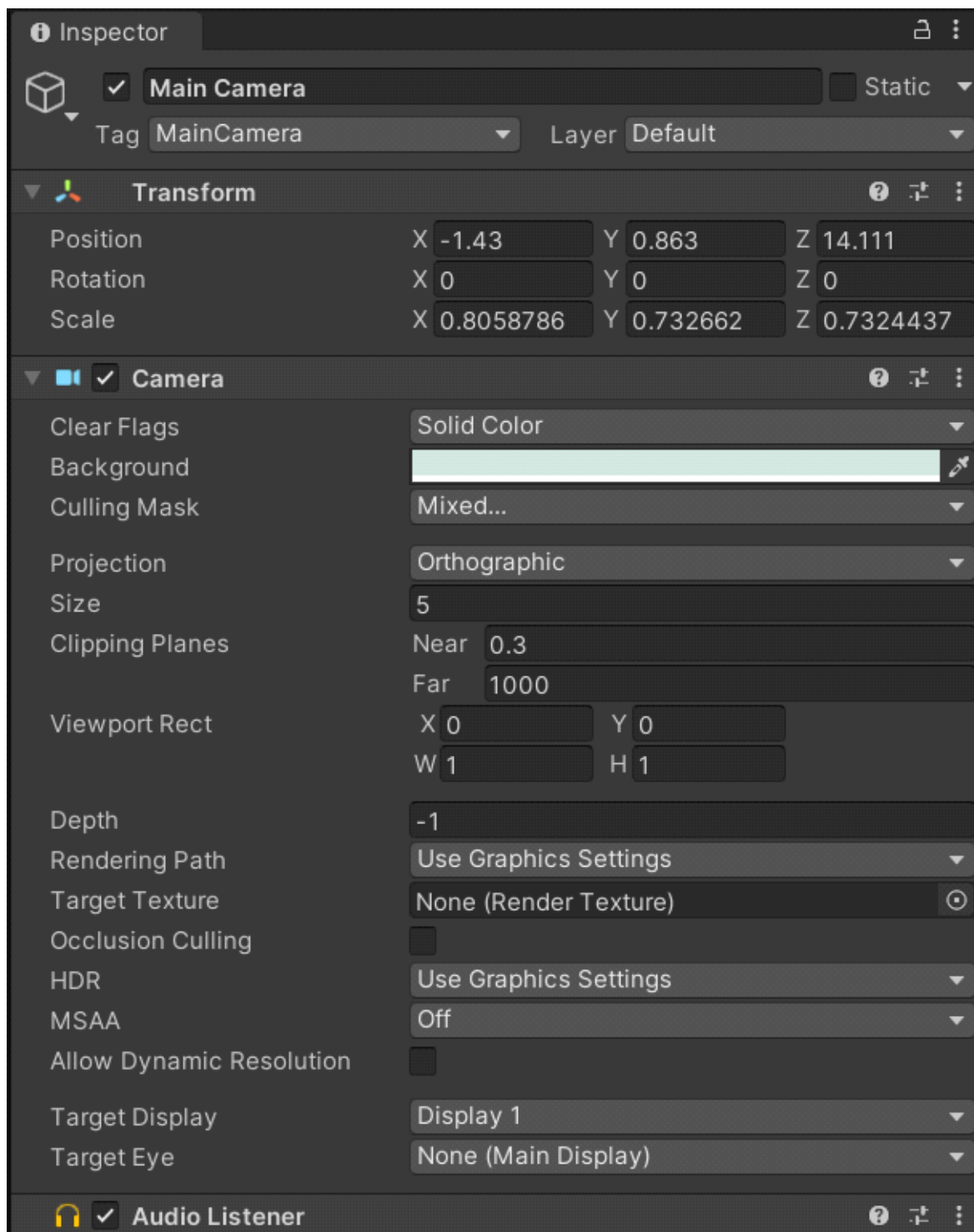


Рисунок 4.3 – Налаткування камери у проєкті

Щоб забезпечити належне функціонування всіх активних елементів тренажеру, необхідно включити в проект об'єкт Canvas. Цей об'єкт є спеціальним "полотном", на якому розташовуються та виконуються всі необхідні функції.

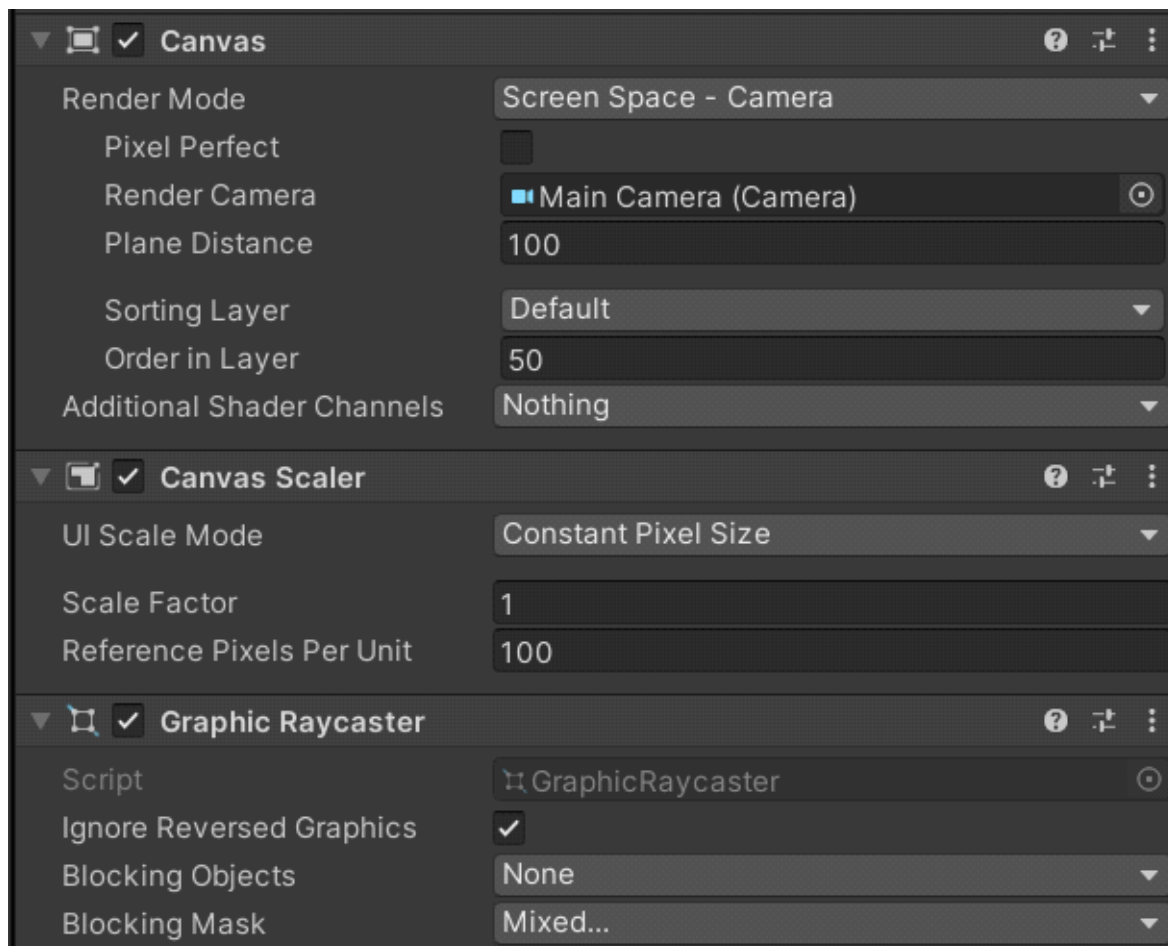


Рисунок 4.4 – Налаштування елементів Canvas

Робота з текстовими елементами в Unity подібна до взаємодії зі звичайними текстовими редакторами, проте має свої переваги, такі як зручне масштабування тексту для різних діагоналей.

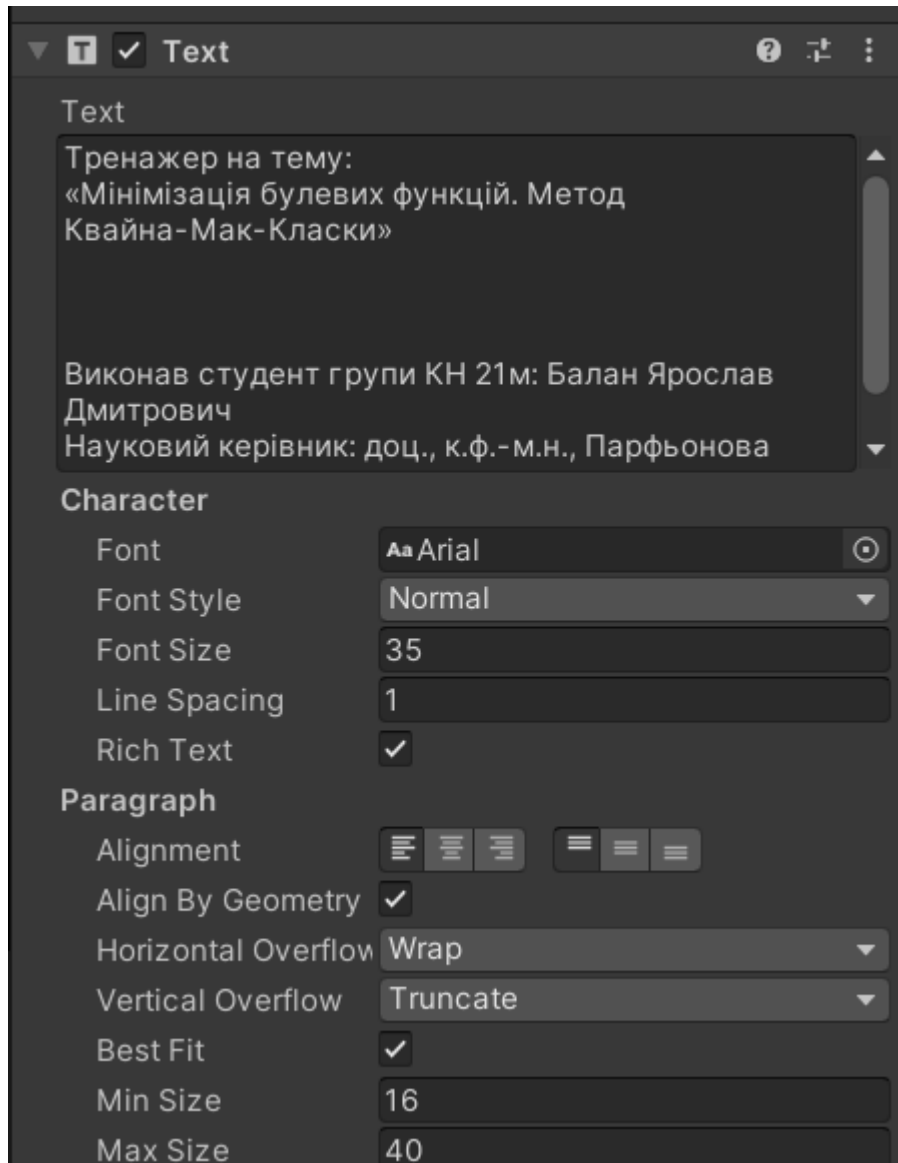


Рисунок 4.5 – Налаштування елемента Текст

Для переміщення між різними слайдами та виконання тестових завдань у тренажері використовується елемент управління, а саме - "кнопка". Налаштування цієї кнопки в початковому меню тренажеру виглядає наступним чином:

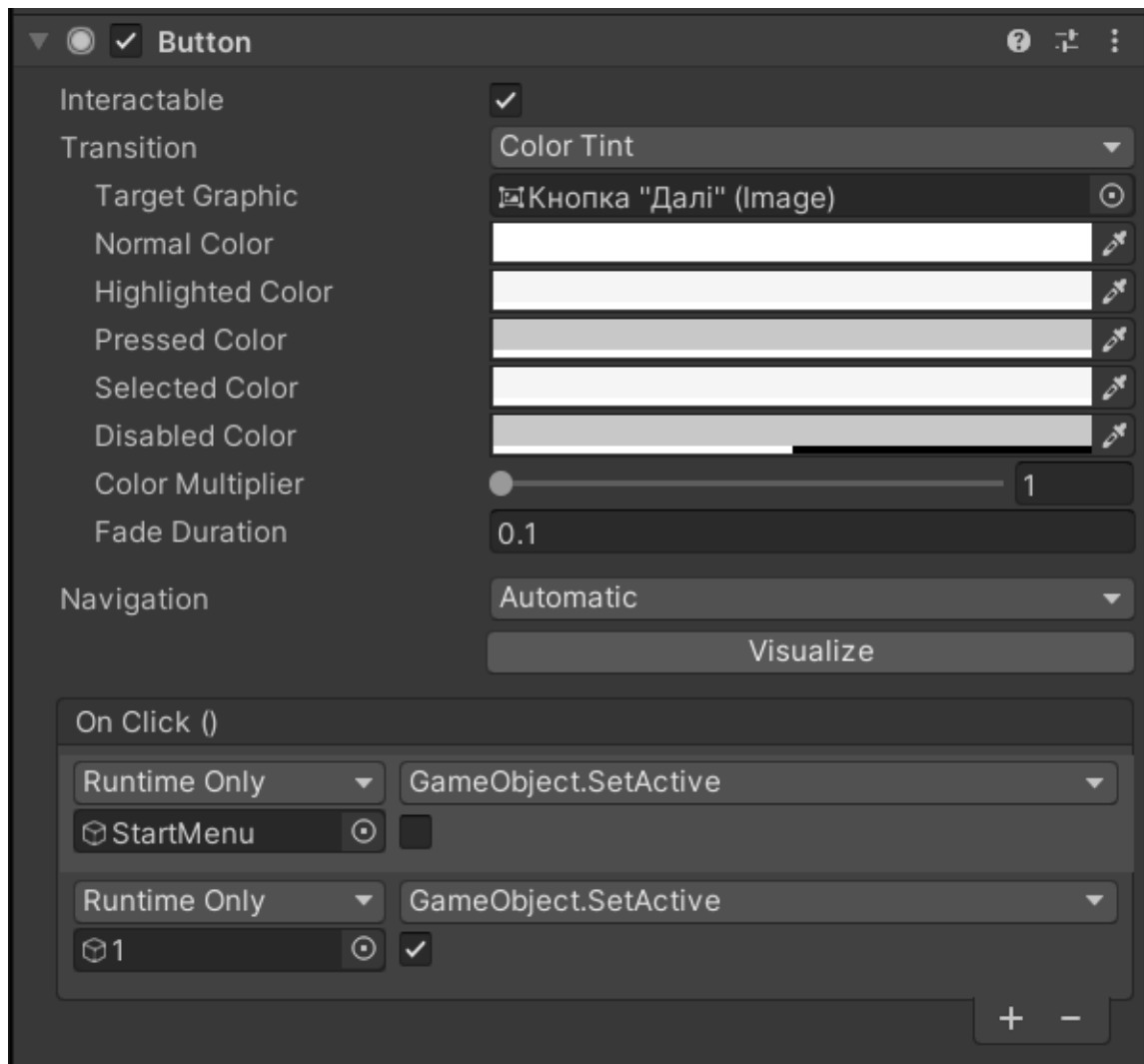


Рисунок 4.6 – Налаштування елементу Кнопка

Кожну кнопку можна оснастити текстовою інформацією, яка не взаємодіє із сценаріями та може змінюватися динамічно при необхідності. Наприклад, текст, пов'язаний із кнопкою "Старт", має такий вигляд:

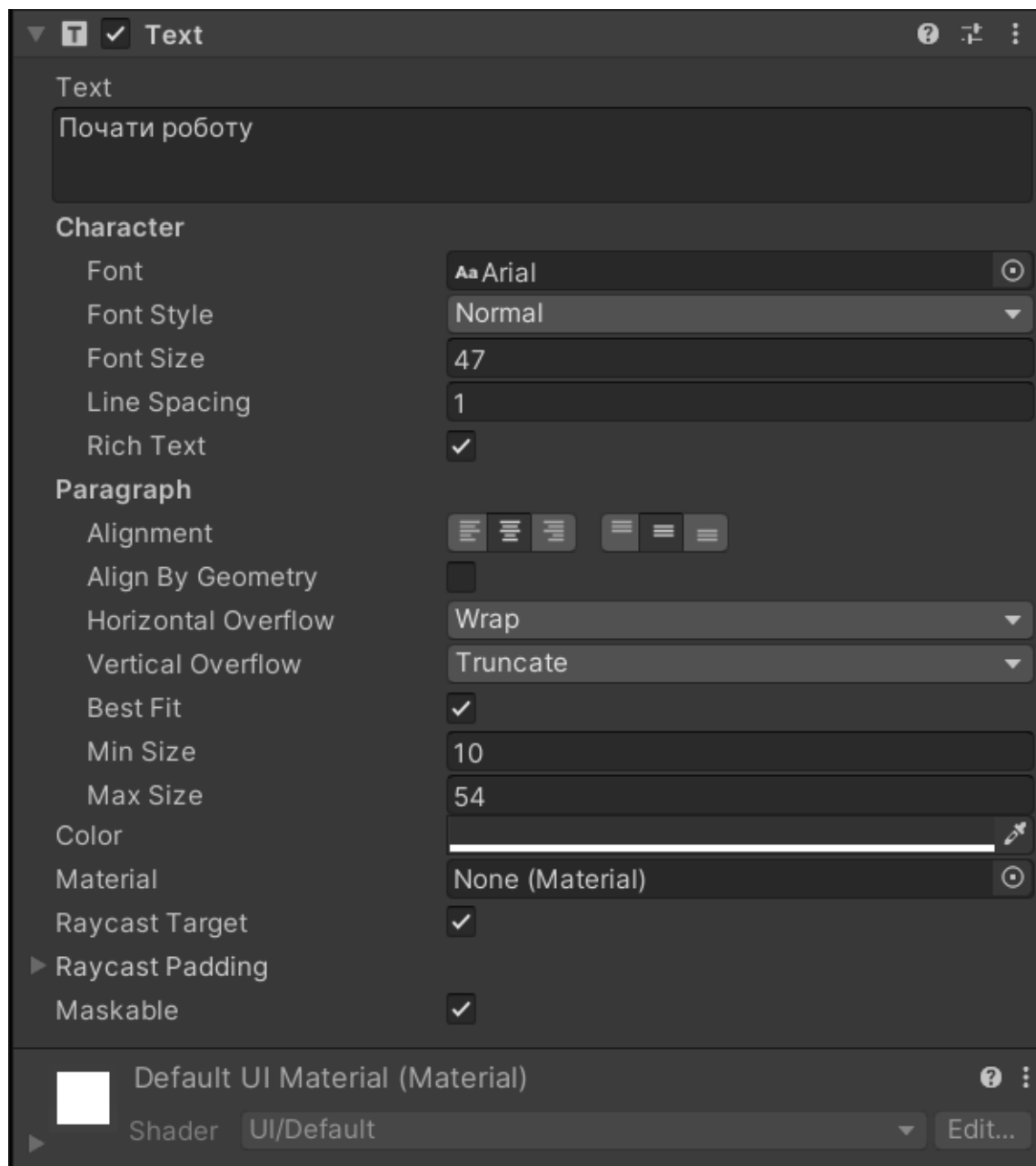


Рисунок 4.7 – Налаштування тексту в кнопці «Почати роботу»

4.2 Інструкція до роботи з тренажером

Після запуску додатку користувач має можливість ознайомитися з тематикою тренажера та інформацією про його розробника, а також розпочати використання за допомогою відповідної кнопки.

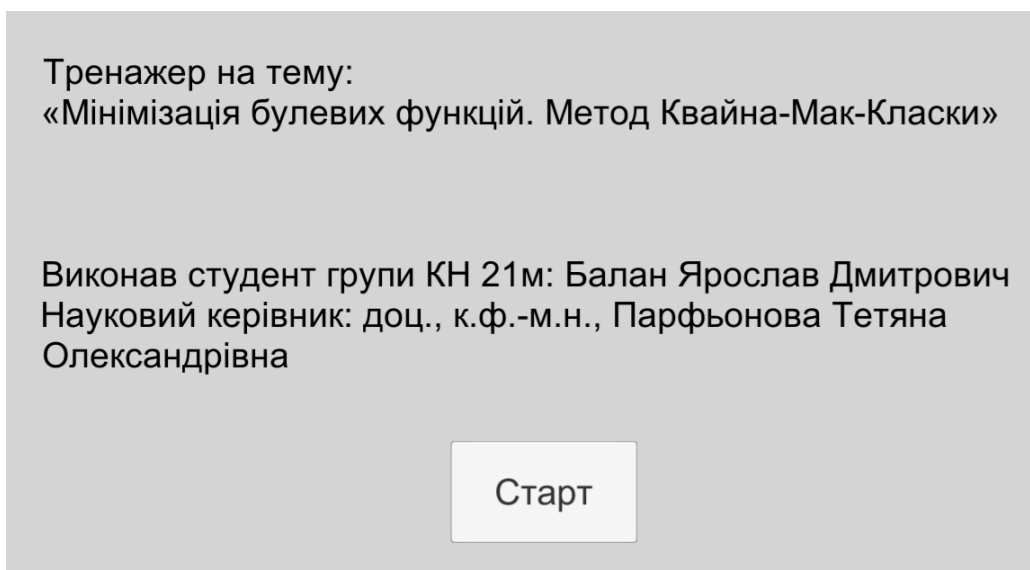


Рисунок 4.8 – Стартове меню

Практичне завдання має наступний вигляд. Для продовження виконання завдання необхідно вибрати правильний варіант відповіді.

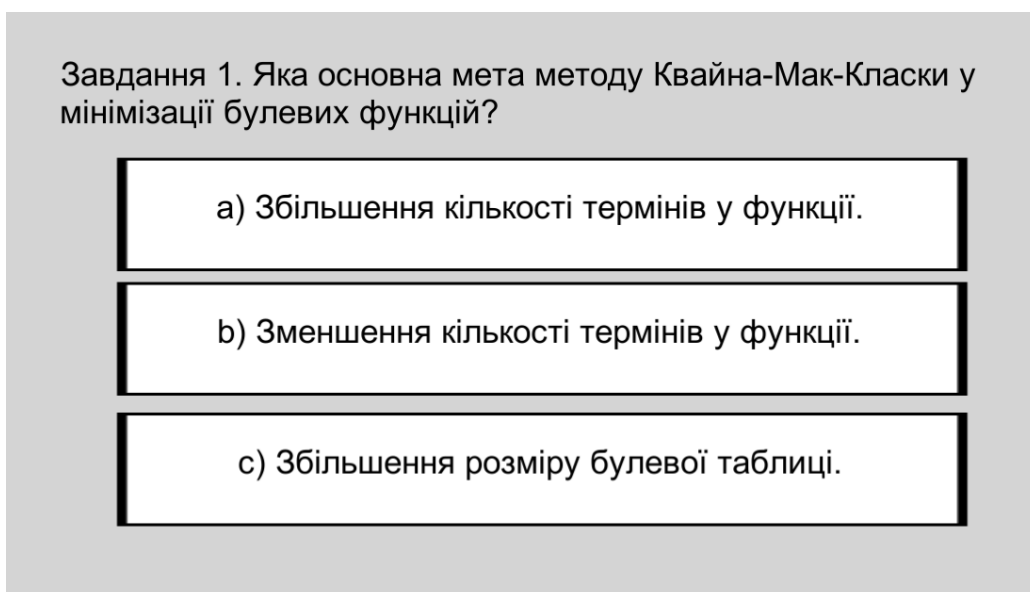


Рисунок 4.9 – Вигляд завдання до вибору відповіді.

Після вибору невірного варіанту відповіді, з'являється повідомлення "Неправильно!"

Завдання 1. Яка основна мета методу Квайна-Мак-Класки у мінімізації булевих функцій?

а) Збільшення кількості термінів у функції.

б) Зменшення кількості термінів у функції.

в) Збільшення розміру булевої таблиці.

НЕВІРНО

Рисунок 4.13 – Неправильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".

Завдання 1. Яка основна мета методу Квайна-Мак-Класки у мінімізації булевих функцій?

а) Збільшення кількості термінів у функції.

б) Зменшення кількості термінів у функції.

в) Збільшення розміру булевої таблиці.

ВІРНО

Далі

Рисунок 4.14 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".

Завдання 2. Які з перерахованих етапів не входять у процес мінімізації методом Квайна-Мак-Класки?

- а) Визначення множин ваг булевої функції.
- б) Побудова вагованого графа Квайна.
- в) Знаходження локальних максимумів ваг функції.

ВІРНО Далі

Рисунок 4.15 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".

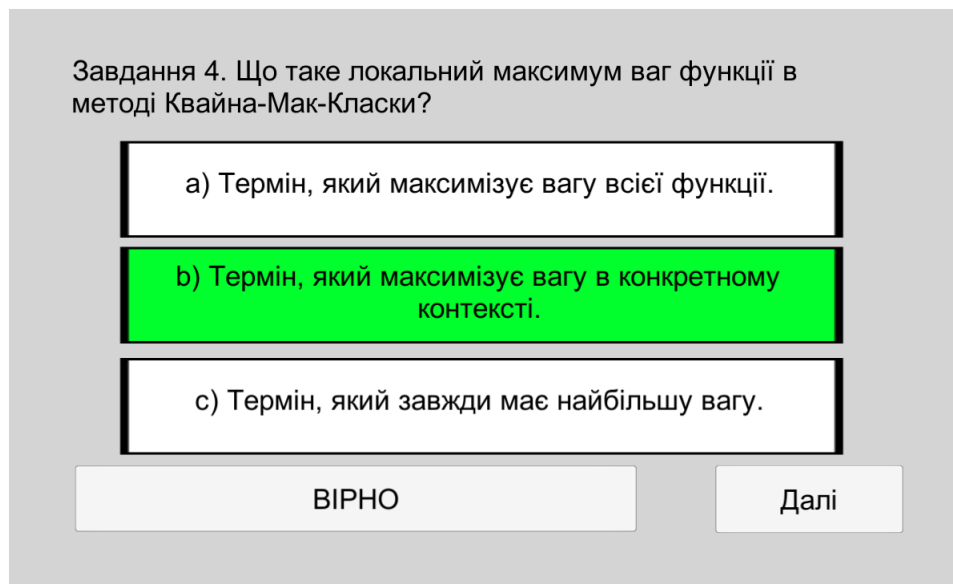
Завдання 3. Що знаходиться на ребрах графа Квайна?

- а) Терміни булевої функції.
- б) Множини ваг.
- в) Глобальні максимуми ваг функції.

ВІРНО Далі

Рисунок 4.16 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".



Завдання 4. Що таке локальний максимум ваги функції в методі Квайна-Мак-Класки?

- a) Термін, який максимізує вагу всієї функції.
- b) Термін, який максимізує вагу в конкретному контексті.**
- c) Термін, який завжди має найбільшу вагу.

ВІРНО Далі

Рисунок 4.17 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".

Завдання 5. Як визначається критерій закінчення процесу мінімізації методом Квайна-Мак-Класки?

- а) Зупинка, коли всі терміни функції стають глобальними максимумами.
- б) Зупинка, коли ваги не змінюються протягом певного кількості ітерацій.**
- с) Зупинка, коли функція стає константою.

ВІРНО Далі

Рисунок 4.18 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".

Завдання 6. Які переваги має метод Квайна-Мак-Класки порівняно з іншими методами мінімізації булевих функцій?

- а) Зменшення кількості використовуваних термінів.
- б) Зручність у визначенні ваг та використання графічного представлення.**
- с) Відсутність обмежень на розмір булевої функції.

ВІРНО Далі

Рисунок 4.19 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".

Завдання 7. Визначте мінімальну кількість термінів, яку може мати мінімізована булева функція.

a) 1

b) 2

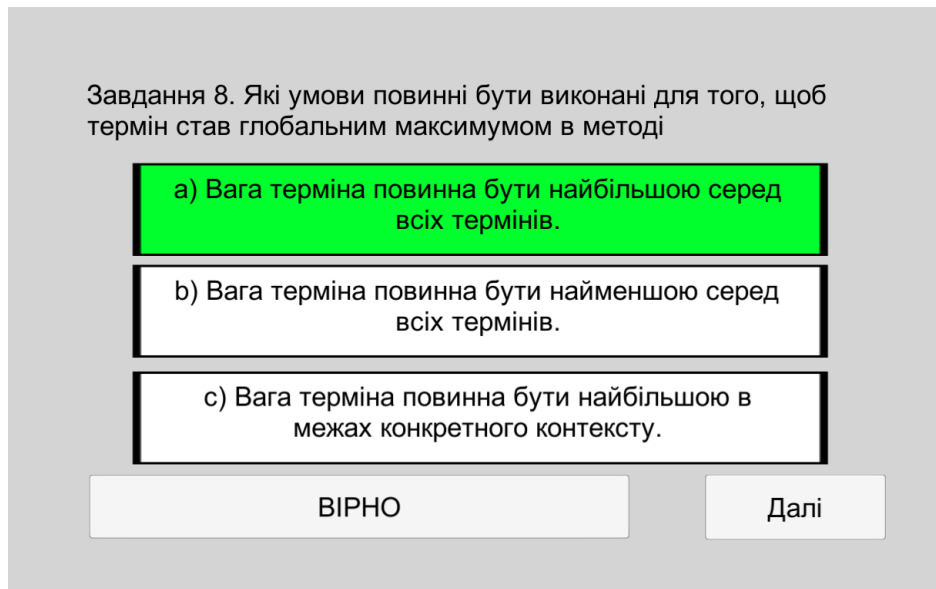
c) 0

ВІРНО

Далі

Рисунок 4.20 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".



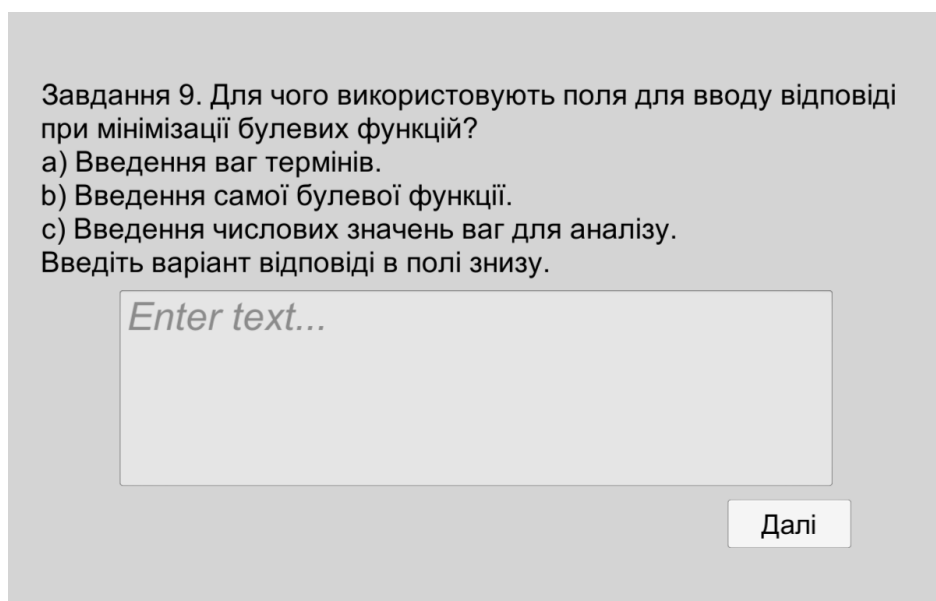
Завдання 8. Які умови повинні бути виконані для того, щоб термін став глобальним максимумом в методі

- a) Вага терміна повинна бути найбільшою серед всіх термінів.
- b) Вага терміна повинна бути найменшою серед всіх термінів.
- c) Вага терміна повинна бути найбільшою в межах конкретного контексту.

ВІРНО Далі

Рисунок 4.21 – Правильний варіант відповіді

Після обрання правильного варіанту відповіді, виводиться повідомлення "Правильно!" і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі". Наступне завдання передбачає введення відповіді через відповідне поле для вводу.



Завдання 9. Для чого використовують поля для вводу відповіді при мінімізації булевих функцій?

- a) Введення ваг термінів.
- b) Введення самої булевої функції.
- c) Введення числових значень ваг для аналізу.

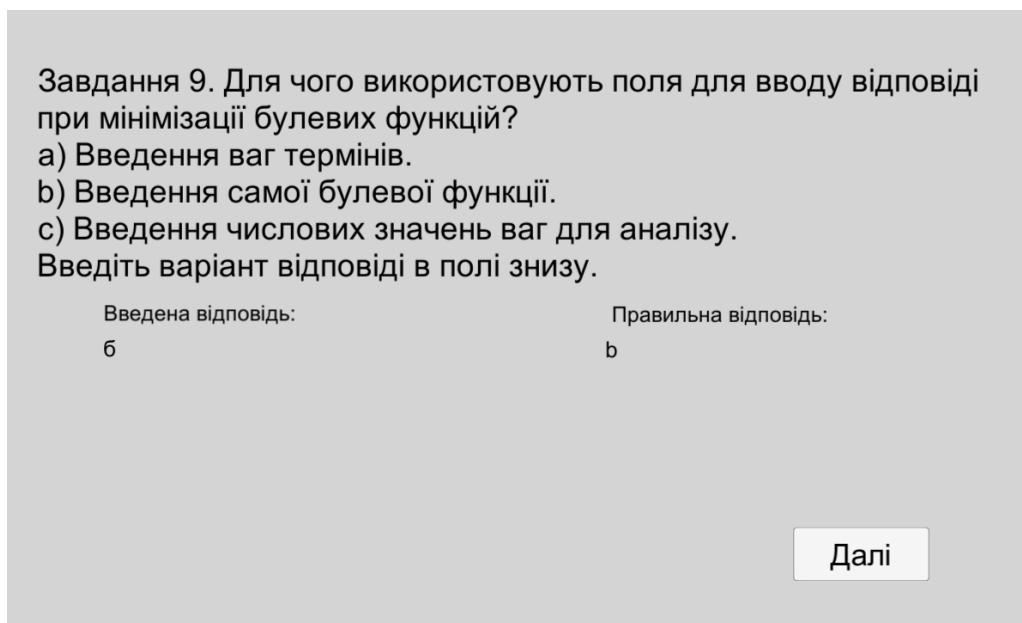
Введіть варіант відповіді в полі знизу.

Enter text...

Далі

Рисунок 4.22 – Завдання з полем для вводу

Після введення правильного варіанту відповіді, виводиться введена та правильна відповідь і з'являється опція продовжити використання тренажеру за допомогою кнопки "Далі".



Завдання 9. Для чого використовують поля для вводу відповіді при мінімізації булевих функцій?

- а) Введення ваг термінів.
- б) Введення самої булевої функції.
- с) Введення числових значень ваг для аналізу.

Введіть варіант відповіді в полі знизу.

Введена відповідь:	Правильна відповідь:
б	б

Рисунок 4.23 – Правильний варіант відповіді

Після завершення взаємодії з навчальними ресурсами тренажеру, користувач переходить до заключного екрану, де відображається повідомлення про завершення роботи та доступна кнопка "Повторити".

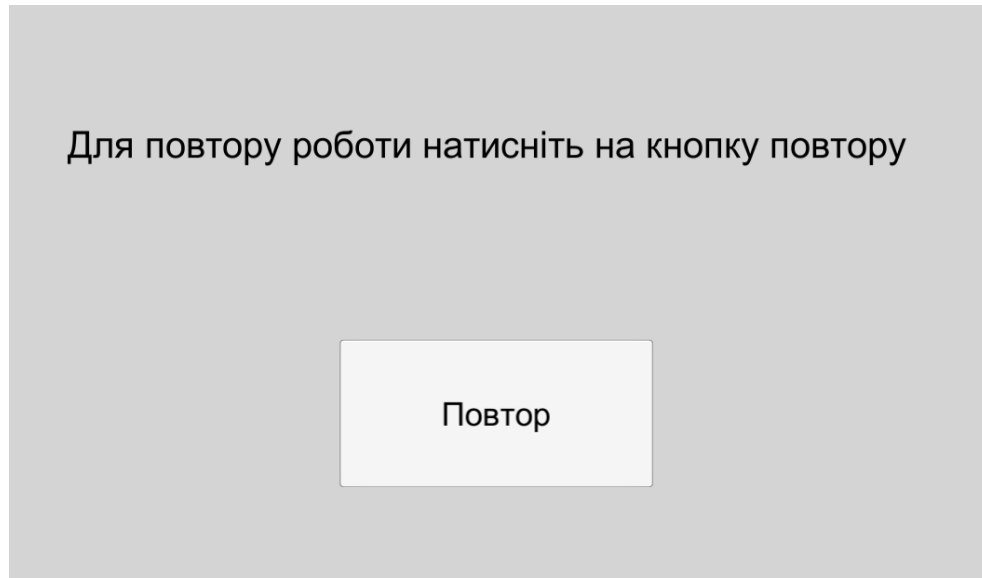


Рисунок 4.24 – Повідомлення, що робота завершена, та опція "Повторити"

ВИСНОВОК

Магістерську роботу успішно завершено, і в результаті розроблено повноцінний програмний симулятор для студентів, спрямований на вивчення теми " Мінімізація булевих функцій. Метод Квайна-Мак-Класки ". Розгляньмо найважливіші аспекти проекту:

Алгоритм роботи: Розроблено та імплементовано ефективний алгоритм роботи тренажеру, що дозволяє студентам систематично освоювати тему.

Завдання: Розроблено різноманітні практичні та теоретичні завдання для глибшого розуміння матеріалу.

Блок-схеми: Створено докладні блок-схеми, що наочно представляють всі етапи роботи тренажеру.

Мова програмування та реалізація: Обрано мову програмування C# і успішно створено програму-тренажер у середовищі Unity 2021.

Інтерфейс: Розроблено зручний та інтуїтивно зрозумілий інтерфейс для полегшення навігації користувачів.

Тестування: Проведено широке тестування тренажеру, включаючи перевірку на валідність та виявлення та усунення всіх можливих помилок.

Інструкції та довідка: Надано детальні інструкції щодо використання тренажеру та довідкову інформацію про правильність відповідей.

Повідомлення про завершення: При успішному завершенні тренажер повідомляє користувача та надає опцію перегляду результатів та повторення тестування.

Налаштування під конкретний комп'ютер: Забезпечено оптимальні налаштування, що найкраще відповідають характеристикам конкретного комп'ютера користувача.

Цей тренажер стане невід'ємним інструментом для студентів, допомагаючи їм систематично вивчати та освоювати складну тему системного аналізу. Після успішного завершення роботи з тренажером, користувач отримає повідомлення про завершення, після чого може переглянути результати та вибрати опцію повторення

тестування. Тренажер стартує з налаштуваннями, що найкраще відповідають конкретному комп'ютеру користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мова програмування C# [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – Режим доступу: <http://uk.wikipedia.org/wiki/C#>.
2. Квайн, У. К. Методи дискретної математики: Навчальний посібник / У. К. Квайн, Г. Ш. Мак-Класки. - К.: Літера, 2009. - 336 с.
3. Величко, О.М. Опрацювання інформаційного потоку взаємодією елементів друкарського контакту: Монографія. – К.: ВПЦ «Київський університет», 2005. – 264 с.
4. Ковальчук, В. С. Дискретна математика: Підручник / В. С. Ковальчук. - К.: Літера, 2017. - 480 с.
5. Кравець, С. О. Основи дискретної математики: Підручник / С. О. Кравець. - К.: Центр учбової літератури, 2003. - 448 с.
6. Шевченко, В. Дискретна математика для інформаційних технологій: Підручник / В. Шевченко. - К.: БІНОМ, 2018. - 432 с.
7. Гринченко, В. В. Математичні основи інформаційних технологій: Підручник / В. В. Гринченко. - К.: Літера, 2008. - 384 с.
8. Мак-Класки, Г. Ш. Лекції по дискретній математиці / Г. Ш. Мак-Класки. - К.: Київський університет, 2002. - 392 с.
9. Яковлев, С. А. Курс дискретної математики / С. А. Яковлев, С. А. Степанов, К. К. Яковлев. - К.: Літера, 2005. - 448 с.
10. Левітін, А. О. Дискретна математика: Вступ у моделювання та алгоритми / А. О. Левітін. - К.: Видавництво "Стиль", 2003. - 496 с.
11. Мірошніченко, В. П. Дискретна математика для інформатиків: Підручник / В. П. Мірошніченко. - К.: Літера, 2014. - 424 с.

ДОДАТОК А