

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

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту
Завідувач кафедри _____ О.О. Ємець
(підпис)
« _____ » _____ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

на тему

**ТРЕНАЖЕР З ТЕМИ «АЛГОРИТМ ПОБУДОВИ ФУНКЦІЇ FOLLOW
ДЛЯ СИМВОЛІВ ГРАМАТИКИ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО
КУРСУ «ТЕОРІЯ ПРОГРАМУВАННЯ» ТА РОЗРОБКА ЙОГО
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Флерчук Вадим Олександрович

_____ « _____ » _____ 2021р.
(підпис)

Науковий керівник доц., каф.-мат. наук, Черненко О.О.

_____ « _____ » _____ 2021р.
(підпис)

ПОЛТАВА 2021р.

РЕФЕРАТ

Записка: 43 стор., в т.ч. основна частина 41 стор., джерел – 10.

Мета бакалаврської роботи – програмування елементів тренажеру «Алгоритм побудови функції FOLLOW для символів граматики» дистанційного навчального курсу «Теорія програмування».

Об'єктом розробки є процес навчання математичним дисциплінам.

Предмет розробки — алгоритм роботи та програмна реалізація тренажеру з теми «Алгоритм побудови функції FOLLOW для символів граматики».

Методи, які були використані для розв'язування задачі – програмна реалізація була здійснена в середовищі розробки IDE Visual Studio та за допомогою платформи Unity.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	6
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
2.1. Типи комп'ютерних тренажерів.....	7
2.2. Актуальність теми роботи.....	10
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА	12
3.1. Огляд матеріалу з теми.....	12
3.2. Алгоритм робити тренажеру.....	16
3.3 Блок-схема алгоритму тренажера.....	24
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА	24
4.1 Опис створення тренажеру	26
4.2 Інструкція для роботи з тренажером.....	29
ВИСНОВКИ.....	41
СПИСОК ЛІТЕРАТУРИ.....	42
ДОДАТОК А. ДИСК З МАТЕРІАЛАМИ	43

ВСТУП

Сучасна система освіти зазнає бурхливий розвиток. Різноманітність і збільшення числа електронних освітніх ресурсів дає педагогу можливість використовувати їх в освітньому процесі. В системі дистанційної освіти для відпрацювання обчислювальних навичок, навичок вирішення типових задач, для самостійного навчання актуальним і ефективним є використання інтерактивних тренажерів.

Використання інтерактивних тренажерів в системі дистанційної освіти:

- доступно для кожного учасника навчального процесу;
- дозволяє ефективно формувати практичні навички учнів;
- підвищує мотивацію до навчання;
- сприяє наочності освітнього процесу;
- підвищує вагу самостійної роботи учнів;
- дозволяє реалізувати диференційований підхід до учнів з різним рівнем готовності до навчання;
- оптимізує навчальний час.

В умовах підключення до мережі Internet кожного учасника навчального процесу в системі дистанційної освіти стає доступним використання online - тренажерів. Можна працювати з таким тренажером і на заняттях, і вдома самостійно, вибравши зручний час.

Мета бакалаврської роботи – програмування елементів тренажеру «Алгоритм побудови функції FOLLOW для символів граматики» дистанційного навчального курсу «Теорія програмування».

Об'єктом розробки є процес навчання математичним дисциплінам.

Предмет розробки — алгоритм роботи та програмна реалізація тренажеру з теми «Алгоритм побудови функції FOLLOW для символів граматики».

Перелік використаних методів полягає в застосуванні побудови функції FOLLOW для символів граматики.

Бакалаврська робота складається з трьох розділів. У першому розділі розглянуто постановку задачі. У другому розділі розглянуто типи комп'ютерних тренажерів, актуальність теми. У третьому розділі представлено огляд матеріалу з теми, алгоритм робити тренажеру. В четвертому розділі описаний процес програмної реалізації тренажеру.

У додатку міститься диск з матеріалами проекту.

Обсяг пояснювальної записки: 43 стор., в т.ч. основна частина - 41 стор., додатки - 1 стор., джерела - 10 назв.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Для досягнення поставленої мети необхідно було вирішити такі завдання:

1. Визначити сучасні вимоги до тренажерів;
2. Розробити концепції функціонування та структуру тренажеру;
3. Запропонувати математичну модель;
4. Виявити вимоги до інструментальних засобів розробки програми на основі створеної моделі;
5. Реалізувати необхідні для програмного втілення моделі інструментальні засоби і компоненти програми;
6. Розробити програмний продукт.

При програмній реалізації пропонується використовувати:

- методи теорії алгоритмів і мов програмування;
- методи об'єктно-орієнтованого програмування;
- методи математичного моделювання та візуалізації.

В роботі слід описати такі пункти:

1. Навести актуальність теми роботи;
2. Розглянути можливі типи тренажерів;
3. Розробити алгоритм роботи тренажеру;
4. Створити блок-схему тренажеру;
5. Описати причини використання якого саме програмного забезпечення при реалізації теми роботи;
6. Реалізувати програму;
7. Описати роботу програми;
8. Вказати процес створення тренажеру.

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Типи комп'ютерних тренажерів

Інтернет тренажери в сфері освіти це програмний комплекс, в основі якого лежить оригінальна методика оцінювання знань, умінь і навичок, спрямованих на багаторазове рішення тестових завдань.

Інтерактивний тренажер призначений для вивчення (або повторення) з одночасним контролем знань з певної теми при самостійній підготовці. Є 2 види роботи тренажера:

1. Наглядно-демонстраційний - включає в себе дидактичний матеріал, який складається з теоретичного і наочно-аналітичного рішення задач.

2. Тренувально-оцінювальний. Задання в таких тренажерах повинні мати зворотний зв'язок, можливість коригування своїх дій. До них можна віднести послідовність питань, підказки та практичні завдання. Інтерактивні тренажери можливо застосовувати на будь-якому етапі. Їх можуть розробляти викладачі самостійно, застосовуючи при цьому програми MicrosoftPowerPoint і MicrosoftExcel [2].

Також можна переглянути інші якості тренажерів.

1. Електронний екзаменатор.

Це найпростіший програмний продукт, що реалізується на всіх видах вітчизняної і зарубіжної обчислювальної техніки. Основна його функція - це заміна живого екзаменатора в строго регламентованих областях (техніка безпеки різних виробництв, правила дорожнього руху і т.п.). Як правило, такий екзаменатор містить набір квитків з декількох питань, пропонованих екзаменованих у випадковому порядку, і ряду неправильних і однієї правильної відповіді на кожне питання. Залежно від складності екзаменатора, він може забезпечувати також наступні можливості:

- показ малюнків в кадрі питання;
- показ мультфільмів (анімація) в кадрі питання;

- аналіз відповіді екзаменованих у вигляді чисел і формул;
- попереднє навчання (показ правильних відповідей);
- редагування старих і створення нових питань.

2. Статичні (або логіко-динамічні) тренажери.

Найбільш широко в цьому класі поширені тренажери по оперативних перемиканням в електричних мережах. Основна особливість полягає в тому, що в таких програмах відсутня фізико-математична модель процесів, що відбуваються в обладнанні, але показується і перевіряється певний порядок дій. Порядок дій зазвичай жорстко задається; в більш складних випадках передбачаються розгалуження в ланцюжку дій, що забезпечується логічними функціями (логіко-динамічна модель).

Головні недоліки:

- неможливість відхилення учня від як завгодно складного, але все одно жорстко заданого ланцюжка дій;
- трудність програмування динамічних ефектів (навіть простої зміни показань приладів).

Ці недоліки несуттєві в жорстко регламентованих оперативних перемиканням в електричних мережах, але визначають неможливість моделювання складних фізичних процесів в котлах і турбінах. Як правило, розробники таких тренажерів спочатку роблять для себе інструментальні засоби (конструктори), дозволяють легко намалювати потрібні електричні схеми і задати ланцюжки правильних дій.

3. Динамічні тренажери.

Мають у своїй основі математичну модель реальних фізичних процесів і тому найбільш корисні для якісного навчання (НЕ натаскування!) персоналу.

Динамічний комп'ютерний тренажер дозволяє:

- провести комп'ютерний експеримент для дослідження різних режимів обладнання, а також для аналізу аварійних ситуацій;

- наочно показати фізичну сутність протікають в обладнанні процесів, їх взаємну залежність, а також ряд істотних тонкощів, яким, на жаль, не завжди надається значення на практиці;
- сформувати і постійно підтримувати кваліфікацію і готовність персоналу, навички дій в складних ситуаціях, в тому числі шляхом регулярних протиаварійних тренувань.

4. Сучасна комп'ютерна технологія (мультимедіа) дозволяє створювати діалогові навчальні програми і тренажери, які включають комп'ютерну мультиплікацію, аудіо та відеотехніку. Як мінімум, це посилить відчуття реальності при роботі з тренажером і відкриє нові можливості в процесі навчання [3].

Наведена класифікація допоможе при виборі програмних засобів в залежності від завдань і цілей навчання:

- для контролю персоналу по набору певних правил (техніка безпеки і т.п.) можна використовувати прості екзаменатори;
- для загального ознайомлення з пристроєм і навчання певному порядку дій можна використовувати статичні і логіко-динамічні тренажери;
- для проведення експериментів, вивчення фізичних основ і способів функціонування пристроїв, для проблемного навчання, протиаварійних тренувань і аналізу аварій слід використовувати динамічні тренажери.

Отже, завдяки доступності засобів створення тренажерів з великим вибором програмних комплексів для навчальних закладів, тренажери займають в сучасній освіті одне з найважливіших місць у формуванні та закріпленні знань, умінь і навичок учнів. Доцільність застосування тренажерів в освітньому процесі установ не викличе сумнівів і виправдає свої педагогічні цілі.

2.2. Актуальність теми роботи

За деякими оцінками, свого розквіту інформаційна цивілізація досягне до середини XXI століття, і вже наші діти будуть жити і працювати в абсолютно новому інформаційному середовищі проживання. Тому сучасна система освіти повинна випереджаюче готувати нове покоління до умов існування і професійної діяльності в глобальному інформаційному суспільстві.

Зазначені об'єктивні чинники і прогнози обумовлюють повсюдне залучення інформаційних і телекомунікаційних технологій для надання освітніх послуг з метою глобального розширення навчається аудиторії. Взаємодія, інтеграція і стандартизація цих послуг призводить до появи середовища відкритої освіти.

Таким чином, одним з основних засобів є дистанційне навчання (ДН), визначене в роботі як «... сукупність технологій, що дозволяють реалізувати освітній процес з віддаленим користувачем». Найголовніше завдання дистанційного навчання - не знижуючи високої якості професійної підготовки, зробити її справді масовою. Однак факти говорять про те, що високої якості професійної підготовки за допомогою засобів ДН вдається досягти аж ніяк не завжди.

Таким чином, актуальність дослідження обумовлена:

1. Бурхливим розвитком, яке переживає сфера електронної освіти в даний час;
2. Відсутністю розвинених стандартів і вимог до комп'ютерних тренажерних програмами;
3. Відсутність формалізованого опису (математичної моделі) комп'ютерних тренажерів в достатній для програмної реалізації ступеня;
4. Великим обсягом наявної пасивної інформації (паперових підручників з математичних дисциплін), яку необхідно перевести в активну, електронну, форму;

5. Наявністю комп'ютерних тренажерних програм в освітньому процесі в кількості, недостатній для сучасного рівня розвитку інформаційних технологій в цілому;

6. Високими вимогами, що пред'являються до готових комп'ютерних тренажерних програм;

7. Трудністю створення без допоміжних засобів навчальних програм, що задовольняють висунутим суспільством вимогам;

8. Необхідністю створення практичної та теоретичної бази, яка б забезпечила можливість створення засобів автоматизованої розробки комп'ютерних тренажерів.

Із застосуванням комп'ютерних систем контроль якості знань учнів відходить від того, щоб бути узагальненим, знаходить характеристики індивідуального підходу, який враховує не тільки знання, але індивідуальний темп навчання учнів. Крім того, автоматизований контроль підвищує об'єктивність самого контролю, дозволяє оцінювати якість знань учнів не тільки поверхнево, але забезпечує кількісну оцінку якості засвоєння учнями того чи іншого розділу навчального розділу. Ще одним важливим аспектом застосування автоматизованого контролю, стає стимулювання і мотивування до самоосвітньої діяльності. Досягнення найкращих результатів у засвоєнні знань з використанням автоматизованих систем навчання і контролю забезпечується грамотним і педагогічно вірним підходом до реалізації таких систем. На кожному етапі потрібно виявляти ефективні способи задоволення основним педагогічним вимогам:

- дидактичним;
- методичним;
- обґрунтуванню вибору тематики навчального курсу;
- перевірці на педагогічну доцільність використання та ефективність застосування.

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Огляд матеріалу з теми

При побудові предиктивного аналізатора корисними виявляються дві функції, зв'язані з граматикою G . Ці функції, FIRST і FOLLOW, дозволяють побудувати таблицю предиктивного розбору для G , якщо, звичайно, це можливо. Множини, що породжуються цими функціями, можуть, крім того, бути використані при відновленні після помилок.

Нехай α – довільний рядок символів граматики. **Функція FIRST(α)** - це множина терміналів, з яких починаються рядки, виведені з α . Якщо $\alpha \Rightarrow^* \varepsilon$, то ε також належить FIRST(α).

Функція FOLLOW(A) для нетерміналу A - це множина терміналів a , що можуть з'явитися безпосередньо праворуч від A у деякій сентенціальній формі, тобто множина терміналів a таких, що існує породження вигляду $S\alpha \Rightarrow Aa\beta$ для деяких α і β . Відзначимо, що між A і a у процесі виведення можуть з'явитися нетермінальні символи, з яких виводиться ε . Якщо A може виявитися крайнім правим символом деякої сентенціальної форми, то $\$$ належить FOLLOW(A).

Для обчислення FOLLOW(A) для нетерміналу A застосуємо такий алгоритм.

Алгоритм. Побудова FOLLOW(X) для всіх X -нетерміналів граматики.

Крок 1 Помістити $\$$ у FOLLOW(S), де S – початковий символ і $\$$ – правий кінцевий маркер.

Крок 2 Якщо є продукція $A \rightarrow \alpha B \beta$, то усі елементи з множини FIRST(β), за винятком ε , додати до FOLLOW(B).

Крок 3 Якщо є продукція $A \rightarrow \alpha B$ або $A \rightarrow \alpha B \beta$, де FIRST(β) містить ε (тобто $\beta \Rightarrow^* \varepsilon$), то усі елементи з множини FOLLOW(A) додати до FOLLOW(B).

Приклад.

Розглянемо, як будуть будуватися FIRST і FOLLOW на прикладі граматики арифметичних виразів. Обмежимося лише операціями додавання, множення і наявністю дужок. Числа будемо позначати однією літерою n для простоти. Інтуїтивна граматика для арифметичних виразів виглядає наступним чином:

$$E \rightarrow E + E \mid E \times E \mid (E) \mid n$$

Однак дана граматика містить ліву рекурсію, праве розгалуження і є неоднозначною. Щоб позбутися від цих проблем неявно, можна придумати більш вдалу граматику для розглянутої мови. Наприклад, вона може мати наступний вигляд:

$$E \rightarrow T + E \mid TT \rightarrow F \times T \mid FF \rightarrow n \mid (E)$$

Дана граматика містить тільки праве розгалуження, від якого можна позбутися за допомогою лівої факторизації, після чого граматика набуде вигляду:

$$E \rightarrow TE'$$

$$E' \rightarrow +E \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow \times T \mid \varepsilon$$

$$F \rightarrow n \mid (E)$$

А потім для простоти аналізування розкрити нетермінали E і T в правилах для E і T .

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow \times FT' \mid \varepsilon$$

$$F \rightarrow n \mid (E)$$

Для неї:

$$\text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{ (, \text{id} \};$$

$$\text{FIRST}(E') = \{ +, \varepsilon \};$$

$$\text{FIRST}(T') = \{ *, \varepsilon \};$$

$$\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{), \$\};$$

$$\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{+,), \$\};$$

$$\text{FOLLOW}(F) = \{+, *,), \$\}.$$

На кроці 1 обчислення множини FOLLOW у $\text{FOLLOW}(E)$ включаємо \$.

На кроці 2, використовуючи продукцію $F \rightarrow (E)$, до $\text{FOLLOW}(E)$ додається також права дужка.

На кроці 3, застосованому до правила $E \rightarrow TE'$, у $\text{FOLLOW}(E')$ включаються \$ і права дужка.

Оскільки $E' \Rightarrow \varepsilon$, вони також потрапляють у $\text{FOLLOW}(T)$. Відповідно до продукції $E \rightarrow TE'$ на кроці 2 у $\text{FOLLOW}(T)$ включається усе з $\text{FIRST}(E')$, відмінне від ε .

Покрокова ітерація:

До циклу while:

Таблиця 3.1 – До циклу while

Правило	FOLLOW
E	{ \$ }
E'	{ }
T	{ }
T'	{ }
F	{ }

Після 1-ої ітерації:

Таблиця 3.2 – Перша ітерація

Правило	FOLLOW
E	{ \$,) }
E'	{ \$ }
T	{ +,\$ }
T'	{ +,\$ }
F	{ *,+,\$ }

Після 2-ої ітерації:

Таблиця 3.3 – Друга ітерація

Правило	FOLLOW
E	$\{ \$,) \}$
E'	$\{ \$,) \}$
T	$\{ +, \$ \}$
T'	$\{ +, \$ \}$
F	$\{ *, +, \$ \}$

Після 3-ої ітерації:

Таблиця 3.4 – Третя ітерація

Правило	FOLLOW
E	$\{ \$,) \}$
E'	$\{ \$,) \}$
T	$\{ +, \$,) \}$
T'	$\{ +, \$,) \}$
F	$\{ *, +, \$,) \}$

Реалізація побудови FOLLOW виходить відразу з лем. Для алгоритму спочатку потрібно виконати побудову FIRST для граматики.

Приклад псевдокоду FOLLOW (рис. 3.1).

```
function constructFOLLOW():
  for ( $A \in N$ )
    FOLLOW[A] =  $\emptyset$ 
  FOLLOW[S] = { $\$$ } // в стартовый нетерминал помещается символ конца строки
  changed = true
  while changed
    changed = false
    for ( $A \rightarrow \alpha \in P$ )
      for ( $B : \alpha = \beta B \gamma$ )
        FOLLOW[B]  $\cup$  = FIRST( $\gamma$ )  $\setminus$  { $\epsilon$ }
        if  $\epsilon \in \text{FIRST}(\gamma)$ 
          FOLLOW[B]  $\cup$  = FOLLOW[A]
        changed = true if FOLLOW[B] изменился
```

Рисунок 3.1 – Приклад псевдокоду FOLLOW

Коректність цього алгоритму доводиться точно так же, як і коректність алгоритму конструювання FIRST.

3.2. Алгоритм робити тренажеру

Крок 1: Користувачу виводиться завдання, варіанти відповіді.

«Функція FOLLOW(A) для нетермінала A - це»

- це множина терміналів, з яких починаються рядки, виведені з α .
Якщо $\alpha \Rightarrow^* \varepsilon$, то ε також належить FIRST(α).
- множина терміналів a , що можуть з'явитися безпосередньо праворуч від A у деякій сентенціальній формі, тобто множина терміналів a таких, що існує породження вигляду $S\alpha \Rightarrow Aa\beta$ для деяких α і β .

Користувач вибирає один з варіантів. При правильній відповіді (2) виводиться наступний крок.

Крок 2: Користувачу виводиться завдання, варіанти відповіді.

«Відзначимо, що між A і a у процесі виведення можуть з'явитися нетермінальні символи, з яких виводиться»

- ε
- $\$$

Користувач вибирає один з варіантів. При правильній відповіді (1) виводиться наступний крок.

Крок 3: Користувачу виводиться завдання, варіанти відповіді.

«Якщо A може виявитися крайнім правим символом деякої сентенціальної форми, то $\$$ належить»

- FIRST (A)
- FOLLOW (A)

Користувач вибирає один з варіантів. При правильній відповіді (2) виводиться наступний крок.

Крок 4: Користувачу виводиться коротка довідка.

«Алгоритм. Побудова FOLLOW(X) для всіх X -нетерміналів граматики.

Крок 1 Помістити $\$$ у FOLLOW(S), де S – початковий символ і $\$$ – правий кінцевий маркер.

Крок 2 Якщо є продукція $A \rightarrow \alpha B \beta$, то усі елементи з множини FIRST(β), за винятком ε , додати до FOLLOW(B).

Крок 3 Якщо є продукція $A \rightarrow \alpha B$ або $A \rightarrow \alpha B \beta$, де FIRST(β) містить ε (тобто $\beta \xRightarrow{*} \varepsilon$), то усі елементи з множини FOLLOW(A) додати до FOLLOW(B).»

Виводиться наступний крок.

Крок 5: Користувачу виводиться умова завдання.

«Розглянемо, як будуть будуватися FIRST і FOLLOW на прикладі граматики арифметичних виразів. Обмежимося лише операціями додавання, множення і наявністю дужок. Числа будемо позначати однією літерою n для простоти. Інтуїтивна граматика для арифметичних виразів виглядає наступним чином:

$$E \rightarrow E + E \mid E \times E \mid (E) \mid n$$

Однак дана граматика містить ліву рекурсію, праве розгалуження і є неоднозначною. Щоб позбутися від цих проблем неявно, можна придумати більш вдалу граматичку для розглянутої мови. Наприклад, вона може мати наступний вигляд:

$$E \rightarrow T + E \mid TT \rightarrow F \times T \mid FF \rightarrow n \mid (E)$$

Дана граматика містить тільки праве розгалуження, від якого можна позбутися за допомогою лівої факторизації, після чого граматика набуде вигляду:

$$E \rightarrow TE' \quad E' \rightarrow +E \mid \varepsilon$$

$$T \rightarrow FT' \quad T' \rightarrow \times T \mid \varepsilon$$

$$F \rightarrow n \mid (E)$$

А потім для простоти аналізування розкрити нетермінали E і T в правилах для E і T .

$$E \rightarrow TE' \quad E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT' \quad T' \rightarrow \times FT' | \varepsilon$$

$$F \rightarrow n|(E) \rangle \rangle$$

Виводиться наступний крок.

Крок 6: Користувачу виводиться завдання.

«Для неї:

$$\text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{ (, \text{id} \};$$

$$\text{FIRST}(E') = \{ +, \varepsilon \};$$

$$\text{FIRST}(T') = \{ *, \varepsilon \};$$

Визначити FOLLOW.»

Виводиться наступний крок.

Крок 7: Користувачу виводиться завдання, варіанти відповіді.

«На кроці 1 обчислення множини FOLLOW у FOLLOW(E) включаємо»

- *
- +
- \$
-)

Користувач вибирає один з варіантів. При правильній відповіді (3) виводиться наступний крок.

Крок 8: Користувачу виводиться завдання, варіанти відповіді.

«На кроці 2, використовуючи продукцію $F \rightarrow (E)$, до FOLLOW(E) додається також»

- *
- +
- \$
-)

Користувач вибирає один з варіантів. При правильній відповіді (4) виводиться наступний крок.

Крок 9: Користувачу виводиться завдання, варіанти відповіді.

«На кроці 3, застосованому до правила $E \rightarrow TE'$, у FOLLOW(E') включаються»

- *, \$
- +,)
- \$,)
-)

Користувач вибирає один з варіантів. При правильній відповіді (3) виводиться наступний крок.

Крок 10: Користувачу виводиться завдання, варіанти відповіді.

«Оскільки $E' \Rightarrow \varepsilon$, вони також»

- потрапляють у FOLLOW(T)
- не потрапляють у FOLLOW(T)

Користувач вибирає один з варіантів. При правильній відповіді (1) виводиться наступний крок.

Крок 11: Користувачу виводиться завдання, варіанти відповіді.

«Відповідно до продукції $E \rightarrow TE'$ на кроці 2 у FOLLOW(T) включається усе з»

- FIRST(E), відмінне від ε
- FIRST(E'), схоже з ε
- FIRST(T'), відмінне від ε
- FIRST(E'), відмінне від ε

Користувач вибирає один з варіантів. При правильній відповіді (4) виводиться наступний крок.

Крок 11: Користувачу виводиться завдання, варіанти відповіді.

«Відповідно до продукції $E \rightarrow TE'$ на кроці 2 у FOLLOW(T) включається усе з»

- FIRST(E), відмінне від ε
- FIRST(E'), схоже з ε
- FIRST(T'), відмінне від ε
- FIRST(E'), відмінне від ε

Користувач вибирає один з варіантів. При правильній відповіді (4) виводиться наступний крок.

Крок 12: Користувачу виводиться завдання, таблиця.

«Заповнимо покрокову ітерацію. Вкажіть FOLLOW. До циклу while:»

Правило	FOLLOW
E	
E'	
T	
T'	
F	

Користувач заповнює поля. При правильній відповіді виводиться наступний крок. Відповідь:

Правило	FOLLOW
E	{ \$ }
E'	{ }
T	{ }
T'	{ }
F	{ }

Крок 13: Користувачу виводиться завдання, таблиця.

«Заповнимо покрокову ітерацію. Вкажіть FOLLOW. Після 1-ої ітерації:»

Правило	FOLLOW
E	
E'	
T	
T'	
F	

Користувач заповнює поля. При правильній відповіді виводиться наступний крок. Відповідь:

Правило	FOLLOW
E	{ \$,) }
E'	{ \$ }
T	{ +, \$ }
T'	{ +, \$ }
F	{ *, +, \$ }

Крок 14: Користувачу виводиться завдання, таблиця.

«Заповнимо покрокову ітерацію. Вкажіть FOLLOW. Після 2-ої ітерації:»

Правило	FOLLOW
E	
E'	
T	
T'	
F	

Користувач заповнює поля. При правильній відповіді виводиться наступний крок. Відповідь:

Правило	FOLLOW
E	{ \$,) }
E'	{ \$,) }
T	{ +,\$ }
T'	{ +,\$ }
F	{ *,+,\$ }

Крок 15: Користувачу виводиться завдання, таблиця.

«Заповнимо покрокову ітерацію. Вкажіть FOLLOW. Після 3-ої ітерації:»

Правило	FOLLOW
E	
E'	
T	
T'	
F	

Користувач заповнює поля. При правильній відповіді виводиться наступний крок. Відповідь:

Правило	FOLLOW
E	{ \$,) }
E'	{ \$,) }
T	{ +,\$,) }
T'	{ +,\$,)}
F	{ *,+,\$,) }

Крок 16: Користувачу виводиться повідомлення.

«Реалізація побудови FOLLOW виходить відразу з лем. Для алгоритму спочатку потрібно виконати побудову FIRST для граматики.

Коректність цього алгоритму доводиться точно так же, як і коректність алгоритму конструювання FIRST.»

3.3 Блок-схема алгоритму тренажера



Рисунок 3.1 – Блок-схема роботи тренажера



Рисунок 3.2 – Блок-схема роботи з практичними завданнями з полями для вводу

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис створення тренажера

Після створення проекту в Unity та додавання всіх робочих елементів в меню програми відбувається задання позиції для їх коректного відображення у вікні тренажера.

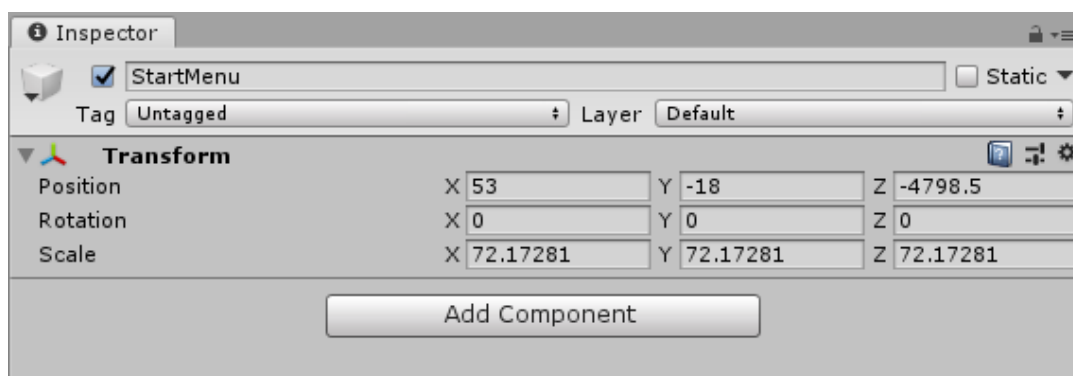


Рисунок 4.1 – Задання позиції елемента «Головне меню» у вікні програми

Для роботи з текстом в Unity використовуються текстові скрипти для роботи з текстовими елементами.

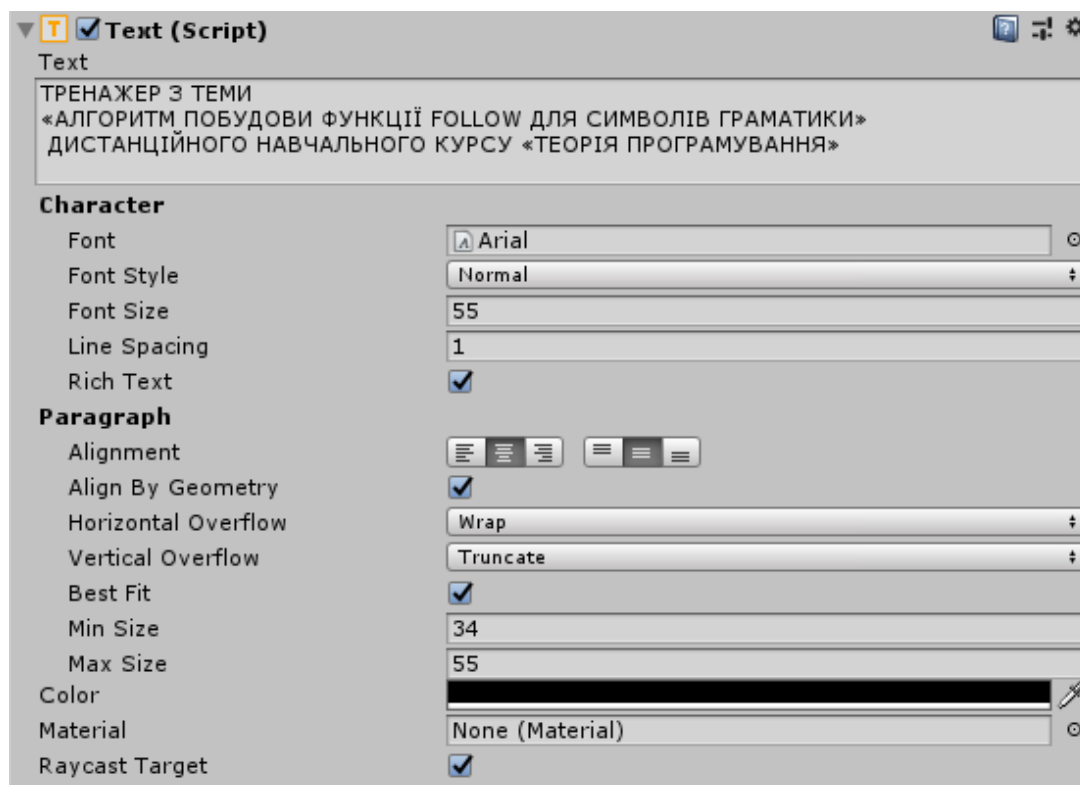


Рисунок 4.2 – Текстовий скрипт для роботи зі всіма текстовими елементами

Скрипти для роботи з кнопками реалізовано через булеві функції, при натисненні на кнопку вмикається певний елемент чи слайд та вмикається інший. Наприклад для кнопки «Початок» при натисненні вмикається вікно з головним меню та вмикається перше практичне завдання.

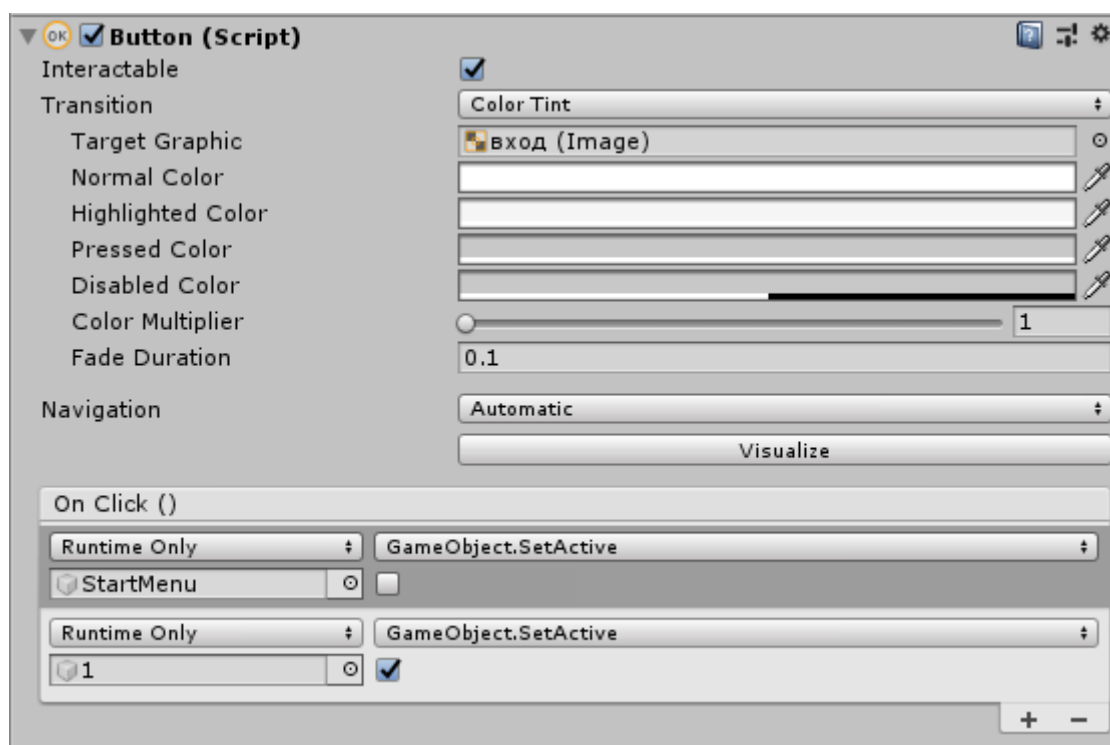
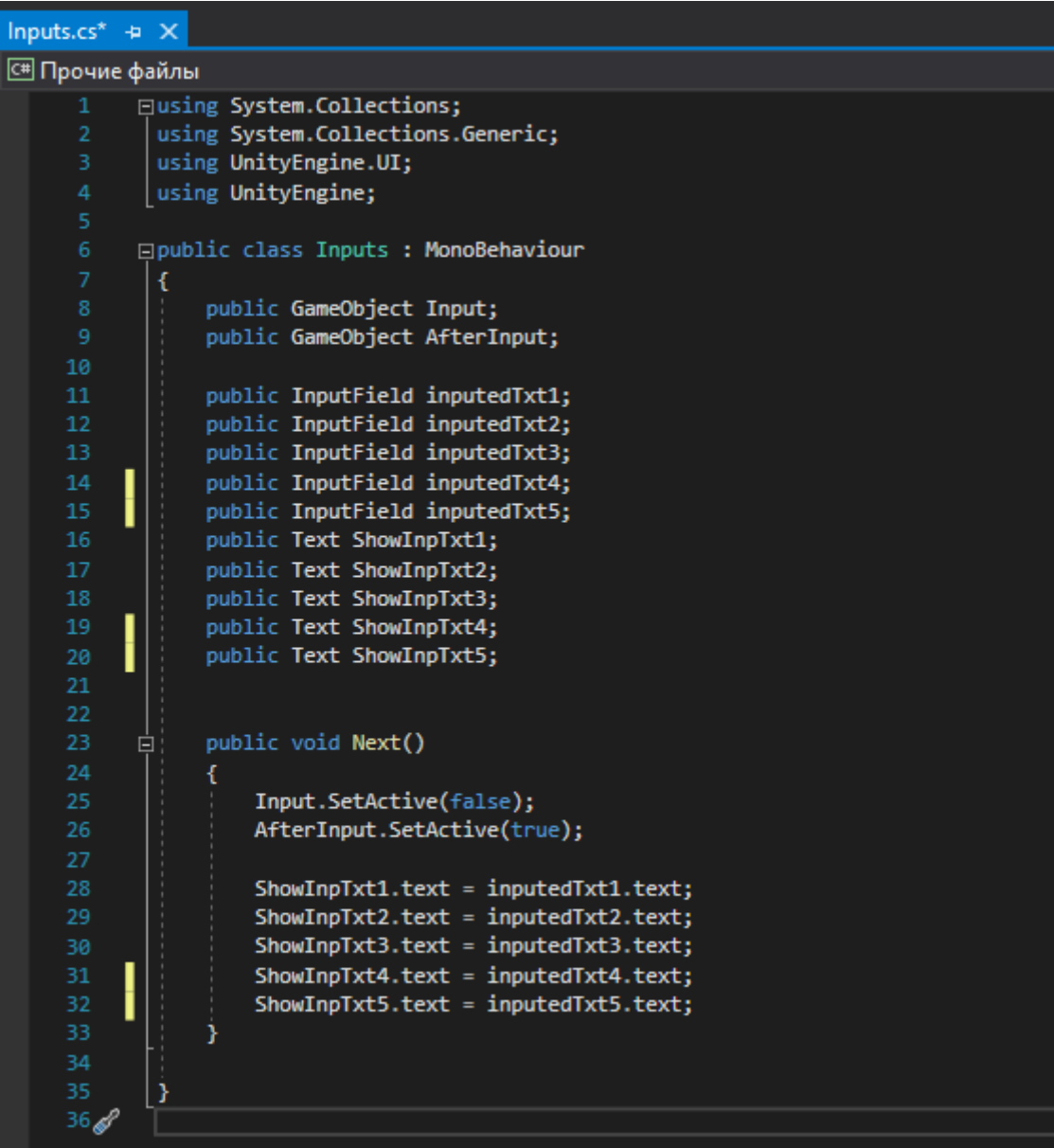


Рисунок 4.3 – Скрипт для кнопки «Початок»

Для роботи з завданнями з полями для вводу було використано наступний скрипт, код якого зображено на рисунку 4.4.



```

Inputs.cs*  + X
C# Прочие файлы
1  using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine.UI;
4      using UnityEngine;
5
6  public class Inputs : MonoBehaviour
7  {
8      public GameObject Input;
9      public GameObject AfterInput;
10
11     public InputField inputedTxt1;
12     public InputField inputedTxt2;
13     public InputField inputedTxt3;
14     public InputField inputedTxt4;
15     public InputField inputedTxt5;
16     public Text ShowInpTxt1;
17     public Text ShowInpTxt2;
18     public Text ShowInpTxt3;
19     public Text ShowInpTxt4;
20     public Text ShowInpTxt5;
21
22
23     public void Next()
24     {
25         Input.SetActive(false);
26         AfterInput.SetActive(true);
27
28         ShowInpTxt1.text = inputedTxt1.text;
29         ShowInpTxt2.text = inputedTxt2.text;
30         ShowInpTxt3.text = inputedTxt3.text;
31         ShowInpTxt4.text = inputedTxt4.text;
32         ShowInpTxt5.text = inputedTxt5.text;
33     }
34
35 }
36

```

Рисунок 4.4 – Скрипт для роботи завданнями з полями для вводу

Роботу даного скрипта можна описати як зчитування введених даних з відповідних полей для вводу та запис їх в текстову змінну. Після натиснення на кнопку продовження користувач отримує свої введені відповіді та має доступ до правильних відповідей. Після отримання першого результату користувач може повернутися до вводу та виправити свою відповідь, але зробити це можливо лише один раз.

4.2 Інструкція для роботи з тренажером

Після запуску тренажера відкривається вікно з головним меню тренажера. На отриманому вікні є інформація про тренажер та кнопка «Початок», що дозволяє перейти до практичних завдань.

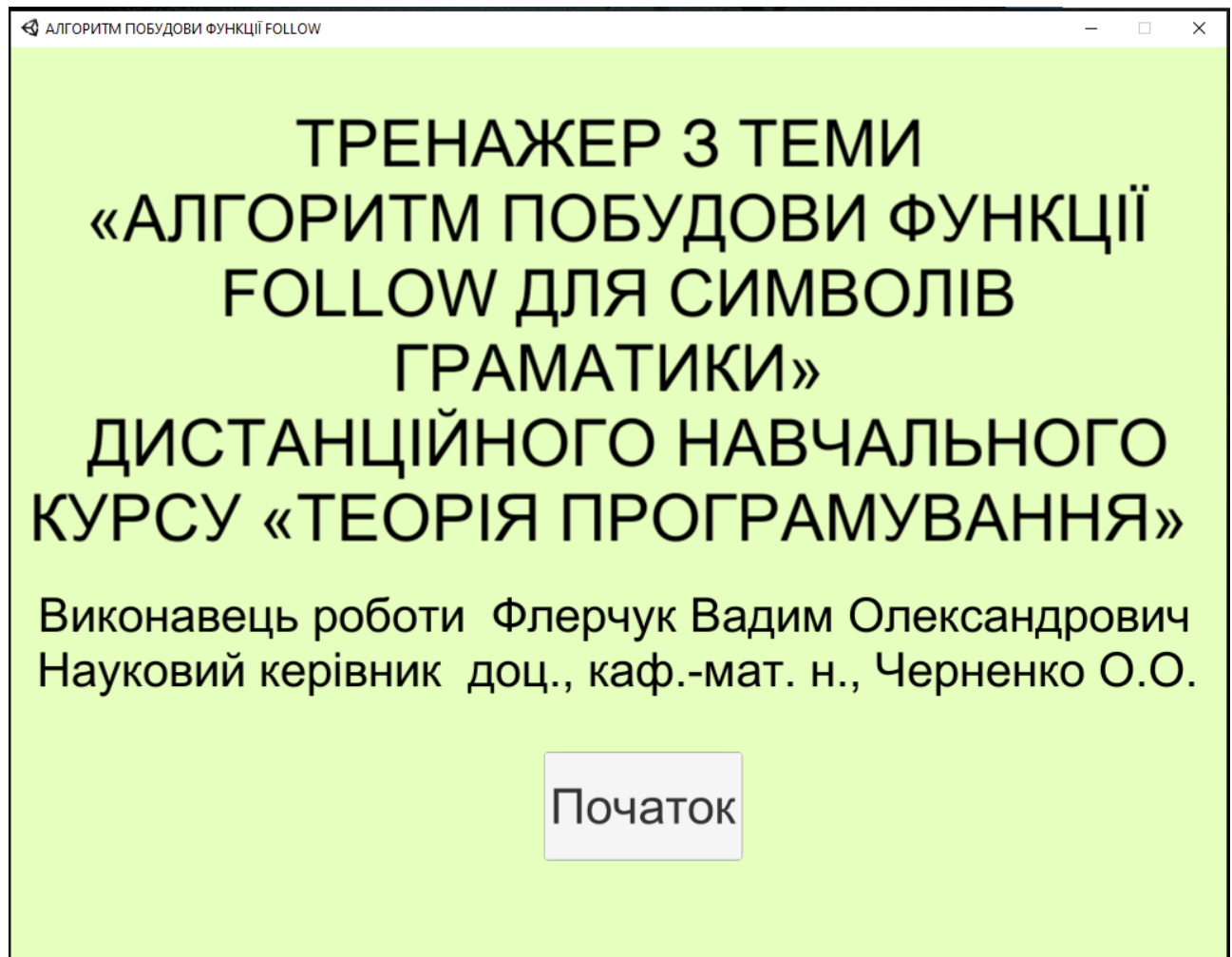


Рисунок 4.5 – Головне меню програми

Після переходу по кнопці «Початок» користувач отримує доступ до вікна з практичними завданнями.

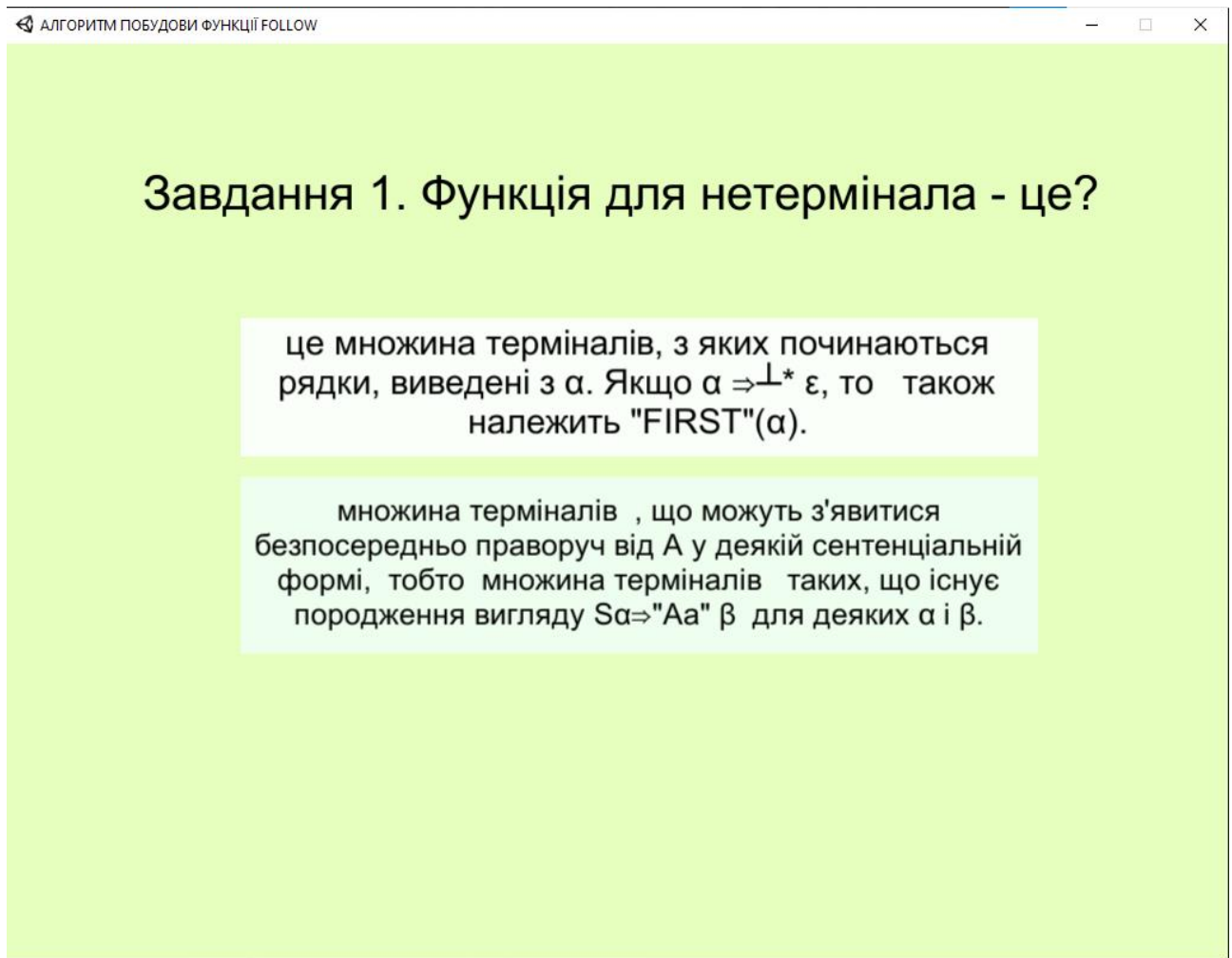


Рисунок 4.6 – Практичне завдання в тренажері

Після вибору неправильної відповіді користувач отримує повідомлення «Неправильно! Правильно - №».

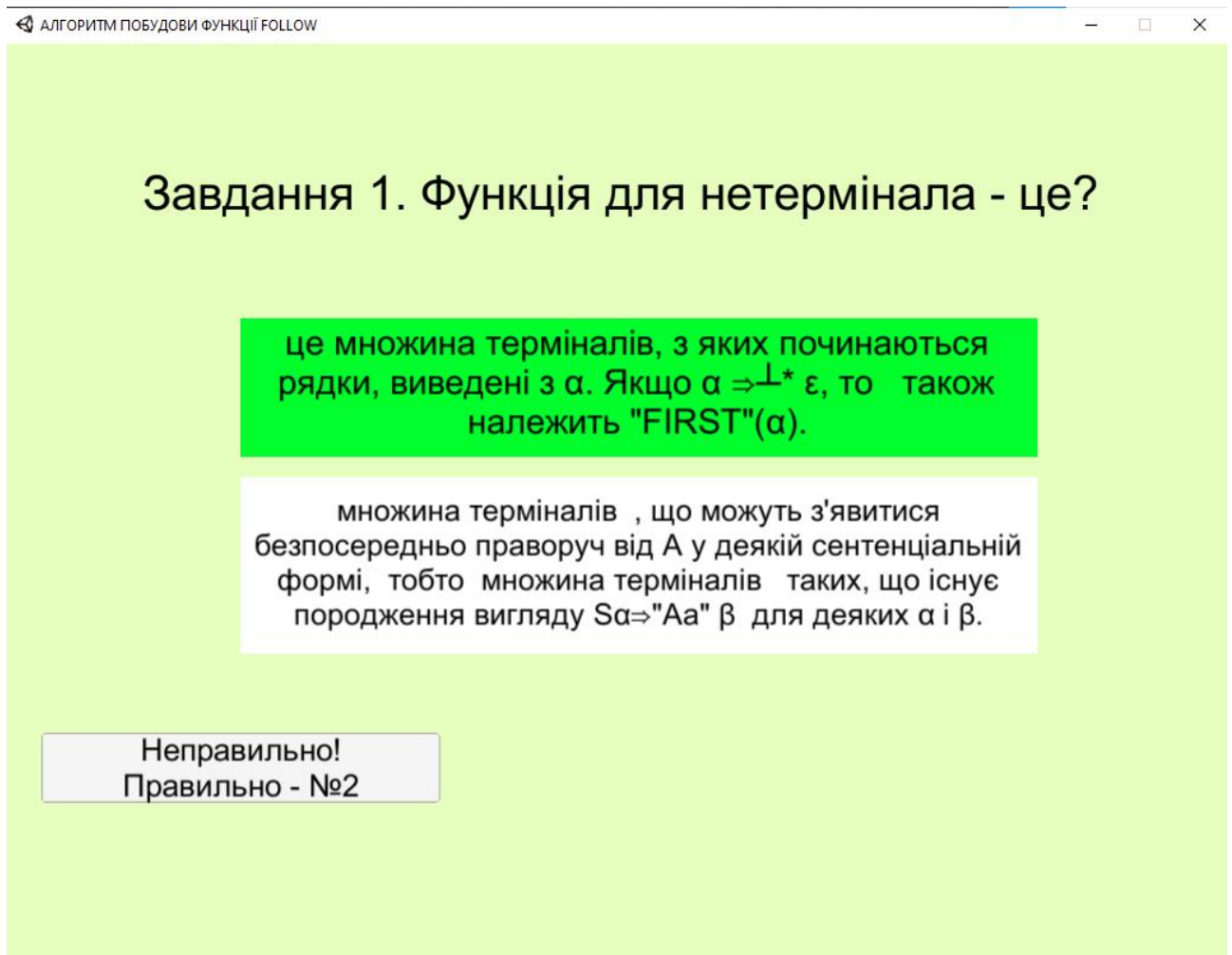


Рисунок 4.7 – Видача повідомлення про помилку

Після вибору правильної відповіді користувач отримує повідомлення «Правильно!» та отримує доступ до кнопки продовження.

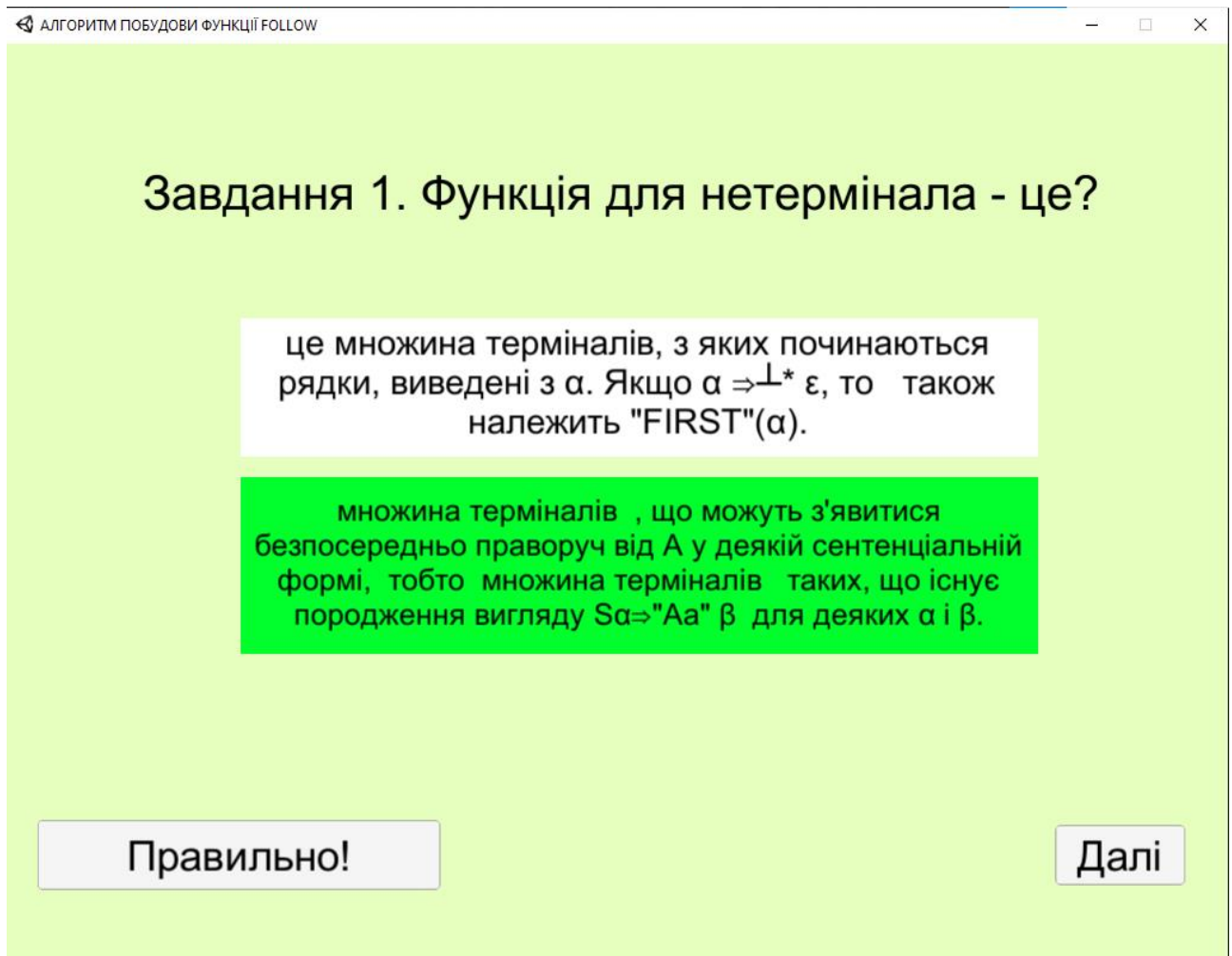


Рисунок 4.8 – Видача повідомлення про правильну відповідь, продовження роботи

Теоретичні матеріали в тренажері видаються наступним чином.

⏪ АЛГОРИТМ ПОБУДОВИ ФУНКЦІЇ FOLLOW — □ ×

Алгоритм. Побудова "FOLLOW"(X) для всіх X-нетерміналів граматика.

Крок 1 Помістити \$ у "FOLLOW"(S), де S – початковий символ і \$ – правий кінцевий маркер.

Крок 2 Якщо є продукція $A \rightarrow \alpha B \beta$, то усі елементи з множини "FIRST"(β), за винятком ϵ , додати до "FOLLOW"(B).

Крок 3 Якщо є продукція $A \rightarrow \alpha B$ або $A \rightarrow \alpha B \beta$, де "FIRST"(β) містить ϵ (тобто $\beta \Rightarrow \perp^* \epsilon$), то усі елементи з множини "FOLLOW" (A) додати до "FOLLOW" (B).»

Далі

Рисунок 4.9 – Видача теоретичних матеріалів

Отримання практичного завдання.

АЛГОРИТМ ПОБУДОВИ ФУНКЦІЇ FOLLOW

Завдання 4. Для неї:
"FIRST"(E)="FIRST" (T)="FIRST"(F)={ (,id};
"FIRST" (E')={+,ε};
"FIRST" (T')={*,ε};
Визначити FOLLOW.»

На кроці 1 обчислення множини "FOLLOW" у "FOLLOW"(E) включаємо

*
+
\$
)

Правильно!

Далі

Рисунок 4.10 – Вибір правильної відповіді на практичне завдання

Практичні завдання з полем для вводу виглядають наступним чином:

АЛГОРИТМ ПОБУДОВИ ФУНКЦІЇ FOLLOW

Заповнимо покрокову ітерацію. Вкажіть FOLLOW.
До циклу while:

Правило	FOLLOW
E	Enter text...
E'	Enter text...
T	Enter text...
T'	Enter text...
F	Enter text...

Рисунок 4.11 – Практичні завдання з полем для вводу

Практичні завдання з заповненими полями для вводу.

АЛГОРИТМ ПОБУДОВИ ФУНКЦІЇ FOLLOW

Заповнимо покрокову ітерацію. Вкажіть FOLLOW.
До циклу while:

Правило	FOLLOW
E	{}
E'	{}
T	{}
T'	{}
F	{}

Рисунок 4.12 – Практичні завдання з заповненими полями для вводу.

Після натиснення на кнопку «Ввести дані» користувач переходить до вікна з введеною відповіддю та підказкою. Існує можливість повернутися до вводу та виправити свою відповідь.

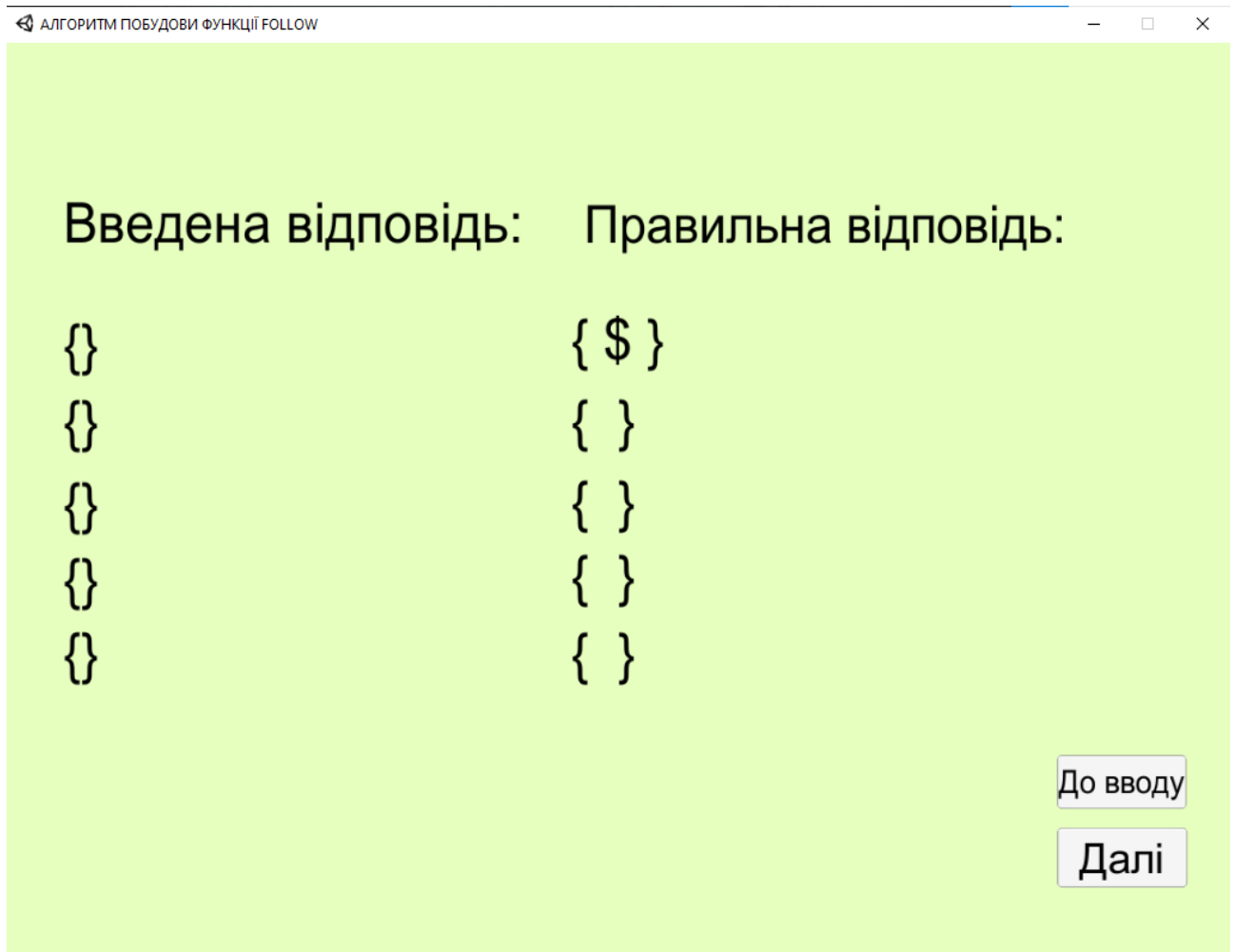


Рисунок 4.13 – Видача підказки в завданнях з полями для вводу

Виправлення відповіді.

АЛГОРИТМ ПОБУДОВИ ФУНКЦІЇ FOLLOW

Заповнимо покрокову ітерацію. Вкажіть FOLLOW.
До циклу while:

Правило

FOLLOW

E

{ \$ }

E'

{ }

T

{ }

T'

{ }

F

{ }

Ввести
дані

Рисунок 4.14 – Виправлення відповіді

Після виправлення відповіді кнопка для повернення до вводу недоступна.

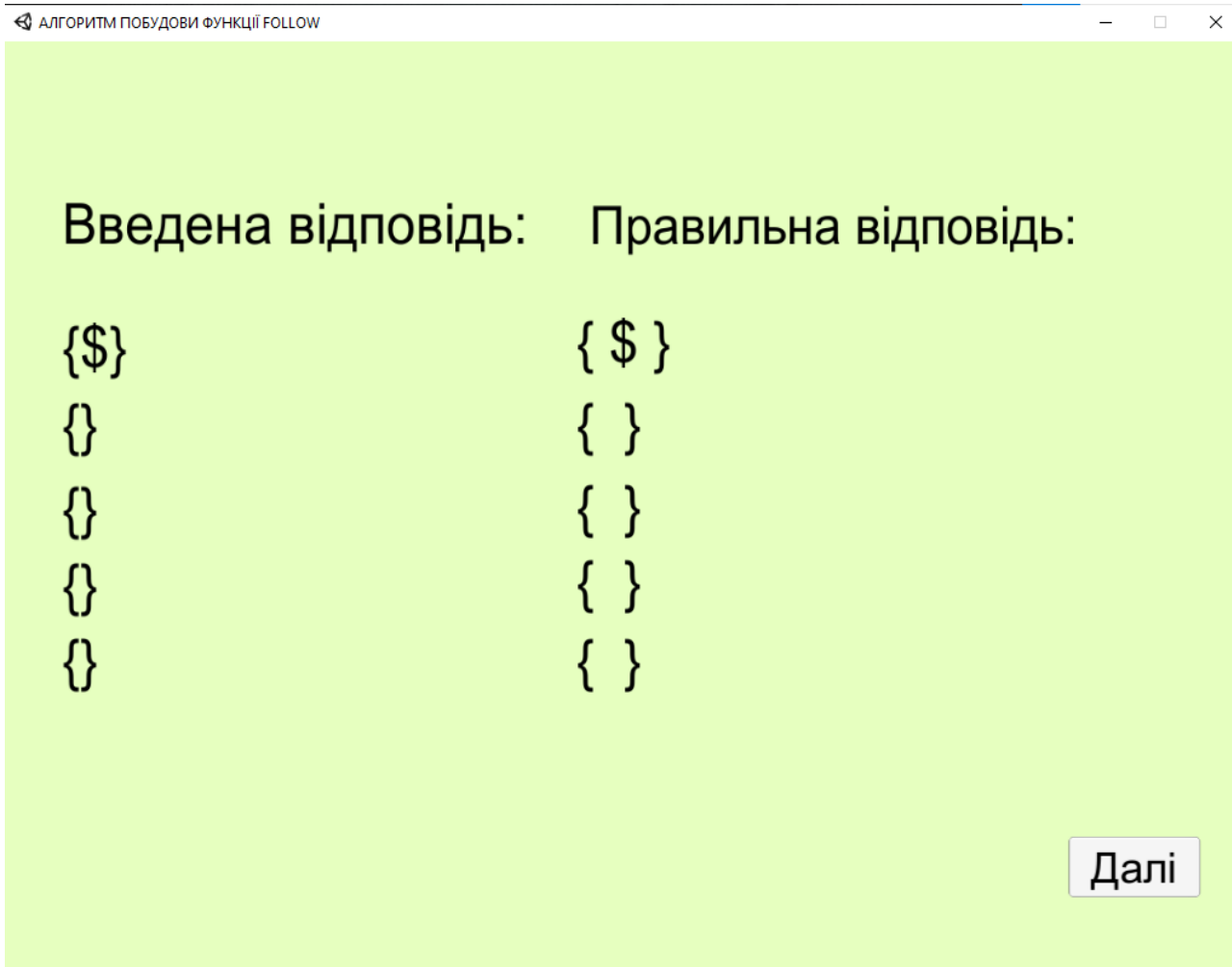


Рисунок 4.15 – Правильна відповідь на завдання з полем для вводу

Після завершення роботи з практичними матеріалами тренажеру користувач отримує висновок з виконаної роботи та повідомлення про завершення роботи. Реалізовано можливість повернутися до головного меню та повторити роботу з тренажером.

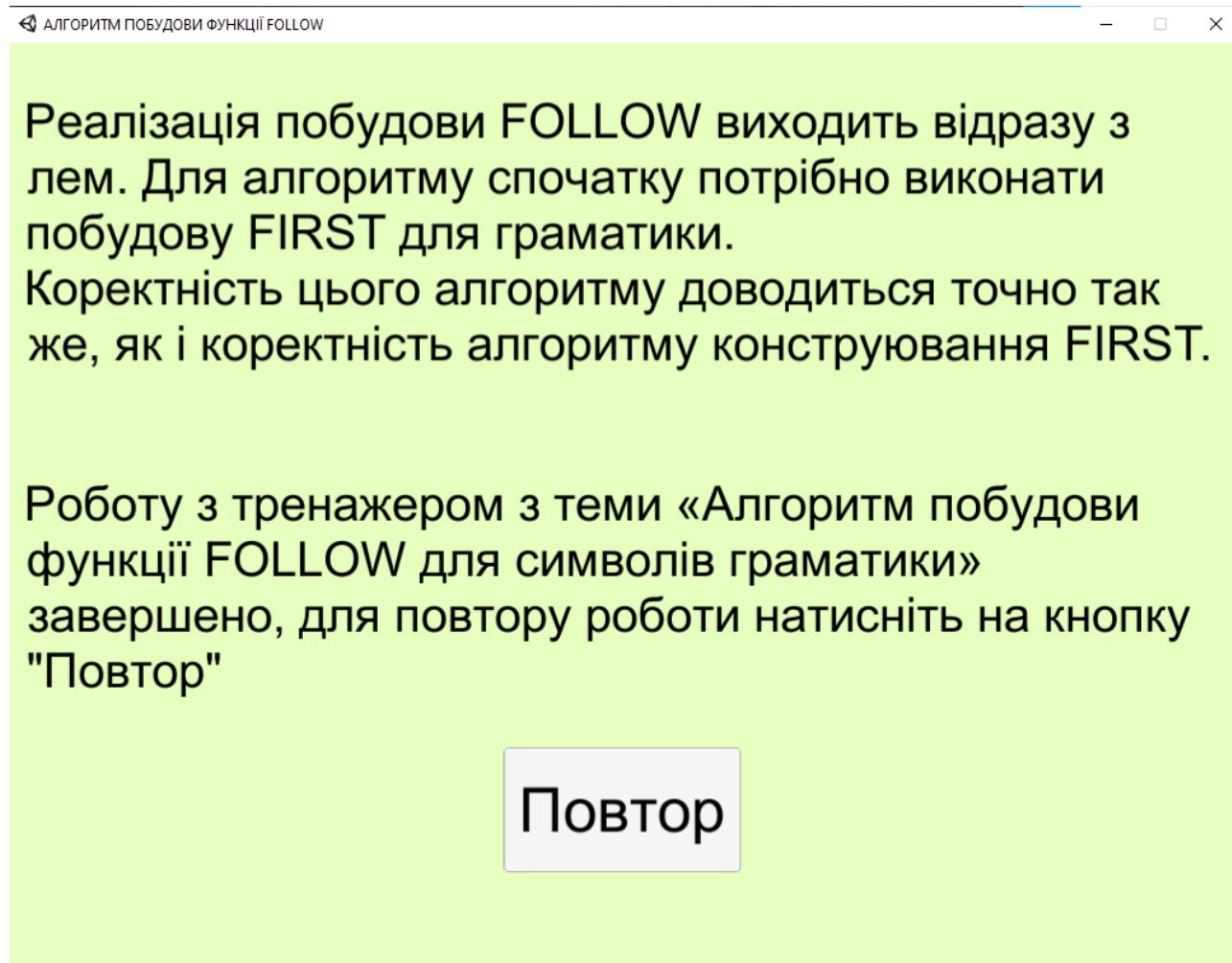


Рисунок 4.16 – Кінець роботи з тренажером, повтор роботи

ВИСНОВКИ

Очевидно, що для ефективного формування соціально-професійної компетентності майбутніх фахівців необхідно навчально-методичне забезпечення, розроблене на основі компетентнісного підходу.

У педагогічній літературі поряд з терміном «електронний тренажер» зустрічаються визначення «комп'ютерна навчальна програма», «симулятор». Найчастіше ці терміни ототожнюються або підміняються одне другим. У зв'язку з цим розглянемо зміст вищевказаних електронних засобів навчання.

Аналіз інтернет-ресурсів показує, що комп'ютерна навчальна програма являє собою ефективний засіб для підготовки фахівців. Комп'ютерні навчальні програми набули широкого поширення при вивченні іноземної мови та дисциплін соціально-гуманітарного циклу. В відмінну від комп'ютерної навчальної програми симулятор (від англ. Simulate-симулювати) визначається як жанр комп'ютерних ігор, в яких симулюються різні види діяльності людини, в основному різні спортивні змагання, пілотування літаків, автомобільні гонки та ін. Головний показник якості гри цього жанру максимальна наближеність ігрового процесу до реальних подій.

В роботі описано такі пункти:

1. Наведено актуальність теми роботи;
2. Розглянуто можливі типи тренажерів;
3. Розроблено алгоритм роботи тренажеру;

Для досягнення поставленої мети необхідно було вирішити такі завдання:

1. Визначити сучасні вимоги до тренажерів;
2. Розробити концепції функціонування та структуру тренажеру;
3. Запропонувати математичну модель;
4. Виявити вимоги до інструментальних засобів розробки програми на основі створеної моделі.

СПИСОК ЛІТЕРАТУРИ

1. Афанасьев А. Н. Модель и метод разработки и анализа компьютер-ных тренажеров [Текст] / А. Н. Афанасьев, Н.Н.Войт, Д. С. Канев // Автоматизация процессов управления. — 2015. — No 2. — С. 64–71.
2. Иорданский М. А. Учебные компьютерные тренажеры — важный класс новых образовательных продуктов[Текст] / М. А. Иорданский, Н.А.Мухин // Вестник Мининского университета. — 2016. — No 2. —С. 1–13.
3. Беляев, М.И. Технология создания электронных средств обучения / М.И. Беляев, В.В. Гриншкун, Г.А. Краснова.—М.: РУДН, 2006. —130 с.
4. Бабій М.С. Теорія програмування: Навчальний посібник [Електронний ресурс] / М.С. Бабій, О.П. Чекалов.— Суми: Вид-во СумДУ, 2009. — 181 с.
5. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. — Київ: КНУ ім. Т.Г. Шевченка, 2009. — 200 с. — Режим доступу: <http://tp.unicyb.kiev.ua/doc/TOP.pdf>.
6. Построение FIRST и FOLLOW [Електронний ресурс]. Режим доступу: https://neerc.ifmo.ru/wiki/index.php?title=_FIRST_%D0%B8_FOLLOW
7. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. — [Чинний від 2007-07-01]. — К. : Держспоживстандарт України, 2007. — 47 с.
8. Платформа Unity. — Режим доступу: <https://store.unity.com/ru/>
9. Основні принципи по роботі з Unity. — Режим доступу: unity.com/ru/

ДОДАТОК А. ДИСК З МАТЕРІАЛАМИ