**Національний технічний університет України**

**„Київський політехнічний інститут** **імені Ігоря Сікорського”**

**Кафедра автоматизації проектування енергетичних процесів і систем**

***КУРСОВА РОБОТА***

**з дисципліни: «Основи програмування - 3»**

**тема: «Комп’ютерна гра «Змійка»**

|  |  |
| --- | --- |
| **Керівник ас.Олєнєва К.М.** | **Виконала Спірідонова А.А.** |
| **Допущено до захисту** | **Студентка 1 курсу** |
| **«25» травня 2020р.** | **Групи ТІ-91** |
| **Захищено з оцінкою** | **залікова книжка №** |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |

**Для групи ТВ керівник: доц., к.т.н. Шпурик В.В**

**Київ – 2020**

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизації проектування енергетичних процесів і систем

**КУРСОВА РОБОТА**

з дисципліни «Основи програмування - 3»

(назва дисципліни)

на тему: «Комп’ютерна гра «Змійка»

Студентки групи \_\_\_\_\_\_\_\_\_\_\_\_ТІ-01\_\_\_\_\_\_\_\_\_\_\_\_

напряму підготовки 6.050103 Програмна інженерія

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка ECTS \_\_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_доц., к.т.н. Шпурик В.В.\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_асист. Олєнєва К.М.\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Київ - 2021 рік

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ТЕПЛОЕНЕРГЕТИЧНИЙ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_\_\_Автоматизації проектування енергетичних процесів і систем\_\_\_\_\_\_\_

Напрям підготовки \_\_\_\_\_\_\_\_\_6.050103 Програмна інженерія\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр і назва)

**ЗАВДАННЯ**

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Іванову Івану Івановичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім’я, по батькові)

1.Тема роботи \_Комп’ютерна гра «Змійка»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

керівник курсової роботи – \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи \_1 червня 2021 року\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до роботи: середовище розробки – \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) – \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів виконання курсової роботи | Строк виконання етапів роботи | Примітка |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
| 4. |  |  |  |
| 5. |  |  |  |
| 6. |  |  |  |
| 7. | Оформлення пояснювальної записки |  |  |
|  | Захист |  |  |

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

**Керівник курсової роботи** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

**АНОТАЦІЯ**

У курсовій роботі були розглянуті можливості розробки простих 2D ігор мовою програмування C++. Спроектований та створений програмний продукт за темою - комп’ютерна гра «Змійка».

Обрані найбільш оптимальні та гнучкі методи та алгоритми (можно написать парочку примеров, использованных технологий, конструкций и т.п. языка) для реалізації логіки програми. Комп’ютерна гра має простий і інтуїтивно зрозумілий інтерфейс, легке управління та швидке оновлення нових рівнів.

Обсяг пояснювальної записки 28 аркушів, кількість ілюстрацій - 5, 3 додатка.

**ANNOTATION**

…

The volume of the explanatory note is 28 sheets, the number of illustrations is 5, 3 applications.

Анотація – завжди 4 сторінка, але НЕ нумерується

Зміст також, нумерувати не потрібно, але він завжди починається з 5 сторінки; а ВСТУП уже – відповідно до довжини змісту – починається з 6 або 7 сторінки та є першою сторінкою, на якій з’являється та продовжується далі нумерація.

Додатки нумеруються відповідно порядку, але титульні сторінки додатків номерів мати не повинні

**ЗМІСТ**

Вcтуп…………………………………………………………………………………...6

1………………………………………………………………………………………...7

2………………………………………………………………………………………...8

3……………………………………………………………………………………….10

3.1…………………………………………………………………………………10

3.2………………………………………………………………………………....11

3.3……………..……………………………………………...…………..…….....19

3.4...…………………..…………..…………………………………………..…...23

4. …………………………………………………….………………………………..24

Висновки ……………………………………………………………………………..27

Список використаних джерел……………………………………………………….28

Додаток 1…………………………...………………………………………………...30

Додаток 2…………………………...………………………………………………...32

Додаток 3…………………………...………………………………………………...40

**ВСТУП**

Актуальність: Актуальність даної роботи полягає в освоєнні методів створення програмного продукту мовою С++. Підвищити знання в розробці програм методами об’єктно орієнтованого програмування.

Мета: Здати курсову роботу. Не вилетіти з КПІ.

Завдання: Створити гру «Поле-чудес». Гра має бути на двох користувачів. де один користувач загадуватиме слово а інший його відгадуватиме по літері. В грі має бути реалізовано: ввід слова першим користувачем, ввід другим користувачем літери яка на його думку міститься в слові, в разі якщо літери в слові немає, вивід на екран повідомлення про відсутність літери в слові, або якщо така літера є то вивід на екран цієї літери одразу на всіх її позиціях. Також має бути реалізований підрахунок за яку найменшу кількість кроків другий користувач може вгадати слово.

Методи дослідження:

**РОЗДІЛ 1. АНАЛІЗ ЗАВДАННЯ**

В ході аналізу завдання було вирішено, що гра «Поле-чудес» буде реалізована в вигляді консольної програми англійською мовою, тому . В програмі має бути меню. Окрім створення можливості грати вдвох вирішили створити можливість грати з комп’ютером.

**1.1. Програмна складова**

Було вирішено розділити програму на три окремі файли.

**РОЗДІЛ 2. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ**

В ході розробки програми

**2.1. Загальний опис продукту**

На початку виконання програми в основному файлі main.cpp відбувається підключення хедерного файлу PoleChudes.h в якому відбувається підключення основних бібліотек, та опис класів.

В функції main() створюється об’єкт класу Menu і викликається його метод menu().

**2.2 Клас Menu**

Клас Menu − клас нащадок класу PoleChudes відповідає за меню в грі.

**2.2.1. Змінні класу Menu**

В класі Menu є змінні типу int menu\_item, y, які відповідають за обраний пункт меню позицію курсора по координаті y для пунктів меню.

**2.2.1. Змінні класу Menu**

В класі

**2.3. Клас PoleChudes**

Клас PoleChudes − головний клас програми який відповідає за гру.

**2.3.1. Змінні класу PoleChudes**

В класі PoleChudes є дві змінні типу string userWord та guessWordByUser, які відповідають за загадане слово, та за слово в якому виводяться лише відгадані літери, відповідно.

Є масив змінних типу char enteredLetters в якому зберігаються літери які користувач, який відгадує слово, вже вводив.

В ньому знаходяться змінні типу int wordLenght, numberOfGuessedLetters, numberOfEnteredLetters та minAttemptions, які відповідають за кількість літер в загаданому слові, кількість відгаданих літер користувачем, кількість літер, які користувач вже вводив та мінімальна кількість кроків за яку можна завершити гру, відповідно.

Також в класі є об’єкт структури з бібліотеки windows.h COORD CursorPosition, в якому будуть зберігатись координати курсора.

**2.3.1. Метод void gotoXY(int x, int y)**

Метод

**2.3.2. Метод void play(bool computerOrPlayer)**

Метод void play(bool computerOrPlayer) запускає гру. Він отримує один параметр типу bool computerOrPlayer яка відповідає за вибір користувача грати з другим користувачем чи з комп’ютером.

За допомогою тернарного оператора змінній userWord надається значення яке повертає одна х функцій randomWord(), або enteringWord(). Якщо змінна computerOrPlayer має значення true, тобто користувач обрав грати з комп’ютером, то відбувається виклик функції randomWord(), а якщо змінна computerOrPlayer має значення false, тобто користувач обрав грати з іншим гравцем, то відбувається виклик функції enteringWord().

Викликається очищення консолі за допомогою функції system("cls") та методи someCalculating(), letsPlay() і final();

Функція нічого не повертає.

**2.3.3. Метод string randomWord()**

Метод string randomWord() відповідає за випадковий вибір слова з файлу words.txt.

**2.3.4. Метод string enteringWord()**

Метод string enteringWord() відповідає за отримання слова від користувача.

Він створює змінні типу string word, в яку буде записуватися слово яке введе користувач та змінна типу bool ind, яка виконує роль індикатора правильності вводу користувачем.

Запускається цикл з післяумовою в якому виконується очищення консолі за допомогою функції system("cls").

Викликається функція gotoXY(20, 2), з параметрами 25 і 5 відповідно.

Виводиться на екран напис «Welcome».

Викликається функція gotoXY(16, 7), з параметрами 16 і 7 відповідно.

Виводиться на екран напис « Please, enter one correct word».

Викликається функція gotoXY(20, 9), з параметрами 20 і 9 відповідно.

Зчитуємо рядок в змінну word за допомогою функції getline(cin, word) з стандартної бібліотеки iostream.

Задаємо змінній ind значення false пробігаємо по слову.

За допомогою оператора if перевіряємо змінну с використовуючи функцію isalpha(c) якщо с − літера, то функція isalpha(c) поверне значення true і if не спрацює, якщо с − не літера, то isalpha(c) поверне false, if спрацює і змінна ind змінить своє значення на true, if завершить свою роботу.

Коли в слові яке ввів користувач не лишиться символів цикл завершить свою роботу.

Далі за допомогою оператора if перевіряємо чи ввів користувач хоча б якісь символи, якщо змінна word починається з символу закінчення рядку, то спрацює if і змінна ind зминить своє значення на true, if завершить свою роботу

За допомогою оператора if перевіряємо змінну іnd, якщо іnd має значення true, тобто гравець ввів не літеру, if спрацює.

Спрацює функцію gotoXY(16, 9), з параметрами 16 і 9 відповідно.

Виведетеся на екран напис «Entering isn't correct».

Спрацює функцію gotoXY(5, 19), з параметрами 5 і 19 відповідно.

Виведеться на екран напис «Use <Enter> to continue».

Викличеться затримка єкрана за допомогою функції system("pause>nul"), GetAsyncKeyState(VK\_RETURN).

Оператор if завершить свою роботу.

Умовою циклу є змінна ind, якщо ind набуває значення true, то цикл продовжить свою роботу, а якщо false, то цикл завершиться.

Метод повертає змінну word.

**2.3.5. Метод void someCalculating()**

Метод void someCalculating() відповідає за ініціалізацію більшості змінних класу PoleChudes. В метод ніякі змінні не передаються.

Він надає змінній wordLenght значення кількості букв слова зі змінної userWord за допомогою методу length() класу string, який рахує кількість символів в рядку.

Надає змінній guessWordByUser значення змінної userWord.

Надає змінній minAttemptions значення яке повертає викликана функція minSteps().

Запускається цикл з лічильником в якості лічильника виступає змінна типу int i початкове значення лічильника 0, умова − поки значення лічильника менше значення змінної wordLenght, з кожним кроком лічильник збільшується на 1. В циклі символу змінної guessWordByUser з індексом і надається значення стмволу «\_».

Змінній numberOfGuessedLetters надається значення 0, тому цо користувач не вгадав ще жодної літери.

Змінній numberOfEnteredLetters надається значення 0, тому що користувач не ввів ще жодної літери.

Запускається цикл з лічильником в якості лічильника виступає змінна типу int i початкове значення лічильника 0, умова − поки лічильник менше змінної wordLenght, з кожним кроком лічильник збільшується на 1. В циклі відбувається перевірка символу змінної userWord з індексом і на належність до символів верхнього регістру за допомогою функції isupper(userWord[i]) з бібліотеки … якщо символ userWord[i] є символом верхнього регістру, то функція isupper(userWord[i]) поверне true і спрацює if. В ньому символу змінної userWord з індексом і надається значення функції tolower(userWord[i]) в яку передається цей символ і вона повертає цей символ в нижньому регістрі. If завершує свою роботу. Коли цикл завершить свою роботу функція завершиться.

Функція нічого не повертає.

**2.3.6. Метод int minSteps()**

Метод int minSteps() рахує за яку мінімальну кількість кроків можна відгадати слово, тобто рахує кількість букв в слові не враховуючи повторення.

В функції створюється змінна типу int count і надається їй значення 0.

**2.3.6. Метод void letsPlay()**

Метод void letsPlay() відповідає за відгадування слова користувачем.

Запускається цикл з передумовою. Умова циклу − значення змінної numberOfGuessedLetters менше за значення змінної wordLenght.

В циклі викликається функція gotoXY(20, 2) , з параметрами 20 і 2 відповідно.

Виводиться на екран «POLE CHUDES».

Викликається функція gotoXY(5, 4) , з параметрами 5 і 4 відповідно.

Виводиться на екран « - - - - - - - - - - - - - - - - - - - - - - - - - -».

В циклі викликається функція gotoXY(5, 5) , з параметрами 5 і 5 відповідно.

Виводиться на екран «табmin steps » , значення змінної minAttemptions, «таб|табyou used», значення змінної numberOfEnteredLetters, «attempsтаб|».

Викликається функція gotoXY(5, 6) , з параметрами 5 і 6 відповідно.

Виводиться на екран « - - - - - - - - - - - - - - - - - - - - - - - - - -».

Викликається функція gotoXY(5, 8) , з параметрами 5 і 8 відповідно.

Викликається метод showEnteredLetters().

Викликається функція gotoXY(5, 11) , з параметрами 5 і 11 відповідно.

Виводиться на екран «The current state of the word: », значення змінної guessWordByUser.

Викликається функція fillWord(enterLetter()), в яку передається значення яке повертає функція enterLetter().

Очищається консольза допомогою функції system("cls")

По закінченню циклу метод закінчує свою роботу. Метод нічого не повертає.

**2.3.7. Метод void showEnteredLetters()**

Метод void showEnteredLetters() відображає на єкрані вже введені користувачем літери.

В ньому виводиться на екран «Your entered letters: ».

За допомогою оператора if перевіряється чи значення змінної numberOfEnteredLetters не дорівнює нулю, якщо значення нулю не дорівнює, то спрацьовує if і запускається цикл з лічильником в якості лічильника виступає змінна типу int i початкове значення лічильника 0, умова − поки значення лічильника менше значення змінної numberOfEnteredLetters, з кожним кроком лічильник збільшується на 1. В циклі виводиться на екран елемент масиву з індексом і та символ « ». По щзавершенню циклу if завершує свою роботу. Якщо ж значення змінної numberOfEnteredLetters дорівнює нулю, то на екран виводиться «no letters».

Функція нічого не повертає.

**2.3.7. Метод void fillWord(char letter)**

Метод void fillWord(char letter) відповідає за заповнення рядку який виводиться на екран літерами, які користувач вже відгадав.

В метод передається змінна типу char letter, яка відповідає за літеру яку ввів користувач.

В методі запускається цикл з лічильником в якості лічильника виступає змінна типу int i початкове значення лічильника 0, умова − поки значення лічильника менше значення змінної wordLenght, з кожним кроком лічильник збільшується на 1. В циклі за допомогою оператора if перевіряється чи дорівнює значення змінної letter із значенням символу змінної userWord з індексом і. Якщо ці значення рівні, то спрацьовує if. Значенню символа змінної guessWordByUser з індексом і присвоюється значення змінної letter. Зеачення змінної numberOfGuessedLetters збільшується на один. If завершує свою роботу

По завершенню циклу робота методу завершується. Метод нічого не повертає.

**2.3.7. Метод char enterLetter()**

Метод char enterLetter() відповідає за отримання нової літери від користувача.

Створюється змінна типу char letter, яка буде набувати значення літери введеної користувачем, та змінна bool ind, яка є індикатором того, що користувач ще не вводив цю літеру.

Викликається функція gotoXY(5, 12) , з параметрами 5 і 12 відповідно.

Виводиться на екран «Enter one of few letters: ».

Змінна letter набуває значення яке повертає метод getLetterFromUser().

Запускається цикл з післяумовою.

В циклі змінна ind набуває значення false.

Запускається цикл з лічильником в якості лічильника виступає змінна типу int i початкове значення лічильника 0, умова − поки значення лічильника менше, або рівне значення змінної numberOfEnteredLetters, з кожним кроком лічильник збільшується на 1. В циклі за допомогою оператора if перевіряється чи значення змінної letter дорівнює значенню елемента масива enteredLetter з індексом і, кщо вони рівні то спрацьовує if.

В ньому викликається функція gotoXY(5, 12) , з параметрами 5 і 12 відповідно.

Виводиться на екран «Letter '», значення змінної letter, «' is already enterred».

Викликається функція gotoXY(5, 13) , з параметрами 5 і 13 відповідно.

Виводиться на екран «Enter enother one: ».

Викликається функція gotoXY(24, 13) , з параметрами 24 і 13 відповідно.

Значення змінної ind змінюється на true.

Значення змінної letter змінюється на значення яке повертає функція getLetterFromUser().

If завершує свою роботу.

По завершенню роботи циклу з лічильником цикл з пусляумовою переходить до перевірки умови. Умовою для нього є значення змінної ind.

По завершенню роботи циклу значення змінної numberOfEnteredLetters збільшується на один.

Значення елемента масиву enteredLetters, індексом якого є значення змінної numberOfEnteredLetters відняти один, змінює своє значення на значення змінної letter.

Функція повертає значення змінної letter.

**2.3.7. Метод char getLetterFromUser()**

Метод char getLetterFromUser() відповідає за ввід користувачем, що вгадує слово літер.

В методі створюються змінна типу char с, в яку дуде зчитуватися ввід користувачем літери, та змінна типу bool ind, яка слугує індикатором правильності вводу користувача.

Запускається цикл з післяумовою.

В циклі в змінну c записується введений користувачем символ.

За допомогою оператора if і функції isalpha(c) в яку передається значення змінної с перевіряється чи введений користувачем символ є літерою, якщо символ є літерою, то функція isalpha(c) поверне значення true, спрацює if і змінна ind набуває значення false, якщо ж символ який ввів користувач не є літерою, то isalpha(c) поверне значення false, спрацює else, змінна ind набуде значення true, викликається функція gotoXY(5, 12) , з параметрами 5 і 12 відповідно, виводиться на екран «That`s exactly not a letter», викликається функція gotoXY(5, 13) , з параметрами 5 і 13 відповідно, виводиться на екран «Try again: », викликається функція gotoXY(16, 13) , з параметрами 16 і 13 відповідно. If else завершає свою роботу. Ввід звільнюється за допомогою while (getchar() != '\n'); Умовою циклу слугує значення змінної ind.

По завершенню роботи циклу за допомогою оператора if і функції isupper(c), в яку передаємо знчення змінної с, перевіряємо чи належить літера, яка міститься в змінній с до літер верхнього регістру. Якщо літера належить, тофункція isupper(c) поверне значення true і спрацює if в якому значення змінної с зміниться на значення що поверне функція tolower(c), в яку передаємо змінну с, вона поверне літеру яка знаїодиться в змінній с в нижньому регістрі. If завершить свою роботу

Метод поверне значення змінної с і завершить свою роботу.

**2.3.7. Метод void final()**

Метод void final() відповідає за повідомлення гравцю про перемогу.

В методі викликається функція gotoXY(0, 2) , з параметрами 0 і 2 відповідно.

Виводиться на екран

« .. .. .... .. .. .. .. .. .. ..

.. .. . . .. .. .. .. .. .. . ..

.. . . .. .. .. . .. .. .. . ..

.. . . .. .. .. .. .. .. .. .. . ..

.. .... ..... ... ... .. .. ..»

Викликається функція gotoXY(7, 10) , з параметрами 7 і 10 відповідно.

Виводиться на екран «Guessed word: », значення змінної guessWordByUser.

Викликається функція gotoXY(7, 12) , з параметрами 7 і 1 відповідно.

Виводиться на екран «You used», значення змінної numberOfEnteredLetters, «attemptions».

Викликається функція gotoXY(7, 13) , з параметрами 7 і 13 відповідно.

Виводиться на екран «Minimal attemptions for this word», значення змінної minAttemptions.

Викликається функція gotoXY(7, 16) , з параметрами 7 і 16 відповідно.

Виводиться на екран «Bye!».

Викликається функція gotoXY(5, 19) , з параметрами 5 і 19 відповідно.

Виводиться на екран «Use <Enter> to continue».

Викличеться затримка єкрана за допомогою функції system("pause>nul"), GetAsyncKeyState(VK\_RETURN).

Метод завершить свою роботу, метод нічого не повертає.