Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 22.01.2022

Programowanie I

**projekt zaliczeniowy**

**"*Sudoku*"**

**Sławomir Szulik gr. lab. 08**

**1. Opis projektu.**

*Program ma za zadanie stworzyć plansze 9x9 z cyframi z przedziału od 1 do 9. Zadaniem użytkownika jest takie uzupełnienie planszy, gdzie w żadnym rzędzie, kolumnie czy też pomniejszym kwadracie 3x3 dana cyfra się nie powtórzy, a sama plansza zostanie w pełni uzupełniona (nie ma już żadnych ‘0’).*

**2. Wymagania**

* *‘wygenerowanie’ tablicy 9x9*
* *Wybór poziomu trudności*
* *Opcja podpowiedzi*
* *Informowanie użytkownika w realnym czasie o jego postępach*
* *Poinformowanie użytkownika o zakończeniu gry*

**3. Przebieg realizacji**

*Na ćwiczeniach z programowaniu udało mi się stworzyć w pełni sprawny system sprawdzania poprawności uzupełnienia tablicy 9x9. Następnym krokiem było tworzenie owej tablicy oraz nadanie jej odpowiedniego poziomu trudności poprzez ‘wymazanie’ poszczególnej ilości cyfr. Na sam koniec pozostało stworzenie pętli decydującej o zakończeniu programu, wyboru poziomu trudności i podpowiedzi przez użytkownika oraz poinformowanie o zakończeniu gry. Ważny krokiem była także walidacja podanych przez użytkownika znaków (jeżeli są wymagane cyfry, to litery nie przejdą(przynajmniej beta-testy nic nie wykryły :D)). Jeżeli chodzi o biblioteki, to <iostream> służy do porozumiewania się z użytkownikiem, a <stdlib.h> ma za zadanie generowanie liczb pseudolosowych.*

**4. Instrukcja użytkownika**

*Użytkownik wybiera poziom trudności:*

Obraz zawierający tekst

Opis wygenerowany automatycznie

*Następnie decyduje się o skorzystaniu z podpowiedzi:*



*Po tych krokach wyświetla się tablica Sudoku z miejscami (zerami) do uzupełnienia i informuje użytkownika o podaniu wartości, aby uzupełnić dane pole o podaną cyfrę:*

Obraz zawierający tekst, sprzęt elektroniczny, klawiatura

Opis wygenerowany automatycznie

*Jeżeli użytkownik wybrał opcję z podpowiedziami, to jego oczom ukażą się pola zaznaczone na czerwono, które to kolidują z poprawnością uzupełnienia planszy:*

Obraz zawierający czarny, sprzęt elektroniczny, klawiatura

Opis wygenerowany automatycznie

*Jeżeli użytkownikowi udało się uzupełnić całą planszę poprawnie, to zastanie go komunikat o zakończeniu programu/wygraniu:*

Obraz zawierający tekst

Opis wygenerowany automatycznie

**5. Podsumowanie i wnioski.**

*Program udało mi się zrealizować w pełni z jednym wyjątkiem – generowaniem tablicy 9x9. Mój program aktualnie posługuje się kilkoma tablicami 9x9, które to już są stworzone, a następnie program losuje na której to ma operować. Stworzeni losowania całkowicie ‘losowego’, to znaczy cyfra o cyfrze, okazało się znacznie trudniejsze niż przypuszczałem i zaimplementowanie tego w moim kodzie wiązało by się z ‘zapożyczeniem’ czyjegoś algorytmu ( a tego bym nie chciał). Sam program można by było jeszcze rozbudować o tablicę wyników, na której widoczny jest czas rozwiązania, czy użyto podpowiedzi i na jakim poziomie trudności się to odbyło. Można także dodać więcej poziomów trudności.*

Uwaga - do dokumentacji proszę nie wklejać całego kodu aplikacji. W sekcji realizacja można zmieścić fragmenty kodu, jeśli chcecie zwrócić uwagę na coś co było bardzo wymagające i konieczne jest dogłębnego jego omówienia.

Poza tym proszę komentować kod programu - to jest istotna część dokumentacji projektu.