Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 26.01.2022

Programowanie I

**projekt zaliczeniowy**

**"*Gra w wisielca*"**

**Bartosz Kała gr. lab. 3/5**

**1. Opis projektu.**

*Gra w wisielca polega na odgadnięciu z jakich liter składa się odgadywany wyraz. Użytkownik ma do wyboru czy sam ustali odgadywane hasło, czy to komputer poda wyraz, który będzie odgadywany. Jeśli użytkownik odgadnie wszystkie litery, które zawiera wyraz; wygrywa. Jeśli użytkownik poda literę, która nie znajduje się w haśle, program poda to do informacji oraz zacznie rysować postać tytułowego wisielca. Użytkownik ma 7 szans na pomyłkę. Jeśli wykorzysta wszystkie szanse; przegrywa.*

**2. Wymagania**

**-**

**3. Przebieg realizacji**

**Program składa się z pliku głównego; „Project\_Bartosz\_Kala.cpp”, zwierającego funkcje główną oraz pliku nagłówkowego; „Funkcje\_gry.h”, w którym znajdują się funkcje, z których korzysta program.**

**Oprócz tego program korzysta z pliku testowego: „słowa.txt”, w którym znajdują się wyrazy, z których korzysta komputer podczas realizacji poszczególnych działań.**

*Na początku program wywołuje funkcje menu()*

*W funkcji menu() dokonujemy wyboru, którą funkcje następnie wywołać*

*W funkcji zasady() i innych możemy wywołać funkcje menu()*

*W funkcji gra1() wywołujemy funkcje losowanie(), która pobiera wyraz pliku tekstowego oraz funkcja zmiana\_znakow(), która zmienia slowo w hasło do odgadnięcia.*

*Funkcja zmiana\_znakow() wywołuje funkcje rysowanie(), która w przypadku pomyłki rysuje wizerunek wisielca.*

*Funkcja gra2() wywołuje funkcje sprwdz(), która sprawdza poprawność wpisanych danych, funkcje zmiana\_znakow().*

*Dokładniejszy opis ich działania jest opisany w pliku Project\_Bartosz\_Kala.cpp*

**4. Instrukcja użytkownika**

*Na początku wyświetlane jest menu z opcją wyboru następnej operacji. Użytkownik powinien wpisać z klawiatury numer 1-4, jeśli wpisze inny znak program go o tym poinformuj, prosząc o ponowne dokonanie wyboru.*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po wpisaniu: ‘1’, użytkownikowi są wyświetlane zasady użytkowania programu.*

*Następnie jest ponownie proszony o dokonanie wyboru operacji. Użytkownik jest zobowiązany do wpisania jednej z wybranych liter. Program akceptuje duża i mała literę. Jeśli użytkownik wpisał znak nieprzewidziany, jest o tym informowany i jest proszony o ponowy wybór.*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Jeśli użytkownik wpisze ‘M’ lub ‘m’ zostanie przeniesiony do menu.*

*Jeśli wpisze ‘G’ lub ‘g’ to pozostanie w miejscu programu, w którym aktualnie się znajduje (jeśli aktualnie był w trybie rozgrywki, program na nowo rozpocznie rogrywkę).*

*Jeśli wpisze ‘R’ lub ‘r’ to program zakończy działanie; wyłączy się.*

*Działa to analogicznie w innych miejscach działania programu.*

*Po wpisaniu ‘2’ w menu program rozpocznie wykonywanie funkcji gra1(), która wywołuje funkcje losowanie() oraz funkcje zmiana\_znakow(). Opis ich działania jest opisany w pliku ”Funkcje\_gry.h”.*

*Obraz zawierający tekst

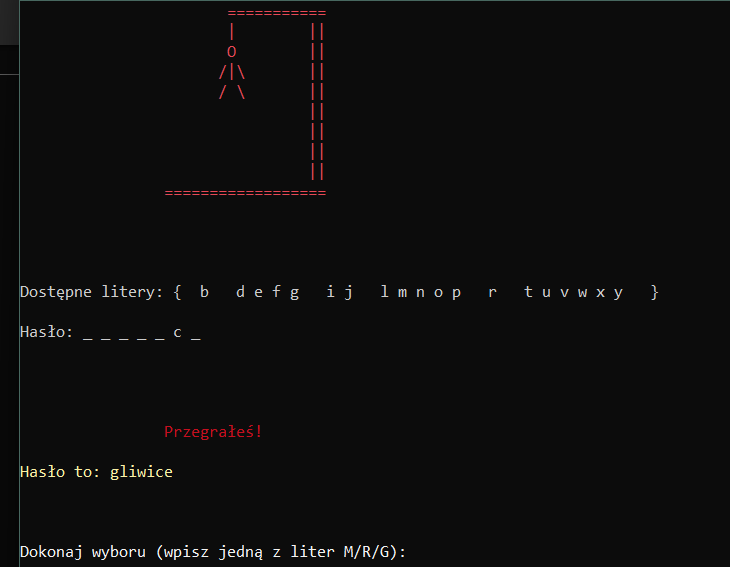
Opis wygenerowany automatycznie*

*Użytkownikowi jest wyświetlane powyższe okno. Jest on proszony o wpisanie znaku (znajdującego się w zbiorze „Dostępne litery”). Jeśli wpisany znak znajduje się w haśle, to zamienia „puste pole” w odgadniętą literę, jeśli się w nim nie znajduje to użytkownik jest o tym informowany, a program rozpoczyna rysowanie wizerunku wisielca.*

*Obraz zawierający tekst

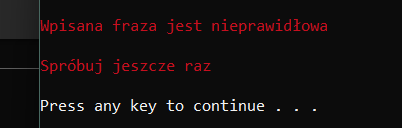
Opis wygenerowany automatycznie*

*Jeśli użytkownik nie odgadł hasła jest o tym informowany, a program podaje poprawne rozwiązanie i prosi o dokonanie wyboru następnej operacji.*

****

*Po wpisaniu ‘3’ w menu, program poprosi o wpisanie słowa, które następnie będzie hasłem, korzysta wtedy z funkcji gra2() oraz funkcji sprawdz(). Jeśli użytkownik wpisał niepoprawny znak lub hasło jest dłuższe niż 20 znaków, jest o tym informowany oraz proszony o ponowne wpisanie słowa (pogram traktuje spacje w tym miejscu jako znak).*

****

****

*Jeśli użytkownik poprawnie wpisał frazę, program działa analogicznie jak po wpisaniu ‘2’ w menu.*

*Jeśli w menu została wpisana ‘4’ program kończy swoje działanie.*

**5. Podsumowanie i wnioski.**

*Zostało zrealizowane główne założenie, czyli zmiana liter wyrazu w ‘\_’ oraz w przeciwną stronę, po uprzednim podaniu przez użytkownika znaku oraz identyczna operacja na słownie które zostało wybrane przez komputer.*

*Oprócz tego zostało dodane menu wyboru oraz „animacja” rysowania wisielca, jeśli podany przez użytkownika znak nie znajduje się w haśle.*

*Dodatkowo zostały dodane efekty zmiany koloru czcionki w konsoli oraz wyświetlanie zbioru liter, które nie zostały jeszcze użyte.*

*Problemy pojawiły się z losowym odczytem słów z pliku tekstowego. Problem polegał na zmianie z typu danych vector na typ danych string, który przyjmuje funkcja, odpowiedzialna za zmianę liter na: ‘\_’. Próbowałem zstąpić to działanie zapisem słów z pliku do dynamicznej tablicy, a następnie losowaniem słowa z tablicy, ale tu wystąpił problem z zapisem słów do tablicy.*

*Następnym problemem była funkcja odpowiedzialna za pobieranie słowa, które następnie będzie hasłem. Funkcja ignorowała fragment, odpowiedzialny za pobranie słowa, od razu oczekiwała wyboru następnej operacji, która następowała po zakończeniu głównego założenia funkcji. Rozwiązaniem okazało się dodanie funkcji z biblioteki string: cin.ignore().*

*Ewentualny rozwój projektu będzie polegał na:*

* *dodaniu czasomierza (liczenia ile czasu użytkownikowi zajęło odgadnięcie hasła)*
* *dodaniu opcji poziomu trudności hasła, które losuje komputer*
* *dodaniu listy rankingowej, z nazwą użytkownika oraz uzyskanym wynikiem*
* *stworzeniu systemu przydziału punktów*

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.