# AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

# Hangman

z przedmiotu

**Języki programowania obiektowego**

Elektronika  I rok

*Michał Zelek*

piątek 14:40

prowadzący: Rafał Frączek

15.06.2022

# 1. Opis projektu

Projekt zawiera implementację gry Hangman. Hangman to gra polegająca na zgadnięciu zaszyfrowanego słowa wylosowanego z wybranej przez gracza kategorii. Co turę gracz podaje literę i każde jej wystąpienie w zaszyfrowanym słowie zostaje odkryte, jeśli zaś litera ta nie występuje w słowie zmniejsza się liczba podejść o 1. Gra kończy się gdy gracz odgadnie całe słowo, lub gdy wyczerpie dostępne podejścia.

# 2. Project description

This project contains an inplementation of Hangman game. In this game player needs to guess an encrypted word which is random word generated from choosen by player category. Every turn player key in a letter and if it matches letters in encrypted word those matchings are reaveled to player. If any matchings of letter is found the number of attempts decrese by 1. Game ends if player guess the whole word of if number of attempts will decrese to 0.

# 3. Instrukcja użytkownika

W menu główny gracz proszony jest o dokonanie wyboru spośród dostępnych opcji za pomocą odpowiedniego znaku z klawiatury, jeśli dla wprowadzonego przez gracza znaku nie znaleziono opcji gra powtórzy żądanie. W przypadku zakończenia gry, bądź wejścia do instrukcji gry gracz proszony jest o wprowadzenie dowolnego znaku lub ciągu znaków i następnie wciśnięcie klawisza [Enter].

# 4. Kompilacja

W celu uruchomienia programu należy wykonać standardową kompilację.

# 5. Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

- *Gui.h*, Gui.*cpp* – deklaracja oraz implementacja klasy `Gui`, odpowiedzialnej za wyświetlenie ekranu startowego, oraz umożliwiającej graczowi poruszanie się po menu i wybór kategorii słowa.
- *Gameplay.h*, Gameplay.*cpp* – deklaracja oraz implementacja klasy `Gameplay`, która stanowi silnik gr, a także zawiera metody i zmienne niezbędne do umożliwienia rozgrywki.
- *Data.h*, Data.*cpp* – deklaracja oraz implementacja klasy Data, która zawiera metodę do wypełnienia zmiennej typu map znajdującej się w tej klasie, a zawierającej bazę słów z wybranej przez gracza kategorii. Wartości klucza danych w zmiennej typu map są używane przez klasę Gameplay do wylosowania słowa z tejże kategorii.
- *main.cpp* – plik w którym zostaje uruchomiona gra.

# 6. Zależności

Brak.

# 7. Opis klas

W projekcie utworzono następujące klasy:

- `Gui` – klasa odpowiedzialna za wyświetlenie menu i umożliwiająca graczowi poruszanie się po nim.
  - `static void menu(void)` – wyświetla logo gry wczytane z pliku logo.txt i opcje menu,

- `static string selectMenuOptions(void)` – funkcja zwracająca wybraną przez gracza opcję menu,
- `static void howToPlay(void)` – wyświetla zawartość opcji menu „How to play”,
- `static int selectCategory(void)` – umożliwia graczowi wybór kategorii i zwraca jej nr wcześniej dokonując konwersji z pomocą funkcji stoi(),

- `Data`– klasa zawierająca bazę danych dla wybranej kategorii.
  - `static void fillDataBase(int)` – uzupełnia zmienną publiczną typu map znajdującą się w tej klasie sparsowanymi danymi z odpowiedniej kategorii – ustalonej za pomocą argumentu metody

- `Gameplay` – klasa zawierająca silnk gry i metody niezbędne do uruchomienia rozgrywki.
  - `static void setSeed(void)` – ustawia wartość zmiennej klasowej int seed na losowy nr z przedziału zawierającego wszystkie klucze ze zmiennej w klasie Data typu map o nazwie dataBase-jest to bieżąca kategoria,
  - `static void setCorrectAnswer(int)` – ustawia zmienną klasową string correctAnswer reprezentującą słowo do zgadnięcia na wartość zmiennej klasy Data dataBase o kluczu seed,
  - `static void fillAlphabet(void)` – uzupełnia alfabetem łacińskim zmienną klasową typu vector<char> o nazwie alphabet
  - `static void decreseAttempts(void)` – zmniejsza wartość zmiennej klasowej int avaliableAttempts reprezentującą dostępne podejścia o 1.
  - `static bool isLetter(string)` – sprawdza czy podany argument jest literą jeśli tak to zwraca 1 w przeciwnym wypadku 0,
  - `static bool wasAlreadyGiven(string)` – sprawdza czy znak podany przez gracza był już wcześniej podany jeśli tak to zwraca 1 w przeciwnym wypadku zwraca 0,
  - `static void guessLetter(void)` – metoda w której odbywa się rozgrywka, zawiera ona nieskończoną pętle wykonującą się do czasu wygranej, bądź porażki. Potem wykonuje się pętla wewnętrzna gdzie gracz jest proszony o wprowadzenie litery, potem odbywa się sprawdzanie czy wprowadzony znak to litera, następnie sprawdzana jest poprawność wprowadzonej litery. Powyższa sekwencja wykonuje się do czasu wyczerpania dostępnych prób, bądź odgadnięcia słowa. Każda iteracja wewnętrznej pętli to kolejna tura na której starcie widoczna jest ilość prób, status szukanego słowa oraz użyte już litery. Każda akcja bądź błąd wejścia powoduje wyświetlenie stosownego komunikatu,
  - `static void runGame(void)` – metoda uruchamiająca grę w niej następuje wywołanie wymienionych metod: menu() - wyświetlenie menu, selectMenuOptions()-poruszanie się po menu, selectCategory()-wybranie kategorii słowa, fillDataBase()-uzupełnienie bazy danych słowami z wybranej kategorii, setSeed()-ustawienie ziarna losowości, setCorrectAnswer()-na podstawie ziarna wybranie odpowiedniego słowa z bieżącej kategorii, fillAlphabet(), guessLetter()-faktyczna rozgrywka. Powyższe instrukcje są zapętlone.

# 8. Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

- `flowers.txt` – plik zawierający bazę słów dla kategorii „Flower”:
  - każda linia zawiera dane które są zapisywane do zmiennej typu map struktura tych danych to: {wartość klucza (kolejne liczby) , wartość (słowo) }

- `motorization.txt` – plik zawierający bazę słów dla kategorii „Motorization”:
  - struktura danych-jak wyżej

- `logo.txt` – plik zawierający logo gry.

# 9. Dalszy rozwój i ulepszenia

Projekt można rozszerzyć o nowe funkcjonalności:

- powiększyć bazę danych
- bazę danych zawrzeć w jednym pliku
- dodać bardziej intuicyjny i wygodny interfejs graficzny
- dodać zmieniające się okienko w którym będzie rysowany tytułowy hangman
- poprawić mechanizm wyświetlania użytych liter

# 10. Inne

Brak.