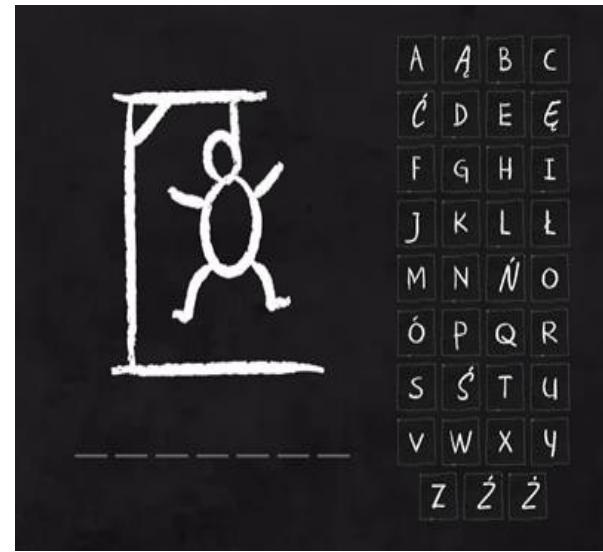


# Gra w wisielca



# Opis projektu

Gra polega na odgadywaniu ukrytego słowa, zgadując po jednej literze, przy ograniczonej ilości prób.

Gracz może wybrać jedną spośród trzech kategorii słów: zwierzęta, zawody i przedmioty.

===== MENU =====

- 1. Play
- 2. Quit

Select an option: |

===== CATEGORY =====

- 1. Animal
- 2. Profession
- 3. Object

Choose a category number:

Następnie użytkownik zgaduje litery, aby odgadnąć hasło. Jeśli odgadnie nieprawidłowo sześć razy gra się kończy.

-----

Guess a letter: e

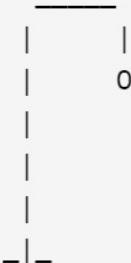
Correct guess!

Used letters: e

\_ e -----

Guess a letter: m

Incorrect guess! Remaining attempts: 5



Used letters: e m

Used letters: e m r a l t

— — e \_ \_

Guess a letter: f

Incorrect guess! Remaining attempts: 0



Used letters: e m r a l t f

— — e \_ \_

Game over! The word was: pigeon

# Jakie napotkałam problemy

- Implementacja losowego wyboru słowa
- Walidacja danych wejściowych



# Z czego jestem najbardziej dumna

- **Funkcja rysująca wisielca** - udało mi się znacznie ją skrócić jednocześnie zachowując jej funkcjonalność. Dzięki temu kod jest bardziej zwięzły i czytelny.

```
void drawHangman(int remainingGuesses) {  
    std::cout << " _ " << std::endl;  
    std::cout << " | " << std::endl;  
    std::cout << " | " << (remainingGuesses < 6 ? "0" : " ") << std::endl;  
    std::cout << " | " << (remainingGuesses < 4 ? "/" : " ") << (remainingGuesses < 5 ? "|" : " ") <<  
        (remainingGuesses < 3 ? "\\" : " ") << std::endl;  
    std::cout << " | " << (remainingGuesses < 2 ? "/" : " ") << " " << (remainingGuesses < 1 ? "\\" : " ") <<  
        std::endl;  
    std::cout << " | " << std::endl;  
    std::cout << "_|_" << std::endl;  
    std::cout << std::endl;  
}
```

- **Obsługa wyjątków**

w kontekście menu,  
kategorii oraz wprowadzania  
liter.

W przypadku wprowadzenia  
nieprawidłowej wartości  
program wyświetla  
użytkownikowi odpowiedni  
komunikat i prosi o ponowne  
wprowadzenie danych.

```
===== CATEGORY =====
```

1. Animal
2. Profession
3. Object

```
Choose a category number: ???
```

```
Invalid category. Please try again.
```

```
-----
```

```
Guess a letter: 123
```

```
Invalid input. Please enter a letter.
```

# Czego się nauczyłem

- **Tworzyć testy jednostkowe**

Każdy komponent gry - biblioteka słów, interfejs i menu - są testowane w celu zapewnienia funkcjonalności. Dzięki temu udało mi się szybko wykryć i naprawić błędy związane z nieprawidłowym działaniem funkcji losujących.

Animal Word Library Test:

Random Word: eagle

Random Word: badger

Random Word: tortoise

Random Word: monkey

Random Word: pigeon

---

Profession Word Library Test:

Random Word: mechanic

Random Word: fisherman

Random Word: surgeron

Random Word: teacher

Random Word: accountant

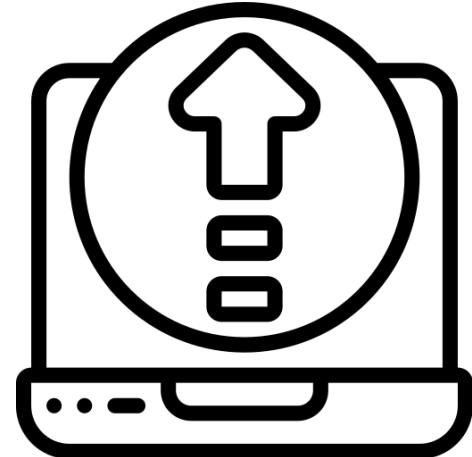
- Przetwarzać ciągi znaków

Wykorzystanie funkcji takich jak tolower pozwoliło na konwersję wprowadzanych dużych liter na małe litery, a funkcja isalpha umożliwiła sprawdzenie, czy wprowadzany znak jest literą.

```
- - - - -  
Guess a letter: A  
Correct guess!  
Used letters: a  
- - - - - a -
```

# Co można ulepszyć

- Dodać system punktacji oraz poziomy trudności
- Stworzyć więcej kategorii słów oraz umożliwić użytkownikowi dodawanie własnych kategorii
- Dodać graficzny interfejs użytkownika dla większej wygody grania



Dziękuję