

Konrad Krupski 310729

Julia Polak 310965

Nazwa projektu: Wisielec

Opis działania programu

Tematem projektu jest gra w wisielca. Gra polega na tym, że użytkownik musi zgadnąć słowo, które jest zupełnie losowe. Użytkownik ma możliwość wyboru poziomu trudności poprzez naciśnięcia odpowiedniego przycisku na interfejsie graficznym. Użytkownik jest ograniczony przez skończoną liczbę żyć wynoszącą 6. Do swojej dyspozycji posiada wyświetloną na interfejsie graficznym klawiaturę z poszczególnymi literami alfabetu. W przypadku gdy nie uda mu się zgadnąć słowa, rozgrywka kończy się przegraną i rozwiązanie jest prezentowane w oknie dialogowym.

Słowa użyte w grze planujemy zrobić angielskie.

Przebieg rozgrywki

Podczas uruchomienia programu nastąpi wyświetlenie okna dialogowego, w którym użytkownik będzie mógł wybrać interesujący go poziom trudności rozgrywki. Poziomy trudności różnią się skomplikowaniem słów. Dla każdego z poziomów słowa odczytywane są z innych plików.

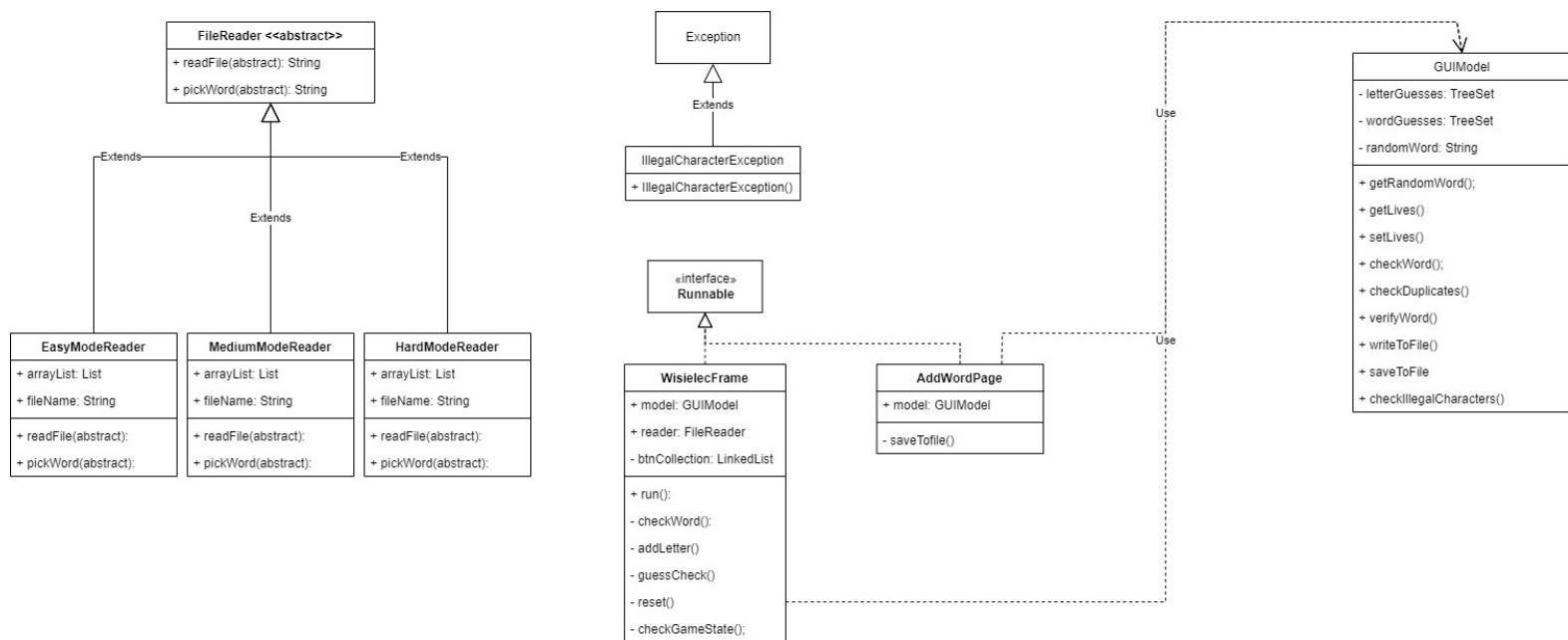
Następnie uruchamiane jest okno główne aplikacji, w którym prowadzona jest rozgrywka. Użytkownik klika na literki znajdujące się a klawiaturze w interfejsie graficznym. Po kliknięciu przycisku jest on deaktywowany, aby zabezpieczyć przed klikaniem kilka razy w tą samą literę. Użyte litery dodawane są do odpowiedniej kolekcji i następnie sprawdzane za pomocą metody, która zajmuje się wyświetlaniem słowa na ekranie. Jeżeli literka w słowie została zgadnięta to jest ona wyświetlana w przeciwnym wypadku widnieje znak „-”.

Jeżeli użytkownik jest w stanie zgadnąć słowo to może do tego użyć specjalnego przycisku „Guess word”, które uruchamia okno dialogowe z możliwością wpisania tekstu albo po prostu użyć klawiatury GUI.

Jeżeli użytkownik zgadnie słowo to rozgrywka się kończy, w przeciwnym wypadku użytkownik informowany jest o swojej porażce i jest mu prezentowane rozwiązanie zagadki.

Podczas rozgrywki użytkownik ma do swojej dyspozycji również przycisk „Restart”, który uruchamia całość od początku.

Diagram UML



Przeznaczenie klas

FileReader

Klasa abstrakcyjna, po której będą dziedziczyć poszczególne klasy odpowiadające za odczytywanie słów o danym poziomie trudności z plików tekstowych. Klasa wymaga zaimplementowania dwóch metod, `readFile()`, która odczytuje wszystkie słowa z pliku tekstowego i zapisuje je do kolekcji oraz metody `pickWord()`, która losuje słowo.

EasyModeReader, MediumModeReader, HardModeReader

Klasy, które wykonują operację odczytania słów z pliku tekstowego dla swojego poziomu trudności oraz losują jedno z tych słów.

IllegalCharacterException

Klasa dziedzicząca po `Exception`, która ma zapewnić obsługę sytuacji wyjątkowej w momencie, gdy użytkownik będzie chciał wpisać do bazy danych nielegalny znak.

WisielecFrame

Klasa, która będzie służyła jako główne okno aplikacji, gdzie będzie prowadzona rozgrywka. Wyświetlane w niej będą przyciski z odpowiednimi literami, słowo, które należy zgadnąć oraz dodatkowe przyciski pozwalające kierować stanem gry np. „Add word”, „Restart”, „Guess word”. Klasa będzie implementowała interfejs `Runnable`, gdyż pozwoli to na uruchomienie jej w osobnym wątku. `AddWordPage`.

Najważniejsze funkcje klasy zostały przedstawione na diagramie UML:

run()

Pozwala na uruchomienie okna w nowym wątku. Implementacja tej metody jest wymuszona poprzez interfejs `Runnable`.

checkWord()

Klasa sprawdzająca wylosowane słowo i wypisująca odpowiednią ilość liter do zgadnięcia na ekranie. Jeżeli, któraś z liter została zgadnięta przez użytkownika to na ekranie jest ona wypisywana, w przeciwnym wypadku wypisywany jest znak „-”.

addLetter()

Metoda pozwalająca na dodanie zgadniętej przez użytkownika litery do kolekcji.

guessCheck()

Metoda realizująca funkcję sprawdzenia słowa zgadniętego przez użytkownika. Użytkownik ma możliwość zgadnięcia całego słowa przy użyciu odpowiedniego guzika GUI. Jeżeli słowo zostanie zgadnięte to rozgrywka kończy się wygraną.

reset()

Metoda pozwalająca użytkownikowi na zrestartowanie rozgrywki, oznacza to:

- 1) Ponowny wybór poziomu trudności
- 2) Ponowne wylosowanie nowego słowa
- 3) Odnowienie ilości żyć

checkGameState()

Metoda realizująca sprawdzenie czy użytkownikowi udało się zgadnąć całe słowo. Natomiast, jeżeli wyczerpał już wszystkie punkty zdrowia to rozgrywka kończy się przegraną.

AddWordPage

Klasa wyświetlająca w osobnym wątku interfejs pozwalający na dodanie nowego słowa do bazy danych. Posiada ona metodę saveToFile(), która realizuje sprawdzenie proponowanego słowa w taki sposób aby nie zawierały się tam nieodpowiednie znaki i następnie za pomocą strumieni zapisuje to słowo do pliku.

GUIModel

Klasa, która będzie przechowywała wszystkie potrzebne metody używane przez okno główne i okno realizujące opcję dodawania nowych słów do bazy danych.

Posiada ona dodatkowo wszystkie potrzebne kolekcje do przechowywania zarówno zgadniętych litera jak i słów, na podstawie których obliczane są pozostałe punkty zdrowia.

CheckIllegalCharacters()

Metoda realizująca sprawdzanie czy słowo zawiera nielegalne znaki np. znaki polskie takie jak ś, ć, ż.