Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 10.07.2020

Programowanie obiektowe i graficzne

**projekt zespołowy**

**"*Familiada”***

**Skład zespołu projektowego:**

**Ewa Lechowicz gr. lab. A**

**Sonia Zieleźny gr. lab. C**

**Maciej Bugajski gr. Lab. C**

**Wykaz zadań i prac zrealizowanych przez zespół projektowy:**

|  |  |  |
| --- | --- | --- |
| Imię Nazwisko | Odpowiedzialny za: | Zrealizował zadania: |
| LECHOWICZ Ewa | 1. Stworzenie bazy danych | 1A. Zaprojektowała strukturę bazy danych |
| 1B. Dodała rekordy do tabel |
| 2. Dodanie części logiki aplikacji | 2A. Dodała funkcjonalność sprawdzania poprawności odpowiedzi |
| 2B. Stworzyła obsługę końca gry |
| ZIELEŹNY Sonia | 1. Stworzenie warstwy modelu z dostępem do danych | 1A. Dodała klasę DataAccess z metodami wyciągającymi dane z bazy MySql |
| 1B. Stworzyła pomocniczą klasę Question |
| 2. Dodanie menu i połączeń między widokiem a modelem widoku | 2A. Dodała rozwijane menu do interfejsu użytkownika |
| 2B. Połączyła część elementów okna z interfejsem ICommand |
| BUGAJSKI Maciej | 1. Zaprojektowanie interfejsu użytkownika i zasobów gry | 1A. Stworzył warstwę widoku, zaprojektował podział okna na kontrolki użytkownika |
| 1B. Opracował grafikę interfejsu |
| 1C. Stworzył podkład muzyczny na bazie oryginalnej melodii z Familiady |
| 2. Stworzenie zamysłu architektury projektu | 2A. Dokonał podziału na pliki według wzorca MVVM |

**1. Opis projektu.**

Za projekt przyjęliśmy stworzenie gry okienkowej „Familiada” wzorowanej na faktycznym teleturnieju. Użytkownik rozgrywa 5 tur polegających na wpisywaniu odpowiedzi jego zdaniem najlepiej pasujących do wyświetlonego pytania i zatwierdzaniu ich przyciskiem. Jeśli dana odpowiedź jest poprawna, wyświetli się na tablicy, a jeśli nie, na tablicy pojawi się krzyżyk świadczący o błędzie. Kolejna runda rozpoczyna się w momencie zgadnięcia wszystkich poprawnych odpowiedzi, dobiegnięcia do końca czasu przeznaczonego na jedną rundę lub udzielenia trzech odpowiedzi błędnych.

**2. Wymagania**

Wymagania funkcjonalne:

* Pytania, odpowiedzi i żarty są pobierane z bazy danych i wyświetlane w oknie użytkownika
* Użytkownik ma możliwość rozpoczęcia nowej gry w dowolnym momencie
* Użytkownik ma dostęp do instrukcji i spisu najlepszych wyników w dowolnym momencie
* Użytkownik może włączać i wyłączać muzykę w tle
* Aplikacja zapisuje najlepsze wyniki do pliku tekstowego
* Rundy rozgrywki zmieniają się przy poprawnym uzupełnieniu wszystkich pól, popełnieniu trzech błędów lub upłynięciu minuty

Wymagania niefunkcjonalne:

* Aplikacja zapewnia rozrywkę użytkownikowi
* Aplikacja jest łatwa w obsłudze i przejrzysta dla użytkownika

**3. Przebieg realizacji**

* Zaprojektowanie wzorca MVVM
* Zaprojektowanie okna
* Połączenie z bazą danych
  + Klasa Question
  + Warstwa dostępu do bazy danych MySQL
* Stworzenie menu
  + Przechowywanie nazwy drużyny
  + Gra rozpoczynająca się dopiero po kliknięciu guzika
* Stworzenie funkcjonalności sprawdzania odpowiedzi
  + Klasa BoardViewModel

Zawiera metodę pobierającą listę poprawnych odpowiedzi i punktów do nich przypisanych oraz metodę z odpowiedziami, które będą wyświetlane podczas rozgrywki. Metoda ustawia wszystkie odpowiedzi na ‘--------------’.

* + MainWindowViewModel

W tej klasie sprawdzana jest poprawność odpowiedzi. Jedna poprawna odpowiedź to ciąg wielu jej wariantów (jedną odpowiedź można wyrazić w wielu różnych słowach). Przy wprowadzeniu słowa sprawdzany jest warunek, czy odpowiedź zawiera się w tym ciągu. Następnie zliczane są punkty, a wyświetlana odpowiedź z ‘--------’ zmienia się na odpowiedź wprowadzoną.

* Koniec gry
  + MainWindowViewModel

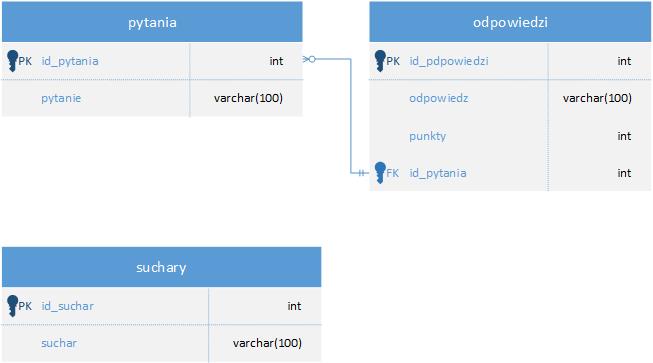
W tej klasie również naliczane są utraty jeśli wprowadzone słowo nie znajdzie się w ciągu z poprawnymi odpowiedziami. Sprawdzany jest zatem warunek czy nastąpiły trzy utraty, a następnie czy wszystkie rundy już zostały zakończone, jeśli nie, następuje wywołanie funkcji, losującej kolejne pytanie . Jeśli wszystkie rundy zostały zakończone, pojawia się kontrolka na której wyświetlana jest nazwa drużyny oraz punkty zdobyte przez nią podczas całej gry. Wywołana zostaje funkcja zapisująca wyniki do pliku.

* Zapisywanie punktów
  + MainWindowViewModel

Nazwa drużyny i wyniki zapisywane są do pliku tekstowego

* Wyświetlanie najlepszych wyników

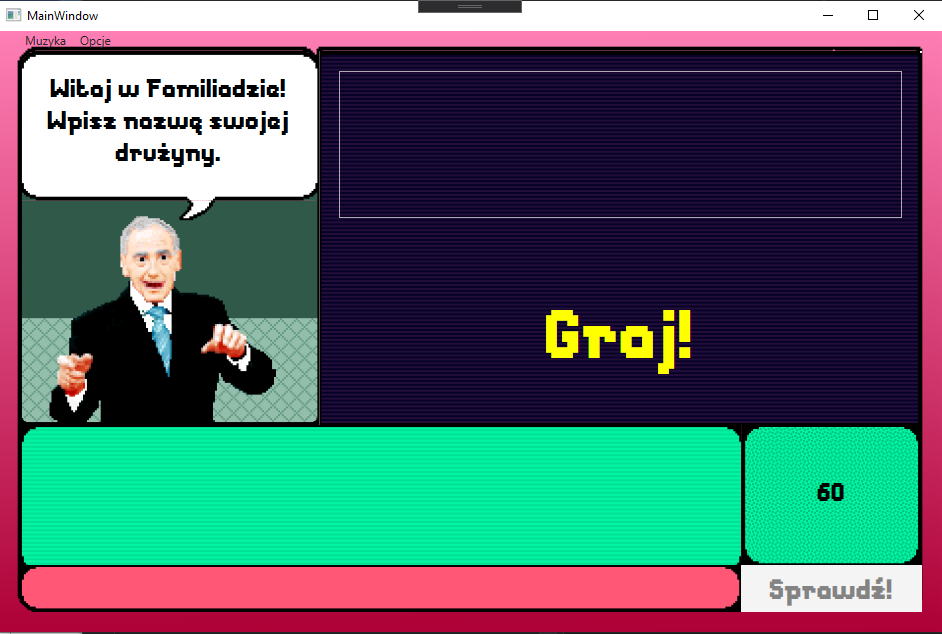
Schemat bazy danych:



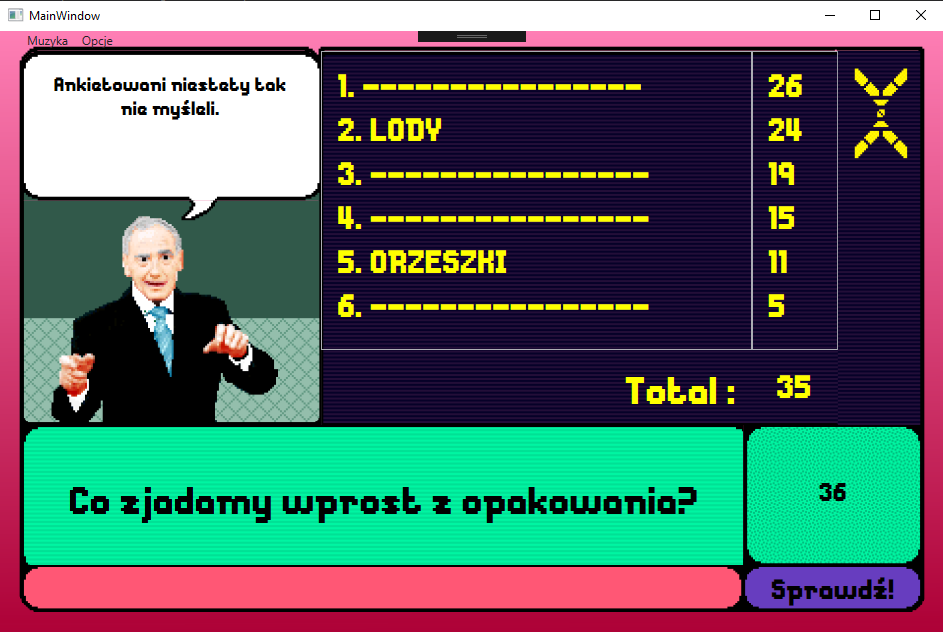
**4. Instrukcja użytkownika**

Opis działania stworzonej aplikacji ze zrzutami ekranów ilustrujące sposób działania aplikacji.

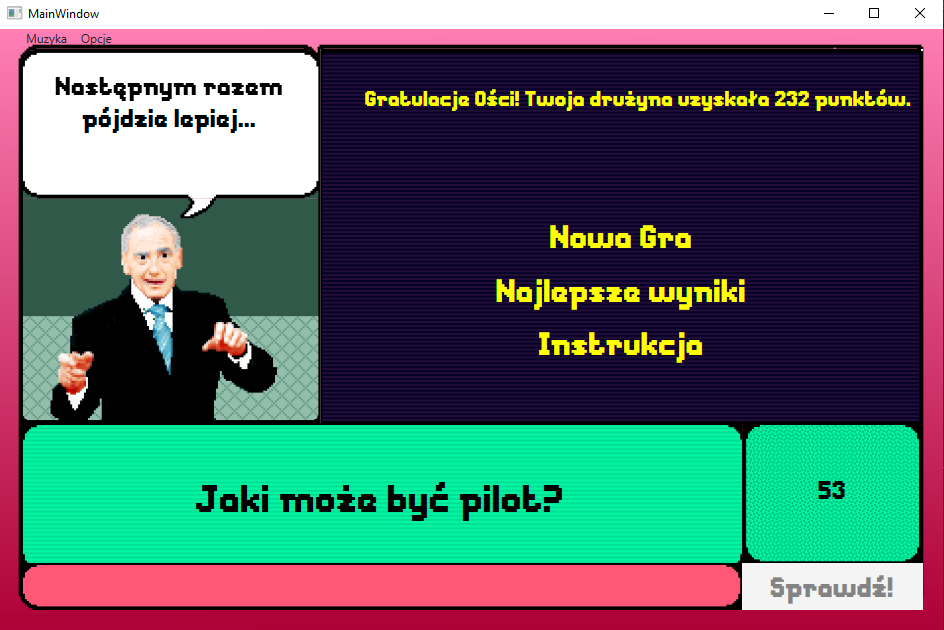
Po uruchomieniu aplikacji użytkownik musi wpisać nazwę swojej drużyny i nacisnąć przycisk ‘Graj’, aby przejść do gry.



Po wpisaniu nazwy drużyny pojawia się pytanie i tablica z możliwą liczbą odpowiedzi, które należy wpisywać w polu do tego przeznaczonym znajdującym się pod polem z pytaniem. Obok znajduje się przycisk sprawdź, który weryfikuje, czy odpowiedź została wprowadzona poprawnie. Jeśli podana odpowiedź jest prawidłowa, pojawia się na tablicy i doliczane są punkty. Jeśli odpowiedź nie jest podana prawidłowo, pojawia się krzyżyk sygnalizujący utratę szansy na nieprawidłową odpowiedź. Użytkownik ma trzy takie szanse. Przejście do następnej rundy następuje po wykorzystaniu wszystkich szans na złą odpowiedź lub po upływie określonego czasu na wpisanie odpowiedzi, który widnieje w polu obok pytania.



W grze jest 5 rund rozgrywanych na tej samej zasadzie. Kiedy zostaną rozegrane wszystkie, pojawia się informacja o zdobytej liczbie punktów przez drużynę użytkownika oraz przyciski umożliwiające rozpoczęcie nowej gry, wyświetlenie najlepszych wyników oraz instrukcji.



**5. Podsumowanie i wnioski.**

Największymi problemami, z jakimi zetknęliśmy się w przypadku tego projektu były:

* Korzystanie ze współdzielonej bazy danych
* Zmiana rundy w momencie upływu wyznaczonego czasu (Użycie klasy DispatcherTimer zamiast Timer)
* Odczekanie dwóch sekund pomiędzy sygnalizacją trzeciego błędu a rozpoczęciem nowej rundy (Użycie funkcji asynchronicznej)

Dalsze kierunki rozwoju aplikacji:

* Możliwość prowadzenia rozgrywki przez internet pomiędzy dwiema drużynami
* Zwiększenie ilości pytań
* Dodanie dźwięków przy naciskaniu guzików

**6. Dodatek - udokumentowanie wykorzystania systemu kontroli wersji.**

Wykorzystanym przez nas systemem kontroli wersji jest platforma Github.

<https://github.com/BugiMa/Familiada_POiG_BD>

