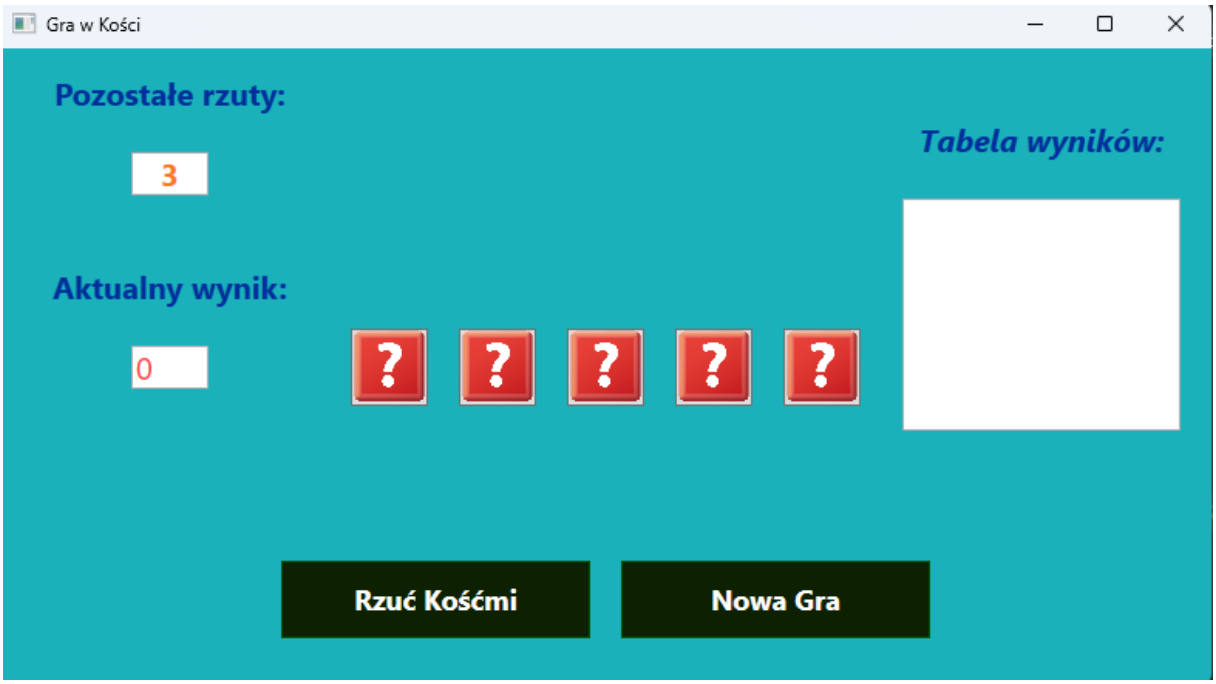
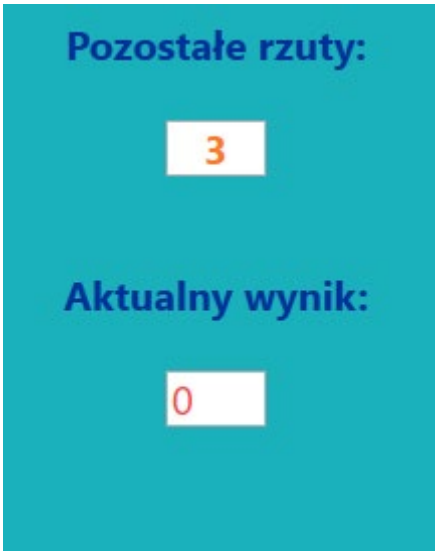
Stworzenie aplikacji gry w kości w WPF

Cel zadania: Stwórz aplikację WPF (Windows Presentation Foundation), która będzie symulować prostą grę w kości, składającą się z 5 kostek. Aplikacja powinna umożliwiać rzut kośćmi, zatrzymywanie wybranych kości oraz obliczanie wyniku na podstawie wartości wyrzuconych kości.



Obrazek1



Obrazek 2

Instrukcja krok po kroku:

1. **Utwórz nowy projekt:**
    - Utwórz nowy projekt typu **WPF App (.NET Framework)** w Visual Studio.
    - Nazwij projekt GraWKosci.
2. **Dodaj obrazy kości:**
    - W folderze projektu GraWKosci utwórz nowy folder o nazwie images.
    - Umieść w tym folderze obrazy dla poszczególnych wartości na kości (pliki graficzne 1.png, 2.png, 3.png, 4.png, 5.png, 6.png) oraz obrazek question.png jako startowy obrazek dla kości.
3. **Zaprojektuj interfejs użytkownika (MainWindow.xaml):**
    - Ustaw tło okna na kolor #FF1AB1BB.
    - Stwórz w centralnej części okna 5 przycisków (Button) o nazwach btnDice1, btnDice2, btnDice3, btnDice4 oraz btnDice5, które będą reprezentować kości. Każdy przycisk powinien zawierać obraz (Image) o nazwie question.png jako początkowy obrazek.
    - Umieść pod przyciskami dwa główne przyciski akcji:
        - Przycisk o nazwie „Rzuć Kośćmi” do wykonania rzutu wszystkimi kośćmi.
        - Przycisk o nazwie „Nowa Gra” do rozpoczęcia nowej gry.
    - Dodaj etykiety (Label) i pola tekstowe (TextBox):
        - Pole tekstowe do wyświetlania liczby pozostałych rzutów.
        - Pole tekstowe do wyświetlania aktualnego wyniku.
    - Dodaj ListBox o nazwie do wyświetlania tabeli wyników.
4. **Utwórz funkcjonalność gry (MainWindow.xaml.cs):**
    - W pliku MainWindow.xaml.cs zaimplementuj następującą logikę:
        - Przypisanie losowych wartości od 1 do 6 do kości, które nie są zatrzymane.
        - Zatrzymanie wybranej kości (zmiana przezroczystości oraz wyłączenie przycisku).
        - Resetowanie gry (wszystkie kości wracają do stanu początkowego).
        - Aktualizowanie obrazów kości na podstawie ich wartości.
        - Obliczanie końcowego wyniku po 3 rzutach.
        - Aktualizowanie bieżącego wyniku po każdym rzucie.

5. **Szczegółowe wymagania funkcjonalne:**
    - **Liczba rzutów**: Gracz może rzucić kośćmi maksymalnie 3 razy w jednej rundzie.
    - **Zatrzymywanie kości**: Gracz może zatrzymać wybrane kości, klikając na kość, co powoduje wyłączenie przycisku oraz zmianę przezroczystości na Opacity = 0.5.
    - **Wyświetlanie wyniku**: Pole wyświetla sumę wartości na kościach po każdym rzucie.
    - **Tabela wyników**: Po 3 rzutach wynik końcowy zostaje dodany do ListBox
6. **Stylizacja i dodatkowe wymagania:**
    - Przyciski kości powinny mieć:
        - Szerokość i wysokość: 50px
        - Margines: 10px
    - Przycisk „Rzuć Kośćmi” oraz „Nowa Gra” powinny mieć:
        - Szerokość: 200px
        - Wysokość: 50px
        - FontSize: 18
        - Tło: #FF0D2000
        - Kolor tekstu: White
    - Etykiety (Label) oraz pola tekstowe (TextBox) powinny być odpowiednio wystylizowane zgodnie obrazkiem 2
7. **Sprawdzanie poprawności:**
    - Uruchom aplikację.
    - Sprawdź, czy po kliknięciu przycisku Rzuć Kośćmi wartości kości są losowane i wyświetlane.
    - Sprawdź, czy po kliknięciu kości zostaje ona zatrzymana.
    - Sprawdź, czy po 3 rzutach wynik końcowy jest obliczany i wyświetlany w ListBox.
8. **Przykładowy interfejs użytkownika:**
    - Interfejs użytkownika powinien wyglądać podobnie jak przedstawiony obrazek1
    - Upewnij się, że wszystkie elementy interfejsu są prawidłowo rozmieszczone i stylizowane.
9. **Kryteria oceny:**
    - Poprawność działania funkcji Rzuć Kośćmi i zatrzymywania kości.
    - Prawidłowe obliczanie wyniku i dodawanie do listy wyników po zakończeniu rundy.
    - Prawidłowe stylizowanie przycisków i elementów interfejsu zgodnie z wymaganiami.
    - Zgodność interfejsu z podanym wyglądem oraz płynność działania aplikacji.