

Projekt aplikacji symulującej popularną grę hazardową. Gra polega na obstawieniu przez gracza sumy pieniędzy i "wyciągnięciu" ich w momencie, w którym licznik (który jest jednocześnie mnożnikiem postawionej sumy) dojdzie do dogodnego według gracza momentu przed zakończeniem gry. Licznik zaczyna zwiększać się od zera a moment zakończenia gry jest losowy.

Baza danych tworzy się lokalnie przy pierwszym zalogowaniu. Do bazy danych została użyta biblioteka Microsoft Entity Framework Core.

Przestrzeń nazw Constants

Statyczne klasy ze stałymi

SettingConstants – Klasa zawierająca stałe ustawienia programu

- HIGH_SCORE_GAME_MAX_AMOUNT – Maksymalna ilość gier uznawanych za rekord
- INITIAL_BALANCE - Początkowa ilość salda nowo tworzonych graczy

Przestrzeń nazw Data

Klasy powiązane z bazą danych

CrashAppContext – Klasa dziedzicząca po DbContext biblioteki EntityFrameworkCore.

- Zawiera ona kolekcję obiektów DbSet dla wszystkich encji stanowiących punkty dostępowe do tabel bazodanowych każdej z nich
- Metoda OnConfiguring – Metoda ustawiająca connection stringa do bazy danych
- Metoda OnModelCreating – Metoda ustawiająca relację między encjami, na których podstawie EF Core jest w stanie wygenerować bazę danych

StaticContext – Statyczna klasa zawierająca instancję klasy CrashAppContext (jest to zrobione, żeby w programie był tylko 1 punkt dostępowy do bazy danych)

- Statyczny konstruktor, który tworzy bazę danych metodą EF Core Database.EnsureCreated()

Przestrzeń nazw Entities

Klasy encji bazodanowych (klasy reprezentujące tabele bazodanowe 1:1)

Entity Base – Klasa bazowa dla wszystkich encji, zawiera id tabel bazodanowych

Contact – Encja zawierająca dane personalne gracza

Game – Encja gry zawierająca dane o rozgrywanych grach przez gracza

HighScore – Encja rekordu zawierająca odniesienie do gry oraz datę, kiedy takowy został ustanowiony

Player – Encja gracza zawierająca saldo oraz dane logowania

Przestrzeń nazw Models

Klasy podobnie jak encje zawierające tylko dane, służące do transefru danych w logice a aplikacji. Zazwyczaj są zwracane przez klasy serwisów

FinishedGameResult – Klasa zawierająca informacje o zakończonej rozgrywce

Właściwość Game – Odniesienie do encji gry

Właściwość IsNewHighScore – Informacja o tym, czy pobito rekord

Właściwość HighScorePlace – Informacja o miejscu danej gry (nawet, jeśli wychodzi poza dopuszczalną liczbę gier rekordowych)

HighScoreGridRowResult - Klasa do wyświetlenia przez kontrolkę WPF DataGrid

Właściwość Number – Numerek wiersza

Właściwość PlayerDescription - Właściwość tylko do odczytu zwracająca imię oraz nazwisko gracza

Właściwość HighScore – Referencja do encji HighScore

Właściwość IsHighlighted – Informacja o tym, czy wiersz powinien być wyświetlony (jeśli jest to rekord aktualnie zalogowanego gracza)

InitialPlayerResult - Klasa rezultatu tworzenia gracza domyślnego

Właściwość InitialPlayer – Referencja do encji nowo utworzonego gracza

Właściwość WasInitialPlayerCreated - Właściwość tylko do odczytu z informacją, czy został utworzony domyślny gracz

SignInResult - Klasa rezultatu logowania

Właściwość SignInSuccess – Informacja o tym, czy logowanie się powiodło

Właściwość SignedInPlayer - Referencja do zalogowanego gracza

Przestrzeń nazw Services

Klasy serwisów z logiką aplikacji

GameService - Najważniejszy serwis w aplikacji zawierający logikę rozgrywki oraz zapis ich danych.

Pole _highScoreService – Pole serwisu

Pole _multiplierInterval – Pole interwału timera _multiplierTimer

Pole _millisecondsToStopTheGame – Pole przechowujące wylosowany czas w milisekundach, po upływie których gra zostanie zakończona z rezultatem niepowodzenia

Pole _currentMillisecondsPassed – Aktualny czas od rozpoczęcia gry w milisekundach

Pole _multiplierTimer – Timer odpowiedzialny za mierzenie przebiegu rozgrywki

Pole _multiplier - Pole przechowujące mnożnik gry, który zwiększa się wraz z upływem czasu

Zdarzenie OnMultiplierChange – Zdarzenie informujące o zmianie licznika. Przekazuje w parametrach nową wartość mnożnika oraz możliwej wygranej

Zdarzenie OnGameFinish – Zdarzenie informujące o zakończonej grze. Przekazuje w argumencie obiekt klasy FinishedGameResult

Pole _player – Obiekt encji Player, która jest aktualnie zalogowanym graczem

Pole _bet - Pole przechowujące początkową stawkę rozgrywki

Pole _isGameRunning – Pole flaga przechowujące informacje, czy gra aktualnie się toczy

Konstruktor GameService – Przyjmuje argument aktualnie zalogowanego gracza. Inicjalizuje _multiplierTimer i podpiną obsługę do jego zdarzenia Tick

Metoda StartNewGame – Metoda rozpoczynająca nową grę. Przyjmuje stawkę początkową rozgrywki. Włącza multiplierTimer

Metoda StopCurrentGameAsync – Przerywa aktualną rozgrywkę triggerując zdarzenie OnGameFinish

Metoda SaveGameAsync – Przyjmuje jako argument encję Game rozgrywki. Zapisuje przekazaną grę. Sprawdza, czy gra pobiła rekord i zapisuje ją do tabeli rekordów poprzez obiekt serwisu _highScoreService, jeśli to prawda.

Metoda _timer_Tick – Metoda obsługująca zdarzenie Tick timera multiplierTimer. Podnosi wartość mnożnika co każdy tick, triggeruje zdarzenie OnMultiplierChange oraz, jeśli wylosowany czas zakończenia rozgrywki dobiegnie końca, kończy ją triggerując zdarzenie OnGameFinish

Metoda CreateFinishedGameResultAsync – Metoda tworząca nowy obiekt encji Game. Zwraca obiekt modelu FinishedGameResult

HighScoreService – Serwis obsługi rekordowych gier. Jego obiekt jest zagnieżdżony w serwisie GameService

Metoda SaveGameAsHighScoreAsync – Jako argument przyjmuje encję Game. Sprawdza, czy powinna usunąć grę z tabeli rekordowych gier, jeśli zostanie przekroczony limit i zapisuje grę jako rekordowa gra

Metoda GetGameHighScorePlaceAsync – Metoda zwracająca miejsce rozegranej gry spośród rekordowych gier

Metoda GetAllHighScoresAsync – Metoda zwracająca listę wszystkich rekordowych gier

PlayerService – Serwis obsługi tworzenia graczy

Metoda CreateInitialUserIfNeededAsync – Metoda tworząca domyślnego gracza, jeśli nie znajdzie żadnych graczy w bazie danych

Metoda CreateNewPlayerAsync – Metoda tworząca i zapisująca w bazie nowy obiekt encji gracza Player

SignInService – Serwis logowania do aplikacji

Metoda SignInAsync – Metoda sprawdzająca zgodność loginu i hasła. W zależności od rezultatu zwraca obiekt modelu SignInResult z uzupełnionym lub nie obiektem encji gracza Player

Przestrzeń nazw Windows

Okna programu:

SignInWindow – Okno logowania do aplikacji. Jest też w nim możliwość do przejścia do okna NewPlayerService.

NewPlayerWindow – Okno tworzenia nowego gracza. Po jego zamknięciu, dane nowo stworzonego gracza zostaną automatycznie wprowadzone do pól tekstowych loginu oraz hasła

GameWindow – Okno profilu gracza oraz gry. Jest w niej możliwość zagrania nowej gry, opcja wylogowania i powrotu do SignInWindow oraz opcja przejścia do HighScoreWindow

HighScoreWindow – Okno gier rekordowych. Gry należące do zalogowanego gracza zostaną podświetlone na kolor jasno-niebieski