Dokumentacja techniczna

Tytuł: Fall of the Sun
Opis: Gra strategiczno-turowa w unity Autorzy: Łukasz Sędkowski, Hubert Rohnka

Technologie:
- API .net 8.0
- Frontend: unity 6000.0.27f1
- Baza Danych: MySQL
- Microsoft IdentityModel.Tokens (JWT)

Instrukcja uruchomienia:
- Gra musi zostać uruchomiona w Unity (wersja 6000.0.27f1), ponieważ projekt wymaga przesłania nieskompilowanej wersji.
- API musi zostać uruchomione lokalnie w aplikacji Visual Studio 2022 i skompilowane, ponieważ nie posiadamy serwera który kompiluje C#/.NET.
- Wersja gry jest skonfigurowana do komunikacji z API
- Krok 1: Pobranie plików:
- Pobierz pliki z GIT.
- Rozpakuj projekt w wybranym folderze na swoim komputerze.
- Upewnij się, że masz dostęp do wszystkich niezbędnych plików projektu (foldery: Assets, ProjectSettings, Packages).

Krok 2: Instalacja Unity (jeśli nie jest zainstalowane):
- Pobierz i zainstaluj Unity Hub z oficjalnej strony Unity.
- W Unity Hub przejdź do zakładki Instalacje i kliknij Dodaj.
- Wybierz wersję 6000.0.27f1 i zainstaluj ją. Jeśli nie ma tej wersji na liście, wybierz Dodaj niestandardową wersję i podaj ścieżkę do instalatora wersji Unity 6000.0.27f1.

Krok 3: Otwieranie projektu w Unity:
- Uruchom Unity Hub.
- Kliknij Otwórz projekt i wskaż folder, w którym rozpakowałeś pliki gry.
- Poczekaj, aż Unity zaimportuje projekt (może to potrwać kilka minut).

Krok 4: Uruchomienie API:
- Otwórz Visual Studio 2022.
- Otwórz projekt API znajdujący się w katalogu API_Fall_Of_The_Sun.
- Kliknij Run lub naciśnij F5, aby skompilować i uruchomić API lokalnie.
- Po uruchomieniu API powinno być dostępne pod adresem https://localhost:7188/swagger.

Krok 5: Uruchomienie gry:
- Upewnij się, że API działa w Visual Studio.
- W Unity kliknij Play w górnej środkowej części ekranu.
- Gra uruchomi się od sceny MainMenu.

Krok 6: Sterowanie:
- Kliknij lewym przyciskiem myszy (LPM) na pionek, aby go wybrać.
- Kliknij LPM na pole na planszy, aby przenieść pionek, jeśli jest to legalny ruch.
- Podświetlenie możliwych pól ruchu po wybraniu pionka (np. na żółto) – HighlightPossibleMoves().
- Po kliknięciu w pionek przeciwnika, który znajduje się w zasięgu ataku, następuje atak.
- Sprawdzenie zasięgu ataku i przeszkód (IsObstacleBetween()).
- Redukcja punktów życia przeciwnika o wartość ataku pionka (AttackEnemyPiece()).
- Naciśnięcie klawisza Q przełącza na kolejną drużynę.
- Po zmianie tury resetują się punkty ruchu pionków drużyny (ResetMovementRangeForTeam()).

- Naciśnięcie klawiszy 1-9 wybiera pionek o odpowiadającym ID.
- Naciśnięcie klawisza Spacja aktywuje specjalną zdolność (np. leczenie przez Kapłankę Priestess).

Wzorzec architektoniczny: Projekt korzysta z adaptacji wzorca Model-View-Controller (MVC), gdzie API pełni rolę Modelu, interfejs Unity jest Widokiem, a logika gry i skrypty Unity pełnią funkcję Kontrolera, dostosowanego do specyfiki silnika Unity

Opis struktury projektu:

Projekt składa się z dwóch głównych części:

API:

Odpowiada za zarządzanie użytkownikami, rejestracją, uwierzytelnianiem za pomocą tokenów JWT, przechowywanie danych graczy w bazie oraz udostępnianie endpointów do komunikacji z grą.

Gra:

Gra w Unity, wykorzystująca rozgrywkę wieloosobową w systemie strategiczno-turowym z kilkoma drużynami i umożliwia rejestrację/logowanie użytkowników poprzez integrację z API.

Modele API:

User: reprezentuje użytkownika w systemie. Model służy do zarządzania danymi użytkownika, w tym danymi logowania i profilu.

Pole	Typ	Opis	Ograniczenia
UserId	Int	Klucz główny, unikalny identyfikator użytkownika	Klucz główny, autoinkrement
Username	String	Unikalny login użytkownika	Wymagane, unikalne, max 50
PasswordHash	String	Hasło zaszyfrowane za pomocą BCrypt	Wymagane, max 256
Email	String	Adres email	Wymagane, unikalne, max 100.
IsEmailConfirmed	Bool	Czy email został potwierdzony	Domyślnie false
CreatedAt	DateTime	Data rejestracji użytkownika.	Domyślnie DateTime.UtcNow

RegisterRequest: Model używany podczas rejestracji użytkownika.

Pole	Typ	Opis	Ograniczenia
Username	string	Nazwa użytkownika	Wymagane
Email	string	Adres email	Wymagane
Password	string	Hasło użytkownika	Wymagane
RepeatPassword	string	Powtórzenie hasła	Wymagane

HallOfFame: Tabela przechowująca wyniki graczy, w tym liczbę zabitych jednostek, strat oraz całkowity wynik punktowy.

Pole	Typ	Opis	Ograniczenia
Id	Int	Unikalny identyfikatr wpisu	Klucz główny, autoinkrement
UserId	Int	Id użytkownika (odwołanie do tabeli Users)	Wymagane
Kills	Int	Liczba zabitych jednostek	Domyślnie 0
Deaths	Int	Liczba straconych jednostek	Domyślnie 0
TotalScore	Int	Całkowity wynik gracza	Domyślnie 0

LoginRequest: Model używany podczas logowania użytkownika.

Pole	Typ	Opis
Email	String	Wymagane
Password	String	Wymagane

ChangePasswordRequest: Model umożliwiający zmianę hasła użytkownika.

Pole	Typ	Opis
CurrentPassword	String	Wymagane
NewPassword	String	Wymagane

Wylistowane kontrolery:

1.POST /api/auth/register: rejestruje nowego użytkownika .

Parametry:

username (string) – Nazwa użytkownika (wymagane)
email (string) – Adres email (wymagane)
password (string) – Hasło użytkownika (wymagane)

Tworzy nowego użytkownika w systemie, zapisuje go w bazie danych i wysyła link aktywacyjny na podany email.

Zwraca:

Sukces: 201 Created w przypadku sukcesu.
Błąd: 400 Bad Request w przypadku błędnych danych wejściowych.

2.POST /api/auth/login: umożliwia użytkownikowi logowanie i zwraca token JWT.

Parametry:

email (string) – Adres email (wymagane)
password (string) – Hasło użytkownika (wymagane)

Sprawdza podane dane logowania, a w przypadku powodzenia generuje token JWT z danymi użytkownika.

Zwraca:

Sukces: OK z tokenem JWT.
Błąd: Unauthorized w przypadku błędnych danych.

3.POST /api/auth/reset-password: Wysyła link resetujący hasło na adres email.

Parametry:

email (string) – Adres email (wymagane)

Weryfikuje istnienie użytkownika w bazie danych i wysyła token resetujący hasło.

Zwraca:

200 OK po pomyślnym wysłaniu wiadomości.

404 Not Found jeśli użytkownik nie istnieje.

4.GET /api/users/{userId}: Pobiera dane użytkownika na podstawie userId.

Parametry:

userId (int) – Identyfikator użytkownika (wymagane)

Zwraca dane profilu użytkownika.

Zwraca:

200 OK z danymi użytkownika (userId, username, email, avatarId).
404 Not Found jeśli użytkownik nie istnieje.

5. GET /api/stats/halloffame: wraca listę użytkowników oraz ich wyniki posortowane malejąco po punktach.

Parametry:

Brak

Łączy dane z tabeli Users oraz HallOfFame i zwraca wyniki w formie rankingu.

Zwraca:

200 OK z listą użytkowników i ich wynikami.
404 Not Found jeśli brak wyników w Hall of Fame

Opis systemu użytkowników Role w systemie:

Goście:

Mogą grać lokalnie bez konieczności rejestracji i logowania
Wyniki ich gier nie są zapisywane w systemie, a ich statystyki nie są uwzględniane w rankingu.

Funkcjonalność: Gra lokalna, Rejestracja

Zarejestrowani użytkownicy:

Posiadają konto z przypisanym loginem i emailem
Mogą brać udział w rankingach oraz ich statystyki są zapisywane na serwerze
Mają dostęp do rankingu globalnego, możliwość rywalizacji z innymi graczami.

Funkcjonalność: Udział w rankingach, Zapisywanie wyników, Logowanie, Reset hasła

Użytkownicy są klasyfikowani na podstawie statusu:

Gość: Domyślny status gracza, który uruchamia grę bez logowania.

Zarejestrowany użytkownik: Status uzyskany po rejestracji i zalogowaniu

Kategoria	Powiązane z użytkownikiem	Globalne Dane
Konto	Login, email, hasło	
Statystyki	Liczba rozegranych gier, wygrane, przegrane, zabite jednostki	Ranking globalny graczy,
Historia gier	Szczegóły gier użytkownika (np. wynik, data)	Ogólna liczba zagranych gier

Najciekawsze funkcjonalności projektu:

1. Dynamiczna generacja planszy, plansza jest generowana proceduralnie przy każdej nowej grze. Przeszkody, jednostki i elementy terenu są rozmieszczane losowo, co sprawia, że każda rozgrywka jest unikalna i wymaga od gracza elastycznego podejścia strategicznego.
2. Sala Chwały (Hall of Fame), system rankingu graczy pozwala na przeglądanie najlepszych wyników użytkowników. Dane są pobierane z API i wyświetlane dynamicznie w grze w dedykowanej zakładce. Wyniki są sortowane malejąco według zdobytych punktów, co umożliwia śledzenie najlepszych graczy.
3. Powiadomienia e-mailowe, po rejestracji użytkownik otrzymuje e-mail z linkiem potwierdzającym adres. Użytkownicy mogą także otrzymywać powiadomienia o ważnych wydarzeniach, np. aktualizacjach rankingów czy resetach hasła.