

Proof of concept. Podstawowa komunikacja i sesja gry dla dwóch graczy

Głównym celem tego PoC jest udowodnienie, że jesteśmy w stanie nawiązać stabilną komunikację klient-serwer w czasie rzeczywistym oraz zaimplementować podstawową logikę gry dla dwóch graczy, co jest kluczowe dla sukcesu całej platformy gier.

Kluczowe elementy do zrealizowania:

1. Rejestracja i logowanie użytkownika:
 - a. Użytkownik A rejestruje się i loguje.
 - b. Użytkownik B rejestruje się i loguje.
 - c. To potwierdzi działanie podstawowego uwierzytelniania i autoryzacji oraz interakcji z bazą danych (przechowywanie danych użytkowników).
2. Komunikacja Klient-Serwer w czasie rzeczywistym (WebSockets):
 - a. Uruchomienie serwera opartego na Spring Framework.
 - b. Ustanowienie połączenia WebSocket między dwoma klientami a serwerem.
 - c. Testowanie wysyłania prostych wiadomości między klientami za pośrednictwem serwera. To jest absolutnie kluczowe dla gier wieloosobowych.
3. Podstawowa logika gry Kółko i Krzyżyk:
 - a. Implementacja minimalnej logiki gry "Kółko i Krzyżyk" na serwerze.
 - b. Umożliwienie dwóm zalogowanym użytkownikom dołączenia do jednej "sesji" gry.
 - c. Przesyłanie ruchów graczy (np. kliknięcia w pole) od klienta do serwera za pomocą WebSockets.
 - d. Serwer przetwarza ruch i wysyła aktualizację stanu gry do obu klientów.
 - e. Klienci wizualizują zmieniony stan gry (np. pojawienie się "X" lub "O" na planszy).
 - f. Nie jest konieczne implementowanie pełnej logiki wygrywania/przegrywania, remisu, czy pełnego UI – skupiamy się na samym przepływie danych.
4. Minimalny interfejs użytkownika:
 - a. Okno logowania/rejestracji.
 - b. Proste okno gry "Kółko i Krzyżyk" z planszą i możliwością wykonywania ruchów (wystarczy kliknięcie w pole, nie trzeba stylizacji czy skomplikowanych animacji).
 - c. Celem jest tylko wizualizacja komunikacji i stanu gry.

Uzasadnienie wyboru PoC:

- Najwyższe ryzyko techniczne: Komunikacja w czasie rzeczywistym między klientami za pośrednictwem serwera (WebSockets) jest często najbardziej złożonym i ryzykownym elementem w aplikacjach wieloosobowych. Potwierdzenie jej działania jest kluczowe.
- Weryfikacja architektury: Testuje podstawowy model klient-serwer z podziałem ról (front-end wysyła, back-end przetwarza i wysyła dalej).
- Integracja kluczowych komponentów: Łączy GUI, logikę biznesową i komunikację sieciową.
- Szybka realizacja: Skupia się na absolutnym minimum, aby potwierdzić wykonalność, bez wdrażania wszystkich funkcjonalności. Nie musimy tworzyć wszystkich gier, wszystkich funkcji profilu, czy zaawansowanego wsparcia językowego.

Oczekiwane rezultaty PoC:

- Działająca aplikacja kliencka, która pozwala na rejestrację i logowanie.
- Działający serwer, który zarządza sesjami użytkowników.
- Możliwość nawiązania połączenia WebSocket między dwoma klientami a serwerem.
- Dwóch graczy może jednocześnie grać w Kółko i Krzyżyk, widząc swoje ruchy w czasie rzeczywistym (bez pełnej logiki zakończenia gry).