

# Dokumentacja - Gra Blackjack Sieciowy

## Opis projektu

Sieciowa implementacja gry Blackjack w architekturze klient-serwer napisana w języku C. Projekt umożliwia graczom logowanie się, rejestrację oraz granie w Blackjaka z systemem tokenów i punktacji.

## Mechanizmy i protokoły

### Protokół komunikacji

- **TCP/IP** - niezawodna komunikacja między klientem a serwerem
- **Port**: 12345
- **Format komunikatów**: tekstowy z terminatorami linii `\n`

### Bezpieczeństwo

- **SHA-256** - hashowanie haseł użytkowników (biblioteka OpenSSL)
- **Przechowywanie**: hashe zapisywane w pliku `users.txt`

### Architektura serwera

- **Fork-based** - każdy klient obsługiwany w osobnym procesie
- **Signal handling** - `SIGCHLD` ignorowany, automatyczne sprzątanie procesów zombie
- **Persistent storage** - dane użytkowników w pliku tekstowym

### Randomizacja

- **PRNG**: `rand()` z seedem `time(NULL) + getpid()`
- **Tasowanie kart**: algorytm Fisher-Yates

## Jak działa

### Proces połączenia

1. **Nawiązanie połączenia** - klient łączy się z serwerem TCP
2. **Autoryzacja** - wybór logowania (l) lub rejestracji (r)
3. **Weryfikacja** - sprawdzenie danych w bazie użytkowników
4. **Rozpoczęcie gry** - wyświetlenie stanu konta gracza

### Rozgrywka

1. **Obstawianie** - gracz stawia zakład ( $1 \leq \text{zakład} \leq \text{tokeny}$ )
2. **Rozdanie kart**:
  - Gracz: 2 karty odkryte
  - Krupier: 1 karta odkryta, 1 zakryta
3. **Tura gracza** - wybory "h" (hit) lub "s" (stand)
4. **Tura krupiera** - automatyczne dobieranie do 17+ punktów
5. **Rozstrzygnięcie**:
  - Wygrana: 2x zakład + 2 punkty score
  - Remis: zwrot zakładu
  - Przegrana: utrata zakładu

### Zasady Blackjaka

- **Wartości kart**: A=11, K/Q/J=10, pozostałe=nominał
- **Bust (przegrana)**: przekroczenie 21 punktów
- **Krupier**: musi dobierać do 17+ punktów
- **Wygrana**: gracz ma więcej punktów niż krupier lub krupier przekroczy 21

### Struktura plików

users.txt	# baza użytkowników
server.c	# kod serwera
client.c	# kod klienta
README.md	# dokumentacja projektu

### Kompilacja i uruchomienie

#### Wymagania

- **System**: Linux/Unix
- **Kompilator**: GCC
- **Biblioteki**: OpenSSL (dla klienta)

#### Instalacja OpenSSL (Ubuntu/Debian)

```
sudo apt-get install libssl-dev
```

## Kompilacja

Za pomocą Makefile:

```
make
```

## Uruchomienie

```
# Uruchomienie serwera
./server

# Uruchomienie klienta
./client 127.0.0.1
```

## Przykład użycia

Rejestracja nowego użytkownika

```
login or register (l/r)?
r
user:
test
pass:
test
```

Logowanie i rozgrywka

```
login or register (l/r)?
l
user:
test
pass:
test
welcome test tokens:1000 score:0
tokens:1000 bet?
100
dealer: 4♥, hidden
your: A♥, 10♦ (total: 21)
h/s?
s
dealer: 4♥, 4♦, 3♠, 3♦, 6♦ (total: 20)
win
play again? y/n
y
tokens:1100 bet?
200
dealer: 3♥, hidden
your: 2♦, 2♥ (total: 4)
h/s?
h
-----
your: 2♦, 2♥, Q♦ (total: 14)
h/s?
h
-----
your: 2♦, 2♥, Q♦, K♥ (total: 24)
dealer: 3♥, 4♠, 4♥, 10♥ (total: 21)
lose
play again? y/n
n
bye tokens:900 score:2
```