Adam Pilawski 155852

Sprawozdanie Sieci Komputerowe 2 Laboratoria

Sieciowa turowa gra logiczna Hnefatafl

## 1. Opis projektu

Zaimplementowana w ramach projektu gra Hnefatafl powstała w Skandynawii najprawdopodobniej jeszcze za czasów wikingów, rozgrywka odbywa się na planszy 9x9. Gra dzieli graczy na atakującego i obrońcę. Zadaniem obrońcy jest bezpieczne przedostanie króla do jednego z rogów planszy z jej środka - tronu, a atakującego wyeliminowanie króla.

Zasady gry są następujące:

- Obrońca ma króla i 8 pionków, atakujący otacza go 16 pionkami.
- Król zajmuje centralne pole na planszy, tzw. tron. Tylko król może tam przebywać, inne pionki mogą jedynie przez niego przechodzić, król może wrócić na tron.
- Grę rozpoczyna atakujący.
- Każdy gracz musi poruszyć jeden pionek w swojej rundzie, nie można pominąć ruchu ani poruszać więcej pionków niż jeden.
- Pionki poruszają się tylko w linii prostej o dowolną liczbę pól, nie można jednak przeskakiwać przez inne pionki.
- Pionki bije się otaczając je z dwóch stron.
- Pionek może wejść pomiędzy dwóch przeciwników w swoim ruchu i nie jest wtedy bity (tzw. szarża).
- Król bierze udział w biciu pionków na normalnych zasadach.
- Król jest bity jeżeli:
    - jest na tronie i zostanie otoczony z czterech stron.
    - jest tuż obok tronu i zostaje otoczony z trzech stron, z czwartej zaś strony przez tron,
    - na planszy tak samo jak inne pionki, otoczony z dwóch stron,
- Obrońca wygrywa jeśli doprowadzi króla do jednego z czterech pól w rogach planszy.
- Pola w rogach planszy i tron działają jak przeciwnicy dla obu stron, tzn. pionek umieszczony tuż obok rogu może zostać zbity jeśli po drugiej stronie znajdzie się przeciwnik.

Gra została napisana w języku programowania C w architekturze klient-serwer za pomocą komunikacji protokołem TCP i standardowych gniazd (socket). Serwer zarządza stanem gry, przechowuje planszę, weryfikuje poprawność ruchów i koordynuje logiką rozgrywki. Obsługuje jednocześnie wiele niezależnych rozgrywek dwóch graczy za pomocą wykorzystania systemu wątków oraz ich rozłączenie. Klient odpowiada za wyświetlanie planszy, wyświetlanie komunikatów odebranych od serwera, pobieranie od gracza ruchów i przesyłanie ich do serwera. W celu czytelnego zaprezentowania planszy wykorzystano kodowanie ANSI, które umożliwia wyświetlanie tekstu w różnych kolorach.

## 2. Opis komunikacji pomiędzy serwerem i klientem

Po uruchomieniu serwer nasłuchuje domyślnie na porcie 1111, oczekuje na połączenie się pierwszego gracza, po jego dołączeniu wysyłany jest do niego komunikat o oczekiwaniu na drugiego gracza. Po zaakceptowaniu połączenia drugiego gracza tworzony jest wątek obsługujący rozgrywkę między nimi, a główna pętla serwera oczekuje na dołączenie kolejnych graczy aby mogli rozpocząć rozgrywkę.

W wątku rozgrywki najpierw inicjowana i wypełniana danymi jest plansza, następnie losowany jest gracz atakujący, a tym samym gracz, który wykonuje pierwszy ruch. Do obu graczy wysyłana jest wiadomość zawierająca informację, którą stroną grają. Wiadomości wysyłane do graczy rozpoczynają się identyfikatorem – tekstem do pierwszej spacji w wiadomości, który określa jakiego typu jest to wiadomość: side – informacja o stronie w rozgrywce, msg – komunikat, board – tablica wysłana jako ciąg znaków powstały z „zlepienia" wszystkich wierszy, move – informacja o ruchu danego gracza. Aby zapobiec niepożądanym błędom w przesyłaniu komunikatów i zapewnieniu synchronizacji rozgrywki klient po odebraniu każdej wiadomości odsyła do serwera potwierdzenie w postaci ciągu znaków „ok", gdy serwer otrzyma potwierdzenie wykonuje dalsze instrukcje. Gdy gracze odbiorą informację o swoich stronach inicjowana jest zmienna *turn* typu int przechowująca informację, którego gracza jest dana tura, początkowo 0 – atakujący, a dla obrońcy 1. Następnie wątek przechodzi do głównej pętli rozgrywki:

1. Wysyła do obu graczy aktualną planszę.
2. Na podstawie flagi *wrong_move* wysyła do gracza, którego jest dana runda informację o błędnym ruchu i potrzebie jego ponownego wykonania. Jeżeli flaga nie jest ustawiona, operacja ta jest pomijana.
3. Na postawie wartości *turn* do gracza, którego jest runda wysyłane jest polecenie wykonania ruchu, a do drugiego gracza informacja o tym, że dana runda jest przeciwnika.
4. Odbierana jest od gracza wiadomość zawierająca ruch przez niego wykonany.
5. Sprawdzana jest poprawność ruchu za pomocą funkcji *move_validation*. Jeżeli ruch nie jest poprawny wszystkie opisane poniżej operacje są pomijane, ustawiana jest flaga *wrong_move* i następuje przejście do kolejnej iteracji pętli. Jeśli ruch jest poprawny za pomocą funkcji *make_move* jest on wykonywany, funkcja *battle* sprawdza czy zachodzi bicie przeciwnika i jeżeli tak wykonuje je.
6. Sprawdzane jest za pomocą funkcji *check_victory* czy, któraś ze stron wygrała, jeżeli tak wysyłane do graczy są odpowiednie komunikaty o zwycięstwie lub przegranej i gra się kończy.
7. Na koniec zmieniana jest wartość zmiennej *turn* na przeciwną (1 lub 0).

## 3. Podsumowanie

Implementacja gry Hnefatafl jako gry sieciowej spełnia wszystkie jej założenia i zasady. Serwer umożliwia rozgrywkę wielu graczy jednocześnie poprzez tworzenie osobnych wątków z dwuosobowymi sesjami. System przesyłania komunikatów do klienta umożliwia łatwą ich edycję po stronie serwera i nie wymaga zmiany kodu klienta. Klient wyświetla informacje w czytelny i przejrzysty sposób poprzez zastosowanie takich usprawnień jak kodowanie ANSI umożliwiające wyświetlanie znaków w kolorze lub czyszczenie konsoli.
Największą trudność sprawiła komunikacja między serwerem a klientem, a konkretnie problem odpowiedniej synchronizacji. Jako rozwiązanie zaimplementowano system potwierdzeń odebrania wiadomości przez klienta i tym samym przekazania do serwera informacji o gotowości do odbierania kolejnych i wykonywania następnych operacji.