

Prowadzący: dr inż. Rafał Biedrzycki

Dokumentacja wstępna

Projekt w ramach przedmiotu ZPR

Warcaby rosyjskie

Paweł Srokosz, Tomasz Nazarewicz, Paulina Pietrzyk

Opis zadania

Tematem projektu są warcaby w wariancie rosyjskim z interfejsem na przeglądarkę. Aplikacja zostanie zrealizowana w architekturze klient-serwer, gdzie rozgrywka będzie się toczyć między użytkownikiem a komputerem.

Zasady gry

Gra jest rozgrywana na planszy 8x8 pól pokolorowanych na przemian na kolor jasny i ciemny. Każdy gracz rozpoczyna grę z dwunastoma pionami (jeden koloru białego, drugi czarnego) ustawionymi na ciemniejszych polach planszy.

Pierwszy ruch wykonuje grający pionami białymi, po czym gracze wykonują na zmianę kolejne ruchy. Celem gry jest zabicie wszystkich pionów przeciwnika albo zablokowanie wszystkich, które pozostają na planszy, pozbawiając przeciwnika możliwości wykonania ruchu. Bicie pionem następuje przez przeskoczenie sąsiedniego pionu (lub damki) przeciwnika na pole znajdujące się tuż za nim po przekątnej (pole to musi być wolne). Zbite piony są usuwane z planszy po zakończeniu ruchu. Piony mogą bić zarówno do przodu, jak i do tyłu. W jednym ruchu wolno wykonać więcej niż jedno bicie tym samym pionem, przeskakując przez kolejne piony przeciwnika. Bicia są obowiązkowe. Jednak jeśli istnieje kilka możliwych bić: wykonane może być dowolne.

Pion, który dojdzie do ostatniego rzędu planszy, staje się damką, nawet w sytuacji, gdy zaszło to w trakcie bicia. Damki mogą poruszać się o dowolną ilość pól, zatrzymując się na wolnych polach. Bicie jest możliwe przez przeskoczenie pionu/damki przeciwnika, za którym znajduje się przynajmniej jedno wolne pole. Podczas bicia nie można przeskakiwać więcej niż jeden raz przez ten sam pion (damkę).

Funkcjonalności

Gra będzie rozgrywała się z serwerem wyposażonym w mechanizm sztucznej inteligencji. Gracz będzie miał możliwość wyboru jakimi pionkami chce grać. Czas rozgrywki będzie odmierzany, a następnie uwzględniany w rankingu jeśli gracz zwycięży.

Zadania do wykonania

Realizowane zadanie	Czas realizacji
1. Zaprojektowanie modelu rozgrywki i protokołu wymiany danych między klientem, a serwerem	
Opracowanie algorytmu wykorzystywanego przez sztuczną inteligencję (serwer)	4 godziny
Zaprojektowanie struktur danych służących do wymiany informacji między klientem a serwerem	2 godziny
Implementacja algorytmu “sztucznej inteligencji” z użyciem pseudokodu	4 godziny
2. Zadania związane z tworzeniem klienta	
Stworzenie szkieletu aplikacji zgodnie ze wzorcem Model-View-Controller	4 godziny
Opracowanie interfejsu i wewnętrznych mechanizmów modelu, zgodnego z zakładaną logiką gry	3 godziny
Napisanie kodu dla modelu w postaci zestawu klas w języku TypeScript	3 godziny
Testy jednostkowe modelu	1 godzina
<i>Zaprojektowanie widoku</i>	
Projekt interfejsu graficznego (stworzenie grafik - sprite’ów, ogólny zarys, pozycja poszczególnych elementów)	3 godziny
Opracowanie logicznej struktury widoku z perspektywy kodu (podział na podwidoki: ładowania gry, menu głównego, widoku rozgrywki)	2 godziny
Napisanie kodu dla widoku, z użyciem biblioteki pixi.js (renderera HTML5 dla grafiki 2D)	4 godziny
Testy jednostkowe poszczególnych elementów widoku	2 godziny
<i>Powiązanie widoku i modelu</i>	
Opracowanie kontrolera sprzęgającego widok z modelem, powiązanie widoku z modelem poprzez funkcje obsługi zdarzeń.	3 godziny
Implementacja klienta sieciowego (element kontrolera), wykorzystującego uprzednio zaprojektowany protokół do komunikacji z serwerem.	4 godziny
Testowanie rozgrywki: czy serwer współpracuje poprawnie z klientem i odwrotnie. Wymuszanie sytuacji wyjątkowych, np. rozłączenia w trakcie rozgrywki.	3 godziny

Testy funkcjonowania na popularnych przeglądarkach.	1 godzina
3. Zadania związane z tworzeniem serwera	
Opracowanie sposobu przechowywania danych (ranking)	1 godzina
Opracowanie wewnętrznych mechanizmów modelu, zgodnego z zakładaną logiką gry (np. kontrola stanu rozgrywki, czasu jej trwania itd.)	4 godziny
Implementacja komunikacji z klientem, ww. mechanizmów, algorytmów rozgrywki	4 godziny
Testowanie działania serwera (np. poprzez wysyłanie JSONów przez klienta WebSocket, rozegranie gry poprzez https://www.websocket.org/echo.html), wymuszanie sytuacji wyjątkowych	4 godziny
4. Dokumentacja projektu	
Opracowanie dokumentacji końcowej	1 godzina
Suma	57 godzin