

Technologie programowania - 2020

Lista 4

Trylma (ang. chinese checkers)

Twoim celem jest zaprojektowanie i implementacja systemu do prowadzenia rozgrywek w chińskie warcaby. System należy tworzyć **w parach wykorzystując repozytorium**. Historia zmian w repozytorium będzie podstawą do oceny wkładu członków zespołu.

Zasady gry

Zasady są dosyć łatwe, więc poniżej przytoczę je jedynie pobieżnie. Przeczytaj opis zasad na wiki i tę instrukcję.

1. W jednej grze uczestniczy 2, 3, 4 lub 6 graczy. Każdy gracz ma 10 pionów.
2. Gra toczy się na planszy w kształcie sześciopromiennej gwiazdy. Każdy promień zawiera 10 pól, w których początkowo umieszczane są piony. Wewnętrzny sześciokąt ma 61 pól, na każdy jego bok składa się 5 pól.
3. Celem każdego gracza jest umieszczenie wszystkich swoich pionów w przeciwległym promieniu. Wygrywa gracz, który dokona tego jako pierwszy. Pozostali gracze mogą kontynuować grę walcząc o kolejne miejsca.
4. Gracz rozpoczynający grę wybierany jest losowo. Następnie gracze wykonują ruchy po kolei zgodnie z kierunkiem ruchu wskazówek zegara.
5. Ruch polega na przesunięciu piona na sąsiednie puste pole lub na przeskakiwaniu pionem innych pionów (swoich lub przeciwnika) tak jak w warcabach, ale bez możliwości zbijania pionów. Ruch może odbywać się w dowolnym kierunku, poza przypadkiem, gdy pion wejdzie do docelowego (przeciwległego) promienia. Wówczas pion może wykonywać ruchy jedynie w obrębie tego promienia.
6. Dopuszczalne jest pozostawianie swoich pionów w promieniu docelowym innego gracza (tzw. blokowanie).
7. Gracz może zrezygnować z ruchu w danej turze.

Wymagania dotyczące systemu

1. System powinien działać w oparciu o architekturę klient-serwer. W razie wątpliwości warto przyjrzeć się przykładom z tej strony.
2. Gracz za pomocą aplikacji klienckiej powinien móc połączyć się z serwerem, dołączyć do gry i przeprowadzić rozgrywkę.
3. Serwer powinien weryfikować poprawność ruchów i pośredniczyć w komunikacji.

Dodatkowe wymagania

1. Postaraj się tworzyć system starannie i w przemyślany sposób. Wykorzystaj UML by zaproponować i rozwijać projekt systemu. Systemy chaotyczne zaprojektowane i pisane na ostatnią chwilę będą nisko ocenione. Bierz również pod uwagę, że to co stworzysz będzie miało wpływ na kolejną listę.
2. Stosuj podejście iteracyjne i przyrostowe.
3. Istnieje bardzo wiele wariantów gry, np. inne mogą być rozmiary planszy, inna liczba pionów, inne zasady dotyczące możliwych ruchów i blokowania. **Projektuj system tak by można go było łatwo rozszerzyć o nowe warianty gry.**
4. Postaraj się wykorzystać poznane narzędzia i wzorce. Im więcej ich zastosujesz, tym wyżej będzie ocenione Twoje rozwiązanie. Ich użycie powinno być jednak zawsze uzasadnione.

Punktacja

Za implementację całej funkcjonalności każdy student może otrzymać maksymalnie **100 punktów**. Poprawność kodu powinna być na bieżąco weryfikowana testami jednostkowymi napisanymi przy użyciu JUnit (**40 punktów**). Stopień pokrycia testami może zostać zweryfikowany przez prowadzącego np. za pomocą EclEmma. Na bieżąco wysyłaj dobrze opisane zmiany do repozytorium (**20 punktów**). Przy oddawaniu projektu powinieneś opisać jego strukturę i działanie istotnych elementów — stwórz w tym celu diagram klas i inne diagramy UML, które mogą być pomocne (**20 punktów**). Stwórz dokumentację Javadoc (**10 punktów**). Łącznie każdy student może otrzymać **190 punktów**.

Sposób zaliczenia

Pokazanie rozwiązania prowadzącemu oraz wrzucenie na ePortal: instrukcji uruchomienia projektu wraz ze wszystkimi potrzebnymi do uruchomienia plikami oraz kodów źródłowych.