STRONA 1

Opis bibliograficzny pracy :
Kuzhel Oleksandr(2015). Teiru MMORTS (Massively multiplayer online real-time strategy). Projekt i implementacja części serwerowej.

Abstrakt

Serwer jest jedną z najważniejszych części każdej gry wieloosobowej czasu rzeczywistego. Programy tego typu przeważnie wykorzystują architekturę typu klient-serwer. Główną wadą takiego rozwiązania są problemy, która objawiają się przy dużym obciążeniu serwera. Duża ilość żądań może powodować błędy, lagi, czyli chwilową utratę synchronizacji gry z poleceniami użytkownika.
Żeby zapobiec tego typu sytuacjom bardzo ważna jest optymalizacja wydajności serwera. W mojej pracy przedstawię etap projektowania oraz implementacji serwera dla wieloosobowej gry z rozgrywką czasu rzeczywistego. Celem tej pracy jest stworzenie projektu i implementacja odpowiedniego serwera dla gry wieloosobowej Teiru MMORTS.

Server is one of the most important part of each multiplayer online game. This type programs for the most part using client-server architecture. A server may receive requests from many different clients in a very short period of time. Because the computer can perform a limited number of tasks it can cause lags and synchronization problems.To prevent this type of situations the server must be optimized.
In my work I will try to show server projecting and implementation stages for multiplayer online game.
The puprose of this work is creating server project and implement it.

Słowa kluczowe

GRA KOMPUTEROWA , UNITY , JAVA , MYSQL , CZĘŚĆ SERWEROWA

COMPUTER GAME, UNITY , JAVA , MYSQL , SERVER PART

SPIS TREŚCI

WSTĘP

Gry czasu rzeczywistego z roku na rok zyskują coraz większą popularność, dlatego implementacja serwera wymaga, aby dużą część uwagi poświęcić optymalizacji wydajności serwera.

Najczęściej spotykana architektura w grach komputerowych jest architektura typu klient-serwer. Klient łączy się z serwerem za pomocą określonych protokołów komunikacyjnych. Komunikacja między serwerem, a klientem odbywa się w następujący sposób : klient nawiązuje połączenie z serwerem , po czym wysyła żądanie w określonym formacie do serwera i oczekuje na odpowiedź. Serwer ciągłe oczekuje na żądania klientów i w momencie otrzymania żądania od jednego z klientów przetwarza je, po czym wysyła odpowiedź.

Celem projektu jest implementacja serwera dla gry wieloosobowej czasu rzeczywistego Teiru MMORTS. Serwer został napisany w języku Java z wykorzystaniem bibliotek Netty[1], wersja 3.6.10 i MySql Connector[2] , wersja 5.1.34. Komunikacja po stronie klienta została napisana w języku C# z wykorzystaniem biblioteki graficznej Unity[3] , wersja 4.6.

Stworzony serwer oczekuje na żądania klientów, opracowuje je i wysyła odpowiedzi (np. odbiera informacje o przebiegu gry od jednego gracza i wysyła je do każdego z osobna). Zmienia, usuwa i dodaje informację do bazy danych. Serwer pozwała graczom na wspólną rozrywkę w świecie gry, pozwala też na swobodną wymianę wiadomości. Podczas pisania serwera skupiałem uwagę przede wszystkim na jego wydajności. Zwiększyć wydajność serwera próbowałem wykorzystując oszczędny protokół komunikacyjny i zostawiając większą część funkcjonalności po stronie klienta (np. zmianę pozycji gracza).

W następnym rozdziale omówię teoretyczną część mojej pracy , p

Rozdział 1

1.Opis projektu

1.1 UML

UML (ang. Unified modeling language) - język formalny wykorzystywany do modelowania i opisywania różnego rodzaju systemów. UML jest zestawem pojęć oraz diagramów, które pozwalają wszechstronnie odwzorować modelowaną dziedzinę problemu, założenia projektowanego systemu informatycznego, oraz większość istotnych aspektów jego konstrukcji. Diagramy UML są bardzo często wykorzystywane do dokumentowania systemów. W mojej pracy najczęściej będę wykorzystywał diagramy przypadków użycia(ang. Use case) i diagramy czynności(ang, Activity diagram).

1.1.1 Diagramy przypadków użycia

1.1.2 Diagramy czynności

1.2 Baza danych

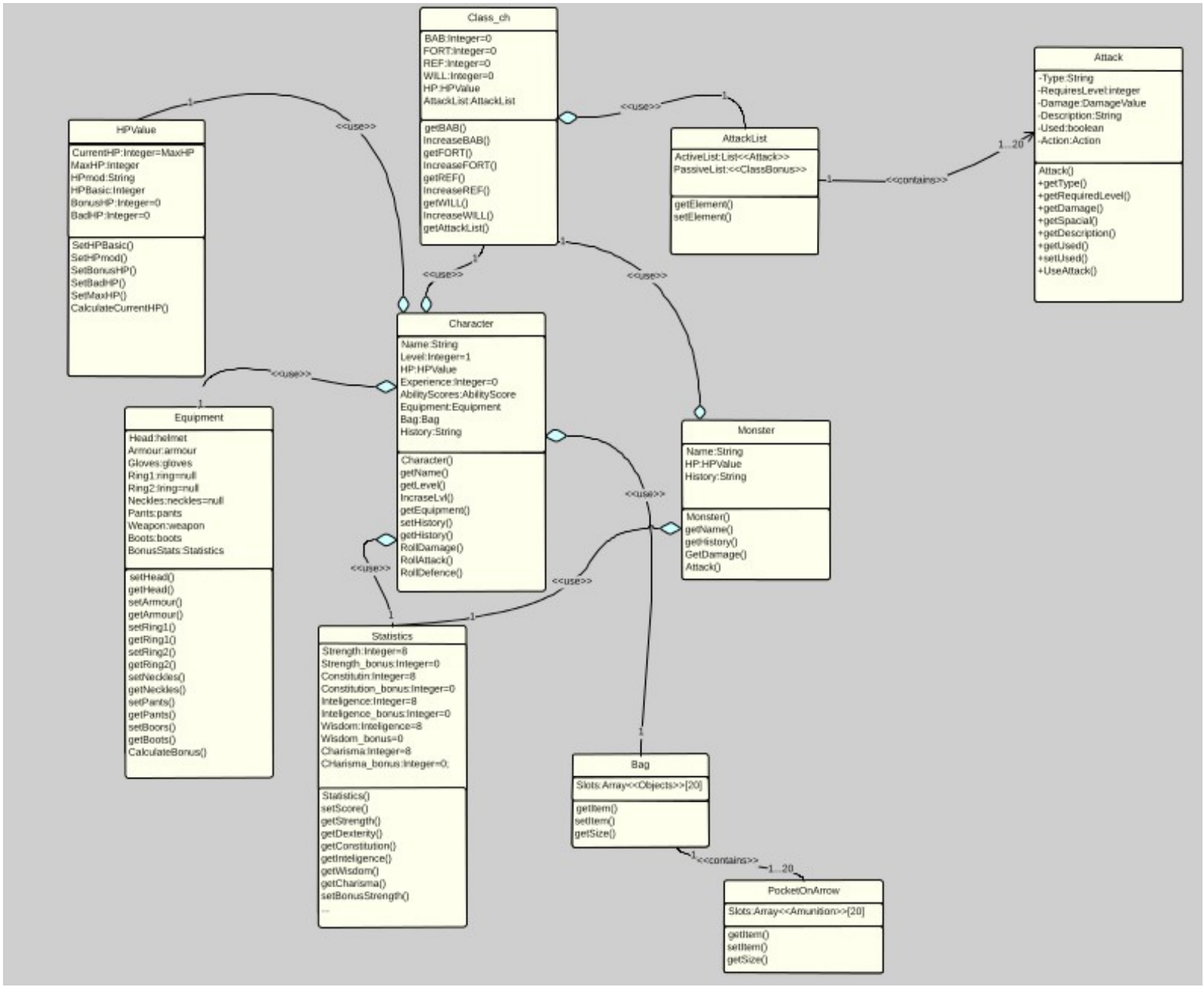
Baza danych to zbiór danych zapisanych zgodnie z określonymi regułami. W naszym przypadku to dane cyfrowe zapisywane zgodnie z zasadami przyjętymi dla danego programu komputerowego specjalizowanego do gromadzenia i przetwarzania danych.

1.2.1 Relacyjne bazy danych

W bazach relacyjnych wiele tablic danych może współpracować ze sobą (są między sobą powiązane). Bazy relacyjne posiadają wewnętrzne języki programowania, wykorzystujące zwykle SQL do operowania na danych, za pomocą których tworzone są zaawansowane funkcje obsługi danych. Relacyjne bazy danych (jak również przeznaczony dla nich standard SQL) oparte są na kilku prostych zasadach:
1.Wszystkie wartości danych oparte są na prostych typach danych.
2.Wszystkie dane w bazie relacyjnej przedstawiane są w formie dwuwymiarowych tabel(w matematycznym żargonie noszących nazwę „relacji”). Każda tabela zawiera zero lub więcej wierszy i jedną lub więcej kolumn („atrybutów”). Na każdy wiersz składają się jednakowo ułożone kolumny wypełnione wartościami, które z kolei w każdym wierszu mogą być inne.
3.Po wprowadzeniu danych do bazy, możliwe jest porównywanie wartości z różnych kolumn, zazwyczaj również z różnych tabel, i scalanie wierszy, gdy pochodzące z nich wartości są zgodne. Umożliwia to wiązanie danych i wykonywanie stosunkowo złożonych operacji w granicach całej bazy danych.
4.Wszystkie operacje wykonywane są w oparciu o algebrę relacji, bez względu na położenie wiersza tabeli. Wiersze w relacyjnej bazie danych przechowywane są w porządku zupełnie dowolnym – nie musi on odzwierciedlać ani kolejności ich wprowadzania, ani kolejności ich przechowywania.
5.Z braku możliwości identyfikacji wiersza przez jego pozycję pojawia się potrzeba obecności jednej lub więcej kolumn niepowtarzalnych w granicach całej tabeli, pozwalających odnaleźć konkretny wiersz. Kolumny te określa się jako „klucz podstawowy” (ang. primary key) tabeli.

1.2.2 Baza danych w naszej grze

Baza danych w grze została oparta na diagramie klas. Każda klasa to tabela, tabele są ze sobą powiązane tak samo jak klasy.



Rys. 1 Diagram klas

Bibliografia

[1] Netty http://netty.io/
[2] MySql Connector http://www.mysql.com/products/connector/
[3] Unity http://unity3d.com/