

Projekt z Baz Danych - Sprawozdanie

Wojciech Zając i Wojciech Wykręt

Cel:

Stworzenie użytecznej bazy danych odpowiadającej rozbudowanej grze typu MMORPG.

Główne założenia:

Każdy zarejestrowany gracz może stworzyć jedną lub więcej postaci.

Gracz może zostać z pewnych powodów zbanowany (traci możliwość logowania się do gry) przez administratora na pewien okres czasu.

Każda postać posiada swój własny ekwipunek oraz swoje statystyki.

Postacie przypisane są do lokacji, lokacje połączone są grafem, postacie mogą poruszać się między lokacjami po krawędziach grafu.

Postacie mogą zdobywać nowe przedmioty poprzez handel, wykonywanie zadań lub walkę.

Przedmioty mogą być sprzedawane między graczami w domu aukcyjnym.

Zadania składają się z jednego etapu i polegają na porozmawianiu z konkretną postacią i/lub przyniesieniu jej konkretnego przedmiotu w określonej ilości.

Dom aukcyjny działa na zasadzie jawnej licytacji.

Poziom przeciwników jest od poziomu lokacji, w której dany przeciwnik się znajduje.

Statystyki postaci są połączeniem jego statystyk podstawowych oraz sumy bonusów za każdy przedmiot w ekwipunku tej postaci.

Po walce postać losuje które (jeśli w ogóle) nagrody za zabicie przeciwnika otrzymuje, a w przypadku śmierci traci całe swoje złoto i odnawia swoje zdrowie do maksimum.

Postać może stworzyć lub dołączyć do gildii, która może skupiać wiele postaci różnych graczy.

Ograniczenia przyjęte przy projektowaniu:

Nie można usunąć aukcji z domu aukcyjnego

Nie można usuwać lokacji (usuwamy tylko połączenia do nich)

Brak weryfikacji adresu email

Dodatkowe więzy integralności:

Player:

1. Pass – Not null
2. Email – Not null, Unique

Guilds:

1. Guild_owner – Not null
2. Name – Not null, Unique
3. Members – Not null

Locations:

1. Name – Not null, Unique
2. Location_lvl – Not null

LocationsConnections:

1. Source_Location_ID – Not null
2. Destination_Location_ID – Not null

Characters:

1. Player_ID – Not null
2. Location_ID – Not null
3. Nick – Not null
4. Max_hp – Not null
5. Hp – Not null
6. Lvl – Not null
7. Character_exp – Not null
8. Gold – Not null

Items:

1. Name – Not null, Unique

Inventory:

1. Character_ID – Not null
2. Item_ID – Not null
3. Item_Amount – Not null

Banned:

1. Player_ID – Not null
2. Start – Not null
3. Finish – Not null

4. Reason – Not null

NPCs:

1. Location_ID – Not null
2. Name – Not null, Unique

Enemies:

1. Hp – Not null
2. Defence – Not null
3. Attack – Not null
4. Kill_exp – Not null

EnemyDrops:

1. Enemy_ID – Not null
2. Item_ID – Not null
3. Drop_chance – Not null

Friends:

1. Friend_ID – Not null
2. Store_ID – Not null, Unique

Stores:

1. Store_ID – Not null
2. Item_ID – Not null
3. Item_lvl – Not null
4. Unit_cost – Not null

AuctionHouse:

1. Offet_ID – Not null
2. Seller_ID – Not null
3. Item_ID – Not null
4. Starting_price – Not null
5. Beggin_date – Not null
6. End_date – Not null

AuctionHouseBids:

1. Offer_ID – Not null
2. Bidder_ID – Not null
3. Bid_amount – Not null

Quests:

1. Min_lvl – Not null
2. Quest_name – Not null, Unique
3. Quest_desc – Not null, Unique
4. Quest_giver – Not null

QuestsTracker:

1. Character_ID – Not null
2. Quest_ID – Not null
3. Quest_Status – Not null

Rewards:

1. Quest_ID – Not null
2. Item_ID – Not null
3. Amount – Not null

Utworzone indeksy:

W każdej z wymienionych wyżej tabel istnieje indeks klastrowy związany z kluczem głównym danej tabeli. Dodatkowo w bazie danych występują następujące indeksy unikalne:

1. Characters - związany z nickiem
2. Guilds - związany z nazwą gildii
3. Items - związany z nazwą przedmiotu
4. Locations - związany z nazwą lokacji
5. NPCs - związany z nazwą NPC
6. Players - związany z adresem email
7. Quests - związany z nazwą zadania
8. Quests - związany z opisem zadania
9. Friends - związany z ID sklepu

Wymaganie własności bazy danych

Tabela Banned zawiera informacje o atrybutach zmieniających się w czasie, tym atrybutem jest informacja o tym czy dany gracz jest zbanowany czy nie, zmienia się ona w czasie ponieważ np. dany gracz może zostać zbanowany na tydzień, powrócić do gry i po trzech dniach znów otrzymać bana na 5 tygodni, wszystkie te interakcje zostaną zaksięgowane w tabeli.

Tabele NPCs, Enemies i Friends implementują schemat dziedziczenia typu "Table per subclass". Tabela NPCs przechowuje własności wspólne dla tabel Enemies i Friends

(lokacja). Klucze główne w tabelach Enemies i Friends są jednocześnie kluczami obcymi odnoszącymi się do klucza głównego tabeli NPCs.

Główne założenie funkcji, widoków i procedur:

Jesteśmy świadomi, że podane poniżej funkcje, procedury i widoki obsługują tylko część funkcjonalności zaprojektowanej przez nas bazy. Zdecydowaliśmy się jednak na ich ograniczenie ze względu na to, jak kompleksowym i rozbudowanym systemem jest w pełni działająca gra komputerowa oraz ze względu na to, że część funkcjonalności można przenieść do aplikacji klienckiej. Oczywiście nasza aplikacja (kliencka i administratorska) jest tylko okrojonym przykładem zastosowania stworzonych funkcji, a zatem ta dodatkowa funkcjonalność nie jest w niej zaimplementowana.

Opis stworzonych widoków:

CurrentlyBanned - Wyświetla listę graczy, którzy aktualnie są zbanowani, czas do końca ich bana, jak i jego powód

Opis funkcji:

TryToLogin (@Email NVARCHAR(64), @Password NVARCHAR(64)) - Funkcja sprawdza czy istnieje gracz o podanym emailu i hasle i czy nie jest zbanowany, jeżeli tak to zwraca jego ID w celach zalogowania, w przeciwnym wypadku zwraca wartość -1

CharacterInventory (@Character_ID INT) -Funkcja zwraca tabele z przedmiotami posiadanymi przez postać o podanym ID

PlayerCharacters (@Player_ID INT) -Funkcja zwraca tabele ze wszystkimi postaciami stworzonymi przez gracza o podanym ID

CharactersInGuild (@Guild_ID INT) - Funkcja zwraca tabele ze wszystkimi postaciami należącymi do gildii o podanym ID

EnemiesInLocation (@Location_ID INT) - Funkcja zwraca tabele ze wszystkimi przeciwnikami w lokacji o podanym ID

FriendsInLocation (@Location_ID INT) - Funkcja zwraca tabele ze wszystkimi przyjaznymi NPC w lokacji o podanym ID

AccessibleLocations (@Character_ID INT) - Funkcja zwraca tabele ze wszystkimi lokacjami połączonymi w grafie do lokacji, w której jest postać o podanym ID

NPCsQuests (@Friend_ID INT) - Funkcja zwraca tabele ze wszystkimi questami które są dawane przez NPC o podanym ID

ItemsInStore (@Store_ID INT) - Funkcja zwraca tabele ze wszystkimi przedmiotami sprzedawanymi w sklepie o podanym ID

RewardsForQuest (@Quest_ID INT) - Funkcja zwraca tabele ze wszystkimi przedmiotami otrzymywanymi w nagrodę za wykonanie questu o podanym ID

Opis procedur składowanych:

Register (@Email NVARCHAR(64), @Password NVARCHAR(64)) -tworzy nowe konto gracza o podanych danych, pod warunkiem ze nie ma już takiego o takim samym e-mailu

CreateCharacter(@PlayerID INT, @Nick NVARCHAR(32)) - tworzy dla podanego gracza nową postać pod warunkiem ze nie ma już postaci o takim samym nicku

BanPlayer(@Nick NVARCHAR(32), @Duration INT, @Reason NVARCHAR(256))
-dodaje rekord do tabeli Banned informujący o zbanowaniu podanego gracza na podany czas z podanego powodu,

AttemptToMove(@Character_ID INT, @Destination_ID INT) - procedura dokonuje próby przesunięcia danej postaci do podanej lokacji, jeżeli postać ma zbyt niski poziom na daną lokację lub jeżeli nie ma w grafie lokacji ścieżki między aktualną lokacją postaci a wybranym celem to procedura kończy się zwracając wartość 0, jeżeli te warunki są spełnione, procedura ustawia wartość w kolumnie Location u danej postaci na docelową lokację i zwraca wartość 1

AttemptToBuy (@Character_ID INT, @Store_ID INT, @Item_ID INT, @Item_lvl INT, @Amount INT) - dokonuje próby kupienia podanego przedmiotu, z podanego sklepu, przez podaną postać, jeżeli postać nie ma wystarczającego poziomu, lub za mało złota zakup nie udaje się i procedura zwraca wartość 0, jeżeli te warunki są spełnione, odpowiedni przedmiot zostaje dodany do ekwipunku postaci, odpowiednia ilość złota pobrana a procedura zwraca wartość 1

RemoveMember(@Character_ID INT, @Guild_ID INT) - procedura ustawia kolumnę Guild_ID u podanej postaci na NULL, usuwając tą postać z danej gildii, dodatkowo zmniejszając liczbę członków tej gildii o 1, jeżeli podana postać była założycielem gildii, gildia ta zostaje usunięta

AddMember(@Character_ID INT, @Guild_ID INT) - procedura ustawia kolumnę Guild_ID u podanej postaci na podaną wartość, dodając postać do danej gildii, dodatkowo zwiększając liczbę członków tej gildii o 1, jeżeli podana postać była członkiem jakiejś gildii wcześniej, zostaje usunięta z niej z użyciem procedury RemoveMember

Death(@Character_ID INT) - procedura uruchamia się, jeżeli dana postać przegra walkę z przeciwnikiem, postać wtedy traci całe swoje złoto a jej życie jest przywracane do maksymalnego poziomu

Level_up(@Character_ID INT) - procedura inkrementuje poziom podanej postaci jak i przyznaje nagrodę w postaci zwiększenie maksymalnego życia o 10 jednostek i pełnego uleczenia postaci

Gain_Exp(@Character_ID INT,@Exp_gain INT) - procedura zwiększa podanej postaci ilość doświadczenia o podaną wartość, jeżeli nowa wartość doświadczenia jest większa niż 1000 procedura wykonuje procedurę Level_up i zmniejsza doświadczenie o 1000 aż do momentu gdy ilość doświadczenia jest mniejsza niż 1000 (za każdy 1000 doświadczenia przyznawany jest nowy poziom)

AcceptQuest (@Character_ID INT,@Quest_ID INT) - procedura dodaje wskazany quest do listy aktywnych questów dla danej postaci

ReturnQuest (@Character_ID INT,@Quest_ID INT) - procedura zmienia status danego quest dla danej postaci na zakończony (1- aktywny, 0-zakończony)

BidOnAuction (@Character_ID INT,@Offer_ID INT, @Gold INT) - procedura dodaje nową ofertę na przedmiot na domu aukcyjnym, jeżeli dana postać już złożyła ofertę na dany przedmiot, procedura zamiast dodawania nowej oferty podnosi ilość złota w starej ofercie do nowej wartości

Opis wyzwalaczy:

AddItem- modyfikuje ilość danego przedmiotu w ekwipunku danej postaci o daną ilość, następnie modyfikuje zmieniane przez nie statystyki postaci (Max HP), pozwala zarówno na zwiększenie ilości danych przedmiotów posiadanych przez daną postać, jak i jej zmniejszenie, jeżeli dodajemy przedmiot którego postać wcześniej nie miała wyzwalacz doda nowy rekord, w przeciwnym przypadku jedynie zmodyfikuje w nim pole Item_amount by odzwierciedlało zmianę, jeżeli końcowy liczba przedmiotów będzie niedodatnia rekord zostanie usunięty

DeleteItem - usuwa podany przedmiot z ekwipunku danej postaci, następnie modyfikuje odpowiednio statystyki danej postaci (Max HP)

DeleteGuild - usuwa podana gildie, przed jej usunięciem usuwa z niej wszystkich jej członków

CreteGuild - po stworzeniu nowej gildii, dodaje jej twórcę jako jej pierwszego członka

DeleteCharacter - wyzwalacz służący usunięciu postaci i wszystkich powiązanych z nią rekordów, przed usunięciem usuwa ją także z gildii używając procedury RemoveMember

Założenia aplikacji klienckiej i administratorskiej

Głównym założeniem obu aplikacji jest pokazanie przykładowych funkcji dostępnych w grze z nastawieniem na możliwość łatwego dodania dodatkowych funkcjonalności w razie potrzeby.

Działanie aplikacji klienckiej:

W momencie włączenia aplikacji klienckiej (user.py) przeprowadzona zostaje symulacja logowania i wybierania postaci (dzieje się to automatycznie, ale aplikacja naprawdę odnosi się do bazy danych przy próbie logowania). Po zalogowaniu aplikacja wyświetla wszystkie postaci przyjazne, potwory oraz lokacje sąsiednie wraz z pokazaniem po prawej ich stronie ID umożliwiających wykonywanie poleceń w aplikacji. Użytkownik ma do dyspozycji pięć poleceń:

1. attack [ID] - zaatakowanie potwora o podanym ID
2. store [ID] - wejście do sklepu o podanym ID
3. move [ID] - przejście do lokacji o danym ID (o ile pozwala na to poziom)
4. show eq - wyświetlenie zawartości ekwipunku gracza
5. exit - zakończenie działania aplikacji klienckiej

Działanie aplikacji administratorskiej:

Po włączeniu aplikacji administratorskiej (admin.py) administrator ma do dyspozycji 3 opcje, które wybiera się poprzez wpisanie odpowiednio liczby 1-3. Ma on możliwość zobaczenia listy graczy, którzy w czasie sprawdzania są zbanowani wraz z powodami tychże banów. Administrator ma również możliwość zbanowania graczy na podstawie nazwy jednej z ich postaci na określony czas i z podaniem przyczyny.

Łączenie aplikacji klienckiej i administratorskiej z serwerem:

W celu poprawnego połączenia zarówno aplikacji klienckiej jak i administratorskiej należy zmienić zawartość pliku "server_name" na prawidłową nazwę serwera SQL, który jest uruchomiony na danym komputerze.

Opis tworzenia bazy danych:

W celu poprawnego utworzenia bazy danych, wypełnienia jej przykładowymi danymi, a następnie wyświetlenia przykładowych zapytań należy wykonać pliki ".sql" w następującej kolejności: "init.sql" -> "inserts.sql" -> "queries.sql". Aplikacji klienckiej ani administratorskiej nie należy używać przed wykonaniem zapytań z pliku "inserts.sql".

Strategia pielęgnacji bazy danych

Przyjęta przez nas strategia pielęgnacji bazy danych polega na krótkiej cotygodniowej przerwie w działaniu bazy danych (w godzinach nocnych) w celu stworzenia kopii

zapasowej. Kopie zapasowe w naszej strategii przechowywane są do dwóch tygodni wstecz oraz znajdują się na osobnych nośnikach, w celu uniknięcia fizycznego zniszczenia kopii.

[illegible]