

RBD – Projekt PuzzledMonkey

Opis projektu

Celem projektu było stworzenie bazy danych odpowiedzialnej za przechowywanie informacji nt. świata tworzonej przez nas sieciowej gry logicznej PuzzledMonkey oraz implementacja warstwy logicznej w postaci procedur SQL zajmujących się następującymi zadaniami:

- Przygotowanie tabel do nowej rozgrywki
- Wygenerowanie świata
- Wylosowanie klucza odpowiedzi
- Obsługa zdarzeń takich jak wysyłanie wiadomości, ruch graczy, podnoszenie, upuszczanie i obrót kostki
- Obsługa logów odbieranych i wysyłanych do graczy
- Rozstrzygnięcie poprawnego rozwiązania

Warstwę logiczną tworzą następujące procedury:

Procedura	Opis
pm_game_make	Czyszczenie tabel, generowanie nowego świata o określonym rozmiarze (tworzenie graczy, rozmieszczenie kostek i slotów) oraz wygenerowanie klucza odpowiedzi
pm_game_join	Dołączenie gracza do gry
pm_game_get	Pobranie listy kostek
pm_game_get_sides	Pobranie kolejności ścianek kostki
pm_game_update	Pobranie listy logów (komend)
pm_cube_take	Podniesienie kostki przez gracza
pm_cube_insert	Wstawienie kostki do slotu
pm_cube_rotate	Obrót kostki
pm_logs_add	Dodanie nowego loga do potoku
pm_chat	Wysyłanie wiadomości

Warstwa przechowująca dane składa się z poniższych tabel:

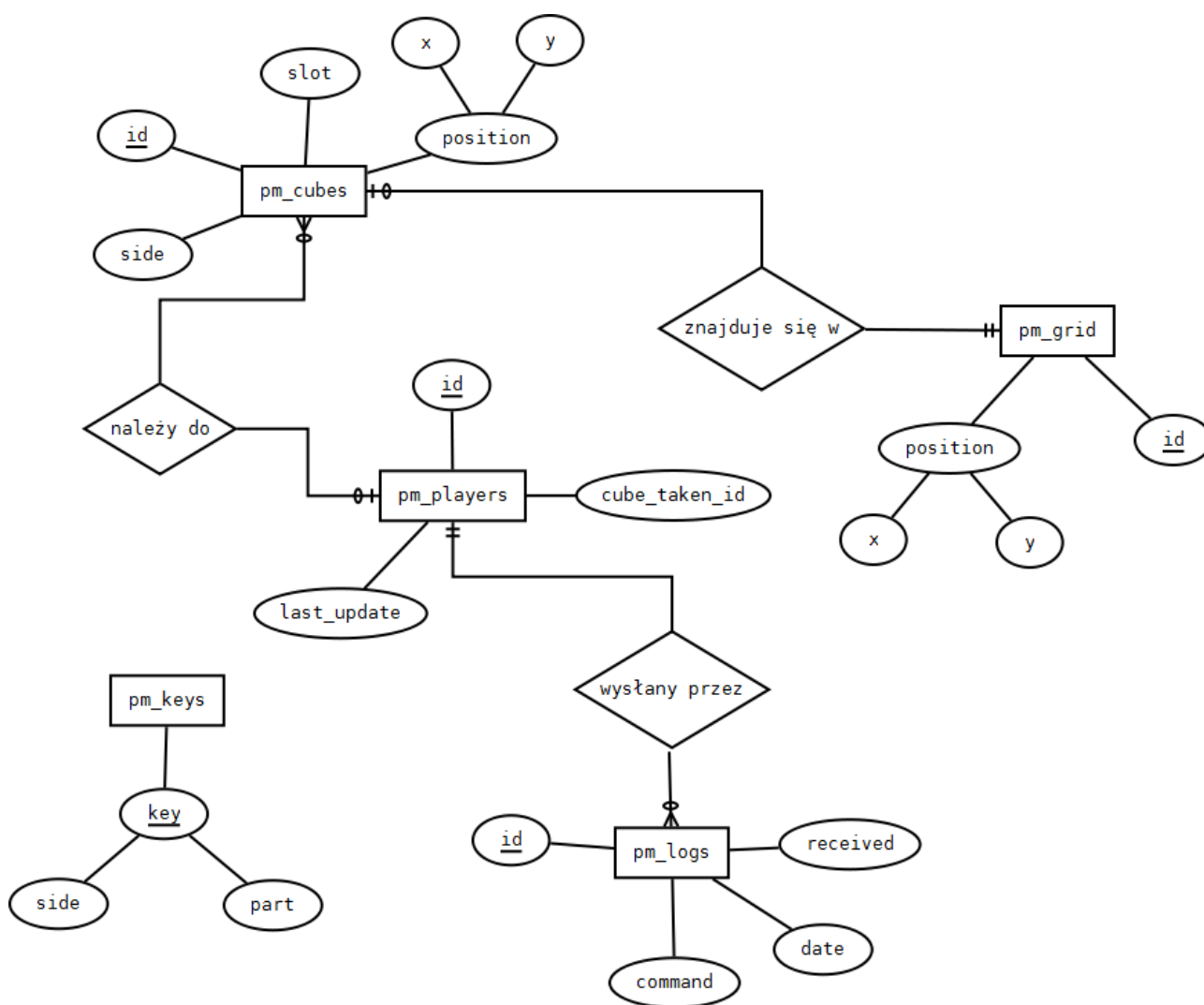
Tabela	Opis
pm_cubes	Kostki znajdujące się na mapie
pm_grid	Tabela pomocnicza podczas generowania świata
pm_keys	Klucz odpowiedzi
pm_logs	Potok logów
pm_players	Lista graczy

Po rozpoczęciu gry wywoływana jest procedura `pm_game_make` czyszcząca wszystkie tabele zawierające dane z poprzedniej rozgrywki. Następnie do tabeli `pm_grid` wstawiane są pozycje graczy oraz slotów - jest to tabela pomocnicza wykorzystywana podczas generowania świata, a dokładniej rozmieszczania kostek na mapie. Po wstawieniu powyższych danych następuje wylosowanie pozycji kostek z uwzględnieniem pól zajętych tj. pozycji zajmowanych przez graczy, sloty oraz już umieszczone kostki. Końcowym etapem zakładania gry jest wylosowanie klucza odpowiedzi tj. kolejności w jakiej ułożone mają zostać kostki w odpowiednich slotach.

Po założeniu gry serwer oczekuje procedury `pm_game_join`, wywoływanej przez drugiego gracza po dołączeniu do gry. W tym momencie następuje regularne odbieranie logów przez obu graczy za pomocą procedury `pm_game_update`. Procedura ta zwraca listę akcji wykonanych przez przeciwnika w postaci komend interpretowanych po stronie klienta. Komendy te dodawane są do potoku logów poprzez procedury `pm_cube_take`, `pm_cube_insert`, `pm_cube_rotate` oraz `pm_chat`. Procedura `pm_cube_insert`, oznaczająca wstawienie kostki do slotu, sprawdza z kluczem odpowiedzi ułożenie kostek - jeśli wszystkie kostki znajdują się na odpowiednich pozycjach, gra zostaje zakończona.

Diagram

Ze względu na specyfikę stworzonej przez nas bazy danych nie zawiera ona żadnych jawnych związków pomiędzy tabelami w postaci kluczy obcych z uwagi na realizację logiki w procedurach SQL - na diagramie przedstawione zostały więc tylko logiczne powiązania pomiędzy poszczególnymi tabelami.



Fragmenty tabel

pm_cubes					
Id	Side	Slot	Player	X	Y
1	1	Null	Null	-23	5
2	4	2	1	-24	-15
3	1	Null	2	19	-24

pm_grid		
Id	X	Y
3467	-1	2
3468	-1	3
3469	0	2

pm_keys	
Side	Part
1	1
2	6
3	5

pm_logs				
Id	Player	Command	Date	Received
107	2	\$ system opponent_join	2011-01-06 18:41:23	0
108	2	\$ opget Cube1	2011-01-06 18:41:33	0
109	2	\$ oprot up	2011-01-06 18:41:40	0

pm_players		
Id	last_update	cube_taken_id
1	2011-01-06 18:40:14	Null
2	2011-01-06 18:54:37	3