Akademia Górniczo-Hutnicza im. Stanisława Staszica

Kraków

Osadnicy z Catan - Gra sieciowa

Autorzy:
Marcin Jędrzejczyk
Sebastian Katszer
Katarzyna Kosiak
Paweł Ogorzały

30 października 2015

Spis treści

1 Wstęp

Niniejszy dokument stanowi dokumentację projektu "Osadnicy z Catanu - Gra sieciowa" z przedmiotu Wprowadzenie do Wzorców Projektowych. Nasz projekt zrealizowaliśmy

1.1 Dlaczego Catan?

Nasza grupa projektowa składa się z osób zafascynowanych światem zarówno gier planszowych jak i komputerowych. Projekt ten umożliwia nam połączenie naszych zainteresowań. Wybór padł na grę "Osadnicy z Catanu" z bardzo prostego powodu: dzięki prostocie i przejrzystości zasad stanowi ona idealny start dla osób, które chcą rozpocząć swoją przygodę z grami planszowymi.

Dzięki wprowadzeniu rozgrywki sieciowej możliwe jest rozegranie partii z przyjaciółmi z całego świata bez dodatkowych wydatków, a także rozpowszechnienie tej gry i ułatwienie wejścia w świat gier planszowych.

1.2 Opis gry

Osadnicy z Catanu (Settlers of Catan) to jedna z najpopularniejszych rodzinno-ekonomicznych gier planszowych na świecie. Gracze są osadnikami na niedawno odkrytej wyspie Catan. Każdy z nich przewodzi świeżo założonej kolonii i rozbudowuje ją stawiając na dostępnych obszarach nowe drogi i miasta. Każda kolonia zbiera dostępne dobra naturalne, które są niezbędne do rozbudowy osiedli.

Gracz musi rozważnie stawiać nowe osiedla i drogi, aby zapewnić sobie dostateczny, ale zrównoważony dopływ zasobów, a jeśli ma ich nadmiar - prowadzić handel z innymi graczami sprzedając im owce, drewno, cegły, zboże lub żelazo a pozyskując od nich te zasoby, których ciągle mu brakuje.

Pierwszy z graczy, który uzyska 10 punktów z wybudowanych przez siebie dróg, osiedli i specjalnych kart - wygrywa.

2 Użyte wzorce projektowe

2.1 Singleton

Singleton jest kreacyjnym wzorcem projektowym, którego zadaniem jest ograniczenie możliwości tworzenia obiektów danej klasy do jednej instancji, oraz zapewnienie globalnego dostępu do stworzonego obiektu.

Singleton implementuje się poprzez stworzenie klasy, która posiada statyczną metodę, która sprawdza czy istnieje instancja danej klasy, w razie potrzeby tworząc ją. Następnie instancja zwracana jest przez referencję. Aby uniemożliwić tworzenie dodatkowych instancji, konstruktor klasy deklaruje się jako prywatny lub chroniony.

Wzorzec ten ma u nas zastosowanie w przypadku klas odpowiadajacych za kostkę, planszę. Zapewnia on globalny dostęp oraz ogranicza możliwość tworzenia większej liczby instancji.

2.2 Builder

Builder jest wzorcem konstrukcyjnym, który ma za zadanie oddzielenie tworzenia obiektów od ich reprezentacji. Proces tworzenia obiektu podzielony jest na mniejsze etapy, a każdy z nich może być implementowany na wiele różnych sposobów. Umożliwia to tworzenie różnych reprezentacji

obiektów w tym samym procesie konstrukcyjnym. Konstruowanie obiektu następuje poprzez wcześniejsze stworzenie jego fragmentów.

Na wzorzec składa się:

- Builder - dostarcza inferfejs do tworzenia obiektów nazywanych produktami,
- ConcreteBuilder - tworzy konkretne reprezentacje produktów przy pomocy zaimplementowanego interfejsu Builder,
- Director - zleca konstrukcję produktów poprzez obiekt Builder.

W naszym projekcie wzorzec ten znajduje zastosowanie przy tworzeniu planszy, która składa się z kafli.

2.3 Abstract Factory

Kreacyjny wzorzec projektowy, który pozwala tworzyć całe rodziny produktów. Dostarcza on interfejs do tworzenia różnych obiektów jednego typu bez specyfikowania ich konkretnych klas.

Wzorzec ten wykorzysytwany będzie przy konstruowaniu kafli oraz generowaniu kart.

2.4 Observer

Wzorzec Óbserver"jest używany jeśli występuje relacja jeden do wielu pomiędzy obiektami. Modyfikacja jednego obiektu powoduje, że zależne obiekty są powiadamiane automatycznie. Wzorzec ten podchodzi pod kategorię wzorców czynnościowych.
Wzorzec ten pomoże nam w komunikacji między graczami. Będziemy wysyłać informacje o tym co się zmieniło na planszy. A OBSERVER każdego gracza będzie je wyłapywał i egzekwował.

2.5 State

We wzorcy Śtateźachowanie klasy zmienia się w zalezności od jej stanu. Ten typ wzorca wchodzi pod wzorce czynnościowe.
Tworzymy obiekty, które reprezentują różne stany i konteksty obiektu, którego zachowanie zmienia się tak jak jego stan. W zależności od tego jaką kartę zagramy stan planszy może się zmienić. Możemy wybudować drogi, wejść w interakcję z innymi graczami, czy zablokować dochód z danego pola. Dlatego potrzebujemy wzorca odpowiedzialnego za zajmowanie się stanem klasy.

2.6 Command

Wzorzec Ćommandópakowuje żądania w obiekty jako rozkaz i przesyła do obiektu wzywającego (z ang. invoker). Obiekt wzywający szuka obiektu właściwego, który może zająć się tym rozkazem i przesyła rozkaz do odpowiedniego obiektu który wykonuje rozkaz.
Wzorzec ten należy do wzorców czynnościowych, czyli opisujących sposób przepływu danych w złożonych aplikacjach.??? Skorzystamy z niego, by móc zagrywać karty rozwoju, które są swoistymi rozkazami. Wzorzec ten będzie musiał u nas współpracować z kodem napisanym pod wzorzec Śtate". TO DO gdzie z tego skorzystamy Commands are an object-oriented replacement for callbacks.

3 Użyte biblioteki zewnętrzne

3.1 Biblioteka graficzna

Jeśli chodzi o bibliotekę graficzną, to nasz wybór padł na libGDX - crossplatformową bibliotekę z licencją opensource (oficjalna strona internetowa: www.libgdx.badlogicgames.com).

Jedną z głównych zalet tej biblioteki jest jej szybkość - twórcy postawili bardzo duży nacisk na rozsądne zarządzanie pamięcią. Dodatkowo libGDX dostarcza narzędzi do obsługi dźwięku i interpretacji wejścia dostarczonego przez użytkownika. Mimo, że renderowanie zachodzi przy użyciu OpenGL ES 2.0, to w większości przypadków nie trzeba znać szczegółów działania tego API, ponieważ biblioteka libGDX dostarcza wygodnych, wysokopoziomowych metod, dzięki którym sprawnie można uzyskać pożądane efekty - od wyświetlania dwuwymiarowych obiektów po tworzenie trójwymiarowych scen.

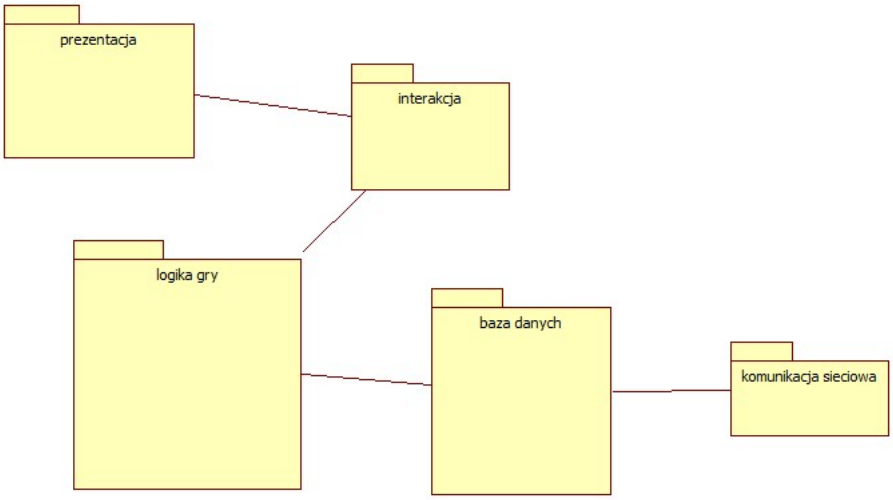
3.2 Biblioteka sieciowa

Jako architekturę do połączeń internetowych wybraliśmy Peer-2-Peer. Powodem jest brak zewnętrznego serwera jak i chęć nauczenia się nowych technologii. Do jej implementacji wykorzystana zostanie biblioteka JXTA, która rozpowszechniona jest na licencji opensource. JXTA jest zbiorem sześciu protokołów, za pomocą których jesteśmy w stanie połączyć się z innymi węzłami w sieci, nawet jeśli znajdują się one za NAT-em czy też za Firewallem. Jest on niezależny od systemu operacyjnego, języka programowania, wykorzystywanego protokołu transportowego oraz jest dla każdego urządzenia cyfrowego.

Nawiązanie połączenia pomiędzy węzłami jest możliwe dzięki peer groups. Dla każdej grupy urządzeń tworzona jest grupa, sieć wirtualna, która jest niezależna od rzeczywistej topologii, w ramach której udostępniane są usługi i następuje wymiana danych. Utworzenie tej grupy jak i jej znalezienie przez innych użytkowników jest możliwe dzięki advertisment, która jest swoistą wizytówką. Każdy korzystający z JXTA znajduje się domyślnie w światowej grupie, w której może wyszukać czy też propagować grupę. Wszystkie informacje są przekazywane za pomocą pipe-ów, które dzielą się na wejściowe i wyjściowe. Umożliwiają one komunikację typu jeden do jeden, jeden do wielu, wielu do wielu.

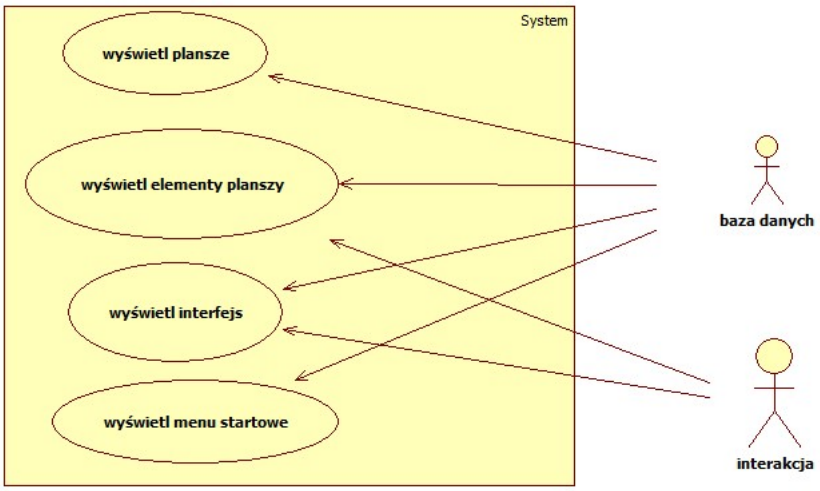
4 Architektura

Poniżej prezentujemy opis wszystkich warstw, na które podzieliliśmy naszą aplikację.

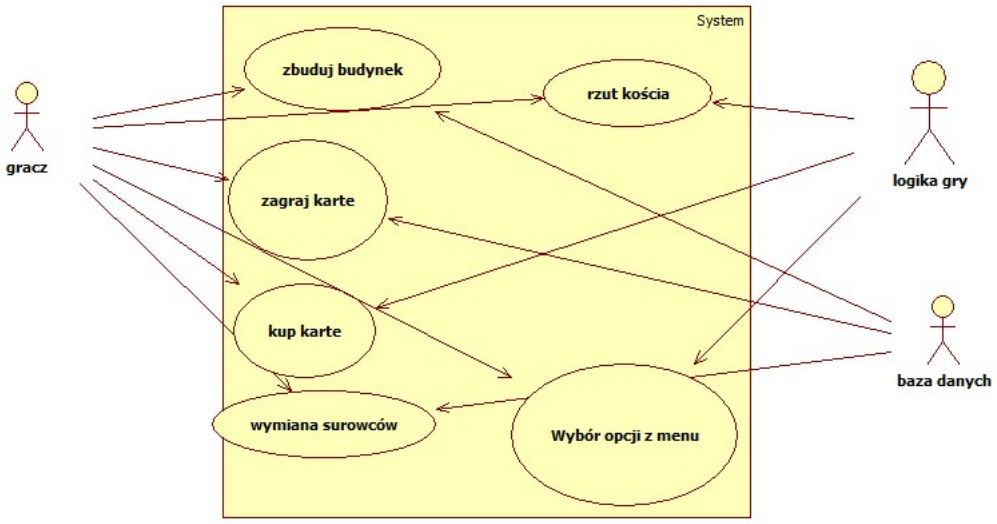


Rysunek 1: Podział na warstwy

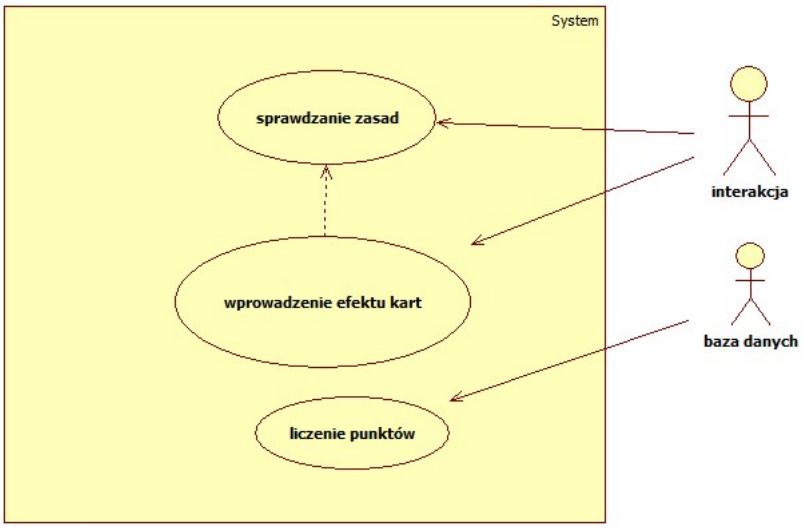
4.1 Warstwa prezentacji



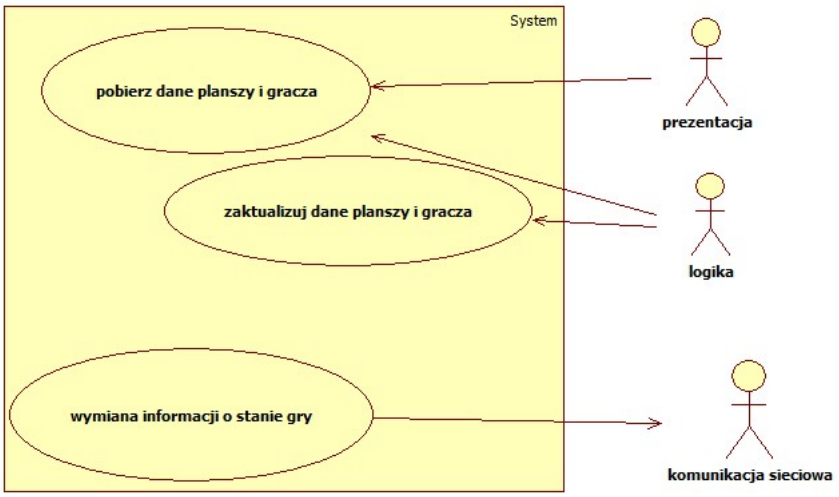
4.2 Warstwa interakcji



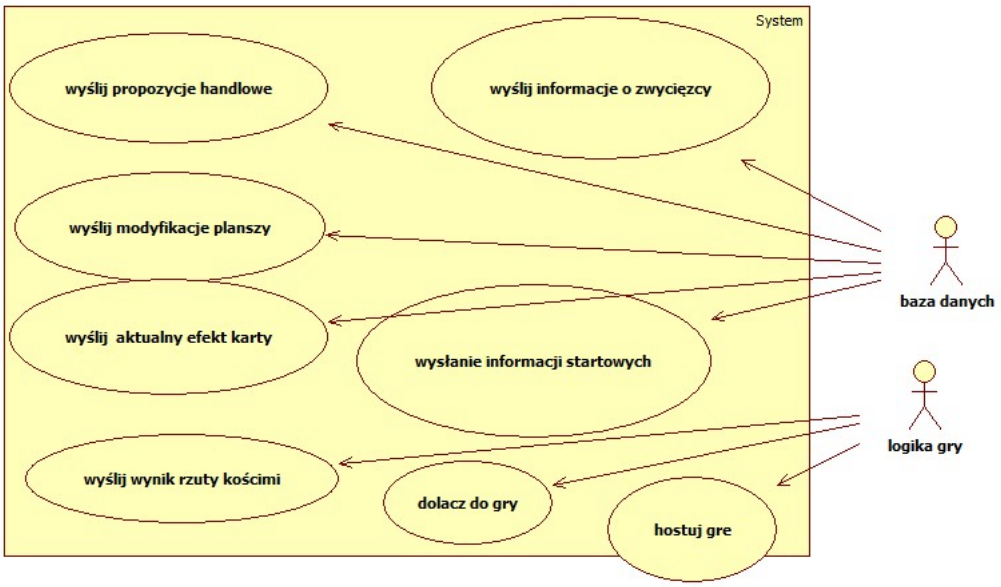
4.3 Warstwa logiki gry



4.4 Warstwa baz danych



4.5 Warstwa komunikacji sieciowej



5 Podsumowanie

6 Testy

7 Wnioski

8 Literatura

Asensio MI, Ferragut L., Simon J.: Modelling of convective phenomena in forest fire. Rev Real Academia de Ciencias, 2002, 96:299–313