

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA
STASZICA

KRAKÓW

Osadnicy z Catan - Gra sieciowa

Autorzy:

Marcin JĘDRZEJCZYK

Sebastian KATSZER

Katarzyna KOSIAK

Paweł OGORZAŁY

28 stycznia 2016

Spis treści

1	Wstęp	2
1.1	Dlaczego Catan?	2
1.2	Opis gry	2
2	Skrócone zasady gry	2
3	Użyte wzorce projektowe	3
3.1	Singleton	3
3.2	Builder	4
3.3	Abstract Factory	4
3.4	Observer	4
3.5	State	4
3.6	Command	4
4	Użyte biblioteki zewnętrzne	5
4.1	Biblioteka graficzna	5
4.2	Biblioteka sieciowa	5
5	Architektura	5
6	Podsumowanie	5
7	Testy	5
8	Wnioski	6
9	Literatura	6

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 Skrócone zasady gry

- Wygrywa gracz, który jako pierwszy zdobędzie 10 punktów zwycięstwa,
- Gracz może handlować z bankiem lub z innymi graczami, ale może zainicjować handel tylko podczas swojej tury,
- Na wierzchołkach sąsiadujących z miastami lub osadami nie można stawiać nowych budynków,
- Do nowej osady musi prowadzić droga. Wyjątek stanowią 2 pierwsze osady gracza,
- Dwie pierwsze osady gracza są darmowe. Dwie pierwsze drogi gracza także są darmowe,
- Na pustym polu, na którym można budować, można postawić osadę lub miasto (z wliczoną ceną osady),
- Miasta mogą nadbudowywać osady,
- Nowa droga musi się zaczynać w mieście, osadzie lub być kontynuacją innej drogi, której gracz jest właścicielem,

- Rzut kością decyduje o tym, które pola przyniosą zasoby w danej turze. Jedna osada produkuje 1 zasób, jedno miasto produkuje 2 zasoby,
- Gracz, który ma więcej niż 5 żołnierzy dostaje kartę największej armii, kolejni gracze muszą mieć o 1 żołnierzy więcej, by odebrać tę kartę właścicielowi,
- Gracz, którego nieprzerwana droga jest dłuższa niż 5 dostaje kartę najdłuższej drogi handlowej, kolejni gracze muszą mieć o 1 dłuższą nieprzerwaną drogę, aby odebrać tę kartę właścicielowi.

Co daje punkty zwycięstwa:

- Osada - 1 punkt zwycięstwa,
- Miasto - 2 punkty zwycięstwa,
- Karta punkty zwycięstwa - 1 punkt zwycięstwa,
- Karta największej armii - 2 punkty zwycięstwa,
- Karta najdłuższej drogi handlowej - 2 punkty zwycięstwa.

Wyjaśnienie kart rozwoju:

- Karta żołnierza-zwiększa liczebność armii gracza o 1, pozwala przesunąć złodzieja,
- Karta monopolu- pozwala wybrać zmonopolizowany surowiec, wszyscy gracze muszą oddać cały stan tego surowca zagrywającemu tę kartę,
- Karta urodzaju- pozwala wybrać 2 darmowe surowce,
- Karta punktu zwycięstwa- zagrywający dostaje 1 punkt zwycięstwa,
- Karta budowy dróg- zagrywający dostaje 2 darmowe drogi do wybudowania.

Koszty:

- Osada- 1 drewna, 1 cegły, 1 wełny, 1 zboża,
- Miasto- 3 rudy i 2 zboża lub 1 drewna, 1 cegły, 1 wełny, 3 zboża i 3 rudy,
- Karta rozwoju- 1 rudy, 1 wełny, 1 zboża,
- Droga- 1 drewno, 1 cegły,

3 Użyte wzorce projektowe

3.1 Singleton

Singleton jest kreatywnym 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.

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

3.2 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 wzorec składa się:

- Builder - dostarcza interfejs 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 wzorec ten znajduje zastosowanie przy tworzeniu planszy, która składa się z kafli.

3.3 Abstract Factory

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

Wzorec ten wykorzystywany będzie przy konstruowaniu kafli oraz generowaniu kart.

3.4 Observer

Wzorec "Observer" jest używany jeśli występuje relacja jeden do wielu pomiędzy obiektami. Modyfikacja jednego obiektu powoduje, że zależne obiekty są powiadamiane automatycznie. Wzorec ten podchodzi pod kategorię wzorców czynnościowych.

Wzorec 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.

3.5 State

We wzorcy "State" zachowanie klasy zmienia się w zależności 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.

3.6 Command

Wzorec "Command" opakowuje 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.

Wzorec 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. Wzorec ten będzie musiał u nas współpracować z kodem napisanym pod wzorec "State". TO DO gdzie z tego skorzystamy Commands are an object-oriented replacement for callbacks.

4 Użyte biblioteki zewnętrzne

4.1 Biblioteka graficzna

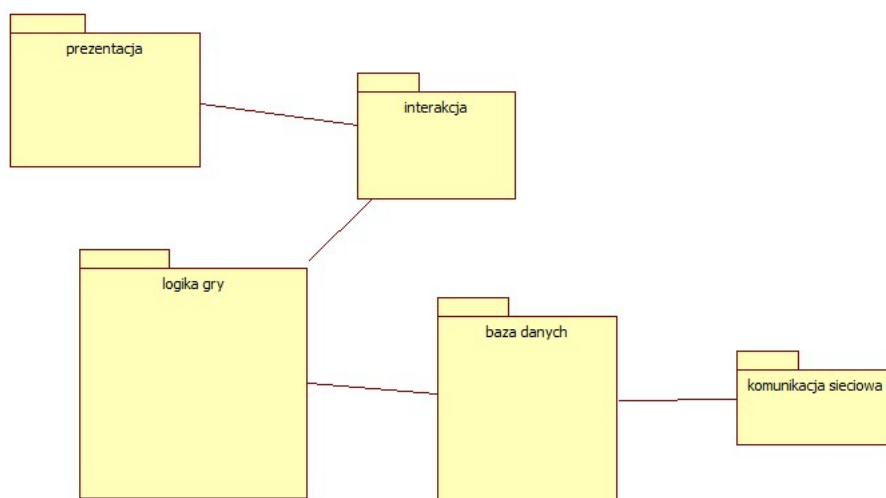
LibGDX to wieloplatformowe narzędzie do tworzenia gier i wizualizacji.
<https://libgdx.badlogicgames.com/>

4.2 Biblioteka sieciowa

Jxta jest to zbiór uogólnionych i otwartych protokołów pozwalającym na komunikację w stylu peer-to-peer.
<https://jxta.kenai.com/>

5 Architektura

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



Rysunek 1: Podział na warstwy

6 Podsumowanie

Wspólna praca pozwoliła nam poprawić zdolność do pracy w zespole i pozwoliła przekonać się jak zmiany w kodzie jednej osoby wpływają na działanie kodu drugiej. Podział pracy na kilka warstw pozwolił na nam działać trochę niezależnie i równolegle, jednakże skutkowało to większym nakładem pracy nad połączeniem poszczególnych części. Przekonaliśmy się jak wzorce projektowe ułatwiają życie i czytelność kody. Ostatecznie udało się nam wyprodukować działająca grę, co cieszy nas niezmiernie.

7 Testy

Dzięki stałemu testowaniu, byliśmy w stanie wyłapywać różne błędy logiki gry na bieżąco. Niektóre wymagały zmian podejścia do danych problemów i innych metod nimi zarządzających.

8 Wnioski

- Po zdefiniowaniu problemu należy sprawdzić, czy jakiś wzorzec nie pomoże w jego rozwiązaniu,
- Nie stosować wzorców na siłę, wzorzec nie musi zawsze pomóc,
- GitHub jest wygodnym narzędziem do pracy grupowej i kontrolowania wersji projektu,
- Trello pomaga wytyczać cele i przydzielać zadania.

9 Literatura

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