

Sprawozdanie z pracowni specjalistycznej

Zaawansowane Techniki Programistyczne

Projekt semestralny

Temat: Gra zręcznościowa na platformę Android

Wykonujący ćwiczenie

Artur Kruszewski, Adrian Małysko, Mirosław Gulewicz

Studia dzienne

Kierunek: Informatyka

Semestr: V

Grupa zajęciowa: PS3

Prowadzący ćwiczenie: **mgr. Inż. Jerzy Krawczuk**

1. Treść zadania

Gra zręcznościowa polegająca na przechodzeniu plansz poprzez zestrzeliwanie obcych statków i unikanie ich ostrzału. W czasie gry można ulepszać swoje działo zbierając odpowiednie bonusy na planszy. Poziom trudności powinien rosnać na kolejnych planszach, np. poprzez pojawianie się nowych statków, zwiększania ich ilości, czy też częstotliwości ostrzału. W grze zliczane są punkty i prezentowany jest ranking najlepszych wyników.

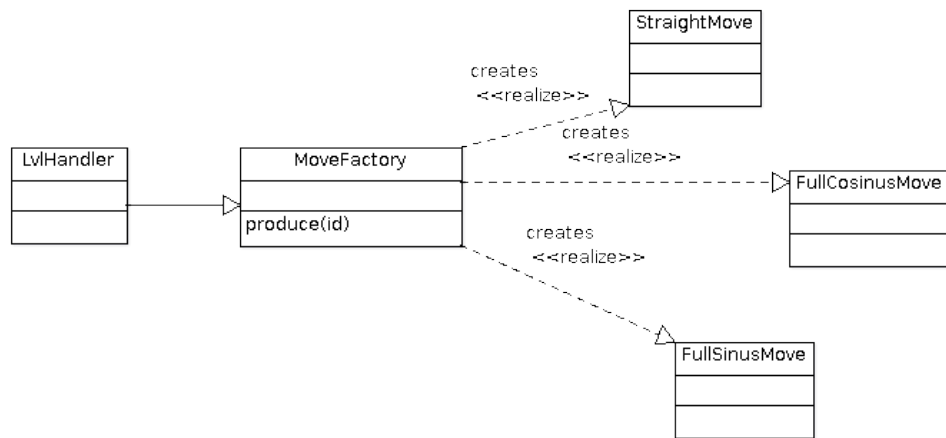
2. Założenia

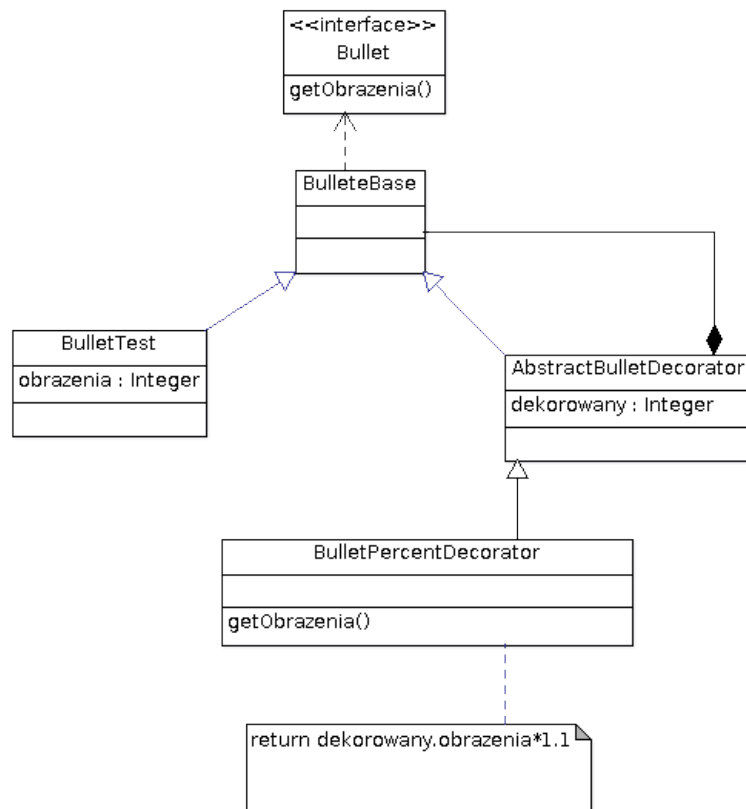
- Wykorzystanie wzorców programistycznych:
 - Singleton
 - Factory method
 - Decorator
 - Iterator
 - Observer
 - Strategy
- Funkcjonalności
 - Wyświetlanie obiektów
 - Detekcja kolizji
 - Możliwość sterowania
 - Możliwość ulepszania
 - Tworzenie statków przeciwnika
 - Rosnący poziom trudności
 - Zliczanie punktów graczy i prezentowanie listy najlepszych

3. Wykorzystane wzorce

- **Factory method**

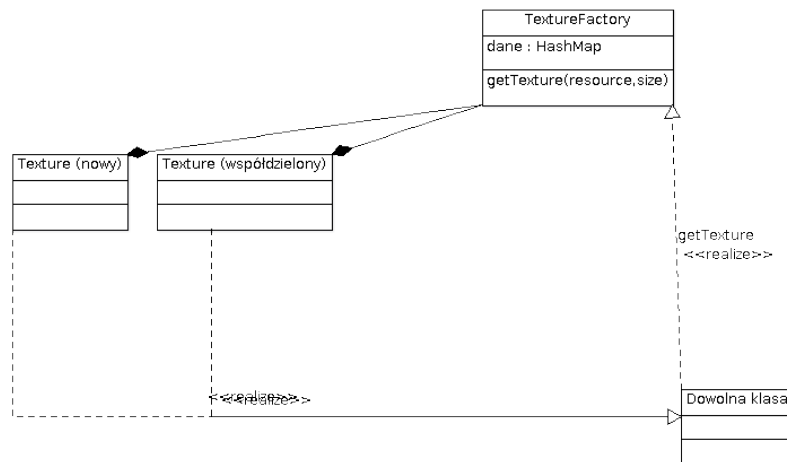
Wykorzystany przy wytwarzaniu ruchu na bazie danych z pliku xml (przekazywany jest id i na jego podstawie wytwarzany jest odpowiedni obiekt ruchu)





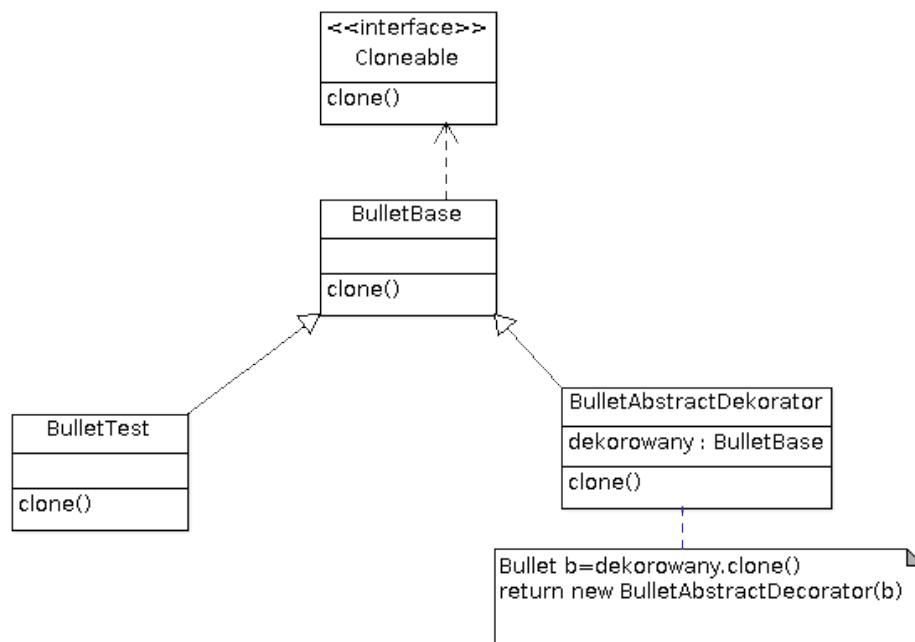
Dekoratora użyliśmy do modyfikacji parametrów broni oraz pocisków. Dekorator użyty został w celu „dekorowania” broni, poprzez modyfikację jej funkcjonalności.

Pylek



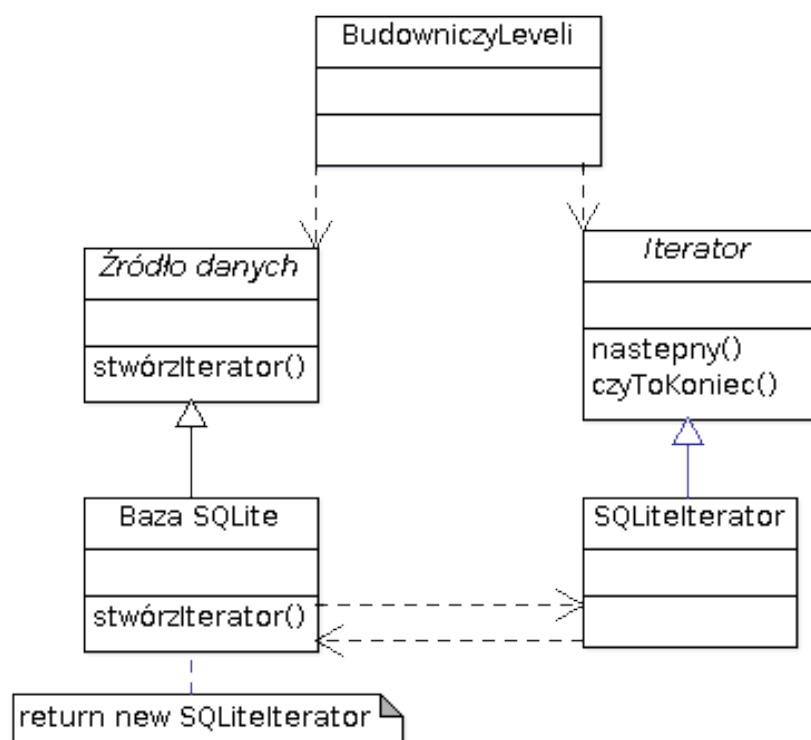
Pylek został użyty w celu redukcji ilości przeładowań tekstur (co wymaga zatrzymania renderowania) oraz zmniejszenia ilości tekstur jednocześnie obecnych w pamięci. Do tego celu wykoszystałem strukturę podobną do HashMapy mianowicie SparseArray w pojedynczym zagnieżdzeniu. Klucze stanowią id tekstury (int) oraz rozmiar (int). Fabryka zawiera HashMapę i w razie wywołania polecenia get najpierw sprawdza czy tekstura o podanych parametrach już nie istnieje, jeśli istnieje to zwraca referencję do tej już istniejącej, jeśli nie to tworzy nowy obiekt, dodaje go do struktury haszującej i zwraca referencję. Tekstura została całkowicie oddzielona od danych zewnętrznych (pozycja statku).

Prototyp



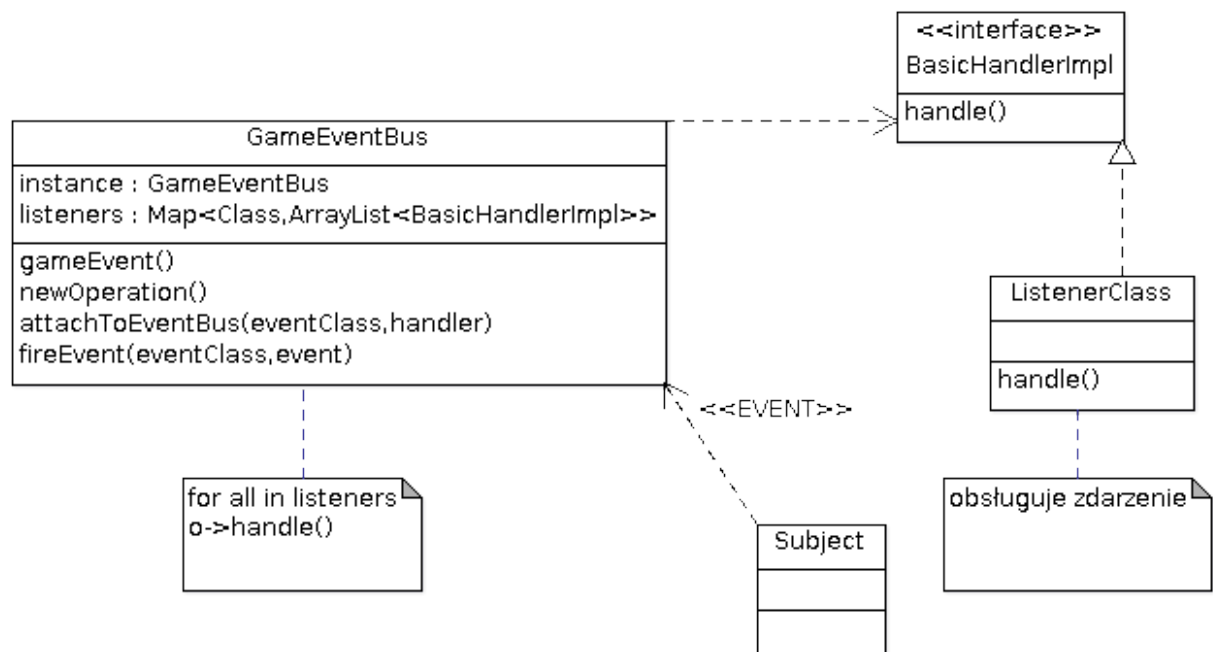
Użyty w celu uzyskania obiektów tego samego typu bez znajomości szczegółów (znamy tylko interfejs). Dodatkowo u nas w połączeniu z dekoratorem (aby sklonować udekorowane pociski). Używany przy wystrzale.

Iterator



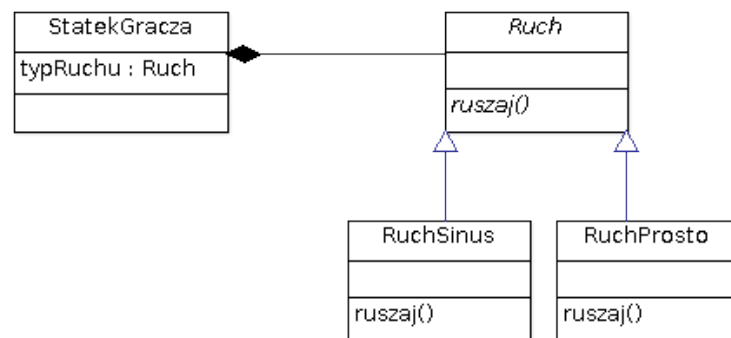
Iteratora używamy w celu unifikacji metod dostępu do danych o kolejnych poziomach. Na potrzeby projektu powstał iterator po poziomach (`levelIterator`) oraz podpoziomach (`stageIterator`).

Obserwator



Wzorca obserwator użyliśmy do ogólnej komunikacji między obiektami, silnikiem, silnikiem audio, zegarami.

Strategia



Wzorca strategia użyliśmy celem możliwości zmiany algorytmu ruchu.

Gra polega na sprawnym manipulowaniu statkiem i przedzierać się przez hordy nieprzyjacielskich statków. Za zniszczone statki wroga otrzymasz od swoich sojuszników kredyty za które możesz ulepszyć swój statek i z większą skutecznością oczyszczać galaktykę z wrogich obcych statków.