

Dokumentacja Projekt Tetris

1. Narzędzia użyte podczas powstawania projektu

Git

Jako system kontroli wersji wybraliśmy Git. Komendy których używamy podczas powstawania projektu:

- I. git init - Inicjalizuje repozytorium GIT w danym katalogu
- II. git add - Dodawanie plików do commita
- III. git commit -m "[treść_commita]" - Commitowanie zmian wraz z dołączoną wiadomością (-m)
- IV. git push origin master - Wysłanie zmian do branacha zdalnego
- V. git pull - Pobranie najnowszych zmian z aktywnego brancha zdalnego
- VI. git status - Wyświetlenie listy zmienionych plików

Jako hosting Git wybraliśmy GitHub. Link do repozytorium:

<https://github.com/THRuM/Tetris-Zesp-2>

IntelliJ IDEA

Użyte przez nas w projekcie IDE to IntelliJ IDEA jest to jedno z najlepszych IDE na rynku ponadto zapewnia ono bardzo dobrą integrację z GitHubem.

2. Dokumentacja powstałego kodu

Pakiety

Projekt został podzielony na pakiety dla lepszej organizacji kodu. Pakiet blocks zawiera klasy implementujące logikę klocków tetrisa, pakiet plane klasy implementujące logikę planszy tetrisa, pakiet users klasę User służącą do reprezentacji użytkownika.

Klasy

Podstawową klasą realizującą logikę gry jest klasa Plane znajdująca się w pakiecie pl.tetris.plane. Klasa zarządza dwuwymiarową tablicą obiektów Square, Posiada mapę odwzorowującą obiekty implementujące interfejs Block na użytkowników reprezentowanych przez klasę User oraz klasę realizującą logikę kolizji klocków implementującą interfejs CollisionDetection.

Implementacje interfejsu Block są reprezentacją klocków tetrisa. Posiadają swój kolor wyrażony w typie enum Square.

Klasa User reprezentuje użytkownika posiada niezbędne pola do zapewnienia pełnej funkcjonalności.

W projekcie istnieje ponadto klasa RandomBlockFactory której zadaniem będzie zwracanie losowych klocków przez co jest niezbędnym elementem tetrisa.

Klasa SimpleCollisionDetection implementacja interfejsu CollisionDetection jest odpowiedzialna za wykrywanie kolizji oraz zdarzenia "pełnej linii".

Interfejsy

Interfejs Block z pakietu pl.tetris.blocks wyraża zachowanie niezbędne do prawidłowego funkcjonowania klocków w systemie. Jego implementacje to: LBlock, SBlock, SquareBlock, TowerBlock.

Interfejs BlocksFactory wyraża zachowanie klas fabryk klocków. Implementacja tego interfejsu to klasa RandomBlockFactory.

Interfejs CollisionDetection definiuje zachowanie klas których zadaniem jest wykrywanie kolizji oraz wykrywanie linii które powinny zostać usunięte. Implementacją tego interfejsu jest klasa SimpleCollisionDetection.

Enum

Typ enum Square jest wyrażeniem możliwych stanów planszy (kolory, puste).

Typ enum Direction jest wyrażeniem możliwych kierunków w których może poruszać się klocek.

Type enum Rotation jest wyrażeniem możliwych kierunków obrotu klocka.

Autorzy projektu:

Maciej Cyrka, Tomasz Gołoś, Łukasz Markiewicz, Tomasz Baran