

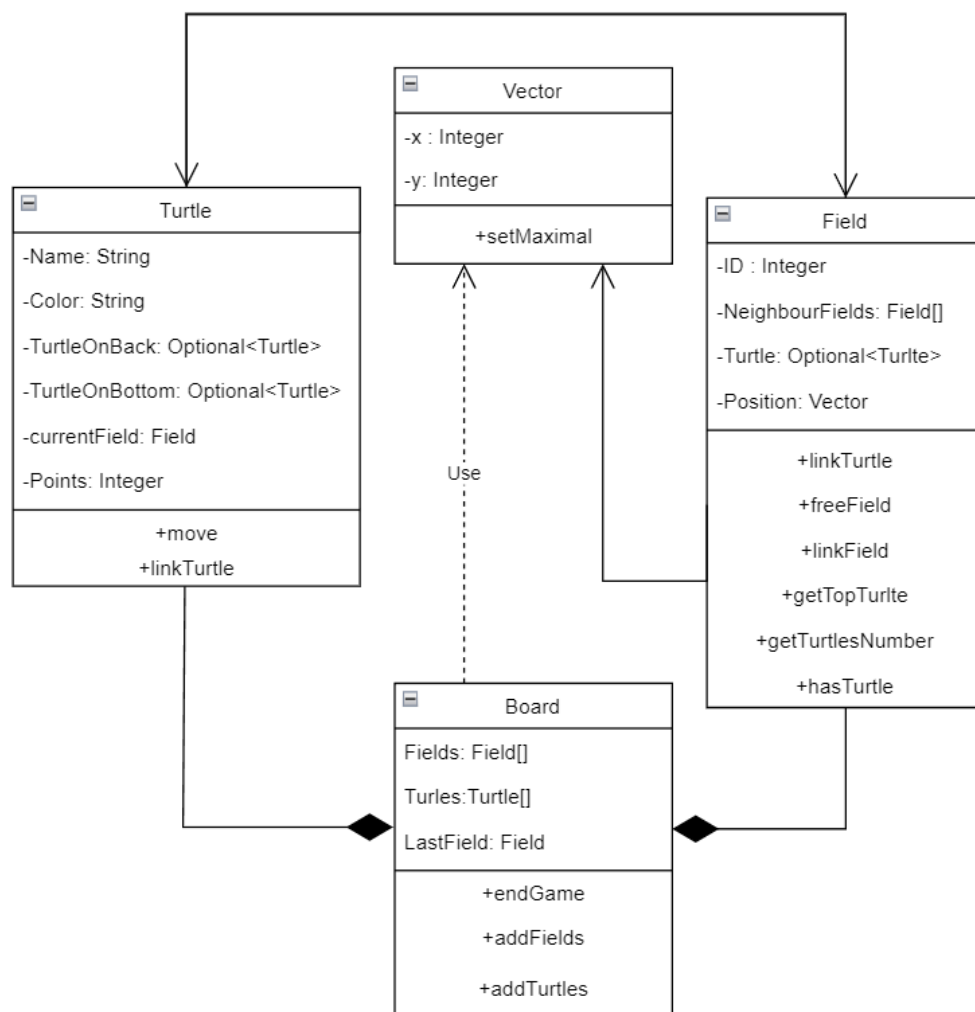
Technologie Obiektowe - projekt

Mateusz Mazur
Wojciech Łącki
Mikołaj Klimek

Milestone 1

Model

Diagram klas modelu



- **Klasa Vector** używana jest do określania pozycji pól na mapie
- **Klasa Turtle** reprezentuje żółwia, wraz z nickiem użytkownika oraz liczbą punktów, a także wskaźniki na opcjonalne żółwie, które mogą znajdować się nad lub pod tym konkretnym żółwiem

- **Klasa Field** reprezentuje pojedyncze pole, na którym przebywać mogą żółwie, pola sąsiadów, do których żółwie mogą przechodzić
- **Klasa Board** reprezentuje planszę, na której toczy się gra, zawiera listę żółwi i pól oraz wskazanie mety, pozwala zakończyć rozgrywkę po osiągnięciu mety, korzystając z warstwy persystencji zapewnia zapis do bazy danych

Gui

W Gui zebrane zostały klasy odpowiedzialne za prezentowanie widoku użytkownikowi:

- **SettingsPanel** - ekran z wstępnymi ustawieniami - liczbą żółwi i rozmiarem planszy
- **PlayersConfiguration** - ekran pobierający dane o graczach - nicki i wybrane kolory
- **BoardPanel** - ekran z planszą, po której poruszają się żółwie w czasie gry

Klasa **App** odpowiada za sterowanie widokami.

Persistence

Zawiera klasy pozwalające na zapis obiektów do bazy danych. W aplikacji zapisywane są logi z podsumowaniem gier po zakończeniu rozgrywki. Zawierają one liczbę graczy oraz pól, pseudonim i ilość punktów zwycięzcy oraz datę. Do reprezentowania logów stworzona została osobna klasa **GameLog**.

Aby umożliwić zapis do bazy wykorzystany został interfejs **GameLogRepository** rozszerzający JpaRepository ze Spring Boot.

Testy

W **TurtlesApplicationTests** zaimplementowane zostały testy sprawdzające poprawność łączenia pól, dołączania żółwi do pól oraz zapisy logów gier do bazy danych.