

Reversi

Opis

Projekt miał na celu odtworzenie gry Reversi (<https://en.wikipedia.org/wiki/Reversi>) przy pomocy silnika Unity oraz dodanie możliwości grania z AI napisanym w C++.

Projekt dzieli się na dwie główne składowe:

- AI - projekt C++ implementujący bot'a do gry
- client - projekt unity z implementacją gry w języku C# wraz z DLL bota i wszystkim potrzebnymi assetami

AI

Spis klas:

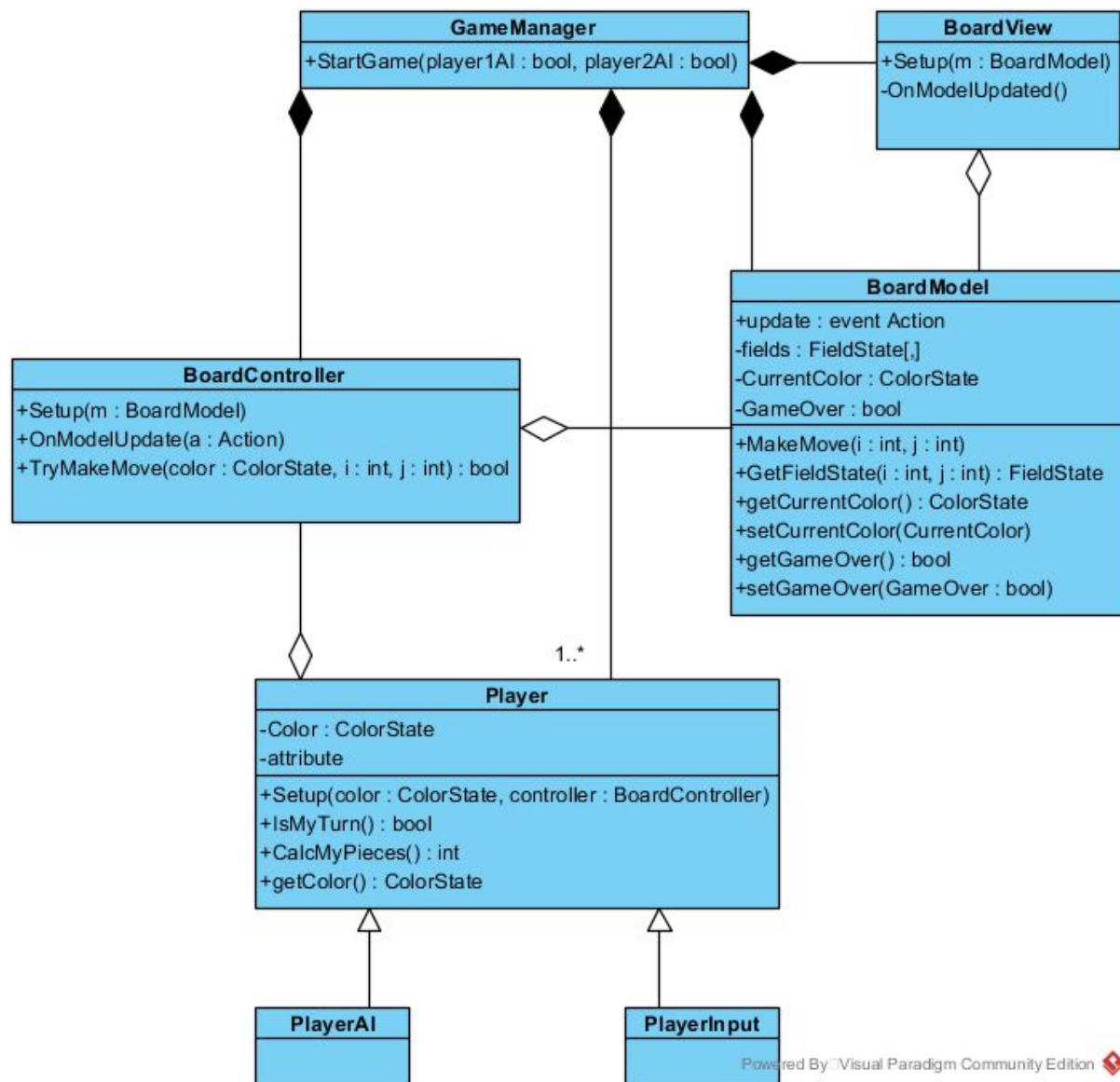
1. Bridge.cpp/h - zawiera eksportowaną metodę send, używaną z poziomu C#. Zczytuje przesłane informacje i przesyła odpowiedź
2. Communicator.cpp/h - parsuje wiadomość od klienta i odpowiada na zapytanie
3. Solver.cpp/h - oblicza optymalny ruch w grze
4. Board.cpp/h - reprezentacja stanu gry

Client

Obsługa planszy została rozbita na trzy klasy:

1. BoardModel.cs - stan planszy i pomocnicze funkcje dla kontrolera
2. BoardController.cs - wystawia API dla widoku do zmieniania modelu
3. BoardView.cs - rysuje planszę przy pomocy informacji w modelu, nasłuchuje na update event wystawiony przez model

Całością zarządza GameManager.cs. Tworzy on obiekty wyżej wymienionych klas i je setup'uje wraz z obiektami dwóch graczy. Bazowa klasa Player.cs ma dostęp do kontrolera oraz informację o tym jaki kolor w grze posiada. Udostępnia również pomocnicze metody np: do sprawdzania czyja aktualnie jest tura. Dwie klasy dziedziczące PlayerAI.cs oraz PlayerInput.cs w swój sposób korzystają z kontrolera do wpływania na stan gry. Całość można zobaczyć na wykresie UML:



Komunikacja z C++ zrobiona jest przy pomocy klas Communicator.cs i BridgeDLL.cs. Działają analogicznie do tych z C++.

Użyte design pattern'y:

1. MVC (model view controller)
2. Observer - zaimplementowany przy pomocy klasy Action
3. Komunikacja pomiędzy językami też przypomina jakiś pattern.