

WOJSKOWA AKADEMIA TECHNICZNA

METODYKI PROGRAMOWANIA .NET

Dokumentacja projektu

Warcaby – gra sieciowa

Autorzy pracy

DOBKOWSKA Milena

NOJSZEWSKI Kacper

WCY18IJ6S1

Prowadzący

mgr inż. MAŁYSZ Kamil

Data oddania projektu

11 stycznia 2020

1 Opis projektu

Tematem projektu jest sieciowa gra w warcaby. Wybraliśmy angielską wersję warcabów, czyli:

- plansza 8 na 8
- królowa może poruszać się o jedno pole w obu kierunkach
- pionki mogą bić tylko do przodu
- występuje przymus bicia

W projekcie zaimplementowaliśmy wersję gry singleplayer, gdzie użytkownik gra przeciwko prostemu algorytmowi oraz wersję multiplayer gdzie użytkownicy grają przeciwko sobie. W grze multiplayer jest dostępny chat. Wyniki gier są zapisywane w bazie danych i jest możliwość ich przejrzenia w panelu statystyk.

2 Przypadki użycia

Użytkownika ma możliwość wybrania trybu gry Multiplayer lub Singleplayer oraz może przeglądać statystyki gier. Po wybraniu gry Multiplayer użytkownik może stworzyć grę lub dołączyć do gry, w obu przypadkach jest możliwe korzystanie z czatu.

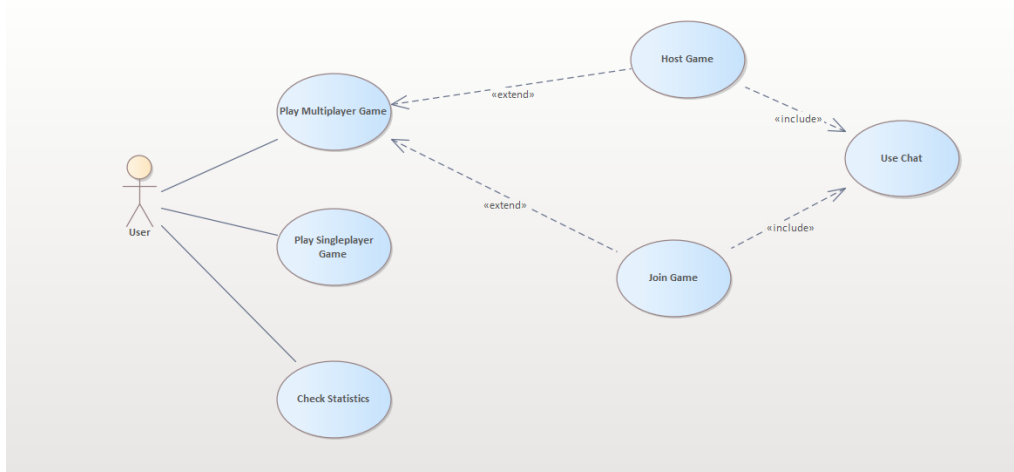


Fig. 1: Diagram przypadków użycia

3 Diagramy klas

3.1 Baza danych

Klasa DatabaseManager odpowiada za komunikację z bazą SQLite. Klasa DatabaseViewManager posiada metody wyświetlające odpowiednie rankingi w panelu statystyk. Klasy DatabaseElementSinglePlayer oraz DatabaseElementMultiPlayer

reprezentują rekordy z bazy danych. Klasa `RankingElementSinglePlayer` reprezentuje element rankingu powstający po analizie danych otrzymanych z bazy danych. Klasy `RankingElementSinglePlayerDisplay` oraz `DatabaseElementMultiPlayerDisplay` są dołączone do odpowiednich prefabrykatów, które wywołują ich metody w momencie utworzenia danego elementu przez metodę klasy `DataManager`.

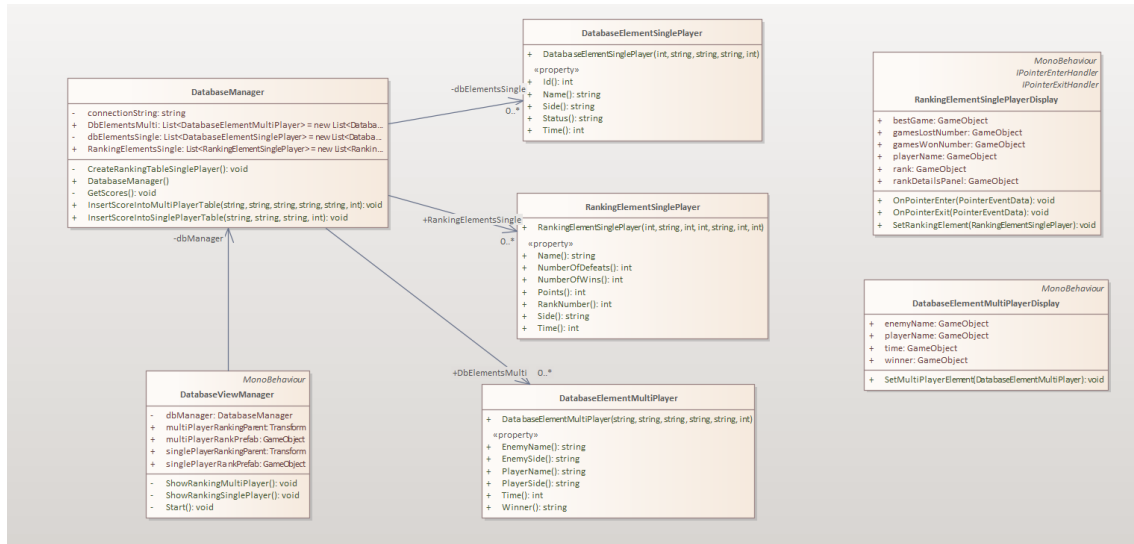


Fig. 2: Diagram klas obsługujących bazę danych

3.2 Gra

Poniżej znajdują się klasy zarządzające samą grą. Klasa `CheckersBoard` dziedziczy z `MonoBehaviour`, czyli klasy z której muszą dziedziczyć wszystkie skrypty Unity, które są połączone z obiektami gry. Zarządza ona wszystkimi graficznymi aspektami naszej gry, zawiera ona macierz obiektów klasy `Piece`, które reprezentują poszczególne pionki na planszy. Klasą w której zawiera się cała logika naszej gry jest klasa `Game`. Enum `PieceColor` jak sama nazwa wskazuje zawiera dostępne kolory dla naszych pionków. Zawiera się tu również `SimpleAlgorithm` który jest bardzo prostym algorytmem sterującym przeciwnikiem w czasie gry `singleplayer`.

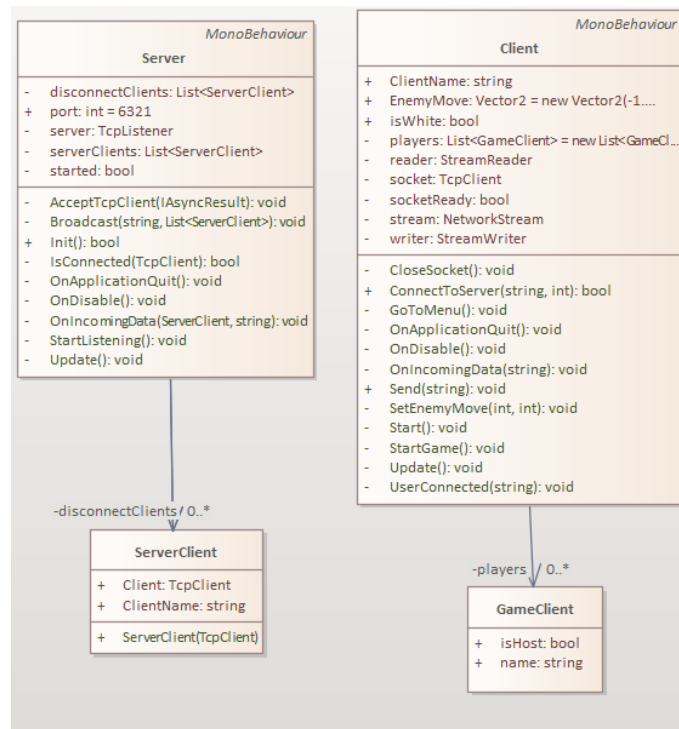


Fig. 4: Diagram klas związanych z komunikacją

3.4 Interfejs użytkownika

UIData jest klasą statyczną przechowującą informacje przekazywane pomiędzy scenami gry. Klasa UIController zawiera metody wywoływane przez przyciski menu w scenie startowej. Klasa EndGame zawiera metody obsługujące przyciski w scenie końcowej oraz zapisujące wyniki gry do bazy danych.

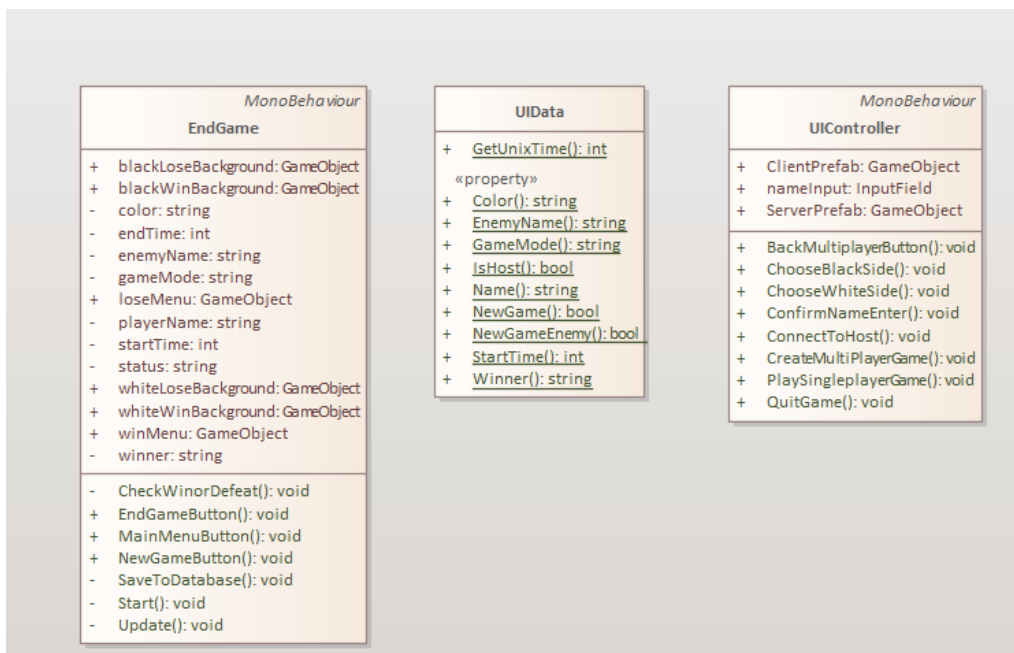


Fig. 5: Diagram klas związanych z interfejsem użytkownika

4 Komponenty

Poniżej znajduje się diagram komponentów. Pokazuje on jak działa nasza gra. Można na nim zauważyć, że komunikacja pomiędzy aplikacją, a serwerem wykonuje się przy użyciu protokołu TCP. Do generowania widoku gry wykorzystywany jest silnik Unity który generuje obraz na podstawie logiki gry. Wyniki naszej gry zapisywane są do bazy danych.

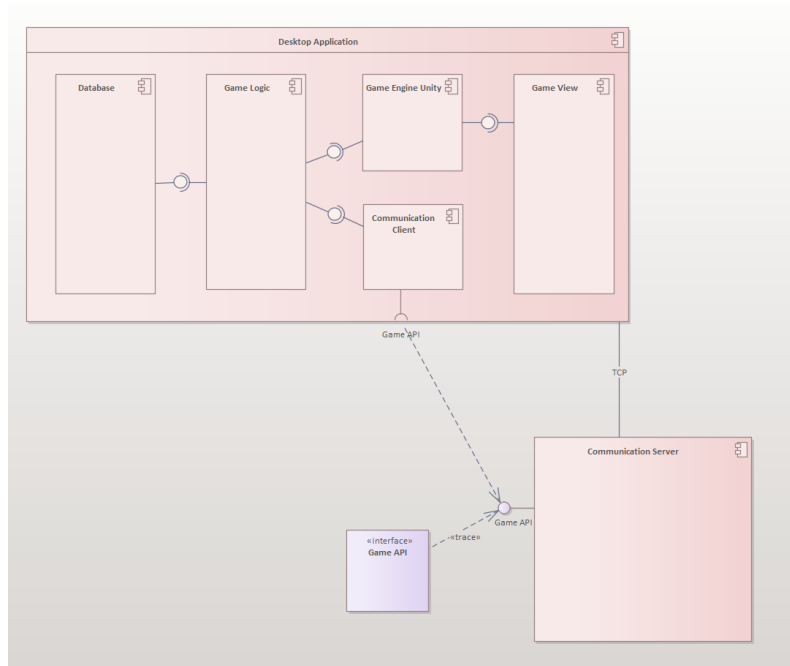


Fig. 6: Diagram komponentów

5 Wdrożenie

Gra jest umieszczana na dwóch komputerach posiadających system operacyjny Windows 10. Na jednym z komputerów który ma rolę hosta jest jednocześnie serwer i klient. Na drugim z rolą gościa znajduje się sam klient. Komunikaty pomiędzy klientami a serwerem są przesyłane za pomocą protokołu TCP.

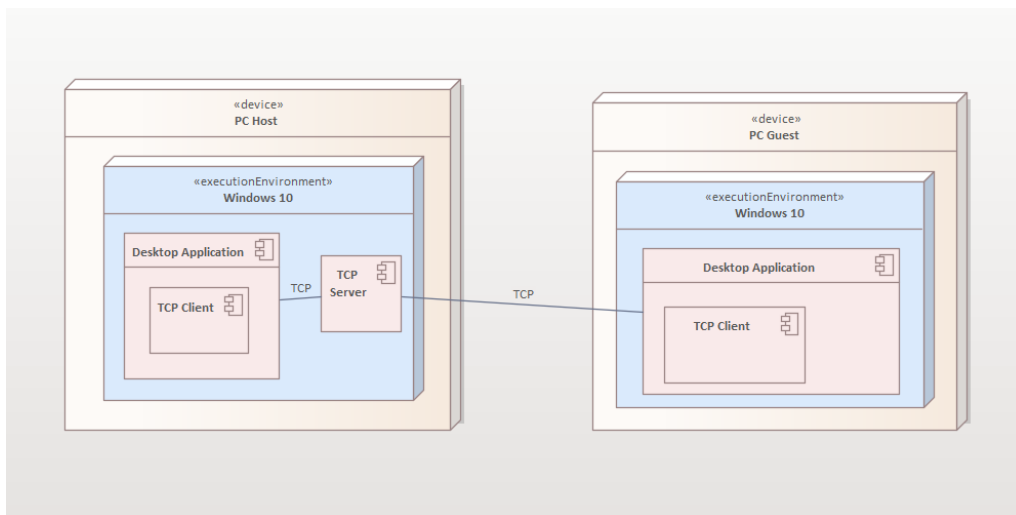


Fig. 7: Diagram wdrożeniowy