

WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ CYBERNETYKI



Dokumentacja

z zajęć laboratoryjnych z przedmiotu
Metody programowania .NET
Temat: *Warcaby sieciowe*

Dokumentację wykonali: Dominik Mucha, Damian Nadolny

Numer grupy: WCY18IJ1N1

Prowadzący: mgr inż. Kamil Małysz

Opis aplikacji Warcaby

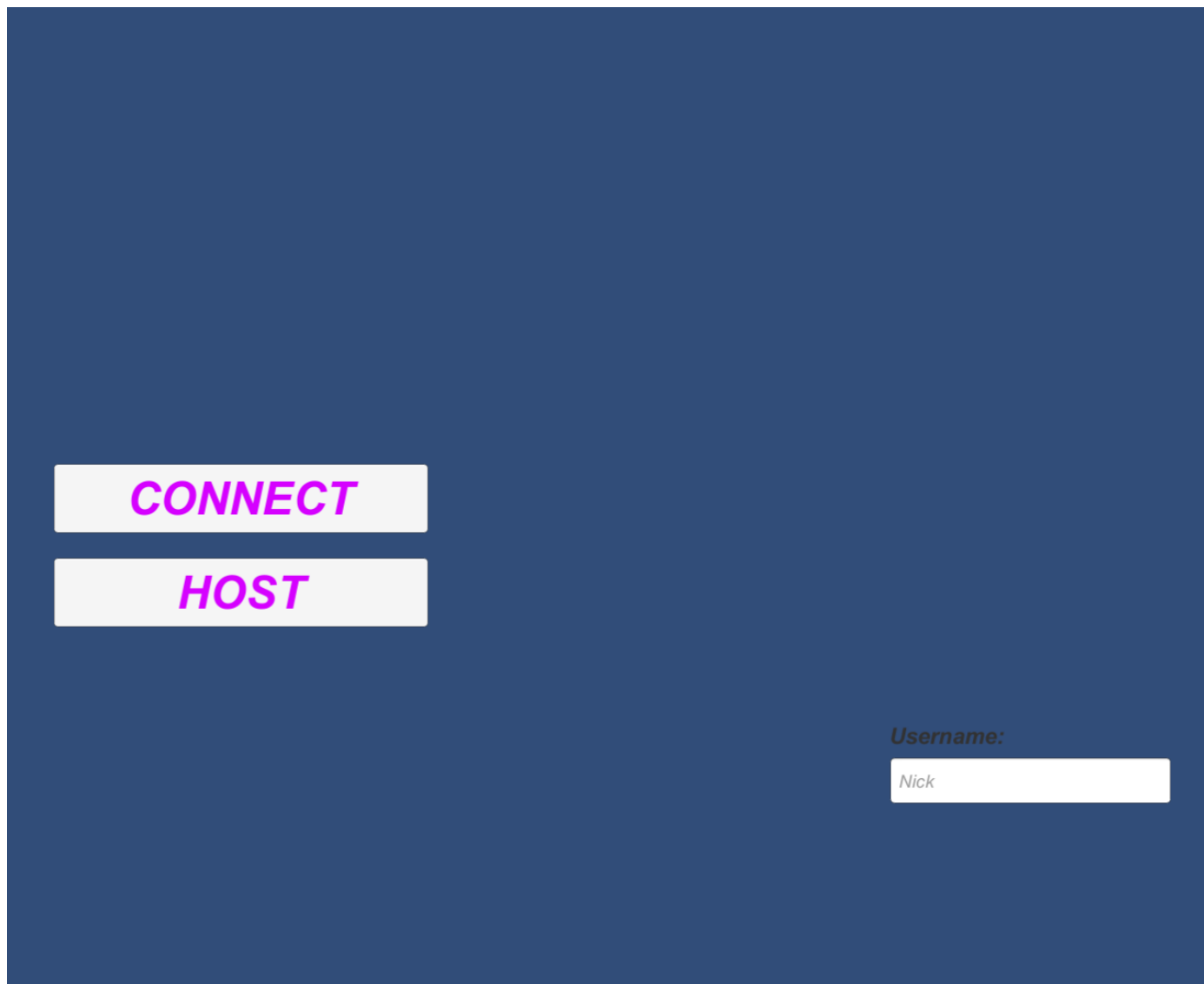
Wymagania techniczne:

System operacyjny Windows x86, myszka, klawiatura

Opis działania aplikacji:

Po uruchomieniu aplikacji pojawia się menu wyboru hosta, klienta.

Host zawsze jest pionkiem białym, klient czarnym. W prawym dolnym rogu mamy pole Username w którym możemy wpisać nazwę gracza.

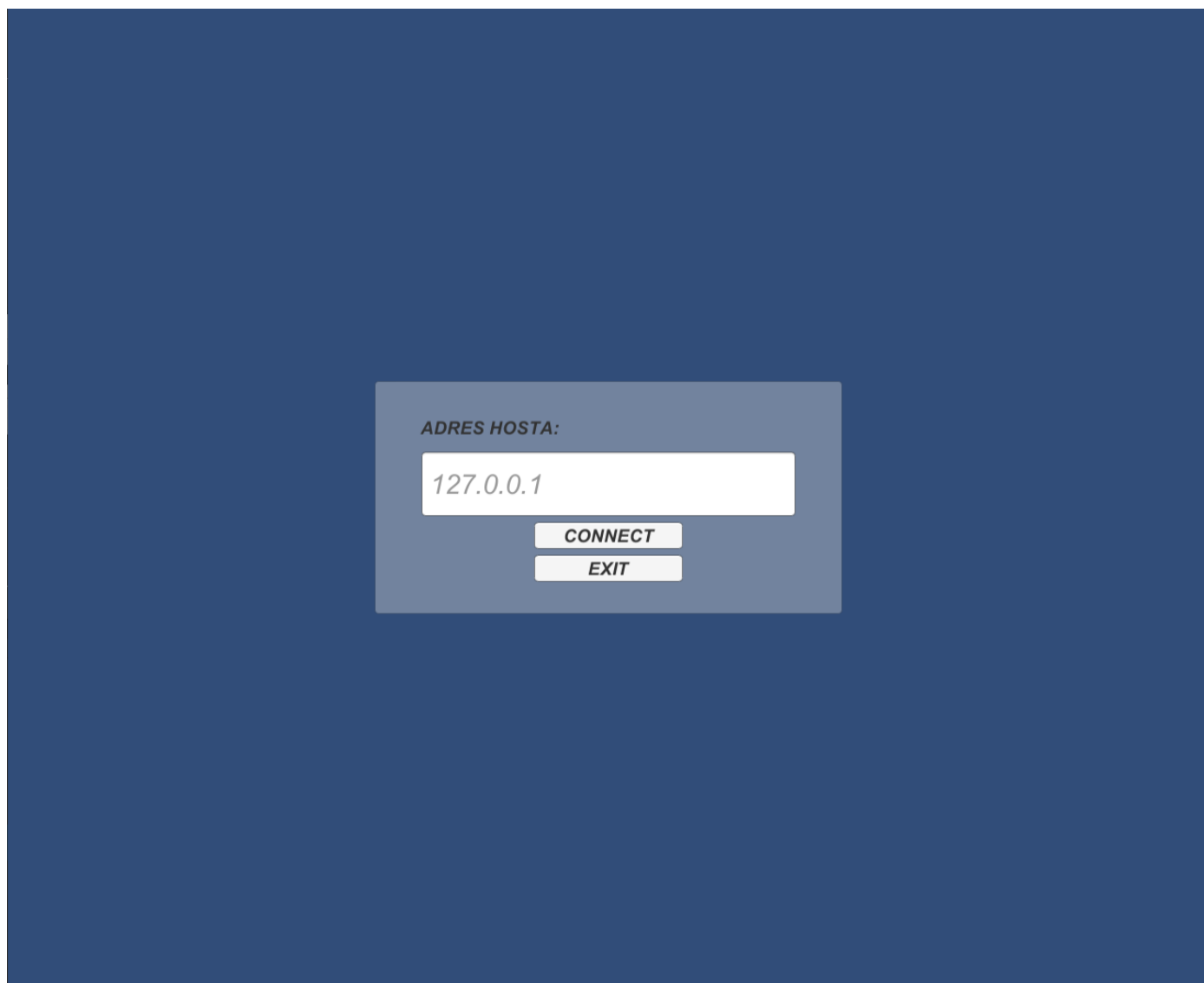


Po wybraniu Host zostajemy przekierowani do drugiego okna w którym czekamy na to aż dołączy klient przez Connect.

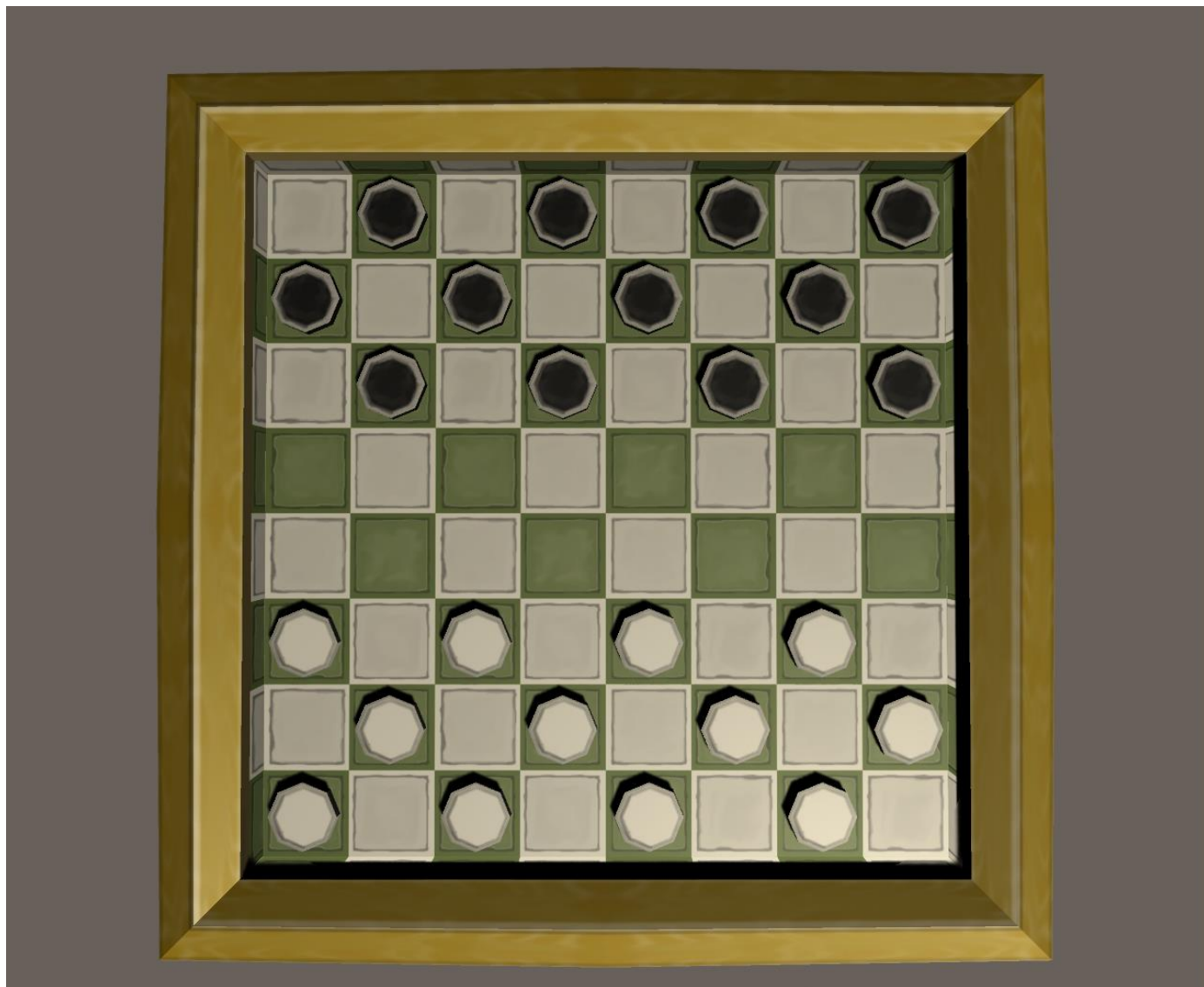
WAITING FOR ANOTHER PLAYER

EXIT

Klient po kliknięciu Connect musi podać adres hosta (w naszym przypadku jest to localhost).



Po kliknięciu Connect jesteśmy przekierowani do planszy gry (host musi być w stanie oczekującym na gracza).



Tak wygląda działająca gra.

Ruchy pionkami wykonuje się klikając a następnie trzymając konkretny pionek. W tym momencie nasz pionek się unosi. Przesuwamy na miejsce w którym chcemy go upuścić i puszczaemy przycisk myszki. W tym momencie pionek upada na wskazane pole.






- ***Wymaszana jest opcja bicia. W momencie wystąpienia owej możliwości wszystkie inne pionki są wyłączone z ruchu. Możemy się ruszyć tylko pionkiem którym aktualnie możemy bić.***
- ***Jest możliwość biciu wielu pionków na raz. Po zbiciu jednego pionka ponownie możemy ruszyć się tym pionkiem jeśli ma dalszą możliwość bicia.***
- ***Pionki mają możliwość tylko bicia do przodu.***
- ***Króla dostajemy w momencie dostania się na koniec przeciwległej krawędzi planszy. Król jest na zasadach angielskich tj. Może ruszać się w każdym kierunku, ale tylko o 1 pole, może bić również w każdym kierunku***

Opis techniczny





Aplikacja została napisana w języku C# z wykorzystaniem unity.

Aplikacja składa się z następujących plików:

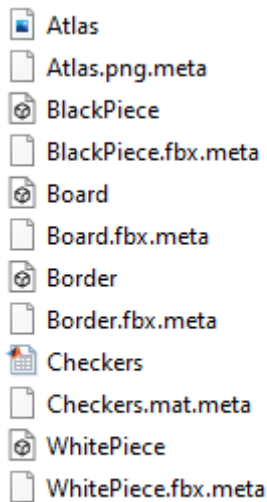
a) Pliki z kodami źródłowymi:

 CheckersBoard.cs
 Client.cs
 GameManager.cs
 Piece.cs
 Server.cs

b) Pliki ze scenami:

 Menu
 Menu.unity.meta
 SampleScene
 SampleScene.unity.meta

c) Pliki elementów graficznych:



W klasie Client mamy następujące metody:

- *Start* – Nie niszczy obiektów podczas ładowania
- *ConnectToServer* – Podłączenie do serwera
- *Update* – Odświeżanie danych
- *Send* – Przesyła dane
- *OnIncomingData* – Sprawdza czy użytkownik jest podłączony, wysyła informacje o username oraz czy jest hostem oraz sprawdza ruch na tablicy
- *UserConnected* – Dodanie klienta do listy graczy, rozpoczęcie gry jeśli liczba graczy jest równa 2
- *OnApplicationQuit* - Rozłączenie
- *OnDisable* - Rozłączenie
- *CloseSocket* – Zamykamy socket, kończymy czytanie i zapisywanie

W klasie CheckersBoard mamy następujące metody:

- *Start* – Utworzenie planszy
- *Update* - Odświeżenie danych
- *UpdateMouseOver* – Sprawdzamy położenie kamery, dzięki czemu będziemy mogli sprawdzić czy możemy coś kliknąć
- *UpdatePieceDrag* – Pionek wisi w powietrzu jak go klikniemy i trzymamy
- *SelectPiece* – Wybranie pionka
- *TryMove* – Próba ruchu, możliwość bicia
- *EndTurn* – Zakończenie ruchu
- *CheckVictory* – Sprawdzanie kto wygrał
- *Victory* – Informacja o tym kto wygrał
- *ScanForPossibleMove* – Stworzone dwie listy z czego jedna jest przeciążeniem drugiej, dodanie pionki do listy wymuszającej ruch
- *GenerateBoard* – Generowanie planszy

- *GeneratePiece* – Generowanie pionków
- *MovePiece* – Ruch pionka

W klasie *Piece* mamy następujące metody:

- *isForceToMove* – Sprawdzanie możliwości bicia
- *ValidateMove* – Sprawdza poprawność ruchu

W klasie *GameManager* mamy następujące metody:

- *Start* – ustawienie scen startowych
- *ConnectButton* – Przycisk przejścia do sceny dla Klienta
- *HostButton* – Przycisk przejścia do sceny dla Hosta
- *ConnectToServerButton* – Połączenie się w scenie Klienta
- *BackButton* – Przycisk powrotu do głównej sceny
- *StartGame* – Po połączeniu rozpoczyna się gra

W klasie *Server* mamy następujące metody:

- *Init* – Tworzenie servera
- *Update* – Aktualizacja danych o tym czy użytkownik jest dalej połączony
- *StartListening* – Rozpoczęcie połączenia
- *AcceptTcpClient* – Akceptacja użytkownika
- *IsConnected* – Sprawdzanie czy Klient jest połączony
- *Broadcast* – Stworzone dwie metody z czego jedna odpowiada za funkcję nadawania a druga jest przeciążeniem pierwszej.
- *OnIncomingData* – Odbieranie danych na serwer

Diagram Klas

