

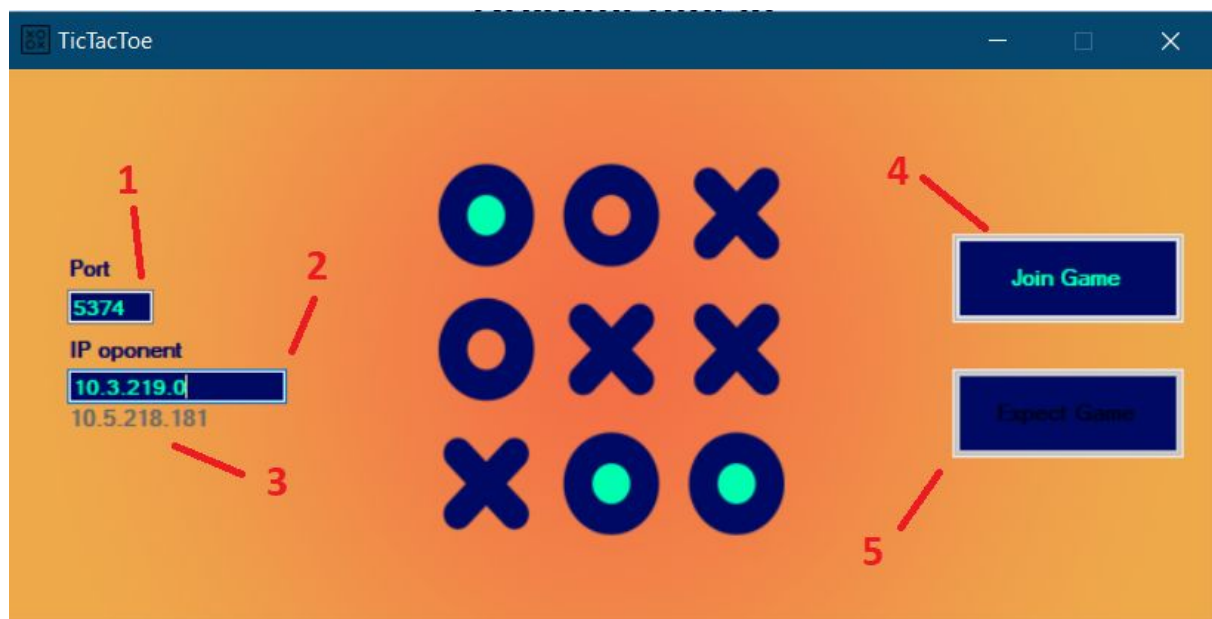
Dokumentacja

Aplikacją jest gra TicTacToe z możliwością grania online z innymi użytkownikami.

Startowe okienko zawiera kontrolki takie jak:

1. **TextBox: Port,**
2. **TextBox: IP,**
3. **Label: Moje IP,**
4. **Button: Join,**
5. **Button Except for Game.**

oraz pozostałe informacyjne.



Kontrolka:

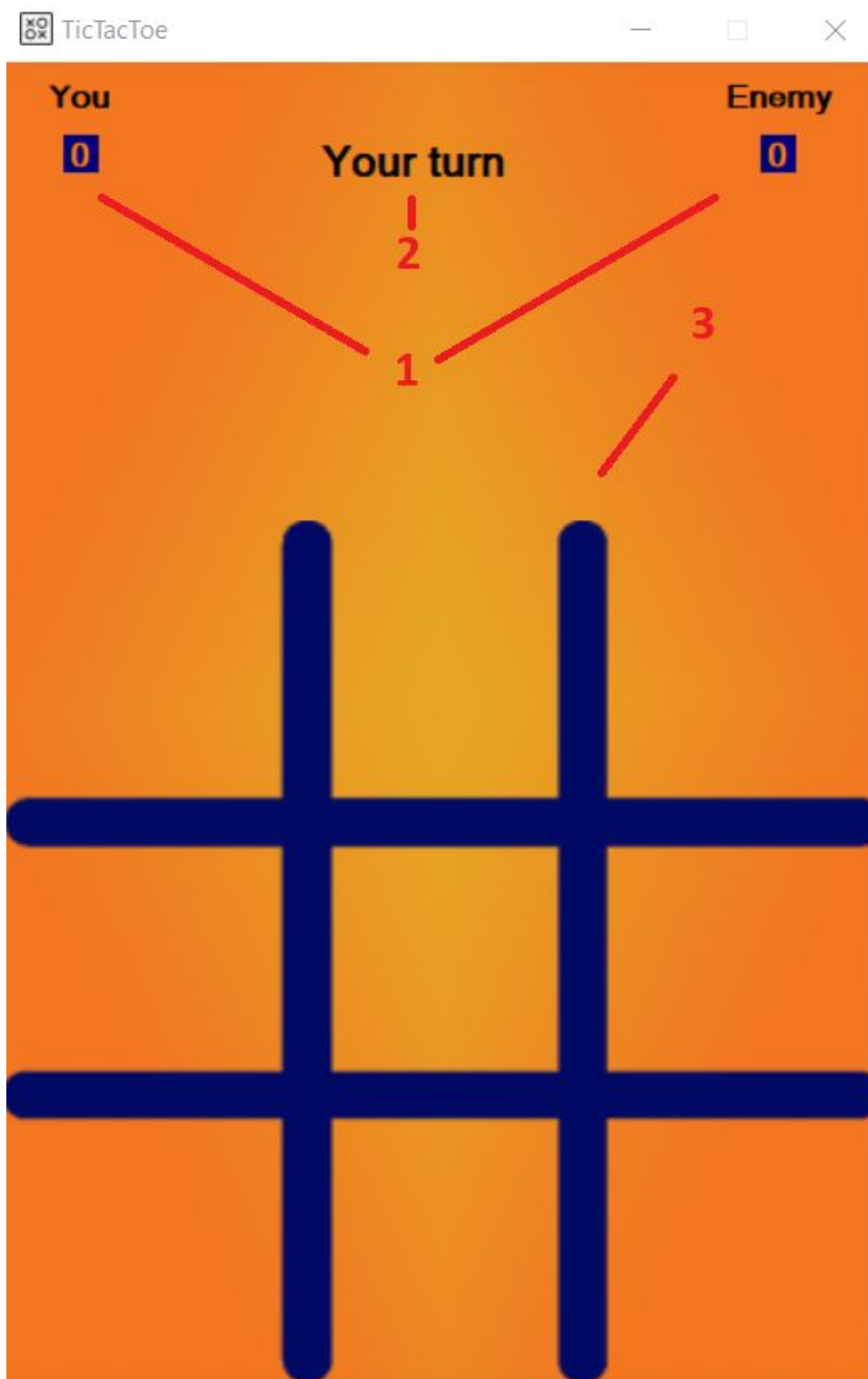
4. Powoduje połączenie się z komputerem o podanym IP na wskazanym Porcie.

5. Powoduje nasłuchiwanie na podanym Porcie z lokalnym IP. W momencie, gdy jakiś klient połączy się z Twoją aplikacją, zachodzi inicjalizacja właściwości gry.

Oba przyciski są na bieżąco walidowane nie pozwalając na wprowadzenie błędnych danych lub danych, które są zbędne.

Po pozytywnej inicjalizacji zostaje uruchomione drugie okno (wyłączając poprzednie).

1. **Label: Punkty poszczególnego gracza,**
2. **Lanel: Informacje o tym kolejności tury,**
3. **Buttons: Tablica gry.**



Formy Informacyjne wyświetlające dynamiczne dane:

Waiting for game...

// Atakujący zawsze otrzymuje *krzyżyk* (agresywny znacznik).

Ruchy gracza następują jeden po drugim, do momentu rozstrzygnięcia rozgrywki.

Po jej ukończeniu wyskakuje komunikat o stanie rozgrywki, cała tablica zostaje przygotowywana do następnej rozgrywki a w przypadku wygranej jednej z osób zwiększona zostaje jego sumeryczna wartość punktów. Po zaakceptowaniu stanu rozgrywki, nowa zostaje rozpoczęta. W tym momencie zostają uruchomione systemu walidacyjne nieprzerwanie badające stan aplikacji odpowiednio reagując.

Krótki opis kodu

Program został odpowiednio zaprojektowany według mnemonika SOLID aby w konieczności poprawek, rozszerzeń były możliwe zmiany w jak najszybszym czasie. Kod został podzielony na warstwy, każde klasy na interfejsy (dla możliwości mockowania przy testach jednostkowych), klasy jedno-instancyjne statyczne. System został zaprogramowany w kontener IoC (Autofac) aby móc zarządzać siecią instancji obiektów oraz jego niechcianych mnożnych wywołań.

Zachowanie kodu jest dynamiczne. W zależności od aktualnego UI (edycji kontrolek, ich ilości itd) przystosowuje się do jego stanu zmieniając logikę kodu.

Części kodu, w których możliwy byłby jego dłuższy przepływ zostały zaprojektowane w systemie współbieżnego programowania. Zachowana została przy tym spójność oraz odpowiednia wymiana danych, ich edycja czy wywołania na wątkach pobocznych, aby asynchronizacja zdarzeń procesu nie spowodowała ThreadException-ów (które jak wiadomo, są uciążliwe i często nie do odratowania).

// Do komunikacji między graczami została użyta biblioteka Socker używająca połączeń Tcp, gdyż są one najlepszym wyborem dla owych rozgrywek (szybkość, bezpieczeństwo, zarządzalność).