Tic Tac Toe – Eine Fingerübung

Mit diesem Programm wird das altbekannte Tic Tac Toe- Spiel oder auch X X O realisiert. Hierbei spielen zwei Spieler gegeneinander an einem PC. Spieler 1 hat als Symbol das X und bedient die Maus. Spieler 2 hat als Symbol das O und bedient auf der Tastatur das NumPad. Hierbei stehen die Zahlen 1 bis 9 für die neun Felder auf dem Spielfeld. Das heißt, wenn Spieler 2 die 7 drückt, wird das Feld oben links markiert. Hat ein Spieler seine Reihe vervollständigt, wird diese Reihe mit einer roten Linie markiert und eine Meldung signalisiert den Gewinn. So lange das Programm läuft, wird der aktuelle Spielstand in der Menüleiste angezeigt.

1. GUI

Das GUI wurde in WPF geschrieben. Das Spielfeld ist ein 3 mal 3 Grid, in dem jedes Feld mit einem Button ausgefüllt ist. Darüber liegt eine Menüleiste mit einem Button »Neu« um das Spiel neu zu starten und drei Label, die den aktuellen Spielstand anzeigen. Dazu ist die Content-Eigenschaft des Labels an die Property »Wins« der Spieler-Klasse gebunden.

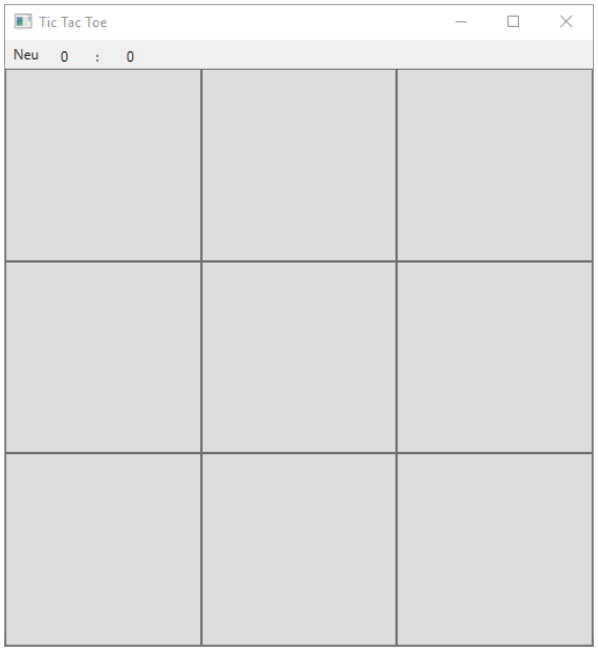

Abb. 1 GUI Start

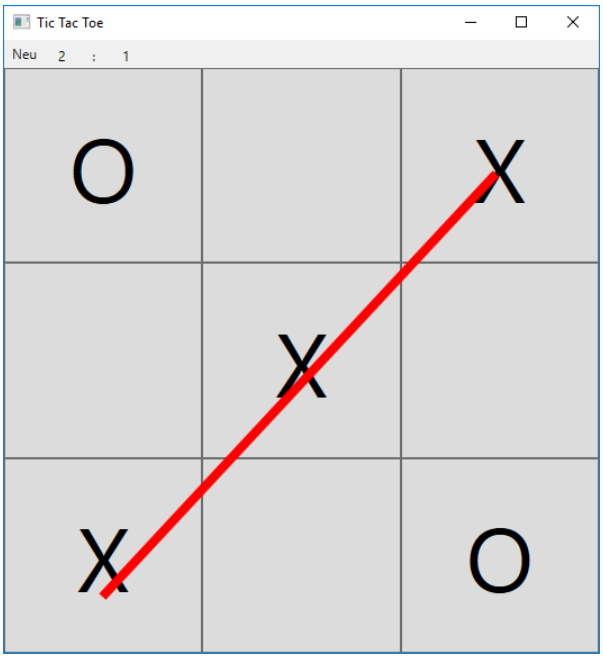

Abb. 2 GUI Spieler 1 hat gewonnen

2. Spieler-Klasse

Die beiden Spieler werden in der Anwendung als Objekte abgebildet.

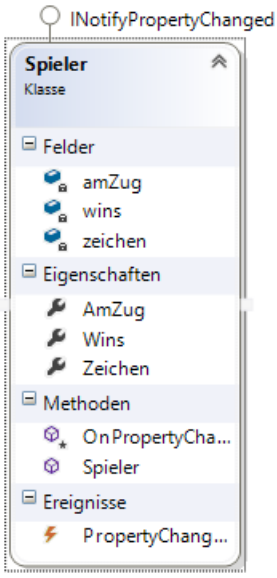

Abb. 3 Klasse Spieler

Die Property *AmZug* ist vom Typ *bool* und wird genutzt, um zu signalisieren, dass der Spieler einen Zug machen darf. Ist der Wert »*false*«, wird das Click-Event der Buttons (für Spieler 1) bzw. das KeyDown-Event (für Spieler 2) nicht ausgelöst.
Wins ist ein Integerwert, der mit 0 initialisiert wird und bei jedem Gewinn inkrementiert wird.
Zeichen ist das Spielerzeichen als *String*, also X oder O.

Die Klasse erbt von der Schnittstelle *INotifyPropertyChanged*, um die Änderung des Wertes von *wins* an das Label weiterzugeben.

3. Code Behind

Beim Initialisieren des *MainWindow* Objekts werden auch die beiden Objekte vom Typ *Spieler* angelegt.
Wenn ein Spieler seinen »Zug« macht, wird dieser, je nach Spieler, in der Methode *Button_Click* bzw. *Window_KeyDown* ausgeführt.
In *Button_Click* wird geprüft, ob Spieler 1 tatsächlich an der Reihe ist und ob das gewählte Feld keinen Inhalt hat. Wenn beide Bedingungen erfüllt sind, wird der *Content*-Eigenschaft des Buttons das Zeichen von Spieler 1 (das X) als String übergeben und damit der nächste Spieler an der Reihe ist, wird die Methode *nextTurn* aufgerufen.
Window_KeyDown hat die gleiche Aufgabe, nur dass hier eine Taste des NumPads dem Button zugeordnet werden muss (Taste 6 → Button in der Mitte Rechts).
Nachdem der gewünschte Button mit dem Zeichen des Spielers markiert wurde, wird die Methode *nextTurn* aufgerufen. Hier wechselt die boolsche Eigenschaft *AmZug* seinen Wert. Damit kann derselbe Spieler nicht zwei Züge hintereinander machen.
Nach jedem Zug wird außerdem geschaut, ob die Siegesbindung eingetreten ist. Dafür ist die Methode *checkForVictory* zuständig, die prüft, ob drei Mal dasselbe Zeichen in einer Reihe vorhanden ist.
Hat ein Spieler gewonnen, wird der Nutzer mit einer MessageBox darüber benachrichtigt.
Zum Abschluss wird eine Linie durch die drei Felder gezogen, mit der ein Spieler gewonnen hat. Dazu werden Start- und Endfeld der Methode drawWinningLine übergeben. Die Methode zeichnet dann eine Linie von Start zum Endpunkt, in dem das Anwendungsfenster vorher gesechstelt wird. So entspricht die erste Reihe ⅓ des Feldes, die zweite 3/6 (½) usw.

Da diese Technik nicht für vertikale Linien funktioniert, wurde die Methode überladen und erwartet nur einen Wert, und zwar die Spalte, in der die Linie gezeichnet werden soll.

Haben die Spieler Lust auf eine neue Runde, können sie in der Menüleiste auf »Neu« klicken. Dann wird die Methode *Neu_Click* aufgerufen, in der die Content-Eigenschaft aller Buttons und die Linie gelöscht wird.