

Java Projekt: Tic Tac Toe + GUI

Rechnerpraktikum aus Programmierung im WS 2005/06
unter der Leitung von Dr. Michael Hahsler

Tadeh Amirian (h0253821)
10.01.2006

1) Problemdefinition

Beim TicTacToe-Spiel versuchen zwei Personen das Spiel zu gewinnen, indem sie abwechselnd Kreuz-Figuren (Spieler A) und Kreis-Figuren (Spieler B) auf einem 3x3-Brett so platzieren, dass ihre Figuren in einer Horizontalen, Vertikalen oder Diagonalen eine 3er- Reihe bilden. Anfangs ist das Brett leer. Das Spiel endet mit einem Unentschieden, wenn alle Felder des Brettes besetzt sind und kein Spieler eine 3er-Reihe bilden konnte.

Zusätzlich zur Spiellogik sollte noch ein Graphisches User-Interface implementiert werden.

2) Analyse

a) Spielablauf

Anzahl der Spieler: 2
Spielfeld: 3x3 Matrix
Beginner: Kann eingestellt werden
Gewinner: eine der folgenden Situationen:

x		x		x					
					x		x		x

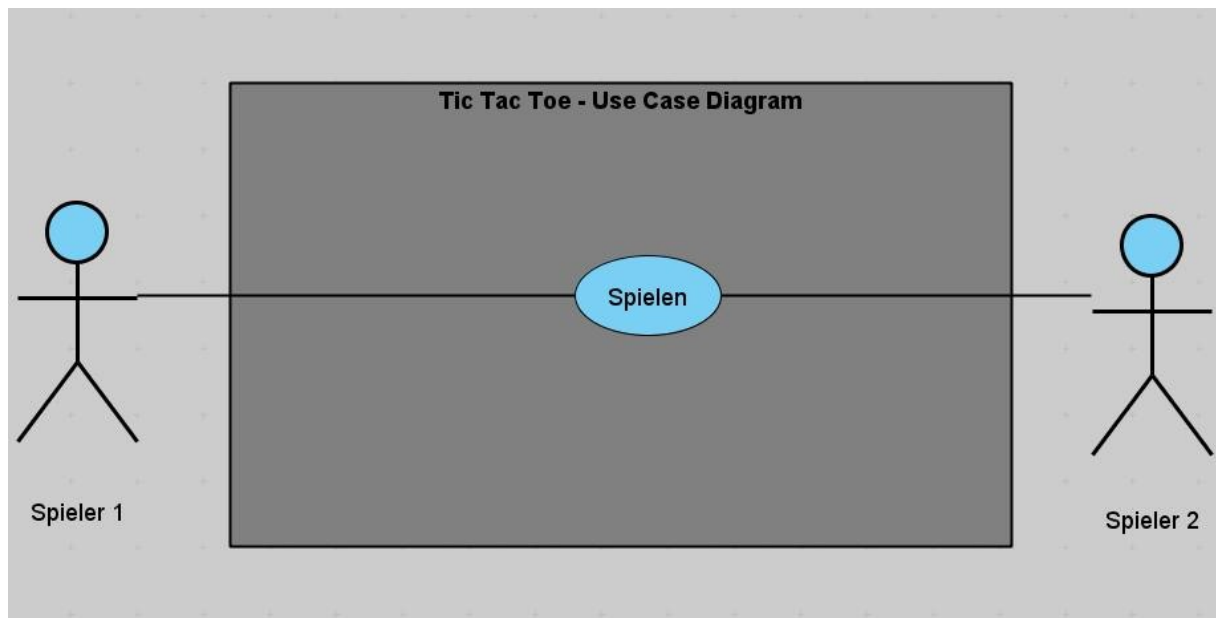
					x				
x		x		x			x		x

				x	x				
x		x			x				
					x				

		x							x
		x							x
		x							x

Unentschieden: "X" und "O" sind 9 mal in der Matrix enthalten

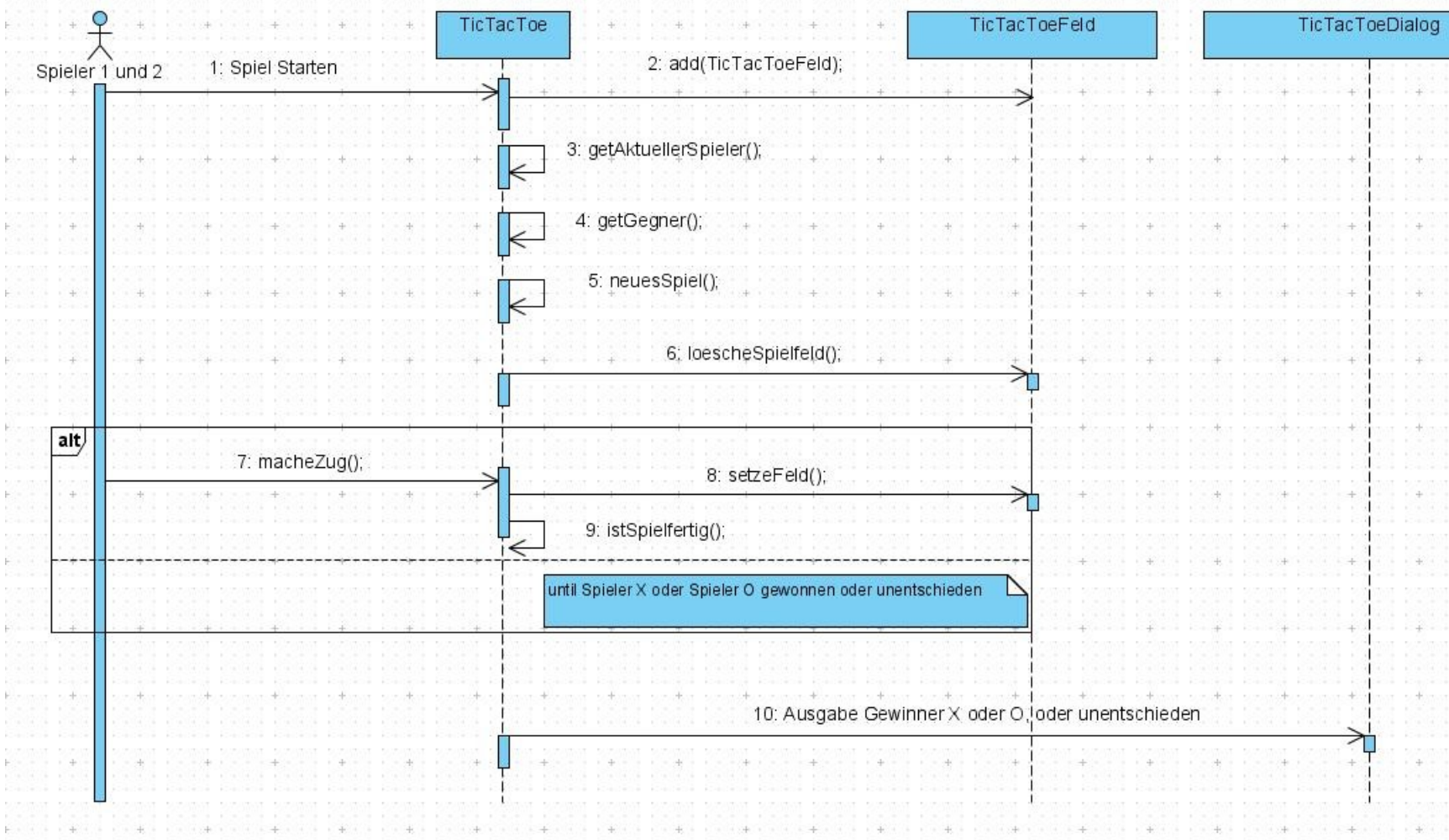
b) Use-Case-Diagramm



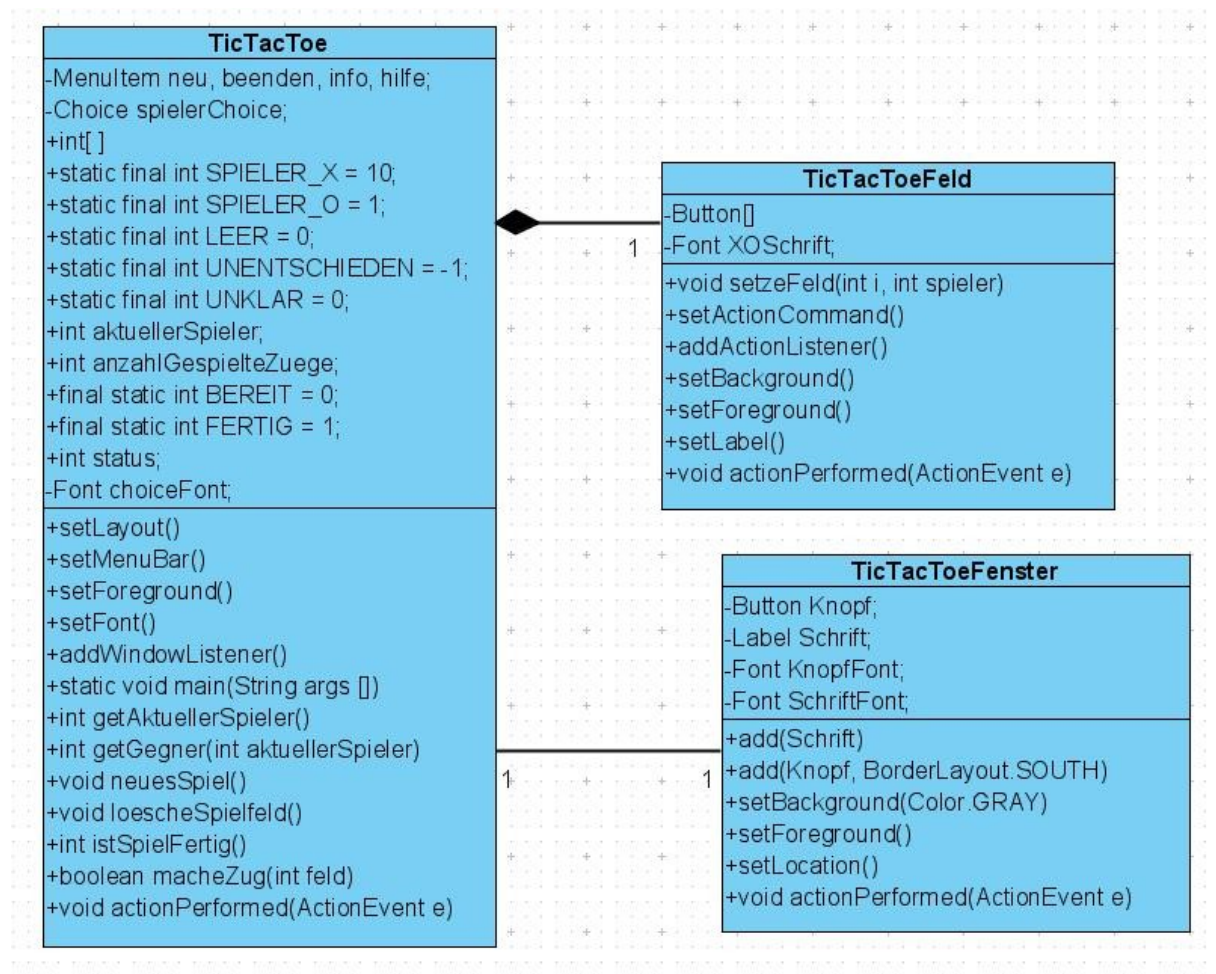
3) Design

Sequenzdiagramm:

sd Sequence Diagramm TicTacToe



Klassendiagramm:



Klassen im Detail:

TicTacToe

Importiert werden:

```
import java.awt.*;
import java.awt.event.*;
```

Variablen:

Menulitem neu, beenden, info, hilfe;
→ Menü-unterpunkte

Choice spielerChoice;
→ Wahlfeld für Beginner

```
public int[] spielfeld;
```

→ Spielfeld wird deklariert: Die Felder sind von links oben nach rechts unten mit 0 bis 8 nummeriert

```
|0|1|2|  
|3|4|5|  
|6|7|8|
```

Codierung der Spielfeldeinträge:

```
public static final int SPIELER_X = 10;  
public static final int SPIELER_O = 1;  
public static final int LEER = 0;  
public static final int UNENTSCHIEDEN = -1;  
public static final int UNKLAR = 0;
```

```
public int aktuellerSpieler;  
public int anzahlGespielteZuege;
```

```
public final static int BEREIT = 0;  
public final static int FERTIG = 1;  
public int status;
```

→ Wichtig für den Spielfortschritt und Gewinnerermittlung

Methoden:

```
TicTacToe();
```

→ Baut GUI auf. Definiert Menü und Menüitems + fügt Ergebnishandlung zu den Menüs hinzu. Definiert das Spielfeld und initiiert auch die Wahlmöglichkeit für den beginnenden Spieler.

```
public static void main(String args[])
```

→ Das ist die Mainmethode, die eine Instanz von TicTacToe erzeugt und mit neuesSpiel() startet.

```
public int getAktuellerSpieler()
```

→ der aktuelle Spieler wird zurückgegeben

```
public int getGegner(int aktuellerSpieler)
```

→ Methode, um den Gegner jeweils zu definieren.

```
public void neuesSpiel()
```

→ startet neues Spiel

```
public void loescheSpielfeld()
```

→ löscht gesamtes Spielfeld (inklusive dem GUI-Spielfeld)

```
public int istSpielFertig()
```

→ prüft ob einer der Spieler gewonnen hat und gibt jenen auch zurück. Falls keiner der Spieler gewonnen hat, wird geprüft ob das Spiel unentschieden ist, oder ob noch Felder frei sind. (dann wird UNKLAR ausgegeben und das Spiel geht weiter)

```
public boolean macheZug(int feld)
```

→ wenn Zug durchgeführt werden kann wird true zurückgegeben. Wenn der Zug möglich ist, wird der Zug gemacht. Der Gegner wird bestimmt und es wird geprüft ob der Spieler gewonnen hat, das Spiel unentschieden ausgegangen ist, oder ob nun der Gegner am Zug ist. Falls es einen Gewinner gibt, oder ein Unentschieden der Fall ist, wird ein Fenster geöffnet in dem das Ergebnis steht.

```
public void actionPerformed(ActionEvent e)
```

→ Deklariert die Ereignishandlungen für das Spielfeld + Menü

TicTacToeFeld

Importiert werde:

```
import java.awt.*;  
import java.awt.event.*;
```

Variablen:

```
Button[] Schaltflaeche;  
→ Die Buttons für das Spielfeld.
```

Methoden:

```
TicTacToeFeld();  
→ das Layout für das Spielfeld wird deklariert. Die 9 Buttons werden erzeugt ( + Ereignisbehandlung)
```

```
public void setzeFeld();  
→ je nachdem welcher Spieler das Feld gewählt hat, wird entweder ein „X“ oder ein „O“ gesetzt. (bei feldlöschen, wird ein leerzeichen gesetzt)
```

```
public void actionPerformed(ActionEvent e);  
→ Der Zug wird auch mit TicTacToe.macheZug ausgeführt!
```

TicTacToeFenster

Importiert werden:

```
import java.awt.*;  
import java.awt.event.*;
```

Variablen:

private Button Knopf;

private Label Schrift;

Methoden:

TicTacToeFenster();

→ Es wird die Location und die Background-Farbe eingestellt. Die GUI – Elemente werden erzeugt und dem Fenster hinzugefügt.

public void actionPerformed(ActionEvent e)

→ Ereignisbehandlung wird definiert.

4) Wartung

Sollte ein Fehler auftreten, bitte eine Mail mit Beschreibung an mich senden:

h0253821@wu-wien.ac.at