Praktikum aus Programmierung
Dr. Michael Hahsler
WS 2006/2007

Dokumentation des Projektes
Tic Tac Toe

Zuzana Gajdosova
0050230

Zuzana Gajdosova
0050230

Java Projekt: Tic Tac Toe

1. Inhaltsverzeichnis

2. Problemdefinition

Das Ziel des Projektes ist das Spiel „TicTacToe” zu implementieren. In diesem Spiel spielen gegeneinander 2 Spieler, die abwechselnd ihre Zeichen (Kreis/Kreuz) auf einem leeren 3x3 Brett platzieren. Wenn es einem der Spieler gelungen ist, 3 Felder waagrecht, senkrecht oder diagonal zu besetzen, hat er sofort gewonnen. Sind alle Felder belegt und kein Spieler konnte 3 Felder waagrecht, senkrecht oder diagonal besetzen so endet das Spiel unentschieden

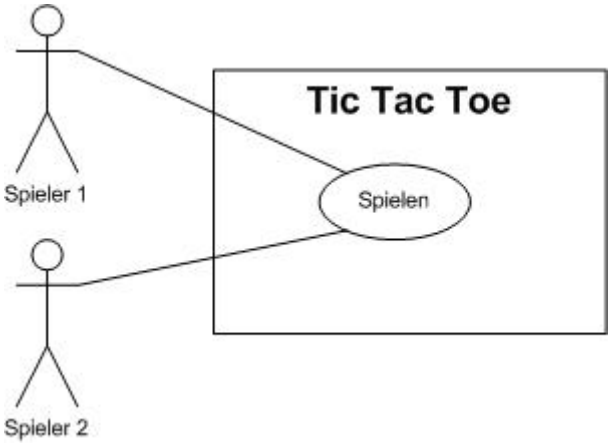
Zuzana Gajdosova
0050230

3. Modellierung

2.1. Beschreibung des Use Case Diagramms:

Das Use Case “Spielen“ stellt den Spielablauf dar. 2 Spieler setzen abwechselnd Züge auf freie Spielfelder bis einer gewinnt oder das Brett voll besetzt ist. Um zu gewinnen müssen 3 Felder waagrecht, senkrecht oder diagonal besetzt werden ansonsten endet das Spiel unentschieden.

Use Case Diagramm:



2.2. Beschreibung des Sequenzdiagramms:

Das Sequenzdiagramm besteht aus den Instanzen *Player, TicTacToe, Field* und *Dialog.*
Im Folgenden wird der zeitliche Ablauf der Nachrichten zwischen diesen Instanzen näher erläutet:

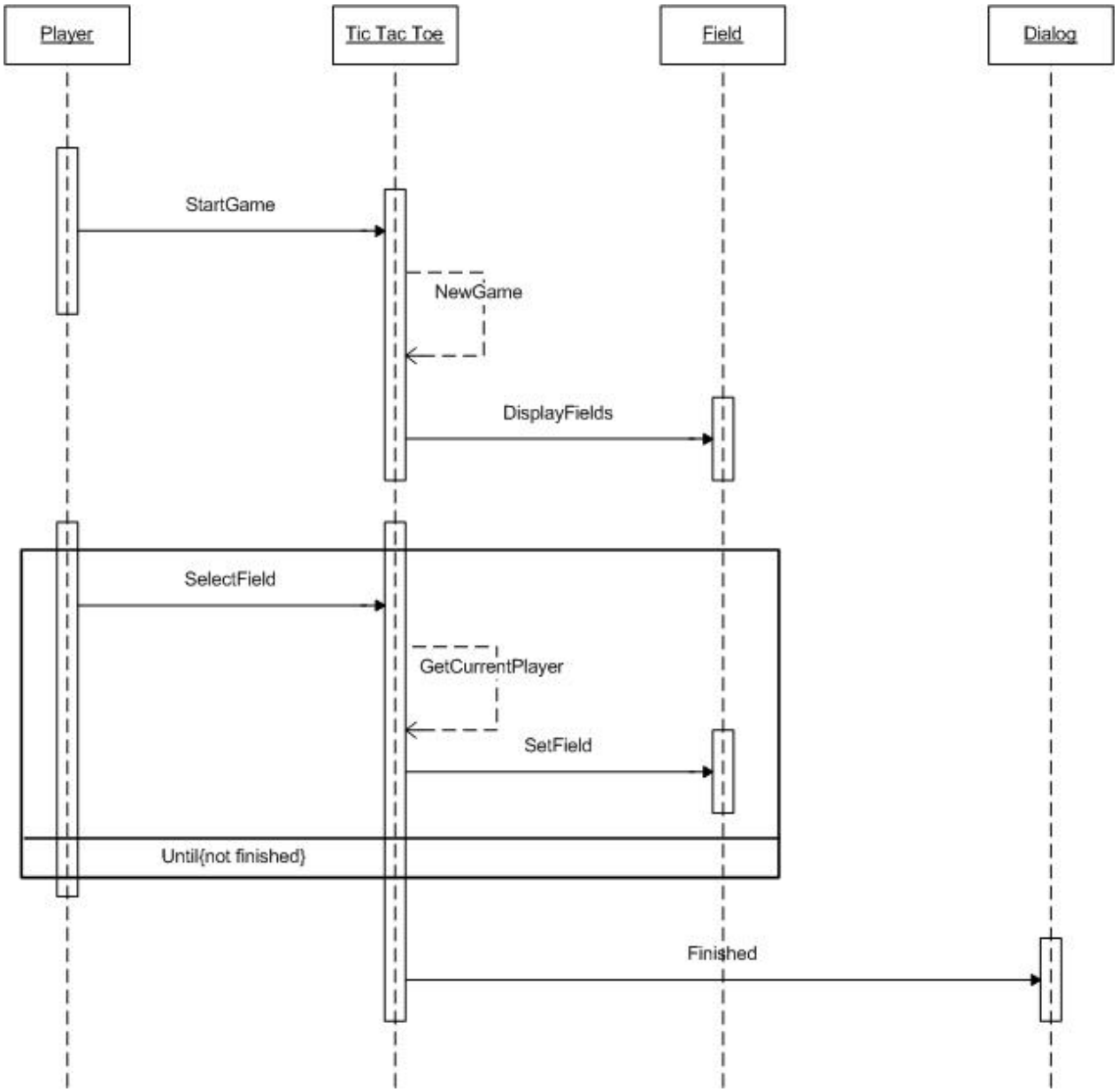
- Objekt Player sendet die Nachricht *StartGame* (= ein neues Spiel soll gestartet werden) auf das Objekt TicTac Toe
- daraufhin sendet Objekt TicTacToe die Nachricht *StartGame* auf sich selbst um eine Nachricht *DisplayFields* (=erzeuge das Spielfeld) an das Objekt Field zu schicken
- weiters sendet das Objekt Player eine Nachricht *Select Field* (=ein Feld wird vom Spieler ausgesucht) an das Objekt TicTac Toe, das wieder sich selbst delegiert mit der

Zuzana Gajdosova
0050230

Nachricht *Get Current Player* und eine Nachricht *Set Field* (=speichere das Zeichen des aktuellen Spielers) an das Objekt Field sendet

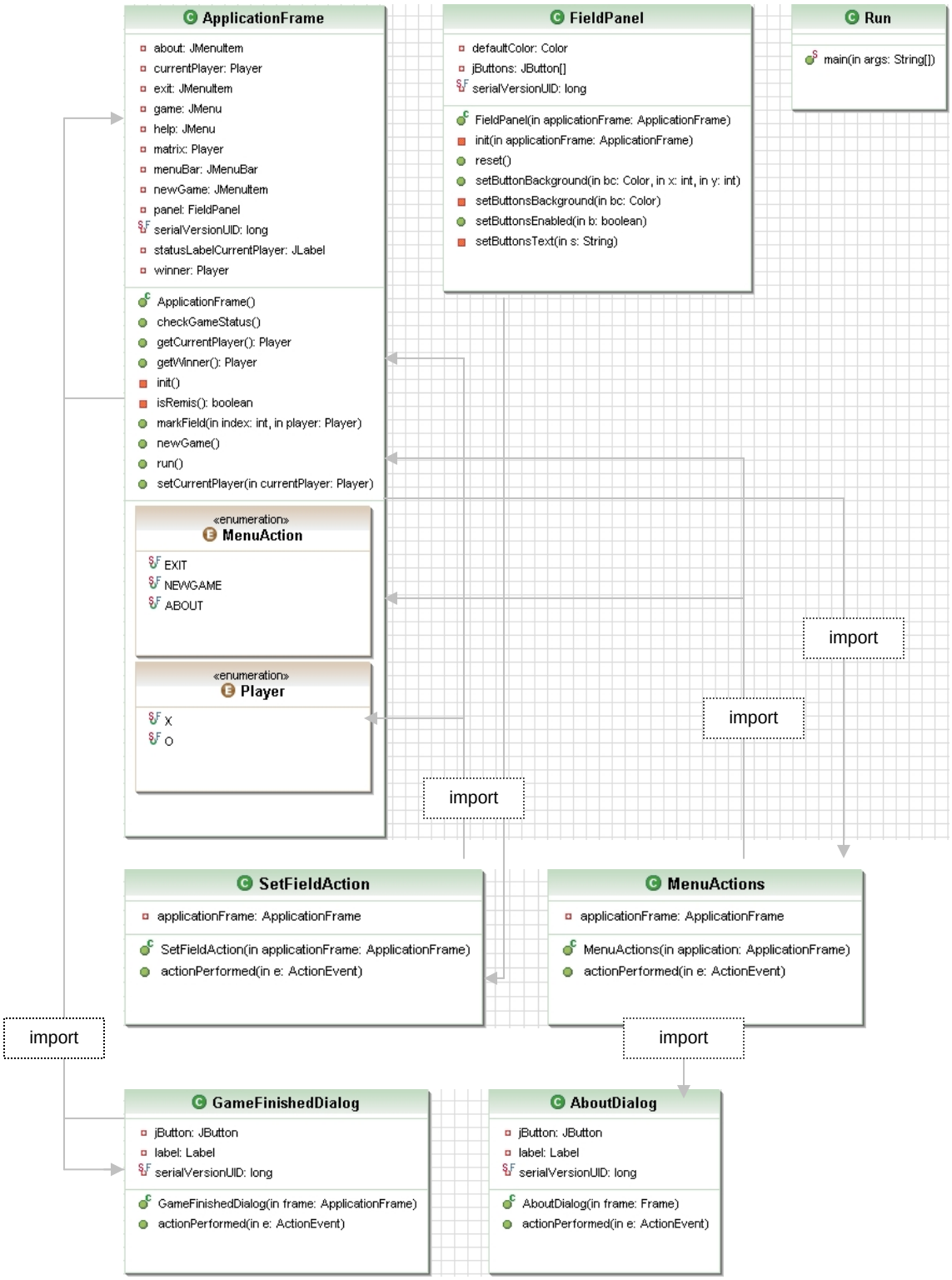
- diese Iteration wird solange wiederholt, bis die Iterationsbedingung erfüllt ist (=entweder hat einer der Spieler gewonnen oder das Feld ist vollständig belegt)
- ist die Bedingung erfüllt sendet das Objekt TicTacToe Nachricht *Finished* an das Objekt Dialog, um das Fenster FinishedDialog zu erzeugen

Sequenzdiagramm:



Zuzana Gajdosova
0050230

Klassendiagramm:



Zuzana Gajdosova
0050230

2.3. Beschreibung des Klassendiagramms:

Package ticTacToe

Klasse ApplicationFrame

Variablen:
public enum Player
→ Aufzählung der Spieler *{ X, O }*
private currentPlayer
private winner
private menuBar
private game
private help
→ Menü Game und Help
private newGame
private exit
→ Menüunterpunkte in Game
private about
→ Menüunterpunkt in Help
private panel
private matrix
→ zweidimensionales Array [3][3]
private statusLabelCurrentPlayer
→ Anzeige des aktuellen Spielers
public enum MenuAction
→ Aufzählung Menuactions { NEWGAME, ABOUT, EXIT }

Methoden:
ApplicationFrame()
→Konstruktor

private void init()
→ erzeugt eine Instanz von FieldPanel

public void run()
→ ruft die Methode init()

public void newGame()
→startet neues Spiel

public void setCurrentPlayer
→ bestimmt den aktuellen Spieler

public Player getWinner
→ gibt den Gewinner zurück

public void markField
→ setzt O/X auf das Spielfeld

public void checkGameStatus()

Zuzana Gajdosova
0050230

→ prüft ob einer der Spieler gewonnen hat. Falls einer Spieler gewonnen hat, werden die drei Felder blau markiert und alle Felder inaktiv. Am Ende des Spiels wird der GameFinished Dialog gezeigt.

private boolean isRemis
→ prüft ob das Spiel mit einem Remis endet

Klase FielPanel

Variablen:
private jButtons
→ die 9 Spielfelder werden in einem eindimensionalem Array gespeichert

private defaultColor

Methoden:

public FieldPanel
→ Konstruktor der die Methode init() aufruft

private void init
→ das Spielfeld 3x3 wird erzeugt

public void reset()
→ diese Methode wird beim Starten des Spiels aufgerufen, der Text wird gelöscht und die Spielfelder werden zum neuen Spiel freigegeben

public void setButtonsEnabled(boolean)
→ Freigabe der Spielfelder

private void setButtonsText(String)
→ es wird 0/X gesetzt

private void setButtonsBackground(Color)
→ die Farbe wir gesetzt

public void setButtonBackground(Color, int, int)
→ die Farbe eines bestimmten Spielfelds wird gesetzt. Hier erfolgt Umrechnung zweidimensionales in eindimensionales Array

Klasse Run

Methode:
public static void main(String[] args)
→ Mainmethode, die eine Instanz von „TicTacToe“ (ApplicationFrame) erzeugt. Es startet mit der Methode run(), die die Methode init() aufruft.

Package ticTacToe.Dialogs

Klasse AboutDialog

Zuzana Gajdosova
0050230

Variablen:
private label
private jButton

Methoden:
public AboutDialog
→ AboutDialog wird erzeugt

public void actionPerformed
→ Ereignishandlung wird definiert; nach dem klicken der Schaltfläche (Button) schließt sich das Dialogfenster zu

Klasse GameFinishedDialog

Variablen:
private label
private jButton

Methoden
public GameFinishedDialog
→ GameFinishedDialog wird erzeugt und je nach dem Ergebnis des Spieles wird “No Winner“ bzw. „Winner: {Spielername}“ ausgegeben

public void actionPerformed
→ Ereignishandlung wird definiert; nach dem klicken der Schaltfläche (Button) schließt sich das Dialogfenster zu

Package ticTacToe.Actions

Klasse SetFieldActions

import ticTacToe.ApplicationFrame
import ticTacToe.ApplicationFrame.Player

Variablen:
private applicationFrame

Methoden:
public SetFieldAction
→ Konstruktor mit der Variablen applicationFrame

public void actionPerformed
→ deklariert die Ereignishandlungen für das Spielfelds; es wird Zeichen des aktuellen Spielers gesetzt, Spielzustand überprüft und falls kein Spieler gewonnen hat und das Spielfeld nicht vollständig besetzt ist, wird der Spieler gewechselt. Nach dem Setzen des Zug wird das Spielfeld inaktiv.

Klasse MenuActions

Zuzana Gajdosova
0050230

import ticTacToe.ApplicationFrame
import ticTacToe.ApplicationFrame.MenuAction
import ticTacToe.Dialogs.AboutDialog

Variablen:
private applicationFrame

Methoden:
public MenuActions
→ Konstruktor mit der Variablen applicationFrame

public void actionPerformed
→ deklariert die Ereignishandlungen für das Menü (NEWGAME, ABOUT, EXIT)

4. Benützung des Programms

Nach dem Starten des Spiels zeigt sich folgendes Fenster:

Um das Spiel anzufangen, muss man auf New Game klicken.

Durch anklicken der Felder wird das Zeichen des aktuellen Spielers gesetzt.
Nach dem Ende des Spieles, z. B. wenn einer der Spieler gewonnen hat, wird folgendes Fenster angezeigt:

5. Installation und Wartung

Als Java Anwendung benötigt dieses Spiel ein JRE 1.5 oder kompatibles. Die exportierte JAR Datei kann man auch außerhalb Eclipse verwenden.
Es ist keine Installation notwendig. Nach dem Abspeichern der JAR Datei kann das Spiel durch Doppelklick auf TicTacToe.jar oder im Dos-Fenster ausgeführt werden.
Für das Programm ist grundsätzlich keine Wartung vorgesehen.