Klasse PlayGround 1/3

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

class PlayGround extends Frame implements ActionListener {
  private static final int     DIMENSION             = 3;
  private static final int     NR_IN_ROW_FOR_SUCCESS = 3;
  private static final boolean SINK_DOWN             = false;

  private final int HEIGHT;
  private final int WIDTH;
  private       int dimension;
  private Figure [][] fig;
  private Evaluator eval;
  private boolean sinkDown;

  public PlayGround ( int dimension, Evaluator eval, boolean sinkDown ) {
    super( "Tic Tac Toe" );
    HEIGHT = dimension*100;
    WIDTH  = dimension*100;
    this.dimension = dimension;
    fig = new Figure[dimension][dimension];
    this.eval = eval;
    this.sinkDown = sinkDown;
    setSize( WIDTH, HEIGHT );
    setLayout( new GridLayout( dimension, dimension ) );
    for ( int i=0; i<dimension; i++ )
      for ( int j=0; j<dimension; j++ ) {
        add( fig[i][j] = new Figure() );
        fig[i][j].addActionListener( this );
      }
```

Klasse PlayGround 2/3

```
    addWindowListener( new WindowAdapter()
                        { public void windowClosing ( WindowEvent e )
{System.exit(0);} }  );
  }

  public void actionPerformed( ActionEvent e ) {
    int row;
    for ( int i=0; i<dimension; i++ )
      for ( int j=0; j<dimension; j++ )
        if ( e.getSource() == fig[i][j] ) {
          if ( sinkDown ) {
            row = 0;
            while ( (row<dimension) && (fig[row][j].symbol == Figure.NONE)
)
              row++;
            row--;
            if ( (row>=0) && (fig[row][j].symbol == Figure.NONE) )
              setFigure( row, j );
          } else if ( fig[i][j].symbol == Figure.NONE ) {
            setFigure( i, j );
          }
        }
  } // actionPerformed
```

Klasse PlayGround 3/3

```
  private void setFigure( int i, int j ) {
    fig[i][j].symbol = Figure.currentSymbol;
    fig[i][j].setText( new Character(Figure.currentSymbol).toString() );
    if ( eval.succeeded( fig, Figure.currentSymbol ) ) {
      for ( int i1=0; i1<dimension; i1++ )
        for ( int j1=0; j1<dimension; j1++ )
          fig[i1][j1].removeActionListener( this );
      new SuccessFrame( Figure.currentSymbol, this );
    } else if ( eval.undecided( fig, Figure.NONE ) ) {
      new UndecidedFrame( this );
    }
    Figure.toggleSymbol();
  }

  public static void main(String[] args) {
    Evaluator  eval = new Evaluator( DIMENSION, NR_IN_ROW_FOR_SUCCESS );
    PlayGround field = new PlayGround( DIMENSION, eval, SINK_DOWN );
    field.show();
  }

}
```

Klasse SuccessFrame

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

class SuccessFrame extends Frame implements ActionListener {
  private final int WIDTH  = 200;
  private final int HEIGHT = 150;
  private JButton stop = new JButton( "OK" );
  private Frame playGround;

  public SuccessFrame( char successNr, Frame playGround ) {
    super( "Gewonnen!" );
    this.playGround = playGround;
    setSize( WIDTH, HEIGHT );
    setLayout( new GridLayout( 2, 1 ) );
    Label lb = new Label( "Spieler '"+successNr+"' hat gewonnen!" );
    lb.setAlignment(Label.CENTER);
    add( lb );
    add( stop );
    stop.addActionListener( this );
    show();
  }

  public void actionPerformed( ActionEvent e ) {
    hide();
    playGround.hide();
    System.exit( 0 );
  }

}
```

Klasse Figure

```java
import javax.swing.JButton;

class Figure extends JButton {
  static final char NONE   = '\0';
  static final char CROSS  = 'X';
  static final char CIRCLE = 'O';
  static       char currentSymbol = CROSS;
  char symbol = NONE;

  Figure() {
    symbol = NONE;
  }

  static void toggleSymbol () {
    if ( currentSymbol == CROSS )
      currentSymbol = CIRCLE;
    else
      currentSymbol = CROSS;
  }

}
```