

Feb 17, 04 12:09

**ATerminalState.java**

Page 1/1

```
// Decompiled by Jad v1.5.8e2. Copyright 2001 Pavel Kouznetsov.
// Jad home page: http://kpdus.tripod.com/jad.html
// Decompiler options: packimports(3) space
// Source File Name:   ATerminalState.java

package model.board;

import model.*;

// Referenced classes of package model.board:
//     IBoardState

abstract class ATerminalState
implements IBoardState
{
    ATerminalState()
    {
    }

    public void map(int i, IBoardLambda iboardlambda, Object obj, IBoardModel iboardmodel)
    {
        iboardlambda.noApply(i, iboardmodel, obj);
    }

    public abstract Object execute(IBoardStatusVisitor iboardstatusvisitor, Object obj, IBoardModel iboardmodel);
}
```

Feb 17, 04 12:09

**DrawState.java**

Page 1/1

```
// Decompiled by Jad v1.5.8e2. Copyright 2001 Pavel Kouznetsov.
// Jad home page: http://kpdus.tripod.com/jad.html
// Decompiler options: packimports(3) space
// Source File Name:   DrawState.java

package model.board;

import model.IBoardModel;
import model.IBoardStatusVisitor;

// Referenced classes of package model.board:
//   ATerminalState

class DrawState extends ATerminalState
{
    private DrawState()
    {
    }

    public Object execute(IBoardStatusVisitor iboardstatusvisitor, Object obj, IBoardModel iboardmodel)
    {
        return iboardstatusvisitor.drawCase(iboardmodel, obj);
    }

    public static DrawState Singleton = new DrawState();
}
```

Feb 17, 04 12:09

**IBoardState.java**

Page 1/1

```
// Decompiled by Jad v1.5.8e2. Copyright 2001 Pavel Kouznetsov.  
// Jad home page: http://kpdus.tripod.com/jad.html  
// Decompiler options: packimports(3) space  
// Source File Name:   IBoardState.java  
  
package model.board;  
  
import model.*;  
  
interface IBoardState  
{  
    public abstract void map(int i, IBoardLambda iboardlambda, Object obj, IBoardM  
odel iboardmodel);  
    public abstract Object execute(IBoardStatusVisitor iboardstatusvisitor, Object  
obj, IBoardModel iboardmodel);  
}
```

Feb 17, 04 9:54

TicTacToeBoard.java

Page 1/2

```
// Decompiled by Jad v1.5.8e2. Copyright 2001 Pavel Kouznetsov.
// Jad home page: http://kpdus.tripod.com/jad.html
// Decompiler options: packimports(3) space
// Source File Name:   TicTacToeBoard.java

package model.board;

import GameIO.IModel;
import model.*;

// Referenced classes of package model.board:
//     ABoardModel, NonTerminalState

public class TicTacToeBoard extends ABoardModel
{
    public TicTacToeBoard(int i, int j, IModel imodel)
    {
        super(i, j, imodel);
        IN_ROW = 3;
    }

    private synchronized int checkWin(int i, int j)
    {
        return winCheck(i, j);
    }

    public synchronized IUndoMove makeMove(final int row, final int col, int i, ICheckMoveVisitor icheckmovevisitor, IBoardStatusVisitor iboardstatusvisitor)
    {
        if (isValidMove(i, row, col))
        {
            super.cells[row][col] = i * 2 - 1;
            chgState(checkWin(row, col));
            icheckmovevisitor.validMoveCase();
            execute(iboardstatusvisitor, null);
            return new IUndoMove() {

                public void apply(IUndoVisitor iundovisitor)
                {
                    undoMove(row, col, iundovisitor);
                }
            };
        } else
        {
            icheckmovevisitor.invalidMoveCase();
            return new IUndoMove() {

                public void apply(IUndoVisitor iundovisitor)
                {
                }
            };
        }
    }

    public synchronized void undoMove(int i, int j, IUndoVisitor iundovisitor)
    {
        int k = super.cells[i][j];
        if (k == 0)
        {
            iundovisitor.noTokenCase();
        } else
        {
            super.cells[i][j] = 0;
            iundovisitor.tokenCase((k + 1) / 2);
        }
        super.state = NonTerminalState.Singleton;
    }
}
```

Feb 17, 04 9:54

TicTacToeBoard.java

Page 2/2

```
    }

    int winCheck(int i, int j)
    {
        int k = super.cells[i][j];
        if (k != 0)
        {
            for (int l = 0; l < super.directions.length; l++)
            {
                int i1 = super.directions[l][0];
                int j1 = super.directions[l][1];
                if (winCheckHelp1(k, IN_ROW - 2, i + i1, j + j1, i1, j1, i - i1, j - j1))
                {
                    return k;
                }
            }
        }
        return 0;
    }

    boolean winCheckHelp1(int i, int j, int k, int l, int i1, int j1, int k1, int l1)
    {
        if (k < 0 || l < 0 || k >= super.cells.length || l >= super.cells[k].length)
            return winCheckHelp2(i, j, k1, l1, -i1, -j1);
        if (super.cells[k][l] != i)
            return winCheckHelp2(i, j, k1, l1, -i1, -j1);
        if (0 == j)
            return true;
        else
            return winCheckHelp1(i, --j, k + i1, l + j1, i1, j1, k1, l1);
    }

    boolean winCheckHelp2(int i, int j, int k, int l, int i1, int j1)
    {
        if (k < 0 || l < 0 || k >= super.cells.length || l >= super.cells[k].length || super.cells[k][l] != i)
            return false;
        if (0 == j)
            return true;
        else
            return winCheckHelp2(i, --j, k + i1, l + j1, i1, j1);
    }

    public boolean isValidMove(int i, int j, int k)
    {
        return 0 == super.cells[j][k];
    }

    private int IN_ROW;
}
```