

Feb 17, 04 10:23

**APlayer.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:   APlayer.java
```

```
package model;
```

```
// Referenced classes of package model:  
//     IRequestor
```

```
public abstract class APlayer
```

```
{
```

```
    public APlayer(IRequestor irequestor, int i)
```

```
    {  
        nextPlayer = this;  
        iRequestor = irequestor;  
        player = i;  
    }
```

```
    public abstract void takeTurn();
```

```
    public IRequestor getRequestor()
```

```
    {  
        return iRequestor;  
    }
```

```
    public int getPlayer()
```

```
    {  
        return player;  
    }
```

```
    public APlayer getNextPlayer()
```

```
    {  
        return nextPlayer;  
    }
```

```
    private void setNextPlayer(APlayer aplayer)
```

```
    {  
        nextPlayer = aplayer;  
    }
```

```
    public void insertAsRest(APlayer aplayer)
```

```
    {  
        aplayer.setNextPlayer(getNextPlayer());  
        setNextPlayer(aplayer);  
    }
```

```
    private IRequestor iRequestor;
```

```
    private int player;
```

```
    private APlayer nextPlayer;  
}
```

Feb 17, 04 10:24

**ComputerPlayer.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:   ComputerPlayer.java

package model;

import GameIO.IModel;
import java.awt.Point;
import java.io.PrintStream;

// Referenced classes of package model:
//     APlayer, INextMoveStrategy, IRequestor, IRejectCommand

public class ComputerPlayer extends APlayer
{
    public ComputerPlayer(IRequestor irequestor, int i, IModel imodel, INextMoveStrategy inextmovestrategy)
    {
        super(irequestor, i);
        model = imodel;
        iNextMoveStrategy = inextmovestrategy;
        System.out.println("ComputerPlayer is using " + inextmovestrategy);
    }

    public void takeTurn()
    {
        System.out.print("Computer player " + getPlayer() + "(" + this + ") takes turn...");
        final Point p = iNextMoveStrategy.getNextMove(model, getPlayer());
        System.out.println("and moves to " + p);
        getRequestor().setTokenAt(p.y, p.x, getPlayer(), new IRejectCommand() {

            public void execute()
            {
                System.out.println("ComputerPlayer: The move at (" + p.x + ", " + p.y + ") is invalid.
");
                takeTurn();
            }
        });
    }

    private INextMoveStrategy iNextMoveStrategy;
    private IModel model;
}
```

Feb 17, 04 10:24

HumanPlayer.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:   HumanPlayer.java

package model;

import java.io.PrintStream;

// Referenced classes of package model:
//     APlayer, ITurnAdmin, IRequestor, IViewRequestor,
//     IRejectCommand

public class HumanPlayer extends APlayer
{
    public HumanPlayer(IRequestor irequestor, int i, ITurnAdmin iturnadmin)
    {
        super(irequestor, i);
        turnAdmin = iturnadmin;
    }

    public void takeTurn()
    {
        System.out.println("Human player " + getPlayer() + " takes turn.");
        turnAdmin.takeTurn(new IViewRequestor() {

            public void setTokenAt(int i, int j, final IRejectCommand rejectCommand)
            {
                getRequestor().setTokenAt(i, j, getPlayer(), new IRejectCommand() {

                    public void execute()
                    {
                        rejectCommand.execute();
                        takeTurn();
                    }

                });
            }

        });
    }

    private ITurnAdmin turnAdmin;
}
```

Feb 17, 04 10:28

**IBoardLambda.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:   IBoardLambda.java  
  
package model;  
  
// Referenced classes of package model:  
//     IBoardModel  
  
public interface IBoardLambda  
{  
    public abstract boolean apply(int i, IBoardModel iboardmodel, Object obj, int  
j, int k, int l);  
    public abstract void noApply(int i, IBoardModel iboardmodel, Object obj);  
}
```

Feb 17, 04 10:25

**IBoardModel.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:   IBoardModel.java

package model;

import java.awt.Dimension;

// Referenced classes of package model:
//      ICheckMoveVisitor, IBoardStatusVisitor, IUndoMove, IBoardLambda,
//      ICommand

public interface IBoardModel
{
    public abstract Dimension getDimension();

    public abstract IUndoMove makeMove(int i, int j, int k, ICheckMoveVisitor iche
ckmovevisitor, IBoardStatusVisitor iboardstatusvisitor);

    public abstract void reset();

    public abstract int[][] getCells();

    public abstract void map(int i, IBoardLambda iboardlambda, Object obj);

    public abstract void mapAll(int i, IBoardLambda iboardlambda, Object obj);

    public abstract int playerAt(int i, int j);

    public abstract Object execute(IBoardStatusVisitor iboardstatusvisitor, Object
obj);

    public abstract boolean isValidMove(int i, int j, int k);

    public abstract void redrawAll(ICommand icommand);

    public abstract boolean isSkipPlayer(int i);
}
```

Feb 17, 04 10:27

**IBoardStatusVisitor.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:   IBoardStatusVisitor.java
```

```
package model;
```

```
// Referenced classes of package model:  
//     IBoardModel
```

```
public interface IBoardStatusVisitor
```

```
{
```

```
    public abstract Object player0WonCase(IBoardModel iboardmodel, Object obj);
```

```
    public abstract Object player1WonCase(IBoardModel iboardmodel, Object obj);
```

```
    public abstract Object drawCase(IBoardModel iboardmodel, Object obj);
```

```
    public abstract Object noWinnerCase(IBoardModel iboardmodel, Object obj);  
}
```

Feb 17, 04 10:25

**ICheckMoveVisitor.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:   ICheckMoveVisitor.java
```

```
package model;
```

```
public interface ICheckMoveVisitor
```

```
{
```

```
    public abstract void invalidMoveCase();
```

```
    public abstract void validMoveCase();
```

```
}
```

Feb 17, 04 10:25

 **ICommand.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:   ICommand.java
```

```
package model;
```

```
public interface ICommand
```

```
{
```

```
    public abstract void setTokenAt(int i, int j, int k);
```

```
    public abstract void clearTokenAt(int i, int j);
```

```
    public abstract void setMessage(String s);
```

```
}
```

Feb 17, 04 10:25

**INextMoveStrategy.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:   INextMoveStrategy.java
```

```
package model;
```

```
import GameIO.IModel;
```

```
import java.awt.Point;
```

```
public interface INextMoveStrategy
```

```
{
```

```
    public abstract Point getNextMove(IModel imodel, int i);
```

```
}
```

Feb 17, 04 10:28

**IRejectCommand.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:   IRejectCommand.java
```

```
package model;
```

```
public interface IRejectCommand
```

```
{
```

```
    public abstract void execute();
```

```
}
```

Feb 17, 04 10:24

**ITurnAdmin.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:   ITurnAdmin.java
```

```
package model;
```

```
// Referenced classes of package model:  
//     IViewRequestor
```

```
public interface ITurnAdmin
```

```
{
```

```
    public abstract void takeTurn(IViewRequestor iviewrequestor);
```

```
}
```

Feb 17, 04 10:28

**IUndoMove.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:    IUndoMove.java
```

```
package model;
```

```
// Referenced classes of package model:  
//     IUndoVisitor
```

```
public interface IUndoMove
```

```
{
```

```
    public abstract void apply(IUndoVisitor iundovisitor);
```

```
}
```

Feb 17, 04 10:28

**IUndoVisitor.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: IUndoVisitor.java
```

```
package model;
```

```
public interface IUndoVisitor
```

```
{
```

```
    public abstract void noTokenCase();
```

```
    public abstract void tokenCase(int i);
```

```
}
```

Feb 17, 04 10:30

**IViewAdmin.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:    IViewAdmin.java
```

```
package model;
```

```
public interface IViewAdmin
```

```
{
```

```
    public abstract void draw();
```

```
    public abstract void win(int i);
```

```
    public abstract void reset();
```

```
}
```

Feb 17, 04 10:24

**IViewRequestor.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:   IViewRequestor.java
```

```
package model;
```

```
// Referenced classes of package model:  
//     IRejectCommand
```

```
public interface IViewRequestor
```

```
{
```

```
    public abstract void setTokenAt(int i, int j, IRejectCommand irejectcommand);
```

```
}
```

Feb 17, 04 10:27

## TurnControl.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:   TurnControl.java
```

```
package model;

import java.io.PrintStream;

// Referenced classes of package model:
//     APlayer

public class TurnControl
{
    private abstract class ANextTurn
    {
        abstract boolean apply();

        private ANextTurn()
        {
        }
    }

    public TurnControl(APlayer aplayer)
    {
        skipPlayer = false;
        proceedTurn = new ANextTurn() {

            boolean apply()
            {
                System.out.println("Proceeding to next player...");
                players = players.getNextPlayer();
                if (skipPlayer)
                {
                    skipPlayer = false;
                    players = players.getNextPlayer();
                }
                setWait();
                players.takeTurn();
                return true;
            }
        };
        waitTurn = new ANextTurn() {

            boolean apply()
            {
                System.out.println("Waiting....");
                try
                {
                    Thread.sleep(500L);
                }
                catch (Exception exception)
                {
                    System.err.println(exception);
                }
                return true;
            }
        };
        haltTurn = new ANextTurn() {

            boolean apply()
            {
                System.out.println("Halting....");
                return false;
            }
        };
    }
}
```

Feb 17, 04 10:27

## TurnControl.java

Page 2/2

```
    }

    };
    nextTurn = proceedTurn;
    players = aplayer;
}

public void addPlayer(APlayer aplayer)
{
    players.insertAsRest(aplayer);
    APlayer aplayer1 = players;
    int i = 1;
    do
    {
        System.out.print("player " + i + "=" + aplayer1 + " ");
        while (players != (aplayer1 = aplayer1.getNextPlayer()));
        System.out.println("");
    }

    public void run()
    {
        nextTurn = proceedTurn;
        (new Thread() {

            public void run()
            {
                try
                {
                    while (nextTurn.apply()) ;
                }
                catch (Exception exception)
                {
                    System.out.println("TurnControl.run() exception: " + exception);
                }
            }
        }).start();

    public synchronized void setProceed()
    {
        nextTurn = proceedTurn;
    }

    public synchronized void setWait()
    {
        nextTurn = waitTurn;
    }

    public synchronized void setHalt()
    {
        nextTurn = haltTurn;
    }

    public synchronized void setSkipPlayer(boolean flag)
    {
        System.out.println("Skip player #" + flag);
        skipPlayer = flag;
    }

    APlayer players;
    boolean skipPlayer;
    private ANextTurn proceedTurn;
    private ANextTurn waitTurn;
    private ANextTurn haltTurn;
    private volatile ANextTurn nextTurn;
}
```