

Permainan TicTacToe Menggunakan Java Applet

Francisco Silvano

franslpster@gmail.com

http://franzeast.com

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

DASAR TEORI

Java Applet adalah sebuah class yang dapat diaktifkan pada Web Browser seperti pada Netscape Communicator atau Internet Explorer. Java Applet di enkapsulasi sebagai bagian dari HTML (HyperText Markup Language).

Perbedaan Java Applet dan Java Application

1. Application berjalan menggunakan Java Interpreter atau command line prompt. Sedang Applet berjalan pada Browser atau pada halaman WEB menggunakan HTML.
2. Java Applet adalah sebuah class yang dapat diaktifkan pada Web Browser seperti Netscape Communicator atau Internet explorer. Java Applet dienkapsulasi sebagai bagian dari HTML.
3. Java Applet mempunyai siklus hidup sedang Java Application tidak mempunyai siklus hidup.

<HTML> dan </HTML> adalah pasangan kunci kata untuk memulai kode HTML.

<APPLET> dan </APPLET> adalah pasangan kunci kata yang digunakan untuk mengaktifkan Applet dan Parameter untuk applet tersebut.

WIDTH dan HEIGHT menentukan jumlah pixel yang dibutuhkan untuk menggambar atau menampilkan applet.

File html ini disimpan dengan nama file yang berekstensi .html dan kemudian dapat dipanggil melalui URL dari web browser atau menggunakan aplikasi **appletviewer**.

Script Program HTML :

```
<html>
<head>
<title>Game Tic Tac Toe</title>
</head>
<body>
<h4>Oleh :</h4>
<h4> Francisco.S.do.R.Ximenes</h4>
<h1>Game Tic Tac Toe</h1>
<hr>
<applet code=TicTacToe.class width=120 height=120>
</applet>
<hr>
</body>
</html>
```

Script Program JAVA :

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.image.*;
import java.net.*;
import java.applet.*;

public class TicTacToe
extends Applet implements MouseListener {

int white;
int black;
final static int moves[] = {
4, 0, 2, 6, 8, 1, 3, 5, 7};
static boolean won[] = new boolean[1 << 9];
static final int DONE = (1<<9) - 1;
static final int OK = 0;
static final int WIN = 1;
static final int LOSE = 2;
static final int STALEMATE = 3;

boolean first = true;

Image notImage;
Image crossImage;
public void init() {
notImage = getImage(getCodeBase(), "images/not.gif");
crossImage = getImage(getCodeBase(), "images/cross.gif");
addMouseListener(this);
}
```

```
public void destroy() {
removeMouseListener(this);
}

/**
 * Paint it.
 */
public void paint(Graphics g) {
Dimension d = getSize();
g.setColor(Color.black);
int xoff = d.width / 3;
int yoff = d.height / 3;
g.drawLine(xoff, 0, xoff, d.height);
g.drawLine(2 * xoff, 0, 2 * xoff, d.height);
g.drawLine(0, yoff, d.width, yoff);
g.drawLine(0, 2 * yoff, d.width, 2 * yoff);

int i = 0;
for (int r = 0; r < 3; r++) {
for (int c = 0; c < 3; c++, i++) {
if ( (white & (1 << i)) != 0) {
g.drawImage(notImage, c * xoff + 1, r * yoff + 1, this);
}
else if ( (black & (1 << i)) != 0) {
g.drawImage(crossImage, c * xoff + 1, r * yoff + 1, this);
}
}
}
}

public void mouseReleased(MouseEvent e) {
int x = e.getX();
int y = e.getY();

switch (status()) {
case WIN:
case LOSE:
case STALEMATE:
play(getCodeBase(), "audio/return.au");
white = black = 0;
if (first) {
white |= 1 << (int) (Math.random() * 9);
}
first = !first;
repaint();
return;
}

// Figure out the row/column
```

```
Dimension d = getSize();
int c = (x * 3) / d.width;
int r = (y * 3) / d.height;
if (yourMove(c + r * 3)) {
    repaint();

    switch (status()) {
    case WIN:
        play(getCodeBase(), "audio/yahoo1.au");
        break;
    case LOSE:
        play(getCodeBase(), "audio/yahoo2.au");

        break;
    case STALEMATE:
        break;
    default:
        if (myMove()) {
            repaint();
            switch (status()) {
            case WIN:
                play(getCodeBase(), "audio/yahoo1.au");
                break;
            case LOSE:
                play(getCodeBase(), "audio/yahoo2.au");
                break;
            case STALEMATE:
                break;
            default:
                play(getCodeBase(), "audio/ding.au");
            }
        }
        else {
            play(getCodeBase(), "audio/beep.au");
        }
    }
    else {
        play(getCodeBase(), "audio/beep.au");
    }
}

public void mousePressed(MouseEvent e) {
}

public void mouseClicked(MouseEvent e) {
}

public void mouseEntered(MouseEvent e) {
```

```
}  
  
public void mouseExited(MouseEvent e) {  
}  
  
boolean myMove() {  
    if ( (black | white) == DONE) {  
        return false;  
    }  
    int best = bestMove(white, black);  
    white |= 1 << best;  
    return true;  
}  
int bestMove(int white, int black) {  
    int bestmove = -1;  
  
    loop:  
    for (int i = 0; i < 9; i++) {  
        int mw = moves[i];  
        if ( ( (white & (1 << mw)) == 0) && ( (black & (1 << mw)) == 0) ) {  
            int pw = white | (1 << mw);  
            if (won[pw]) {  
  
            }  
            for (int mb = 0; mb < 9; mb++) {  
                if ( ( (pw & (1 << mb)) == 0) && ( (black & (1 << mb)) == 0) ) {  
                    int pb = black | (1 << mb);  
                    if (won[pb]) {  
  
                    }  
                    continue loop;  
                }  
            }  
            if (bestmove == -1) {  
                bestmove = mw;  
            }  
        }  
        if (bestmove != -1) {  
            return bestmove;  
        }  
  
        for (int i = 0; i < 9; i++) {  
            int mw = moves[i];  
            if ( ( (white & (1 << mw)) == 0) && ( (black & (1 << mw)) == 0) ) {  
                return mw;  
            }  
        }  
    }  
}
```

```
return -1;
}
/**
 * User move.
 * @return true if legal
 */
boolean yourMove(int m) {
if ( (m < 0) || (m > 8)) {
return false;
}
if ( ( (black | white) & (1 << m)) != 0) {
return false;
}
black |= 1 << m;
return true;
}
static void isWon(int pos) {
for (int i = 0; i < DONE; i++) {
if ( (i & pos) == pos) {
won[i] = true;
}
}
}

static {
isWon( (1 << 0) | (1 << 1) | (1 << 2));
isWon( (1 << 3) | (1 << 4) | (1 << 5));
isWon( (1 << 6) | (1 << 7) | (1 << 8));
isWon( (1 << 0) | (1 << 3) | (1 << 6));
isWon( (1 << 1) | (1 << 4) | (1 << 7));
isWon( (1 << 2) | (1 << 5) | (1 << 8));
isWon( (1 << 0) | (1 << 4) | (1 << 8));
isWon( (1 << 2) | (1 << 4) | (1 << 6));
}

int status() {
if (won[white]) {
return WIN;
}
if (won[black]) {
return LOSE;
}
if ( (black | white) == DONE) {
return STALEMATE;
}
return OK;
}
}
```

Referensi

- Applet(1)
- Bab 1 Pengenalan Kecerdasan Buatan
- Bab 2 Representasi Pengetahuan
- Bab 3 Reasoning, Semantic Network, Frame
- Bab 4 Algoritma Pencarian
- Bab 5 Natural Language Processing
- Bab 7 Algoritma Genetika
- Bab 8 Jaringan Syaraf Tiruan

Biografi Penulis



Francisco Silvano, Saat ini tercatat Aktif sebagai mahasiswa Jenjang Sarjana Program Studi Teknik Informatika, Jurusan Teknik Informatika Fakultas Teknologi Industri, Di Salah Satu PTS Yogyakarta. Memiliki ketertarikan di bidang image processing, e-learning, smart client development, information system dan Networking. Prestasi Saat ini Belum memiliki prestasi apa – apa. Menulis, Sharing pengetahuan IPTEK adalah hobbi saya saat ini Untuk saling sharing saya memiliki web <http://franzeast.com> dan silahkan bagi teman teman yang ingin sharing .