**JAVA 프로그래밍 #11**

담당 교수님 : 천인국 교수님

이름 : 현 지 원

학과 : 컴퓨터공학과

학번 : 20174069

* TIC-TAC-TOE 게임을 작성하여 보자. TIC-TAC-TOE 게임은 3 X 3 칸을 가진 게임판을 만들고, 경기자가 동그라미 기호(O)와 가위표 기호(X)를 번갈아 가며 게임판에 놓는 게임이다. 가로, 세로, 대각선 상관 없이 직선으로 동일한 기호 3개를 먼저 만들면 승리한다. 모든 버튼에 동일한 액션 리스너 객체를 등록한다. 액션 리스너에서는 버튼 클릭 이벤트를 처리하고 아직 클릭 안 된 것을 확인한 후에 동그라미 차례이면 ‘O’으로, 가위 차례이면 ‘X’로 버튼의 텍스트를 변경한다.

**Source code – 초기 버전**

package ddfdff;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class TicTacToe extends JFrame implements ActionListener {

private JPanel panel, panell;

private JButton[][] button = new JButton[3][3];

private JLabel label;

int count = 0;

public TicTacToe() // 생성자

{

setTitle("Tic Tac Toe");

setSize(300, 350);

panel = new JPanel(); // 버튼

panel.setLayout(new GridLayout(3, 3, 2, 2));

// 버튼 생성

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

button[i][j] = new JButton(" ");

button[i][j].addActionListener(this);

panel.add(button[i][j]);

}

}

panell = new JPanel(); // 게임 결과 출력

add(panel, BorderLayout.CENTER);

add(panell, BorderLayout.SOUTH);

setVisible(true);

}

// ActionListener 인터페이스 구현

@Override

public void actionPerformed(ActionEvent e) {

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

if (e.getSource() == button[i][j] && button[i][j].getText().equals(" ")) {

button[i][j].setText("O"); // 나

count++;

}

}

}

while (true) {

int rand1 = (int) (Math.random() \* 3), rand2 = (int) (Math.random() \* 3);

if (count == 9)

break;

if (button[rand1][rand2].getText().equals(" ")) {

button[rand1][rand2].setText("X");

count++;

break;

}

}

// 가로

for (int i = 0; i < 3; i++) {

if ((button[i][0].getText() == "X" || button[i][0].getText() == "O")

&& button[i][0].getText().equals(button[i][1].getText())

&& button[i][1].getText().equals(button[i][2].getText())) {

for (int i1 = 0; i1 < 3; i1++) {

for (int j = 0; j < 3; j++) {

button[i1][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

}

// 세로

for (int i = 0; i < 3; i++) {

if ((button[0][i].getText() == "X" || button[1][i].getText() == "O")

&& button[0][i].getText().equals(button[1][i].getText())

&& button[1][i].getText().equals(button[2][i].getText())) {

for (int i1 = 0; i1 < 3; i1++) {

for (int j = 0; j < 3; j++) {

button[i1][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

}

// 대각선

if ((button[0][0].getText() == "X" || button[0][0].getText() == "O")

&& button[0][0].getText().equals(button[1][1].getText())

&& button[1][1].getText().equals(button[2][2].getText())

&& button[0][0].getText().equals(button[2][2].getText())) {

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

button[i][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

if ((button[0][2].getText() == "X" || button[0][2].getText() == "O")

&& button[0][2].getText().equals(button[1][1].getText())

&& button[1][1].getText().equals(button[2][0].getText())

&& button[0][2].getText().equals(button[2][0].getText())) {

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

button[i][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

}

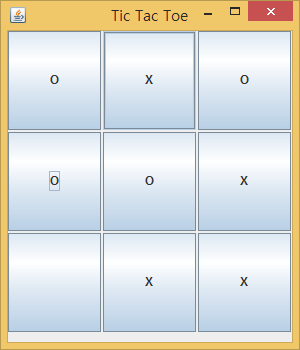
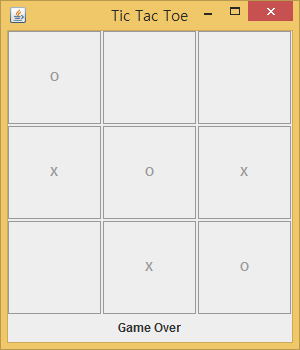
public static void main(String argv[]) {

TicTacToe t = new TicTacToe();

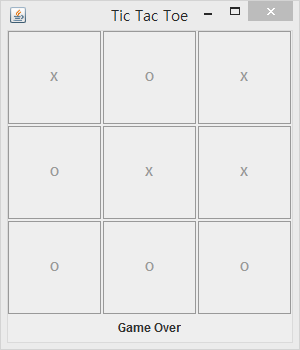
}

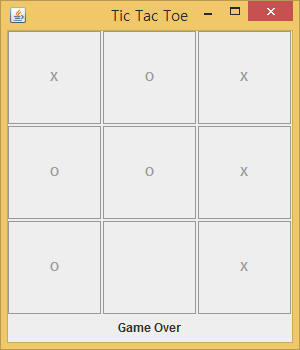
}

**실행결과**



- O가 승리 -게임 중





- x가 승리 - O가 승리

**해설**

모든 버튼에 동일한 액션 리스너 객체를 등록한다. 액션 리스너에서는 버튼 클릭 이벤트를 처리하고 아직 클릭 안 된 것을 확인한 후에 동그라미 차례이면 ‘O’으로, 가위 차례이면 ‘X’로 버튼의 텍스트를 변경한다. 배치 관리자는 GridLayout으로 설정하였다.

가로, 세로, 대각선의 경우로 나눠 x나 o가 승리하는 경우 button의 Enabled를 false로 설정하여 버튼을 누를 수 없게하고, 밑에 Game Over가 나오게 해주었다.

**Source code – 무승부 알고리즘 사용**

package lab12;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class TicTacToe extends JFrame implements ActionListener {

private JPanel panel, panell;

private JButton[][] button = new JButton[3][3];

private JLabel label;

int count = 0;

public TicTacToe() // 생성자

{

setTitle("Tic Tac Toe");

setSize(300, 350);

panel = new JPanel(); // 버튼

panel.setLayout(new GridLayout(3, 3, 2, 2));

// 버튼 생성

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

button[i][j] = new JButton(" ");

button[i][j].addActionListener(this);

panel.add(button[i][j]);

}

}

panell = new JPanel(); // 게임 결과 출력

add(panel, BorderLayout.*CENTER*);

add(panell, BorderLayout.*SOUTH*);

setVisible(true);

}

// 검사

int check() {

if (button[1][1].getText() == " ") { // 가운데가 비어있는 경우

return 0;

}

// defense

// 세로

else if (button[0][0].getText() == "O" && button[1][0].getText() == "O") { // 3개중 2개가 같은 경우

return 1;

} else if (button[0][0].getText() == "O" && button[2][0].getText() == "O") { // 3개중 2개가 같은 경우

return 2;

} else if (button[1][0].getText() == "O" && button[2][0].getText() == "O") { // 3개중 2개가 같은 경우

return 3;

} else if (button[0][1].getText() == "O" && button[1][1].getText() == "O") { // 3개중 2개가 같은 경우

return 4;

} else if (button[0][1].getText() == "O" && button[2][1].getText() == "O") { // 3개중 2개가 같은 경우

return 5;

} else if (button[1][1].getText() == "O" && button[2][1].getText() == "O") { // 3개중 2개가 같은 경우

return 6;

} else if (button[0][2].getText() == "O" && button[1][2].getText() == "O") { // 3개중 2개가 같은 경우

return 7;

} else if (button[0][2].getText() == "O" && button[2][2].getText() == "O") { // 3개중 2개가 같은 경우

return 8;

} else if (button[1][2].getText() == "O" && button[2][2].getText() == "O") { // 3개중 2개가 같은 경우

return 9;

}

// 가로

else if (button[0][0].getText() == "O" && button[0][1].getText() == "O") { // 3개중 2개가 같은 경우

return 10;

} else if (button[0][0].getText() == "O" && button[0][2].getText() == "O") { // 3개중 2개가 같은 경우

return 11;

} else if (button[0][1].getText() == "O" && button[0][2].getText() == "O") { // 3개중 2개가 같은 경우

return 12;

} else if (button[1][0].getText() == "O" && button[1][1].getText() == "O") { // 3개중 2개가 같은 경우

return 13;

} else if (button[1][0].getText() == "O" && button[1][2].getText() == "O") { // 3개중 2개가 같은 경우

return 14;

} else if (button[1][1].getText() == "O" && button[1][2].getText() == "O") { // 3개중 2개가 같은 경우

return 15;

} else if (button[2][0].getText() == "O" && button[2][1].getText() == "O") { // 3개중 2개가 같은 경우

return 16;

} else if (button[2][0].getText() == "O" && button[2][2].getText() == "O") { // 3개중 2개가 같은 경우

return 17;

} else if (button[2][1].getText() == "O" && button[2][2].getText() == "O") { // 3개중 2개가 같은 경우

return 18;

}

// 대각선

else if (button[0][0].getText() == "O" && button[1][1].getText() == "O") { // 3개중 2개가 같은 경우

return 19;

} else if (button[0][0].getText() == "O" && button[2][2].getText() == "O") { // 3개중 2개가 같은 경우

return 20;

} else if (button[1][1].getText() == "O" && button[2][2].getText() == "O") { // 3개중 2개가 같은 경우

return 21;

} else if (button[2][2].getText() == "O" && button[2][0].getText() == "O") { // 3개중 2개가 같은 경우

return 22;

} else

return 23;

}

// ActionListener 인터페이스 구현

@Override

public void actionPerformed(ActionEvent e) {

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

if (e.getSource() == button[i][j] && button[i][j].getText().equals(" ")) {

button[i][j].setText("O"); // 나

count++;

}

}

}

while (true) {

if (count == 9)

break;

if (check() == 0) {

button[1][1].setText("X");

} else if (check() == 1) {

button[2][0].setText("X");

} else if (check() == 2) {

button[1][0].setText("X");

} else if (check() == 3) {

button[0][0].setText("X");

} else if (check() == 4) {

button[2][1].setText("X");

} else if (check() == 5) {

button[1][1].setText("X");

} else if (check() == 6) {

button[0][1].setText("X");

} else if (check() == 7) {

button[2][2].setText("X");

} else if (check() == 8) {

button[1][2].setText("X");

} else if (check() == 9) {

button[0][2].setText("X");

}

else if (check() == 10) {

button[0][2].setText("X");

} else if (check() == 11) {

button[0][1].setText("X");

} else if (check() == 12) {

button[0][0].setText("X");

} else if (check() == 13) {

button[1][2].setText("X");

} else if (check() == 14) {

button[1][1].setText("X");

} else if (check() == 15) {

button[1][0].setText("X");

} else if (check() == 16) {

button[2][2].setText("X");

} else if (check() == 17) {

button[2][1].setText("X");

} else if (check() == 18) {

button[2][0].setText("X");

} else if (check() == 19) {

button[2][2].setText("X");

} else if (check() == 20) {

button[1][1].setText("X");

} else if (check() == 21) {

button[0][0].setText("X");

} else if (check() == 22) {

button[1][1].setText("X");

}

// attack

else {

if (button[0][0].getText() == " ") {

button[0][0].setText("X");

} else if (button[0][2].getText() == " ") {

button[0][2].setText("X");

} else if (button[2][0].getText() == " ") {

button[2][0].setText("X");

} else if (button[2][2].getText() == " ") {

button[2][2].setText("X");

} else if (button[0][1].getText() == " ") {

button[0][1].setText("X");

} else if (button[2][0].getText() == " ") {

button[2][0].setText("X");

} else if (button[2][1].getText() == " ") {

button[2][1].setText("X");

}

}

count++;

break;

}

// 가로

for (int i = 0; i < 3; i++) {

if ((button[i][0].getText() == "X" || button[i][0].getText() == "O")

&& button[i][0].getText().equals(button[i][1].getText())

&& button[i][1].getText().equals(button[i][2].getText())) {

for (int i1 = 0; i1 < 3; i1++) {

for (int j = 0; j < 3; j++) {

button[i1][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

}

// 세로

for (int i = 0; i < 3; i++) {

if ((button[0][i].getText() == "X" || button[1][i].getText() == "O")

&& button[0][i].getText().equals(button[1][i].getText())

&& button[1][i].getText().equals(button[2][i].getText())) {

for (int i1 = 0; i1 < 3; i1++) {

for (int j = 0; j < 3; j++) {

button[i1][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

}

// 대각선

if ((button[0][0].getText() == "X" || button[0][0].getText() == "O")

&& button[0][0].getText().equals(button[1][1].getText())

&& button[1][1].getText().equals(button[2][2].getText())

&& button[0][0].getText().equals(button[2][2].getText())) {

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

button[i][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

if ((button[0][2].getText() == "X" || button[0][2].getText() == "O")

&& button[0][2].getText().equals(button[1][1].getText())

&& button[1][1].getText().equals(button[2][0].getText())

&& button[0][2].getText().equals(button[2][0].getText())) {

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

button[i][j].setEnabled(false);

}

}

JLabel label = new JLabel("Game Over");

panell.add(label);

}

}

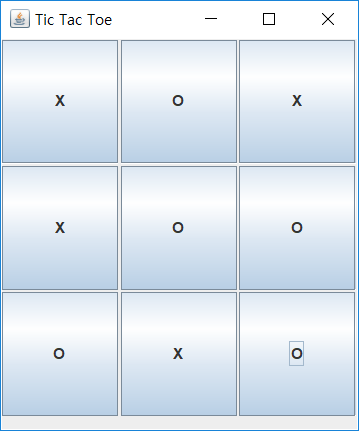
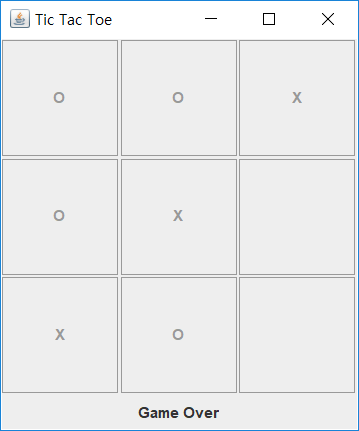
public static void main(String argv[]) {

TicTacToe t = new TicTacToe();

}

}

**실행결과**



- x가 승리 - 무승부

**해설**

컴퓨터가 X를 놓는 위치에 대한 알고리즘은 먼저 가운데가 비어있으면 그 자리에 X를 놓는다. 그리고 만약 가로 세로 대각선 중에 한 줄에서라도 2개가 같은 기호일 경우 빈 곳에 놓는다. 그 후에 대각선 순서대로 빈 곳에 놓고 남은 칸을 채우면 된다. 아직 학습이 부족하여 인공지능을 구현하지는 못하고, 모든 경우를 일일이 적어 X를 표시해 주었다. 추후에 더 공부를 하여 인공지능으로 이 프로그램을 다시 만들어 볼 것이다.