Computer Science 201 – Final Assignment

Author – Ali Vira

Date – 10/29-2011

**Project Overview**

The aim of the project was to program an interface that would allow a user to play a game of Tic-Tac-Toe against a computer. The computer would be able to recognize and effectively respond to player moves, preventing the player from winning. Thus, the computer is said to be unbeatable. Using a GUI, computer and player moves can be effectively and aesthetically presented to the user. A loss and tie display, and a reset button are also included so that the user can keep track of their progress.

**Programming Components**

*1. Arrays*

Arrays are used to track player and computer moves. For instance, if a player activates the top left square, the ‘playermove’ array changes the value in its first slot to one. If the computer responds by moving to the middle square, the ‘computermove’ array changes its value in the fifth slot to one.

*2. Variables*

Numerous variables are used throughout the program in order to keep track of when and where the player moves, as well as when the computer is allowed to move. The computer move portion of the program relies upon a series of nested loops. Each loop represents a different type of computer response to the player’s move, the structure of which will be elaborated upon further below. If the computer makes a move based on the information in the first loop, the variable representing the computer’s turn will be set to false, and the rest of the loops will not trigger. Alternatively, if the first loop does not trigger, the variable remains true and the computer will check the next loops for available moves.

*3. Loops*

The whole program is based on a series of if/else loops. The computer is constantly checking where the player has moved, and makes a response accordingly. For instance, if the player moves in a corner, the computer moves to the center. If the computer sees the player is about to win, it moves to intercept. If the computer sees that it can win, it moves to do so. The GUI itself is also based on a series of if else statements. If a value in the player/computermove array is set to 1, the picture in the GUI changes to the corresponding X or O image.

*4. JPanel and JFrame*

The Tic-Tac-Toe interface is made using javax.swing. JPanels and JFrames are utilized to create the representation of the board. Buttons and icons are then added to the frame so as to allow user input. A final component of the JPanel is a portion of text which changes based on whether the computer wins or ties. While the game is in progress the text simply reads: Tic Tac Toe Rev.7

*5. Buttons, Icons and ActionListeners*

The Tic-Tac-Toe grid is represented by a 3x3 set of buttons with embedded icons displaying a blank tile, an X, or an O. If a player clicks a button, the action listener triggers and the button’s icon changes to a black X, and the button is disabled to prevent cheating. When the computer responds, the button’s icon changes to a green O, and the button is also disabled to prevent cheating. When the computer wins, all the O icons change their color from green to red, allowing the user to see clearly where they have lost. The final button is the clear button, which, when clicked, resets the player/computermove arrays, the computer turn variables, and the icons.

*6. Methods, Classes, and Public Variables*

The code for each computer response is set in a series of public classes and methods. Each set of code uses a series of public variables defined at the beginning of the code to track its progress. This is useful because the variables are constant between the various classes and can be modified and referenced consistently.

**How it works**

The code is based on the fact that there are only 24 possible winning combinations of Tic-Tac-Toe as illustrated below:



Figure : Possible win combinations

The computer works as follows:

1. Is the player about to win?
   1. Yes: Move to stop the player and end turn
   2. No: Check next loop
2. Is the computer about to win
   1. Yes: Move to win and end turn
   2. No: Check next loop
3. Did the player move to a corner?
   1. Yes: Is the center already taken?
      1. Yes: Perform failsafe and end turn
      2. No: Move to the center and end turn
   2. No: Check next loop
4. Did the player move to the center?
   1. Yes: Is the left corner already taken?
      1. Yes: Move to next corner available corner and end turn
         1. No available corner?
            1. Yes: Perform the failsafe and end turn
            2. No: Move to next loop
      2. No: Move to the left corner and end turn
   2. No: Check next loop
5. Did the player select an edge?
   1. Yes: Is the top left corner already taken?
      1. Yes: Move to next corner available corner and end turn
         1. No available corner?
            1. Yes: Perform the failsafe and end turn
            2. No: Move to next loop
      2. No: Move to the left corner and end turn
   2. No: Check next loop
6. Computer failsafe
   1. Check to see if to left space is free
      1. Yes: Move there and end turn
      2. No: Repeat for next square

If the computer moves at the end of any of these loops, it sets the computer turn variable to zero and does not move until the player moves.

**Code Testing**

In order to ensure that the program truly was unbeatable, various contests were held with other members of the class to see who could beat it. These challenges exposed several flaws with the program, the biggest one being that sometimes the computer would fail to move only once, and would change the playermove array to make sure that it won. Through continued testing, this, and other failures in the basic algorithm were brought to light and eventually fixed for the final iteration of the program. The two biggest challenges presented during coding were, a) how to get the computer to respond properly and only once, and b) how to get the GUI together in the first place. Through continued testing and research however, these issues were rectified.

**Sample Outputs:**

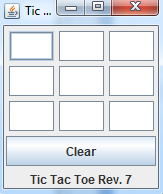
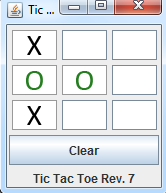
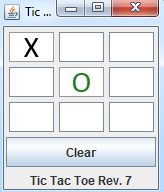


Figure : Blank interface Figure : First move Figure : Second move

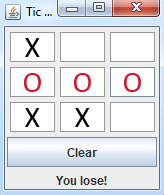
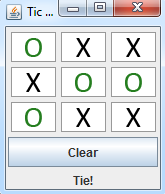


Figure : Loss Figure : Tie

**Reflection and improvements**

1. A display panel that shows wins and losses
2. Different colored backgrounds
3. Gifs for the icons (animations)
4. Sound for the game
5. 2d arrays

**Code Analysis**

*1. Makes the Tic Tac Toe game:*

**public** **static** **void** main(String[] args) {

TicTacToe c1 = **new** TicTacToe();

}

*2. Initialize variables:*

**public** **static** **final** **int** *FRAME\_WIDTH* = 170;

**public** **static** **final** **int** *FRAME\_HEIGHT* = 200;

**public** **static** **int** *playermove*[] = **new** **int**[10];

**public** **static** **int** *compmove*[]=**new** **int** [10];

**public** **static** **int** *compturn*=0;

**public** **static** **int** *compturn1*=0;

**public** **static** **int** *compturn2*=0;

**public** **static** **int** *compturn3*=0;

**public** **static** **int** *compturn4*=0;

**public** **static** **int** *compvictory*=0;

**public** **static** ImageIcon *blank* = **new** ImageIcon("images\\blank.png");

**public** **static** ImageIcon *ex* = **new** ImageIcon("images\\ex.png");

**public** **static** ImageIcon *oh* = **new** ImageIcon("images\\oh.png");

**public** **static** ImageIcon *ohwin* = **new** ImageIcon("images\\ohwin.png");

**public** **static** JButton *clear* = **new** JButton("Clear");

**public** **static** JButton *button00* = **new** JButton(*blank*);

**public** **static** JButton *button01* = **new** JButton(*blank*);

**public** **static** JButton *button02* = **new** JButton(*blank*);

**public** **static** JButton *button10* = **new** JButton(*blank*);

**public** **static** JButton *button11* = **new** JButton(*blank*);

**public** **static** JButton *button12* = **new** JButton(*blank*);

**public** **static** JButton *button20* = **new** JButton(*blank*);

**public** **static** JButton *button21* = **new** JButton(*blank*);

**public** **static** JButton *button22* = **new** JButton(*blank*);

**public** **static** JLabel *losswin* = **new** JLabel("Tic Tac Toe");

*3. Sample – set button size:*

JFrame frame = **new** JFrame("Tic Tac Toe Rev. 7 - Ali Vira");

JPanel panel = **new** JPanel();

*clear*.setPreferredSize(**new** Dimension (150,30));

*button00*.setPreferredSize(**new** Dimension(45, 30));

*4. Sample – add buttons to frame:*

panel.add(*button22*);

panel.add(*clear*);

panel.add(*losswin*);

frame.add(panel);

*5. Set player and computer move arrays to 0:*

*playermove*[1]=0;

*playermove*[2]=0;

*playermove*[3]=0;

*playermove*[4]=0;

*playermove*[5]=0;

*playermove*[6]=0;

*playermove*[7]=0;

*playermove*[8]=0;

*playermove*[9]=0;

*compmove*[1]=0;

*compmove*[2]=0;

*compmove*[3]=0;

*compmove*[4]=0;

*compmove*[5]=0;

*compmove*[6]=0;

*compmove*[7]=0;

*compmove*[8]=0;

*compmove*[9]=0;

*compvictory*=0;

*6. Set window in the middle of the screen:*

// Get the size of the screen

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

// Determine the new location of the window

**int** w = frame.getSize().width;

**int** h = frame.getSize().height;

**int** x = (dim.width-w)/2;

**int** y = (dim.height-h)/2;

// Move the window

frame.setLocation(x, y);

*7. Sample – Button Action Listener:*

*button00*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button00*.getIcon().equals(*blank*)) {

*button00*.setIcon(*ex*);

*button00*.setDisabledIcon(*ex*);

*button00*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[1]=1;

compturn();

}

});

*8. Sample – computer victory:*

**public** **void** compvictory(){

/\* (1)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|\_

x|x|\_

x|x|

\*/

**if**(*compmove*[1]==1&&*compmove*[2]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*ohwin*);

*button01*.setDisabledIcon(*ohwin*);

*button00*.setDisabledIcon(*ohwin*);

*compmove*[3]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

*9. Sample – Stop player victory:*

**public** **void** compcounter(){

/\* (1)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|\_

x|x|\_

x|x|

\*/

**if**(*playermove*[1]==1&&*playermove*[2]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*oh*);

*button02*.setDisabledIcon(*oh*);

*button02*.setEnabled(**false**);

*compmove*[3]=1;

*compturn1*=0;

}

*10. Sample – Player moves to the corner*

**public** **void** compcorner()

{

**if**((*playermove*[1]==1||*playermove*[3]==1||*playermove*[7]==1||*playermove*[9]==1)&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn2*=0;

}

*10. Computer turn loops:*

**public** **void** compturn() {

**if** (*compvictory* == 0) {

compvictory();

**if** (*compturn* == 1) {

compcounter();

**if** (*compturn1* == 1) {

compcorner();

**if** (*compturn2* == 1) {

compcenter();

**if** (*compturn3* == 1) {

compedge();

**if** (*compturn4* == 1) {

compfailsafe();

}

}

}

}

}

}

*11.Prevents additional moves during a tie*

**if**((*playermove*[1]==1||*compmove*[1]==1)&&(*playermove*[2]==1||*compmove*[2]==1)&&(*playermove*[3]==1||*compmove*[3]==1)&&

(*playermove*[4]==1||*compmove*[4]==1)&&(*playermove*[5]==1||*compmove*[5]==1)&&(*playermove*[6]==1||*compmove*[6]==1)&&

(*playermove*[7]==1||*compmove*[7]==1)&&(*playermove*[8]==1||*compmove*[8]==1)&&(*playermove*[9]==1||*compmove*[9]==1)&&*compvictory*==0)

{

*losswin*.setText("Tie!");

}

**if**(*compvictory*==1&&*button00*.getIcon().equals(*blank*))

{

*button00*.setDisabledIcon(*blank*);

*button00*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button01*.getIcon().equals(*blank*))

{

*button01*.setDisabledIcon(*blank*);

*button01*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button02*.getIcon().equals(*blank*))

{

*button02*.setDisabledIcon(*blank*);

*button02*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button10*.getIcon().equals(*blank*))

{

*button10*.setDisabledIcon(*blank*);

*button10*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button11*.getIcon().equals(*blank*))

{

*button11*.setDisabledIcon(*blank*);

*button11*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button12*.getIcon().equals(*blank*))

{

*button12*.setDisabledIcon(*blank*);

*button12*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button20*.getIcon().equals(*blank*))

{

*button20*.setDisabledIcon(*blank*);

*button20*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button21*.getIcon().equals(*blank*))

{

*button21*.setDisabledIcon(*blank*);

*button21*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button22*.getIcon().equals(*blank*))

{

*button22*.setDisabledIcon(*blank*);

*button22*.setEnabled(**false**);

}

}

}

**FULL CODE**

**package** TicTacToe;

**import** java.util.\*;

**import** javax.swing.\*;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**import** java.awt.event.\*;

**public** **class** TicTacToe {

**public** **static** **void** main(String[] args) {

//Makes the game

TicTacToe c1 = **new** TicTacToe();

}

//All variables stated here

**public** **static** **final** **int** *FRAME\_WIDTH* = 170;

**public** **static** **final** **int** *FRAME\_HEIGHT* = 200;

**public** **static** **int** *playermove*[] = **new** **int**[10];

**public** **static** **int** *compmove*[]=**new** **int** [10];

**public** **static** **int** *compturn*=0;

**public** **static** **int** *compturn1*=0;

**public** **static** **int** *compturn2*=0;

**public** **static** **int** *compturn3*=0;

**public** **static** **int** *compturn4*=0;

**public** **static** **int** *compvictory*=0;

**public** **static** ImageIcon *blank* = **new** ImageIcon("images\\blank.png");

**public** **static** ImageIcon *ex* = **new** ImageIcon("images\\ex.png");

**public** **static** ImageIcon *oh* = **new** ImageIcon("images\\oh.png");

**public** **static** ImageIcon *ohwin* = **new** ImageIcon("images\\ohwin.png");

**public** **static** JButton *clear* = **new** JButton("Clear");

**public** **static** JButton *button00* = **new** JButton(*blank*);

**public** **static** JButton *button01* = **new** JButton(*blank*);

**public** **static** JButton *button02* = **new** JButton(*blank*);

**public** **static** JButton *button10* = **new** JButton(*blank*);

**public** **static** JButton *button11* = **new** JButton(*blank*);

**public** **static** JButton *button12* = **new** JButton(*blank*);

**public** **static** JButton *button20* = **new** JButton(*blank*);

**public** **static** JButton *button21* = **new** JButton(*blank*);

**public** **static** JButton *button22* = **new** JButton(*blank*);

**public** **static** JLabel *losswin* = **new** JLabel("Tic Tac Toe");

**public** TicTacToe() {

//Set button properties

JFrame frame = **new** JFrame("Tic Tac Toe Rev. 7 - Ali Vira");

JPanel panel = **new** JPanel();

*clear*.setPreferredSize(**new** Dimension (150,30));

*button00*.setPreferredSize(**new** Dimension(45, 30));

*button01*.setPreferredSize(**new** Dimension(45, 30));

*button02*.setPreferredSize(**new** Dimension(45, 30));

*button10*.setPreferredSize(**new** Dimension(45, 30));

*button11*.setPreferredSize(**new** Dimension(45, 30));

*button12*.setPreferredSize(**new** Dimension(45, 30));

*button20*.setPreferredSize(**new** Dimension(45, 30));

*button21*.setPreferredSize(**new** Dimension(45, 30));

*button22*.setPreferredSize(**new** Dimension(45, 30));

*losswin*.setText("Tic Tac Toe Rev. 7");

//Add things to the frame

panel.add(*button00*);

panel.add(*button01*);

panel.add(*button02*);

panel.add(*button10*);

panel.add(*button11*);

panel.add(*button12*);

panel.add(*button20*);

panel.add(*button21*);

panel.add(*button22*);

panel.add(*clear*);

panel.add(*losswin*);

frame.add(panel);

frame.setSize(*FRAME\_WIDTH*, *FRAME\_HEIGHT*);

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

frame.setVisible(**true**);

//Set all moves to zero

*playermove*[1]=0;

*playermove*[2]=0;

*playermove*[3]=0;

*playermove*[4]=0;

*playermove*[5]=0;

*playermove*[6]=0;

*playermove*[7]=0;

*playermove*[8]=0;

*playermove*[9]=0;

*compmove*[1]=0;

*compmove*[2]=0;

*compmove*[3]=0;

*compmove*[4]=0;

*compmove*[5]=0;

*compmove*[6]=0;

*compmove*[7]=0;

*compmove*[8]=0;

*compmove*[9]=0;

*compvictory*=0;

// Get the size of the screen

Dimension dim = Toolkit.*getDefaultToolkit*().getScreenSize();

// Determine the new location of the window

**int** w = frame.getSize().width;

**int** h = frame.getSize().height;

**int** x = (dim.width-w)/2;

**int** y = (dim.height-h)/2;

// Move the window

frame.setLocation(x, y);

//Sets action listeners for the buttons and disables button when clicked

*button00*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button00*.getIcon().equals(*blank*)) {

*button00*.setIcon(*ex*);

*button00*.setDisabledIcon(*ex*);

*button00*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[1]=1;

compturn();

}

});

*button01*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button01*.getIcon().equals(*blank*)) {

*button01*.setIcon(*ex*);

*button01*.setDisabledIcon(*ex*);

*button01*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[2]=1;

compturn();

}

});

*button02*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button02*.getIcon().equals(*blank*)) {

*button02*.setIcon(*ex*);

*button02*.setDisabledIcon(*ex*);

*button02*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[3]=1;

compturn();

}

});

*button10*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button10*.getIcon().equals(*blank*)) {

*button10*.setIcon(*ex*);

*button10*.setDisabledIcon(*ex*);

*button10*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[4]=1;

compturn();

}

});

*button11*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button11*.getIcon().equals(*blank*)) {

*button11*.setIcon(*ex*);

*button11*.setDisabledIcon(*ex*);

*button11*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[5]=1;

compturn();

}

});

*button12*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button12*.getIcon().equals(*blank*)) {

*button12*.setIcon(*ex*);

*button12*.setDisabledIcon(*ex*);

*button12*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[6]=1;

compturn();

}

});

*button20*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button20*.getIcon().equals(*blank*)) {

*button20*.setIcon(*ex*);

*button20*.setDisabledIcon(*ex*);

*button20*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[7]=1;

compturn();

}

});

*button21*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button21*.getIcon().equals(*blank*)) {

*button21*.setIcon(*ex*);

*button21*.setDisabledIcon(*ex*);

*button21*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[8]=1;

compturn();

}

});

*button22*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

**if** (*button22*.getIcon().equals(*blank*)) {

*button22*.setIcon(*ex*);

*button22*.setDisabledIcon(*ex*);

*button22*.setEnabled(**false**);

}

*compturn*=1;

*compturn1*=1;

*compturn2*=1;

*compturn3*=1;

*compturn4*=1;

*playermove*[9]=1;

compturn();

}

});

//Clear button, activates buttons, resets arrays, changes computer victory

*clear*.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

*compvictory*=0;

*playermove*[1]=0;

*playermove*[2]=0;

*playermove*[3]=0;

*playermove*[4]=0;

*playermove*[5]=0;

*playermove*[6]=0;

*playermove*[7]=0;

*playermove*[8]=0;

*playermove*[9]=0;

*compmove*[1]=0;

*compmove*[2]=0;

*compmove*[3]=0;

*compmove*[4]=0;

*compmove*[5]=0;

*compmove*[6]=0;

*compmove*[7]=0;

*compmove*[8]=0;

*compmove*[9]=0;

*button00*.setEnabled(**true**);

*button01*.setEnabled(**true**);

*button02*.setEnabled(**true**);

*button10*.setEnabled(**true**);

*button11*.setEnabled(**true**);

*button12*.setEnabled(**true**);

*button20*.setEnabled(**true**);

*button21*.setEnabled(**true**);

*button22*.setEnabled(**true**);

*button00*.setIcon(*blank*);

*button01*.setIcon(*blank*);

*button02*.setIcon(*blank*);

*button10*.setIcon(*blank*);

*button11*.setIcon(*blank*);

*button12*.setIcon(*blank*);

*button20*.setIcon(*blank*);

*button21*.setIcon(*blank*);

*button22*.setIcon(*blank*);

*losswin*.setText("Tic Tac Toe Rev. 7");

}

});

}

//Method for computer to win if it sees the chance

**public** **void** compvictory(){

/\* (1)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|\_

x|x|\_

x|x|

\*/

**if**(*compmove*[1]==1&&*compmove*[2]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*ohwin*);

*button01*.setDisabledIcon(*ohwin*);

*button00*.setDisabledIcon(*ohwin*);

*compmove*[3]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[4]==1&&*compmove*[5]==1&&*playermove*[6]==0&&*compmove*[6]==0)

{

*button12*.setIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button10*.setDisabledIcon(*ohwin*);

*compmove*[6]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[7]==1&&*compmove*[8]==1&&*playermove*[9]==0&&*compmove*[9]==0)

{

*button22*.setIcon(*ohwin*);

*button21*.setDisabledIcon(*ohwin*);

*button20*.setDisabledIcon(*ohwin*);

*compmove*[9]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

/\* (2)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_|x|x

\_|x|x

\_|x|x

\*/

**else** **if**(*compmove*[2]==1&&*compmove*[3]==1&&*playermove*[1]==0&&*compmove*[1]==0)

{

*button02*.setDisabledIcon(*ohwin*);

*button01*.setDisabledIcon(*ohwin*);

*button00*.setIcon(*ohwin*);

*compmove*[1]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[5]==1&&*compmove*[6]==1&&*playermove*[4]==0&&*compmove*[4]==0)

{

*button12*.setDisabledIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button10*.setIcon(*ohwin*);

*compmove*[4]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[8]==1&&*compmove*[9]==1&&*playermove*[7]==0&&*compmove*[7]==0)

{

*button22*.setDisabledIcon(*ohwin*);

*button21*.setDisabledIcon(*ohwin*);

*button20*.setIcon(*ohwin*);

*compmove*[7]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

/\* (3)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|x

x|x|x

| |

\*/

**else** **if**(*compmove*[1]==1&&*compmove*[4]==1&&*playermove*[7]==0&&*compmove*[7]==0)

{

*button20*.setIcon(*ohwin*);

*button10*.setDisabledIcon(*ohwin*);

*button00*.setDisabledIcon(*ohwin*);

*compmove*[7]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[2]==1&&*compmove*[5]==1&&*playermove*[8]==0&&*compmove*[8]==0)

{

*button21*.setIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button01*.setDisabledIcon(*ohwin*);

*compmove*[8]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[3]==1&&*compmove*[6]==1&&*playermove*[9]==0&&*compmove*[9]==0)

{

*button22*.setIcon(*ohwin*);

*button12*.setDisabledIcon(*ohwin*);

*button02*.setDisabledIcon(*ohwin*);

*compmove*[9]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

/\* (4)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_|\_|\_

x|x|x

x|x|x

\*/

**else** **if**(*compmove*[4]==1&&*compmove*[7]==1&&*playermove*[1]==0&&*compmove*[1]==0)

{

*button20*.setDisabledIcon(*ohwin*);

*button10*.setDisabledIcon(*ohwin*);

*button00*.setIcon(*ohwin*);

*compmove*[1]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[5]==1&&*compmove*[8]==1&&*playermove*[2]==0&&*compmove*[2]==0)

{

*button21*.setDisabledIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button01*.setIcon(*ohwin*);

*compmove*[2]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[6]==1&&*compmove*[9]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button22*.setDisabledIcon(*ohwin*);

*button12*.setDisabledIcon(*ohwin*);

*button02*.setIcon(*ohwin*);

*compmove*[3]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

/\* (5)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|\_|x

x|\_|x

x| |x

\*/

**else** **if**(*compmove*[1]==1&&*compmove*[3]==1&&*playermove*[2]==0&&*compmove*[2]==0)

{

*button02*.setDisabledIcon(*ohwin*);

*button01*.setIcon(*ohwin*);

*button00*.setDisabledIcon(*ohwin*);

*compmove*[2]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[4]==1&&*compmove*[6]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button10*.setDisabledIcon(*ohwin*);

*button11*.setIcon(*ohwin*);

*button12*.setDisabledIcon(*ohwin*);

*compmove*[5]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[7]==1&&*compmove*[9]==1&&*playermove*[8]==0&&*compmove*[8]==0)

{

*button20*.setDisabledIcon(*ohwin*);

*button21*.setIcon(*ohwin*);

*button22*.setDisabledIcon(*ohwin*);

*compmove*[8]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

/\* (6)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|x

\_|\_|\_

x|x|x

\*/

**else** **if**(*compmove*[1]==1&&*compmove*[7]==1&&*playermove*[4]==0&&*compmove*[4]==0)

{

*button00*.setDisabledIcon(*ohwin*);

*button10*.setIcon(*ohwin*);

*button20*.setDisabledIcon(*ohwin*);

*compmove*[4]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[2]==1&&*compmove*[8]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button01*.setDisabledIcon(*ohwin*);

*button11*.setIcon(*ohwin*);

*button21*.setDisabledIcon(*ohwin*);

*compmove*[5]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[3]==1&&*compmove*[9]==1&&*playermove*[6]==0&&*compmove*[6]==0)

{

*button02*.setDisabledIcon(*ohwin*);

*button12*.setIcon(*ohwin*);

*button22*.setDisabledIcon(*ohwin*);

*compmove*[6]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

/\* (7)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|\_|x

\_|x|\_

x| |x

\*/

**else** **if**(*compmove*[1]==1&&*compmove*[5]==1&&*playermove*[9]==0&&*compmove*[9]==0)

{

*button00*.setDisabledIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button22*.setIcon(*ohwin*);

*compmove*[9]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[7]==1&&*compmove*[5]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button20*.setDisabledIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button02*.setIcon(*ohwin*);

*compmove*[3]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[3]==1&&*compmove*[5]==1&&*playermove*[7]==0&&*compmove*[7]==0)

{

*button02*.setDisabledIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button20*.setIcon(*ohwin*);

*compmove*[7]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[9]==1&&*compmove*[5]==1&&*playermove*[1]==0&&*compmove*[1]==0)

{

*button22*.setDisabledIcon(*ohwin*);

*button11*.setDisabledIcon(*ohwin*);

*button00*.setIcon(*ohwin*);

*compmove*[1]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

/\* (8)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|\_|x

\_|\_|\_

x| |x

\*/

**else** **if**(*compmove*[1]==1&&*compmove*[9]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button22*.setDisabledIcon(*ohwin*);

*button11*.setIcon(*ohwin*);

*button00*.setDisabledIcon(*ohwin*);

*compmove*[5]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

**else** **if**(*compmove*[3]==1&&*compmove*[7]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button02*.setDisabledIcon(*ohwin*);

*button11*.setIcon(*ohwin*);

*button20*.setDisabledIcon(*ohwin*);

*compmove*[5]=1;

*losswin*.setText("You lose!");

*compvictory*=1;

*compturn*=0;

}

}

//Method to block player winning

**public** **void** compcounter()

{

/\* (1)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|\_

x|x|\_

x|x|

\*/

**if**(*playermove*[1]==1&&*playermove*[2]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*oh*);

*button02*.setDisabledIcon(*oh*);

*button02*.setEnabled(**false**);

*compmove*[3]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[4]==1&&*playermove*[5]==1&&*playermove*[6]==0&&*compmove*[6]==0)

{

*button12*.setIcon(*oh*);

*button12*.setDisabledIcon(*oh*);

*button12*.setEnabled(**false**);

*compmove*[6]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[7]==1&&*playermove*[8]==1&&*playermove*[9]==0&&*compmove*[9]==0)

{

*button22*.setIcon(*oh*);

*button22*.setDisabledIcon(*oh*);

*button22*.setEnabled(**false**);

*compmove*[9]=1;

*compturn1*=0;

}

/\* (2)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_|x|x

\_|x|x

\_|x|x

\*/

**else** **if**(*playermove*[2]==1&&*playermove*[3]==1&&*playermove*[1]==0&&*compmove*[1]==0)

{

*button00*.setIcon(*oh*);

*button00*.setDisabledIcon(*oh*);

*button00*.setEnabled(**false**);

*compmove*[1]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[5]==1&&*playermove*[6]==1&&*playermove*[4]==0&&*compmove*[6]==0)

{

*button10*.setIcon(*oh*);

*button10*.setDisabledIcon(*oh*);

*button10*.setEnabled(**false**);

*compmove*[4]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[8]==1&&*playermove*[9]==1&&*playermove*[7]==0&&*compmove*[7]==0)

{

*button20*.setIcon(*oh*);

*button20*.setDisabledIcon(*oh*);

*button20*.setEnabled(**false**);

*compmove*[7]=1;

*compturn1*=0;

}

/\* (3)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|x

x|x|x

| |

\*/

**else** **if**(*playermove*[1]==1&&*playermove*[4]==1&&*playermove*[7]==0&&*compmove*[7]==0)

{

*button20*.setIcon(*oh*);

*button20*.setDisabledIcon(*oh*);

*button20*.setEnabled(**false**);

*compmove*[7]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[2]==1&&*playermove*[5]==1&&*playermove*[8]==0&&*compmove*[8]==0)

{

*button21*.setIcon(*oh*);

*button21*.setDisabledIcon(*oh*);

*button21*.setEnabled(**false**);

*compmove*[8]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[3]==1&&*playermove*[6]==1&&*playermove*[9]==0&&*compmove*[9]==0)

{

*button22*.setIcon(*oh*);

*button22*.setDisabledIcon(*oh*);

*button22*.setEnabled(**false**);

*compmove*[9]=1;

*compturn1*=0;

}

/\* (4)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_|\_|\_

x|x|x

x|x|x

\*/

**else** **if**(*playermove*[4]==1&&*playermove*[7]==1&&*playermove*[1]==0&&*compmove*[1]==0)

{

*button00*.setIcon(*oh*);

*button00*.setDisabledIcon(*oh*);

*button00*.setEnabled(**false**);

*compmove*[1]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[5]==1&&*playermove*[8]==1&&*playermove*[2]==0&&*compmove*[2]==0)

{

*button01*.setIcon(*oh*);

*button01*.setDisabledIcon(*oh*);

*button01*.setEnabled(**false**);

*compmove*[2]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[6]==1&&*playermove*[9]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*oh*);

*button02*.setDisabledIcon(*oh*);

*button02*.setEnabled(**false**);

*compmove*[3]=1;

*compturn1*=0;

}

/\* (5)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|\_|x

x|\_|x

x| |x

\*/

**else** **if**(*playermove*[1]==1&&*playermove*[3]==1&&*playermove*[2]==0&&*compmove*[2]==0)

{

*button01*.setIcon(*oh*);

*button01*.setDisabledIcon(*oh*);

*button01*.setEnabled(**false**);

*compmove*[2]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[4]==1&&*playermove*[6]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[7]==1&&*playermove*[9]==1&&*playermove*[8]==0&&*compmove*[8]==0)

{

*button21*.setIcon(*oh*);

*button21*.setDisabledIcon(*oh*);

*button21*.setEnabled(**false**);

*compmove*[8]=1;

*compturn1*=0;

}

/\* (6)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|x|x

\_|\_|\_

x|x|x

\*/

**else** **if**(*playermove*[1]==1&&*playermove*[7]==1&&*playermove*[4]==0&&*compmove*[4]==0)

{

*button10*.setIcon(*oh*);

*button10*.setDisabledIcon(*oh*);

*button10*.setEnabled(**false**);

*compmove*[4]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[2]==1&&*playermove*[8]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[3]==1&&*playermove*[9]==1&&*playermove*[6]==0&&*compmove*[6]==0)

{

*button12*.setIcon(*oh*);

*button12*.setDisabledIcon(*oh*);

*button12*.setEnabled(**false**);

*compmove*[6]=1;

*compturn1*=0;

}

/\* (7)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|\_|x

\_|x|\_

x| |x

\*/

**else** **if**(*playermove*[1]==1&&*playermove*[5]==1&&*playermove*[9]==0&&*compmove*[9]==0)

{

*button21*.setIcon(*oh*);

*button21*.setDisabledIcon(*oh*);

*button21*.setEnabled(**false**);

*compmove*[9]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[7]==1&&*playermove*[5]==1&&*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*oh*);

*button02*.setDisabledIcon(*oh*);

*button02*.setEnabled(**false**);

*compmove*[3]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[3]==1&&*playermove*[5]==1&&*playermove*[7]==0&&*compmove*[7]==0)

{

*button20*.setIcon(*oh*);

*button20*.setDisabledIcon(*oh*);

*button20*.setEnabled(**false**);

*compmove*[7]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[9]==1&&*playermove*[5]==1&&*playermove*[1]==0&&*compmove*[1]==0)

{

*button00*.setIcon(*oh*);

*button00*.setDisabledIcon(*oh*);

*button00*.setEnabled(**false**);

*compmove*[1]=1;

*compturn1*=0;

}

/\* (8)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x|\_|x

\_|\_|\_

x| |x

\*/

**else** **if**(*playermove*[1]==1&&*playermove*[9]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn1*=0;

}

**else** **if**(*playermove*[3]==1&&*playermove*[7]==1&&*playermove*[5]==0&&*compmove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn1*=0;

}

}

//Responds if the player moves to a corner

**public** **void** compcorner()

{

System.*out*.println("compcorner");

**if**((*playermove*[1]==1||*playermove*[3]==1||*playermove*[7]==1||*playermove*[9]==1)&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn2*=0;

}

**else** **if**(*playermove*[1]==1&&*compmove*[5]==1&&*playermove*[6]==0&&*compmove*[6]==0)

{

System.*out*.println("TEST");

*button12*.setIcon(*oh*);

*button12*.setDisabledIcon(*oh*);

*button12*.setEnabled(**false**);

*compmove*[6]=1;

*compturn2*=0;

}

**else** **if**(*playermove*[3]==1&&*compmove*[5]==1&&*playermove*[4]==0&&*compmove*[4]==0)

{

System.*out*.println("TEST1");

*button10*.setIcon(*oh*);

*button10*.setDisabledIcon(*oh*);

*button10*.setEnabled(**false**);

*compmove*[4]=1;

*compturn2*=0;

}

**else** **if**(*playermove*[7]==1&&*compmove*[2]==0&&*playermove*[2]==0&&*compmove*[5]==1)

{

System.*out*.println("TEST2");

*button01*.setIcon(*oh*);

*button01*.setDisabledIcon(*oh*);

*button01*.setEnabled(**false**);

*compmove*[2]=1;

*compturn2*=0;

}

}

//Responds if the player moves to the centre

**public** **void** compcenter()

{

**if**(*playermove*[5]==1&&*compmove*[5]==0)

{

System.*out*.println("compcenter");

**if**(*playermove*[1]==0&&*compmove*[1]==0)

{

*button00*.setIcon(*oh*);

*button00*.setDisabledIcon(*oh*);

*button00*.setEnabled(**false**);

*compmove*[1]=1;

*compturn3*=0;

}

**else** **if**(*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*oh*);

*button02*.setDisabledIcon(*oh*);

*button02*.setEnabled(**false**);

*compmove*[3]=1;

*compturn3*=0;

}

**else** **if**(*playermove*[7]==0&&*compmove*[7]==0)

{

*button20*.setIcon(*oh*);

*button20*.setDisabledIcon(*oh*);

*button20*.setEnabled(**false**);

*compmove*[7]=1;

*compturn3*=0;

}

**else** **if**(*playermove*[9]==0&&*compmove*[9]==0)

{

*button22*.setIcon(*oh*);

*button22*.setDisabledIcon(*oh*);

*button22*.setEnabled(**false**);

*compmove*[9]=1;

*compturn3*=0;

}

}

}

//Responds if the player moves to an edge

**public** **void** compedge()

{

**if**(*playermove*[2]==1||*playermove*[4]==1||*playermove*[6]==1||*playermove*[8]==1)

{

System.*out*.println("compedge");

**if**(*playermove*[8]==1&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[2]==1&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[6]==1&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[4]==1&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[8]==1&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button21*.setIcon(*oh*);

*button21*.setDisabledIcon(*oh*);

*button21*.setEnabled(**false**);

*compmove*[5]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[8]==1&&*compmove*[5]==0&&*playermove*[5]==0)

{

*button21*.setIcon(*oh*);

*button21*.setDisabledIcon(*oh*);

*button21*.setEnabled(**false**);

*compmove*[5]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[8]==1&&*playermove*[6]==1&&*compmove*[5]==1&&*playermove*[9]==0&&*compmove*[9]==0)

{

*button22*.setIcon(*oh*);

*button22*.setDisabledIcon(*oh*);

*button22*.setEnabled(**false**);

*compmove*[9]=1;

*compturn4*=0;

}

}

}

//Fail-safe if above methods don't trigger

**public** **void** compfailsafe()

{

System.*out*.println("failsafe");

**if**(*playermove*[1]==0&&*compmove*[1]==0)

{

*button00*.setIcon(*oh*);

*button00*.setDisabledIcon(*oh*);

*button00*.setEnabled(**false**);

*compmove*[1]=1;

*compturn4*=0;

}

**else** **if** (*playermove*[2]==0&&*compmove*[2]==0)

{

*button01*.setIcon(*oh*);

*button01*.setDisabledIcon(*oh*);

*button01*.setEnabled(**false**);

*compmove*[2]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[3]==0&&*compmove*[3]==0)

{

*button02*.setIcon(*oh*);

*button02*.setDisabledIcon(*oh*);

*button02*.setEnabled(**false**);

*compmove*[3]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[4]==0&&*compmove*[4]==0)

{

*button10*.setIcon(*oh*);

*button10*.setDisabledIcon(*oh*);

*button10*.setEnabled(**false**);

*compmove*[4]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[5]==0&&*compmove*[5]==0)

{

*button11*.setIcon(*oh*);

*button11*.setDisabledIcon(*oh*);

*button11*.setEnabled(**false**);

*compmove*[5]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[6]==0&&*compmove*[6]==0)

{

*button12*.setIcon(*oh*);

*button12*.setDisabledIcon(*oh*);

*button12*.setEnabled(**false**);

*compmove*[6]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[7]==0&&*compmove*[7]==0)

{

*button20*.setIcon(*oh*);

*button20*.setDisabledIcon(*oh*);

*button20*.setEnabled(**false**);

*compmove*[7]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[8]==0&&*compmove*[8]==0)

{

*button21*.setIcon(*oh*);

*button21*.setDisabledIcon(*oh*);

*button21*.setEnabled(**false**);

*compmove*[8]=1;

*compturn4*=0;

}

**else** **if**(*playermove*[9]==0&&*compmove*[9]==0)

{

*button22*.setIcon(*oh*);

*button22*.setDisabledIcon(*oh*);

*button22*.setEnabled(**false**);

*compmove*[9]=1;

*compturn4*=0;

}

}

//Computer turn method, executes above algorithms

**public** **void** compturn() {

**if** (*compvictory* == 0) {

compvictory();

**if** (*compturn* == 1) {

compcounter();

**if** (*compturn1* == 1) {

compcorner();

**if** (*compturn2* == 1) {

compcenter();

**if** (*compturn3* == 1) {

compedge();

**if** (*compturn4* == 1) {

compfailsafe();

}

}

}

}

}

}

//Freezes the board from moving if there is a tie

**if** ((*playermove*[1]==1||*compmove*[1]==1)&&(*playermove*[2]==1||*compmove*[2]==1)&&(*playermove*[3]==1||*compmove*[3]==1)&&

(*playermove*[4]==1||*compmove*[4]==1)&&(*playermove*[5]==1||*compmove*[5]==1)&&(*playermove*[6]==1||*compmove*[6]==1)&&

(*playermove*[7]==1||*compmove*[7]==1)&&(*playermove*[8]==1||*compmove*[8]==1)&&(*playermove*[9]==1||*compmove*[9]==1)&&*compvictory*==0)

{

*losswin*.setText("Tie!");

}

**if**(*compvictory*==1&&*button00*.getIcon().equals(*blank*))

{

*button00*.setDisabledIcon(*blank*);

*button00*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button01*.getIcon().equals(*blank*))

{

*button01*.setDisabledIcon(*blank*);

*button01*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button02*.getIcon().equals(*blank*))

{

*button02*.setDisabledIcon(*blank*);

*button02*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button10*.getIcon().equals(*blank*))

{

*button10*.setDisabledIcon(*blank*);

*button10*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button11*.getIcon().equals(*blank*))

{

*button11*.setDisabledIcon(*blank*);

*button11*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button12*.getIcon().equals(*blank*))

{

*button12*.setDisabledIcon(*blank*);

*button12*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button20*.getIcon().equals(*blank*))

{

*button20*.setDisabledIcon(*blank*);

*button20*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button21*.getIcon().equals(*blank*))

{

*button21*.setDisabledIcon(*blank*);

*button21*.setEnabled(**false**);

}

**if**(*compvictory*==1&&*button22*.getIcon().equals(*blank*))

{

*button22*.setDisabledIcon(*blank*);

*button22*.setEnabled(**false**);

}

}

}