Noughts and Crosses Game for Android Devices

My program

Upon Opening

On opening up the application the user is shown the title ‘Noughts And Crosses’ at the top of the screen, this shows clearly what the application is as everyone has heard of the game. The title is clearly visible in 25sp, bold, white writing on a purple background.

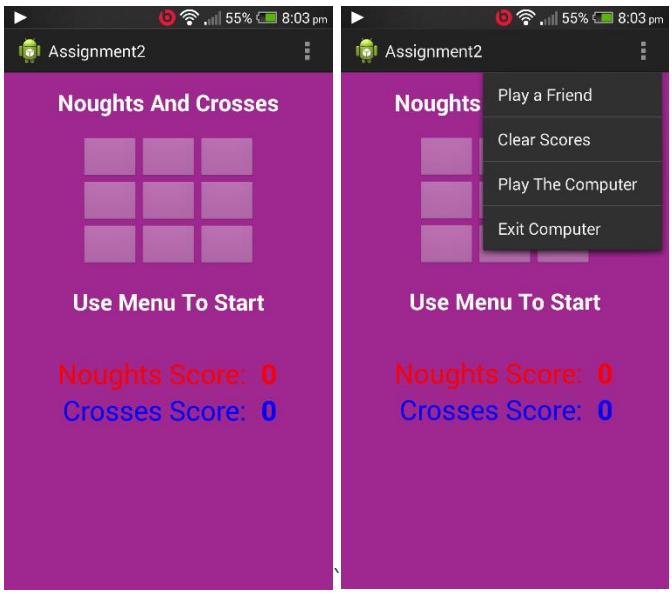
Below this is a grid of buttons (3x3) which are disabled so cannot be pressed, they are equally spaced and centred horizontally on the screen so it doesn’t look messy.

Underneath the grid is another textView, this is very similar to the title text, same size, colour and is also centred and in bold, this helps the application look simple and formal. This textView displays the message ‘Use Menu To Start’ which clearly instructs the user that they must use the menu to start the game. This message will change throughout the game to give the user commands such as whose go it is, the result of a game etc.

At the bottom is the results textViews, they display the current score so far. As the application has just been opened they show ‘Noughts Score:’ ‘0’ (in red) and Crosses Score:’ ‘0’ (in blue). The textViews are colour coded because they correspond to the O’s (red) and X’s (blue) placed on the screen which easily shows who is winning. When a player (or the computer) wins a game the score of the corresponding score is increased by 1.

The menu shows 4 options ‘Play a Friend’, ‘Clear Scores’, ‘Play The Computer’, and ‘Exit Game’. If the user is in the middle of a game the only option available is ‘Exit application’ which simply exits the application. Trying to select a different option during a game displays a toast notification saying ‘You Must Finish The Game First’. But out of a game selecting ‘Play a Friend’ Starts a 2 player game as described below.

App upon opening **With Menu Open**



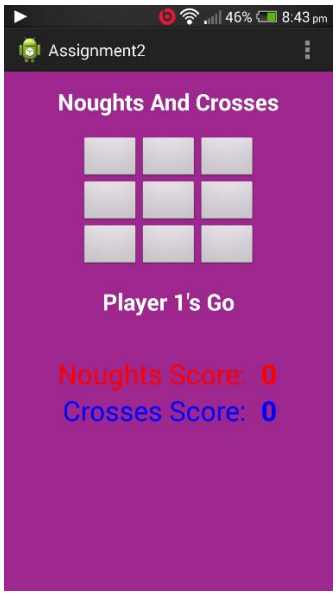
After Pressing 'Play A Friend' on the Menu

If you select 'Play a Friend' on the menu while currently out of a game the grid of buttons will enable allowing the first player to select where on the grid they would like to place their nought, the TextView that originally displayed 'Use Menu To Start' will also change to 'Player 1's Go' in case they're unsure what to do or have forgotten who's go it is.

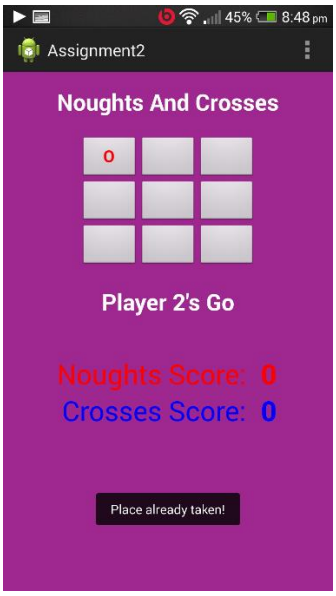
After the first player takes there go by selecting a button on the grid a O in red will be placed in there box and if they haven't won the textView will be change to 'Player 2's go' and the second player will be able to select a button on the grid to place their X.

After the second player takes their go a blue X will be placed in the button on the grid they selected (unless they chose a place already taken in which case a toast notification will display 'Place already Taken!' and they will have to place it somewhere else to end their turn) and they haven't won it will go back to the first players go and this will go on until a win or draw is detected.

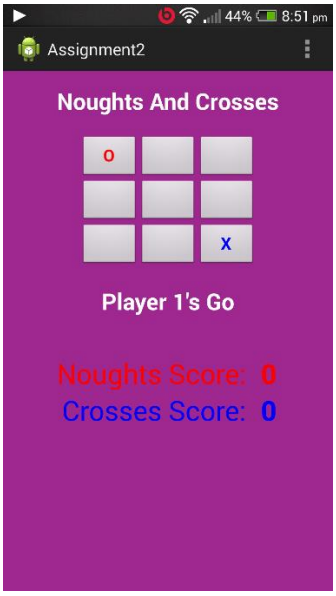
Start of Game



Toast message

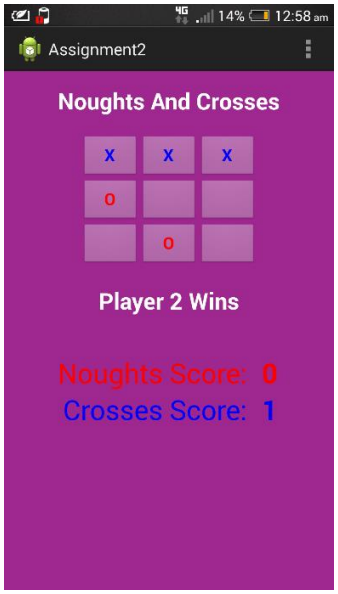


back to Player 1's go

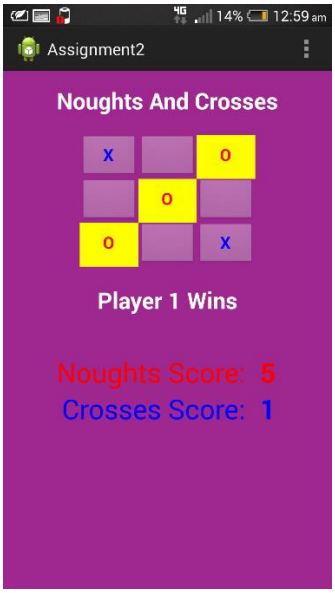


Once a win is detected the grid of buttons is disabled because the game has ended, the textView displays the message 'Player (1 or 2) wins' (or 'Computer Wins' if playing against the computer), 1 is added to the score of the winner and the winning line on the grid flashes 3 times to show the winning spot. If the game ends in a draw the game ends as usual without changing the score and displaying draw in the textView.

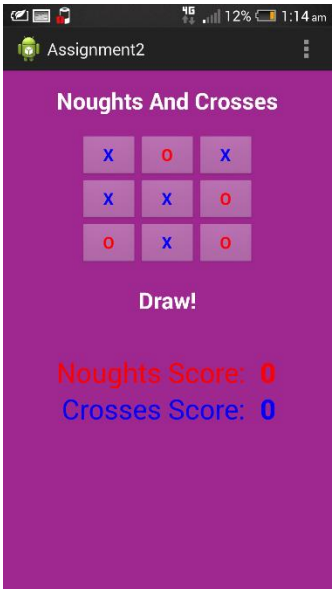
After Winning the Game



Winning Line Flashing

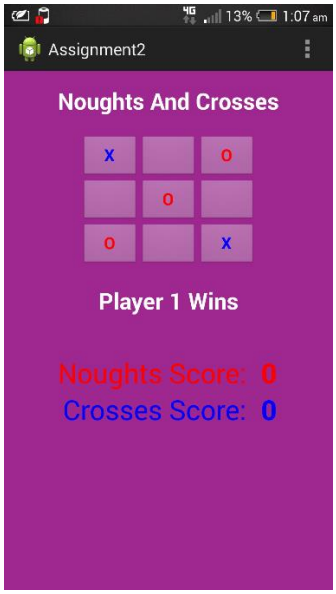


After a Draw



The user can then use the menu to play another game or clear the score.

After Clearing the Score



Playing the Computer

If you choose to play against the computer after you take your turn as normal but after you do the computer automatically takes its turn placing a blue X in an available place on the grid.

Added Feature

I have added a feature which alternates who gets to go first, when the application is first run player 1 (Noughts) will get the first go, however if he plays another game either the second player or the computer (both Crosses) will take the first go. I have added this feature as in real life this is how you would play as it makes it fair for both players (and the computer).

The Code

Activity_main.XML

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/purple"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="@string/title"
        android:textColor="@color/white"
        android:textSize="25sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/TopM"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="21dp"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <Button
        android:id="@+id/MidL"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/MidM"
        android:layout_alignBottom="@+id/MidM"
        android:layout_alignRight="@+id/TopL"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <Button
        android:id="@+id/MidR"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/MidM"
        android:layout_alignBottom="@+id/MidM"
        android:layout_alignLeft="@+id/TopR"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <Button
        android:id="@+id/BotM"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/MidM"
        android:layout_below="@+id/MidM"
        android:background="@android:drawable/btn_default"
```

```
        android:textStyle="bold" />

    <Button
        android:id="@+id/MidM"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/TopM"
        android:layout_below="@+id/TopM"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <Button
        android:id="@+id/BotL"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/BotM"
        android:layout_alignBottom="@+id/BotM"
        android:layout_alignLeft="@+id/MidL"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <Button
        android:id="@+id/BotR"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/MidR"
        android:layout_alignTop="@+id/BotM"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <Button
        android:id="@+id/TopR"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/MidR"
        android:layout_toRightOf="@+id/TopM"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <Button
        android:id="@+id/TopL"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/TopM"
        android:layout_alignBottom="@+id/TopM"
        android:layout_toLeftOf="@+id/TopM"
        android:background="@android:drawable/btn_default"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/nScore"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView2"
        android:layout_marginLeft="16dp"
        android:layout_toRightOf="@+id/textView2"
        android:text="0"
        android:textColor="@color/red"
        android:textSize="30sp"
        android:textStyle="bold" />
```

```
    <TextView
        android:id="@+id/cScore"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView3"
        android:layout_alignBottom="@+id/textView3"
        android:layout_alignLeft="@+id/nScore"
        android:text="0"
        android:textColor="@color/blue"
        android:textSize="30sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView2"
        android:layout_toLeftOf="@+id/nScore"
        android:text="@string/txtCScore"
        android:textColor="@color/blue"
        android:textSize="30sp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_below="@+id/txtState"
        android:layout_marginTop="42dp"
        android:text="@string/txtNScore"
        android:textColor="@color/red"
        android:textSize="30sp" />

    <TextView
        android:id="@+id/txtState"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/BotM"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:text="@string/state"
        android:textColor="@color/white"
        android:textSize="25sp"
        android:textStyle="bold" />

</RelativeLayout>
```

Strings.XML

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Assignment2</string>
    <string name="action_settings">Settings</string>
    <string name="title">Noughts And Crosses</string>
     <string name="state">Use Menu To Start</string>
    <string name="txtNScore">Noughts Score:</string>
    <string name="txtCScore">Crosses Score:</string>

    <string name="menuNewGame">Play a Friend</string>
```

```
    <string name="menuClear">Clear Scores</string>
    <string name="menuCompGame">Play The Computer</string>
    <string name="menuExit">Exit Game</string>

    <color name="white">#FFFFFF</color>
    <color name="blue">#0000FF</color>
    <color name="red">#FF0000</color>
    <color name="purple">#9E2890</color>
    <color name="yellow">#FFFF00</color>

</resources>
```

Options_menu.XML

```
<?xml version="1.0" encoding="UTF-8"?>
    <menu xmlns:android="http://schemas.android.com/apk/res/android">
        <item android:id="@+id/TwoPlayer"
android:title="@string/menuNewGame"></item>
        <item android:id="@+id/Clear"
android:title="@string/menuClear"></item>
        <item android:id="@+id/OnePlayer"
android:title="@string/menuCompGame"></item>
        <item android:id="@+id/ExitGame"
android:title="@string/menuExit"></item>
    </menu>
```

MainActivity.Java

```
package com.example.assignment2;

import java.util.Random;

import java.util.Timer;

import java.util.TimerTask;

import android.R.color;

import android.R.drawable;

import android.os.Bundle;

import android.os.Handler;

import android.app.Activity;

import android.graphics.Color;

import android.graphics.drawable.Drawable;

import android.view.ContextMenu;

import android.view.ContextMenu.ContextMenuInfo;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.View.OnClickListener;
```

```
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity implements OnClickListener{
	//declare Variables for buttons, textviews, ints etc.
	private TextView txtNScore, txtCScore, txtState;
	private Button TopL, TopM, TopR, MidL, MidM, MidR, BotL, BotM, BotR, place, flash1, flash2, flash3;
	private Random random;
	private Timer timer;
	private Handler handler;
	private int NoughtWins, CrossWins, bClicked;
	private int whosGo = 1;
	private String go;
	private boolean stop, compgame, menuEnabled;
	private String[] boardArray = new String[10];
	private int counter = 0;
	private Drawable buttonColor;

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);
		//make declared vars equal to the buttons and textviews
		txtNScore = (TextView) findViewById(R.id.nScore);
		txtCScore = (TextView) findViewById(R.id.cScore);
		txtState = (TextView) findViewById(R.id.txtState);

		TopL = (Button) findViewById(R.id.TopL);
		TopM = (Button) findViewById(R.id.TopM);
		TopR = (Button) findViewById(R.id.TopR);
		MidL = (Button) findViewById(R.id.MidL);
		MidM = (Button) findViewById(R.id.MidM);
```

```
        MidR = (Button) findViewById(R.id.MidR);
        BotL = (Button) findViewById(R.id.BotL);
        BotM = (Button) findViewById(R.id.BotM);
        BotR = (Button) findViewById(R.id.BotR);

        random = new Random();
        buttonColor = TopL.getBackground();
        handler = new Handler();
        //empty the N&C's board and diable clicking it until a game is started
        resetBoard();
        disableBoard();
        //add button click listeners
        TopL.setOnClickListener(this);
        TopM.setOnClickListener(this);
        TopR.setOnClickListener(this);
        MidL.setOnClickListener(this);
        MidM.setOnClickListener(this);
        MidR.setOnClickListener(this);
        BotL.setOnClickListener(this);
        BotM.setOnClickListener(this);
        BotR.setOnClickListener(this);
        //makes user be able to select from menu
        menuEnabled = true;
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.options_menu, menu);
        return true;
    }
    public boolean onOptionsItemSelected(MenuItem item){
        switch(item.getItemId()){
```

```
                //when an option is selected do the function.
                        case R.id.TwoPlayer:
                                if (menuEnabled == true) {
                                        //go to the method
                                        doNewGame();
                                }
                                else
                                {
                                        //if in-game dont allow player to use menu (other than to exit)
                                        Toast.makeText(this,"You Must Finish The Game First",
Toast.LENGTH_SHORT).show();
                                }
                                return true;
                        case R.id.Clear:
                                if (menuEnabled == true) {
                                        //go to the method
                                        doClearGame();
                                }
                                else
                                {
                                        //if in-game dont allow player to use menu (other than to exit)
                                        Toast.makeText(this,"You Must Finish The Game First",
Toast.LENGTH_SHORT).show();
                                }
                                return true;
                        case R.id.OnePlayer:
                                if (menuEnabled == true) {
                                        //go to the method
                                        doCompGame();
                                }
                                else
                                {
                                        //if in-game dont allow player to use menu (other than to exit)
```

```
                                        Toast.makeText(this,"You Must Finish The Game First",
Toast.LENGTH_SHORT).show();
                                }
                                return true;
                        case R.id.ExitGame:
                                //exit application
                                this.finish();
                                return true;
                        default:return false;
                }
        }

        public void doNewGame(){
                //play game with a friend
                //set compgame to false as youre not playing with the computer
                compgame = false;
                //stop is set to false during a game
                stop = false;
                //empty the N&C's board
                resetBoard();
                //disable the menu to start games, clear score and play the computer during the game.
                menuEnabled = false;
                //enable the board so players can click places
                enableBoard();
                //whosGo changes who goes first in the game, which changes game to game
                //set go to which player goes first then change WhoGos so next game thwe other will go first
                if (whosGo == 1) {
                        go = "player1";
                        txtState.setText("Player 1's Go");
                        whosGo = 2;
                }else
                {
                        go = "player2";
```

```
                txtState.setText("Player 2's Go");
                whosGo = 1;
            }
        }
        //reset the scores for both players to 0
        public void doClearGame() {
            //clear scores set back to 0
            NoughtWins = 0;
            CrossWins = 0;
            txtCScore.setText(String.valueOf(CrossWins));
            txtNScore.setText(String.valueOf(NoughtWins));
        }
        //play a game against the computer
        public void doCompGame() {
            //compgame set to true so computer takes player 2's go instead
            compgame = true;
            //stop set to false during the game
            stop = false;
            //empty N&C's board
            resetBoard();
            //disable menus during
            menuEnabled = false;
            //enables the N&C's board
            enableBoard();
            //sets who goes first this game
            if (whosGo == 1) {
                go = "player1";
                txtState.setText("Player 1's Go");
                whosGo = 2;
            }
            else
            {
                go = "player2";
```

```
                txtState.setText("Computer's Go");
                whosGo = 1;
                compGo();
            }
        }
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            if (v==TopL) {
                //place equals the button pressed
                place = TopL;
                bClicked = 1;
                takeSpot();
            }
            if (v==TopM) {
                //place equals the button pressed
                place = TopM;
                bClicked = 2;
                takeSpot();
            }
            if (v==TopR) {
                //place equals the button pressed
                place = TopR;
                bClicked = 3;
                takeSpot();
            }
            if (v==MidL) {
                //place equals the button pressed
                place = MidL;
                bClicked = 4;
                takeSpot();
            }
            if (v==MidM) {
```

```
                //place equals the button pressed
                place = MidM;
                bClicked = 5;
                takeSpot();
            }
            if (v==MidR) {
                //place equals the button pressed
                place = MidR;
                bClicked = 6;
                takeSpot();
            }
            if (v==BotL) {
                //place equals the button pressed
                place = BotL;
                bClicked = 7;
                takeSpot();
            }
            if (v==BotM) {
                //place equals the button pressed
                place = BotM;
                bClicked = 8;
                takeSpot();
            }
            if (v==BotR) {
                //place equals the button pressed
                place = BotR;
                bClicked = 9;
                takeSpot();
            }
        }

        public void disableBoard() {
            //disables all buttons on the N&C's board
```

```
                TopL.setEnabled(false);
                TopM.setEnabled(false);
                TopR.setEnabled(false);
                MidL.setEnabled(false);
                MidM.setEnabled(false);
                MidR.setEnabled(false);
                BotL.setEnabled(false);
                BotM.setEnabled(false);
                BotR.setEnabled(false);

        }
        public void enableBoard() {
                //enables all buttons on the N&C's board
                TopL.setEnabled(true);
                TopM.setEnabled(true);
                TopR.setEnabled(true);
                MidL.setEnabled(true);
                MidM.setEnabled(true);
                MidR.setEnabled(true);
                BotL.setEnabled(true);
                BotM.setEnabled(true);
                BotR.setEnabled(true);

        }
public void resetBoard() {
                //empty the board
                TopL.setText("");
                TopM.setText("");
                TopR.setText("");
                MidL.setText("");
                MidM.setText("");
                MidR.setText("");
                BotL.setText("");
```

```
            BotM.setText("");
            BotR.setText("");
        }

        public void takeSpot() {
            if (place.getText() == "") {
                //if the place on the N&C's board is free
                if (go=="player1") {
                    //if its player 1s go
                    //place a O on the button selected and set that place in the array to O
                    place.setText("O");
                    boardArray[bClicked] = "O";
                    place.setTextColor(Color.RED);
                    //check win to see if this move has won the game
                    checkWin();
                    //if game not finished change to player 2s go
                    if (stop == false) {
                        go = "player2";
                        txtState.setText("Player 2's Go");
                    }
                    //if playing the computer change to the computers go
                    if (compgame == true){
                        go = "player2";
                        txtState.setText("Computer's Go");
                        //do the computers go
                        compGo();
                    }
                }
                else
                {
                    //if player 2's go, do the same but width an X
                    place.setText("X");
                    boardArray[bClicked] = "X";
```

```
                                place.setTextColor(Color.BLUE);
                                checkWin();
                                //change to player 2's go
                                if (stop == false) {
                                        go = "player1";
                                        txtState.setText("Player 1's Go");
                                }
                        }
                }
                else
                {
                        //toast message saying place taken
                        Toast.makeText(this,"Place already taken!", Toast.LENGTH_SHORT).show();
                }
        }
        public void checkWin() {
                //check if there is a win on the top row
                if (TopL.getText() != "" && TopL.getText() == TopM.getText() && TopM.getText() ==
TopR.getText())
                {
                        //set the winning places to flash1,2,3
                        flash1 = TopL;
                        flash2 = TopM;
                        flash3 = TopR;
                        //finish game
                        win();
                }
                //check if there is a win on the middle row
                if (MidL.getText() != "" && MidL.getText() == MidM.getText() && MidM.getText() ==
MidR.getText())
                {
                        //set the winning places to flash1,2,3
                        flash1 = MidL;
                        flash2 = MidM;
```

```
                flash3 = MidR;
                //finish game
                win();
            }
            //check if there is a win on the bottom row
            if (BotL.getText() != "" && BotL.getText() == BotM.getText() && BotM.getText() ==
BotR.getText())
            {
                //set the winning places to flash1,2,3
                flash1 = BotL;
                flash2 = BotM;
                flash3 = BotR;
                //finish game
                win();
            }
            //check if there is a win on the left column
            if (TopL.getText() != "" && TopL.getText() == MidL.getText() && MidL.getText() ==
BotL.getText())
            {
                //set the winning places to flash1,2,3
                flash1 = TopL;
                flash2 = MidL;
                flash3 = BotL;
                //finish game
                win();
            }
            //check if there is a win on the middle column
            if (TopM.getText() != "" && TopM.getText() == MidM.getText() && MidM.getText() ==
BotM.getText())
            {
                //set the winning places to flash1,2,3
                flash1 = TopM;
                flash2 = MidM;
                flash3 = BotM;
```

```
                    //finish game
                    win();
                }
                //check if there is a win on the right column
                if (TopR.getText() != "" && TopR.getText() == MidR.getText() && MidR.getText() ==
BotR.getText())
                {
                    //set the winning places to flash1,2,3
                    flash1 = TopR;
                    flash2 = MidR;
                    flash3 = BotR;
                    //finish game
                    win();
                }
                //check if there is a win on the diagonal going left-to-right
                if (TopL.getText() != "" && TopL.getText() == MidM.getText() && MidM.getText() ==
BotR.getText())
                {
                    //set the winning places to flash1,2,3
                    flash1 = TopL;
                    flash2 = MidM;
                    flash3 = BotR;
                    //finish game
                    win();
                }
                //check if there is a win on the diagonal going right-to-left
                if (TopR.getText() != "" && TopR.getText() == MidM.getText() && MidM.getText() ==
BotL.getText())
                {
                    //set the winning places to flash1,2,3
                    flash1 = TopR;
                    flash2 = MidM;
                    flash3 = BotL;
                    //finish game
```

```
                win();
            }
            if (TopL.getText() != "" && TopM.getText() != "" && TopR.getText() != "" && MidL.getText() != "" && MidM.getText() != "" && MidR.getText() != "" && BotL.getText() != "" && BotM.getText() != "" && BotR.getText() != "") {
                //if all locations on the board have been used and there is no winner end game in a tie
                if (stop != true) {
                    //reenable the buttons
                    menuEnabled = true;
                    //disable the board
                    disableBoard();
                    //stop game
                    stop = true;
                    //end compgame
                    compgame = false;
                    //write to screen that it was a draw
                    txtState.setText("Draw!");
                }

            }

        }
        public void win() {
            //the game has been won!
            if (go == "player1") {
                //if player 1's go then display message saying player 1 wins add one to there score
                txtState.setText("Player 1 Wins");
                NoughtWins++;
                txtNScore.setText(String.valueOf(NoughtWins));
            }
            else {
                if (compgame == true) {
                    //else if this is against the computer then display the computer wins
```

```
                txtState.setText("Computer Wins");
            }
            else
            {
                //else player 2 must have won
                txtState.setText("Player 2 Wins");
            }
            //so increase player 2/computers score
            CrossWins++;
            txtCScore.setText(String.valueOf(CrossWins));
        }
        //reenable buttons
        menuEnabled = true;
        //flash meathod is repeated until counter > 5
        handler.postDelayed(new Runnable() {
            public void run(){

                flash();
                handler.postDelayed(this, 400);
                counter++;
                if (counter > 5){
                    //end and set counter back to 0 for next time
                    handler.removeCallbacks(this);
                    counter=0;
                }
            }
        }, 400);

        //disable the N&C's board
        disableBoard();
        //end game by stop = true
        stop = true;
        //end comp game
```

```
            compgame = false;
            for (int i=0; i<10; i++) {
                  //set the values of the array back to null for next game
                  boardArray[i] = "";
            }


}
public void flash() {
            //flash 1, 2, 3 are the winning places, make them flash 3 times and return to normal
            switch (counter) {
            case 0: flash1.setBackgroundColor(Color.YELLOW);
                              flash2.setBackgroundColor(Color.YELLOW);
                              flash3.setBackgroundColor(Color.YELLOW);
                  break;
            case 1: flash1.setBackgroundDrawable(buttonColor);
                              flash2.setBackgroundDrawable(buttonColor);
                              flash3.setBackgroundDrawable(buttonColor);
                  break;
            case 2: flash1.setBackgroundColor(Color.YELLOW);
                              flash2.setBackgroundColor(Color.YELLOW);
                              flash3.setBackgroundColor(Color.YELLOW);
                  break;
            case 3: flash1.setBackgroundDrawable(buttonColor);
                              flash2.setBackgroundDrawable(buttonColor);
                              flash3.setBackgroundDrawable(buttonColor);
                  break;
            case 4: flash1.setBackgroundColor(Color.YELLOW);
                              flash2.setBackgroundColor(Color.YELLOW);
                              flash3.setBackgroundColor(Color.YELLOW);
                  break;
            case 5: flash1.setBackgroundDrawable(buttonColor);
                              flash2.setBackgroundDrawable(buttonColor);
```

```
                    flash3.setBackgroundDrawable(buttonColor);
                break;
            }
        }
        public void compGo() {
            //disable board during comps go
            disableBoard();
            //create int sqaure
            int square;
            //set compPlayed to false
            boolean compPlayed = false;
            //while compPlayed is false do a loop
            while (compPlayed == false){
                //set square to random number between 1-9
                square = random.nextInt(9)+1;
                //if the spot on the array matching rhe random number is free place comps X in that
place
                if (boardArray[square] != "X" && boardArray[square] != "O")
                {
                    switch (square) {
                    case 1: place = TopL;
                    break;
                    case 2: place = TopM;
                    break;
                    case 3: place = TopR;
                    break;
                    case 4: place = MidL;
                    break;
                    case 5: place = MidM;
                    break;
                    case 6: place = MidR;
                    break;
                    case 7: place = BotL;
```

```
                                        break;
                                        case 8: place = BotM;
                                        break;
                                        case 9: place = BotR;
                                        break;
                                        }
                                        boardArray[square] = "X";
                                        place.setText("X");
                                        place.setTextColor(Color.BLUE);
                                        //set compPlayed to true to end loop
                                        compPlayed = true;
                                }
                                //else redo loop until computer has placed a X
                        }
                        //check if comp has won
                        checkWin();
                        //if game not won set to player 1's go and enable the board
                                if (stop == false){
                                        go = "player1";
                                        txtState.setText("Player 1's Go");
                                        enableBoard();
                                }

        }
}
```