Steps to create a Tic-Tac-Toe game

First, find and download (or create) an image we can use for the game's icon. Create a new project named “**Tic-Tac-Toe**” using the package name of “**edu.uah.pcs.android**”. Select the image using the wizard and at the final step ask the wizard to create a blank activity.

Next, edit the ***activity_main.xml*** file to read:

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_margin="30dp"
    android:orientation="vertical"
    android:padding="30dip"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textViewTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="25dp"
        android:text="@string/main_title"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textSize="24sp" />

    <Button
        android:id="@+id/continue_button"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/continue_label" />

    <Button
        android:id="@+id/new_button"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/new_game_label" />

    <Button
        android:id="@+id/about_button"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/about_label" />

    <Button
        android:id="@+id/exit_button"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/exit_label" />

</LinearLayout>
```

View the layout in landscape mode and see that not all buttons are visible. Create a new file named ***res/layout-land/activity_main.xml*** and edit to read:

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp" >

    <TextView
        android:id="@+id/textViewTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/main_title"
        android:textSize="24sp" />

    <TableLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:stretchColumns="*" >

        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

            <Button
                android:id="@+id/continue_button"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/continue_label" />

            <Button
                android:id="@+id/new_button"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/new_game_label" />
        </TableRow>

        <TableRow
            android:id="@+id/tableRow2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

            <Button
                android:id="@+id/about_button"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/about_label" />

            <Button
                android:id="@+id/exit_button"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/exit_label" />
        </TableRow>
    </TableLayout>
```

```
</LinearLayout>
```

Create a new layout file name ***res/layout/about.xml*** and edit to read:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp" >

    <TextView
        android:id="@+id/textViewAbout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/about_text" />

</ScrollView>
```

Update the ***res/values/strings.xml*** to read:

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

    <string name="app_name">Tictactoe</string>
    <string name="action_settings">Settings</string>
    <string name="settings_title">Tic-Tac-Toe Settings</string>
    <string name="settings_shortcut">s</string>
    <string name="colors_title">Inverted Colors</string>
    <string name="colors_summary">Use light or dark colors for the board.</string>
    <string name="main_title">Tic-Tac-Toe</string>
    <string name="continue_label">Continue</string>
    <string name="new_game_label">New Game</string>
    <string name="about_label">About</string>
    <string name="exit_label">Exit</string>
    <string name="about_title">About Tic-Tac-Toe</string>
    <string name="about_text">\
Tic-tac-toe is game of 9 squares where players alternately
place an \'X\' or an \'O\' on a <i>3x3</i> grid.  To win, you must
get three X\'s or O\'s in a row.</string>

</resources>
```

Also, create the file ***res/values/colors.xml*** and edit to read:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="background">#3500FFFF</color>
</resources>
```

Next, create a new class named **About** that extends *android.app.Activity* and edit to read:

```
package com.bamafolks.android.games.tictactoe;

import android.app.Activity;
import android.os.Bundle;
```

```
public class About extends Activity {

      @Override
      protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.about);
      }

}
```

Modify the ***MainActivity.java*** source file to read:

```
package com.bamafolks.android.games.tictactoe;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;

public class MainActivity extends Activity implements OnClickListener {

      @Override
      protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);

            findViewById(R.id.continue_button).setOnClickListener(this);
            findViewById(R.id.new_button).setOnClickListener(this);
            findViewById(R.id.about_button).setOnClickListener(this);
            findViewById(R.id.exit_button).setOnClickListener(this);

      }

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

      @Override
      public void onClick(View v) {
            switch(v.getId()) {
            case R.id.about_button:
                  Intent i = new Intent(this, About.class);
                  startActivity(i);
                  break;
                  // More code later...
            }

      }
}
```

Run the application using an emulator and click the **About** button. It should crash, because we did not add the **About** activity to the manifest file. Fix that and try again.

Next, change the About activity's theme to **@android:style/Theme.Dialog** in the manifest and run the application again to see the change in style.

Add the following to the ***res/values/strings.xml*** file:

```
    <string name="settings_title">Tic-Tac-Toe Settings</string>
    <string name="settings_shortcut">s</string>
    <string name="colors_title">Light Colors</string>
    <string name="colors_summary">Use light or dark colors for the board.</string>
```

Use the wizard (New → Android XML File) to create the ***res/xml/settings.xml*** file of type **PreferenceScreen**. Use the editor to add a **Checkbox** preference like this:

```
    <CheckBoxPreference android:key="color"
android:summary="@string/colors_summary" android:title="@string/colors_title"
android:defaultValue="true"/>
```

Create the class **Prefs** that extends the *android.preference.PreferenceActivity* class and edit to read:

```
package com.bamafolks.android.games.tictactoe;

import android.os.Bundle;
import android.preference.PreferenceActivity;

public class Prefs extends PreferenceActivity {

      @Override
      protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            addPreferencesFromResource(R.xml.settings);
      }

}
```

Next, add the following code to the ***MainActivity.java*** file:

```
      @Override
      public boolean onOptionsItemSelected(MenuItem item) {
            switch(item.getItemId()) {
            case R.id.action_settings:
                  startActivity(new Intent(this, Prefs.class));
                  return true;
                  // Add more as needed
            }
            return false;
      }
```

Register the new **Prefs** activity in the manifest and test it out. We will add code to load the use the setting later.

Add the following strings to the ***res/values/strings.xml*** file:

```
    <string name="player_label">Player</string>
    <string name="computer_label">Computer</string>
    <string name="random_label">Random</string>
```

Create the file ***res/values/arrays.xml*** and add the following:

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

    <array name="first_move">
        <item>@string/player_label</item>
        <item>@string/computer_label</item>
        <item>@string/random_label</item>
    </array>

</resources>
```

Edit the *onClick* method in ***MainActivity.java*** file by adding this case:

```
            case R.id.new_button:
                  openNewGameDialog();
                  break;
```

Now add the following methods to the end of the **MainActivity** class:

```
      private void openNewGameDialog() {
            new AlertDialog.Builder(this)
            .setTitle(R.string.new_game_title)
            .setItems(R.array.first_move, new DialogInterface.OnClickListener() {

                  @Override
                  public void onClick(DialogInterface dialog, int which) {
                        startGame(which);
                  }
            });
      }

      protected void startGame(int which) {
            Log.d(this.getClass().getSimpleName(), "clicked on " + which);
            // Start game here...
      }
```

Implement the **Exit** button by adding this code to the *onClick* method in the **MainActivity** class:

```
            case R.id.exit_button:
                  finish();
                  break;
```

Run the game and try the new game and exit buttons.

Now we need to work on drawing the board and using some of Android's graphics features. Start by completing the *startGame* method in the **MainActivity** class:

```
    protected void startGame(int which) {
        Log.d(this.getClass().getSimpleName(), "clicked on " + which);
        Intent intent = new Intent(this, Game.class);
        intent.putExtra(Game.FIRST_MOVE, which);
        startActivity(intent);
    }
```

Next, create a new **Game** class that extends *Activity* and add the following code:

```
package com.bamafolks.android.games.tictactoe;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class Game extends Activity {

    public static final String FIRST_MOVE =
"com.bamafolks.android.games.tictactoe.first_move";
    public static final int PLAYER_FIRST = 0;
    public static final int COMPUTER_FIRST = 1;
    public static final int RANDOM_FIRST = 2;

    private String[] cells;
    private Board board;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(getClass().getSimpleName(), "onCreate");

        int first = getIntent().getIntExtra(FIRST_MOVE, RANDOM_FIRST);
        cells = getCells(first);

        board = new Board(this);
        setContentView(board);
        board.requestFocus();
    }

}
```

Add a new string to ***res/values/strings.xml***:

```
   <string name="game_title">Game</string>
```

Remember to also register the new **Game** activity in the manifest. Next, define the **Board** class as follows:

```
package com.bamafolks.android.games.tictactoe;

import android.content.Context;
import android.graphics.Rect;
import android.util.Log;
import android.view.View;

public class Board extends View {
      private final Game game;
      private float width;
      private float height;
      private int selX;
      private int selY;
      private final Rect selRect = new Rect();

      public Board(Context context) {
            super(context);
            this.game = (Game) context;
            setFocusable(true);
            setFocusableInTouchMode(true);
      }

      @Override
      protected void onSizeChanged(int w, int h, int oldw, int oldh) {
            width = w / 3f;
            height = h / 3f;
            getRect(selX, selY, selRect);
            Log.d(getClass().getSimpleName(), "onSizeChanged: width " + width
                        + ",height " + height);
            super.onSizeChanged(w, h, oldw, oldh);
      }

      private void getRect(int x, int y, Rect rect) {
            rect.set((int) (x * width), (int) (y * height),
                        (int) (x * width + width), (int) (y * height + height));
      }

}
```

In order to draw the board, add the following colors to the ***res/values/colors.xml*** file:

```
   <color name="board_background">#ffe6f0ff</color>
   <color name="board_hilite">#ffffffff</color>
   <color name="board_light">#64c6d4ef</color>
   <color name="board_dark">#6456648f</color>
   <color name="board_foreground">#ff000000</color>
   <color name="board_selected">#64ff8000</color>
```

Override the *onDraw* method in the **Board** class and define as follows:

```
    @Override
    protected void onDraw(Canvas canvas) {
        // Draw the background
        Paint background = new Paint();
        background.setColor(getResources().getColor(R.color.board_background));
        canvas.drawRect(0,  0, getWidth(), getHeight(), background);

        // Draw the board
        Paint dark = new Paint();
        dark.setColor(getResources().getColor(R.color.board_dark));

        Paint hilite = new Paint();
        hilite.setColor(getResources().getColor(R.color.board_hilite));

        Paint light = new Paint();
```

```
            light.setColor(getResources().getColor(R.color.board_light));

            for (int i = 0; i < 3; i++) {
                  canvas.drawLine(0, i * height + 1, getWidth(), i * height + 1,
hilite);
                  canvas.drawLine(i * width, 0, i * width, getHeight(), dark);
                  canvas.drawLine(i * width + 1, 0, i * width + 1, getHeight(),
hilite);
            }

            // Draw the X's and O's...
            Paint foreground = new Paint(Paint.ANTI_ALIAS_FLAG);
            foreground.setColor(getResources().getColor(R.color.board_foreground));
            foreground.setStyle(Style.FILL);
            foreground.setTextSize(height * 0.75f);
            foreground.setTextScaleX(width / height);
            foreground.setTextAlign(Paint.Align.CENTER);

            FontMetrics fm = foreground.getFontMetrics();
            float x = width / 2;
            float y = height / 2 - (fm.ascent + fm.descent) / 2;
            for (int i = 0; i < 3; i++) {
                  for (int j = 0; j < 3; j++) {
                        canvas.drawText(this.game.getCellString(i,j), i * width +
x, j * height + y, foreground);
                  }
            }

            // Draw the selected (active) cell
            Log.d(getClass().getSimpleName(), "selRect=" + selRect);
            Paint selected = new Paint();
            selected.setColor(getResources().getColor(R.color.board_selected));
            canvas.drawRect(selRect, selected);
      }
```

Also add the following methods to the **Board** class:

```
      private void select(int x, int y) {
            invalidate(selRect);
            selX = Math.min(Math.max(x, 0), 2);
            selY = Math.min(Math.max(y, 0), 2);
            getRect(selX, selY, selRect);
            invalidate(selRect);
      }

      @Override
      public boolean onTouchEvent(MotionEvent event) {
            if ( event.getAction() != MotionEvent.ACTION_DOWN)
                  return super.onTouchEvent(event);

            select((int) (event.getX() / width), (int) (event.getY() / height));
            setSelectedCell(this.game.getPlayerSymbol());
            Log.d(getClass().getSimpleName(), "onTouchEvent: x " + selX + ", y " +
selY);
            return true;
      }
```

```
    public void setSelectedCell(String symbol) {
        if (game.setCellIfValid(selX, selY, symbol)) {
            invalidate();
        } else {
            Log.d(getClass().getSimpleName(), "setSelectedCell: invalid
selection");
            startAnimation(AnimationUtils.loadAnimation(game, R.anim.shake));
        }
    }
```

Create the new XML resource named ***res/anim/shake.xml*** and define as:

```
<translate
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromXDelta="0"
    android:toXDelta="10"
    android:duration="1000"
    android:interpolator="@anim/cycle_7" />
```

Also create the ***res/anim/cycle_7.xml*** file and define as:

```
<cycleInterpolator
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:cycles="7" />
```

Update the *onCreate* method in the **Game** class to read as follows:

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(getClass().getSimpleName(), "onCreate");

        int first = getIntent().getIntExtra(FIRST_MOVE, RANDOM_FIRST);
        cells = getCells(first);

        board = new Board(this);
        setContentView(board);
        board.requestFocus();

        if (computerSymbol == SYMBOL_O) {
            doComputerMove();
        }
    }
```

Add the following methods to the **Game** class:

```
    private void doComputerMove() {
        // Dumb IA.  Just find the next available unused cell...
        for (int i = 0; i < cells.length; i++) {
            if (cells[i] == SYMBOL_SPACE) {
                cells[i] = computerSymbol;
                break;
            }
        }

        board.invalidate();
        isGameOver();
    }

    private void isGameOver() {
        // Add game win/tie check here...
    }

    public String[] getCells(int first) {
        // TODO: Restore previously stored game...
        String[] grid = new String[9];
        for (int i = 0; i < 9; i++)
            grid[i] = SYMBOL_SPACE;

        switch(first) {
        case PLAYER_FIRST:
            playerSymbol = SYMBOL_X;
            computerSymbol = SYMBOL_O;
            break;
        case COMPUTER_FIRST:
            playerSymbol = SYMBOL_O;
            computerSymbol = SYMBOL_X;
            break;
        case RANDOM_FIRST:
            if (random.nextBoolean()) {
                playerSymbol = SYMBOL_X;
                computerSymbol = SYMBOL_O;
            } else {
                playerSymbol = SYMBOL_O;
                computerSymbol = SYMBOL_X;
            }
        }

        return grid;
    }

    public boolean setCellIfValid(int x, int y, String symbol) {
        int index = (y * 3) + x;
        if (cells[index] != SYMBOL_SPACE)
            return false;
        cells[index] = symbol;
        return true;
    }

    public String getPlayerSymbol() {
        return playerSymbol;
    }

    public String getComputerSymbol() {
        return computerSymbol;
    }

    public String getCellString(int i, int j) {
        int index = (j * 3) + i;
        return cells[index];
    }
```

At this point, you should be able to run the application and try the New Game button out.