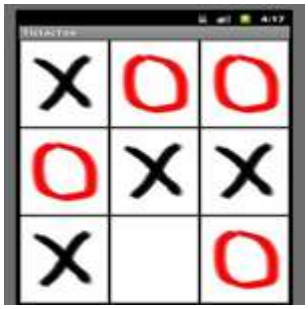
An Android TicTacToe Game



This project describes how to use 2D graphics and draw a TicTacToe board. It also implements the functionality for playing the game and determining a winner!

The Underlying Implimentation:

Basic description of algorithm in step by step form:

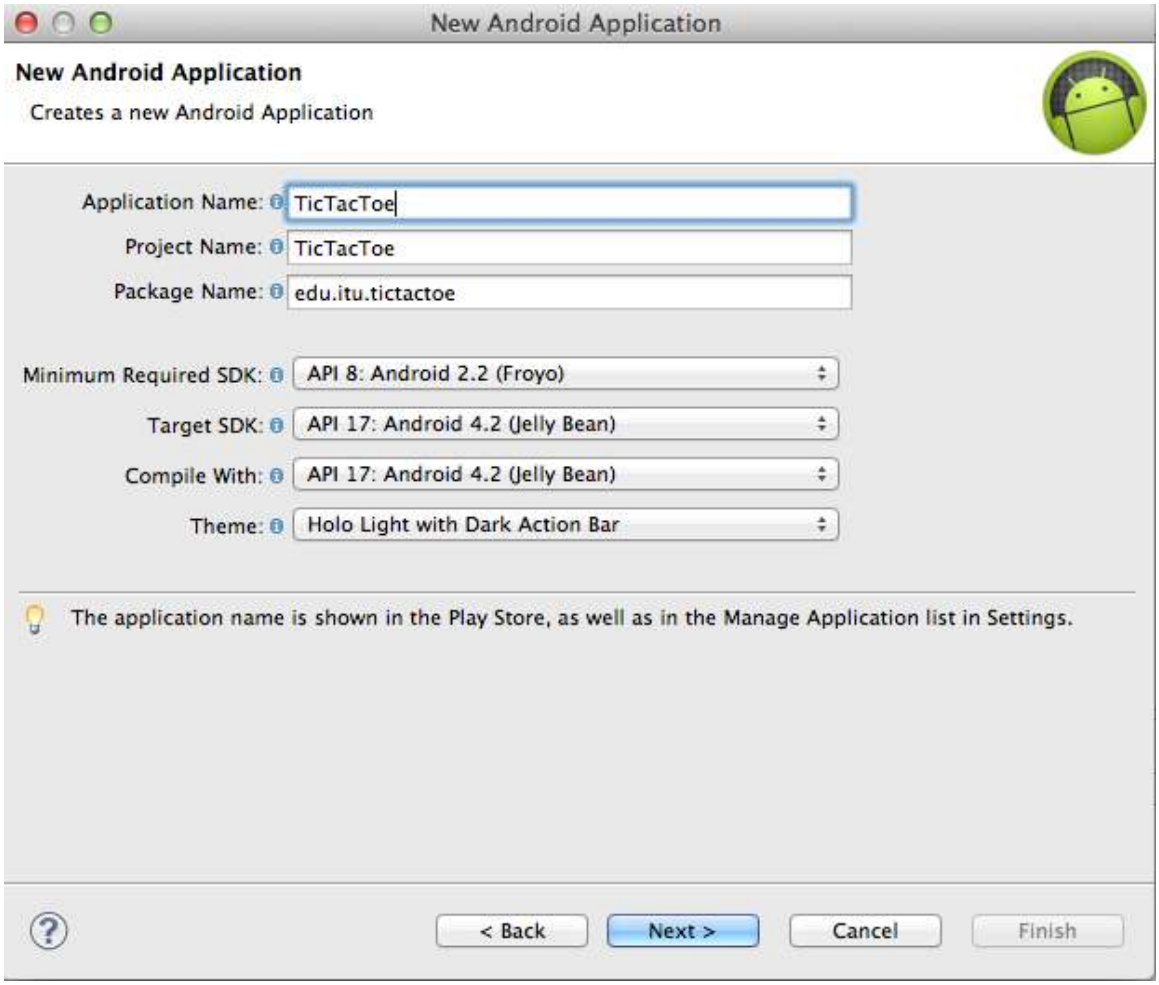
1. Create a Project TicTacToe

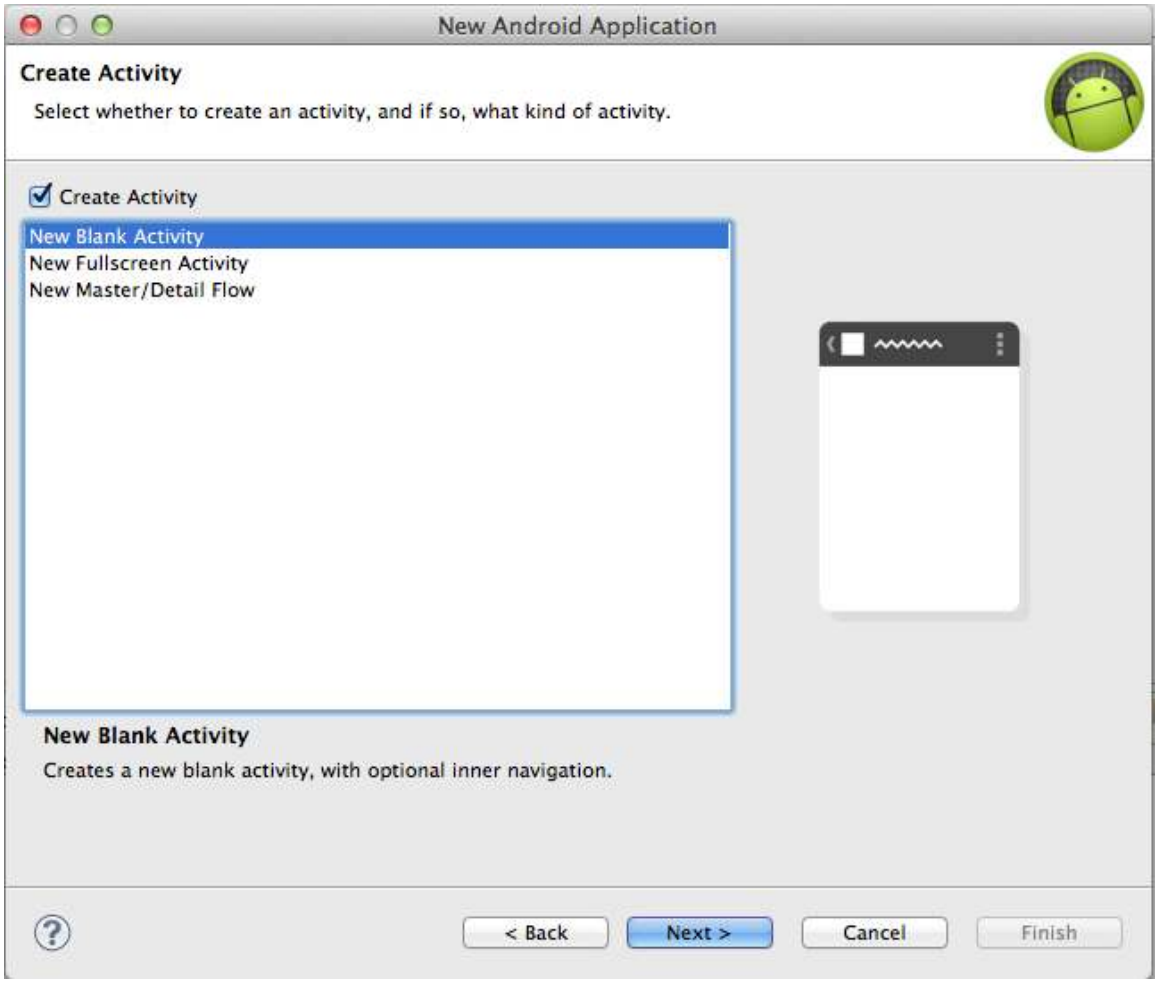
2. Add some images into your drawable folder for drawing X and O images. Program the game logic.

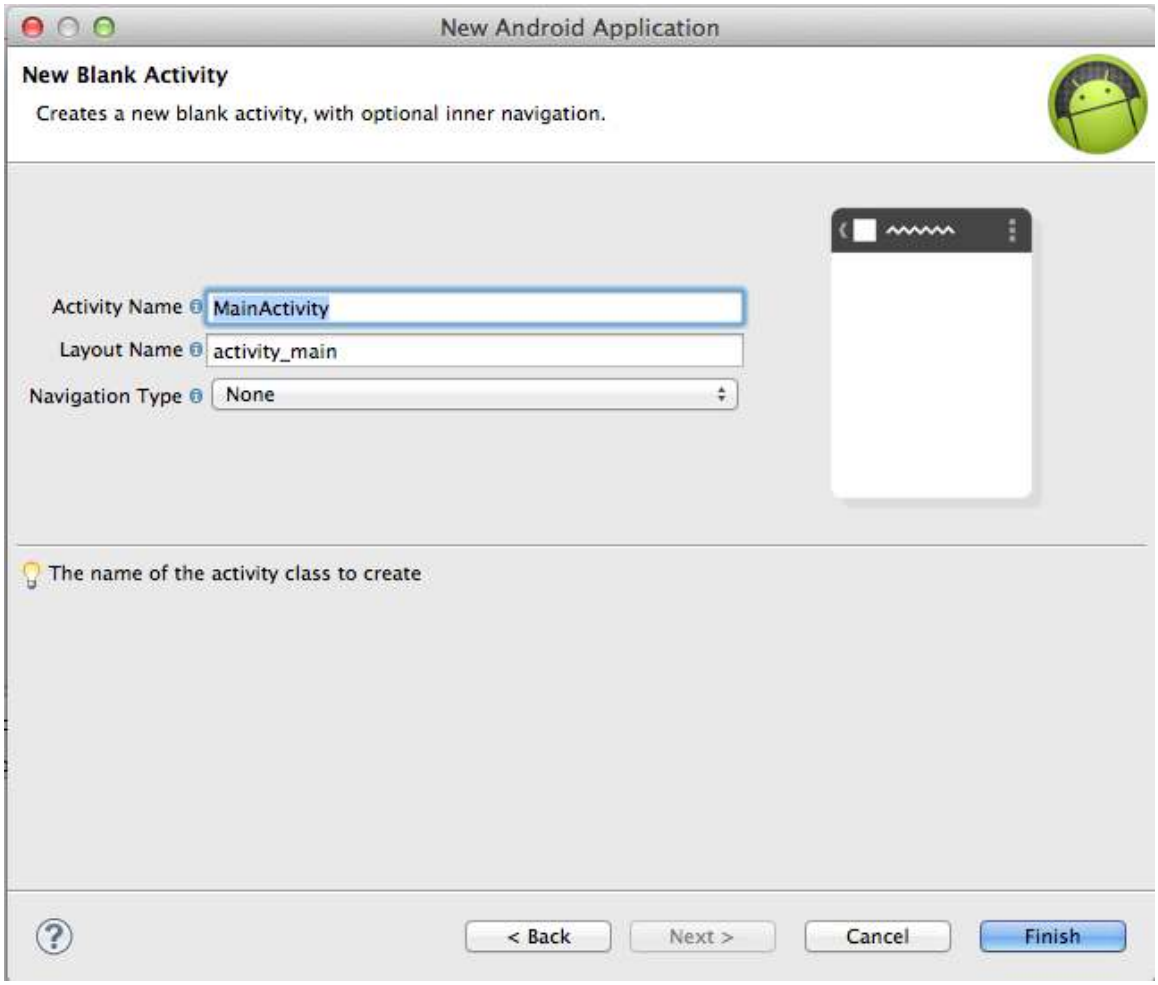
3. Run the application.

Steps to Create:

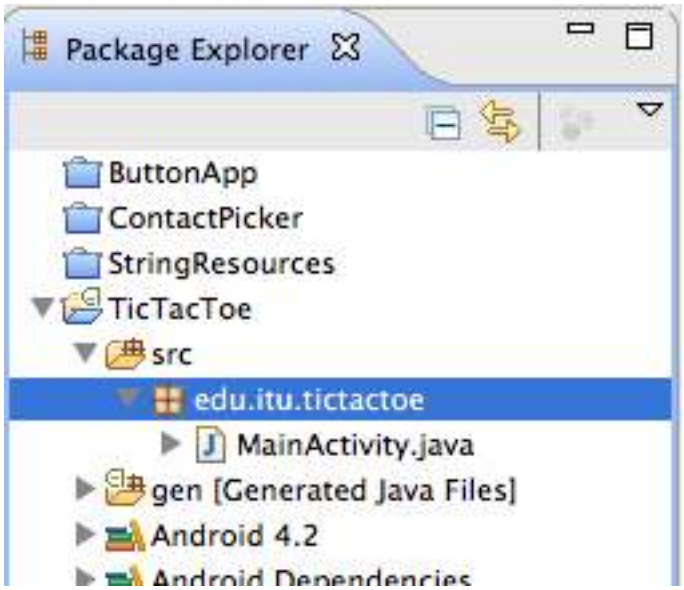
Step 1: Open Eclipse. Use the New Project Wizard and select Android Project Give the new project name like TicTacToe. Enter following information:


New Android Application
New Android Application
Creates a new Android Application
Application Name: TicTacToe
Project Name: TicTacToe
Package Name: edu.itu.tictactoe
Minimum Required SDK: API 8: Android 2.2 (Froyo)
Target SDK: API 17: Android 4.2 (Jelly Bean)
Compile With: API 17: Android 4.2 (Jelly Bean)
Theme: Holo Light with Dark Action Bar
The application name is shown in the Play Store, as well as in the Manage Application list in Settings.
< Back
Next >
Cancel
Finish

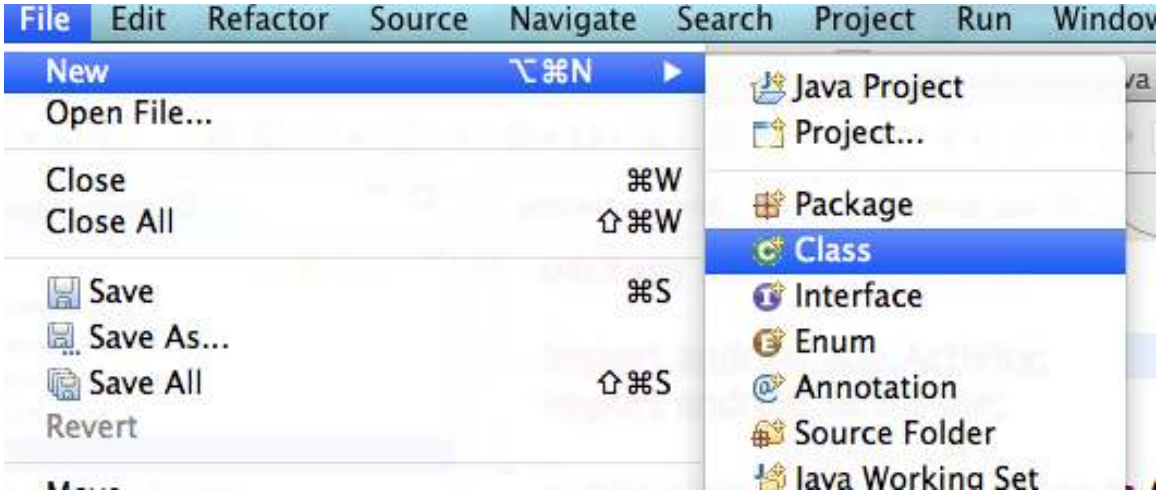

New Android Application
Create Activity
Select whether to create an activity, and if so, what kind of activity.
Create Activity
New Blank Activity
New Fullscreen Activity
New Master/Detail Flow
New Blank Activity
Creates a new blank activity, with optional inner navigation.
< Back
Next >
Cancel
Finish



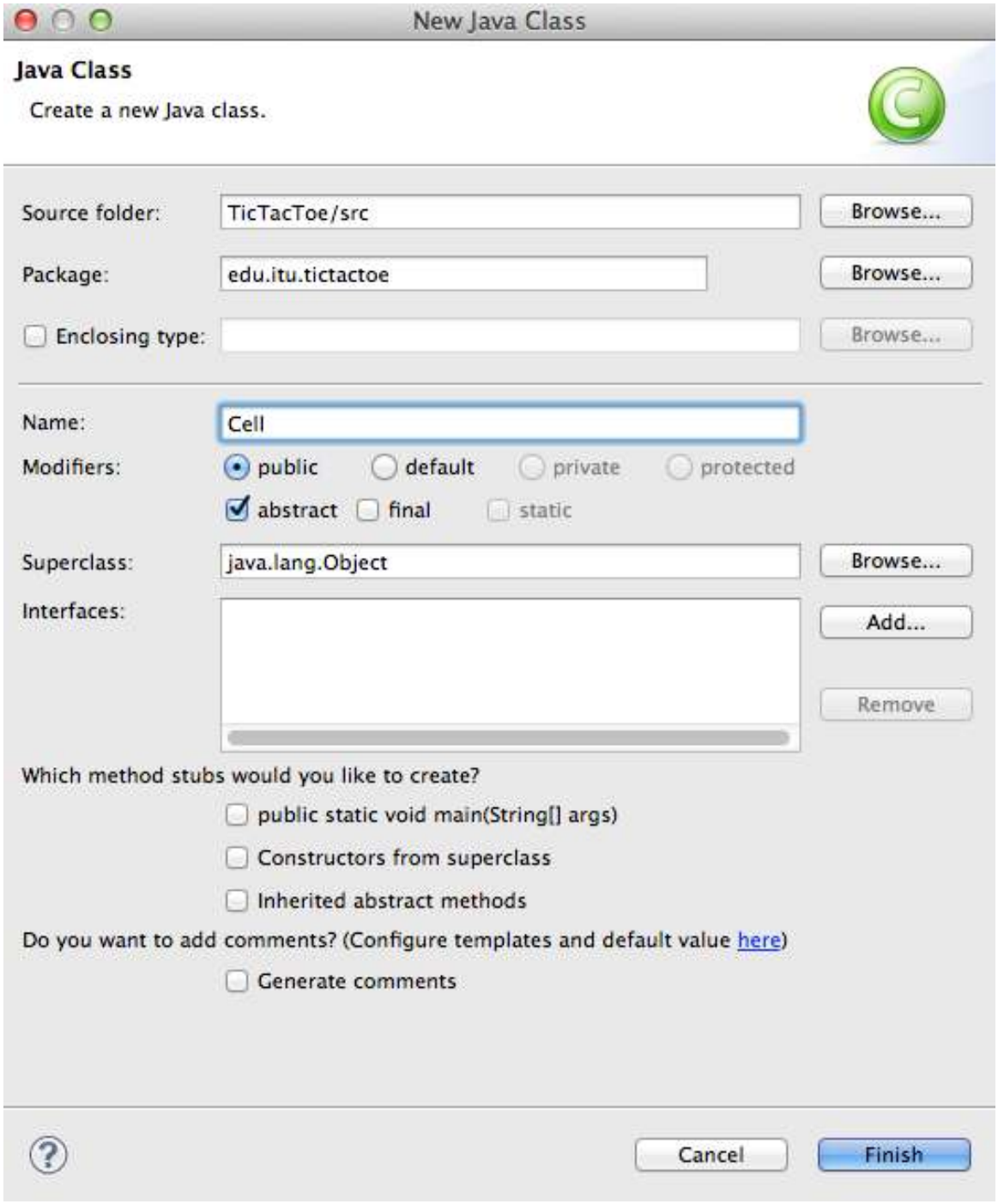
Step 2: Lets add a class to the project. Click on the package in the source "src" folder and we will add a new class to this package.



Select File->New->Class as follows:



We are first going to create an abstract class named, **Cell** using the **abstract** check box as follows:



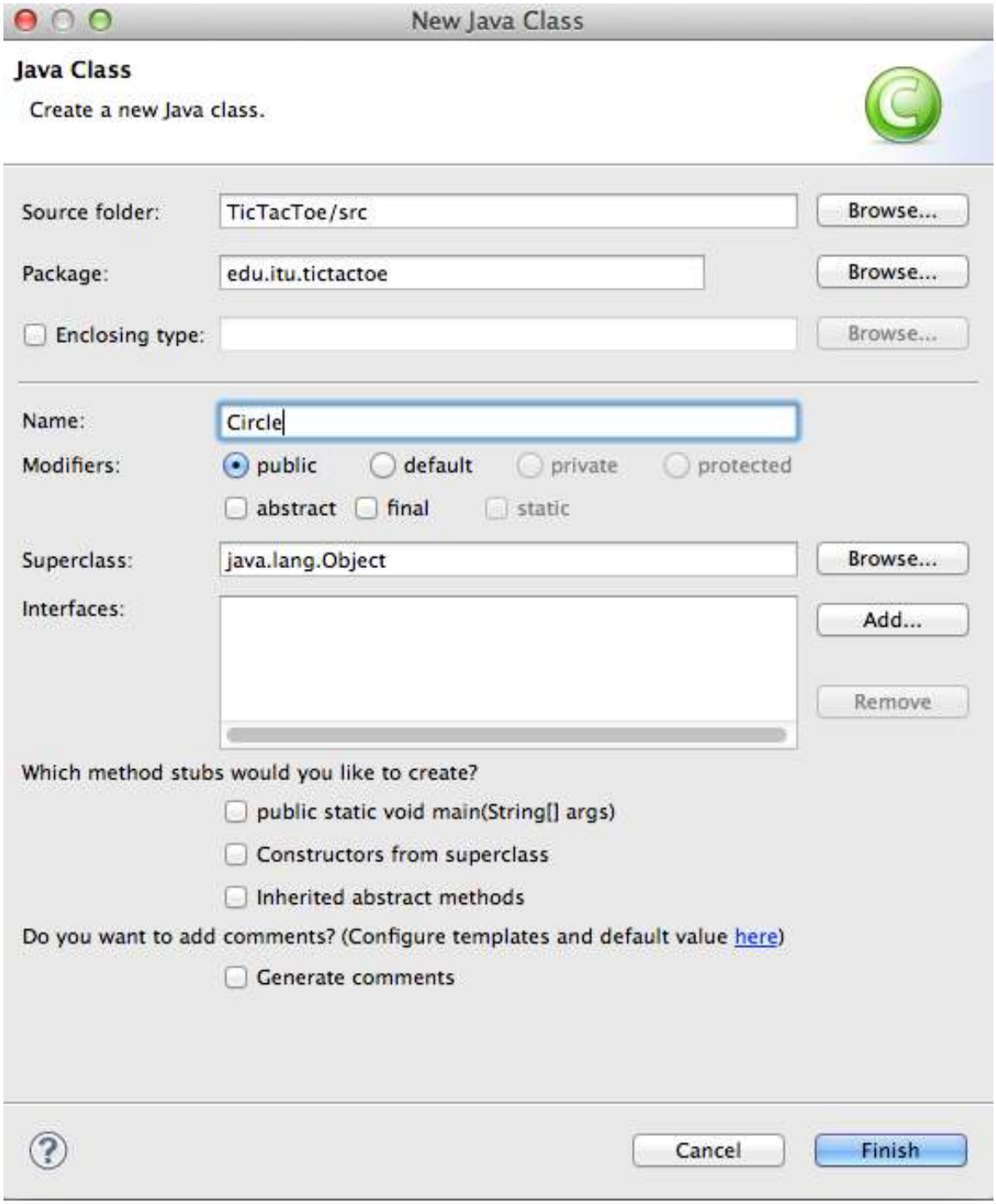
Change the text in this Cell.java to the following pre-written code:

```
package edu.itu.tictactoe;

import android.content.res.Resources;
```

```
import android.graphics.Canvas;

import android.graphics.Point;



public abstract class Cell extends Point {

    public Cell(int x, int y) {

        super(x, y);

    }

    abstract public void draw(Canvas g,Resources res, int x, int y, int w,
int h);

}
```

You should not receive any errors or warnings. Change your package name if its different then what is in this example.

Step 3: Create a new class and name it **Circle**. This class is NOT abstract but it will use the **cell** abstract class in its implementation. Use the following settings:



Change the text in Circle.java to the following pre-written code:

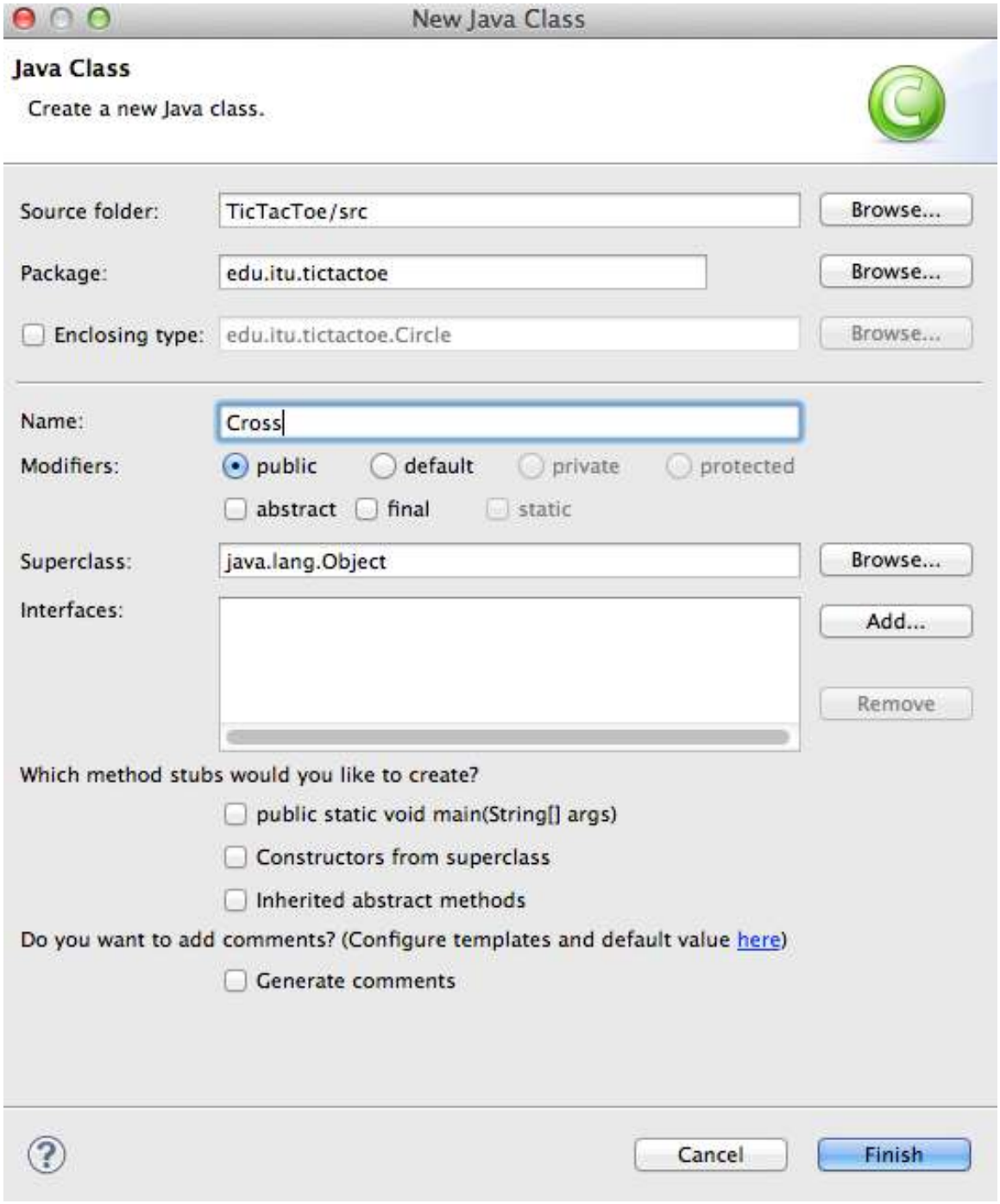
```
package edu.itu.tictactoe;

import android.content.res.Resources;

import android.graphics.Bitmap;
```

```
import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.Paint;

import android.graphics.Rect;


public class Circle extends Cell {

    public Circle(int x, int y) {

        super(x, y);

    }

    public void draw(Canvas g, Resources res, int x, int y, int w, int h) {

    Bitmap im = BitmapFactory.decodeResource(res,
R.drawable.circle);

    g.drawBitmap(im, null, new Rect(x*w, y*h, (x*w)+w, (y*h)+h),
new Paint());

    }

    @Override

    public boolean equals(Object obj) {

        if (obj instanceof Circle) {

            return true;

        } else {

            return false;

        }

    }

    @Override
```

```
    public String toString() {

        return "O";

    }

}
```

Note: You might be an error message if you have not added the circle.png file to the project yet. This is OK, we will add the images last.

Step 4: Create a new class and name it **Cross**. This class is NOT abstract but it will use the **cell** abstract class in its implementation. Use the following settings:



Change the text in Cross.java to the following pre-written code:

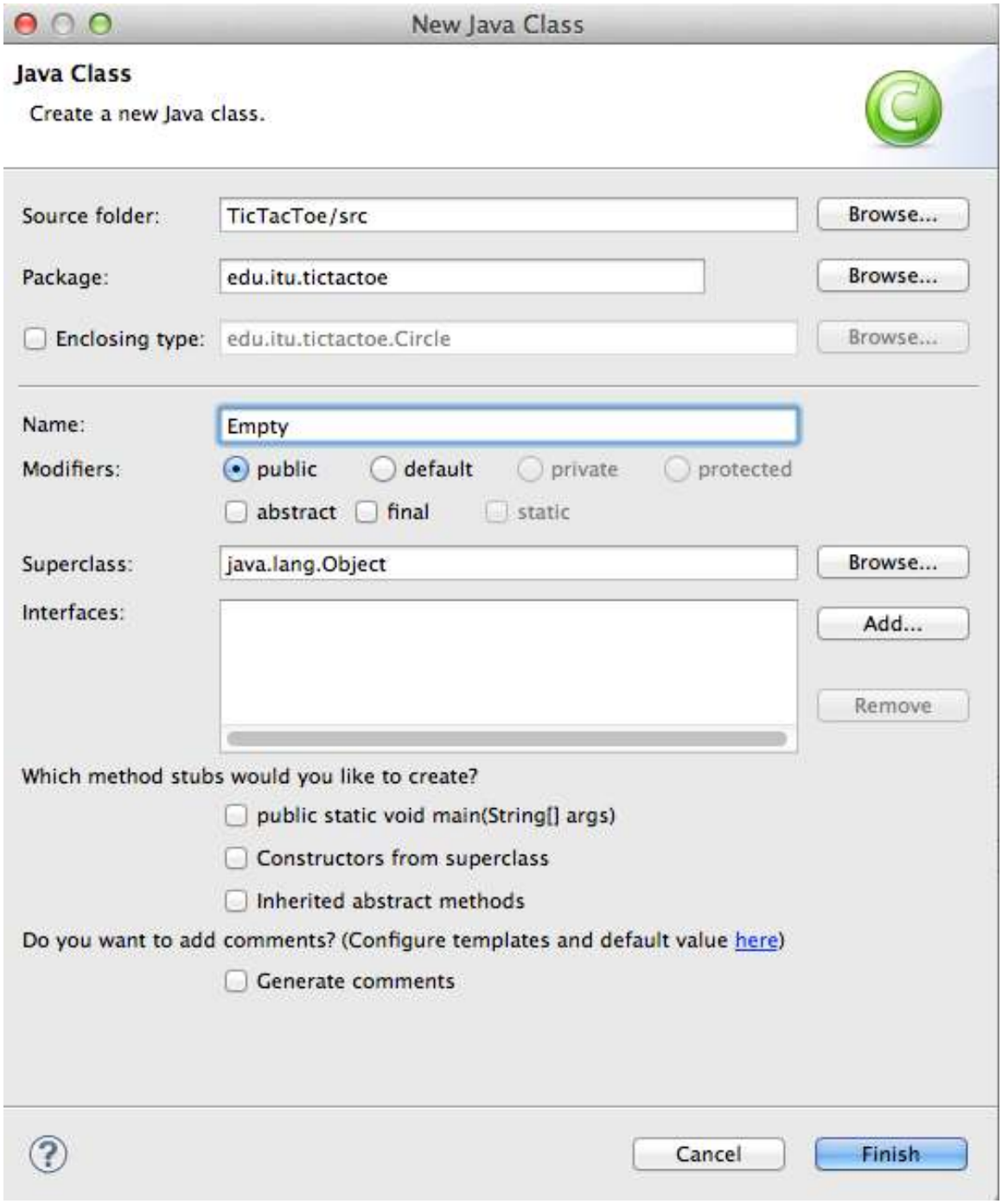
```
package edu.itu.tictactoe;

import android.content.res.Resources;

import android.graphics.Bitmap;
```

```
import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.Paint;

import android.graphics.Rect;


public class Cross extends Cell {

    public Cross(int x, int y) {

        super(x, y);

    }

    public void draw(Canvas g, Resources res, int x, int y, int w, int h) {

        Bitmap im = BitmapFactory.decodeResource(res,
R.drawable.cross);

        g.drawBitmap(im, null, new Rect(x*w, y*h, (x*w)+w, (y*h)+h),
new Paint());

    }

    @Override

    public boolean equals(Object obj) {

        if (obj instanceof Cross) {

            return true;

        } else {

            return false;

        }

    }

    @Override
```

```
    public String toString() {

        return "X";

    }

}
```

Note: You might be an error message if you have not added the cross.png file to the project yet. This is OK, we will add the images last.

Step 5: Add another class and call it **Empty**. This class is NOT abstract but it will use the **cell** abstract class in its implementation. Use the following settings:



Change the text in Empty.java to the following pre-written code:

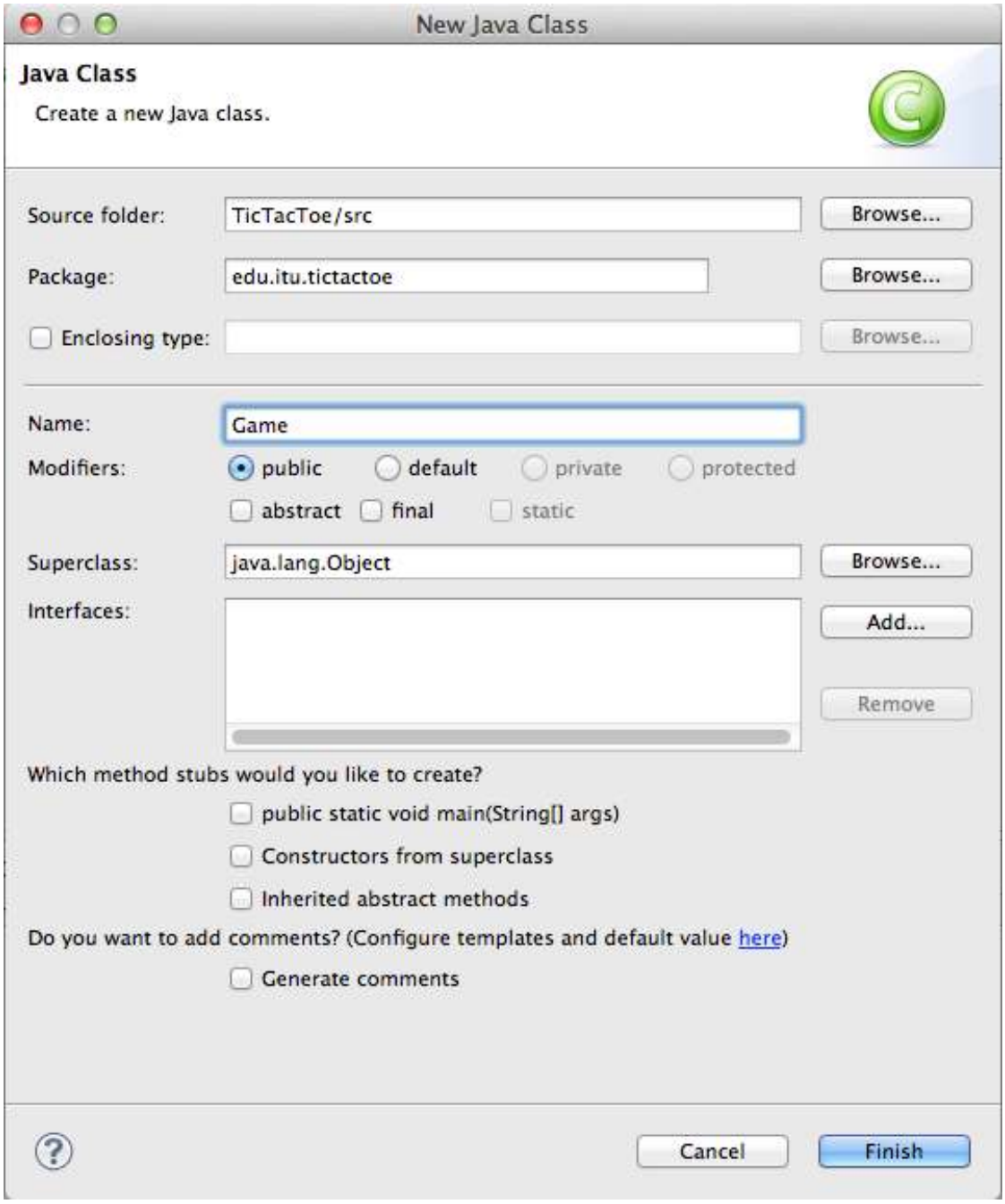
```
package edu.itu.tictactoe;

import android.content.res.Resources;

import android.graphics.Bitmap;
```

```
import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.Paint;

import android.graphics.Rect;


public class Empty extends Cell {

    public Empty(int x, int y) {

        super(x, y);

    }

    public void draw(Canvas g, Resources res, int x, int y, int w, int h) {

    Bitmap im = BitmapFactory.decodeResource(res,
R.drawable.empty);

    g.drawBitmap(im, null, new Rect(x*w, y*h, (x*w)+w, (y*h)+h),
new Paint());

    }

    @Override

    public boolean equals(Object obj) {

        if (obj instanceof Empty) {

            return true;

        } else {

            return false;

        }

    }

    @Override
```

```
    public String toString() {

        return " ";

    }

}
```

Note: You might be an error message if you have not added the empty.png file to the project yet. This is OK, we will add the images last.

Step 6: Add another class and call it **Game**. This class is NOT abstract but it will use the **cell** abstract class in its implementation. Use the following settings:



Change the text in Game.java to the following pre-written code:

```
package edu.itu.tictactoe;

import android.content.Context;

import android.graphics.Canvas;
```

```
import android.graphics.Paint;
import android.graphics.Paint.Style;
import android.os.Handler;
import android.os.Message;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Toast;

public class Game extends View {
    private Cell[][] singlesquare = null;
    int x = 3;
    int y = 3;
    private int l;
    private int a;
    private boolean whatdrawn = false;
    private int playerwin = 3;
    private Paint caneta;

    Handler handler = new Handler() {
        // @Override
        public void handleMessage(Message msg) {
            switch (msg.what) {
            case 0:
```

```
            invalidate();

            break;

         case 1:

            Toast.makeText(getContext(), "You Win!",
Toast.LENGTH_SHORT).show();

            break;

         case 2:

            Toast.makeText(getContext(), "Computer Win!",
Toast.LENGTH_SHORT).show();

            break;

         case 3:

            Toast.makeText(getContext(), "Loose!",
Toast.LENGTH_SHORT).show();

            break;

         default:

            break;

         }



         super.handleMessage(msg);

      }

   };



   public int getGameSize() {

      return x;
```

```
}

public Game(Context context) {
    super(context);

    caneta = new Paint();
    this.caneta.setARGB(255, 0, 0, 0);
    this.caneta.setAntiAlias(true);
    this.caneta.setStyle(Style.STROKE);
    this.caneta.setStrokeWidth(5);

    l = this.getWidth();
    a = this.getHeight();

    singlesquare = new Cell[x][y];

    int xss = l / x;
    int yss = a / y;

    for (int z = 0; z < y; z++) {
        for (int i = 0; i < x; i++) {
            singlesquare[z][i] = new Empty(xss * i, z * yss);
        }
```

```
    }
}

@Override
protected void onDraw(Canvas canvas) {
    for (int i = 0; i < singlesquare.length; i++) {
        for (int j = 0; j < singlesquare[0].length; j++) {
            singlesquare[i][j].draw(canvas, getResources(), j, i, (this
                    .getWidth() + 3)
                    / singlesquare.length, this.getHeight()
                    / singlesquare[0].length);
        }
    }

    int xs = this.getWidth() / x;
    int ys = this.getHeight() / y;
    for (int i = 0; i <= x; i++) {
        canvas.drawLine(xs * i, 0, xs * i, this.getHeight(), caneta);
    }
    for (int i = 0; i <= y; i++) {
        canvas.drawLine(0, ys * i, this.getWidth(), ys * i, caneta);
    }
```

```
    super.onDraw(canvas);
}

@Override
public boolean onTouchEvent(MotionEvent event) {
    int x_aux = (int) (event.getX() / (this.getWidth() / x));
    int y_aux = (int) (event.getY() / (this.getHeight() / y));
    drawimage(x_aux, y_aux);
    return super.onTouchEvent(event);
}

public String getPiece(int player) {
    switch (player) {
    case 1:
        return "x";
    case -1:
        return "o";
    }
    return null;
}

public void drawimage(int x_aux, int y_aux) {
    Cell cel = null;
```

```
        if (whatdrawn)

        {

            cel = new
Cross(singlesquare[x_aux][y_aux].x,singlesquare[x_aux][y_aux].y);

            whatdrawn = false;

        }

        else

        {

            cel = new Circle(singlesquare[x_aux][y_aux].x,
singlesquare[x_aux][y_aux].y);

            whatdrawn = true;

        }



        singlesquare[y_aux][x_aux] = cel;



        handler.sendMessage(Message.obtain(handler, 0));



        if (validate_game()) {

            if (whatdrawn) {

                System.out.println("You Win");

                handler.sendMessage(Message.obtain(handler, 1));

            } else {

                System.out.println("Computer Win");

                handler.sendMessage(Message.obtain(handler, 2));
```

```
        }
        resizegame(x);

    } else if (isFull()) {
        System.out.println("Loose");
        handler.sendMessage(Message.obtain(handler, 3));
        resizegame(x);

    }
}

private boolean validate_game() {
    int contador = 0;
    Cell anterior = null;

    for (int i = 0; i < singlesquare.length; i++) {
        for (int j = 0; j < singlesquare[0].length; j++) {
            System.out.print(singlesquare[i][j]);
            if (!singlesquare[i][j].equals(anterior)
                    || singlesquare[i][j] instanceof Empty) {

                anterior = singlesquare[i][j];
                contador = 0;
```

```
        } else {
            contador++;
        }
        if (contador >= getPlayerwin() - 1) {
            return true;
        }

    }
    System.out.println("");
    anterior = null;
    contador = 0;
}

anterior = null;
for (int j = 0; j < singlesquare[0].length; j++) {
    for (int i = 0; i < singlesquare.length; i++) {
        System.out.print(singlesquare[i][j]);
        if (!singlesquare[i][j].equals(anterior)
                || singlesquare[i][j] instanceof Empty) {
            anterior = singlesquare[i][j];
            contador = 0;
        } else {
            contador++;
```

```
        }

        if (contador >= getPlayerwin() - 1) {
            return true;
        }

    }
    System.out.println("");
    anterior = null;
    contador = 0;
}

anterior = null;
for (int j = singlesquare[0].length - 1; j >= 0; j--) {
    int yau = 0;
    for (int z = j; z < singlesquare[0].length; z++) {
        if (!singlesquare[yau][z].equals(anterior)
                || singlesquare[yau][z] instanceof Empty) {
            anterior = singlesquare[yau][z];
            contador = 0;
        } else {
            contador++;
        }
```

```
            if (contador >= getPlayerwin() - 1) {
                return true;
            }
            yau++;
        }
        contador = 0;
        anterior = null;
    }

    anterior = null;
    for (int j = 0; j < singlesquare[0].length; j++) {
        int yau = 0;
        for (int z = j; z >= 0; z--) {
            if (!singlesquare[yau][z].equals(anterior)
                    || singlesquare[yau][z] instanceof Empty) {
                anterior = singlesquare[yau][z];
                contador = 0;
            } else {
                contador++;
            }

            if (contador >= getPlayerwin() - 1) {
```

```
                return true;
            }
            yau++;
        }
        contador = 0;
        anterior = null;
    }
    return false;
}

public boolean isFull() {
    for (int i = 0; i < singlesquare.length; i++) {
        for (int j = 0; j < singlesquare[0].length; j++) {
            if (singlesquare[i][j] instanceof Empty) {
                return false;
            }
        }
    }
    return true;
}

public void resizegame(int s) {
    x = s;
```

```
        y = s;


        singlesquare = new Cell[x][y];
        int xss = l / x;
        int yss = a / y;


        for (int z = 0; z < y; z++) {
            for (int i = 0; i < x; i++) {
                singlesquare[z][i] = new Empty(xss * i, z * yss);
            }
        }
        handler.sendMessage(Message.obtain(handler, 0));
    }
    public int getPlayerwin() {
        return playerwin;
    }
}
```

Note: You should not have any errors in this code since all of its sub components are already added to the project.

Step 7: Change the source code in the MainActivity.java class to the following pre-written code:

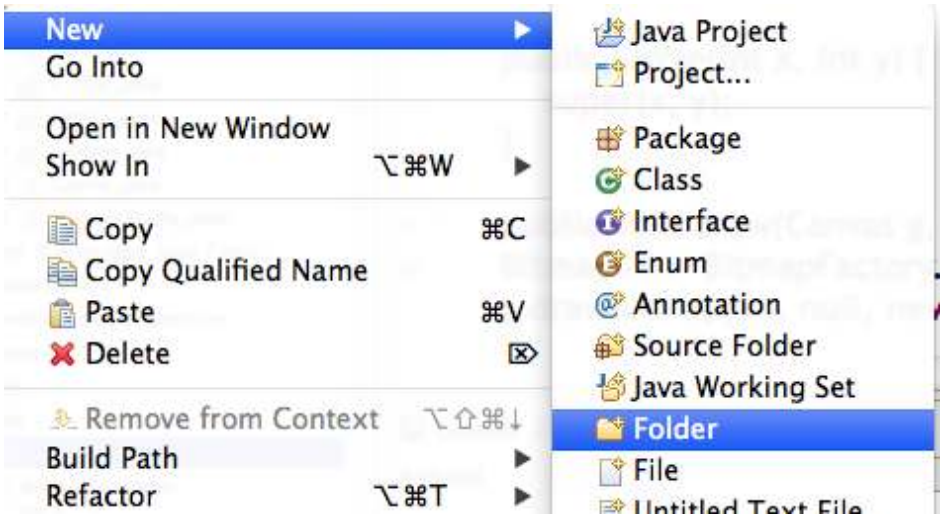
```
package edu.itu.tictactoe;

import android.app.Activity;

import android.os.Bundle;
```

```
public class MainActivity extends Activity {

    private Game game1;


    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        game1 = new Game(this);

        setContentView(game1);

    }

}
```

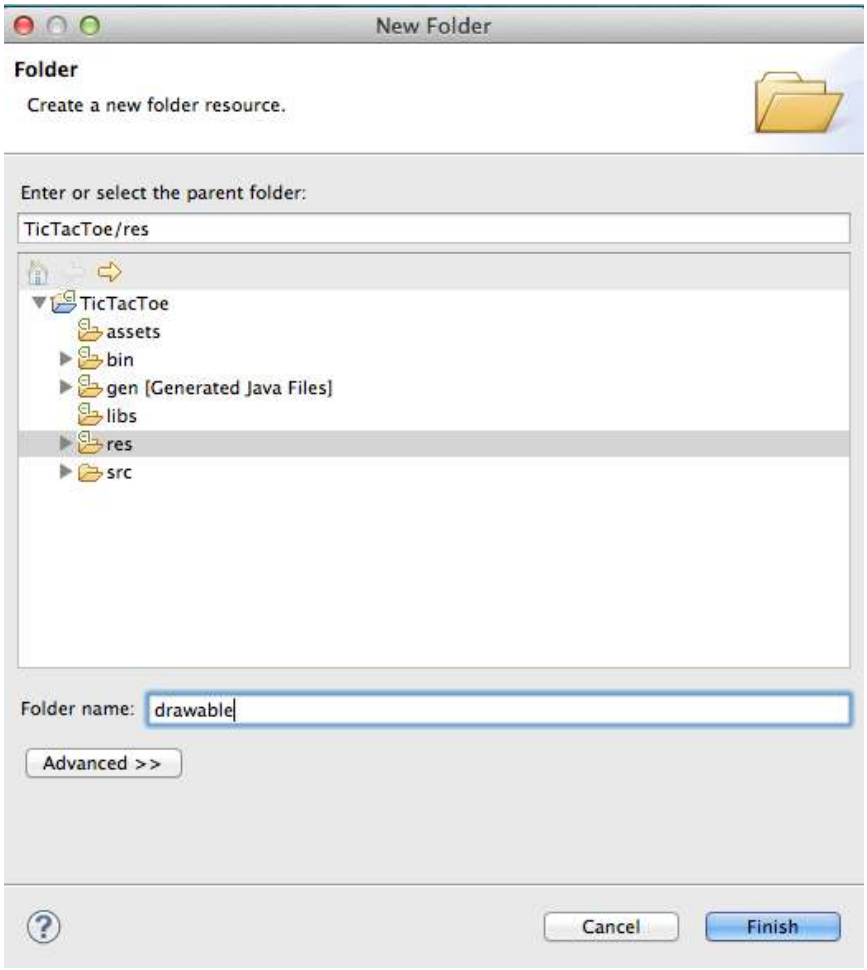
Note: You should not receive any errors since all of this code and sub-components is already implemented in previous steps.

Step 8: Now its time to add the images to the project.

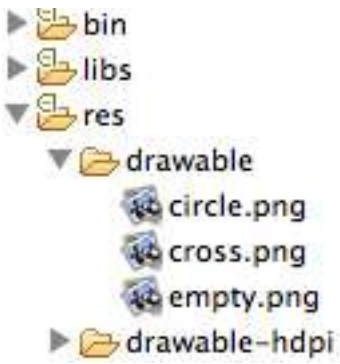
Create the **drawable** folder if it does not exist as follows:



Label the folder **drawable** as follows:



Drag 3 images into this new folder so you have the following:



Step 9: Run the project and play a game of Tic-Tac-Toe!

Congratulations, you are done!