ITIS/ITCS 4180/5180 Mobile Application Development
Homework 3
Date Posted: 06/01/2012 at 01:00am
Due Date: 06/08/2012 at 11:59pm

Basic Instructions:

1. In every file submitted you MUST place the following comments:
    a. Assignment #
    b. File Name.
    c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your Android project as follows:
    a. From eclipse, choose "*Export…*" from the *File* menu.
    b. From the Export window, choose *General* then *File System.* Click *Next*.
    c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
    d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
    e. When exporting make sure you select *Create directory structure for files*.
    f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the `.java` and resource files.
5. Submission details:
    a. When you submit the assignment, compress your exported Android project into a single zip file. The format of compressed file name is HW#.zip
    b. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

Homework 3 (120 Points)

In this assignment you will develop a Tic-Tac-Toe game for Android. You will get familiar with event listeners and how to handle Android Intents. For this project you will have 4 activities, a **Main, Game, ScoreBoard**, and **Credits Activity**. You will design the game for 2-player mode, i.e. 2 human players play against each other, rather than one playing against the computer.

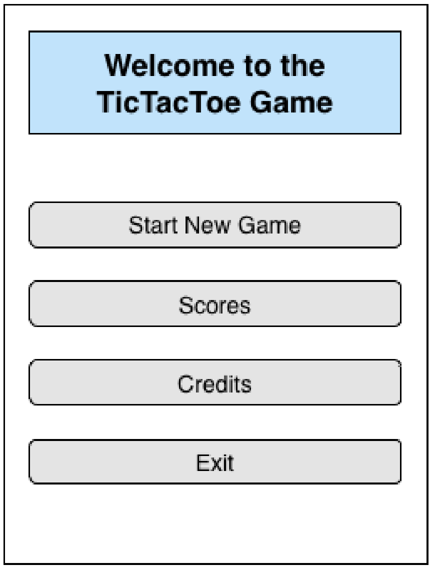
Notes:

1. All strings should be read from your strings.xml, all dimensions from dimens.xml, and all images from drawable-ldpi.
2. The recommended Android Virtual Device (AVD) should have a Target = Android 4.0.3, and a Skin = HVGA.

Part A: Main Activity (20 Points):

The Main Activity displays the app root activity the gets displayed when the application is first activated. The Figure shows a wireframe of the Main Activity interface which includes:

1. The "Start New Game" button should show the Game Activity. To do so you will need to use Android Intents. Use an explicit intent to show the Game Activity.
2. The "Score" button should show the ScoreBoard Activity. Use an explicit intent to show the ScoreBoard Activity.
3. The "Credits" button should show the Credits Activity. Use an implicit intent to show the Credits Activity.
4. The "Exit" button should finish the Main Activity and exit the application.



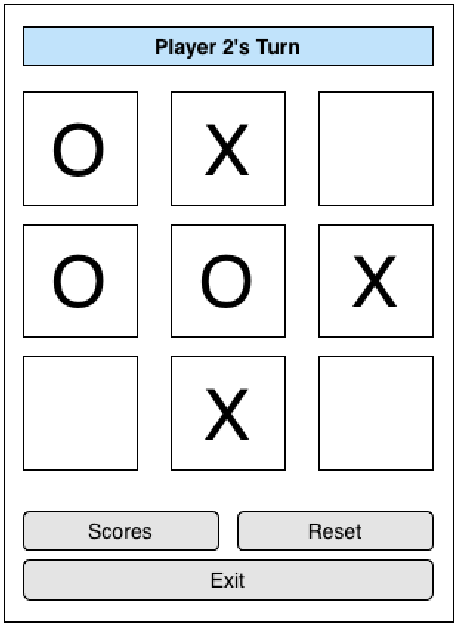
Part B: Game Activity (60 Points):

The Game activity contains the main elements of the Tic-Tac-Toe game. The Figure shows a wireframe of the game's interface which includes:

1. The Player Turn indicator should show the player who currently has the turn to play (Player 1 or Player 2).
2. There are nine slots as indicated in the figure. The X and O figures are provided (use the provided letterx.png and lettero.png) and should be represented as instances of ImageView with appropriate event listeners. At the start of the game, these slots should be set to empty (Use the empty.png from provided support files). When a player clicks a slot, the image is changed to either an X or O based on the player. Assume that Player 1 is X and Player 2 is O. The dimensions of the

ImageViews should be read from the provided dimens.xml file.

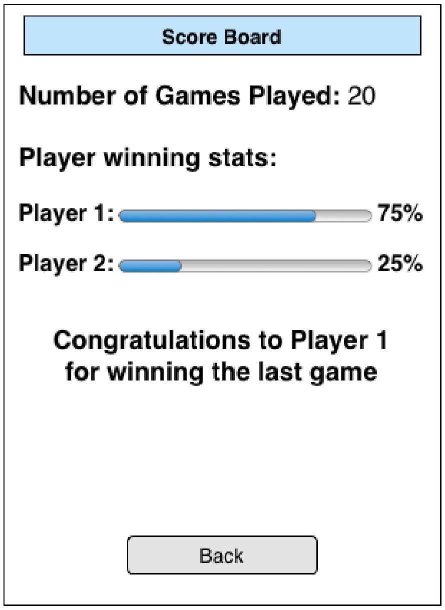
3. The players take turns until one of the players succeeds in occupying 3 slots horizontally, vertically or diagonally. Each time a player makes a move, your code should verify if there is a winner.
4. When a winner is detected or when there is a draw, you should display a message congratulating the winning player via ScoreBoard activity. In case it is a draw, indicate there is no winner.
5. The “Scores” button should show the ScoreBoard activity. Use an explicit intent to show the ScoreBoard activity.
6. The “Reset” button should reset the game to its initial state. This only resets the game UI but should not reset the score statistics.
7. The “Exit” button should quit the Game activity and exits the application.


Player 2's Turn

O	X	
O	O	X
	X	

Part C: ScoreBoard Activity (30 Points):

The ScoreBoard activity shows the overall scores for players 1 and 2. It is called if a user clicks the Scores button, or when a winner is detected, or when draw is detected. A wireframe of the ScoreBoard interface, the activity should indicate the total number of games played so far, the winning stats of each player and a congratulation or draw message. The overall player scores can either be managed by the Game activity. The scores should be passed to the ScoreBoard via the putExtra() method. In addition the Game activity should provide the ScoreBoard activity with the required information to display the correct congratulations message or draw message.


Score Board

Number of Games Played: 20

Player winning stats:

Player 1: **75%**

Player 2: **25%**

Congratulations to Player 1 for winning the last game

1. When a winner is detected, the overall scores need to be updated. You should then show the ScoreBoard activity and send it information indicating who is the winner and the current updated overall scores.
2. If the ScoreBoard activity is started via the “Scores” button in the Main or the Game activity, it simply shows the current overall score stats for Players 1 and 2, with no congratulation message.
3. The “Back” button should finish the ScoreBoard activity.

Part D: Credits Activity (10 Points)

The Credits activity shows the names of the project group members. It also has a “Back” button to return to the previous activity. The Credits activity should be started using an implicit Intent object.

Part E (Bonus): Finding a Fork (BONUS 10 Points)

A fork is when a player has setup two winning moves, so the opponent can only block one of them. To get the bonus points for this question, your program should be able to detect a fork and to declare a winner based on the detected fork then to indicate this in the ScoreBoard activity.