

## Lab 5: Pig Game

### Using Different Layouts for Different Device Orientations

CS235AM, Intermediate Mobile Application Development: Android

#### Overview

This tutorial gives you practice working with:

- Linear UI layouts
- Relative UI layouts
- Loading either a layout with widgets arranged for landscape orientation or one with an arrangement for portrait depending on which way the device is rotated.

#### Part 1: Rotation Tutorial

The tutorial for this lab doesn't guide you through building a project, but source-code for the app described in the tutorial is available. Download the code from GitHub, build it, and run it. Put in break-points so you can see what is happening at various points in the code when you start the app or rotate it. Take at least two screen-shots showing the app with different orientations.

Tutorial:

[https://developer.xamarin.com/guides/android/application\\_fundamentals/handling\\_rotation/](https://developer.xamarin.com/guides/android/application_fundamentals/handling_rotation/)

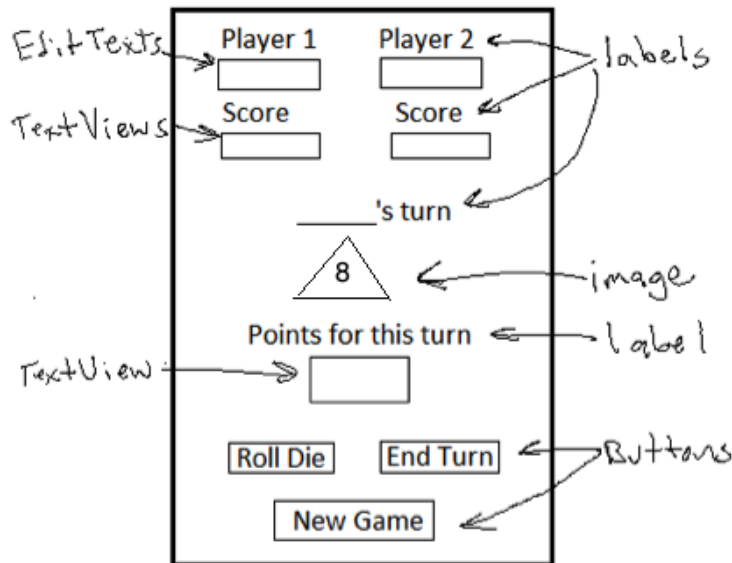
Tutorial source code:

<https://github.com/xamarin/monodroid-samples/tree/master/ApplicationFundamentals/RotationDemo/RotationDemo>

## Part 2, Group A: Big Pig – An 8-sided die, relative layout for portrait, linear for landscape

Create an app for playing a version of the dice game "Pig". Here is an example of one version of the game: <http://nrich.maths.org/1260>

The UI for your game should be similar to the one shown below:



- Use a Frame Layout to hold a background image for the app (you can provide your own image).
- Images will be provided for each side of the die.
- Provide different layouts for landscape and portrait orientation
- Use the names entered in the editable text views for Player 1 and Player 2 to display the name of the player whose turn it is.
- Write a separate Pig class that will contain the logic for the game.

The “Big Pig” version of the game is a dice game with 2 players and one die, but it is an 8-sided die. Each player rolls the die as many times as they wish or until they roll an 8. Their score for the turn is the sum of their rolls unless they get an 8, then it’s zero. When they roll an 8, the “Roll Die” button should be disabled, re-enable it when the next player starts their turn. Players keep taking turns rolling until someone gets to 100. If the player who had the first roll reaches 100, the second player may still take their turn. The player with the highest score wins.

- Portrait orientation: Arrange the widgets using a Relative Layout.
- Landscape orientation: Arrange the widgets using nested Linear Layouts. (Try to use as few levels of nesting as possible).

## Submission

### *Beta Version*

Post the following to the Beta + Code Review Forum:

- 1) For part 1: A document containing screen-shots of the tutorial app with each screen-shot labeled. (Please use .docx or .pdf format.)
- 2) For part 2: A zip file containing your app's Visual Studio solution folder. (Make your solution smaller by deleting the *obj* and *bin* folders.)  
Or, optionally, a link to a repository containing your solution source code. (You can put the link on the same document with the screen-shots for part 1.)
- 3) A copy of your lab instructions (so the lab partner who reviews your work will know what the requirements were for your app).

### *Production Version*

1. Items 1 and 2 above, but revised as needed.
2. The code review of your work (the one done by your lab partner) with the second column ("Release") completed by you.