CS 4985 – Fall 2013
Assignment 2, 20 pts.
Due: Tuesday, Sept. 17th at 9a.

Objectives

- Be able to display and use simple UI widgets according.
- Add functionality to respond to interaction with the various UI widgets.

Specifications

In this assignment you will make a simple programmer's calculator.

A. Getting Started

1. Create a new Android project as follows:
    a. Project name to be: YourNameAssignment2
    b. Package name to be: edu.uwg.*firstnamelastname*.calculator
    c. app_name: WolfBytes Calculator by *Firstname Lastname*
2. Do NOT downloand and use another Android calculator project written by someone else as a starting point for this project.

B. Functionality

1. The calculator must have the following buttons for interaction:
    a. Numbers 0 to 9.
    b. Six buttons with the letters A-F on them for hexadecimal input.
    c. Clear to clear the operation and display.
    d. Input/output display.
    e. Buttons to support the following operations:
        i. Addition, subtraction, multiplication, division, modulus, equal, shift left, shift right, not, and, or, xor, sign (to change the sign of the displayed number).

f. Three radio buttons that allow the user to specify the number base that will be used. The three bases to use are: binary, decimal, and hexadecimal.

    i. Only the legal characters for the active base should be enabled. For example, if binary is selected then the A-F and 2-9 buttons should be grayed out.

2. Implement the calculator functionality, so the calculator will work as follows:

    a. All arithmetic will be integer arithmetic. The arithmetic should work in all three bases.

    b. To simplify the calculator, the arithmetic operations can be evaluated in the order they are entered. In other words, there is no operator precedence.

    c. The answer to the operation should be displayed when the equal button is invoked or the next operator is invoked.

    d. The math operator that is invoked should be displayed in the input/output display.

    e. The result from a previous operation should become the first operand of any subsequent operation.

    f. The clear button should clear out the answer display and any pending operations.

    g. When switching bases, if a number is in the answer display that number should be converted to the new base.

        i. All operations should work in any base.

C. UI requirements

1. Make a clean and professional looking calculator that displays nicely in both portrait and landscape mode for both phones and tablets.

2. The calculator must have a picture of a wolf behind it. This picture must display and scale correctly for both portrait and landscape mode for both phones and tablets.

D. Implementation requirements

1. Follow clean coding practices.

2. Add a status.txt file to your Eclipse project and in this file explain what works, any known bugs, and any issues that you ran into.

Suggestions

- Implement all the calculator functionality using base 10 first, before handling the other bases.
- For handling the other bases, the only thing that should need to be modified is the display of the numbers. You should be able to keep the math functionality in your code as base 10.
- You do not need a separate event handler for each control. Similar types of controls can be handled in the same method.
- The Integer::parseInt method is overload to convert a string to a different base number.
- The Integer::toHexString and Integer::toBinaryString methods return the base equivalent string.
- You can read about bitwise operators here: http://en.wikipedia.org/wiki/Bitwise_operation

Grading breakdown

- 4 pts. – The UI design in both landscape and portrait mode on both phones and a tablet.
- 3 pts. – Enabling and disabling of controls according to the number base specified.
- 10 pts. – Calculator can convert between the three bases and perform all the operations in each base.
- 3 pts. – Clean coding practices used; status.txt file completed; correct filename for submission.

Submission

Export your project in Eclipse, naming the archive file *FirstnameLastName*Assignment2.zip, e.g., JohnDoeAssignment2.zip and submit the file in Moodle by the due date.