CS 501, Mobile Application Development

Worksheet 4

Gestures, Teamwork, Fundamental Programming

Date: _____

Team Members: _____

This worksheet is to be done in collaboration with your project team. Although we are working with Android, the concepts apply for any device.

Part 0: Getting to know your Team + Fundamental Programming – Design Process (basic)

Utilize your skills as a competent programmer to implement a Flashlight App. Use the information/requirements specified below in 1b.

1. **What design questions do you need to ask before beginning?**
2. **What are the functions that need to be written?**
3. **What exceptions do you need to handle and how should you handle them.**
4. **What types of feedback do you need to provide the user as they interact with your App.**
5. **Be ready to briefly present to the class.**

Part 0b: Fundamental Programming – Implementation

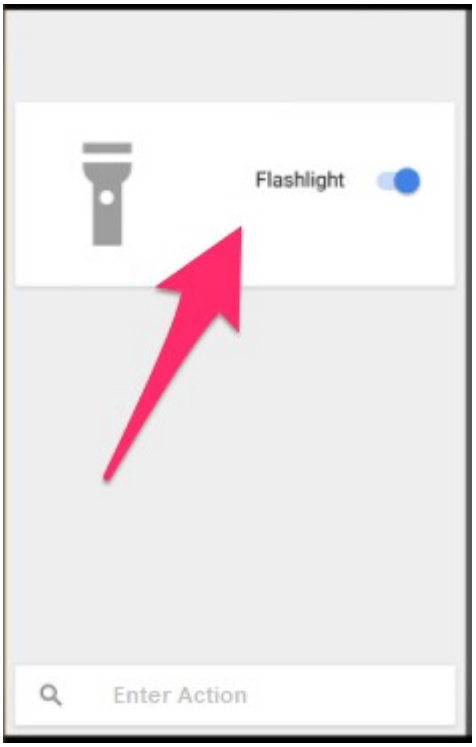
Turning on your flashlight 3 ways.

Implement an android app that turns on your flashlight.

1. Use any layout you like, but make it look similar to the diagram below.
2. Use a simple toggle switch to turn the flashlight on/off.
3. Make it so entering text into the Action Text Box will also affect the flashlight. 'ON' turns on the flashlight, 'OFF' turns off the flashlight.
    a. Ensure that entering 'ON' or 'OFF' appropriately flips the toggle switch, if needed.
4. Modify your code to dynamically render the toggle button at startup.
5. Use a fling, fling up, turns on the flashlight, fling down, turns off the flashlight. Ensure the fling is "significant" and not an accident as a result of the user lifting their finger too quickly off of the screen.
    a. Again ensure that the toggle switch, toggles to the appropriate setting.
6. Ensure you handle exceptions appropriately, for example flash light not available, etc.

Additional Notes:

> *You only need to implement this for portrait mode. You do not need to implement speech recognition. It also does not matter if the flashlight turns off when the app is closed. Simply follow the steps above.*



Depending on the model of your device, this app may require some permissions.

```
<uses-permission android:name="android.permission.FLASHLIGHT" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<uses-feature android:name="android.hardware.camera.flash" />
```

Add these lines, just below the package statement in your manifest file. For example:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sse.lectx_flashlight">
    <uses-permission android:name="android.permission.FLASHLIGHT" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />
    <uses-feature android:name="android.hardware.camera.flash" />
```

Part 1: Initial Project Ideas:

- **Meet with your team for 30 minutes**
- **Based on what you know how to do now, and what you would like to learn, come up with 3 ideas for projects. Be as descriptive as possible.**

- **Describe third party libraries you would like to work with (e.g., - facebook, Instagram, ebay, paypal, etc.).**
- **Identify what the pinch points might be, and what you would need to learn to implement your project ideas.**
- **Be ready to briefly present your ideas to the class.**

Part 2: Flinging Money.

1. **Currency converter. Write an app that uses scrolls and flings to calculate currency values for the US Dollar vs. 4 other currencies of your choosing. The Simple App will have one primary editable text box that holds currency in US Dollars, and 4 other text boxes (read only) that will be used to display the converted currency value. Each box will be for a different currency, eg, Dollars, Yen, Florin, Rupi's, etc.**

    A. **As the value in the edit text changes, the other 4 textboxes must get updated, regardless of whether a gesture was used to make the change or, if a user entered the value directly.**
    B. **A scroll up or down, anywhere in the app, would produce 10 cent increments/decrements**
    C. **A fling up or down, would make 1 dollar increments/decrements.**
    D. **Ensure the value of the US Dollars EditText never goes below $0. If a user enters a negative value, it should snap back to $0.**

Part 3: Accelerometer Measurements.

1. **an app that measures movement along the x,y,and z planes. When the device moves "significantly" in any one direction report this in a log and via a Toast. You don't want to report every small movement, just large ones. You can define what is considered *significant* via experimentation.**

2. **Write an app that measures movement along the x,y,and z planes. When the device moves "significantly" in any one direction report this in a log and via a Toast. You essentially don't want to report every small movement, just large ones. You can define what is considered *significant* via experimentation.**

3. **Add a SeekBar to the App to allow users to *tweak* what is considered a significant move.**

4. **Add a WebView Component to the screen, and have it load the following urls based upon the size of the biggest move. If the biggest move is in the:**
    a. **x direction, you will load** https://www.ecosia.org/

b. **y direction, you will load** https://www.dogpile.com/

c. **z direction, you will load** https://buzzsumo.com/

Notes: (1) It might be best to test the WebView first, ensure you can load web pages with a simple button click. You may need to enable Javascript and/or provide permissions for internet access. Refer to the Android Documentation as needed. (2)Only perform these actions if the change in direction was "significant". (3) Feel free to lock the screen in portrait or landscape to avoid intermediate destruction of Views. Use

```
  setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
or
  setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

5. **Make it so that if you shake your device <u>really, really fast</u>, it displays the following URL:** https://jumpingjaxfitness.files.wordpress.com/2010/07/dizziness.jpg

Part 4: Pulling it all together.

- **Create an App that Has 5 Activities, Name them Home, North, East, South, West.**
- **Put some sort of text or image in each of the Activities. Feel free to use whatever you like, just make clear which activity the user is on.**
- **The Launcher Activity should be the Home. Use flings to open the other Activities.**
    - **Right Fling: Opens East Activity**
    - **Left Fling: Opens West Activity**
    - **Upwards Fling: Opens North Activity**
    - **Downwards Fling: Opens South Activity.**
    - **Shaking the device feverishly should cause the image to also shake for at least 2 seconds (OPTIONAL)**

Part 5: Hangman Game with Advanced Hints.

1. **Create a Hangman Game. This is in preparation for our Lesson on Fragments next week. Target both a Portrait and Landscape views as shown below. This is a design challenge as well, so please make design choices on your own, (eg.,- how to render the hangman stick figure, how to tell the user they won or lost, etc.) Your game should be able to run at least one complete game and handle both winning and losing, no scoring is necessary. *Include a "New Game" Button which will reset the current game and start a new and <u>different</u> game when clicked.***
2. **Your game must remember state when rotating!**

Panel 1: Contains a "Choose the letter" set of buttons. Provide a menu of letters, just use buttons. Be sure to disable the button after it's been selected.

Panel 2: Contains a Hint Button.

Clicking the Hint Button can do two things.

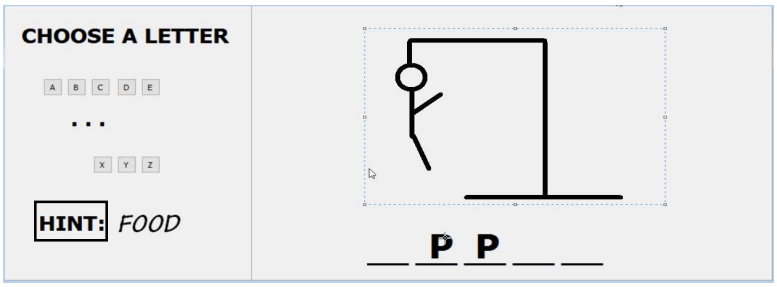
a. The first time it is clicked it displays a hint message BUT it costs the user a turn.

b. The second time it is clicked it disables half of the remaining letters (that are not part of the word) BUT it costs the user a turn.

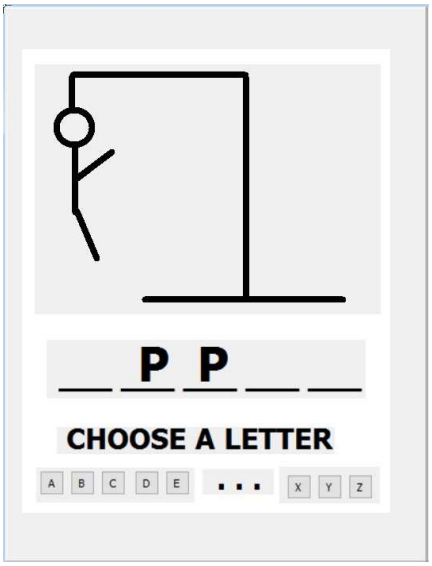
c. After the second click it should be disabled.

Panel 3: Contains the Main Game Play Screen.

For a Tablet in Landscape mode the display should appear as below.



For a phone or a Tablet in Portrait Mode, the interface should be simpler, laid out vertically, and there is no hint button.



General Game Play:

After choosing a letter, the user will be notified whether the letter was correct or not. If correct the letter will be displayed in the proper area, if incorrect, another body part will hang. The game ends when the user successfully identifies the word or is completely hung. **IMPLEMENT <u>ONLY</u> THE REQUIREMENTS ABOVE.**