

ITIS/ITCS 4180/5180 Mobile Application Development

Homework 3

Date Posted: 09/16/2013 at 06:00am

Due Date: 09/23/2013 at 11:55pm

Basic Instructions:

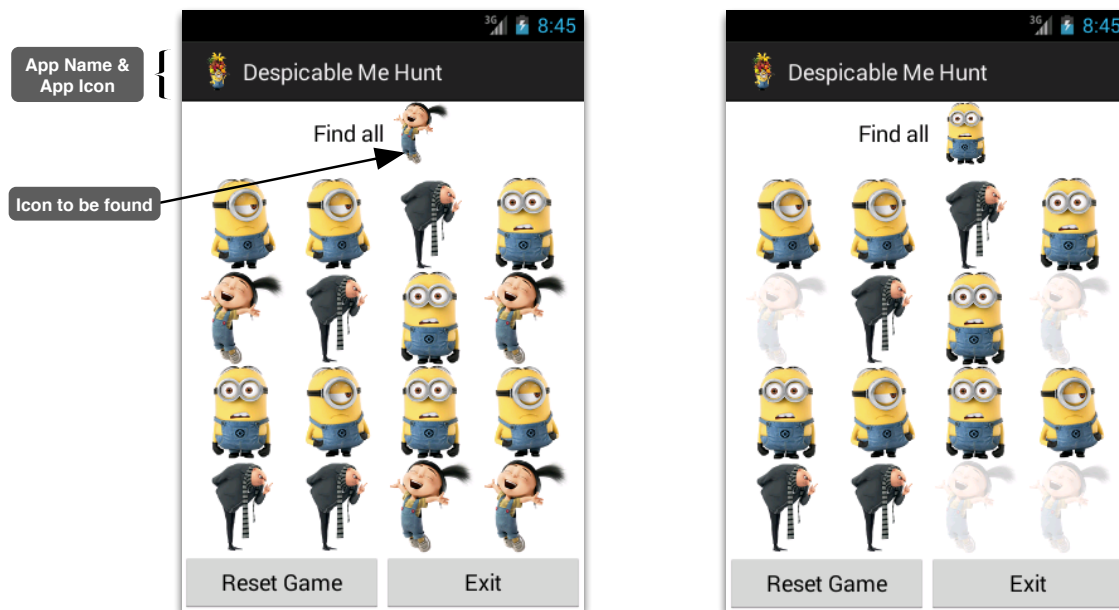
1. In every file submitted you **MUST** place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. 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.
6. 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.
- 7. Failure to follow the above instructions will result in point deductions.**

Homework 3 (100 Points)

In this assignment you will develop a simple selection game for Android called the “Despicable Me Hunt” game. The game is a puzzle played on a 4-by-4 grid with 16 square blocks labeled with characters from the Despicable Me movie. The game indicates to the player a specific character, and the player’s goal is to find all the blocks that display the indicated character. When the player finds all the blocks for the specified character, another character is selected and the game ends when the player finds all the specified characters. This is a single player game. You will get familiar with event listeners, Handling GUI Objects, and how to handle Android Intents. For this project you will have 2 activities, a **Game** activity and a **Result** activity.

Notes:

1. The recommended Android Virtual Device (AVD) should have minimum SDK version set to 16 and target SDK at least 18. The app should display correctly on 3.2” QVGA (ADP2) (320x480: mdpi).
2. All strings should be read from your strings.xml, all dimensions from dimens.xml, and all images from drawable-ldpi. You will be provided with a folder containing all images needed for the game.



(a) Start Screen

(b) Second Character to be found is selected randomly.

Figure 1, Application User Interface

Part 1: Game Activity (80 Points)

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, strings.xml, and drawable files to create the user interface. The layout XML file can be modified through the raw xml, or through the GUI tools

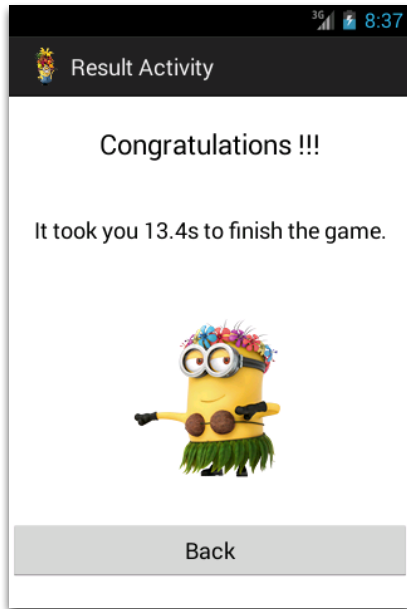
provided within eclipse. To build the UI, please follow the following tasks:

1. Your application should have an application launcher icon, please select your launcher icon to represent your app. Set the application title to “Despicable Me Hunt”.
2. The string values used for the text labels, and button labels should be read from the strings.xml file and should not be hardwired in the layout file. Use the provided character images.
3. **Game board initialization:**
 - a. You are provided with 4 icons, each icon should be displayed in 4 randomly selected locations on the game board.
 - b. *Selecting the focus image:* Randomly select the the character to be found, see Figure 1(a). The selected character is referred to as the focus character. The player should hunt all characters that match the focus character. Hunting a character should change the transparency of the ImageView as shown in Figure 1(b). After the user hunts all the characters that match the focus character, randomly select the next focus character and update the interface as indicated in Figure 1(b).
 - c. The selected focus character should be indicated on the top of the game layout, see Figure 1(a).
 - d. *Deciding on the order of images:* the selected 16 images should be randomly placed on the 4x4 game board. Each of the provided 4 character icons should be displayed in 4 randomly selected locations on the game board.
4. When the player clicks on an image that matches the focus image, it's alpha value should be reduced to indicate it is selected. See Figure 1(a).
5. Tapping on an image that does not match the focus image, or an image that was previously selected should have no effect on the game or on the images.
6. When the player selects all the images that match the indicated focus image, the game should randomly select the next focus image and the interface should be updated as indicated in Figure 1(b). If all the images are selected, the total amount of time taken to select all the images should be recorded, the Results Activity should be started and should be sent the recorded time.
7. When the “Reset” button is clicked a new game board should be created by reinitializing the game.
8. The application should be terminated if the “Exit” button is clicked.

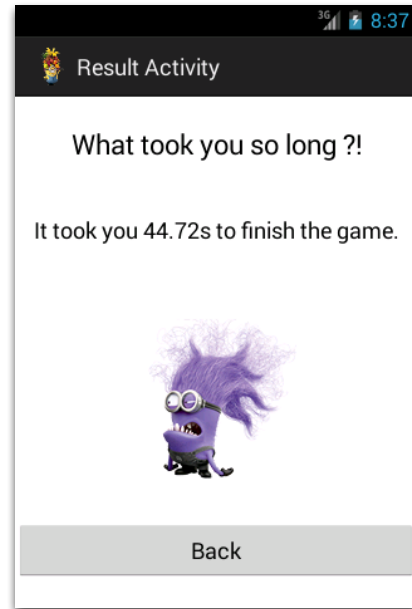
Part 2: Result Activity (20 Points)

The **Result** activity appears when the user successfully completes hunting all characters in the correct order. Game activity should pass the time in seconds that took the user to complete the game. Below are the main implementation requirements:

1. The top message TextView and the ImageView content will differ according to the time passed by the Game Activity. If the time take by the player to complete the game is less than 20 seconds then generate a UI similar to Figure 2(a), otherwise generate the UI based on Figure 2(b).
2. The second message TextView should display the time the player needed to complete the game.
3. Tapping on the “Back” button should finish the Result Activity and start a **new** game.



(a) Congratulations screen



(b) Took too long screen.

Figure 2, The Results Activity