Simulation of Insects Classification Based on Android Computing Platform

Baydaa F. Dhahir ¹, Alaa Hussein ²

¹Computer Department, Thi-Qar University, Iraq
²Biology Department, Thi-Qar University, Iraq

¹Baydaafadhil@gmail.com, ²Thiqarbutterfly@gmail.com

Abstract—Handheld systems, such as smartphones and tablets are now the most common way for people to access and interact with computing services. The demand for application development skills is therefore growing at a breath-taking pace. Since, we live in the “age of insects”. Humans have walked on earth for only a mere fraction of the 350 million years that insects have crawled, burrowed, jumped, bored, or flown on the planet, and classification of insects can be mainly realized as the most important and confusing subject for students of the biology department. So, to bring these skills alive and learn the details of the largest group of animals on Earth (with over 1.5 million species known to science up to now) the article will involve in-depth, hands-on application under title classification of insects, implemented using the Android Platform, the fastest growing segment of the handheld system user base. This application will cover the fundamental programming principles underlying handheld software applications by designing artistic and engaging user interfaces that help students to master insects taxonomy readily. In conclusion the classification of insects app was successfully developed, offering multiplayer capabilities on various devices in the real world and getting run with no need for internet connection.

Keywords—Android Computing Platform, Insects Class, Motivations and Objective, Research Contribution, Software Procedures & Results
I. INTRODUCTION

Computing continues to become more and more personalized and accessible. Handheld devices have largely transformed into computing platforms. Mobile phones are no longer just for talking they have been capable of carrying data and video for some time. Be it a phone or a tablet, the mobile device is now so capable of general-purpose computing that it’s becoming more like a PC. A number of traditional PC manufacturers such as ASUS, HP, and Dell are producing devices of various form factors based on the Android OS.

The battles between operating systems, computing platforms, programming languages, and development frameworks are being shifted and reapplied to mobile devices. We are also seeing a surge in mobile programming as more and more IT applications start to offer mobile counterparts. Lastly, developers are excited about Android [1], because it is an advanced platform that introduces a number of new paradigms in framework design (even with the limitations of a mobile platform).

Mobile phones use a variety of operating systems, such as Symbian OS, Microsoft’s Windows Mobile, Mobile Linux, iPhone OS (based on Mac OS X), Moblin (from Intel), and many other proprietary OSes. So far, no single OS has become the de facto standard. The available APIs and environments for developing mobile applications are too restrictive and seem to fall behind when compared to desktop frameworks [2]. In contrast, the Android platform promised openness, affordability, open-source code, and, more important, a high-end, all-in-one-place, consistent development framework.

This work, shows how implement an android application framework and simulate it on an android platform to catalog and classify the insects class into subclasses, orders, suborders, families, and subfamilies according to their morphology, each category will consist of stand-alone section. Students are not required to grasp all of these categories. Those who do, however, will gain a much more detailed, end-to-end understanding of insects properties and their taxonomy derived from hierarchy shown in Fig. 1.

![Fig. 1 The Hierarchy of Insect Class, described at [3].](image-url)
II. MOTIVATIONS AND OBJECTIVES

Insects represent nearly one-half of all plants and animals. Although scientists do not know how many insect species there are and probably will never know, some researchers believe the number of species may reach 10 to 30 million. Even a typical backyard may contain several thousand species of insects, and these populations may number into the millions [3].

Despite the huge existence for this group of animals, there is no simulation tool depicts them up to now. So our research efforts have been focused on this group in order to demonstrate it and provide a common ground for prototyping by means of simulation. The primary goal beyond the use of android platform is the ability to write the android application once and have it run on various devices because the android operating system is standardized for various devices. Finally, for more in-depth understanding and further information about the general insects subject, our android simulator offers a good framework and visualization interface.

III. RESEARCH CONTRIBUTION

Use of android devices has greatly expanded for the portability of these devices, their networking capability, computing power, available sensors (location, motion, optical, acoustic, etc.), gives them great potential for design software. More than ever, they are performing tasks that were traditionally performed on desktop or laptop computers. So, the main contribution of this research to design and implement simulation software is based on android platform.

We are going to start by introducing the android user interface (UI) model. In its UI, android is quite different from some other paradigms that we may be familiar with. The unique feature is its dual approach to UI via both Java and XML languages. Our approach will focus on best practices in UI development so that the application looks good and works well on any android device, regardless of screen size and resolution.

An Android application is a loose collection of Activities, Services, Content Providers, and Broadcast Receivers. These are the components from which we put together an application. Fig. 2, shows the design of the entire (Classification of Insects) application, which incorporates most of the main Android building blocks.

First, we’ll create the screen, which is an activity, a basic Android building block. Then, we’ll use Intents which send intents to open up a specific activity. It turns out that some of the cool features Android services provide can make our application service much simpler. So we’re going to use Alarm service to fire off these intents on regular basis. Additionally, we’ll add a feature to notify the user of new action by putting a notification in the notification bar. For that, we’ll use the Notification service. All this will prove to be a substantially more elegant solution to our application service need.

Finally, Lists and Adapters will tie the data to our user interface. They form a very powerful component that allows our tiny UI to connect to potentially large datasets in an efficient and scalable manner. In other words, users will be able to use this application in the real world without any performance hits in the long run.
Fig. 2. GUI Diagram for The Insects Classification Software
IV. SOFTWARE PROCEDURES & RESULTS

This section presents the major algorithms and results of a study in which the insects classification software is built and deployed on a real device (HTC_Sensation) and emulator (AVD) as follows:

A. Sleep Activity Algorithm

**Input:** first.xml file composed of linear layout and button.

**Output:** Sleep Screen.

**Process:**

1. **Step 1.** Apply the layout (first.xml) to the sleep activity.
2. **Step 2.** Set a custom theme for the activity style in manifest file.
3. **Step 3.** Create timer.
4. **Step 4.** Try sleep for 5000ms.
5. **Step 5.** Create toast message.
6. **Step 6.** Catch interrupted exception.
7. **Step 7.** Finally create an intent in order to lunch the next activity.

B. Starting Point Activity Algorithm

**Input:** main.xml file (relative layout and three buttons), and dialog.xml file (relative layout, text view and image view).

**Output:** Starting point Screen.

**Process:**

1. **Step 1.** Set content view using main.xml layout.
2. **Step 2.** Define quit button by its id.
3. **Step 3.** Set onClick method to do quit button actions.
4. **Step 4.** Create layout inflator to apply the layout (dialog.xml) for alert dialog view.
5. **Step 5.** Create builder for the previously mentioned dialog.
6. **Step 6.** If negative button of builder is pressed then cancel dialog.
7. **Step 7.** Create a toast message as a feedback to the user.
8. **Step 8.** If positive button of builder is pressed then finish the starting point activity.
9. **Step 9.** Find about button view by its id.
10. **Step 10.** Set onClickListener for about button.
Step 11. Create an intent to get context from about class.

Step 13. Define go button by its id.

Step 12. Set onClick method to do go button actions.

Step 13. Create an intent to lunch insects subclasses activity.

Step 14. Lastly create a custom theme and apply it for the activity style attribute.

C. About Activity Algorithm

Input: about.xml file (linearlayout, textview, imageview, and button).

Output: About Screen.

Process:

Step 1. Apply the layout (about.xml) to the about activity.

Step 2. Change the title of the activity to be custom.

Step 3. Set activity style attribute to be as dialog theme.

Step 4. Set textview autoLink attribute to get web service.

Step 5. Define close button by its id and set onClickListener for it.

Step 6. If close button is pressed then finish about activity is done.

D. Subclasses Activity Algorithm

Input: subclass.xml file (tablelayout, textview, and radiobutton).

Output: Insects Subclasses Screen.

Process:

Step 1. Set content view using subclass.xml layout.

Step 2. Set a custom theme for the activity style in manifest file.

Step 3. Find 1st and 2nd radio button views by their id.

Step 4. React to events from the radio group.

Step 5. Send some feedback to the user through toast message.

Step 6. If the 1st radio button is pressed then pterygota division activity is lunched.

Step 7. Otherwise, perform the 2nd button action to get context from apterygota orders activity.
E. Pterygota Divisions Activity Algorithm

**Input:** pterygota.xml file (tablelayout, textview and radiobutton).

**Output:** Pterygota Divisions Screen.

**Process:**

**Step 1.** Set content view using pterygota.xml layout.

**Step 2.** Change the style of the activity to be custom.

**Step 3.** Find 1st and 2nd radio button views by their id.

**Step 4.** React to events from the Radio Group.

**Step 5.** Send some feedback to the user through toast message.

**Step 6.** If the 1st radio button is clicked then pass the control to the exopterygota orders.

**Step 7.** Else if the 2nd radio button is pressed then create an intent in order to lunch the next activity (endopterygota orders).

F. Apterygota Orders Activity Algorithm

**Input:** main_list.xml file (RelativeLayout, and ListView),
list_item.xml file (LinearLayout,ImageView, and TextView),
getItem, title, array of string for items names and array of integer for items images, and listview.

**Output:** Apterygota Orders Screen.

**Process:**

**Step 1.** Prepare data source for list by defining array of string for items names and array of integer for images.

**Step 2.** Override onCreate method at first created.

**Step 3.** Set content view for Apterygota Orders activity using main_list.xml file.

**Step 4.** Change the title of the activity to be custom.

**Step 5.** Define listview by it’s id.

**Step 6.** Create a custom adapter( external class extend from the ArrayAdapter) that maps data from the data source to single row.
Step 7. Give it the context.

Step 8. Give it the appearance for a single row or child view after defining it in a separate layout file called (list_item.xml).

Step 9. Give it a data source.

Step 10. Change the name for each item based on the position value.

Step 11. Change the image for each item based on the position value.

Step 12. Set the adapter and let it render.

Step 13. Define what happens once the list view item is clicked using OnItemClickListener method.

Step 14. If getItem equals first item then get me the dialog for the first row, by creating a builder and set a custom title and message for it.

Step 15. Else If getItem equals second item then get me the dialog for the second row, by creating a builder and set a custom title and message for it., etc.

Step 16. Repeat step 15 until we reach the last item in the array.

Step 17. Finally, get the dialog and show it.

G. Exopterygota Orders Activity Algorithm

Input: main_list.xml file (RelativeLayout, and ListView),
list_item.xml file (LinearLayout, and two TextView), getItem,

Fig. 9 Apterygota Orders

Fig. 10 Shows what the Apterygota Orders List looks like when the OnItemClickListener method run.

(a) Thysanura Insects  (b) Protura Insects  (c) Collembolla Insects

G. Exopterygota Orders Activity Algorithm

Input: main_list.xml file (RelativeLayout, and ListView),
list_item.xml file (LinearLayout, and two TextView), getItem,
title, string array for list items names, and listview.

**Output:** Exopterygota Orders Screen.

**Process:**

**Step 1.** Call `onCreate` method when the activity is first created.

**Step 2.** Apply the layout (`main_list.xml`) to the apterygota orders activity.

**Step 3.** Change the title of the activity to be custom.

**Step 4.** Get the listview by its id.

**Step 5.** Creating our own custom subclass called `MyAdapter` to extend the `ArrayAdapter` of the list items array from it.

**Step 6.** Set the adapter and let it render.

**Step 7.** Set an `OnItemClickListener` to react to selections in the list.

**Step 8.** If `getItem` equals first item then get me the activity for the first row, by creating an intent for that activity.

**Step 9.** Else if `getItem` equals second item then get me the activity for the second row, by creating an intent for that activity, etc.

**Step 10.** Repeat step 9 until we reach the last item in the list.

**Step 11.** Override `getView()` method of an Adapter to be called as needed for each position in the list.

**Step 12.** Let the `ArrayAdapter` (`MyAdapter`) in `getView()` method inflate the row layout using `list_item.xml` file.

**Step 13.** Finally, change the name for each item based on the position value.

![Exopterygota Orders](image-url)
Fig. 12 Shows what the Exopterygota Orders List looks like when the OnItemClickListener method run.

**H. Endopterygota Orders Activity Algorithm**

In the same way we build the Endopterygota Orders activity to be looks like figure below after running.

---

© 2014, IJCSMC All Rights Reserved
V. CONCLUSIONS AND FUTURE WORKS

The distribution of smartphones and tablet PCs running operation systems like Apple iOS or Android, attracts the interest in many application fields. In this paper we present insights into the development process of a Google Android-based system, designed for prototyping almost completely insects class categories and everybody, anywhere anytime can download it through smartphone and tablet PC web markets for free. By using this system we provided a solid foundation for validating our simulation with the demands of the recent device directive for software. The application proposed in this paper is developed using the SDK to simulate Insects taxonomy. So, in future works a simulation tool, integrating all these categories with their metamorphosis, should be developed. We believe that the future work on this topic can take advantage from the results presented in this article.

REFERENCES

