**TITLE:** Bugs Exhibit Application

**AUTHORS:** Michael Evans and Ethan Hearne

**ABSTRACT**

The Fort Lauderdale Museum of Discovery and Science is looking to use technology such as smart phones and tablets to improve the educational aspects of its exhibits. Currently the description in the museum exhibits are lacking in descriptive information regarding many of its creatures in the Ecoscapes section. Our goal here is to develop an app that will bring up more details about the insects that are featured, primarily to sketch out a general activity flow that can be used by any developer interested in displaying additional information about any exhibit. We will discuss the framework and methods that are used to complete our objectives. We will utilize the NFC sensor as an additional option to make the app more user friendly by allowing the user to easily scan tags for easy access of detailed information. With some graphics added, the user can visually tell the difference from one insect to another.

**BACKGROUND**

We got a thorough tour of the Museum of Discovery and Science, and we decided that the insect section was the best area to research and develop an app that enhances the learning and brings more to the visitor experience. This exhibit is in a section of the Florida Ecoscapes department located on the first floor of the building. Many of the insects present are land creatures such as scorpions, spiders and millipedes.

The goal of the Insect App is to present additional descriptions and facts of the featured insects that are displayed inside the MODS. We utilize tablet and phone features such as tapping and swiping mechanisms to make the user interface easier. Since the descriptions can be quite lengthy, we add a scrolling feature so that the user can read more about the insects.

**METHODS**

Our design involves a sequence of fragments that can be swiped through by the user. Each of these fragments contains a ScrollView, and each of these ScrollViews contains an ImageView and a TextView. The ImageView contains an image evoking the exhibit -- in our case we have artistic representations of insects. Below this, the TextView contains descriptive information about the particular part of the exhibit represented by the fragment -- in our case we divided the exhibit up by insect.

The main focus of our design was on the integration of NFC/RFID tags into the user experience of the application. We hypothesized that if RFID tags were placed around the museum at each exhibit, then this would encourage visitors, especially young children, to use their Android devices to enhance their museum experience. Thus, we wanted to build an application that could react to these RFID tags in an expected way related to the exhibit to which they were attached.

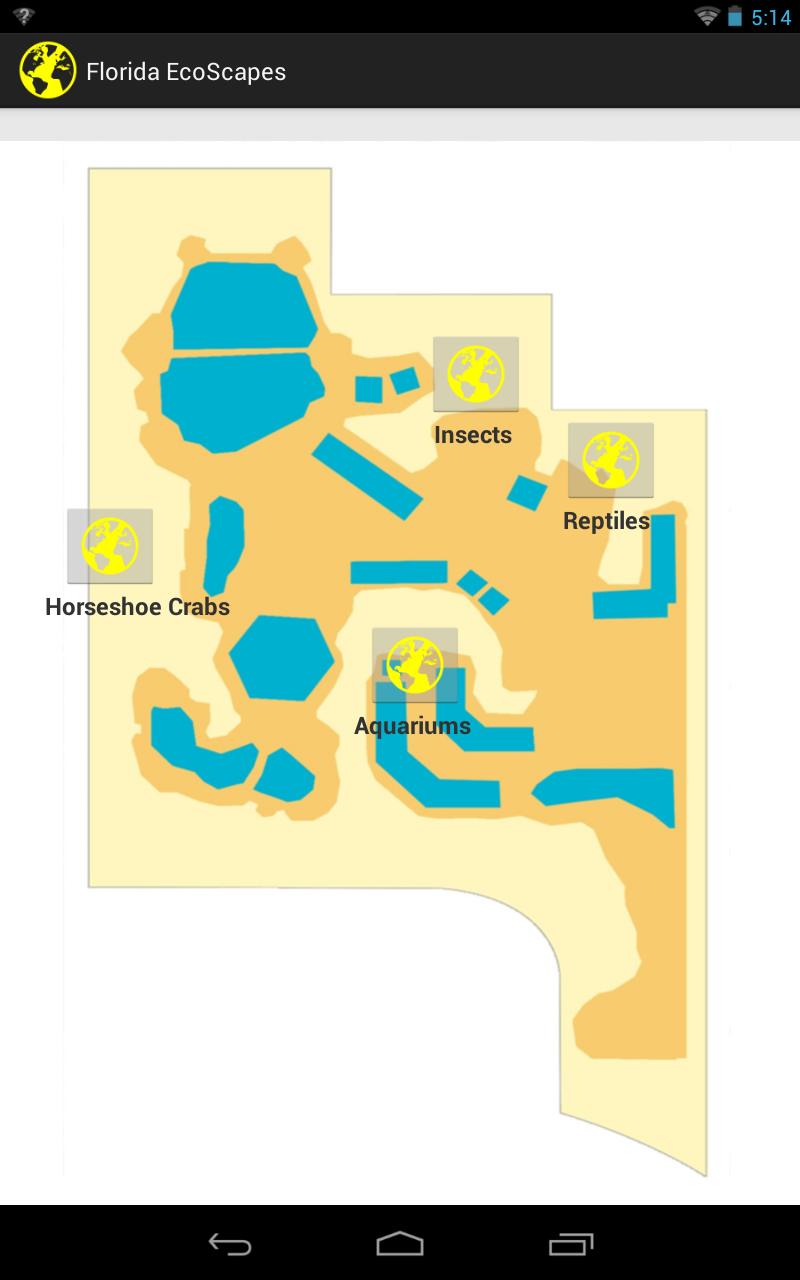
**RESULTS**

We successfully implemented the design, and it is a good start for developers interested in setting up a basic informational application. The application successfully allows the user to swipe through each fragment, and each ImageView and TextView is displayed visibly.



*Three of the many fragments that the user can swipe through to get more information about each insect in the exhibit.*

The application also successfully launches from the MODS framework application if the user taps on the icon representing the insects exhibit, which can be found in the activity containing the floor map representing the Ecoscapes theme. It also launches successfully if the user scans an RFID tag, with different tags launching different fragments within the application, according to the specific insect described by the URI encoded on the tag.



*The Ecoscapes floor map containing the insects icon from which our exhibit app can be launched*

There were two main challenges in the implementation of our Android application. The first was the management of memory when using media-rich fragments. Our application consists of one activity that uses about ten fragments. These fragments each contain an image. To begin with, the application worked perfectly, but after swiping through multiple fragments the application threw an OutOfMemory exception. In order to work around this, we had to crop and resize the images to an eighth of their original sizes. This error was caused by our inefficient management of fragments, and we conclude that they are an advanced feature to be studied in greater depth for any developer interested in using them.

The second challenge was correcting the unintuitive behavior that arises when the user scans an NFC tag while the application is already in the foreground. When the application is already in the foreground, the mechanism used to launch a specific fragment no longer works, because it relies on the launching of a broadcast from the MainMenu activity of the framework application. This broadcast is detected by the listener declared in the manifest of our exhibit application, and an intent is dispatched containing an extra integer indicating the fragment to display. The downside to this mechanism is that this intent is only used to create, start, or resume an application that is not created or is in the background. This means that if the user leaves the application open on a specific fragment while they scan an RFID tag, an intent is launched correctly, but the application is already running and so it ignores it and the extra integer it came packaged with. We tried to work around this with the OnNewIntent method, but this did not seem to behave as indicated.