**“Green on the Go” Museum of Discovery and Science (MODs)**

By

Abhinav Komaravelli (Programmer), Alec Handal (Main Artist), David Hernandez (Co-artist), and Oluwapelumi Omole (Project Manager)

# Abstract

The purpose of this project was an attempt to create an Android app for the “Go Green” exhibit of the Museum of Discovery and Science. This app is merely to educate society about the environment and the impact on it. The exhibit in the museum is the central idea of the project. This is a mobile version of the information provided on their webpage and museum. The app contains a component on recycling purely meant to inform children for the better of our environment.

# Background

Mobile technology has substantially improved over the last decade or so. In contemporary society, nearly everyone wields their own mobile device, widely known as “smartphones”. These user-friendly devices have been given such a name due to their wide range of capabilities and connectivity such as web browsing, a digital camera, a GPS navigator, and the use of touchscreen. A multitude of new features and advances have been made in the mobile industry. Basically, the entire world can be seen through a mobile device in one’s possession. Since there is such a high interest for this technology, we made an application in the same genre as a guide for the exhibit with some additional information about the environment. The overall goal is to send a message about how important the environment is and how essential it is to the human population. As part of the Math, Science and Engineering (MSE) program in Suncoast Community High School, it was mandatory to enroll and take any course in the Engineering Scholars Program (ESP). This project is a partial fulfillment of the “Mobile Applications to Google’s Android” course in the ESP program at Florida Atlantic University in 2014.

# Methods and Results

Several tools were used to develop this android app, which included Integrated Development Environment (IDE) using Eclipse, plugin for the Eclipse IDE called Android Developer Tools (ADT) which included an android emulator, Processing 2.0, Android Software Development Kit (SDK). Java version 1.8.0\_05 was used in the development of this app. The minimum of Android 2.3.3 Gingerbread is required to run this application, but Android 4.0 Ice Cream Sandwich was used to develop this app. For the development of this Android application, the menu screen and the contents of the app were predesigned through a storyboard. An Asus Nexus 7 (2012 model) was used to test the application in an android environment. The basic debugging was done using the emulator which was installed along with the Android SDK. All the layouts are defined using the XML format. Our team has selected a few pictures to use as part of the application display in various screens. They were pre-selected to signify the beauty of nature and the environment so that many more people may value it. The programming logic was developed using Java. We tried to make the app very user-friendly by allowing it to respond to horizontal and virtual scrolling as well.

The total application comprises of XML files used for layouts, pictures used for backgrounds, java programs for logic and program controls, and most of the text has been initialized in the strings. Most of the Java code imports Android app activity and Android OS bundle classes. Other classes are exported from the java libraries when needed.

When the user launches the application, the MainActivity class is initiated and loads the initial screen. This initial screen layout is defined in activity\_main XML. This displays the logo of the app on top of the screen with a stunning background showing how beautiful Mother Nature is. It also displays four main buttons: “Interactive Map”, “Go Green”, “Recycle”, and “About Us.” Each of these buttons take the user to a different screen and show the components of the selected button. This MainActivity Java class has a couple of methods, onCreate and onClick, with a switch statement to control the main logic and select the proper class based on the button pressed.

The “Interactive Map” button, when selected, runs the InteractiveMapContents class. This java class is a simple class with one OnCreate method and loads the screen for which the layout is defined in the floor\_maps XML file. This will display the floor maps of the first and second floors of the museum. These screens can be swiped horizontally when displayed. These maps are extremely useful and can practically take the reader, not just through the Go Green exhibit, but anywhere they desire within the entire museum. This application development involves taking the advantage of the “Go Back” button of the android device. This button needs to be tapped when the user wishes to go to the previous screen.

When the “Go Green” button is selected from the menu on the initial screen, it runs the GoGreen Java class and goes to another screen. This class contains some simple Java code to control the flow of the program, and couple of methods, onCreate and onClick. This class displays another screen with two more buttons, “About Exhibit” and “Fun Facts”. This display layout is defined in the gogreen XML file. This has a striking background screen with a leaf-like frame around the buttons. This class will invoke two other classes based on the button selected. It invokes the AboutExhibitContetnts class when the “About Exhibit” button is pressed and invokes the FunFactsContents class when the “Fun Facts” button is pressed. The “About Exhibit” screen gives a lot of information about what is in this exhibit. It mainly explains Harry the Heap, the environmentally friendly robot, and also briefly explicates about making bright recycling choices. This screen is programmed so that the user can scroll up and down. The display content for this layout is defined in the fragment\_go\_green\_main XML file. On the other hand, when the user select the “Fun Facts” button from this screen, it will run the FunFactsContetnts Java class, which eventually displays a few facts about the nature and ecosystem. This screen is also programmed to be able to be scrolled vertically, and it is launched using the layout defined in the fun\_facts\_contents XML file. These interesting facts are displayed with a preselected nature background. Both of the classes, AboutExhibitContetnts and FunFactsContents, are written with one onCreate method.

When the “Recycle” button is selected, it launches the Recycle class, which only has one method. It redirects the user to a screen where it displays random items that one could come across in their daily lifestyle. This screen layout is defined in the recycle XML layout file. This layout is also programed to be scrollable vertically. This part of the app is actually unique because the background was edited. The purpose of the modifications to the background was to help give the

From the main menu when “About Us” button is pressed, it leads to another screen to know about the details of the project developed. It initializes the AboutUsContetns java class for which the layout it is defined in AboutUs XML file. This screen shows the details the team, who developed this application and also gives information about the persons who helped for this project.

# Discussion

The main purpose for the choosing of the theme of the app to be environmental is to promote a greater care for nature in people everywhere. The museum has this exhibit, so this was the easiest way to start. Although the app has been completed, there is no denying that the final version was accomplished to the fullest extent. One of the group members (Oluwapelumi) was originally going create a full floor-plan that would allow filtering of certain areas such as restrooms and fast food stops, but due to both limitations in time and experience, it could not have been done. Another group member (Abhinav) was planning on including a much smoother User Interface with more menus and an added section containing interesting articles pertaining to “Go Green”, but it was also not completed in time. The group’s main artist, Alec, attempted to make a game to educate children about the many benefits to recycling in a harmless way, but due to difficulties and restraints, he cut it down into a simple list of items that are recyclable and not recyclable. His game was originally supposed to be where the “Recycle” section of the app is now. The group still did collaborate excellently in the short time given and got experience, so the app can definitely be improved to the fullest extent. The group was planning on adding features to the map such as markers and a zoom in function, as well as completing the game if give more time. The group also improved on their skills over time due to the rigorous tasks. In a nutshell, the group accomplished a lot together and all pitched in to make this app.

# Conclusion

This project has been successfully executed to show some of the information about the Go Green exhibit in the museum of discovery in Fort Lauderdale, Florida. This android application is accommodated with easy touch screen buttons to explore the information to be utilized on android mobile devices. The information provided in this app is very limited because of the time taken to develop this application. This application can be extended further to incorporate complete information about Go Green exhibit if needed. Overall this project has been completed in the given limited time. This project was challenging to our team since it is developed in less than a week by a team of four students, who do not have in depth knowledge and learned about the android programming in less than three weeks of time. The authors are grateful to the teacher and the teaching assistants who helped us do achieve this. If time permits, the same team will further develop this project to incorporate more interactive information, add more enhancements and more features.

# References

Sources of Background:

<http://battleforge.wikia.com/wiki/File:CardFrame_Nature.png>

<http://upload.wikimedia.org/wikipedia/commons/1/1a/Bachalpseeflowers.jpg>

# http://files.all-free-download.com/downloadfiles/wallpapers/1920\_1200\_widescreen/peaceful\_lake\_wallpaper\_landscape\_nature\_wallpaper\_1920\_1200\_widescreen\_1208.jpg

# Appendices

The full Appendices can be seen with the attached code (project).

1. MainActivity.java

package com.example.testkk;

import android.content.Intent;

import android.os.Bundle;

import android.support.v4.app.Fragment;

import android.support.v7.app.ActionBarActivity;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.View.OnClickListener;

import android.view.ViewGroup;

public class MainActivity extends ActionBarActivity implements OnClickListener{

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// Set up click listeners for all the buttons

View aboutButton = findViewById(R.id.aboutUsButton);

aboutButton.setOnClickListener(this);

View goGreenButton = findViewById(R.id.goGreenButton);

goGreenButton.setOnClickListener(this);

View interactiveMapButton = findViewById(R.id.interactiveMapButton);

interactiveMapButton.setOnClickListener(this);

View gameButton = findViewById(R.id.gameButton);

gameButton.setOnClickListener(this);

}

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

switch (v.getId()) {

case R.id.aboutUsButton:

Intent i = new Intent(getApplicationContext(), AboutUs.class);

startActivity(i);

break;

case R.id.goGreenButton:

startActivity(new Intent(this, GoGreen.class));

break;

case R.id.interactiveMapButton:

startActivity(new Intent(this, InteractiveMapContents.class));

break;

case R.id.gameButton:

startActivity(new Intent(this, Recycle.class));

break;

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

/\*\*

\* A placeholder fragment containing a simple view.

\*/

public static class PlaceholderFragment extends Fragment {

public PlaceholderFragment() {

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View rootView = inflater.inflate(R.layout.fragment\_main, container, false);

return rootView;

}

}

}

1. AboutUs.java

package com.example.testkk;

import android.app.Activity;

import android.os.Bundle;

public class AboutUs extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.aboutus);

}

}

1. AboutExhibitContents.java

package com.example.testkk;

import android.app.Activity;

import android.os.Bundle;

public class AboutExhibitContents extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.fragment\_go\_green\_main);

}

}

1. FunFactsContents.java

package com.example.testkk;

import android.app.Activity;

import android.os.Bundle;

public class FunFactsContents extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.fun\_facts\_contents);

}

}

1. GoGreen.java

package com.example.testkk;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

public class GoGreen extends Activity implements OnClickListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.gogreen);

View exhibitsButton = findViewById(R.id.aboutExhibit);

exhibitsButton.setOnClickListener(this);

View funFactsButton = findViewById(R.id.funFacts);

funFactsButton.setOnClickListener(this);

}

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

switch (v.getId()) {

case R.id.aboutExhibit:

Intent i = new Intent(getApplicationContext(), AboutExhibitContents.class);

startActivity(i);

break;

case R.id.funFacts:

startActivity(new Intent(this, FunFactsContents.class));

break;

}

}

}

1. InteractiveMapContents.java

package com.example.testkk;

import android.app.Activity;

import android.os.Bundle;

public class InteractiveMapContents extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.floor\_maps);

}

}

1. Recycle.java

package com.example.testkk;

import android.app.Activity;

import android.os.Bundle;

public class Recycle extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.recycle);

}

}