Speech Based Accessibility - Medium

Background

This module demonstrates the use using code to set the contentDescription attribute of UI controls on the screen. The motivation for this activity, is that there could be instances where the UI is built or changes dynamically and hence it's not feasible to hardcode the contentDescription value in the layout xml.

A working version of this app is available at: https://github.com/milk-modules/Apps/tree/master/accessible/DemoApp06

Prerequisite

1. Android studio is installed on the development workstation
2. A working Android emulator is available for testing
3. TalkBack is enabled on the emulator.
    a. Details on installing and activating TalkBack: https://milk-modules.github.io/activities/general/Android_TalkBack_Install.pdf

Activity Instructions

This activity will utilize a pre-created version of this project and only applies accessibility attributes to the UI controls. Download the code for DemoApp06 from: https://github.com/milk-modules/Apps/tree/master/non-accessible

The project contains only one screen (activity). The primary user interface (UI) elements of this screen are:

1. Two panels displaying the weather forecast for two days ('Today' & 'Tomrrow'). Each panel contains:
    a. Text labels displaying the high and low temperature
    b. An image of the weather forecast

Apply the below code in the file: **fragment_weather.xml**

Steps

1. Add the code highlighted in yellow to the '`FrameLayout`' property listing:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/frameLayout_fragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
        android:accessibilityLiveRegion="polite"
        tools:context="edu.rit.se.milk.demoapp06.WeatherFragment">
```

2. Add the code highlighted in yellow to the ‘`textViewHigh`’ property listing:

```
<TextView
    android:id="@+id/textViewHigh"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/imageView"
    android:layout_alignStart="@+id/imageView"
    android:layout_alignTop="@+id/imageView"
    android:layout_marginLeft="13dp"
    android:layout_marginStart="13dp"
    android:layout_marginTop="20dp"
    android:clickable="false"
    android:importantForAccessibility="no"
    android:text="TextView" />
```

3. Add the code highlighted in yellow to the ‘`textViewLow`’ property listing:

```
<TextView
    android:id="@+id/textViewLow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textViewHigh"
    android:layout_alignStart="@+id/textViewHigh"
    android:layout_below="@+id/textViewHigh"
    android:layout_marginTop="19dp"
    android:importantForAccessibility="no"
    android:text="TextView" />
```

Apply the below code in the file: **WeatherFragment.java**

Steps

1. Update the method ‘`generateRandomWeatherForecast`’ by adding the code highlighted in yellow:

```
public void generateRandomWeatherForecast(String day){
    Random rand = new Random();
    int weather = rand.nextInt(4) + 1;
    int high = rand.nextInt(80 + 1 - 50) + 50;
    int low = rand.nextInt(high + 1 - 20) + 20;

    txtHigh.setText("High Temperature: "+high+"F");
    txtLow.setText("Low Temperature: "+low+"F");

    Resources res = getContext().getResources();
    Drawable img;
    String text;
    switch (weather){
        case 1:
            img = res.getDrawable(R.drawable.weather_icon_01);
            text = "sunny with a few thunderstorms";
            break;
        case 2:
            img = res.getDrawable(R.drawable.weather_icon_02);
            text = "sunny with a few clouds";
            break;
        case 3:
            img = res.getDrawable(R.drawable.weather_icon_03);
            text = "a bright and sunny day";
            break;
        default:
            img = res.getDrawable(R.drawable.weather_icon_04);
            text = "thunderstorms throughout the day";
            break;
    }

    day = day.equalsIgnoreCase("today")?"Today's":"Tomorrow's";

    String accessibilityText = day +
            " weather is set to be  " + text +
            "with a high of " + high +
            "and a low of " + low;
    frameLayout.setContentDescription(accessibilityText);
    weatherIcon.setImageDrawable(img);
}
```

Apply the below code in the file: **MainActivity.java**

Steps

1. Update the method ‘`generateWeather`’ by adding the code highlighted in yellow:

```
private void generateWeather(){
    weatherTimer = new Timer();
    weatherTimer.schedule(new TimerTask() {
        @Override
        public void run() {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    todayWeather = (WeatherFragment)
getSupportFragmentManager().findFragmentById(R.id.weather_fragment_today);
                    todayWeather.generateRandomWeatherForecast("today");


                    tomorrowWeather = (WeatherFragment)
getSupportFragmentManager().findFragmentById(R.id.weather_fragment_tomorrow);
                    tomorrowWeather.generateRandomWeatherForecast("tomorrow");
                }
            });
        }
    }, 0, 30000);
}
```

Deploy the app into the emulator. Touch the weather panels to hear the weather forecast. The forecast will auto update every 30 seconds with random values. When the panel auto updates, it will automatically announce/speak the forecast.