Now that we’ve learned how to use SharedPreferences to be able to store our points in our Cookie Clicker game, the next thing to create is point shop for us to spend our points.

To do that, we’re going to learn about Intents to switch activities, specifically from our main activities to our new shop activity.

So let’s get started.

# What is Intent?

There are two ways for you to change the View of your app and that’s with: Intents and Fragments.

* **Intent** is Android’s way to signal your app that you want to change Activities.
* **Fragments** on the other hand keep your app on the same Activity but you would be creating your own Fragment class that you can easily re-use in your Activities whenever you want.

We’ll talk about Fragments some other tutorial, so let’s talk about Intents!

## How to use Android Intent:

Here’s how you use Intents.

You get your Intent object:  
Intent i = **new** Intent(getApplicationContext(), NewActivity.**class**);

You would give your Intent the context of your app and the class of the Activity that you want to switch to.

You can also add data to pass in to other activities by doing:

i.putExtra(“key”, “value”);

Similar to how PlayerPreference works, you can store data in a key-value pairing. Your Key is a String, but your value can be any primitive type (String, int, double, etc).

Once you’re done with everything you call:

startActivity(i);

to start the new Activity.

## Getting Intent Data

Now that you have changed the Activity, the next thing you have to do is to grab the data from the Activity.

In the new Activity in your onCreate() function You have to get a Bundle object:

Bundle extra = getIntent().getExtras();

Check if your Bundle is null or not in the event that you didn’t reach the Activity via an Intent.

If the bundle isn’t null, you can get the data that you passed in by:

extra.getInts(“key”, “**default**”);

Similar to SharedPreferences you use your key to get your value that you store. If there is no value, you would specify a default value that would be used instead.

**Special note:** when we switch activities. Our current activity will finish it’s life cycle and close, and our new activity will start its own cycle

And that’s all! Now let’s see this in practice.

# The Code

The first thing we’re going to have to do is create a new Shop Activity that we can later use to buy upgrades for our Cookie Clicker game.

For our Shop Activity, all we need are:

* A TextView to show the points we have to spend
* A back button to bring us back to our Main Activity.

We’ll create a new XML file: activity\_store.xml in res -> layout to create the interface of our ShopActivity.  
When you’re done, you should have something like this:

Activity\_shop.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:text="Points: 0"  
 android:id="@+id/tvShopPoints"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="157dp"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Back"  
 android:id="@+id/btnBack"  
 android:onClick="backToMain"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
</**RelativeLayout**>

You’ll get something like this:



Afterwards, let’s create our Activity class. Right click **java** -> **<your package name>** and select **new** -> **Java Class** and create ShopActivity.java

Now before we jump into ShopActivity.java, let’s use Intents inside our MainActivity

Inside our activity\_main.xml we want to add:

* A button that will allow us to take us to our ShopActivity.

After you’ve added your button, you should have something like this:

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="net.joshchang.josh.week8.MainActivity"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:text="Score: 0"  
 android:id="@+id/score"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Click"  
 android:id="@+id/button"  
 android:layout\_marginTop="37dp"  
 android:layout\_below="@+id/score"  
 android:layout\_alignRight="@+id/score"  
 android:layout\_alignEnd="@+id/score"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Store"  
 android:id="@+id/store"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"** />  
</**RelativeLayout**>

You’ll get something like this:



Now let’s go back to our MainActivity.java to add to add a onClickListener to our Shop button that will use the Intent code.

MainActivity.java

**package** net.joshchang.josh.cookieclicker;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** TextView **tvPoints**;  
 **private** Button **btn**;  
 **private int points**;  
 **private** SharedPreferences **pref**;  
 **private** Button **store**; *// !!!! Change #1  
  
 // Hard-coded values to prevent confusion with variable types* **private final** String **PREF\_NAME** = **"pref"**;  
 **private final** String **POINTS** = **"totalPoints"**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *// Basic setup* **tvPoints** = (TextView) findViewById(R.id.***tvPoints***);  
 **btn** = (Button) findViewById(R.id.***btnPoints***);  
 **store** = (Button) findViewById(R.id.***btnShop***); *// !!!! Change #2  
  
 // Restoring data with pref* **pref** = getSharedPreferences(**PREF\_NAME**, Context.***MODE\_PRIVATE***);  
 *// Get the points that we store in the pref. Otherwise the default value* **points** = **pref**.getInt(**POINTS**, 0);  
 *// Sets the text* **tvPoints**.setText(**"Score: "** + **points**);  
 }  
  
 *// opens the new ShopActivity* **public void** shopButton(View v) {  
 Intent intent = **new** Intent(getApplicationContext(), ShopActivity.**class**);  
 intent.putExtra(**"points"**, **points**);  
 startActivity(intent);  
 }  
  
 *// Increment our points and displays it* **public void** btnClick(View v) {  
 **points**++;  
 **tvPoints**.setText(**"Points: "** + **points**);  
 }  
  
 *// Whenever the app is paused we would save our points* @Override  
 **protected void** onPause() {  
 **super**.onPause();  
  
 *// Gets a Editor from our shared preference* SharedPreferences.Editor edit = **pref**.edit();  
 *// Add our points with our editor to be saved* edit.putInt(**POINTS**, **points**);  
 *// Save our changes* edit.commit();  
 }  
}

Now let’s go to our ShopActivity.Java and receive the data that we pass over.

ShopActivity.java

**package** net.joshchang.josh.cookieclicker;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**public class** ShopActivity **extends** AppCompatActivity {  
 **private int points**;  
 **private** TextView **tvPoints**;  
  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_shop***);  
  
 *// Get the data that our Intent passed in* Bundle extra = getIntent().getExtras();  
 **points** = 0;  
 *// check if we received valid data* **if** (extra != **null**) {  
 *// get the points that we stored or else get a default value of 0* **points** = extra.getInt(**"points"**, 0);  
 }  
  
 *// Creates and set our TextView to show our points* **tvPoints** = (TextView) findViewById(R.id.***tvShopPoints***);  
 **tvPoints**.setText(**"Points: "** + **points**);  
  
 *// Creates our back button to send us back to MainActivity* Button back = (Button) findViewById(R.id.***btnBack***);  
 }  
  
 *// brings us back to the main activity* **public void** backToMain(View v) {  
 Intent intent = **new** Intent(getApplicationContext(), MainActivity.**class**);  
 startActivity(intent);  
 }  
}

Now before we run our app…

…It is **EXTREMELY** important that we add our Activity in,otherwise our app wouldn’t know about the activity and we would crash whenever we try to use the activity,

AndroidManifiest.XML

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="net.joshchang.josh.cookieclicker"**>  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity android:name=".StoreActivity"**></**activity**>  
 </**application**>  
  
</**manifest**>

All we have to do is create a new activity tag and add the name of our StoreActivity class:

<**activity android:name=".StoreActivity"**></**activity**>

Now run the emulator or on your phone and you should be able to switch activities now.

# Conclusion

Now with that we have a Cookie Clicker game that allows us to gain points and go to a shop.

Right now our shop is empty, but next time, we’ll learn about using SQLite to be able to store upgrade prices in our app to allow us to spend our points.

It’s going to be a big article next time. Rest up and prepare yourself to finish our Cookie Clicker game!

Like always, leave comments if you have any questions!