Integrate Kiip

Adding Kiip Resources
In the KiipSDKXamarin directory, you'll find the KiipDroidArtResources folder. Copy the contain of KiipDroidArtResources to your project resources folder.

Adding Kiip Library Reference
There is two methods to reference the library:

1) Adding the KiipSDKXamarin/KiipSDKXamarin.Droid solution directly to your project and referencing the solution directly.
2) Precompiling the KiipSDKXamarin/KiipSDKXamarin.Droid solution and referencing the generated dll.

Adding Kiip Permissions
In your Properties/AndroidManifest.xml file, you will need to request the following permissions (if you haven't already) to use the Kiip SDK:

```
<!-- Kiip -->
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<!-- END -->

<!-- Kiip Optional Permissions -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<!-- END Kiip Optional Permissions -->
```

Creating a BaseActivity Class
The best way to let the Kiip SDK know whenever a new activity in your application is started or stopped is to create an alternative Activity that all the activities will derive. See the sample BaseActivity.cs in KiipSDKXamarin/DroidSimplesample/BaseActivity.cs

Creating an Application Class
Create a new class named App.cs and have it inherit Application. See the sample App.cs in in KiipSDKXamarin/DroidSimplesample/App.cs

Initializing Kiip in your Application's onCreate()
Now that we have an application class, we'll need to implement the onCreate() method and add in the Kiip initialization. Modify your onCreate() to look like this:

```
public override void OnCreate ()
{
	base.OnCreate ();
	Kiip kiip = Kiip.Init(this, APP_KEY, APP_SECRET);
	Kiip.Instance = kiip;
}
```

Call a Kiip Moment

Now that your application lifecycle is all setup, it only takes one simple call to start receiving rewards.

You can call a Kiip moment by doing this in a subclass of BaseActivity.cs:

```
Kiip.Instance.SaveMoment(momentId,this);

public override  void OnFailed (Kiip kiip, Java.Lang.Exception exception)
{

}

public override  void OnFinished (Kiip kiip, Poptart poptart)
{
	OnPoptart(poptart);
}
```

Reward Virtual Currency

To enable virtual currency, you will need to create a currency listener and let Kiip know you can take Virtual Rewards. This listener will be called whenever a Virtual Reward is served and the user redeems it.

```
public class MyContentListener : Kiip.IOnContentListener
{
	public void OnContent (Kiip kiip, string content, int quantity,
		string transactionId, string signature)
	{
		// Give the currency to the user by using your in-app currency management.
	}

}
Kiip.Instance.SetOnContentListener(new MyContentListener());
```