AUTODESK UNIVERSITY 2013

Collect Data with Phone Sensors and Bring It Inside Autodesk® AutoCAD® Civil 3D® with API

Augusto Goncalves – Augusto Goncalves

DV3016 This class is a continuation of the AU 2012 class "CP3325: From Mobile and Through the Cloud to AutoCAD® Civil 3D®" that showed how to collect coordinate points using mobile with GPS-enabled devices running Android™ or Apple® iOS, and then how to store and merge those in the cloud to generate Autodesk AutoCAD Civil 3D objects. This year's class shows you how to collect more data with mobile device sensors running Android™, such as GPS, compass and camera, and then how to store and merge this data in the cloud to finally use inside Civil 3D software to create TinSurface, Parcel, and other site features. Prior knowledge of .NET programming and the Civil 3D API are required. Note that the 2012 class is available online at http://au.autodesk.com/au-online/classes-on-demand/class-catalog/2012/autocad-civil-3d/from-mobile-and-through-the-cloud-to-autocad-civil-3d.

Learning Objectives

At the end of this class, you will be able to:

- Connect mobile and web-based applications with the Civil 3D API
- Consume data with the Civil 3D API
- Create a mobile application

About the Speaker

Augusto is member of Autodesk DevTech team since 2008 based at Sao Paulo office. He is Civil Engineer with a Master in Computer Science, also a specialist in AutoCAD, Civil3D, Revit and Inventor APIs.
augusto.goncalves@autodesk.com

Overview

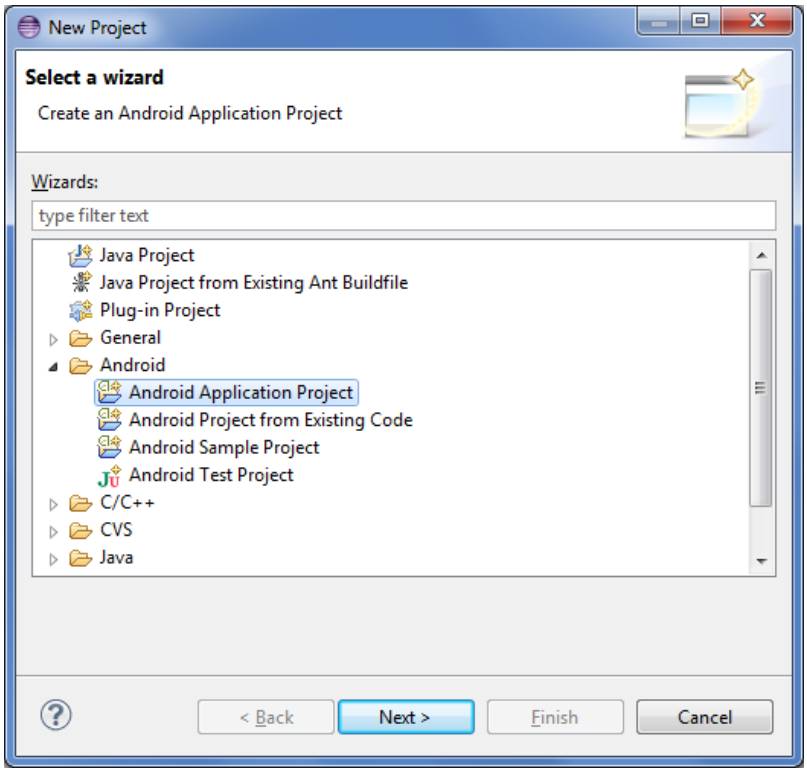
This material will focus on the basic references required to create Android apps and how work with this data on Civil 3D. Basically, it's required to access the GPS from the mobile device and convert this information to a drawing coordinate system.

Mobile

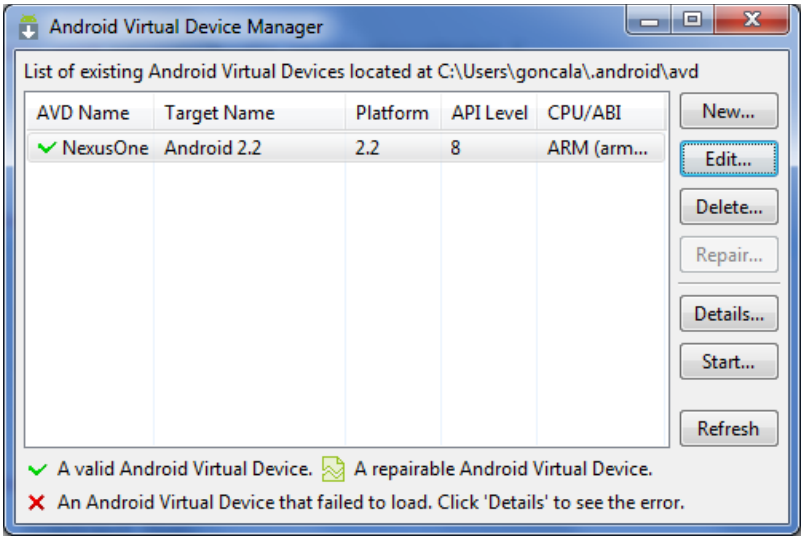
This module was can be developed for any device as the first module is platform independent. As mentioned before, this material will use Android platform from Google, which can be deployed on phones and tablets.

Different from the Cloud module, this requires a Java IDE. Google suggest and offer an interesting Android SDK for Eclipse, which includes the templates required to create projects and a good emulator to test the projects. It also has a package of drivers to debug on an actual device, which is recommended. The image below shows the Android template at the list of templates after we install the SDK.

The emulator, shown below, can be configured for any version of the OS. From Android website dashboard, the version 2.2 is used on more than 98% of the devices. It is very important to pay attention to this, as there are many changes between 2 and 3, and 3 and 4.



"New Project" wizard with Android template



Emulators available at Eclipse after installing the Android SDK

The Android development is based on Activities. Basically any new feature added to the project will be implemented on a separated class that implements this interface. One important change between version 2 and 4 is that Internet requests cannot be made from the main Activity, but to maintain the same code stream for both versions, this sample code added a security feature to allow this connection.

The GPS is tracked at the **LocationChanged** event. There is a strategy to improve the accuracy at Android Develop webpage. To use the location, several permission are required, so please make sure all are specified and the manifest XML. The sample code of this project has all permissions required, below is the relevant part.

```
<uses-permission android:name="android.permission.ACCESS_GPS" />
<uses-permission android:name="android.permission.ACCESS_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

<uses-feature android:name="android.hardware.camera" />
```

Now the camera also requires some permissions and features. The sample code above shows it. Like other features, the Camera is an Activity, and we can use the native (built-in camera) or build a custom one. This sample will use the native for simplicity. To use it, just start a camera activity and wait for the response at **onActivityResult** event.

Desktop

This module was implemented using regular Civil 3D APIs. This class will not focus on getting start with this API, but there are some training materials available at the Developer Center, including previous webcasts and blog.

To convert between coordinate systems, more specifically from LL84 (default for mobile GPS) to the current Civil 3D project, two new references are required: **OSGeo.MapGuide.Geometry.dll** and **OSGeo.MapGuide.Foundation.dll**, both available at Civil 3D installation folder. Below is a sample code demonstrating the approach to make this conversion. Before execute it, make sure the coordinate system is set on the drawing settings of Civil 3D.

```
public static Point3d ConvertPointCoordinate(Point3d point_LL84)
{
  MgCoordinate coord = null;
  if (CSTransform != null && !double.IsNaN(point_LL84.Z))
    coord = CSTransform.Transform(point_LL84.X, point_LL84.Y, point_LL84.Z);
  else
    coord = CSTransform.Transform(point_LL84.X, point_LL84.Y);
  return new Point3d(coord.X, coord.Y, coord.Z);
}

private static MgCoordinateSystemTransform _cstransform = null;

private static MgCoordinateSystemTransform CSTransform
{
  get
  {
    if (_cstransform != null) return _cstransform;
    // get Civil3D current coordinate system
    CivilDocument civilDoc = CivilApplication.ActiveDocument;
    string code = civilDoc.Settings.DrawingSettings. _
                      UnitZoneSettings.CoordinateSystemCode;
    // get the transformation
    MgCoordinateSystemFactory coordSysFactory = new MgCoordinateSystemFactory();
    MgCoordinateSystem coordSys = coordSysFactory.CreateFromCode(code);
    MgCoordinateSystem wgs84Sys = coordSysFactory.Create("LL84");
    // convert the coordinates
    _cstransform = coordSysFactory.GetTransform(wgs84Sys, coordSys);
    return _cstransform;
  }
}
```

Further reading

Some useful Civil 3D® online material:

- Civilized Development blog - http://civilizeddevelopment.typepad.com
  Isaac Rodrigues's Civil 3D API focused blog includes several example codes for .NET

- Civil 3D® Online help with documentation of the entire API

- Civil 3D® Developers Center – http://www.autodesk.com/developcivil
  Training material, recorded presentations and tools.

- Discussion Groups monitored by Autodesk team.

- Information about the Autodesk Developer Network - http://www.autodesk.com/joinadn
  ADN members can ask unlimited API questions through our DevHelp Online interface

- API Training - http://www.autodesk.com/apitraining
  Information about upcoming training classes and webcasts, also download of webcasts

- Watch out for our regular ADN DevLab events. DevLab is a programmers' workshop (free to ADN and non-ADN members) where you can come and discuss your AutoCAD programming problems with the ADN DevTech team.

Conclusion

Thank you for attending this session on Civil 3D, Cloud and Mobile. I hope you found the class enjoyable and valuable. In this handout I have presented information that will be very helpful when redoing the sample.

You have seen how the architecture was idealized and separated into 3 modules, then a summary of relevant information to implement this on your project. Please also review the slide deck (Power Point file) and the code project sample, for web site with ASP.NET, for Android with Java and for Civil 3D with .NET.

I'm also glad you are considering integrate your Civil 3D development with Cloud and Mobile, we believe the best approach is use what's best for each scenario: with our extremely connect technology, the cloud is everywhere and accessible almost any time with mobile devices. Take advantage of that! Good luck!