



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΣΥΣΤΗΜΑΤΑ ΓΕΩΓΡΑΦΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ

Εισαγωγή στο Microsoft Virtual Earth

Δρ. Ηλίας Φρέντζος

Όπως περιγράφεται από την Microsoft στην αρχική σελίδα του αντίστοιχου Software Development Kit (<http://msdn.microsoft.com/en-us/virtualearth/default.aspx>), το Microsoft Virtual Earth είναι ένα σύνολο υπηρεσιών βασισμένων στο διαδίκτυο που δίνουν τη δυνατότητα στους χρήστες τους – προγραμματιστές να δώσουν στις εφαρμογές τους ένα ολοκληρωμένο τρόπο θέασης της χωρικής υπόστασης των δεδομένων τους και να αναπτύξουν σελίδες που αλληλεπιδρούν με τους τελικούς χρήστες μέσα από ένα πλούσιο γεω-χωρικό περιβάλλον.

Το βασικό πλεονέκτημα του Microsoft Virtual Earth, όπως και του Google Maps, είναι ότι παρέχουν ένα πλήρες μοντέλο δεδομένων που μπορεί να διαχειριστεί / οπτικοποιήσει χωρικά δεδομένα, κάτω από ένα ενιαίο, πλούσιο χωρικό υπόβαθρο που αποτελείται από διανυσματικά δεδομένα (δρόμοι, ακτογραμμές, διοικητικά όρια), ψηφιακά μοντέλα εδάφους, και ορθοανηγμένα raster δεδομένα όπως δορυφορικοί ορθοφωτοχάρτες (aerial και hybrid view) και αεροφωτογραφίες μεγάλης κλίμακας (bird's eye). Το μοντέλο δεδομένων που παρέχεται μπορεί να χρησιμοποιηθεί για να διαχειριστεί τον τρόπο θέασης του χάρτη, τον τρόπο που εμφανίζονται τα επιπλέον δεδομένα που «τοποθετεί» ο χρήστης επάνω στο διατιθέμενο υπόβαθρο, καθώς και να εκτελέσει πιο προηγμένες λειτουργίες, όπως γεωκωδικοποίηση (δηλαδή υπολογισμός των συντεταγμένων μίας γεωγραφικής διεύθυνσης), αντίστροφη γεωκωδικοποίηση, δρομολόγηση κ.α.

Ένα γρήγορο και απλό παράδειγμα των δυνατοτήτων που παρέχονται από το Microsoft Virtual Earth SDK μπορεί να βρεθεί παρακάτω. Ο κώδικας που παρατίθεται παρακάτω τοποθετεί επάνω στο χάρτη τρεις εκκλησίες, και δημιουργεί ένα τρίγωνο με τις κορυφές του επάνω σε αυτές.

Ο κώδικας μπορεί να βρεθεί και στο URL: http://isl.cs.unipi.gr/courses/gis/lab/data/ve_lab_data.zip

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <script type="text/javascript"
src="http://dev.virtualearth.net/mapcontrol/mapcontrol.ashx?v=6.1"></script>
    <script type="text/javascript">
      var map = null;

      function GetMap()
      {
        //Create a new instance of VEMap, and specify the control
        //on the Web page that will contain the map. For example,
        //it is most common to display the map in a DIV control.
        map = new VEMap('myMap');
        map.LoadMap();
        // Display map in Initial zoom
        DoInitZoom();
      }

      function DoInitZoom()
      {
        // Create a new VELatLong for the 1st church
        var Church1LL=new VELatLong(38.355252,22.12446)
        // Create a new shape for the 1st church and add attributes
        var shape = new VEShape(VEShapeType.Pushpin, Church1LL);
        shape.SetTitle('Εκκλησία');
        shape.SetDescription('Προφήτης Ηλίας');
        map.AddShape(shape);

        // Create a new VELatLong for the 2nd church
        var Church2LL=new VELatLong(38.347508,22.156914)
        // Create a new shape for the 2nd church and add attributes
        var shape = new VEShape(VEShapeType.Pushpin, Church2LL);
        shape.SetTitle('Εκκλησία');
        shape.SetDescription('Άγιος Νικόλαος');
        map.AddShape(shape);

        // Create a new VELatLong for the 3rd church
        var Church3LL=new VELatLong(38.375162,22.141762)
        // Create a new shape for the 3rd church and add attributes
        var shape = new VEShape(VEShapeType.Pushpin, Church3LL);
        shape.SetTitle('Εκκλησία');
        shape.SetDescription('Άγιος Νεκτάριος');
        map.AddShape(shape);

        // create a new polygon shape to display a triangle
        // connecting the three churches
        var polshape = new VEShape(VEShapeType.Polygon,
                                   [Church1LL,Church2LL,Church3LL]);
        polshape.SetTitle='Απλό πολύγωνο';
        polshape.SetDescription('Απλό πολύγωνο');
        map.AddShape(polshape);

        // calculate initial zoom position as the middle of the three churches
        var clat = (Church1LL.Latitude+Church2LL.Latitude+Church3LL.Latitude)/3;
        var clon = (Church1LL.Longitude+
                    Church2LL.Longitude+Church3LL.Longitude)/3;
```

```
    var zoom = 12;
    map.SetCenterAndZoom(new VELatLong(clat, clon), zoom);
    // set the map style..
    map.SetMapStyle(VEMapStyle.Shaded);
}
</script>
</head>
<body onload="GetMap();" >
    <div id='myMap' style="position:relative; width:1000px;
height:700px;"></div>
</body>
</html>
```