CREATING A WEBPAGE WITH AN EMBEDED INTERACTIVE WEB MAP FOR DISPLAYING EMERGENCY INFORMATION

Zachary Strack

Master of Science in Geographic Information Science (MSGISci)

Department of Geography, California State University, Long Beach

Introduction

Emergencies on any college campus can become chaotic due to the large number of people who are affected. This is why campuses like California State University, Long Beach (CSULB) have procedures and plans in place to create order during emergency events. Even though procedures and plans are established, a large number of people may not be familiar with the campus or its emergency procedures. The purpose of this project is to take the existing static CSULB emergency map and create a proof of concept webpage with an embedded interactive emergency map.

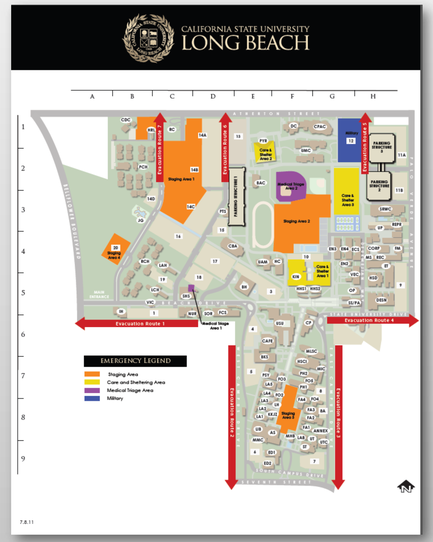

Figure 1. *Image of existing CSULB static emergency map.*

Data and Data Sources

The emergency information used in the interactive web map was provided by CSULB in the form a static PDF map. The static map contained emergency routes and various areas including care and sheltering, staging, medical, and military areas. The map also contained roads, buildings, and other structures for reference. A file geodatabase containing a basemap feature class of the CSULB campus was provided by the CSULB geography department to be used for processing of the PDF map features into feature classes using ArcMap.

Table 1. *List of data and data sources used in the project*

Dataset	Source
CSULB Emergency Map PDF	California State University, Long Beach
CSULB Basemap feature class	California State University, Long Beach
Aerial Imagery	Esri World Imagery

Methodology

The first step completed was heads-up digitizing the PDF features and their attributes into feature classes. Being a static map, the PDF did not have a geospatial reference, so the emergency map was georeferenced using ArcMap 10.3.1 Once the static map image was georeferenced, feature classes were created and loaded into CSULB's enterprise geodatabase.

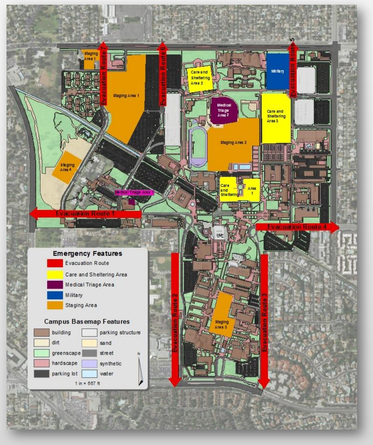

Figure 2. *CSULB emergency features and campus basemap shown in ArcMap*

Evacuation routes and staging areas were the two feature classes created. A ArcMap Map Document (MXD) file was created that held symbology information for each feature layer. Once the feature classes were published as a feature service using ArcGIS for Server they were displayed on the webpage with the symbology set in the MXD. Feature services also allowed for querying of the datasets through Esri's ArcGIS for JavaScript API making it possible to include this functionality in the webpage code. The returned queried information could then be displayed in a popup page element.

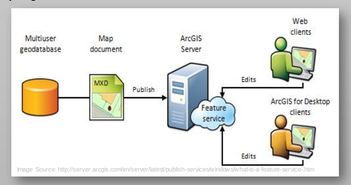

Figure 3. *A model of how webpages can access feature attributes through feature services.*

To build the web page that would incorporate the interactive web map, w3schools.com was the main resource for free online tutorials and references for Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript.

```
<!doctype html>

<html>

<head>
    <title>CSULB Emergency Map</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="stylesheet.css">
    <link rel="stylesheet" href="http://js.arcgis.com/3.12/esri/css/esri.css">
    <script src="http://js.arcgis.com/3.12/"></script>

    <script type="text/javascript">
    </script>

</head>

<body onload="initialize()">


<div id="contain_header">
  <div id="contain_header_setwidth">
  <div id="contain_witeid">
     <div id="siteid_content"><a href="http://web.csulb.edu"><img
     src="http://web.csulb.edu/minr/depot/assets/UT/hdr_blk_id.gif"
     alt="California State University, Long Beach"
     title="California State University, Long Beach" height="50" width="287" /></a><span>California State University, Long Beach</span></div>
    </div>


<div id="title" onclick="titlemsjs()">
California State University-Long Beach Emergency Management Map
</div>

<div id="mapcontainer">
</div>

</body>
</html>
```

Figure 4. *Example of a simple HTML document*

Results

Figure 2 shows the emergency evacuation routes and the various staging areas on top of the CSULB basemap in ArcMap 10.3.1. Figure 5 shows the webpage with the embedded interactive web map after an evacuation route has been clicked.

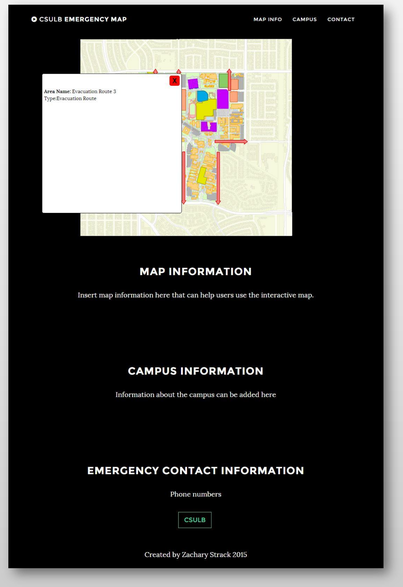

Figure 5. *Image of webpage in a browser showing popup with feature attributes after it has been clicked*

```
//SET GEOMETRY SERVICE AND INITIAL MAP EXTENT

    initialeXtent = new Extent(-13150000, 3999000, -13146800, 4000000, SpatialReference({wkid:3857}));

//CREATE MAP

    map = new Map("mapcontainer", {  autoResize:true,   extent:initialeXtent,   logo:false,   slider:false,
    basemap:"",   showAttribution:false});
    mapcenterpoint = new Point([-13122880,4012924],new SpatialReference({ wkid:3857 }));
    basemap = new
    ArcGISTiledMapServiceLayer('http://134.139.94.24:6080/arcgis/rest/services/CSULB/basemap4/MapServer');
    buildings  = new
    FeatureLayer('http://134.139.94.24:6080/arcgis/rest/services/CSULB/buildings/FeatureServer/0');
    stagingareas  = new
    FeatureLayer('http://134.139.94.24:6080/arcgis/rest/services/CSULB/Emergency_Response_Staging_Areas/FeatureSe
    ver/0');
    evacroutes  = new
    FeatureLayer('http://134.139.94.24:6080/arcgis/rest/services/CSULB/evacuationroutes/FeatureServer/0');

    map.addLayer(basemap);
    map.addLayer(Buildings);
    map.addLayer(stagingareas);
    map.addLayer(evacroutes);

    buildings.on("mouse-over",function (){map.setMapCursor("pointer")});
    buildings.on("mouse-out", function (){map.setMapCursor("default")});
    buildingsinfotemplate = new InfoTemplate("buildings", "Names: ${name}<br> Abbreviated Name:
    ${short_name}<br><br><img src='${media_dir}${photo_file}' />");
    buildingsquerytask = new
    QueryTask('http://134.139.94.24:6080/arcgis/rest/services/CSULB/buildings/FeatureServer/0');
    buildings.on("click", function (evt){ layer='buildings'; selectinfotemplate = buildingsinfotemplate;
    selectquerytask = buildingsquerytask; x = evt.clientX;  y = evt.clientY;
    showinfopopup(x,y,layer,selectinfotemplate,selectquerytask)});

    stagingareas.on("mouse-over",function (){map.setMapCursor("pointer")});
    stagingareas.on("mouse-out", function (){map.setMapCursor("default")});
    stagingareasinfotemplate = new InfoTemplate("stagingareas", "Area Name: ${area_name}");
    stagingareasquerytask = new
    QueryTask('http://134.139.94.24:6080/arcgis/rest/services/CSULB/Emergency_Response_Staging_Areas/FeatureServe
    0');
    stagingareas.on("click", function (evt) {layer = 'stagingareas';  selectinfotemplate =
    stagingareasinfotemplate; selectquerytask = stagingareasquerytask; x = evt.clientX; y = evt.clientY;
    showinfopopup(x,y,layer,selectinfotemplate,selectquerytask)});

    evacroutes.on("mouse-over", function (){ map.setMapCursor("pointer")});
    evacroutes.on("mouse-out", function () {map.setMapCursor("default")});
    evacrouteinfotemplate = new InfoTemplate("stagingareas", "<b>Area Name:</b> ${area_name}<br>
    Type:${area_type}");
    evacroutequerytask = new
    QueryTask('http://134.139.94.24:6080/arcgis/rest/services/CSULB/evacuationroutes/FeatureServer/0');
    evacroutes.on("click", function (evt) {layer = 'evacroutes'; selectinfotemplate = evacrouteinfotemplate;
    selectquerytask = evacroutesquerytask; x = evt.clientX; y = evt.clientY;
    showinfopopup(x,y,layer,selectinfotemplate,selectquerytask)});

    function printerror(err){
        alert(err);
    }

    function showfeatureinfo(results){

        var featureselectgraphic = results.features[0];
        featureselectgraphic.setInfoTemplate(selectinfotemplate);
        var info = featureselectgraphic.getContent();
        document.getElementById('popupwindowcontent').innerHTML = info;
    }

    function showinfopopup(x,y,layer,selectinfotemplate,selectquerytask){
            aquery = new Query();
            sp = new ScreenPoint(x,y);
            map.y = map.toMap(sp);
            newmapxy = new Extent(mapxy.x-5,mapxy.y-5,mapxy.x+5,mapxy.y+5, SpatialReference({wkid:3857}));
            aquery.geometry = newmapxy;
            aquery.outFields = ["*"];
            document.getElementById("popupwindow").style.display="initial";
            selectquerytask.execute(aquery,showfeatureinfo,printerror);

    }

    document.getElementById("popupwindow").style.display="none";

});
```

Figure 6. *Sample of JavaScript Code in HTML document*

Timeline

Nearly all major milestones were completed by their expected dates with the exception of completion of the webpage and web map. This was due to difficulties debugging errors in the JavaScript code. Though the issues extended the completion time, both webpage and web map were completed.

Task	Start	Finish
Learn HTML, CSS, JavaScript	29-Jun	10-Jul
Complete Web Page Layout and Design	29-Jun	6-Jul
Incorporate Web Map into Webpage	13-Jul	17-Jul
Incorporate Map Features/ Write Methods	6-Jul	25-Jul
Finalize Project Web Application	6-Jul	14-Aug
Results Section	25-Jul	1-Aug
Discussion Section	1-Aug	8-Aug
Abstract & Introduction	8-Aug	10-Aug
Poster	10-Aug	12-Aug
Applied Research Project Document	6-Jul	14-Aug
Final Project Presentation	12-Aug	15-Aug

Table 2. Expected dates of completion for major project milestones

Discussion

The free open source tutorials and examples provided by W3schools.com allowed for examples of code to be copied and were used to learn how to create HTML, CSS, and JavaScript documents in a short amount of time.

JavaScript was the most difficult aspect of the project. The main cause of difficulties was debugging errors in the JavaScript code. Some issues would take over a week to solve which extended expected completion time and often required the assistance of CSULB faculty.

The webpage is not a finished product to be distributed and utilized. The users can click on features in the web map and additional information will show in a popup, but the attributes available are limited. With a framework now established, additional attribute information can be added to feature classes and be incorporated in the web map.

Conclusion

While the web map created through this project is not a finished product, it can serve as an effective framework that can be further developed into a functional interactive emergency response web map to be deployed at CSULB.

In creating a webpage with an interactive web map, a good understanding of JavaScript is necessary. Taking JavaScript courses, especially those incorporating the Esri ArcGIS for Javascript API, would be of significant benefit.

In the development of the webpage, the skills necessary to develop a working webpage with little initial knowledge proved to be possible with the assistance of w3schools. The site allows people to learn HTML, CSS, and JavaScript at their own pace and obtain basic web programming knowledge.

Future work on this project could include further developing the feature attributes so they can be added to information that appears in the popup element. The inclusion of feature labels and legend for the web map would help users interpret and utilize the map more effectively. Finally, sections of the web page were left reserved and that could be filled with additional campus emergency information.

Submitted in partial fulfillment of the requirements of the Masters of Science in Geographic Information Science(MSGISci), August 15, 2015.

For additional information please contact: Zachary Strack at zstra399@gmail.com