JavaScript
Map Overlays (Markers)

Waterford Institute of Technology

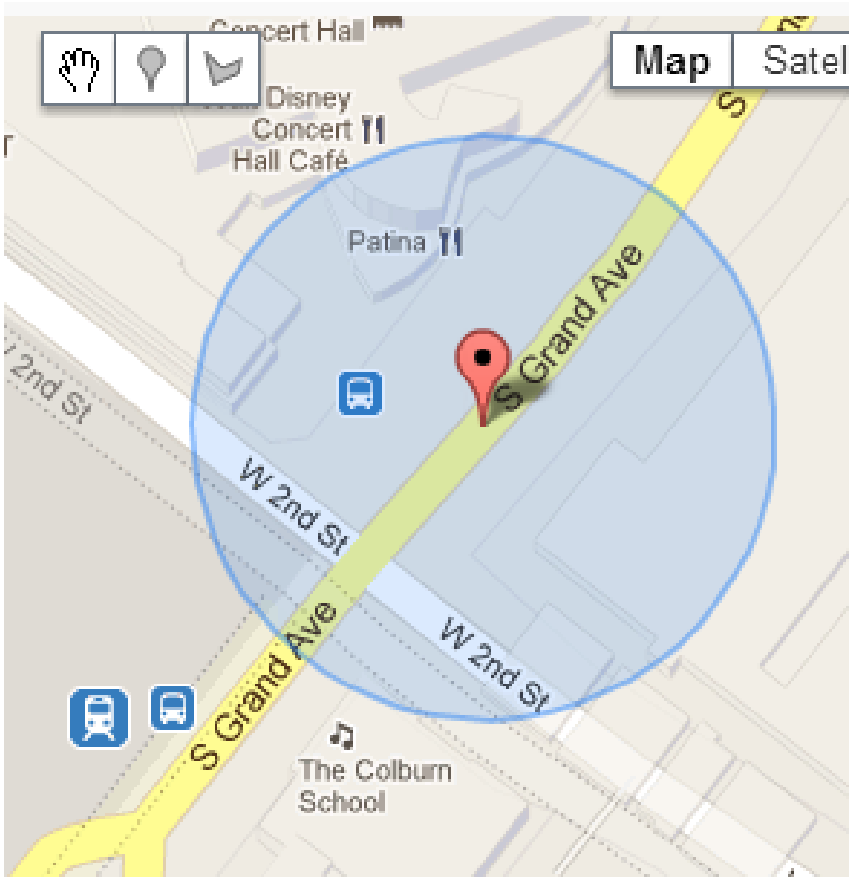
May 10, 2016

John Fitzgerald

Using JavaScript

Drawing on the map

- Add objects (overlays) to map to designate:
  - points
  - lines
  - areas
  - markers



Circle overlay

circledata.html

Begin with simple html file, using scripts:

- jQuery
- Google Map
- Bespoke *circledata.js*

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
  </head>
  <body>
    <div id="map-canvas"></div>
    <br>
    <script src="js/jquery-2.1.1.js"></script>
    <script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></script>
    <script src="js/circledata.js"></script>
  </body>
</html>
```

Map container

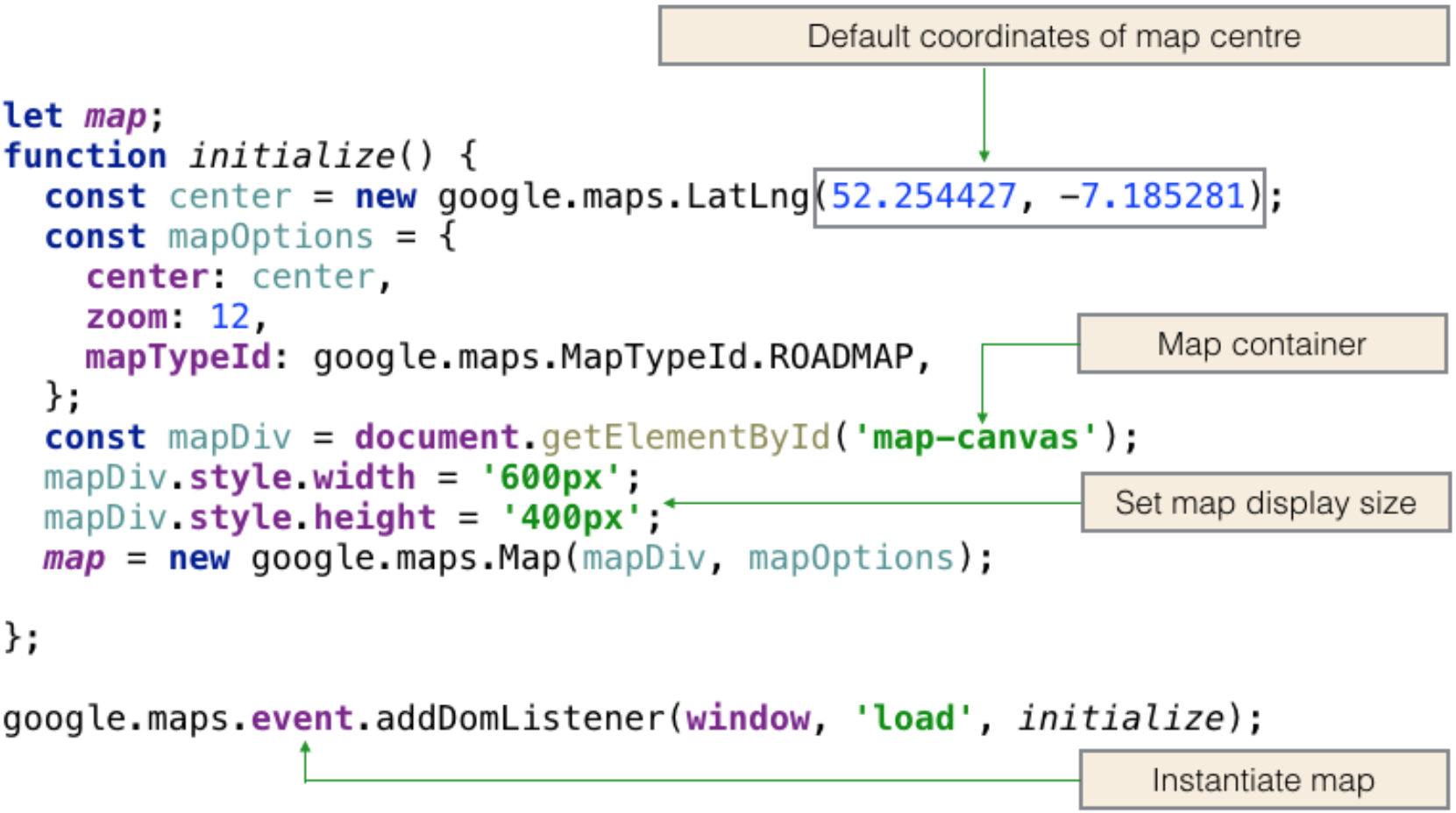
jQuery file

Bespoke JavaScript file

Google Map

Circle overlay

Baseline *circledata.js*




```
let map;
function initialize() {
  const center = new google.maps.LatLng(52.254427, -7.185281);
  const mapOptions = {
    center: center,
    zoom: 12,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
  };
  const mapDiv = document.getElementById('map-canvas');
  mapDiv.style.width = '600px';
  mapDiv.style.height = '400px';
  map = new google.maps.Map(mapDiv, mapOptions);

};

google.maps.event.addDomListener(window, 'load', initialize);
```

Circle overlay

Scripts

Locate scripts in folder *WebContent/js*:

- jQuery
- Bespoke *circledata.js*
- Launch *circledata.html* in browser

Circle overlay

Introduce circle overlay

Define circle object in *circledata.js*:

- Draggable
- Resizable

```
let circle;
let map;
function initialize() {

  map = new google.maps.Map(mapDiv, mapOptions);
  const initRadius = 2000;
  circle = new google.maps.Circle({
    center: center,
    radius: initRadius,
    strokeColor: '#0000FF',
    strokeOpacity: 0.4,
    strokeWeight: 1,
    fillColor: '#0000FF',
    fillOpacity: 0.4,
    draggable: true,
  });

  // User may vary radius by dragging anchor point
  circle.setEditable(true);
  circle.setMap(map);
};

google.maps.event.addDomListener(window, 'load', initialize);
```

Set initial radius

Configure circle

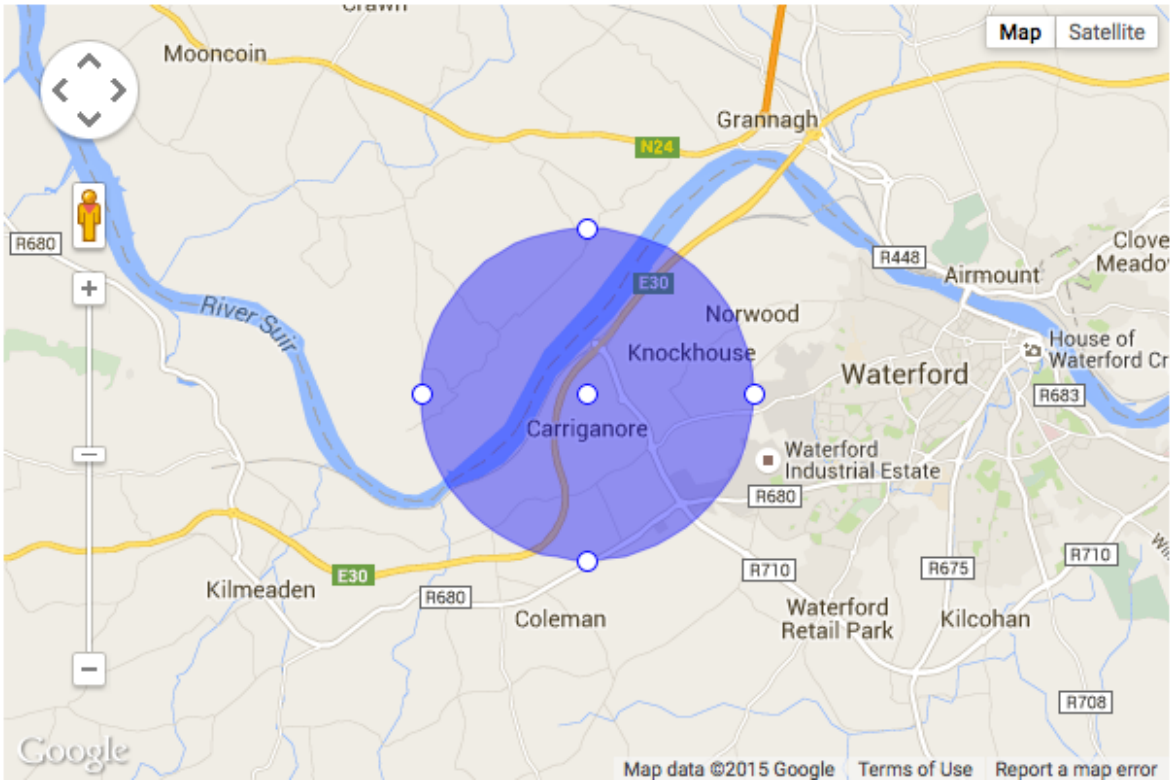
Circle may be dragged and dropped

Circle may be resized

Circle overlay

Display circle overlay

Refresh browser to display modified map.



Circle overlay

Display circle overlay

Check overlay behaves as expected:

- Change radius.
- Drag to new location.
- Change map zoom level.

To resize circle:
Hover over handle.
Hold down mouse button.
Drag handle.

To move circle:
Hover over interior circle.
Hold down mouse button.
Drag circle.

Circle overlay

Capture data

Additional html & script function

```
<div id="circledata">
  <input type="text" id="radius" size="10">
  <input type="text" id="latcenter" size="25">
  <input type="text" id="lngcenter" size="25">
</div>
```

Display the circle parameters

```
function circleData() {
    const center = circle.getCenter();
    const latcenter = center.lat().toString();
    const lngcenter = center.lng().toString();
    var radius = circle.getRadius().toString();
    $("#radius").val(radius);
    $("#latcenter").val(latcenter);
    $("#lngcenter").val(lngcenter);
}
```

Obtain the centre and radius of circle presently rendered

Display the circle parameters on the web page

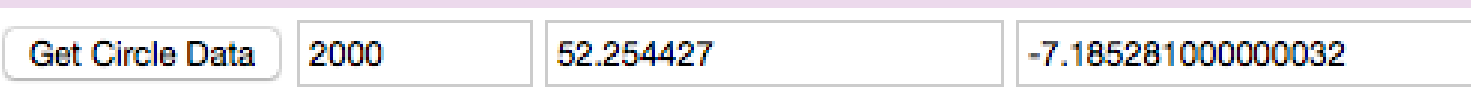
```
<button type="button" onclick="circleData()">Get Circle Data</button>
```

Circle overlay

Capture data

Add button and click to display circle parameters.

```
<button type="button" onclick="circleData()">Get Circle Data</button>
```



Marker overlay

Introduce additional overlay - a marker

Function to create map marker object (overlay)

```
function addMarker(map, center)
{
  // create a marker
  marker = new google.maps.Marker({
    map: map,
    position: center,
    title: 'Drag and drop on your property!',
    draggable: true,
  });
  marker.setMap(map);
};
```

Marker overlay

Introduce additional overlay - a marker

Capture and display marker position

```
<!-- Marker overlay -->
<div id="markerdata">
  <input type="text" id="geolocation" readonly class="input-medium" size="25">
</div>
```

Display marker coords in this readonly text box.

```
function markerData() {
  let latLng = marker.getPosition();
  let lat = latLng.lat().toString().substring(0, 10);
  let lon = latLng.lng().toString().substring(0, 10);
  let latlong = lat + ',' + lon;

  // render lat long in geolocation text box on html page
  $('#geolocation').val(latlong);
}
```

Obtain the centre and radius of circle presently rendered

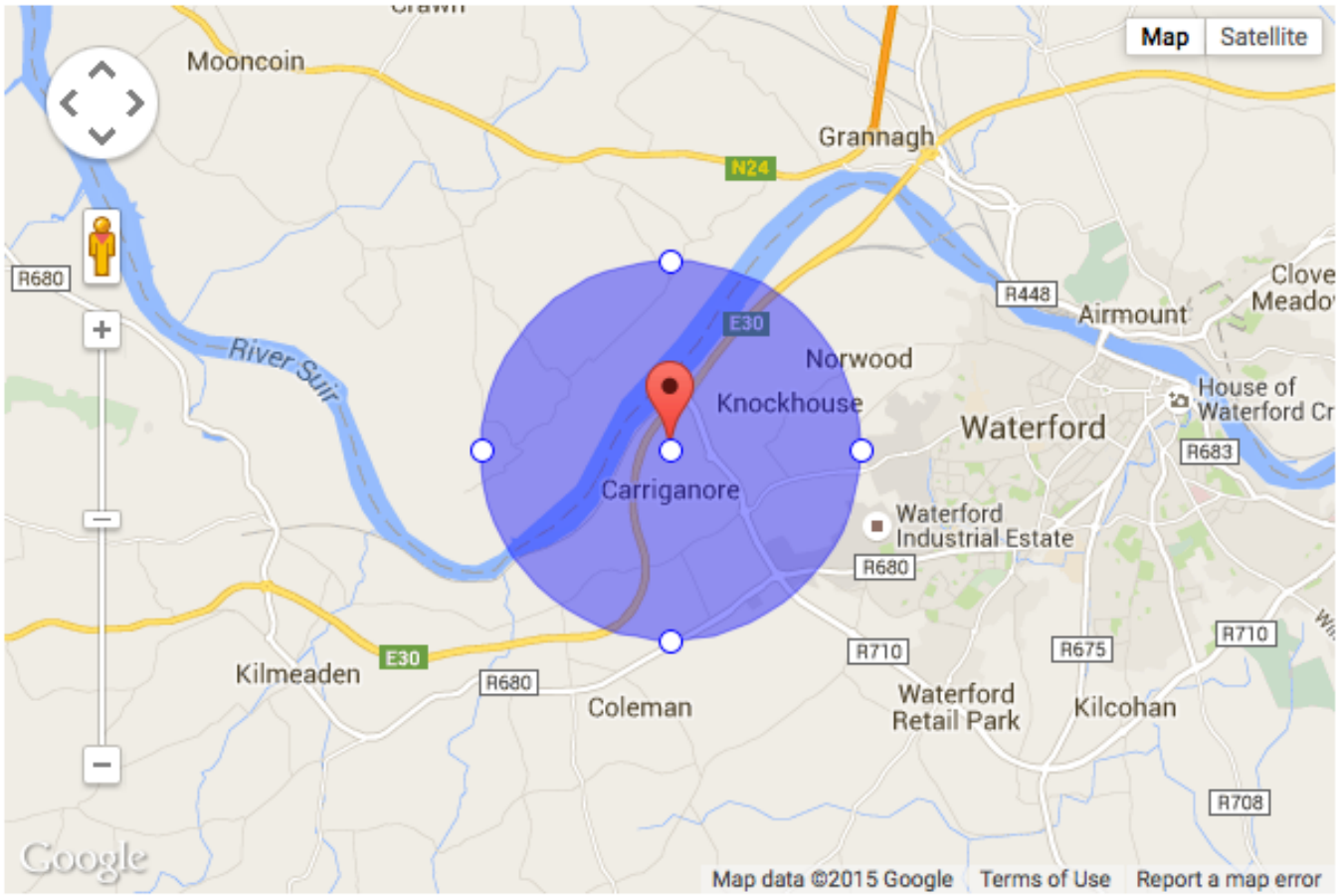
Marker overlay

Introduce additional overlay - a marker

To display marker:

```
let marker; //place at begininning circledata.js
```

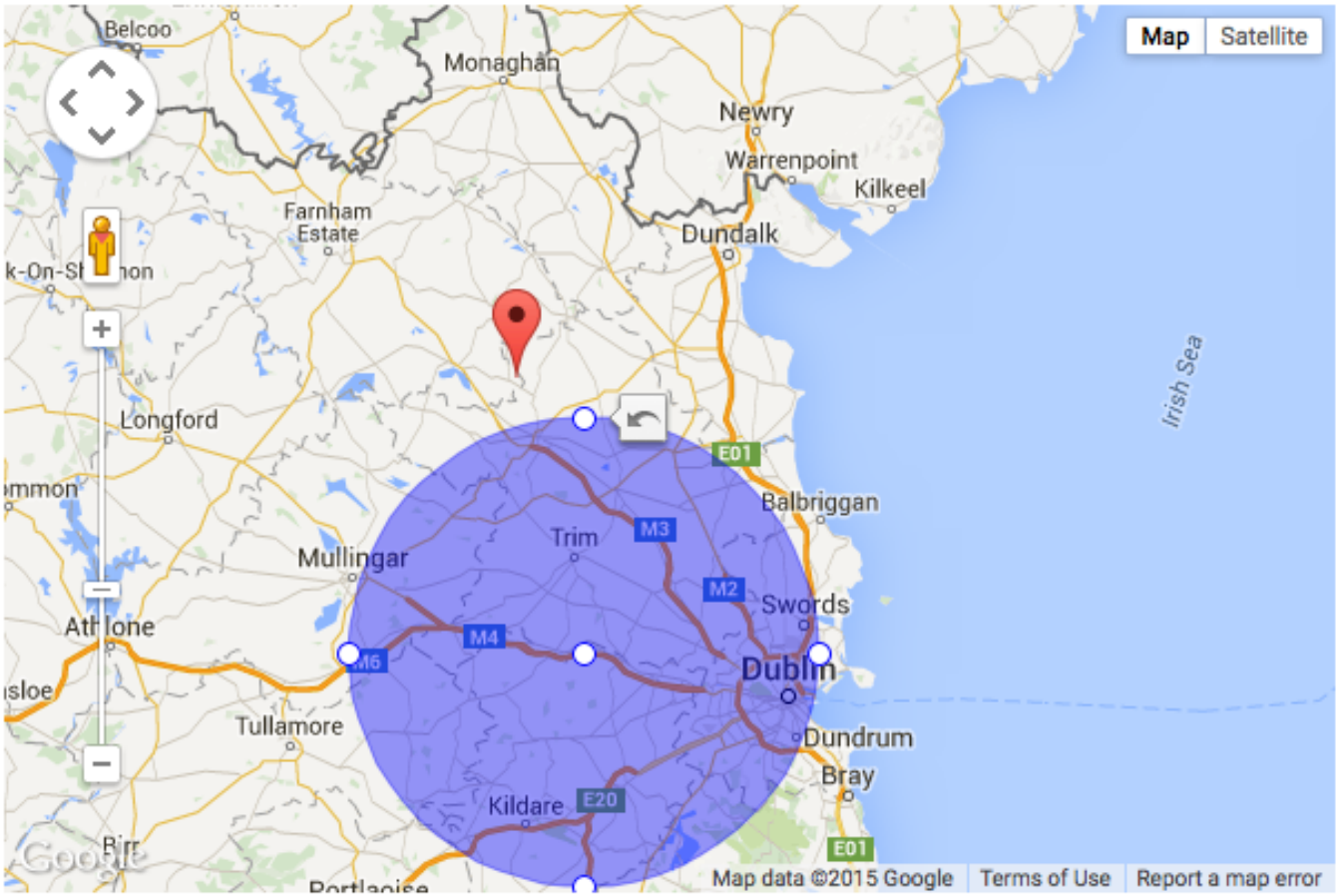
```
addMarker(map, center); //place end initialize()in circledata.js
```



Marker overlay

Introduce additional overlay - a marker

Marker and Circle overlays behave independently.



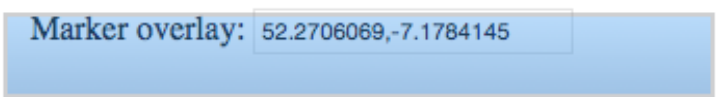
Marker overlay

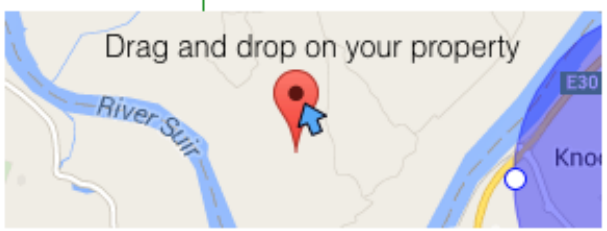
Listener added

Invoke `markerListener()` at end `initialize()`.

```
function markerListener()  {
  google.maps.event.addListener(marker, 'dragend', function () {
        markerData();
    });
}
```

Updates Marker overlay coordinates when drag ends





Overlays

Is marker located within circle?

Invoke `circleListener()` at end `initialize()`.

```
function circleListener()  {
    google.maps.event.addListener(circle, 'center_changed', function () {
        circleData();
    });
    google.maps.event.addListener(circle, 'radius_changed', function () {
        circleData();
    });
}
```

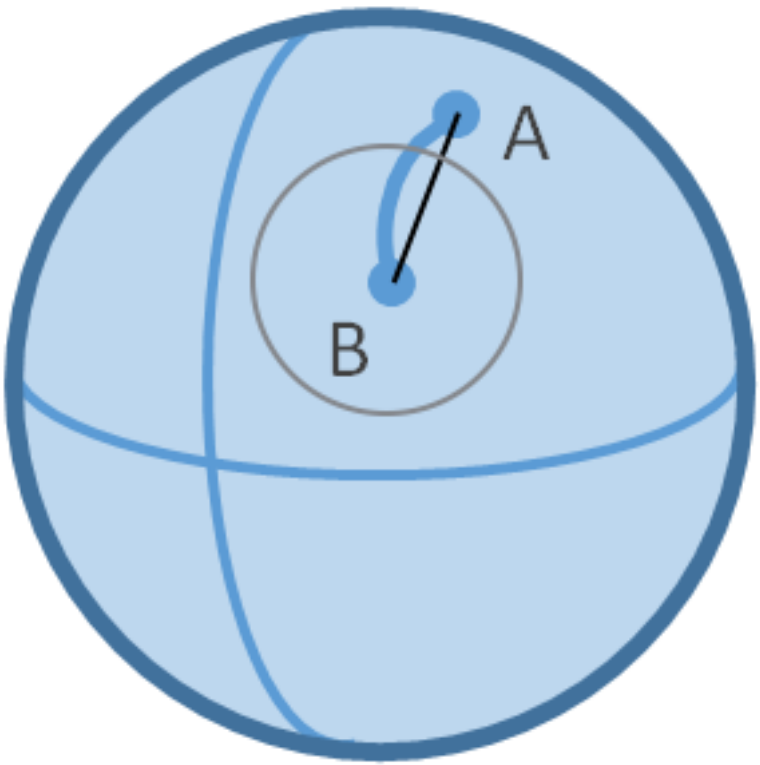
Listen for both change in position (center_changed) and change in size (radius_changed).

Overlays

Is marker located within circle?

Evaluate distance between centre circle (B) and marker (A):

- If distance A-B exceeds radius then point falls outside circle.
- Note necessity of measuring distance along surface of sphere.



Overlays

Is marker located within circle?

- Assume radius is measured along surface sphere.
- Use Google library method *computeDistanceBetween*.
- This requires script addition `&libraries=geometry`

```
<script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false&libraries=geometry"></script>

function isMarkerInCircle()
{
    const ccenter = circle.getCenter();
    const mcenter = marker.getPosition();
    const distance = google.maps.geometry.spherical.computeDistanceBetween(ccenter, mcenter);
    const is_in_circle = distance <= circle.getRadius();
    if(is_in_circle)
        return true;
    else
        return false;
};
```