**Example:** (**A Hexagon-Triangle Hinge)** In the diagram below, Q is the center of a regular hexagon with a vertex B at which it touches an equilateral triangle with center R. Let P be the midpoint of AB. Show that $∠RPQ$ is a right angle.

(Source: Konhauser, J. D. E., Velleman, D., Wagon, S. *Which Way Did the Bicycle Go? … and Other Intriguing Mathematical Mysteries*. The Mathematical association of America, Dolciani Mathematical Expositions – No. 18. 1996.)



The solution to the example, i.e. proving the given statement, is left to the reader. Here we will deal only with the programming of the illustrative dynamic figure.

Programming:

First, we define the Bounding box as follows:

var board = JXG.JSXGraph.initBoard('jxgbox', {boundingbox: [-4, 3, 4, -3]});

There will be eleven points in the diagram, namely vertices of polygons, their centers and a midpoint. We set them all to be the same size, specifically 1, as follows:

board.options.point.size = 1;

*Note:* In this way we can also set other common attributes of objects in the board, for example the appearance of points using the command board.options.point.face = '[]'; etc.

To obtain the regular hexagon we first construct points A and B, its future two adjacent vertices, and then color them orange to highlight them as movers.

var A = board.create('point' , [-1.2,-2], {color: 'orange'});

var B = board.create('point', [0.25,-0.5], {color: 'orange'});

Then, using these points as the input parameters to the ‘regularpolygon’ (https://jsxgraph.org/docs/symbols/RegularPolygon.html) object, we create the desired hexagon.

var hexagon = board.create('regularpolygon', [A,B, 6]);

The center of the regular polygon is the center of its circumcircle, therefore we have to identify at least three vertices of the hexagon to determine its center. We chose D, the fourth vertex from A (index number is 3 because the software starts to count from 0)

var D = hexagon.vertices[3];

to determine the center Q of the regular hexagon as the center of the circumcircle to the triangle ABD:

var Q = board.create('circumcenter', [A, B, D], {name:'Q'});

In an analogous way, we then create an equilateral triangle BGH with center N:

var G = board.create('point', [3,-2], {name: 'G', color: 'orange'});

var rtr = board.create('regularpolygon', [B, G, 3]);

var R = board.create('circumcenter', [B, G, H], {name: 'R'});

Then, we add the triangle AGB, the midpoint P of its side AB and the lines PQ and PR, which we have to prove are always perpendicular:

var tr = board.create('polygon', [A, G, B], {color: 'pink'});

var P = board.create('midpoint', [A, G], {name: 'P'});

var H = rtr.vertices[2];

var q = board.create('line', [P, Q], {name: 'q', withLabel: true});

var r = board.create('line', [P, R], {name: 'r', withLabel: true});

Finally, we can identify the angle RPQ, the rightness of which we have to prove. JSXGraph recognizes it as a right angle, so it marks it accordingly, as can be seen in figure above:

var angle = board.create('angle', [R, P, Q], {radius: 0.4, color: 'red', fillOpacity: 0});

If this symbolic expression of the right angle is not enough for us, we can enter the code to display the size of the respective angle:

board.create('text', [-3, -3,

 function () {return '&theta;\_1 = ' + (arc2.Value() \* 180 /Math.PI).toFixed(1) + '&deg;';}
]);

The task solution code is now complete.