CLASS YUIDOC EXAMPLE (http://students.cs.byu.edu/~cs340ta/winter2014/notes/12-YUIDoc/Rodham/js/Point3D.js)

/\*\*  
 The Point3D class represents a three-dimensional cartesian point   
 <pre>   
 Domain:  
 X: the point's x coordinate, Number  
 Y: the point's y coordinate, Number  
 Z: the point's z coordinate, Number  
   
 Invariants:  
 INVARIANT: Something that is always true about Point3D objects  
 INVARIANT: Something else that is always true about Point3D objects  
   
 Constructor Specification:  
 PRE: !isNaN(xValue)  
 PRE: !isNaN(yValue)  
 PRE: !isNaN(zValue)  
 POST: getX() == xValue  
 POST: getY() == yValue  
 POST: getZ() == zValue  
 </pre>  
  
 @class Point3D  
 @extends Point  
 @constructor  
   
 @param {Number} xValue The point's x coordinate  
 @param {Number} yValue The point's y coordinate  
 @param {Number} zValue The point's z coordinate  
 \*/

METHOD IN CLASS YUIDOC EXAMPLE (http://students.cs.byu.edu/~cs340ta/winter2014/notes/12-YUIDoc/Rodham/js/Point3D.js)

/\*\*  
 Scales the point's coordinates by the specified factor   
 <pre>  
 PRE: !isNaN(factor)  
 POST: getX() == factor X OLD(getX())  
 POST: getY() == factor X OLD(getY())  
 POST: getZ() == factor X OLD(getZ())  
 </pre>  
   
 @method scale  
 @param {Number} factor The scaling factor  
 \*/

//////////////// ACTUAL CODE STARTS HERE ////////////////

/\*\*

The Map Class represents the board and holds all of the hexagons

<pre>

TODO…

</pre>

\*/

function canPlaceRoad() {

//TODO

}

/\*\*  
 Place the road in a specific position   
 <pre>  
 PRE: got validation from canPlaceRoad()

POST:

</pre>  
   
 @method placeRoad  
 @param {location} where the road will be placed

@param {playerID} ID of the player who is placing the road

@return {boolean} returns true if road is placed. false otherwise

\*/

function placeRoad(location, playerID) {

//TODO

}

function canPlaceSettlement() {

}

/\*\*  
 Place the settlement in a specific position   
 <pre>  
 PRE: …

POST: ...  
 POST: ...  
 POST: ...  
 </pre>  
   
 @method placeSettlement  
 @param {location} where the settlement will be placed

@param {playerID} ID of the player who is placing the settlement

@return {boolean} returns true if settlement is placed. false otherwise

\*/

function placeSettlement() {

//TODO

}

function canPlaceCity() {

}

/\*\*  
 Place the city in a specific position  
 <pre>  
 PRE: checked if can place city canPlaceCity()

POST:   
 </pre>  
   
 @method placeCity  
 @param {location} where the city will be placed

@param {playerID} ID of the player who is placing the city

@return {boolean} returns true if city is placed. false otherwise

\*/

function placeCity() {

//TODO

}

/\*\*

Move the robber to a specified position and set robber’s position

<pre>

</pre>

@method moveRobber

@param {location} where the robber is moving to

@return {boolean} returns true if the robber is moved. false otherwise

\*/

function moveRobber(){

//todo

}