

```

40     lines = csv.reader(file, delimiter=' ')
41     correctClasses = [ int(line[2]) for line in lines if line[2] ]
42     return correctClasses
43
44
45
46 # NOTE: - class definitions: ***** clas
47
48 class XYPoint():
49     def __init__(self, listPoint):
50         # self.rawValue = listPoint
51         self.x, self.y = listPoint[0], listPoint[1]
52
53     zero = nu.array([0, 0])
54
55     @property
56     def rawValue(self):
57         return [self.x, self.y]
58
59     @property
60     def ndarrayValue(self):
61         return nu.array(self.rawValue).reshape(2,1)
62
63     def distanceFrom(self, point, weight=nu.diag([1,1]) ):
64         return nu.linalg.norm( nu.dot(weight, (self.ndarrayValue - point.reshape(2,1))
65
66     def distanceFromXYPoint(self, point):
67         return ma.sqrt( (self.rawValue[0]-point.rawValue[0])**2 + (self.rawValue[1]-po
68
69     def similarityToPoint(self, point):
70         inner = nu.dot(self.ndarrayValue.reshape(1,2), point.reshape(2,1))
71         distanceProduct = self.distanceFrom(XYPoint.zero) * nu.linalg.norm(point)
72         angle = ma.acos(inner / distanceProduct)
73         return angle
74
75     def move(self, direction):
76         global STEP, validDirections
77         if direction in validDirections:
78             return XYPoint([ self.x + validDirections[direction][0], self.y + validDir
79         else:
80             return None
81
82
83 class ClassData():
84     def __init__(self, file):

```