```
# Creates a Magic Square of odd size bewteen 3 and 15
# Limited size for space considertions
import random

def squareMaker(size):

    magic_square = [[0 for x in range(size)] for y in range(size)]

    #intitialize counts

    incount = 0
    row_out = 0
    col_out = 0
    both_out = 0
    blocked = 0

    row = 0
    col = size // 2

    magic_square [row][col] = 1

    for num in range (2 , size**2 + 1) :

        col = col + 1
        row = row - 1

        if col >= 0 and col <= size - 1 and row >=0 and row <= size -1:
          if magic_square [row][col] == 0:
             magic_square [row][col] = num
             incount = incount + 1
          else:
             row = row + 2
             col = col - 1
             magic_square [row][col] = num
             blocked = blocked + 1
        elif row < 0 and col >=0 and col <= size -1:
          row = size - 1
          magic_square [row][col] = num
          row_out = row_out +1
        elif col > size - 1 and row >= 0 and row <= size - 1:
          col = 0
          magic_square [row][col] = num
          col_out = col_out + 1
        elif row < 0 and col > size - 1:
          row = row + 2
          col = col - 1
          magic_square [row][col] = num
          both_out = both_out +1
    print ("incount: ",incount)
    print ("blocked: ",blocked)
    print ("row out: ",row_out)
    print ("col out: ",col_out)
    print ("both out: ",both_out)


    sum = size * ((size**2)//2 +1)
    print  ("\n\n The sum of each row, column, and diagonal is: ", sum)
    return magic_square


def createRandomMatch(size):
    flat_magic = []                    # Flatten the Square
    for row in range(size):
        for col in range(size):
            flat_magic.append(magic_square[row][col])
    print ("\n\n Target 3 x 3 Square",flat_magic, end = " ")

    tries = 0               # counts the attempts to match
    match = False
    while match == False:     # Finding a match loop
        rnd_flat = []              # Create a candidiate list
        for index in range(size*size):
            hit = False            # Used to see if a ;p;otential list entry is unique
            while hit == False:
                num = random.randint(1,9)
                if num not in rnd_flat:
                    rnd_flat.append(num)
                    hit = True
                    break
            if rnd_flat[index] != flat_magic[index]:    # Chacking to candidate list matches so far
                #print (rnd_flat)
                break
        tries = tries + 1
        if tries % 100000 == 0:
            print ("\n",rnd_flat, "\n", tries)
        if rnd_flat == flat_magic:
            match = True    # Found a match
            break       # Break out of While match == False: loop

    #print (flat_magic, "\n")
    print (rnd_flat)

    print ("It took ", tries, "tries to create a random square to match the Magic Square")
```