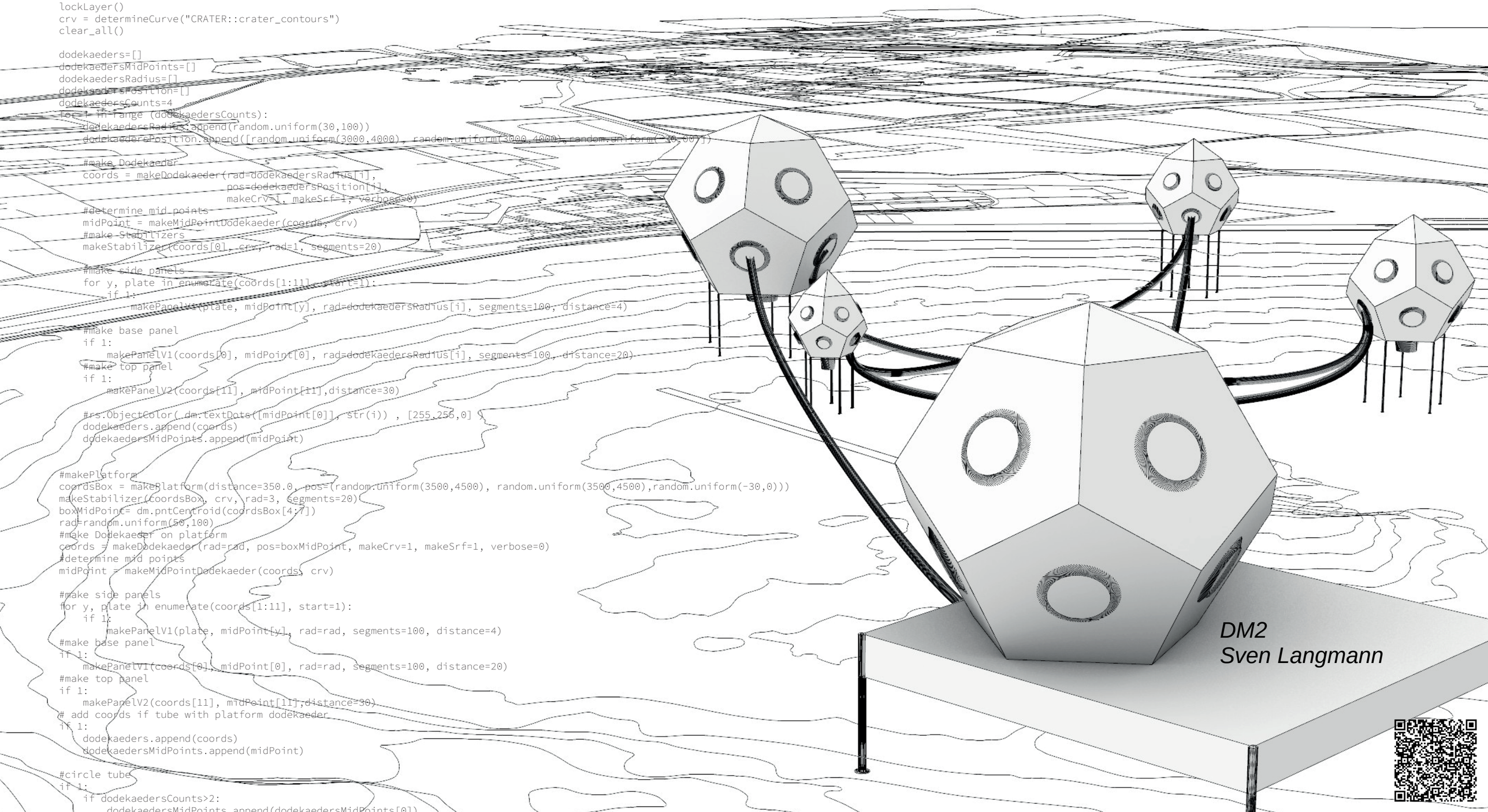
```
#Supp
def lockLayer():
def determineCurve(layer_name):
def clear_all():
def makeDodekaeder( rad=10.0, pos=[0,0,0], makeCrv=1, makeSrf=1, verbose=1):
def makeStabilizer(baseplate, crv, rad=1, segments=20):
def find_center(points):
def makeMidPointDodekaeder(dodekaeder, crv):
def makeTubeV1(dodekaedersMidPoints, rad=5, countCircle=10, countCirclePts=10):
def makeTubePanel(midPoints, rad=5, countCircle=10, countCirclePts=10):
    if midPoints:
def makeTubeV2(dodekaedersMidPoints, rad=5, countCircle=10, countCirclePts=10):
def distance_to_point(point, ref_point):
def makePanelV1(plate, midPoint, rad=10.0, segments=10, distance=10):
def makePanelV2(plate, midPoint, distance=10):
def makePlatform(distance=10.0, pos=[0,0,0]):

#Main
def main():
    lockLayer()
    crv = determineCurve("CRATER::crater_contours")
    clear_all()

    dodekaeders=[]
    dodekaedersMidPoints=[]
    dodekaedersRadius=[]
    dodekaedersPosition=[]
    dodekaedersCounts=4
    for i in range (dodekaedersCounts):
        dodekaedersRadius.append(random.uniform(30,100))
        dodekaedersPosition.append([random.uniform(3000,4000), random.uniform(3000,4000),random.uniform(-30,0)])

        #make Dodekaeder
        coords = makeDodekaeder(rad=dodekaedersRadius[i],
                                pos=dodekaedersPosition[i],
                                makeCrv=1, makeSrf=1, verbose=0)
        #determine mid points
        midPoint = makeMidPointDodekaeder(coords, crv)
        #make Stabilizers
        makeStabilizer(coords[0], crv, rad=1, segments=20)

        #make side panels
        for y, plate in enumerate(coords[1:11], start=1):
            if 1:
                makePanelV1(plate, midPoint[y], rad=dodekaedersRadius[i], segments=100, distance=4)
        #make base panel
        if 1:
            makePanelV1(coords[0], midPoint[0], rad=dodekaedersRadius[i], segments=100, distance=20)
        #make top panel
        if 1:
            makePanelV2(coords[11], midPoint[11],distance=30)

        #rs.ObjectColor( dm.textDots([midPoint[0]], str(i)) , [255,255,0] )
        dodekaeders.append(coords)
        dodekaedersMidPoints.append(midPoint)


    #makePlatform
    coordsBox = makePlatform(distance=350.0, pos=(random.uniform(3500,4500), random.uniform(3500,4500),random.uniform(-30,0)))
    makeStabilizer(coordsBox, crv, rad=3, segments=20)
    boxMidPoint= dm.pntCentroid(coordsBox[4:7])
    rad=random.uniform(50,100)
    #make Dodekaeder on platform
    coords = makeDodekaeder(rad=rad, pos=boxMidPoint, makeCrv=1, makeSrf=1, verbose=0)
    #determine mid points
    midPoint = makeMidPointDodekaeder(coords, crv)

    #make side panels
    for y, plate in enumerate(coords[1:11], start=1):
        if 1:
            makePanelV1(plate, midPoint[y], rad=rad, segments=100, distance=4)
    #make base panel
    if 1:
        makePanelV1(coords[0], midPoint[0], rad=rad, segments=100, distance=20)
    #make top panel
    if 1:
        makePanelV2(coords[11], midPoint[11],distance=30)
    # add coords if tube with platform dodekaeder
    if 1:
        dodekaeders.append(coords)
        dodekaedersMidPoints.append(midPoint)


    #circle tube
    if 1:
        if dodekaedersCounts>2:
            dodekaedersMidPoints.append(dodekaedersMidPoints[0])
```

Moon Crater Station



DM2
Sven Langmann

Zufällige Größe,
Zufällige Anordnung,
Zufällige Position
in X,Y,Z