```
import time, numpy, matplotlib, csv, cv2, sys
import matplotlib.pyplot as plt
Control_Points=[]
Curves=[]

def Curve_deCasteljau(b,t):

    """Compute point on Bezier curve with Casteljau algorithm
    Input:  Control points array (P),curve degree(n),parameter(u)
    Output: A point on curve corresponding to t """
    n=len(b)
    for r in range(1,n):
        for i in range(0,n-r):
            b[i]=(1.0-t)*b[i]+t*b[i+1]

    return b[0]

def Save_ControlPoints(fn, CP):
    with open(fn,'a',newline='') as csvfile:
        csv_writer=csv.writer(csvfile)
        for i in range(len(CP)):
            each_row=[CP[i][0], CP[i][1]]
            csv_writer.writerow(each_row)

def Save_Curve(fn, CPN):
    with open(fn,'a',newline='') as csvfile:
        csv_writer=csv.writer(csvfile)
        csv_writer.writerow(CPN)
```

```
def Retrive_Curves(fn):
    C=[]
    with open(fn,newline='') as csvfile:
        csv_reader=csv.reader(csvfile)
        for each_row in csv_reader:
            if len(each_row) < 2:
                return False,0
            else:
                C.append(each_row)
    return True, C

def Read_ControlPoints(fn):
    global Control_Points
    with open(fn,newline='') as csvfile:
        csv_reader=csv.reader(csvfile)
        for each_row in csv_reader:
            if len(each_row) != 2:
                Control_Points=[]
                return False
            else:
                Control_Points.append([float(each_row[0]),float(each_row[1])])
    return True

def Draw_Curve(CPoints):
    x=[]; y=[]; Px=[]; Py=[]
    for j in range(len(CPoints)):
            Px.append(CPoints[j][0])
            Py.append(CPoints[j][1])
    for i in range(1,102):
```

```
        t=(i-1)/(100);
        x.append(Curve_deCasteljau(Px,t))
        y.append(Curve_deCasteljau(Py,t))
    plt.plot(x,y)
    plt.draw()

def Draw_Curves():

        Draw_Curve(numpy.asarray(pts))

def tellme(s):
    print(s)
    plt.title(s, fontsize=16)
    plt.draw()



SR=input('Start new session(S) or Reload an existing model(R): ')
matplotlib.use('tkagg')
plt.figure()
plt.axis([0,300,0,300])
plt.show()
if SR == 's':
    Point_Number=0
    while True:
        pts = []
        while len(pts) < 2:
            tellme('Select Control Points with mouse')
            pts = numpy.asarray(plt.ginput(-1, timeout=-1))
            if len(pts) == 1:
```

```
            tellme('Too few points, starting over')
            time.sleep(1)  # Wait a second
        elif len(pts) == 0:
            break
    if len(pts) == 0:
        break
    CPN=[]
    for i in range(Point_Number, Point_Number+len(pts)):
        CPN.append(i)
    Point_Number += len(pts)
    Save_ControlPoints('CPdata.csv', pts)
    Save_Curve('CurveData.csv', CPN)
    Draw_Curve(pts)

elif SR == 'r':
    ret, Curves=Retrive_Curves('Curvedata.csv')
    Read_ControlPoints('CPdata.csv')

    for i in range(len(Curves)):
        Pn=[]; pts=[]
        Pn=Curves[i]
        for j in range(len(Pn)):
            pts.append([Control_Points[int(Pn[j])][0],Control_Points[int(Pn[j])][1]])
        Draw_Curve(pts)
```