punktinterp

March 22, 2017

1 Punktinterp

Script for weighted interpolation of a few points in 3D space

-Gerald Schuller, Nov. 2014

```
In [ ]: from OpenGL.GL import *
        from OpenGL.GLUT import *
        from OpenGL.GLU import *
        import numpy as np
```

```
In [ ]: def displayFun():
            glMatrixMode(GL_PROJECTION)
            glLoadIdentity()

            #Orthogonale Projektionsmatrix,
            #Orthogonal Projection Matrix
            #glOrtho(left, right, bottom, top, znear, zfar);
            glOrtho(-3.0,3.0,-1.5, 1.5, 0.1, 100)

            glMatrixMode(GL_MODELVIEW)
            glLoadIdentity()

            #Position der virtuellen Kamera:
            #Position of the virtual camera:
            #gluLookAt( eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , upY , upZ )
            gluLookAt(0.0, 0.0, 2.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0)

            glClear(GL_COLOR_BUFFER_BIT)

            glColor3f(1.0,0.0,0.0)

            #Kreis aus Punkten, Vertices berechnet innerhalb von glBegin und glEnd:
            #Circle of points, vertices calculated within glBegin and glEnd:
            glPointSize(5)
            glBegin(GL_POINTS)
            #3 Punkte die ein Dreieck aufspannen:
            #3 Points that span a triangle:
            P0=np.matrix([-1.0,-1.0,0.0])

            #glVertex3fv takes an Array or Matrix as input!:
            #glVertex3fv takes an array or matrix as input !:
            glVertex3fv(P0)
            P1 = np.matrix([1.0, -1.0, 0.0])
            glVertex3fv(P1)
            P2 = np.matrix([0.0, 0.5, 0.0])
            glVertex3fv(P2)

            glColor3f(0.0, 0.0, 0.0)
            f = 0.0
            for u in np.linspace(0.1 ,0.9, 10):
                b0 = u**2
                b1 = 0.3-0.3*u**2
                b2 = 0.7-0.7*u**2
                print(b0, b1, b2)
                print(b0 + b1 + b2)

                #linear interpolation of the 3 points inside the convex hull:
                glVertex3fv(b0*P0 + b1*P1 + b2*P2)

            glEnd()

            glFlush()
```

- **Test displayfun()**

```
In [ ]: if __name__ == '__main__':
            glutInit()
            glutInitWindowSize(640,480)
            glutCreateWindow("3D")
            glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
            glClearColor(1.0,1.0,1.0,0.0)
            glutDisplayFunc(displayFun)
            glutMainLoop()
```