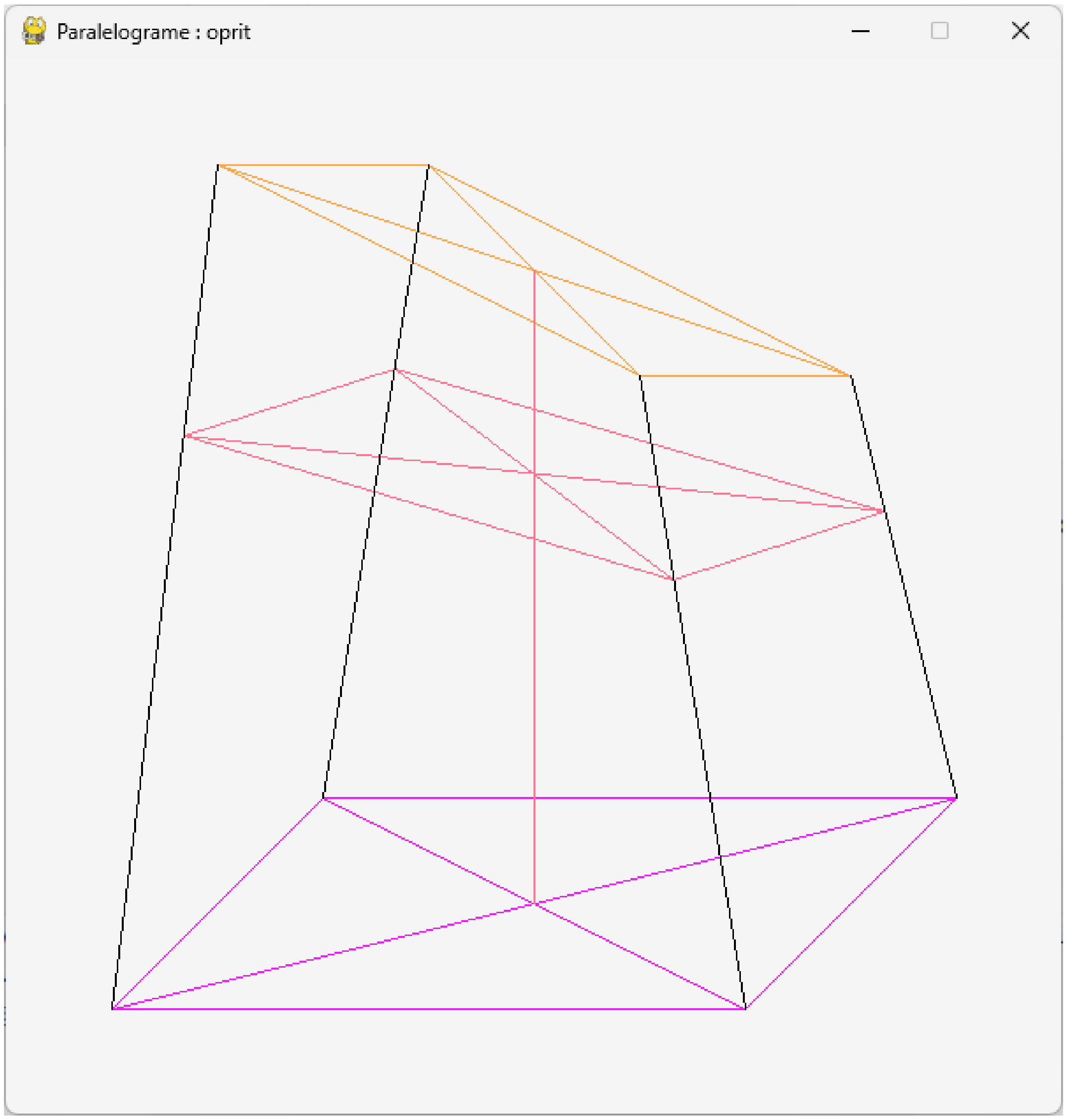
Curs 13

(*plan de curs*)

1 ½ oră Mulţimi Julia

½ oră: Paralelograme:



```
import ComplexPygame as C
import Color
import math
```

```
def Paralelograme():
    class Paralelogram:
        def __init__(self, a=0j, b=0j, c=0j):
            self._v = [a, b, c, a + c - b]
        def __getitem__(self, k):
            return self._v[k]
        # def __setitem__(self, k, varf):
        #     self._v[k] = varf
        def __add__(self, other):  # aduna self + other
            if isinstance(other, Paralelogram):
                return Paralelogram(*[self[k] + other[k] for k in range(3)])

        def __rmul__(self, other):  # returneaza other*self
            if isinstance(other, int) or isinstance(other, float):
                return Paralelogram(*[other * self[k] for k in range(3)])

        def centru(self):
            return sum(self._v) / 4

        def show(self, col):
            for k in range(4):
                for h in range(k + 1, 4):
                    C.drawLine(self[k], self[h], col)

    C.setXminXmaxYminYmax(0, 10, 0, 10)
    C.fillScreen()
    a = Paralelogram(1 + 1j, 7 + 1j, 9 + 3j)
    b = Paralelogram(2 + 9j, 6 + 7j, 8 + 7j)
    delta = math.pi / 10000
    a_col=400
    b_col=300
    for tk in range(10 ** 5):
        C.fillScreen()
        gama = math.sin(tk * delta)**2
        c = gama * a + (1-gama) * b
        c_col=int(gama*a_col +(1-gama)*b_col)
        a.show(Color.Index(a_col))
        b.show(Color.Index(b_col))
        c.show(Color.Index(c_col))
        for k in range(4):
            C.drawLine(a[k], b[k], Color.Black)
        C.drawLine(a.centru(), b.centru(), Color.Index(c_col))
        if C.mustClose(): return

if __name__ == '__main__':
    C.initPygame()
    C.run(Paralelograme)
```