



```

    Polys
    Receptacle_and_Primordia

    FailedPolygon=0
    Do While n<=tl
    Order_Fractions
    If FailedPolygon>0 Then Exit Do (RepairPolygon)
    Define_PPoly
    Define_FlatRings
    Translate_FlatRings & 'Clover'
    *Compile_Receptacle (Ring 0...1) & 'Str/Twist'
    *Compile_Primordium (Ring1...Rings_Prim)
    Loop
    
```

\*naar OGL; niet in arrays

```

    Polys
    Polygon

    Reparations=0
    VascularUnits
    Rings_Leaf=Int(PolyRings*(RingTotal-1))+1
    If RingTotal=8 then 'lowest resolution
    Rings_Prim=Rings_Leaf
    Else
    Rings_Prim=0.5*Rings_Leaf
    End If
    Receptacle_and_Primordia
    Do While FailedPolygon>0 and Reparations<Rep
    Repair_Polygon
    Loop

    0<=PolyRings<=1; 8<=RingTotal<=128;
    2<=Vertices<=32;
    8<=Rings_Prim<=64; 8<=Rings_Leaf<=128
    
```

```

    Polys
    Repair_Polygon

    Reparations=Reparations+1
    CalcPermission
    nv=FailedPolygon
    TransCalculation xx -> xxx, not 'Clover'
    ColorCalculation
    FailedPolygon=0
    If Core.Lock then glNewList Core
    If Spheres.Lock then glNewList Spheres
    VascularUnits
    Receptacle_and_Primordia
    glNewList Polygons
    
```

