APEX POLYS FLOW *frank '20/02, '21/07, '22/08, '23/11, '24/01*

VascularUnits
towards translated fractions with thickness

Do While n<=nt
 Tria
 Quad (example)
 Pent
 Hexa
 Sept
 Octa
 Nona
Loop

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<=64;
8<=Rings_Prim<=64; 8<=Rings_Leaf<=128

Receptacle_and_Primordia

FailedPolygon=0
Do While n<=tl
 Order_Fractions
 If FailedPolygon>0 Then Exit Do *(RepairPolygon)*
 Define_PPoly
 Smooth_FlatRing_0
 Define_FlatRings
 Translate_FlatRings ***& 'Clover'***
 Compile_Receptacle *(Ring 0...1)*
 Compile_Primordium *(Ring1...Rings_Prim)*
Loop

Compile_Receptacle

Define_RecSection (2x)
Glue_Ring 0
Glue_Ring 1
Show_Section (2x)

Compile_Primordium

ProtoPrimordium
ProtoLeaf
Get_Extensions
Define_SeedSection (multi)
Define_LeafSection (2x)
[Glue_BaseRim] (2x)
Show_Section (multi)

Define_RecSection

input: u (0-1), v (F-LVertex)
XB(u,v), Y, Z
output: XE(u,v,0), Y, Z
Thick_Receptacle

Define_SeedSection

input: u (2-7), v (F-LVertex)
XB(u,v), Y, Z
output: XE(u,v,0), Y, Z
Shape_SeedSection
Define_PrimTip (u=7,8)
Smooth_SeedProfile
Thick_SeedSection

Define_LeafSection

input: u (2-7), v (F-LVertex)
XB(u,v), Y, Z
output: XE(u,v,0), Y, Z
Shape_LeafSection
Define_PrimTip (u=7,8)
Smooth_LeafProfile
Thick_LeafSection

Shape_SeedSection

Extend
XE(u,v,0), Y, Z (u=2-7)

Shape_LeafSection

DiffLeaf
Extend
XE(u,v,0), Y, Z (u=2-6)
stem leaves: tip lift (u=2-7)

Thick_Receptacle

input: u (0-1), v (F-LVertex)
XE(u,v,0), Y, Z, dXb(v), Y, Z
output: XE(u,v,1), Y, Z

Thick_SeedSection

input: u (1-RingTot), v (F-LVertex)
XE(u,v,0), Y, Z
output: XE(u,v,1), Y, Z

Thick_LeafSection

input: u (1-RingTot), v (F-LVertex)
XE(u,v,0), Y, Z
output: XE(u,v,1), Y, Z
Shape_LeafEdge

Extend

Extend_RingSection (u=2-7)

Extend_RingSection

input: u (1-RingTot), v (F-LVertex)
XE(u,v,0), Y, Z
output:
XE(u,v,0), Y, Z *(val.changed)*

Order_Fractions

on error: FailedPolygon=n, exit sub
v has lowest Z_drop(n,v) of all first fraction's vertices; w=0
Transfer_Vertex
Do While v is not empty space or Fractions<4
 w has coords like last fraction's v
 Transfer_Vertex
 v=last place
Loop

Transfer_Vertex

glue fraction behind the last one
input: fraction/ normals with first vertex v
X_drop(n,v), dX_drop(n,v), Y, Z
output: mother polygon/ normals with ordered indices
XB(0,w), dXB(w), Y, Z

Smooth_FlatRing_0

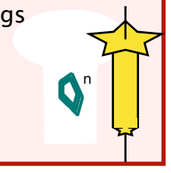
Do While v<=PPoly
 DuoPoint (multi)
 TrioPoint (multi)
 => XB(0,v), Y, Z
Loop

Define_FlatRings

(FlatRing 2)
Do While v<=PPoly
 TrioPoint (multi)
 => XB(2,v), Y, Z
Loop
Balance_FlatRing_2
(FlatRing 1)
 => XB(1,v), Y, Z
(FlatRing 7)
 => XB(7,v), Y, Z
(FlatRings 3,4,5,6)
 => XB(3,v), Y, Z

Translate_FlatRings

input: u (0-8), v (0-PPoly)
(on cylinder and hemisphere)
XB(u,v), Y, Z
output:
(on transformed core)
XB(u,v), Y, Z *(values changed)*



Show_Section

If 3D-Print Then
 Write_Solid (-> Write_Entity .STL)
 *(GL_triangles -> **Put_VertexS**)*
 Triangles_Colorless
Else
 If CheckSolid then
  "meshS"
   *(GL_triangle_strip -> **Put_VertexS**)*
   RadialRimS, TangentRimS
   LeafPlane0, LeafPlane1
  "meshT"
   *(GL_triangles -> **Put_VertexS**)*
   Triangles_Colored
  "meshQ"
   *(GL_quads -> **Put_VertexS**)*
   Quads_Colored
 End If
 If CheckWireFrame Then
  (glColor3d)
  If FullMesh Then
   Radial_EdgesW
  Else
   *(GL_lines -> **Put_VertexW**)*
   Radial_EdgesW, Connect_TangentsW,
   TangentsW, RadialsW, Connect_RadialsW,
   DiagonalsW, Radials_combine_DiagonalsW
  End If
  Tangent_EdgesW
 End If
End If

Put_VertexS

Solid OpenGL
(glColor3d)
(glNormal3f)
(glVertex3f)
& 'Stretch/Twist'

Put_VertexW

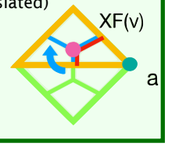
Line OpenGL
(glNormal3f)
(glVertex3f)
& 'Stretch/Twist'

Quad

PolyPieces
(Ghost_Trapped...)

PolyPieces

nx,ny,nz (normal)
XF(Vertices/2), Y, Z (gravity center)
Do While k<3
 (Define Fraction)
 input:
 XF(0), XF(Vertices), Y, Z
 output:
 XF(tussenpunten), Y, Z
 If DropSpace<VMaxFractions Then
  w is first empty space in ZD(a,w)
  input:
  XF(v), Y, Z
  output: see below
  Do While v<=Vertices And w<=VMaxFractions
   (Drop Fraction)
   X_drop(a,w)=XF(v), Y, Z
   (Thick Fraction)
   Get_Fraction_Normal (translated)
   dX_drop(a,w)=nx, Y, Z
  Loop
  DropSpace=w
 End If
Loop



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