module Bud(int time, int phyllo, int order) extends Sphere(0.1)

{

{setShader(nodemat);}

}

module Node extends Sphere(0.07)

{

{setShader(GREEN);}

}

const ShaderRef leafmat = new ShaderRef("Lambert");

const ShaderRef petalmat = new ShaderRef("Lambert 2");

const ShaderRef internodemat = new ShaderRef("Lambert 3");

const ShaderRef nodemat = new ShaderRef("Lambert 4");

module Leaf (super.length, super.width, float al, int age)

extends Box(length,width,0.01).(setShader(new AlgorithmSwitchShader(new RGBAShader(0,1,0), GREEN )));

module NiceLeaf (float al) extends Parallelogram(2,2).(setShader(leafmat));

module NiceNode extends Sphere(0.07)

{{setShader(nodemat);}}

module Internode extends F(1,0.1,7);

//module NiceInternode extends Cylinder(1,0.08).(setShader(internodemat));

module NiceInternode extends Cylinder(1,0.08).(setShader(new RGBAShader(1,0.5,0.1)));

module Flower ==>

RU(180) Cone(0.3,0.3).(setColor(0x82B417)) M(-0.25) RL(90)

[ for (int i=1; i<=5; i++) ([RU(i\*360/5) RL(20) Parallelogram(2,1).(setColor(0xFF00FF))])]

RU(45)

[ for (int j=1; j<=5; j++) ([RU(j\*360/5) RL(40) F(0.3,0.1,14) RV(-0.3) F(0.3,0.1,14) RV(-0.3) F(0.3,0.1,14)])

]

RU(-45)

[ for (int k=1; k<=5; k++) ([RU(k\*360/5) RL(70) Frustum(0.7,0.2,0.05).(setColor(0x8DAF58))])

];

module NiceFlower ==>

RU(180) Cone(0.3,0.3).(setShader(internodemat)) M(-0.25) RL(90)

[ for (int i=1; i<=5; i++) ([RU(i\*360/5) RL(20) Parallelogram(2,1).(setShader(petalmat))])]

RU(45)

[ for (int j=1; j<=2; j++) ([RU(j\*360/3) RL(40) F(0.3,0.1,14) RV(-0.3) F(0.3,0.1,14) RV(-0.3) F(0.3,0.1,14)])

]

RU(-45)

[ for (int k=1; k<=5; k++) ([RU(k\*360/5) RL(70) Frustum(0.7,0.2,0.05).(setColor(0x8DAF58))])

];

//Introducing leaf growth

//derivative of logistic function

public float logistic (float maxdim, int time, float phylloM, float slope)

{

return (slope\*maxdim\*Math.exp(-slope\*(time-phylloM)))/

((Math.exp(-slope\*(time-phylloM))+1)\*\*2);

}

const DatasetRef light = new DatasetRef("Light intercepted by canopy");

LightModel lm = new LightModel(100000,5);

const int phyllo = 14;

protected void init ()

[

{

light.clear();

chart(light,XY\_PLOT);

}

Axiom ==> Bud(1,phyllo, 1);

]

public void grow ()

{

for(apply(1)) run();

//lm.setSeed(irandom(1,100000));

lm.compute();

for(apply(1)) la();

light.addRow().set(0,sum((\* Leaf \*)[al]));

}

protected void run ()

[ Bud(r,p,o),(r<10 && p>0) ==> Bud(r,p-1,o);

Bud(r,p,o),(r<10 && p==0 && o<4) ==>

RV(-0.1) NiceInternode NiceNode [ RL(50) Bud(r+1,phyllo,o+1) ]

[RL(70) Leaf(0.1,0.07,0,1)] RH(137) RV(-0.1) NiceInternode Bud(r+1,phyllo, o);

Bud(r,p,o), (r==10) ==> RV(-0.1) NiceInternode NiceInternode NiceFlower;

]

protected void la ()

[ lf:Leaf ::> {

lf[al] = lm.getAbsorbedPower3d(lf).integrate();

lf.(setShader(new AlgorithmSwitchShader(new RGBAShader((float) lf[al]/5.0f, (float) lf[al]\*2, (float) lf[al]/100.0f), GREEN)));

println(lf[al]);

lf[age]++;

lf[length] += logistic(2,lf[age],10,0.5);

lf[width] = lf[length]\*0.7;

}]