Assignment 5 – Building Trees

Nate Paternoster

EEN318

#include "library.h"

const double pi = acos(-1.0);

const double ninety = pi/2, degrees = pi/180;

void block1(double p1x, double p1y, double p2x, double p2y, double len, double beta, double end, int level) {

double base = sqrt((p1x-p2x)\*(p1x-p2x) + (p1y-p2y)\*(p1y-p2y));

double alpha = atan2(p2x-p1x, p1y-p2y) - ninety;

if (sin(beta)>2\*len/base || end < 0.3) {

if (random\_in\_range(0,500)==500) {

set\_pen\_color(color::red);

set\_pen\_width(20);

draw\_point();

}

else {

set\_pen\_color\_int(10,random\_in\_range(100,200),10);
set\_pen\_width(10);

draw\_point();

}

return;

}

double p3x = (p1x + p2x) \* 0.5;

double p3y = (p1y + p2y) \* 0.5;

double p4x = p3x + len \* sin(alpha+beta);

double p4y = p3y - len \* cos(alpha+beta);

double dx = 0.5 \* end \* sin(ninety-alpha-beta);

double dy = 0.5 \* end \* cos(ninety-alpha-beta);

double p5x = p4x - dx;

double p5y = p4y - dy;

double p6x = p4x + dx;

double p6y = p4y + dy;

set\_pen\_color(color::black);

set\_pen\_width(1);

start\_shape();

move\_to(p1x, p1y);

note\_position();

draw\_to(p2x, p2y);

note\_position();

draw\_to(p6x, p6y);

note\_position();

draw\_to(p5x, p5y);

note\_position();

draw\_to(p1x, p1y);

note\_position();

set\_pen\_color(color::brown);

fill\_shape();

if (level<100) {

set\_pen\_color(color::black);

move\_to(p1x, p1y);

draw\_to(p2x, p2y);

draw\_to(p6x, p6y);

draw\_to(p5x, p5y);

draw\_to(p1x, p1y);

}

block1(p5x,p5y,p6x,p6y,len/1.4,beta+20\*degrees,end/2,level+1);
block1(p5x,p5y,p6x,p6y,len/1.5,beta-20\*degrees,end/2,level+1);
block1(p5x,p5y,p6x,p6y,len/1.3,beta-10\*degrees,end/2,level+1);

if (random\_in\_range(0,1)==1)

block1(p5x,p5y,p6x,p6y,len/1.2,beta+10\*degrees,end/2,level+1);

}

void main() {

int width = 800;

int height = 800;

make\_window(width, height);

fill\_rectangle(0,0,width,height-height/6,color::light\_blue);
fill\_rectangle(0,height-height/6,width,height,color::dark\_green);
set\_pen\_color(color::yellow);

set\_pen\_width(width/10);

draw\_point(width/10,height/10);

block1(width\*0.5-width/20, height, width\*0.5+width/20, height, height/5, 0, width/10, 0);

}

