Cristian Brinza FAF-212

Report

Final Eman

of Computer Graphics

Made by:

Cristian Brinza, *university student FAF-212*

DISA, FCIM, UTM

Checked by:

Olga Grosu, *university assistant*

DISA, FCIM, UTM

Chișinău – 2022

1. The condition / conditions of the problems:

Make a tree. Set the angles of the branches of the tree according to Perlin noise values. Adjust the noise values over time to animate the tree. See if you can get it to appear as if it is blowing in the wind. Can you add leaves or flowers to the end of the branches? What if the leaves can fall off the tree. And comment to the program.

2. The code with comments:

```
The program code, having relevant comments in it: float angle;
float time = 0;
float wind = 0.01;
int leaves = 100;
int[] leafX = new int[leaves];
int[] leafY = new int[leaves];
boolean[] leafFallen = new boolean[leaves];

void setup() {
  size(600, 600);
  strokeWeight(2);
  angle = radians(20);

  // initialize leaf positions and fallen state
  for (int i = 0; i < leaves; i++) {
    leafX[i] = int(random(width));
    leafY[i] = int(random(height));
    leafFallen[i] = false;
  }
}

void draw() {
  background(9, 204, 230);
  rect(0, height-20,width,height);
  // set the angle of the branches using Perlin noise
  float noiseVal = noise(time);
  angle = map(noiseVal, 0, 1, radians(20), radians(40));

  // animate the tree by incrementing the time variable
  time += 0.01;

  // draw the fractal tree
  translate(width/2, height);
  branch(150);

  // add leaves to the end of the branches with a random chance
  for (int i = 0; i < leaves; i++) {
    if (random(1) < 0.05 && !leafFallen[i]) {
      fill(0, 255, 0);
      ellipse(leafX[i], leafY[i], 20, 10);
      rotate(20);
    }
}
```

```
    }

    // simulate leaves falling off the tree in the direction of the wind
    for (int i = 0; i < leaves; i++) {
      if (leafFallen[i]) {
        leafY[i] += wind;
        if (leafY[i] > height) {
          leafFallen[i] = false;
          leafX[i] = int(random(width));
          leafY[i] = int(random(height));
        }
      }
    }
}

void branch(float len) {
  // draw the current branch
  line(0, 0, 0, -len);
  translate(0, -len);

  // if the length of the current branch is greater than a certain value, draw its
child branches
  if (len > 4) {
    pushMatrix();
    rotate(angle);
    branch(len * 0.67);
    popMatrix();
    pushMatrix();
    rotate(-angle);
    branch(len * 0.67);
    popMatrix();
  }
}
```

3. Screen printing of program execution;