

CA1 – Computer Graphics

For computer graphics we were required to draw a 3d image with 4 objects and 3 of them had to rotate.

Peasy Cam

For this I had to download the peasy cam library from the processing application itself. After that I imported peasy and created an object.

```
import peasy.*;
PeasyCam movement;
```

I then set it up in the void setup method. I created the minimum and maximum distance.

```
movement = new PeasyCam(this, 800);
movement.setMinimumDistance(50);
movement.setMaximumDistance(600);
```

I then went on to create all my points. I did this using the float array function and defined the x,y, z and the 1.0.

```
//island points
float[] island1 = {150,250,60, 1.0};
float[] island2 = {200,200,60, 1.0};
float[] island3 = {250,200,60, 1.0};
float[] island4 = {300,250,60, 1.0};
float[] island5 = {150,250,0, 1.0};
float[] island6 = {200,200,0, 1.0};
float[] island7 = {250,200,0, 1.0};
float[] island8 = {300,250,0, 1.0};

//tree points
float[] tree1 = {210,200,50, 1.0};
float[] tree2 = {240,200,50, 1.0};
float[] tree3 = {210,200,10, 1.0};
float[] tree4 = {240,200,10, 1.0};

float[] tree5 = {210,0,50, 1.0};
float[] tree6 = {240,0,50, 1.0};
float[] tree7 = {210,0,10, 1.0};
float[] tree8 = {240,0,10, 1.0};

//leaves
float[] leaves1 = {225,40,100, 1.0};
float[] leaves2 = {270,40,25, 1.0};
float[] leaves3 = {180,40,25, 1.0};
float[] leaves4 = {225,40,-20, 1.0};

//bird points
float[] bird1 = {100,100,60, 1.0};
float[] bird2 = {120,120,60, 1.0};

float[] bird3 = {110,110,70, 1.0};
float[] bird4 = {110,110,50, 1.0};

//person points
float person1[] = {325,200,60, 1.0};
float person2[] = {360,200,60, 1.0};
float person3[] = {360,225,60, 1.0};
float person4[] = {380,225,60, 1.0};
float person5[] = {325,225,60, 1.0};
float person6[] = {305,225,60, 1.0};

float person7[] = {325,200,45, 1.0};
float person8[] = {360,200,45, 1.0};
float person9[] = {360,225,45, 1.0};
float person10[] = {380,225,45, 1.0};
float person11[] = {325,225,45, 1.0};
float person12[] = {305,225,45, 1.0};
```

Next I created the void setup method. In this I defined the size of the canvas, the camera and its positions. As discussed before I set up the peasy cam here as well. Then I set smooth to make it appear less pixelated and set it to a loop to repeat the translations. Then I set the framerate to 15 as my personal preference.

```

void setup()
{
    size(900, 900, P3D);
    camera(170.0, 325.0, 280.0, 50.0, 50.0, 0.0, 0.0, 1.0, 0.0);

    movement = new PeasyCam(this, 800);
    movement.setMinimumDistance(50);
    movement.setMaximumDistance(600);
    smooth();

    loop();

    frameRate(15);
}

```

Then I went on to make some draw methods for the shapes. For these I made an island, a bird, a tree and a person in the water. I used line and connected each of the points defined above using line to create a wireframe.

```

void drawIsland()
{
    //draws line between 2 points
    stroke(0, 255, 0);
    line(island1[0], island1[1], island1[2], island2[0], island2[1], island2[2]);
    line(island2[0], island2[1], island2[2], island3[0], island3[1], island3[2]);
    line(island3[0], island3[1], island3[2], island4[0], island4[1], island4[2]);
    line(island4[0], island4[1], island4[2], island1[0], island1[1], island1[2]);
    line(island1[0], island1[1], island1[2], island5[0], island5[1], island5[2]);
    line(island2[0], island2[1], island2[2], island6[0], island6[1], island6[2]);
    line(island3[0], island3[1], island3[2], island7[0], island7[1], island7[2]);
    line(island4[0], island4[1], island4[2], island8[0], island8[1], island8[2]);
    line(island5[0], island5[1], island5[2], island6[0], island6[1], island6[2]);
    line(island6[0], island6[1], island6[2], island7[0], island7[1], island7[2]);
    line(island7[0], island7[1], island7[2], island8[0], island8[1], island8[2]);
    line(island5[0], island5[1], island5[2], island8[0], island8[1], island8[2]);
}

void drawTree()
{
    //log
    stroke(0, 255, 0);
    line(tree1[0], tree1[1], tree1[2], tree3[0], tree3[1], tree3[2]);
    line(tree2[0], tree2[1], tree2[2], tree4[0], tree4[1], tree4[2]);
    line(tree3[0], tree3[1], tree3[2], tree4[0], tree4[1], tree4[2]);
    line(tree1[0], tree1[1], tree1[2], tree2[0], tree2[1], tree2[2]);
    line(tree1[0], tree1[1], tree1[2], tree5[0], tree5[1], tree5[2]);
    line(tree2[0], tree2[1], tree2[2], tree6[0], tree6[1], tree6[2]);
    line(tree3[0], tree3[1], tree3[2], tree7[0], tree7[1], tree7[2]);
    line(tree4[0], tree4[1], tree4[2], tree8[0], tree8[1], tree8[2]);
    line(tree5[0], tree5[1], tree5[2], tree7[0], tree7[1], tree7[2]);
    line(tree6[0], tree6[1], tree6[2], tree8[0], tree8[1], tree8[2]);
    line(tree5[0], tree5[1], tree5[2], tree6[0], tree6[1], tree6[2]);
    line(tree7[0], tree7[1], tree7[2], tree8[0], tree8[1], tree8[2]);

    //leaves
    line(tree5[0], tree5[1], tree5[2], leaves1[0], leaves1[1], leaves1[2]);
    line(tree6[0], tree6[1], tree6[2], leaves1[0], leaves1[1], leaves1[2]);

    line(tree6[0], tree6[1], tree6[2], leaves2[0], leaves2[1], leaves2[2]);
    line(tree8[0], tree8[1], tree8[2], leaves2[0], leaves2[1], leaves2[2]);

    line(tree5[0], tree5[1], tree5[2], leaves3[0], leaves3[1], leaves3[2]);
    line(tree7[0], tree7[1], tree7[2], leaves3[0], leaves3[1], leaves3[2]);

    line(tree8[0], tree8[1], tree8[2], leaves4[0], leaves4[1], leaves4[2]);
    line(tree7[0], tree7[1], tree7[2], leaves4[0], leaves4[1], leaves4[2]);
}

```

```

void drawBird()
{
    stroke(0, 255, 0);
    line(bird1[0], bird1[1], bird1[2], bird2[0], bird2[1], bird2[2]);
    line(bird3[0], bird3[1], bird3[2], bird4[0], bird4[1], bird4[2]);
}

void drawPerson()
{
    line(person1[0], person1[1], person1[2], person2[0], person2[1], person2[2]);
    line(person2[0], person2[1], person2[2], person3[0], person3[1], person3[2]);
    line(person3[0], person3[1], person3[2], person4[0], person4[1], person4[2]);
    line(person1[0], person1[1], person1[2], person5[0], person5[1], person5[2]);
    line(person5[0], person5[1], person5[2], person6[0], person6[1], person6[2]);
    line(person1[0], person1[1], person1[2], person7[0], person7[1], person7[2]);
    line(person2[0], person2[1], person2[2], person8[0], person8[1], person8[2]);
    line(person3[0], person3[1], person3[2], person9[0], person9[1], person9[2]);
    line(person4[0], person4[1], person4[2], person10[0], person10[1], person10[2]);
    line(person5[0], person5[1], person5[2], person11[0], person11[1], person11[2]);
    line(person6[0], person6[1], person6[2], person12[0], person12[1], person12[2]);

    line(person3[0], person3[1], person3[2], person5[0], person5[1], person5[2]);
    line(person9[0], person9[1], person9[2], person11[0], person11[1], person11[2]);

    line(person7[0], person7[1], person7[2], person8[0], person8[1], person8[2]);
    line(person8[0], person8[1], person8[2], person9[0], person9[1], person9[2]);
    line(person9[0], person9[1], person9[2], person10[0], person10[1], person10[2]);
    line(person7[0], person7[1], person7[2], person11[0], person11[1], person11[2]);
    line(person11[0], person11[1], person11[2], person12[0], person12[1], person12[2]);
}

```

Then I created the draw method. In this I defined the background and set it to black. The stroke weight which is the thickness of the lines. The m variable was applied to the matrices and called apply transform. This was not put in until nearly last though. I first put in the draw methods calls to draw my static image first.

```

void draw()
{
    background(0);
    strokeWeight(5);
    //drawAxes();
    drawIsland();

    m = translateToOriginMatrix;
    applyTransform1();

    m = scalingMatrix;
    applyTransform1();

    m = scalingBackMatrix;
    applyTransform1();

    //m = rotYMatrix;
    //applyTransform1();

    m = translateFromOriginMatrix;
    applyTransform1();

    drawBird();

    m = translateToOriginMatrix;
    applyTransform2();

    m = rotYMatrix;
    applyTransform2();

    m = translateFromOriginMatrix;
    applyTransform2();

    drawTree();

    m = translateToOriginMatrix;
    applyTransform3();

    m = rotYMatrix;
    applyTransform3();

    m = translateFromOriginMatrix;
    applyTransform3();

    drawPerson();
}

```

Next then I went onto the transforming. I created matrices for scaling and translating. Then I made 2 more for setting it back to its original position name `translatetooriginmatrix` and `transatefromoriginmatrix`. Then I made a float `theta` using radians to convert from degrees to radians. Then I made 3 rotation matrices for rotation. I also made a variable `m` to set to the matrices in the `draw` method.

```
//scaling & translate
float[][] scalingMatrix = {{0.9,0.0,0.0,0.0},{0.0,0.9,0.0,0.0},{0.0,0.0,0.9,0.0},{0.0,0.0,0.0,1.0}}; // scale by 0.9
float[][] scalingBackMatrix = {{1.0,0.0,0.0,0.0},{0.0,1.0,0.0,0.0},{0.0,0.0,1.0,0.0},{0.0,0.0,0.0,1.0}}; // scale back to normal size

float[][] translateMatrix = {{1.0,0.0,0.0,0.0},{0.0,1.0,0.0,0.0},{0.0,0.0,1.0,0.0},{-10,-10,-10,1.0}}; // translate by distance of -10

float[][] translateToOriginMatrix = {{1.0,0.0,0.0,0.0},{0.0,1.0,0.0,0.0},{0.0,0.0,1.0,0.0},{-75,-50,-20,1.0}}; // translate back to originy distance of -10
float[][] translateFromOriginMatrix = {{1.0,0.0,0.0,0.0},{0.0,1.0,0.0,0.0},{0.0,0.0,1.0,0.0},{75,50,20,1.0}}; // translate back to originy distance of -10

float theta = radians(10); // convert degrees to radians

float[][] rotXMatrix = {{1.0,0.0,0.0,0.0},{0.0,cos(theta),sin(theta),0.0},{0.0,-sin(theta),cos(theta),0.0},{0.0,0.0,0.0,1.0}}; // roatate around x axis
float[][] rotYMatrix = {{cos(theta),0.0,-sin(theta),0.0},{0.0,1.0,0.0,0.0},{sin(theta),0.0,cos(theta),0.0},{0.0,0.0,0.0,1.0}}; // roatate around y axis
float[][] rotZMatrix= {{cos(theta),sin(theta),0.0,0.0},{-sin(theta),cos(theta),0.0,0.0},{0.0,0.0,1.0,0.0},{0.0,0.0,0.0,1.0}}; // roatate around z axis

float[][] m;
```

To conclude I feel I learned a bit doing this project mathematically with all the matrices and I have a good understanding of how to create static 3d images in processing.

This was the finished product without the transforming:

