Points

divide points into 3 equal parts:

```
i@part = floor(fit(rand(@ptnum+.258), 0, 1, 0, 2.9));
```

Nage replacement

```
@nage = fit(@age,0,@life,0,1);
```

Group from class:

```
string grpname = sprintf("class_%d", i@class);
setprimgroup(0, grpname, @primnum, 1); // use setpointgroup() if running over points
```

Connect close points:

```
float max_dist = chf("connect_distance");

int npts = @numpt;
for (int i = 0; i < npts; i++) {
    vector pi = point(0, "P", i);
    for (int j = i+1; j < npts; j++) {
        vector pj = point(0, "P", j);
        if (distance(pi, pj) < max_dist) {
            int prim = addprim(0, "polyline");
            addvertex(0, prim, i);
            addvertex(0, prim, j);
            setprimattrib(0, "create_frame", prim, @Frame);

        }
    }
}
```

Shift 0 -1 to 0 - 1 - 0:

```
float mask = f@mask;

float midpoint = 0.5;

float result = 1.0 - abs(mask - midpoint) * 2.0;
result = clamp(result, 0.0, 1.0);

f@newmask = result;
```

Confine points to sphere:

```
vector center = prim(1, "P", 0);
float radius = ch("scale");

vector dir = @P - center;
float dist = length(dir);

if (dist > radius) {
    @P = center + normalize(dir) * radius;
}
```

Connect points with lines by dist:

```

float max_dist = chf("connect_distance");

// Loop through all points in Group A
int npts = @numpt;
for (int i = 0; i < npts; i++) {
    if (!inpointgroup(0, "groupA", i)) continue;

    vector pi = point(0, "P", i);

    // Loop through all points in Group B
    for (int j = 0; j < npts; j++) {
        if (!inpointgroup(0, "groupB", j)) continue;

        // Avoid self-connection in case of overlapping group membership
        if (i == j) continue;
```

```
        vector pj = point(0, "P", j);

        if (distance(pi, pj) < max_dist) {
            int prim = addprim(0, "polyline");
            addvertex(0, prim, i);
            addvertex(0, prim, j);
            @primid = @primnum;

        }
    }
}
```

Revision #12
Created 2023-04-25 12:05:42 UTC by Anthony
Updated 2025-05-29 16:39:43 UTC by Anthony