
From: Lesley Groff [REDACTED]
Sent: Wednesday, February 1, 2017 11:08 PM
To: Jeffrey Epstein
Subject: Work from home tomorrow?

Since wheels up is 11am should I just work from home tomorrow? I can certainly come in at 9am if you want me to!!

Sent from my iPhone=?xml version=.0" encoding=TF-8"?> <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version=.0">

<dict>

<key>date-last-viewed</key>

<integer>0</integer>

<key>date-received</key>

<integer>1485990454</integer>

<key>flags</key>

<integer>8590195717</integer>

<key>gmail-label-ids</key>

<array>

<integer>5</integer>

<integer>18</integer>

</array>

<key>remote-id</key>

<string>684096</string>

</dict><coordOffset.w>-jsl_pixCoordOffset.y);

const int MAX_FRACTIONS =2;

const float TEXTURE_WIDTH =6.0;

const float FULL_TEXEL_X =.0 / TEXTURE_WIDTH; const float HALF_TEXEL_X =ULL_TEXEL_X / 2.0; uniform vec4

fractions[12]; uniform sampler2D colors; uniform float offset;

vec4 sampleGradient(float dist) {

int i;

float relFraction =.0;

{

relFraction +=lamp((dist - fractions[0].x) * fractions[0].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[1].x) * fractions[1].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[2].x) * fractions[2].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[3].x) * fractions[3].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[4].x) * fractions[4].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[5].x) * fractions[5].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[6].x) * fractions[6].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[7].x) * fractions[7].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[8].x) * fractions[8].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[9].x) * fractions[9].y, 0.0, 1.0); } { relFraction +=lamp((dist - fractions[10].x) * fractions[10].y, 0.0, 1.0); } float tc =ALF_TEXEL_X + (FULL_TEXEL_X * relFraction); return texture2D(colors, vec2(tc, offset)); }

vec4 cycleNone(float dist) {

if (dist <=.0){

return texture2D(colors, vec2(0.0, offset)); } else if (dist >=.0){ return texture2D(colors, vec2(1.0, offset)); } else { return sampleGradient(dist); }

vec4 cycleReflect(float dist) {

dist =.0 - (abs(fract(dist * 0.5) - 0.5) * 2.0); return sampleGradient(dist); }

```

vec4 cycleRepeat(float dist) {
    dist = fract(dist);
    return sampleGradient(dist);
}
uniform vec4 m0;
uniform vec4 m1;
uniform vec3 precalc;
uniform vec3 perspVec;
lowp vec4 paint(vec2 winCoord) {
    vec3 fragCoord = vec3(winCoord.x, winCoord.y, 1.0); float wdist = dot(fragCoord, perspVec); float x = dot(fragCoord, m0.xyz) /
    wdist + m0.w; float y = dot(fragCoord, m1.xyz) / wdist + m1.w; float xfx = - precalc.x; float dist = precalc.x * xfx + sqrt(xfx *
    xfx + y * y * precalc.y) * precalc.z; return cycleNone(dist); } void main() { gl_FragColor = aint(pixcoord) * perVertexColor;
} PK

```