```
// "3D Calligraphy"
// This is a sketch of Processing to draw 3D Calligraphy in a virtual space.
// A CSV file "syodo.csv" containig the original or revised data should be in the same folder of this sketch.
// Copyright (C) 2014 ArduinoDeXXX All Rights Reserved.

int LENGTH;
String [][] csv;
int i = 0;
int h = 0;
float x, y, z;
float zx, zy,zz;
float rec_x, rec_y, rec_z;
float rec_zx, rec_zy, rec_zz;
float dx, dy, dz;
float dzx, dzy,dzz;
float wdL;
byte numWrite = 0;
int obsEnd = 0;
int [] obsWrtFn = new int[32];
int winW = 1200;
int winH = winW * 10 / 16;

void setup(){
  size(winW, winH, P3D);
  int csvWidth = 0;
  String lines[] = loadStrings("syodo.csv");
  for (int i=0; i < lines.length; i++) {
    String [] chars = split(lines[i],',');
    if (chars.length > csvWidth){
      csvWidth = chars.length;
    }
  }
  csv = new String [lines.length][csvWidth];
  LENGTH =lines.length;
  println(LENGTH);
  for (int i=0; i < lines.length; i++) {
    String [] temp = new String [lines.length];
    temp= split(lines[i], ',');
    for (int j=0; j < temp.length; j++){
      csv[i][j]=temp[j];
    }
  }
  for ( int i=0; i<32; i++ ) { obsWrtFn[i] = -1; }
  frameRate(300);
  scanCSV();
  i=0;
  h=0;
  rec_x = ( Float.parseFloat(csv[0][1]) - dx ) * 3 - 600*2;
  rec_y = ( Float.parseFloat(csv[0][2]) - dy ) * 3 - 375*2;
  rec_z = ( Float.parseFloat(csv[0][3]) - dz ) * 3 + 600*2;
}

void draw(){
  if ( i == 0 ){
    background(255);
    camera( 200+mouseX*3, -1000+mouseY*4, 800,  400+mouseX*2, 200, -900,   0, 1, 0);
  }
  for( i=1; i< LENGTH; i++){
    if ( i > obsWrtFn[h] ) { h++; }
    dx = ( i - ( obsWrtFn[ h - 1 ] + 1 ) )
         * ( Float.parseFloat( csv[ obsWrtFn[ h ] ][1] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][1] ) )
         / ( Float.parseFloat( csv[ obsWrtFn[ h ] ][0] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][0] ) );
    dy = ( i - ( obsWrtFn[ h - 1 ] + 1 ) )
         * ( Float.parseFloat( csv[ obsWrtFn[ h ] ][2] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][2] ) )
         / ( Float.parseFloat( csv[ obsWrtFn[ h ] ][0] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][0] ) );
    dz = ( i - ( obsWrtFn[ h - 1 ] + 1 ) )
         * ( Float.parseFloat( csv[ obsWrtFn[ h ] ][3] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][3] ) )
         / ( Float.parseFloat( csv[ obsWrtFn[ h ] ][0] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][0] ) );
    dzx = ( i - ( obsWrtFn[ h - 1 ] + 1 ) )
         * ( Float.parseFloat( csv[ obsWrtFn[ h ] ][10] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][10] ) )
         / ( Float.parseFloat( csv[ obsWrtFn[ h ] ][0]  ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][0]  ) );
    dzy = ( i - ( obsWrtFn[ h - 1 ] + 1 ) )
         * ( Float.parseFloat( csv[ obsWrtFn[ h ] ][11] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][11] ) )
         / ( Float.parseFloat( csv[ obsWrtFn[ h ] ][0]  ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][0]  ) );
    dzz = ( i - ( obsWrtFn[ h - 1 ] + 1 ) )
         * ( Float.parseFloat( csv[ obsWrtFn[ h ] ][12] ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][12] ) )
         / ( Float.parseFloat( csv[ obsWrtFn[ h ] ][0]  ) - Float.parseFloat( csv[ obsWrtFn[ h - 1 ] + 1 ][0]  ) );
    x = ( Float.parseFloat(csv[i][1]) - dx ) * 3 - 600*2;
    y = ( Float.parseFloat(csv[i][2]) - dy ) * 3 - 375*2;
    z = ( Float.parseFloat(csv[i][3]) - dz ) * 3 + 600*2;
    wdL=sqrt((x-rec_x)*(x-rec_x)/100+(y-rec_y)*(y-rec_y)/100+(z-rec_z)*(z-rec_z)/100);
    zx = ( Float.parseFloat(csv[i][10]) -dzx ) * 120 * sq( 0.15 / ( wdL + 0.12 ) );
    zy = ( Float.parseFloat(csv[i][11]) -dzy ) * 120 * sq( 0.15 / ( wdL + 0.12 ) );
    zz = ( Float.parseFloat(csv[i][12]) -dzz ) * 120 * sq( 0.15 / ( wdL + 0.12 ) );
    if( x!=600 || y!=375 || z!=-600 ){
       if ( Float.parseFloat(csv[i][13]) ==1 ) {
        beginShape( QUADS );
        noStroke();
        fill( 0,   0,   0,   255*(1.3-wdL*7/3) );
        vertex(x+zx,  y+zy,  z+zz);
        vertex(x-zx,  y-zy,  z-zz);
        vertex(rec_x-rec_zx,  rec_y-rec_zy,  rec_z-rec_zz);
        vertex(rec_x+rec_zx,  rec_y+rec_zy,  rec_z+rec_zz);
        endShape();
      } else {
      }
    }
    rec_x = x;
    rec_y = y;
    rec_z = z;
    rec_zx = zx;
    rec_zy = zy;
    rec_zz = zz;
  }
  i=0;
  h=1;
}

void scanCSV() {
  int endWrt = 0;
  numWrite = 0;
  for ( int i = 2; i < LENGTH ; i++ ) {
    endWrt = (int)Float.parseFloat(csv[i][14]) - (int)Float.parseFloat(csv[i-1][14]);
    if ( endWrt == 1 ) {
      numWrite++;
      obsWrtFn[ numWrite ] = i;
    }
  }
}
// Copyright (C) 2014 ArduinoDeXXX All Rights Reserved.
```