```
import processing.opengl.*;
import processing.video.*;



//----GLOBAL VARIABLES----
String path;                      // this will hold the file path of the sketch
//MovieMaker MainMovie;             // this will record the main screen
int ImageCount = 0;               // used for calling the image files
int Factor = 4;

int ColorScale = 70000
int PixelCount = 0;
int W = 800                      // screen resolution
int H = 600
int MainMovieTimer = 0            // used to trigger saving the main movie
int MainMovieCounter = 0;
int ImageSaveCount = 0
PImage img;
char ActiveKey = 'R';
char ActiveKey2 = 'S';
int s= second();


//----MULTI THREAD----
Thread t1, t2;
//----MULTI THREAD----


//----CAMERA VARIABLES----
int CamSaveCount = 0;
//----CAMERA VARIABLES----

//----DISPLAY VARIABLES----
Display_Cam display;
int cx = W ;
int cy = (H / 2) * 7;
int cz = 100;
int tx = W;
int ty = H / 2;
int tz = 90;
//----DISPLAY VARIABLES----
//----GLOBAL VARIABLES----

//--------------------------------------------------------------------------------
void setup() {

  path = sketchPath("");                    // this is the folder directory of the sketch
  //println ("Sketch path: " + path);
  CreateFolders(path + "Pixel_Images");
  CreateFolders(path + "Cam_Images");
  //CreateFolders(path + "Movie");


  //img = loadImage("C:\\Users\\Crashnorun\\Documents\\Image Sketch\\Image_03\\_01_Save_Images_01\\Cam_Images\\Img_" + ImageCount + ".jpg");       // Get the image from the camera
  img = loadImage("Cam_Images\\Img_" + ImageCount + ".jpg");      //load the first image
  size(img.width/2, img.height/2, P3D);
  frameRate(15);
  background(0);
```

```
  //MainMovie = new MovieMaker(this, width, height,  "Movie\\Main_" + MainMovieCounter + ".mov");     // record the main screen

  directionalLight(255, 255, 255, 1, 1, -1);                                  //rgb values andx,y,z coord
  ambientLight(255, 255, 255);                                                //ambient light

  display = new Display_Cam(cx, cy, cz, tx, ty, tz, 0.0, 1.0, 0.0);           // Create a display camera

  cam = new SaveImg(this, W, H, path);
  t1 = new Thread(cam);
  t1.start();
  cam.run(CamSaveCount);

  //cam.SaveImages(CamSaveCount);
  CamSaveCount ++;
  //cam.SaveImages(CamSaveCount);
  cam.run(CamSaveCount);
  CamSaveCount ++;
}
//-----------------------------------------------------------------------------

//-----------------------------------------------------------------------------
void draw() {
  background(0);
  camera(display.CX, display.CY, display.CZ, display.TX, display.TY, display.TZ, display.UpX, display.UpY, display.UpZ);    // Use the display properties to set the camera
  img = loadImage("Cam_Images\\Img_" + ImageCount + ".jpg");

  //----SAVE IMAGES FROM CAMERA----
  cam.run(CamSaveCount);
  if (CamSaveCount > 100) {
    CamSaveCount = 0;
  } else {
    CamSaveCount ++;
  }
//----SAVE IMAGES FROM CAMERA----


if (key == '0') {                                    //  Number 2 button shows the image
  image(img, 0, 0);
  display.CX = img.width / 2;
  display.CY = img.height / 2;
  display.CZ = img.height;
  display.TX = img.width / 2;
  display.TY = img.height / 2;
  display.TZ = 0;
} else {

  img.loadPixels();
  PixelCount = 0;
  for (int i = 0; i < img.height; i += Factor) {
    for (int j = 0; j < img.width; j += Factor) {

      println("Pixel Count = " + PixelCount + " X= " + i + " Y= " + j);

      color c = color(red(img.pixels[PixelCount]), green(img.pixels[PixelCount]), blue(img.pixels[PixelCount]));
      stroke(c);
      fill(c);

      pushMatrix();
      translate(j*2, i*2, abs(c / ColorScale));
      rect(0, 0, Factor*2, Factor*2);
      popMatrix();
      PixelCount+=Factor;
```

```
      }           // close J loop
      PixelCount = i * img.width ;
    if (i == 4) {
      //System.exit(0);
    }
  }             // close I loop
}               // close keyprssed conditional statement

if (ImageCount >= 100) {
  ImageCount = 0;
} else {
  ImageCount++;
}


//----SAVE IMAGES----
if (second() % 2 == 0) {
  PImage img2 = createImage(width, height, RGB);
  //img2 = display.Image(width, height);
  img2.loadPixels();
  //println(img2.pixels[0]);
  save("Pixel_Images\\Image_" + ImageSaveCount + ".png");
  //img2.save("Pixel_Images\\Image_" + ImageSaveCount + ".png");

  if (ImageSaveCount >= 100) {
    ImageSaveCount = 0;
  } else {
    ImageSaveCount++;
  }
}
//----SAVE IMAGES----
  System.gc();
}
//--------------------------------------------------------------------------------



//--------------------------------------------------------------------------------
// Create the appropiate file paths
private void CreateFolders(String FilePath) {

  File folder = new File(FilePath);
  if (!folder.exists()) {                           //check if file path exists
    println("creating directory: " + FilePath);
    folder.mkdir();                                 //make the file path
  } else {
    println("Path: '" + FilePath + "' already exists");
  }
}
//--------------------------------------------------------------------------------


//--------------------------------------------------------------------------------
// When the space bar is pressed the main movie saves and exits processing
void keyPressed() {
  int temp;

  switch (key) {
  case ' ':
    exit();                                 // Quit running the sketch once the file is written
```

```
    break;

  case '+':                             // Zoom in
    temp = display.CY;
    display.CY = temp - 10;
    temp = display.TY;
    display.TY = temp - 10;
    //println(display.CY);
    break;

  case '-':                             // Zoom out
    temp = display.CY;
    display.CY = temp + 10;
    temp = display.TY;
    display.TY = temp + 10;
    //println(display.CY);
    break;

  case '1':                             // Reset camera to original position
    display.CX = cx;
    display.CY = cy;
    display.CZ = cz;
    display.TX = tx;
    display.TY = ty;
    display.TZ = tz;
    break;

  case '3':                             // Show original image
    image(img, 0, 0);
    break;

  case '4':                             // Alternate view
    display.CZ = 1000;
    break;

  case '5':                             // Alternate view
    display.CZ = 100;
    display.CY = 750;
    break;


  case '*':                             // this chanes the color height;
    temp = ColorScale;
    ColorScale = temp - 1000
    break;

  case '/':
    temp = ColorScale;
    ColorScale  = temp + 1000;
    break;

  case '2':                             // Reset color scale
    ColorScale = 70000;
    break;

  default:

    switch (keyCode) {
    case UP:                           // Move Camera Up
      temp = display.CZ;
      display.CZ = temp + 10;
      //println(display.CZ);
```

```
      break;

    case DOWN:                              // Move Camera Down
      temp = display.CZ;
      display.CZ = temp - 10;
      //println(display.CZ);
      break;

    case LEFT:                              // Move Camera Left
      temp = display.CX;
      display.CX = temp - 10;
      temp = display.TX;
      display.TX = temp - 10;
      //println(display.CX);
      break;

    case RIGHT:                            // Move Camera Right
      temp = display.CX;
      display.CX = temp + 10;
      temp = display.TX;
      display.TX = temp + 10;
      //println(display.CX);
      break;
    }
    break;
  }
  //println("key: " + key);
}
//------------------------------------------------------------------------------
```