

PLAGIARISM SCAN REPORT



Date 2020-03-12

Words 396

Characters 4770

Content Checked For Plagiarism

```
import processing.serial.*; // library imported for serial_communication import java.awt.event.KeyEvent; // library imported to read value from
serial_port import java.io.IOException; Serial my_port; String ang=""; String dis=""; String value=""; String n_object; float pixs_dis; int iang, idis;
int index_1=0; int index_2=0; PFont orcFont; void setup() { size (1200, 700); // size of the screen smooth(); my_port = new Serial(this,"COM5",
9600); // start serial communication from arduino to processing IDE my_port.bufferUntil('.'); // date read from serial port until character '.'.
ie.,angle,distance } void draw() { fill(98,245,31); // create blur_motion and slow_fade for the line noStroke(); fill(0,4); rect(0, 0, width, height-
height*0.065); fill(98,245,31); // green_color // function call for drawing the radar drawRadar(); drawLine(); drawObject(); drawText(); } void
serialEvent (Serial my_port) { // read the value from Serial Port // value is stored in the variable called "value". value =
my_port.readStringUntil('.'); value = value.substring(0,value.length()-1); index_1 = value.indexOf(","); // identifies the character ',' and stores in
variable called "index_1" ang= value.substring(0, index_1); // the value from 0th position to position of index_1 is measured dis=
value.substring(index_1+1, value.length()); // the value of distance is measured // String variables are converted into Integer variables iang =
int(ang); idis = int(dis); } void drawRadar() { pushMatrix(); translate(width/2,height-height*0.074); // move the coordinates to a new_location
noFill(); strokeWeight(2); stroke(98,245,31); // the arc lines are drawn arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);
arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI); arc(0,0,(width-width*0.479),(width-width*0.479),PI,TWO_PI); arc(0,0,(width-
width*0.687),(width-width*0.687),PI,TWO_PI); //the angle lines are drawn line(-width/2,0,width/2,0); line(0,0,(-width/2)*cos(radians(30)),(-
width/2)*sin(radians(30))); line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60))); line(0,0,(-width/2)*cos(radians(90)),(-
width/2)*sin(radians(90))); line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120))); line(0,0,(-width/2)*cos(radians(150)),(-
width/2)*sin(radians(150))); line((-width/2)*cos(radians(30)),0,width/2,0); popMatrix(); } void drawObject() { pushMatrix();
translate(width/2,height-height*0.074); // move the coordinates to a new_location strokeWeight(9); stroke(255,10,10); // red color pixs_dis =
idis*((height-height*0.1666)*0.025); // covers the dis from sensor from centimetres to pixls //the range is limited to 40 cms using if condition
if(idis<40){ // the object is drawn according to the value of ang and the distance measured line(pixs_dis*cos(radians(iang)),-
pixs_dis*sin(radians(iang)),(width-width*0.505)*cos(radians(iang)),(-width-width*0.505)*sin(radians(iang))); } popMatrix(); } void drawLine() {
pushMatrix(); strokeWeight(9); stroke(30,250,60); translate(width/2,height-height*0.074); // move the coordinates to a new_location
line(0,0,(height-height*0.12)*cos(radians(iang)),(-height-height*0.12)*sin(radians(iang))); //the angle lines are drawn popMatrix(); } void
drawText() { //the text are drawn pushMatrix(); if(idis>40) { n_object = "Out_of_Range"; } else { n_object = "In_Range"; } fill(0,0,0); noStroke();
rect(0, height-height*0.0648, width, height); fill(98,245,31); textSize(25); text("10cm",width-width*0.3854,height-height*0.0833);
text("20cm",width-width*0.281,height-height*0.0833); text("30cm",width-width*0.177,height-height*0.0833); text("40cm",width-
width*0.0729,height-height*0.0833); textSize(40); text(" VIRAL SCIENCE ", width-width*0.875, height-height*0.0277); text("Angle: " + iang + " °",
width-width*0.48, height-height*0.0277); text("dis: ", width-width*0.26, height-height*0.0277); if(idis<40) { text(" " + idis + " cm", width-
width*0.225, height-height*0.0277); } textSize(25); fill(98,245,60); translate((width-width*0.4994)+width/2*cos(radians(30)), (height-
height*0.0907)-width/2*sin(radians(30))); rotate(-radians(-60)); text("30",0,0); resetMatrix();
translate((width-width*0.503)+width/2*cos(radians(60)), (height-height*0.0888)-width/2*sin(radians(60))); rotate(-radians(-30)); text("60",0,0);
resetMatrix(); translate((width-width*0.507)+width/2*cos(radians(90)), (height-height*0.0833)-width/2*sin(radians(90))); rotate(radians(0));
text("90",0,0); resetMatrix(); translate(width-width*0.513+width/2*cos(radians(120)), (height-height*0.07129)-width/2*sin(radians(120)));
rotate(radians(-30)); text("120",0,0); resetMatrix(); translate((width-width*0.5104)+width/2*cos(radians(150)), (height-height*0.0574)-width/
2*sin(radians(150))); rotate(radians(60));text("150",0,0);popMatrix();}
```

Matched Source

No plagiarism found

