

```
/*
```

The ant coordinate game - for PCS 5th grade

K. Davis & Z. Wood 2014

Students must change the value of the points (x1, y1) thru (x5, y5) in order to make the ant pick up all the crumbs. Note the ant always travels in x direction first then y and the ant will only move to the crumb if the values are correct (otherwise the incorrect path is shown to help students figure out the coordinates). See TODO instructions below

```
*/
```

```
void setXY() {
```

```
/*
```

TODO Change the value of the points x and y coordinates so the ant reaches all the crumbs on the grid. The ant must pick up the crumbs in a certain order starting at the origin:

#1) (x1, y1) upper left quadrant (white) ->

#2) (x2, y2) upper right quadrant (pink) ->

#3) (x3, y3) bottom right quadrant (green) ->

#4) (x4, y4) bottom left quadrant (blue) ->

#5) (x5, y5) bottom right corner of the window

Change the value of the variables below one at a time and run the code to confirm you got the right answer - the ant will only dance when each crumb's coordinates are correct

```
*/
```

```
x1 = 0;
```

```
y1 = 0;
```

```
x2 = 0;
```

```
y2 = 0;
```

```
x3 = 0;
```

```
y3 = 0;
```

```
x4 = 0;
```

```
y4 = 0;
```

```
x5 = 0;
```

```
y5 = 0;
```

```
}
```

```
boolean reset = true;
boolean hard = false;
```

```
/****** Do not change anything in the file below this line *****/
```

```
int[] list;
int wdh = 400, hght = 400;
int x1, y1, x2, y2, x3, y3, x4, y4, x5, y5;
int transX = 0, transY = 0;
int curCrumb = 1;
int rotateAng = 0;
int ellipseNum = 1;
int numC = 5;
boolean disco;

color qs[] = {
  color(250, 250, 250, 128), color(250, 150, 150, 128), color(150, 250, 150, 128),
  color(150, 150, 250, 128)
};
void setup() {
  list = new int[10];

  size(400, 400);
  background(255);

  setXY();

  if (!hard)
    genListEasy();
  else
    genListHard();
  list[8] = wdh;
  list[9] = hght;

  drawGrid();
```

```
drawAnt();  
disco = false;  
}
```

```
void drawGrid() {  
  noStroke();  
  fill(qs[0]);  
  rect(0, 0, 200, 200);  
  
  fill(qs[1]);  
  rect(200, 0, 200, 200);  
  
  fill(qs[2]);  
  rect(0, 200, 200, 200);  
  
  fill(qs[3]);  
  rect(200, 200, 200, 200);  
  
  strokeWeight(1);  
  //the horizontal lines  
  stroke(0, 0, 204);  
  line(0, 50, width, 50);  
  line(0, 100, width, 100);  
  line(0, 150, width, 150);  
  line(0, 200, width, 200);  
  line(0, 250, width, 250);  
  line(0, 300, width, 300);  
  line(0, 350, width, 350);  
  //the vertical lines  
  stroke(10, 10, 10);  
  line(50, 0, 50, height);  
  line(100, 0, 100, height);  
  line(150, 0, 150, height);  
  line(200, 0, 200, height);  
  line(250, 0, 250, height);  
}
```

```
line(300, 0, 300, height);  
line(350, 0, 350, height);  
}
```

```
void drawCrumbs() {  
  int j = 0;  
  stroke(0);  
  strokeWeight(1);  
  for (int i = 0; i < 10; i++) {  
    fill(0);  
    //text("x"+ (j+1) + " y"+(j+1), list[i]+10, list[i+1]+10);  
    fill(qs[j%4]);  
    ellipse(list[i++], list[i], 10, 10);  
    j++;  
  }  
}
```

```
void drawAnt() {  
  pushMatrix();  
  
  translate(transX, transY);  
  rotate(radians(rotateAng));  
  if (disco)  
    scale(random(0.3, 0.7));  
  else  
    scale(.5);  
  translate(100, -100);  
  rotate(radians(90));  
  
  strokeWeight(1);  
  stroke(0);  
  if (disco)  
    fill(random(20, 120), random(30, 140), random(10, 160));  
  else  
    fill(0);
```

```
//body
ellipse(100, 76, 28, 28);
ellipse(100, 102, 16, 30);
ellipse(100, 128, 36, 40);

//head
ellipse(100, 46, 28, 32);
fill(255);
//antennas
bezier(104, 32, 118, 28, 110, 22, 122, 18);
bezier(96, 32, 82, 28, 90, 22, 78, 18);
//eyes
ellipse(90, 38, 10, 14);
ellipse(110, 38, 10, 14);
fill(0);
ellipse(90, 36, 4, 6);
ellipse(110, 36, 4, 6);

//legs
noFill();
strokeWeight(2);
bezier(102, 90, 132, 76, 130, 76, 142, 92);
bezier(98, 90, 68, 76, 70, 76, 58, 92);
bezier(110, 70, 124, 60, 128, 60, 132, 34);
bezier(90, 70, 76, 60, 72, 60, 68, 34);
bezier(108, 102, 134, 122, 126, 120, 132, 148);
bezier(92, 102, 66, 122, 74, 120, 68, 148);

popMatrix();
}
```

```
void antDance(int finalX, int finalY) {
  if (transX - finalX < 0)
    transX += 2;
  else
    transX -=2;
}
```

```

if (transY - finalY < 0)
    transY += 2;
else
    transY -=2;
}

void antMarch(int curX, int curY, int crumbX, int crumbY, int stuX, int stuY) {
    if (stuX == crumbX && stuY == crumbY) {
        if (transX == crumbX && transY > crumbY - 45) { //make crumb disappear
            ellipseNum = 0;
        }
        else if (transY == crumbY && transX > crumbX - 45) { //make crumb disappear
            ellipseNum = 0;
        }
        if (transX == crumbX && transY == crumbY) { //switch to next crumb
            curCrumb++;
            ellipseNum = curCrumb;
            if (curCrumb == 3)
                rotateAng = 180;
            else
                rotateAng = 0;
        }
        else if (transX == crumbX) { //move in y direction
            rotateAng = 90;
            transY += 2;
        }
        else if (rotateAng == 0) { //move in x direction
            transX += 2;
        }
    }
    else {
        strokeWeight(3);
        stroke(255, 0, 0);
        line(curX, curY, stuX, curY);
        line(stuX, curY, stuX, stuY);
    }
}

```

```
}
```

```
void antMarchReverse(int curX, int curY, int crumbX, int crumbY, int stuX, int stuY) {  
  if (stuX == crumbX && stuY == crumbY) {  
    if (transX == crumbX && transY > crumbY - 45) { //make crumb disappear  
      ellipseNum = 0;  
    }  
    else if (transY == crumbY && transX < crumbX - 45) { //make crumb disappear  
      ellipseNum = 0;  
    }  
    if (transX == crumbX && transY == crumbY) { //switch to next crumb  
      curCrumb++;  
      ellipseNum = curCrumb;  
      rotateAng = 0;  
    }  
    else if (transX == crumbX) { //move in y direction  
      rotateAng = 90;  
      transY += 2;  
    }  
    else if (rotateAng == 180) { //move in x direction  
      transX -= 2;  
    }  
  }  
  else {  
    strokeWeight(3);  
    stroke(255, 0, 0);  
    line(curX, curY, stuX, curY);  
    line(stuX, curY, stuX, stuY);  
  }  
}
```

```
void draw() {  
  background(255);  
  drawGrid();
```

```
  if (x1 == 0 && y1 == 0 && x2 == 0 && x3 == 0 && x4 == 0) {
```

```

    drawCrumbs();
}

if (curCrumb == 1) {
    antMarch(0, 0, list[0], list[1], x1, y1);
}
else if (curCrumb == 2) {
    antMarch(list[0], list[1], list[2], list[3], x2, y2);
}
else if (curCrumb == 3) {
    antMarchReverse(list[2], list[3], list[4], list[5], x3, y3);
}
else if (curCrumb == 4) {
    antMarch(list[4], list[5], list[6], list[7], x4, y4);
}
else if (curCrumb == 5) {
    antMarch(list[6], list[7], list[8], list[9], x5, y5);
}
else {
    if (transX != width/2) {
        rotateAng = -135;
        antDance(width/2, height/2);
    }
    else if (transX == width/2 && transY == height/2) {
        rotateAng += 4;
        disco = true;
        frameRate(10);
    }
}

stroke(0);
strokeWeight(1);
if (ellipseNum == 1) {
    fill(0);
    //text("x1 y1", list[0]+10, list[1]+10);
    fill(qs[0]);
}

```



```

    ellipse(list[0], list[1], 10, 10);
}
else if (ellipseNum == 2) {
    fill(0);
    //text("x2 y2", list[2]+10, list[3]+10);
    fill(qs[1]);
    ellipse(list[2], list[3], 10, 10);
}
else if (ellipseNum == 3) {
    fill(0);
    //text("x3 y3", list[4]+10, list[5]+10);
    fill(qs[2]);
    ellipse(list[4], list[5], 10, 10);
}
else if (ellipseNum == 4) {
    fill(0);
    //text("x4 y4", list[6]+10, list[7]+10);
    fill(qs[3]);
    ellipse(list[6], list[7], 10, 10);
}
else if (ellipseNum == 5) {
    fill(0);
    //text("x5 y5", list[8]-40, list[9]-10);
    fill(qs[0]);
    ellipse(list[8], list[9], 10, 10);
}

drawAnt();
}

/*****

```

GO BACK UP TO THE TOP

GO BACK UP TO THE TOP

GO BACK UP TO THE TOP

GO BACK UP TO THE TOP

GO BACK UP TO THE TOP

*/

```
void genListEasy() {  
    list[0] = 100; list[1] = 50; list[2] = 350;  
    list[3] = 150; list[7] = 350;  
    list[4] = 100; list[5] = 250; list[6] = 250;  
}
```

```
void genListHard() {  
    list[0] = 30; list[7] = 370; list[1] = 110; list[6] = 380;  
    list[2] = 340; list[5] = 260; list[3] = 180; list[4] = 90;  
}
```