Modeling Polygons in Processing: Adding drawPolygon() function to Actor Class

Mr. Michaud
Marist School
Modeling Polygons

• Allows programmer / animator to draw any shape.
• Allows for rotation and scaling of shape.
• Uses a ‘radius’ and trigonometry to plot vertices.
• We will create a drawPolygon() Function
  – Draw any polygon
  – Translation, Scaling, Rotation
  – Line and Fill Colors
Drawing Irregular Shapes

beginShape(); // start drawing polygon
    vertex(x1, y1); // Vertices of polygon
    vertex(x2, y2);
    vertex(x3, y3);
    vertex(x4, y4);
    vertex(x5, y5);
endShape(CLOSE); // Close and color polygon

The beginShape() and endShape() functions allow us to draw any polygon shape as long as we calculate or define the vertices.
Calculating the Vertices

The radius is like a Segment – the vertices are the endpoints calculated from angles and length of radius.

Properties:
- x, y position
- radius $\rightarrow r$
- direction $\rightarrow d$
- sides $= 4$

Examples:

\[
\begin{align*}
    x_1 &= x + r \cdot \cos \left( \alpha - a \right) \\
    y_1 &= y + r \cdot \sin \left( \alpha - a \right) \\
    x_2 &= x + r \cdot \cos \left( \alpha + d \right) \\
    y_2 &= y + r \cdot \sin \left( \alpha + d \right)
\end{align*}
\]

How would you complete this pattern?
Calculating Values for Angle \( a \)

\[
a = \frac{360}{\text{sides}}
\]
Add Field to Actor

// For Polygons
float heading = 0;
Algorithm for drawPoly()

- Create 2D array to hold the x and y of vertices
- Calculate the angle ‘a’ by \( a = \frac{360}{\text{sides}} \)
- For loop counting by sides
  - add \( a \times \text{index} + d \) (direction)
  - calculate new x and y and place in 2D array
- Set fill
- Set stroke
- beginShape()
- For loop to call the vertex() function with 2D Array
- endShape()
Function: drawPolygon(int sides)

public void drawPolygon(int sides) {
    // Calculate the vertices
    float points [][] = new float [sides][2];
    float a = 360/sides;
    for (int p = 0; p < sides; p++) {
        float angle = heading + (p*a);
        points[p][0] = x + size*cos(radians(angle));
        points[p][1] = y + size*sin(radians(angle));
    }

    // Draw Shape
    fill(fillRed, fillGreen, fillBlue);
    stroke(penRed, penGreen, penBlue);
    strokeWeight(lineWidth);
    beginShape();
    for (int v = 0; v < sides; v++) {
        vertex(points[v][0], points[v][1]);
    }
    endShape(CLOSE);
} // end drawPolygon
public void drawLine() {
    float x2 = x + size*cos(radians(heading));
    float y2 = y + size*sin(radians(heading));
    stroke(penRed, penGreen, penBlue);
    strokeWeight(lineWidth);
    line(x, y, x2, y2);
} // end drawLine