CS/COE 1501

cs.pitt.edu/~bill/1501/

Point in Polygon
Determining if a point is in/out of a polygon

Is (5, 8) inside or outside the polygon?
Raytracing (Point inside example)

- (5, 9) Crossed 1x
- (7, 6)
- (6, 4) Crossed 3x
- (9, 9) Crossed 1x
- (2, 3) Crossed 1x
- (11, 2)
Raytracing (Point inside example)

(2, 3) → (6, 4) → (5, 9) → (9, 9) → (11, 2)

- Crossed 4x
- Crossed 2x
- Crossed 0x
How can we implement raytracing?

- Assume we are given:
  - The point to check
  - An array of n points that make up the vertices of the polygon
    - In order so that vertices next to one another in the array are the endpoints of an edge of the polygon
Raytracing approach
Typically slopes are rise/run

\[ x \times \text{(rise/run)} \text{ yields vertical distance line will travel after } x \text{ horizontal units} \]

\[ \text{Hence } y = x \times \text{(rise/run)} + y\text{-intercept is the equation for a line} \]

\[ y \times \text{(run/rise)} \text{ yields horizontal distance line will travel after } y \text{ vertical units} \]

\[ \text{Hence } (a[i].y - p.y) \times \text{(run/rise)} + a[i].x \text{ gives x-intercept at y coordinate } p.y \]
public boolean contains2(Point p) {
    int crossings = 0;
    for (int i = 0; i < N; i++) {
        int j = i + 1;
        boolean cond1 = (a[i].y <= p.y) && (p.y < a[j].y);
        boolean cond2 = (a[j].y <= p.y) && (p.y < a[i].y);
        if (cond1 || cond2) {
            if (p.x < (a[j].x - a[i].x) * (p.y - a[i].y) / (a[j].y - a[i].y) + a[i].x)
                crossings++;
        }
    }
    if (crossings % 2 == 1) return true;
    else return false;
}

- Point has defined x and y attributes
- a is an array of N-1 different vertices (as Point objects)
Another raytracing implementation (in C)

int pnpoly(int nvert, float *vertx, float *verty, float testx, float testy) {
    int i, j, c = 0;
    for (i = 0, j = nvert-1; i < nvert; j = i++) {
        if ( ((verty[i]>testy) != (verty[j]>testy)) &&
             (testx < (vertx[j]-vertx[i]) * (testy-verty[i])
              / (verty[j]-verty[i]) + vertx[i]) )
            c = !c;
    }
    return c;
}
Raytracing oddity