

1 Winding Number Algorithm

The winding number algorithm was the algorithm that we first built to test insideness. Let C represent a polygon with n number of vertices. Let V_i be any vertex of C where $0 \leq i \leq n$. We define the winding number as a function W . A point $P = (x, y)$ is inside the polygon C , if for all V_i on C :

$$W(P, C) = \frac{1}{2\pi} \sum_{i=1}^{n-1} \arccos \frac{(V_i - P) \cdot (V_{i+1} - P)}{|V_i - P| \cdot |V_{i+1} - P|}$$

If a point is inside the polygon C then the winding number has to be a non-zero integer. We decided to use this algorithm as a way to approximate a circle by increasing the amount of vertices on the polygon.

2 Ray Casting Algorithm

The ray casting algorithm at its core works by taking a given point Q and extending that point in the positive x direction. Lets call that ray \vec{Q} . If $Q = (a, b)$ where a and b are constants, then $\vec{Q} = \langle x, b \rangle$ where $x \in R$. Let C be a polygon with n vertices, each line segment formed by connecting vertices we consider to be "walls". If Q lies outside the polygon, the \vec{Q} will either not intersect the polygon at all, or there will be at least two intersections which cancel each other out. To cancel each other out means that an intersection will occur when the ray enters the shape and there will be an exiting intersection. Where as, a point that is inside the shape could intersect the shape in multiple places, although there will only be one exiting intersection and no entry intersection. Thus, the algorithm thinks about this as a index. Let's call this index I , I is an integer which will incremented by one on an entry and decremented when exiting the shape. I if a point lies outside the shape I will be zero. A point is inside the shape when I is not zero.

2.1 Bounding Box Implementation

The pure ray casting algorithm's issues arise when the amount of vertices becomes large. A large number of vertices slow down the algorithm too much. The Ray casting algorithm was designed for polygons of smaller amounts of vertices. We needed a scalable algorithm like the winding number to compute insideness. Thus, we needed to adjust the ray-casting algorithm. Rather than extending the a point Q only in the positive x direction, we extend the point in both directions. An intersection from an extended point in the negative x direction will subtract one from the intersection index. Where as an intersection from the positive x direction adds one to the intersection index. Thus, a point lies within a polygon if the intersection index is zero. We also throw out all points that are greater than the maximums of the shape or lower than the minimums of the polygons. Throwing out those points creates a box around the shape. Points within that box will be the only points tested.

References