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 vertice of C where $0 \leq i \leq n$. We define the winding number as a function f. A point $P = (x, y)$ where $x, y \in R$ is inside the polygon C, if for all V_i on C:

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

If a point is inside the curve C then the winding number has to be greater than or equal to 1.

2 Ray Casting Algorithm

The ray casting algorithm works by extening a given point Q to the right infinitly. Lets call that ray $\overrightarrow{\mathrm{Q}}$. The approach here is to think of one point as an infinite ray. If the point lies outside the polygon, that implies that the sum of intersections that occured should add up to zero. If the intersection index is anything but zero, then the point is outside the polygon. A intersection index is defined as a counter which counts the intersections. An intersection from the right adds one to the intersection index. An intersection from the left subtracts one from the sum. Therefore, the sum of the intersections of a point inside the polygon is 0. Whereas, the intersections are not zero of a point lying outside the circle. Thus, lets take two points from the curve C, lets call the points P_1 and P_2. P_1 has to be greater in some context. For example, if we have a parameterized circle with one variable t, and P_1 is defined at t_1 and P_2 is defined at t_2 then the algorithm requires that $t_1 > t_2$. Thus, the line segment created from the points P_1 and P_2 is $\overline{\mathrm{P_1P_2}}$. So, lets say that the infinite ray $\overrightarrow{\mathrm{Q}}$ intersects the line segment $\overline{\mathrm{P_1P_2}}$, then if that is the only intersection, the point Q is inside the curve C. However, if there exists a line segment, $\overline{\mathrm{P_iP_j}}$ where $\overrightarrow{\mathrm{Q}}$ intersects the curve C again, then the point lies outside of the C.

2.1 Ray Casting Implementation: Bounding-Box Testing

Ray casting is powerful for a polygon with a small amount of edges. A circle is a infinitly edged shape, when trying to parameterize a circle in a finite environment the ray casting alogrithm has problems. Depending on the size of the finite domain, the ray casting algorithm's accuracy varies. A solution lies in a implementation [1] which takes the ray casting algorithm and applies a faster intersection index. The intersection index is a number that counts how many intersections occured between the extended ray $\overrightarrow{\mathrm{Q}}$ and $\overline{\mathrm{P_iP_j}}$. This implementation differs by restricting the domain to a smaller size based on a user defined box. Now, the ray $\overrightarrow{\mathrm{Q}}$ will only check intersections within that bouding boxes. This implementation allows for us to be able to focus on a smaller area while increasing the points.

References

[1] A. Mathias, U. Kanther, and R. Heidger. Insideness and collision detection algorithms. In *2008 Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles*, pages 1–7, Sep. 2008.