BCA 608 Final Assignment

Bugrahan Akbulut

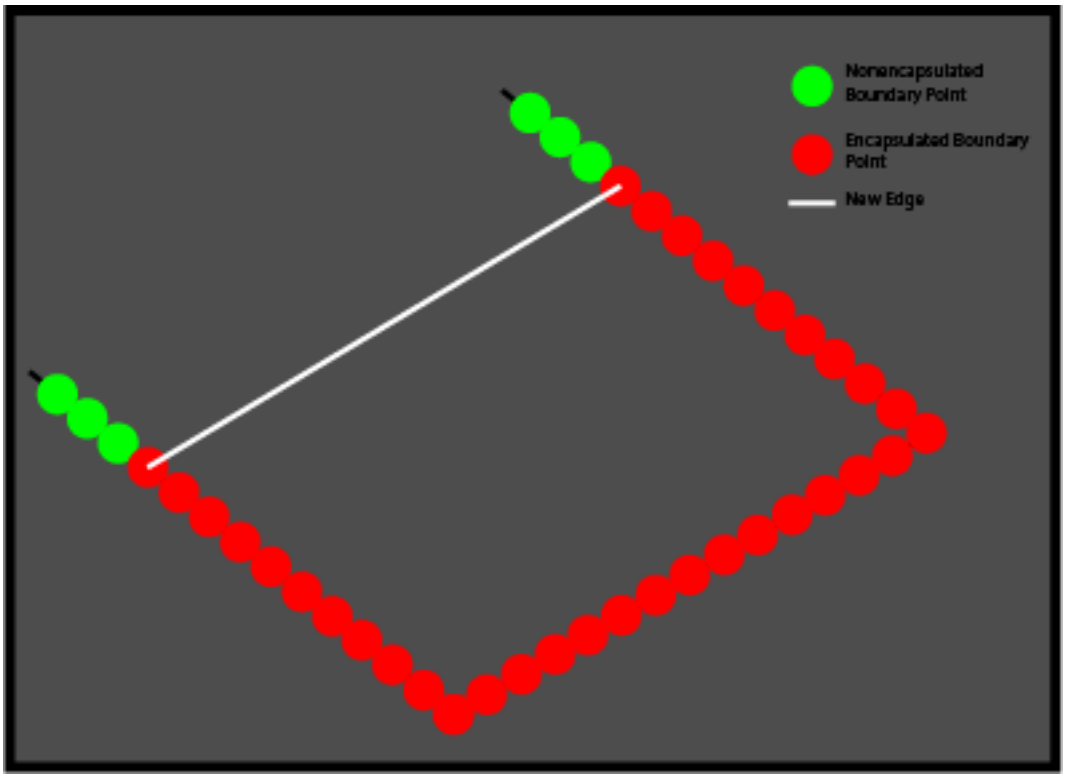
January 20, 2021

1 Introduction and Overview

When I was implementing this assignment, I used Unity2D with C#.
I used Rider JetBrains for IDE. To keep project clean and solid I put project on **github.** To avoid any kind of cheating I kept repository private until due date.

2 Implementation Details

Firstly I build polygon according to the user input, then I created boundary points in edges and for each boundary point by using ray-casting(from boundary point to every direction) determined every boundary point is encapsulated or not. After found every boundary points' encapsulation information by traversing them I found encapsulated part of polygon. It is working as by following this steps :
1. Found encapsulated boundary point
2. Until found non-encapsulated boundary traverse boundary points
3. When found non-encapsulated boundary point add new edge between the encapsulated boundary point (from Step 1) and previous boundary point
4. Determine encapsulated inputs by using winding points algorithm



3 Solution Analysis

In first step I used distance based method to create boundary points. I leave some space between each boundary point then calculated their encapsulation information by ray-casting in every direction(360°). This step has complexity $O(n * 360) = O(n)$(n is count of total boundary points). In second and third steps has total complexity $O(n)$. In fourth step has complexity $O(E)$ E refers total edge counts in final encapsulation polygon. So total complexity is

$$O(n) \ + \ O(n) \ + \ O(E) = O(2n \ + \ E) = O(n)$$