BCA 608 Assignment 2

Bugrahan Akbulut

December 25, 2020

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.

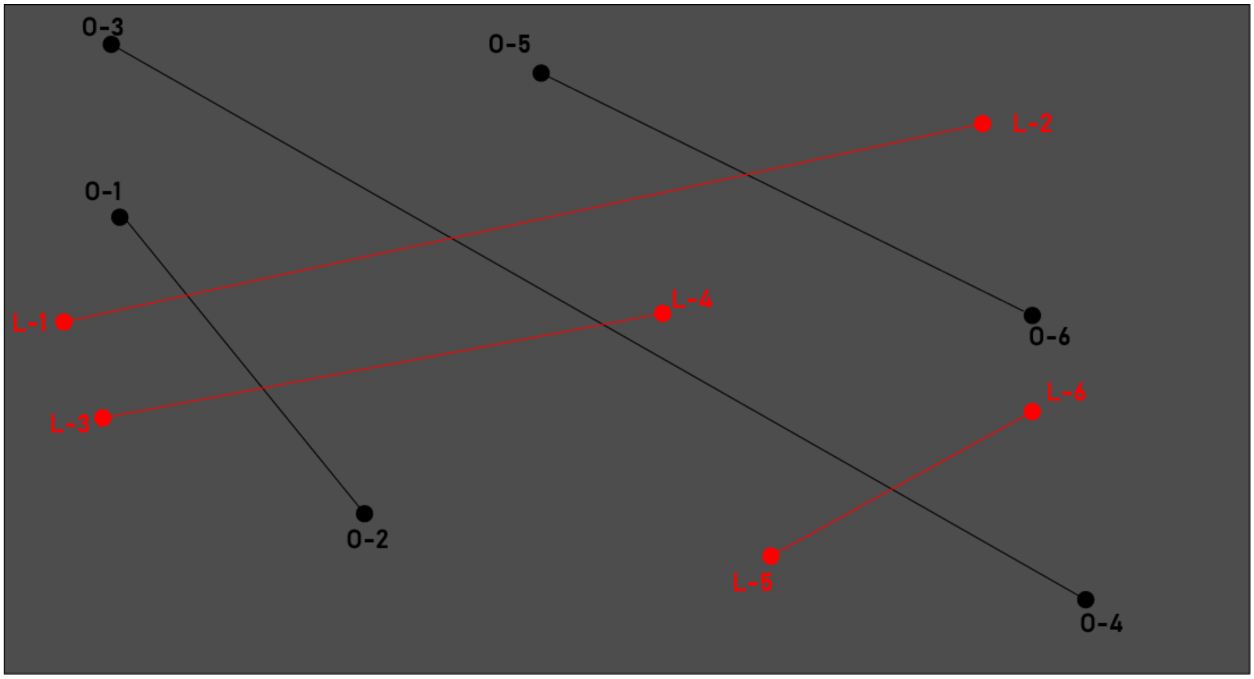
2 Implementation Details

To implement this assigned I created following structers : Input Controller, Triangulation Controller and Path Controller. As their names represented their duties Input Controller controls input creating obstacles by mouse and skipping states by space key, Triangulation Controller controls triangulation steps and lastly path controller creating walkable paths and calculates shortest path Starting point to Destination point.

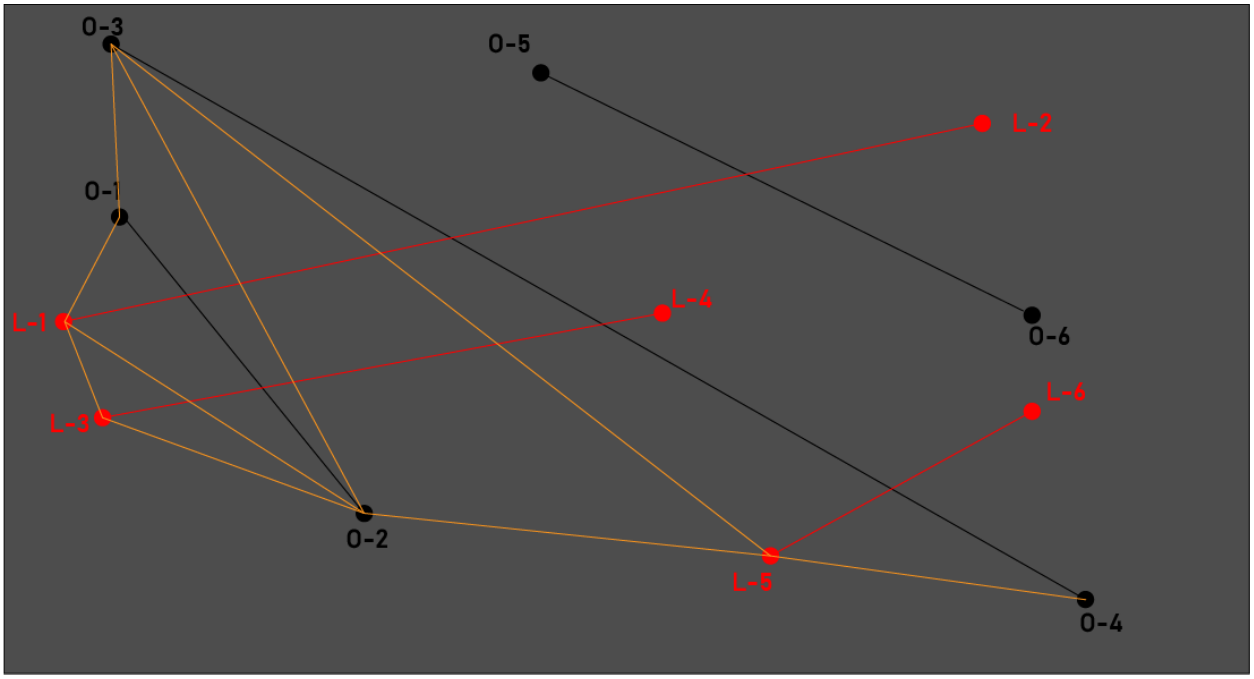
In my program I apply Delaunay Triangulation firstly by using obstacle start, finish points and application window corner points. After that my program controls is there any obstacle intersecting with triangulation lines. If there is no intersection this means obstacle lines becomes part of triangulation. Otherwise triangulation lines are crossing with the obstacles and we should avoid that to create proper paths which are not passing by obstacle lines. To get proper triangulation I implemented following algorithm :
1. Detect Intersected Obstacles and Triangulation Lines
2. Control for each Triangulation line intersected multiple obstacle if line crossed with multiple obstacle gather all other obstacle and their intersections.
3. Order all gathered obstacle left to right.
4. For each triangulation line (which are involved intersection with these obstacles) categorize them according to their area(by saying area I mean since there is minimum one obstacle, it divides points in two different area: left of this obstacle and right of this obstacle).
5. Triangulate every area by itself.

To be explain algorithm in more details we could investigate following figure. There three obstacles and First line (from L-1 to L-2) intersecting with all the obstacles. According to the my algorithm it firstly controls for obstacles and intersected triangulation lines Obstacle-1 intersecting with Line-1 and this line crosses all the other obstacles. Which means my algorithm gather all the obstacles and their intersected triangulation line and created segments. In first segment there are four points which are L-1, L-3, O-1 and O-2, in second segment there are five points O-1, O-2, O-3, O-4 and L-5 and so on. after created segments it start to triangulate every segment by it self and removes intersected triangulation lines and get clean triangulation in these areas.

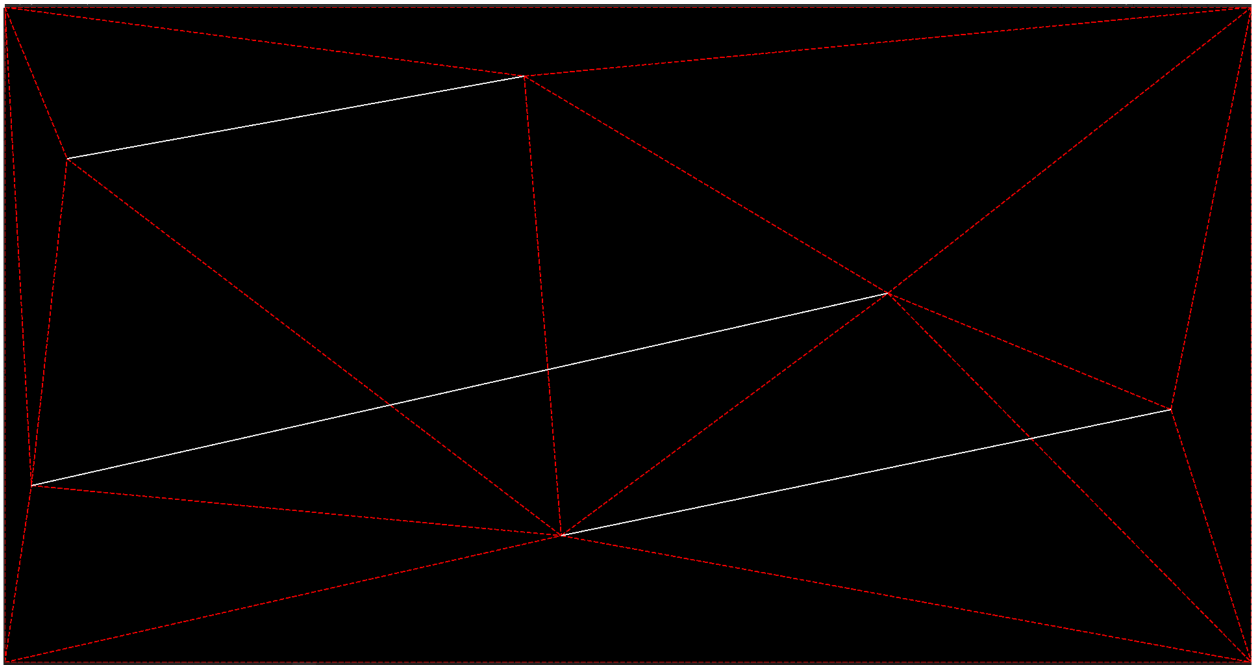
(a) Dividing Area



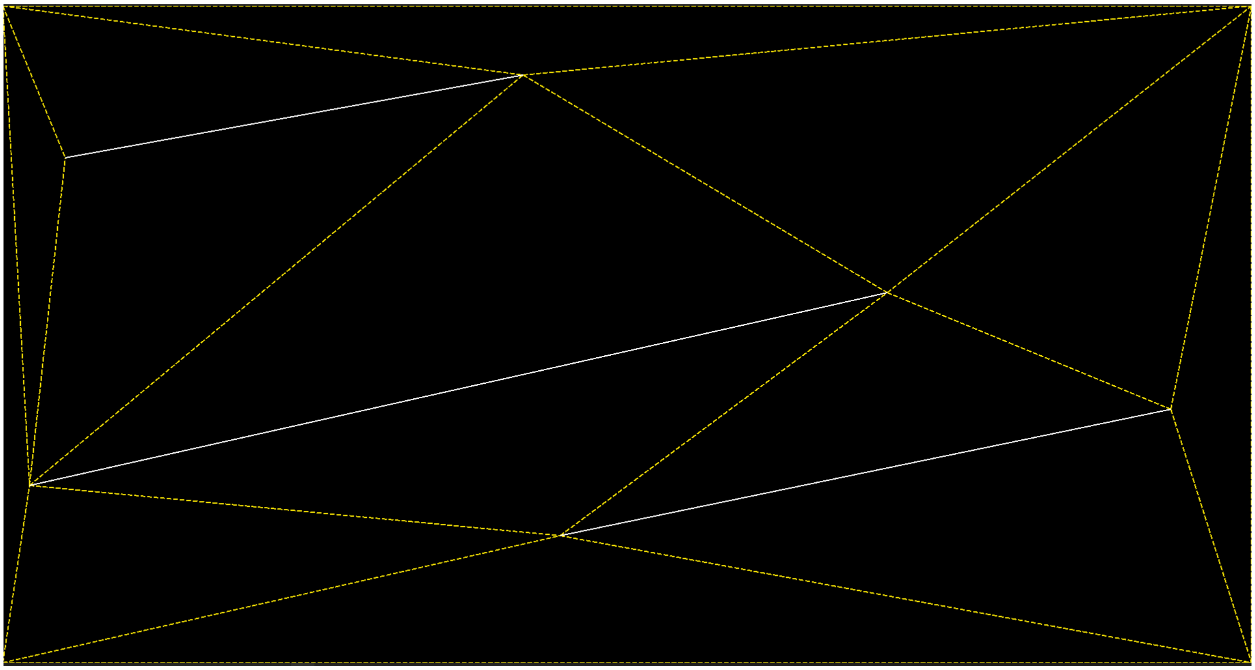
(b) After triangulation segment 0 and 1



(a) Red lines in here result of Vanilla Deluanay Triangulation



(b) Yellow lines in here result after applied my algorithm into Deluanay Triangulation



3 Solution Analysis

First Deluanay Triangulation has complexity $O(nlog(n))$. After that controls for intersection between obstacles and triangulation has complexity is $O(n \times v)$. After intersection test we need to triangulate every segment again in worst case there are n intersected obstacles which means (n+1) segments than its complexity $O(n^2log(n))$. Building triangles has complexity $O(n^3)$. To find shortest path I used Djiskstra's Shortest Path Algorithm complexity : $O(n^2)$ Than we have complexity : $O(n^3)$