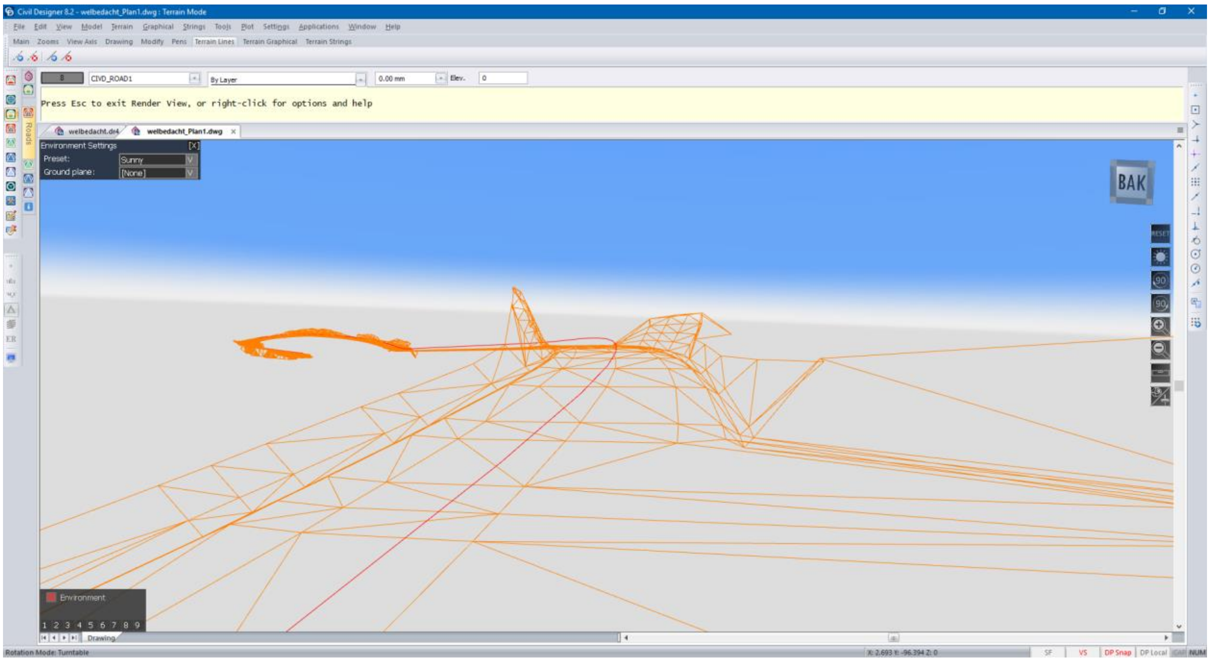
# How far can you see along a road?

## Problem Description:

Road designers have to design roads in such a way that the driver of a vehicle traveling on the road, can always see a minimum distance in front of him. His vision can be obstructed by a curve in the road, cut banks or by crested vertical curves.

## Inputs:



Civil Designer 8.2 - welbedacht_Plan1.dwg : Terrain Mode

Press Esc to exit Render View, or right-click for options and help

Environment Settings

Preset: Sunny

Ground plane: [None]

1. A CSV file containing triangle data **“TIN Model Road Surface.csv”**. The triangles defines the surface of the road (Orange triangles in the above image). Each row in the CSV file represents one triangle i.e. 3 x 3D Coordinates (9 X floating point values). All coordinates are in metres.
2. A CSV file containing a list of 3D coordinates (3D polyline) **“polyline.csv”**. This polyline defines the route by which the driver’s eye will be traveling along the road (See the Red line in the above image). To simplify the problem, the object which the driver must be able to see will also be on the same line.
3. A DXF drawing file (welbedacht_Plan1.dxf) for your reference. This drawing contains the above data in a drawing form for your own visualization and verification purposes. You can view this in any CAD program, or download the AllyCAD student version.

## Task:

Write a C++ program that will output the areas along the road where the driver cannot see 350m in front of him, so let:

C The polyline described in 2, contained in **“polyline.csv”**

S The Start of the road – first vertex of C

A The position of the driver’s eyes.

B Be an object on the road which the driver must be able to see.

AB The distance from A to B, along the polyline C

L The Sight line, so a straight line between A and B

T Any triangle on the road surface as described in 1, contained in **TIN Model Road Surface.csv”**

IP Intersecting point with the road surface

## Algorithm:

1. Let A follow the polyline, so iterate through the vertices of C, starting at S
2. For each position of A, let B move further along the vertices of C – until AB > 350m.
3. Check if there is an intersection of L with all the triangles T
4. Whenever you find a T intersection (IP) with L, report the IP as well as the distance AB along the polyline.

## Output:

1. For every IP that you find, report AB.

## Submission:

A successful submission should include:

1. C++ Source code that:
    a. Compiles, runs and outputs the correct results.
    b. Display Proper use of C++ principles.
2. Documentation:
    a. What compiler was used?
    b. User guide (how to compile and run the program)
    c. Algorithm description.
    d. List of sources used for research.