Project report

Prepared by:Naman jain(160050032) ,Utkarsh gupta(160050032),Sharvik mital(160050059).

25 April 2017

1).**Description of the problem**:- Our project aims at solving the art gallery problem where the cameras are allowed only at vertices for simple polygons .

The Original Art Gallery Problem :-Given the layout (map) of an art gallery what is the minimum number of stationary guards (CC TV cameras) needed to guard every point in the art gallery.

We use art gallery theorem as a guideline.

The Art Gallery Theorem (AGT) :-For any gallery with n walls [n/3] cameras are always sufficient to guard the gallery.

2).Design of program:-

We have created a main file with canvas package of racket. On running the file we get input window in which user can give input to the polygon by marking points on screen. Program takes points as input and opens the output window with tabs corresponding to triangulation, monotone partition, colouring and camera position. We have created objects corresponding to a polygon to which we call monotoning partitioning function. New monotone polygon is instantiated in separate monotone class inherited from polygon class. Monotone polygon is triangulated to give us triangle. Triangle is instance of node class inherited from monotone class.

3).Outline of algorithm:-

->We first break our polygon into monotone polygons. to do so we first identify the vertices that causes the non-monotonicity of the polygon and then we resolve them accordingly. We simply iterate through the vertices of polygon and break it into monotone polygons.

-> After partitioning our polygon Into monotone polygons we triangulate each polygon separately by chopping of triangles one by one and then recursively calling the function on remaining polygon .

->We then make a graph whose vertexes (of graph ,not polygon) are the triangles and two vertexes are joined by an edge(of graph ,not polygon) if they share a side in common

->We then run a depth-first-search on the graph and in this process we go to each vertex of the graph (that is each triangle) and colour the vertices by three different colours. This results in a three colouring of a graph whose vertices are the vertices of the polygon and whose edges are the edges of the the graph.

->After three colouring of the polygon we identify the colour that is used least no of times and identify the vertices of the polygon that were coloured by this colour ,placing cameras at these vertices provides the solution of our problem.

4).Limitations:-

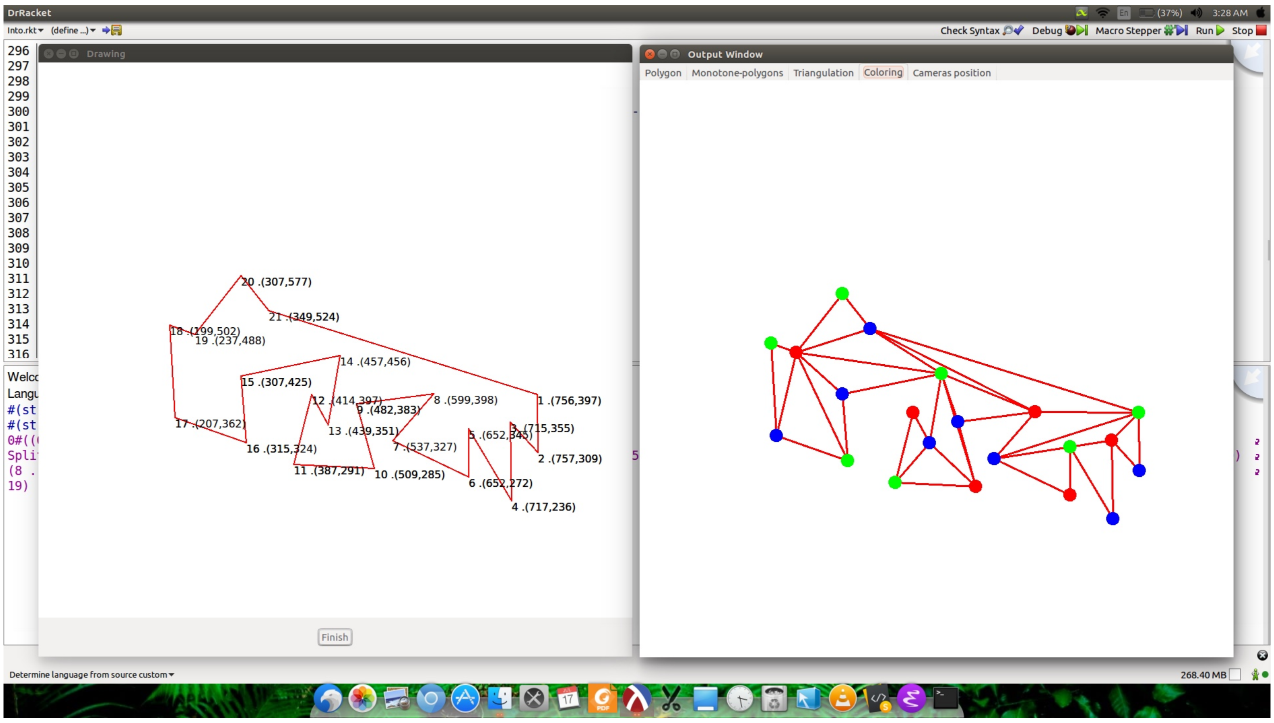
->The art gallery problem and all its standard variations are NP-hard thus our algorithm does not finds the minimum no.of cameras but it founds a number and places that is always sufficient and occasionally necessary

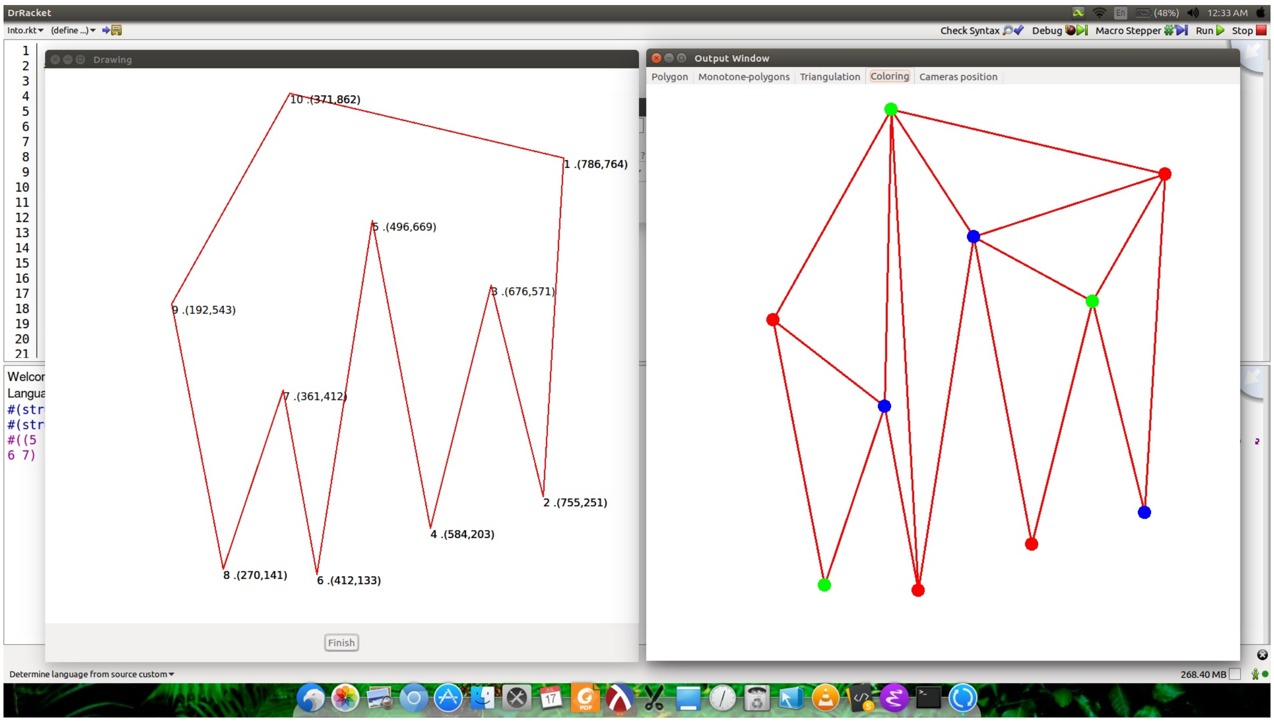
-> Program thus overestimates for some cases like star shaped polygon.

5)Interesting Features:-

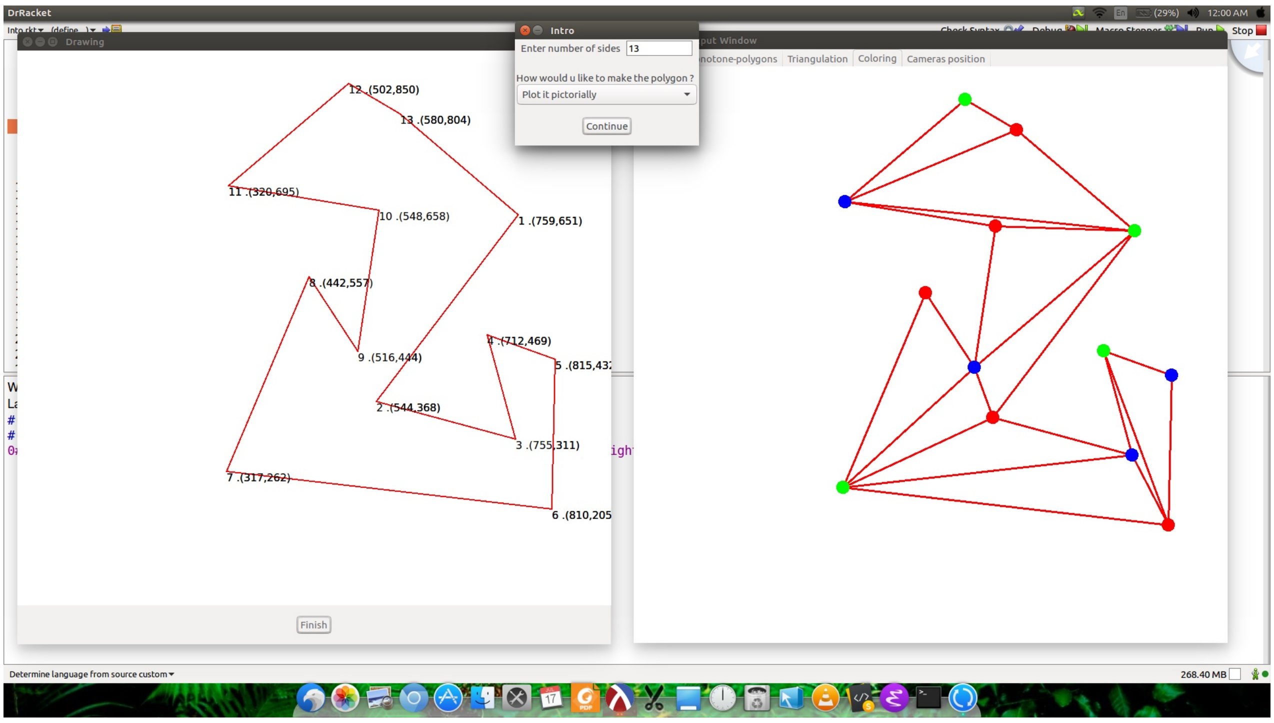
We have some higher order functions and higher order data structures in our program.
Some examples :
->gnodify is perhaps the most interesting function in program. It's input is a very nice data structure that is a vector of mpairs, where head of mpair is instance of node class and tail of mpair is a gvector of elements of node class.
->coloring is also a good algorithm in which we take node as defined above.
There was a very nice way to create gnode from vector of nodes. We memoized the vertices corresponding to the edges that were broken while creating monotone polygons. Then eventually it was very easy to create gnode (will explain better in demonstration).
There were other interesting points also for example when we needed to sort vector of vertices by y coordinates we consed the vertices with their y coordinates, and using the key features provided in vector sort it was a nice way to sort functions, also this consing was done for searching for vertices in same y coordinate.

6).Sample Inputs and outputs:-





```
#(str
#(str
#((5
6 7)
```



Enter number of sides 13

How would u like to make the polygon ? Plot it pictorially