**Project Subject**

For this project, we aim to place “communication objects” in a wide area: the field. These communicating objects are either servers or mobile clients (drones).

We consider that servers are linked by a wired network which connects each server to its neighbors. Two servers are neighbors if they reach a common border.

We have also drones which are linked to the closest server by a wireless network. They are mobile, autonomous, they detect close obstacles.

**Our Approach**

**Classes Used**

Field

FieldDraw (extends GlutWindow class)

Server

Drone

Triangle

Polygon

Vector2D

**Algorithms Used**

Simple triangulation

Graham’s scan

Delaunay Triangulation

Derive Voronoi Diagram from Delaunay Mesh

**Organization**

Kwabena BAMFO – assigned to drawing the field

Patrice GADEGBE – assigned to physics of drone movements and collision detection

Rabah GILES – assigned to dynamic distribution of drones in the field

Zihan SHEN – assigned to ensure clean merging of code on GitHub

**Our Solution**

**Server Part**

First, we compute the convex hull of a given set of severs using Graham’s scan algorithm.

**Image**

We then triangulate with this convex hull, add the interior points, and form a new triangulation with the interior points.

**Image**

After this normal triangulation, we apply the Delaunay triangulation algorithm on our list of triangles.

**Image**

Given our Delaunay triangles, we draw the Voronoi diagram using the algorithm proposed in Figure A.

**Image**

**Figure A**

A screenshot of text

Description automatically generated

**Drone Part**

**Challenges**

**Solutions**

**Existing Issues (Limits of our Solution)**

* Voronoi diagram is buggy for some combination of server locations, particularly because of minor issues with finding and adding the right corner points.
* Collision detection force looks smooth for some server areas and cumbersome for others.
* Distribution of drones in the field not optimized as per the project subject (that is, servers can only reach neighbours). Our implementation is close enough, but not ideal. We however have an ideal solution separate from the master branch, as demonstrated during our presentation.