



**EPS**

Escola Politècnica  
Superior

## Projecte/Treball Fi de Carrera

**Estudi:** Enginyeria Informàtica. Pla 1997

**Títol:** Algorismes per a la resolució de problemes de proximitat en terrenys

**Document:** Resum

**Alumne:** Eduard Oliver Juanola

**Director/Tutor:** Joan Antoni Sellarès Chiva  
**Departament:** Informàtica i Matemàtica Aplicada  
**Àrea:** LSI

**Convocatòria** (mes/any): Febrer - 2007

## Algorismes per a la resolució de problemes de proximitat en terrenys

El modelatge, visualització i anàlisi de terrenys (visibilitat, càlcul de camins mínims, visualització Diagrames de Voronoi...) és de gran importància en els Sistemes d'Informació Geogràfica (GIS). Actualment és de gran interès per a aquesta comunitat disposar de software que permeti analitzar terrenys. Aquest projecte final de carrera consisteix, a grans trets, en realitzar l'anàlisi, disseny i implementació d'un programari que sigui capaç de resoldre problemes de proximitat en terrenys, com poden ésser: determinació de camins òptims entre una o un conjunt de seus (que geomètricament es representen per punts, segments, polígons, poligonals, ...) i un o diversos punts del terreny, càlcul de la distància geodèsica entre una seu i un punt, visualització de Diagrames de Voronoi per a un conjunt de seus, .... Una part important per tal de poder obtenir tot això també ha d'ésser la de poder generar, visualitzar i modificar un model 3D d'un terreny a partir de dades introduïdes per l'usuari o obtingudes des d'un fitxer.

Per tal de poder construir l'aplicació desitjada ha calgut dissenyar una interfície gràfica d'usuari que permetés realitzar de forma interactiva la introducció, modificació i esborrat de les diferents seus (punts, segments, polígons, poligonals, ...) o restriccions del terreny, així com la seva visualització. El disseny de la interfície s'ha dut a terme tenint en compte una entrada de dades fàcil i intuïtiva, per tal que qualsevol tipus d'usuari pugui ésser capaç d'utilitzar-la. A partir d'aquesta interfície es podran dur a terme tots els objectius que ens havíem marcat en iniciar el projecte.

L'objectiu principal del projecte és desenvolupar una aplicació per a la **resolució de diversos problemes de proximitat en terrenys**. Per tal d'arribar a aquest objectiu final, ens caldrà bàsicament:

- Construir una interfície que ens permeti inserir i escollir seus (punts) sobre un terreny.
- Dissenyar i implementar diferents algorismes per tal de determinar camins òptims sobre terrenys.
- Calcular la distància geodèsica entre un punt del terreny i la seva seu més propera.
- Poder visualitzar tant en 2D com en 3D camins òptims sobre terrenys.
- Calcular i visualitzar Diagrames de Voronoi sobre terrenys.

Comentar també el fet que per al desenvolupament d'aquesta aplicació només s'ha usat software lliure (Llibreries **Qt**, **Open GL** i **CGAL**).

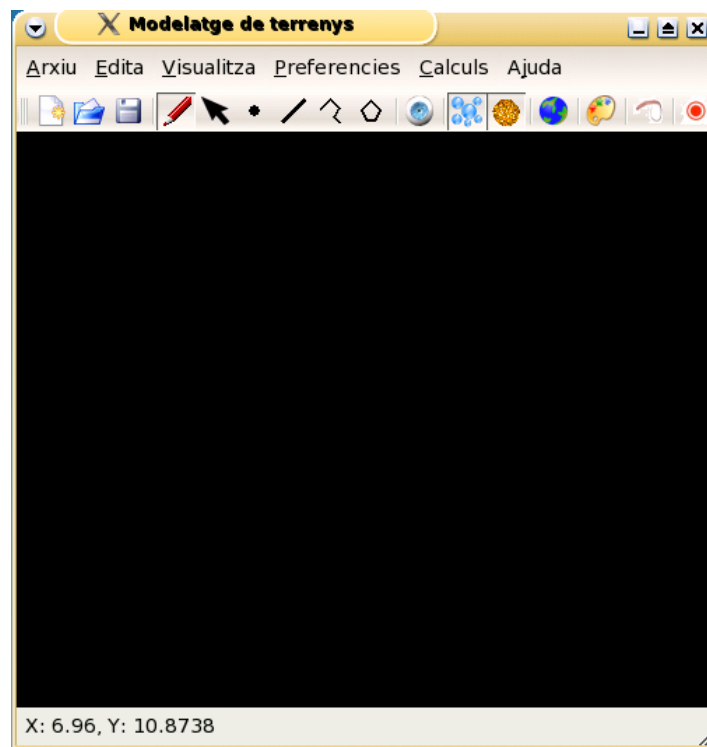
Per tal de poder complir amb el nostre objectiu principal, necessitarem disposar d'una interfície gràfica que ens permeti generar, visualitzar, editar i emmagatzemar terrenys. Per aconseguir això s'ha partit d'una interfície realitzada en un projecte anterior, la qual ha estat adaptada tot eliminant funcionalitats pròpies d'altres projectes, afegint-hi les noves funcionalitats del nostre projecte o bé canviant el comportament de funcionalitats ja existents, adequant-les a la nostra problemàtica.

Podem dividir bàsicament la interfície en dues parts:

→ **Editor 2D**

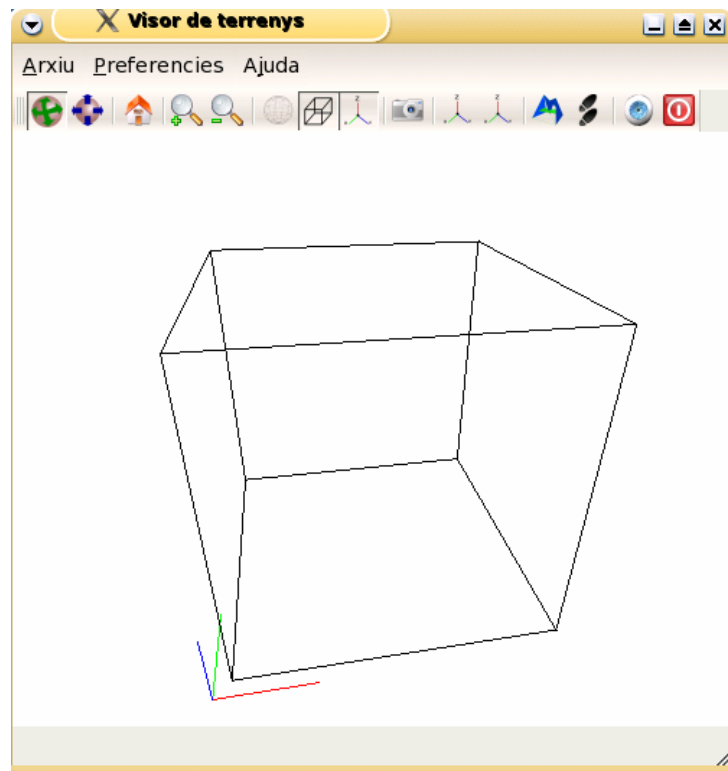
→ **Visor 3D**

*L'Editor 2D* serà qui ens permetrà realitzar totes aquelles operacions pròpies de l'edició de terrenys, càlcul de distàncies, càlcul de camins òptims, càlcul dels Diagrames de Voronoi, ... de forma interactiva. Ens mostrarà la triangulació del terreny a tractar i ens permetrà modificar-lo (inserir seus, eliminar-les, inserir punts de camí mínim, calcular el que hem comentat anteriorment, ...). Gairebé tots els canvis que hem introduït s'han realitzat en la finestra de *l'Editor 2D*, que és la finestra amb la qual més interactuarà l'usuari de l'aplicació i allà on es realitzen tots els procediments i totes les opcions principals del nostre projecte. La distribució dels nous controls afegits dins de la finestra de l'aplicació s'han dut a terme seguint una certa lògica de processos. S'ha tingut també en compte la comprensió del funcionament de cadascuna de les operacions perquè qualsevol tipus d'usuari sigui capaç d'utilitzar l'aplicació. En la *Figura 1* podem observar l'aparença de *l'Editor 2D*



**Figura 1** : Interfície de l'Editor 2D

D'altra banda, el *Visor 3D* serà qui ens permetrà representar el terreny triangulat en 2.5D amb el color i/o textures escollides. Del *Visor 3D* hem aprofitat bàsicament la carcassa i hi hem introduït els canvis pertinents per tal d'adequar-lo també a la nostra problemàtica. Aquesta finestra només serà de visualització i en tot moment reflectirà instantàniament tots els canvis que efectuem en l'*Editor 2D*. Així doncs, podem dir que a través del *Visor 3D* no podran realitzar-se modificacions sobre el terreny. Com que volem que tots els canvis que efectuem en l'*Editor 2D* es vegin reflectits també en el *Visor 3D*, requerirem que tots dos vegin el mateix objecte, cosa que ha estat possible gràcies a la utilització del patró **Singleton**. En la *Figura 2* podem observar l'aparença del *Visor 3D*.



**Figura 2** : Interfície del Visor 3D

En aquest projecte ens hem centrat bàsicament en resoldre de forma exacta dos problemes relacionats amb la proximitat en terrenys:

→ **Problema 1**: Trobar i reconstruir el **camí més curt** entre una seu i un punt qualsevol del terreny.

→ **Problema 2**: Trobar i visualitzar el **Diagrama de Voronoi** per a un conjunt de seus.

## Càlcul camí òptim entre seu i punt

Per a poder realitzar el càlcul del camí òptim entre una seu i un punt el primer que ens caldrà és disposar d'un terreny amb una seu inserida. Els camins mínims són visualitzats com a rajos que surten d'un vèrtex seu  $v_s$  en totes les direccions tangents de la superfície polièdrica  $P$  al vèrtex  $v_s$ . La funció de distància del camí més curt definit per un punt que sigui seu  $s$ , és una funció  $D_s$  tal que per a cada punt  $q$  de la superfície polièdrica  $P$ ,  $D_s(q)$  és la llargada del camí més curt de  $q$  a la seu  $s$ .

La idea bàsica, consisteix en agrupar els camins mínims d'una manera que puguin ésser parametritzats de forma automàtica. Amb aquest objectiu, els costats es parteixen en una sèrie d'interval·ls anomenats **Finestres**. Una altra estructura que haurem de emmagatzemar, són els **PseudoSources** o seus puntuals, que seran els que es formaran quan realitzem la propagació de finestres, i que sempre seran vèrtexs de la triangulació del terreny. Per tal d'emmagatzemar aquestes dues estructures de dades, s'ha creat una classe general abstracta anomenada **Event**, on hi haurà definits tots els mètodes utilitzats a les classes **Finestra** i **PseudoSource**.

Per a la reconstrucció del camí mínim, ens caldrà realitzar la inicialització i posterior propagació de les finestres al llarg del terreny, en ordre segons una cua de prioritat que haurem definit. També haurem de tenir en compte casos en què possibles finestres puguin intersecar o solapar-se. Un cop haguem propagat les *Finestres* i *PseudoSources* al llarg del terreny, utilitzant una tècnica de **Backtracing** podrem obtenir el camí més curt entre la seu i el punt.

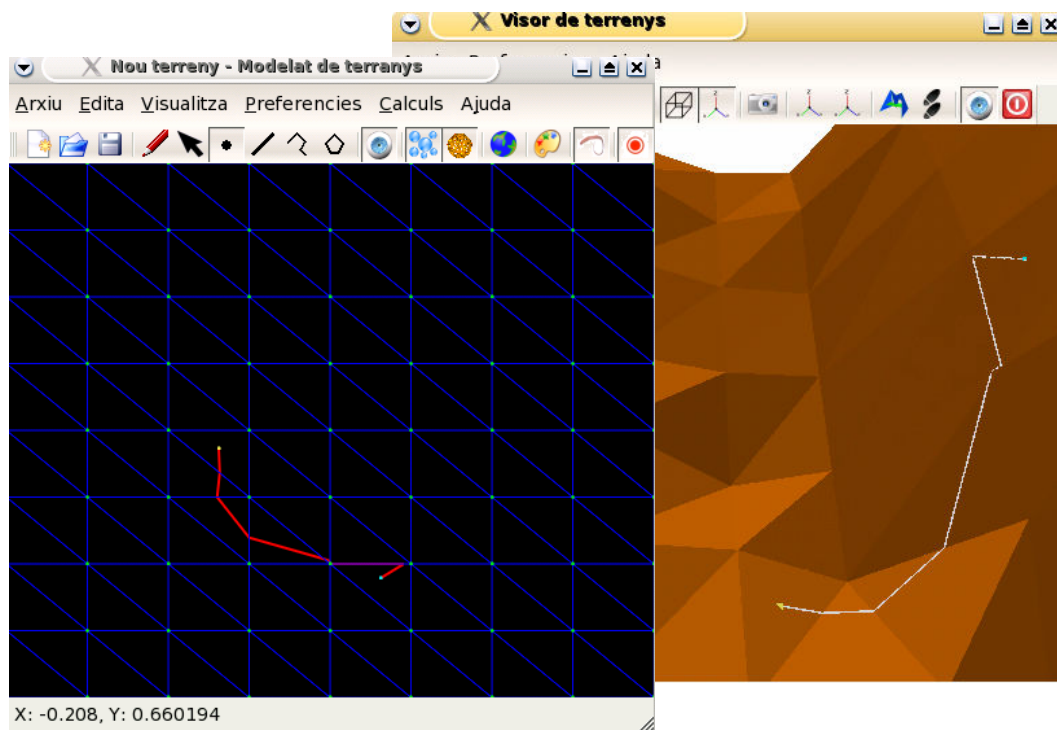


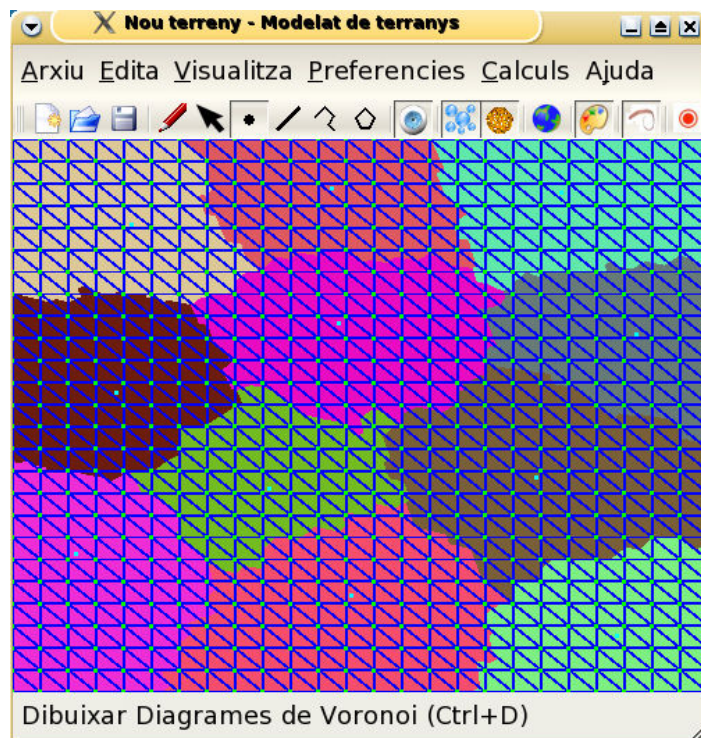
Figura 3 : Camí més curt entre seu i punt

## Diagrames de Voronoi

L'algorisme utilitzat anteriorment ens trobava la distància entre una seu i un punt qualsevol del terreny. Per tant, l'haurem de fer extensible al cas d'un terreny que disposi d'un conjunt de seus, fent que per a cada punt del terreny sapiguem la distància mínima a la seu més propera que tingui.

Això es durà a terme considerant totes les seus a la vegada, i ens caldrà conèixer de quina seu prové cada finestra o cada pseudosource nou que anem creant i propagant al llarg del terreny. Amb aquesta finalitat, es modificarà l'estructura de les **Finestres** i dels **PseudoSources**, afegint un atribut anomenat **index** que anirà de 1 fins a  $k$  (essent  $k$  el nombre de seus), que serà l'índex de la seu de la qual la finestra prové. D'aquesta manera, propaguem simultàniament el camp de distàncies definit per cada una de les diferents seus. Automàticament podrem obtenir una codificació de la funció de distàncies generalitzada que ens produeix una representació implícita del *Diagrama de Voronoi* del conjunt de seus.

Per tal de visualitzar una aproximació del *Diagrama de Voronoi*, calcularem una aproximació del camp de distàncies utilitzant només la distància als vèrtexs del terreny i hardware gràfic. Durant aquest procés, necessitarem guardar a cada vèrtex del terreny la distància a la seu més propera i l'índex d'aquesta seu. Tal i com es pot observar en la *Figura 4*, tots els punts del terreny que estan sota la superfície d'un mateix color tindran com a seu més propera la que està continguda també dins la superfície amb aquell color. Així doncs, donat un punt qualsevol del terreny sabrem dins quina regió del *Diagrama de Voronoi* el podem englobar i quina és la seu que té més propera.



**Figura 4** : Exemple de Diagrama de Voronoi per a un conjunt de seus