

Introducció

Avui en dia el modelatge de ciutats és un problema obert pel que no existeixen solucions estàndards ni de codi lliure. Aquest és un problema força important per a indústries com la del cinema, la realitat virtual o els videojocs. Dins d'aquest problema es poden presentar sub-problemes interessants, dels quals el modelatge de les xarxes de carrers pren un rol fonamental. La complexitat geomètrica que té una ciutat fa que el modelatge d'aquest tipus d'estructures sigui una tasca gens menyspreable.

Qualsevol persona, amb més o menys relació amb el món dels ordinadors, haurà jugat a algun joc d'aventures gràfiques on l'escenari escollit és una gran ciutat, o haurà vist alguna pel·lícula relativament actual on el protagonista viu la seva aventura envoltat de grans edificis. Cada vegada més es fan servir eines informàtiques per a crear aquests escenaris monumentals. Molts d'aquests escenaris han estat creats manualment, és a dir, muntant els edificis un per un fins aconseguir una ciutat.



El modelatge procedural pretén solucionar bona part d'aquest problema utilitzant les seves funcionalitats per a generar de forma sistemàtica la geometria necessària en cada cas. A grans trets, el procediment consisteix en deixar a la màquina la feina de crear la ciutat pròpiament dita i així l'usuari només s'ha de preocupar d'introduir els paràmetres necessaris per aconseguir el seu propòsit.

Existeixen eines de desenvolupament procedural, però el seu ús es troba restringit a unes quantes aplicacions que, generalment, no inclouen xarxes de carrers.

Objectius

L'objectiu d'aquest projecte és el desenvolupament d'una eina de generació de xarxes de carrers a partir d'exemples. L'eina ha de permetre generar una xarxa de carrers nova que sigui semblant a l'existent en un mapa vectorial donat.

A més, també es preten unir aquesta aplicació amb l'urbanEngine per tal de poder generar vistes en 3D sobre aquestes xarxes de carrers, a més d'ampliar les opcions de l'urbanEngine a l'hora de crear ciutats.

Programari Utilitzat

Per a la realització d'aquest projecte bàsicament s'han utilitzat tres eines diferents:

- El Houdini com a modelador 3D per tal de poder mostrar els carrers i les ciutats 3d que és genera l'urbanEngine.
- El Python com a llenguatge de programació base. Juntament amb ell s'han utilitzat les llibreries miniDom, per llegir i generar arxius en format OpenStreetMap, i les WxPython, per construir la interfície d'usuari.
- El format de mapes OpenStreetMap com a format d'entrada del exemples.

Modelatge de xarxes de carrers a partir d'exemples

El modelatge de xarxes de carrers a partir d'exemples consta de tres fases:

- **Entrada:** Exemples de fragments de ciutats (mapa OSM)
- **Processament:** L'usuari marca les pautes perquè l'aplicació generi una nova xarxa de carrers.
- **Sortida:** Nova xarxa de Carrers.

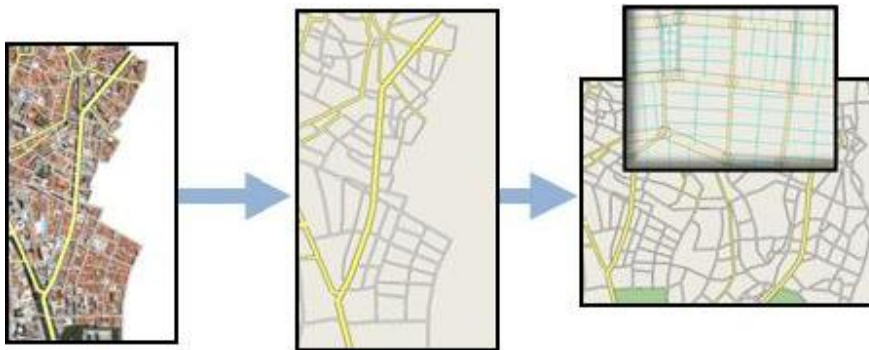


Diagrama on es mostra el procés que s'ageix l'aplicació.

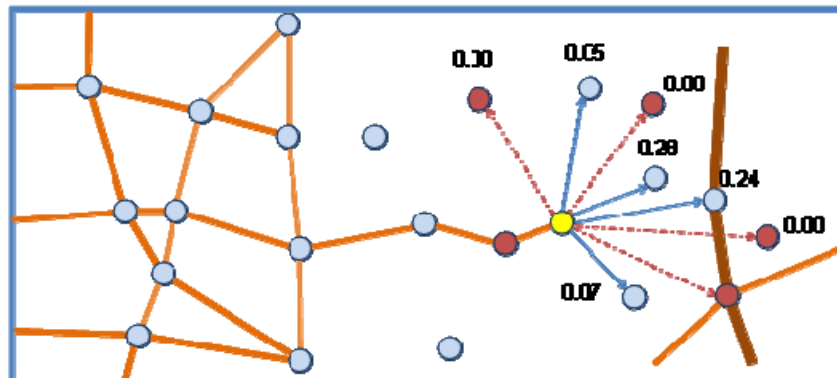
Un cop s'ha carregat el mapa OSM a l'aplicació aquesta calcula, per a cadascun dels carrers, les següents propietats:

- S'assigna un nivell a cada carrer depenent del tipus de via que es tracti.
- La mitjana de la distància d entre dues interseccions consecutives del mateix carrer.
- La mitjana de l'angle α entre dos segments (tant dels segments intersecció com dels que no) consecutius del mateix carrer.
- La varianza de la distància d entre els segments que passen per cadascuna de les interseccions respecte al promig de les mitjanes de la distància de tots els carrers que passen per ella.
- La varianza de l'angle α entre dos segments (tant dels segments intersecció com dels que no) consecutius del mateix carrer respecte a la mitjana de l'angle del carrer.
- També es calcula un valor al que hem anomenat "tortuositat", que representa la mitjana de la relació entre la distància acumulada entre els dos segments i la distància en línia recte entre els punts inicial d'un segment i final de l'altre.

Seguidament s'inicia la generació dels carrers, primer s'escull un punt dels punts dels carrers amb el nivell més alt i s'intenta connectar amb algun dels altres punts del mateix nivell, si un punt produeix algun d'aquests casos és descartat:

- El segment candidat creua amb un segment ja existent.
- La intersecció no admet més segments (màxim 4)
- La transició és excessivament estranya (angle o distància massa gran o massa petita)

- o Si un dels punts és la intersecció anterior



Exemple on es mostren una serie de punts invalids.

Un cop descartats aquests punts, es calcula la probabilitat dels que queden utilitzant la funció Gaussiana de grau 2 següent:

$$f(x, y) = e^{-\frac{1}{2} \left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2} \right)}$$

Finalment es tria el punt següent aplicant una funció de distribució acumulada (CDF) als punts dels que tenim la probabilitat.

Un carrer s'acaba quan totes les probabilitats de l'últim punt són 0 o s'han descartat tots els punts. Quan s'ha acabat un carrer s'inicia la creació d'un altre carrer del mateix nivell, això es repeteix fins que no es poden realitzar més connexions entre els punts d'aquell nivell. Un cop passa això es continua amb els punts d'un nivell més baix, fins que ja no queden més nivells.

Implementació

Durant l'implementació d'aquest projecte s'ha utilitzar les classes del diagrama següent:

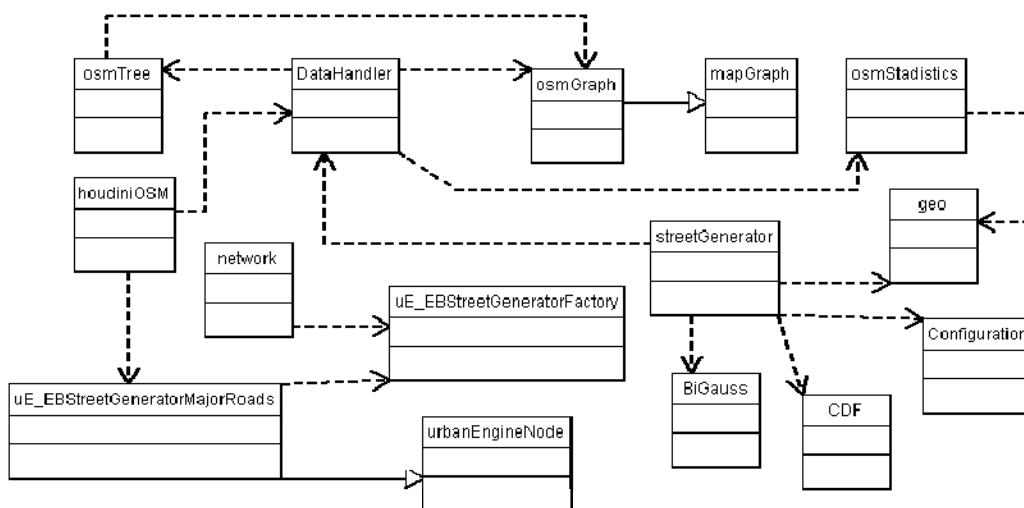


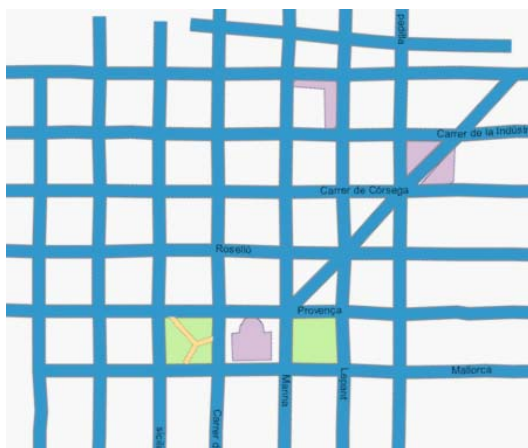
Diagrama d'activitat generat xarxa de carrers

A continuació s'exposaran les classes més importants i la funció principal que desenvolupen.

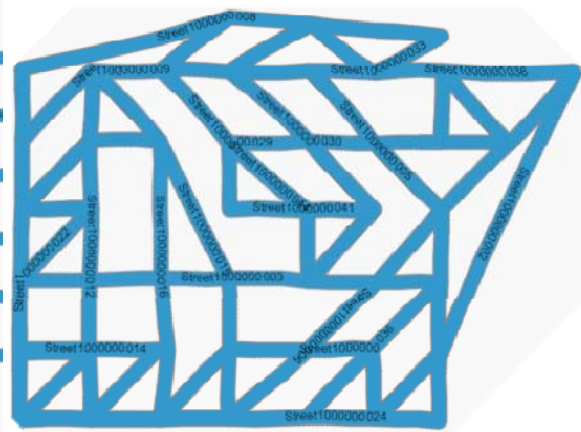
- osmTree
Aquesta classe s'utilitza per convertir arxius OSM en arbres miniDOM i viceversa.
- osmGraph
Aquesta classe s'utilitza per convertir arbres miniDOM en grafs i viceversa.
- osmStatistics
Aquesta classe s'utilitza per calcular una sèrie d'estadístiques dels carrers i les interseccions.
- streetGenerator
Aquesta classe s'utilitza per generar una nova xarxa de carrers a partir de les estadístiques calculades.
- geo
Aquesta classe agrupa tots els mètodes que treballen amb dades geomètriques i geogràfiques.
- uE_EBStreetGeneratorFactory
Aquesta classe s'utilitza per generar tots els nodes urbanEngine per a la construcció de la ciutat
- network
Aquesta classe marca la seqüència per a la creació de ciutats en 3D de l'urbanEngine.

Resultats

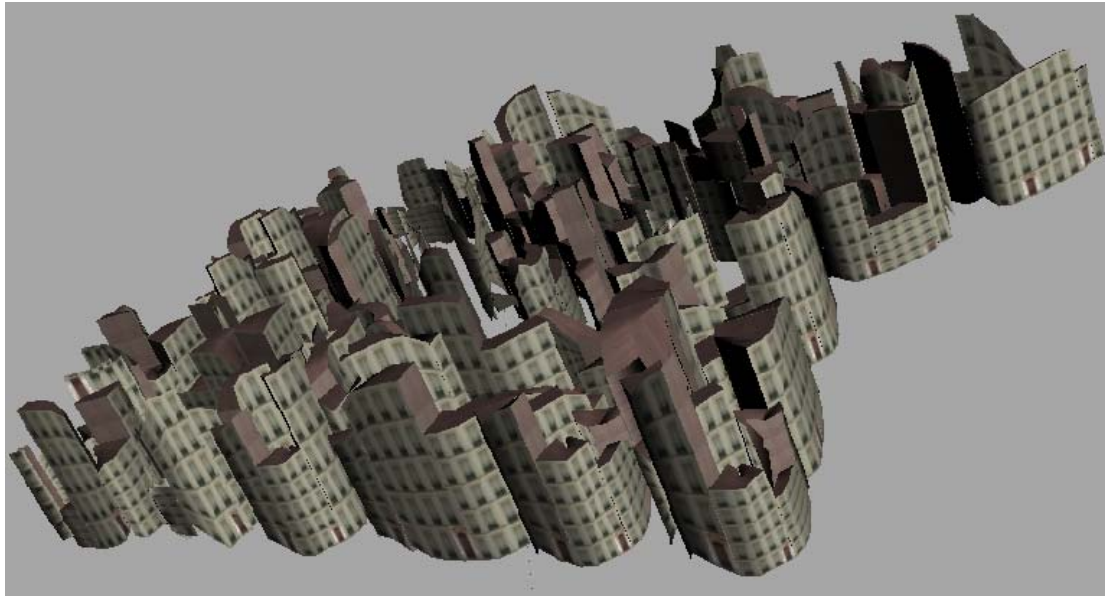
A continuació es mostrarà un exemple d'un dels resultats obtinguts.



Mapa OSM d'un districte de Barcelona



Mapa resultant d'utilitzar el districte de Barcelona anterior



Vista en 3D de la xarxa de carrers de la figura anterior

Conclusions

L'objectiu principal d'aquest projecte consistia en crear un programari que permetés enerar xarxes de carrers a partir d'exemples. Aquest objectiu ha estat complert de forma satisfactòria així com tot el conjunt de requeriments definits des d'un principi.

En un principi, el camí a seguir era el marcat per l'article "Interactive Example-Based Urban Layout Synthesis" (D. Aliaga, C. Vanegas i B. Benes, 2008). S'ha aconseguit un sistema per a crear xarxes de carrers basades en exemples bastant semblant al descrit per l'article, tot i que s'han hagut de solventar algunes problemàtiques no explicades completament a l'article.

A més també s'ha aconseguit crear un conjunt de classes genèriques per a treballar amb mapes OSM que poden ser utilitzades per a futures aplicacions que treballin amb aquest format.

El desenvolupament d'aquest projecte ha permès aconseguir dos tipus d'objectius. Per una banda els relacionats amb els requeriments de l'aplicació i per l'altre els referents a l'aprenentatge de temes que tenen a veure amb la geometria, l'estadística i dels diferents softwares utilitzats.

Pel que fa a l'aprenentatge s'ha aconseguit ampliar els coneixaments relacionats amb els conceptes següents:

- Algorismes de càlcul de relació entre vectors, segments i punts.
- Les funcions utilitzades en estadística: "Cumulative Distribution Function" i la funció Gaussiana.
- Eines de software com ara wxPython, miniDom, urbanEngine, Houdini...
- El llenguatge de programació Python