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 d'edificis i altres estructures arquitectòniques 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.

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 per 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 edificis.

Objectius

L'objectiu d'aquest projecte és el desenvolupament d'una eina pel modelatge procedural de ciutats i xarxes de carrers. El modelatge de carrers és, per si sol, un bon tema on aplicar-hi la programació procedural. Les ciutats solen comptar amb patrons que es van repetint al llarg del territori. El fet de "repetir" una tasca suggereix sempre l'aplicació d'algun tipus de procediment per tal de simplificar i reduir la feina de l'usuari a l'hora de desenvolupar aquesta tasca.

L'objectiu principal d'aquest projecte es centra en el desenvolupament d'una eina de fàcil ús que permeti a l'usuari obtenir de manera ràpida l'estructura bàsica d'una ciutat.

Eines

En la elaboració d'aquest projecte s'han utilitzat bàsicament tres eines diferents: Un modelador 3D, un llenguatge d'scripting i un llenguatge flexible per a definir els edificis.

Inicialment, es va proposar fer servir només el Houdini com a eina de treball per a modelar la geometria. Tot i això, aquest programa sol no donava la possibilitat de programar l'eina de manera sistemàtica, ja que està orientat a donar servei a través de la interfície gràfica d'usuari.

Per tal de poder implementar un mòdul recuperable i editable externament, es va optar per fer servir el Python que ja està incorporat dins el mateix Houdini. D'aquesta manera obtenim una perfecte connexió entre les dues eines. Primer es crea un script que construeixi la geometria, ja sigui directament amb la consola o usant un editor extern, i automàticament podem visualitzar el resultat dins de Houdini.

Finalment, s'ha optat per dotar al sistema d'un procediment de modelatge basat en el llenguatge XML. Aquesta tecnologia permet a l'usuari treballar amb una sintaxi coneguda i relativament senzilla.

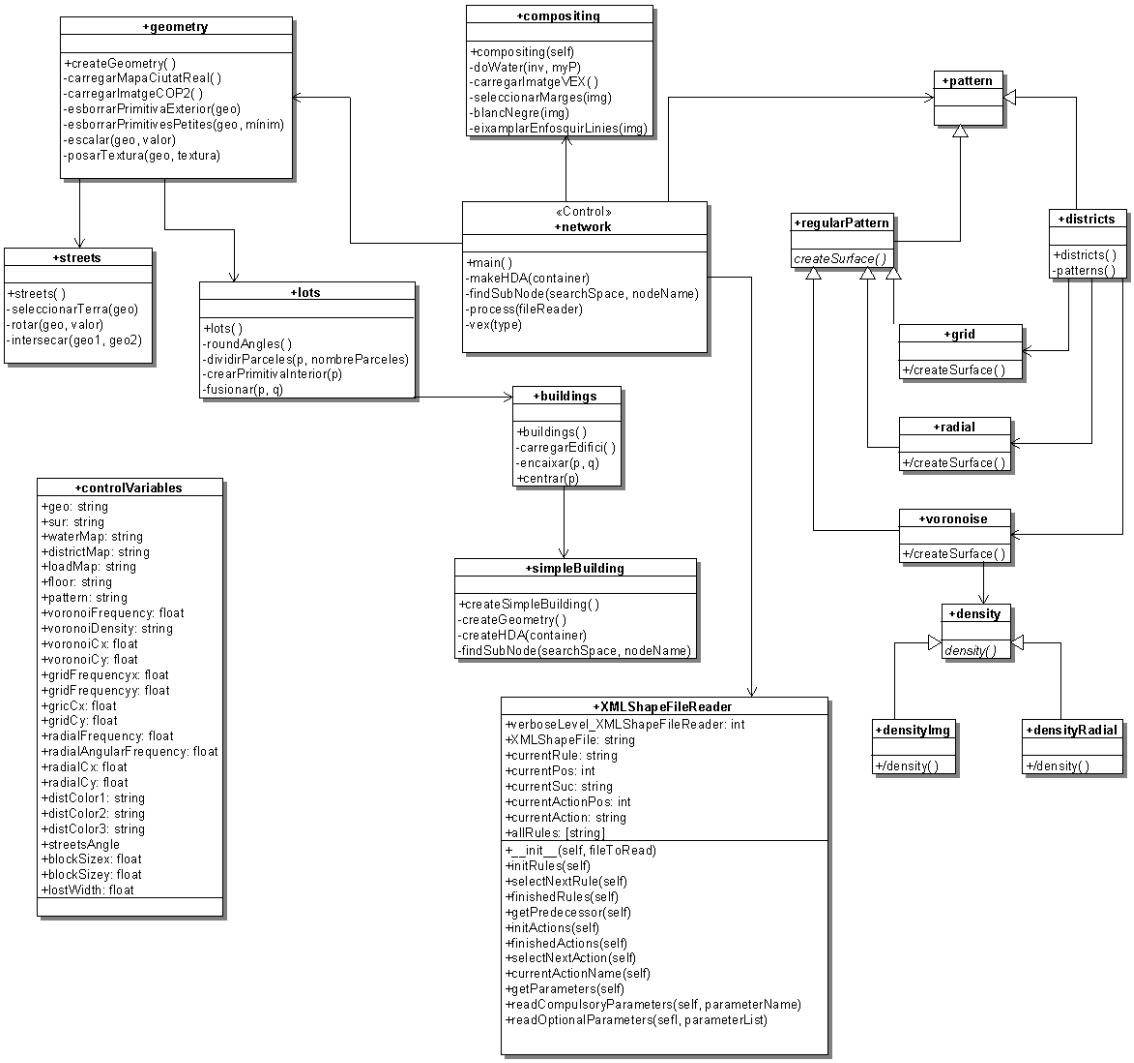
Treball previ

El projecte s'ha basat sobretot en l'article *Procedural Modeling of Cities* (Müller & Parish, 2001). L'objectiu de Müller i els seus companys és crear un sistema per a construir ciutats de manera automàtica i amb uns resultats realistes. Usant una sintaxi anomenada *CGA shape* i seguint les regles introduïdes amb aquesta sintaxi, s'aconsegueix crear la geometria d'una ciutat de manera ràpida i fàcil.

Les xarxes de carrers de les ciutats formen patrons de gran complexitat. Aquest patrons estan condicionats per motius geogràfics (rius, llacs, elevacions del terreny), històrics i culturals. Les carreteres de les zones més habitades també haurien de ser més denses per alleujar la càrrega del trànsit, i la gent de tot arreu hauria de tenir un accés fàcil a les carreteres.

Anàlisi

A continuació podem veure el diagrama de classes del sistema:



Disseny

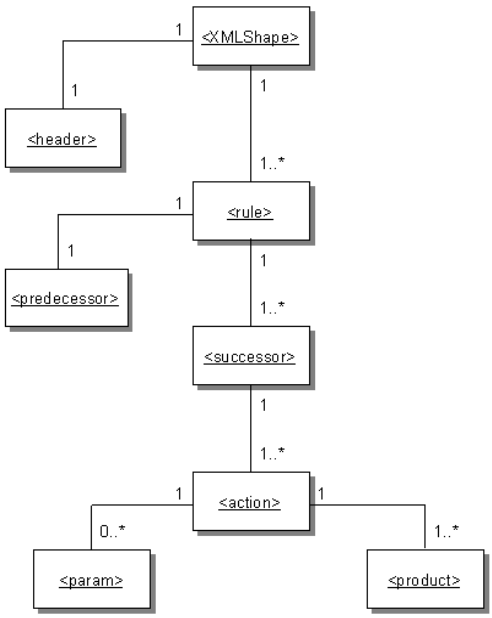
XMLShape funciona bàsicament igual que CGA Shape, exceptuant algunes funcionalitats que no ha calgut dissenyar ja que el mateix Houdini les proporcionava.

Conceptualment, la sintaxi d'ambdós sistemes és similar:

Id: predecessor → *successor*

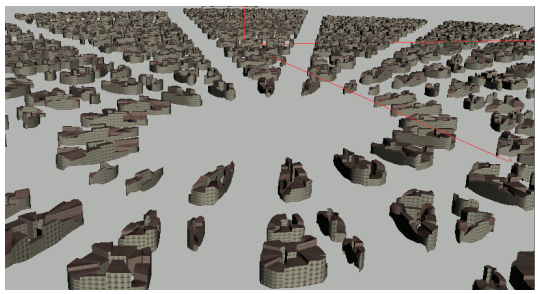
D'aquesta manera s'obtenen el conjunt de regles necessàries per a dissenyar les xarxes de carrers.

A continuació podem veure una representació dels tags XMLShape i quines relacions tenen entre ells.

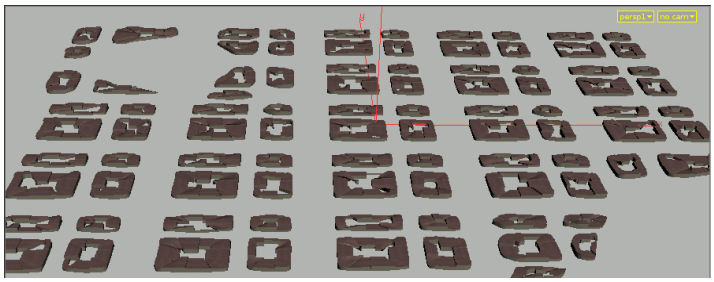


Resultats

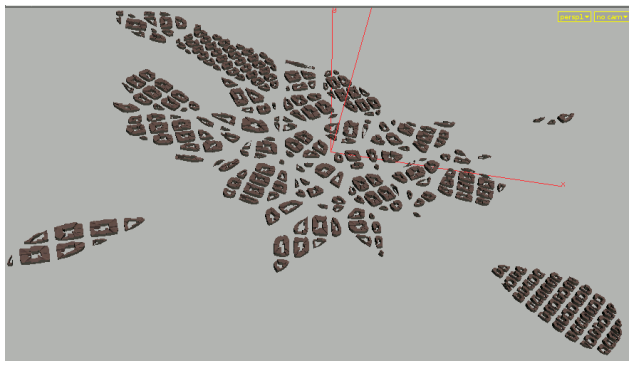
- Ciutat amb patró radial.



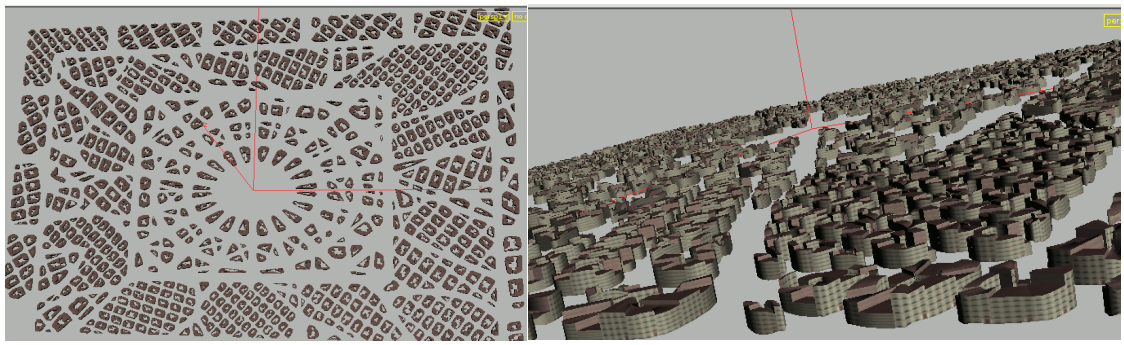
- Ciutat amb patró grid amb mapa d'ús de la terra:



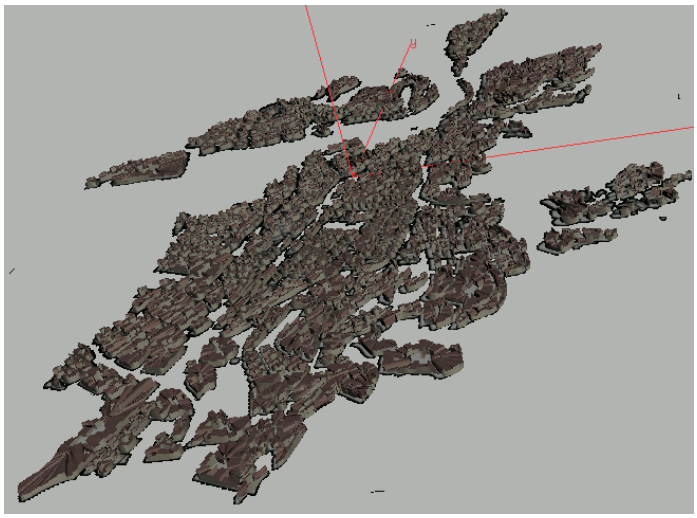
- Ciutat amb patró voronoise amb densitat centrada:



- Ciutat amb mapa de districtes:



- Ciutat a partir del mapa de Girona:



Conclusions

Els objectius i abast d'aquest projecte s'han pogut assolir satisfactòriament. Aquests objectius eren els següents:

- Aplicació d'un conjunt de regles que generi iterativament la xarxa de carrers.

  S'ha definit un sistema de regles (XMLShape) per a generar xarxes de carrers que, a més, divideixen les illes resultants en parcel·les i permeten aixecar-hi edificis.

Aquest sistema ens permet definir, a més de triar quin tipus de ciutat volem generar, tota una sèrie de paràmetres per a totes les fases de la generació. Així, podem triar paràmetres generals de cada patró, com ara on estarà ubicat el centre de la ciutat i la freqüència de les illes, i paràmetres més específics de cada geometria, com ara l'amplada dels carrerons o la mida dels patis interiors dels edificis.

- Aplicacions d'aquest sistema per generar ciutats a partir del mapa d'una ciutat real.

Podem generar geometria a partir d'un mapa d'una ciutat real, a més de poder-hi posar una textura al terra. En fer servir una imatge que ens dóna l'usuari, el sistema resulta ser molt genèric i configurable, ja que qualsevol imatge amb dos colors és vàlida per a ser usada per aquest propòsit.

- Aplicacions per generar ciutats sintètiques amb mapes de districtes, ús de la terra i densitat de població com a entrades.

  Podem generar ciutats sintètiques a través d'aquestes entrades. Amb un mapa de districtes podem definir fronteres a la ciutat i fer que aquesta consti de diferents patrons en cada part. Amb un mapa d'ús de la terra podem definir en quines parts de la ciutat no s'hi pot edificar a causa de rius, llacs, muntanyes, etc. Amb un mapa de densitat podem definir les diferents densitats de població a través de la ciutat.

  A més, hem implementat tres patrons diferents per a les ciutats sintètiques: voronoise, grid i radial.

  Apart de tot això, també s'ha desenvolupat un HDA (Houdini Digital Asset), que actua com a interfície d'usuari. Aquest, un cop generada la ciutat, ens permet modificar-ne alguns paràmetres que canviaran immediatament la geometria. Aquesta eina permet que alguns dels paràmetres entrats per l'usuari en el fitxer XML no hagin de ser definitius i que aquest pugui fer canvis sense haver d'escriure un altre fitxer XML i tornar a executar el programa.