

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 subproblemes 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 PFC és el desenvolupament d'una eina pel modelatge procedural d'edificis i altres estructures arquitectòniques. El modelatge d'edificis és, per si sol, un bon tema on aplicar-hi la programació procedural. Un edifici normal compta sempre amb elements que es repeteixen en altura i amplada. 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 feina.

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 el treball realitzat per Pascal Müller et Al. Concretament a l'article *Procedural Modeling of Buildings* [Müller '06]. 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 *CGAshape* i seguint les regles introduïdes amb aquesta sintaxi, s'aconsegueix crear la geometria de qualsevol edifici d'una manera ràpida i fàcil.

Disseny

Aquesta gramàtica resultant, que a partir d'ara anomenarem XML shape, funciona bàsicament igual que la seva antecessora, 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 els edificis:

1: \emptyset → *createBase*(type, x, y, z [, dx, dy, dz]){geo}

1: *pred* → *translate*(tx, ty, tz){geo}

1: *pred* → *comp*() {f1, f2, f3, f4, f5, f6}

1: *pred* → *subdiv*(div1...divN){p1...pN}

1: *pred* → *import*(type [, params]){geo}

Dins el fitxer XML la sintaxi segueix el següent patró:

```
<XMLshape>
  <rule id="1">
    <predecessor name="pred"></predecessor>
    <successor>
      <action name="act">
        <param name="p1" value="x"/>
        <param name="p2" value="y"/>
        ...
        <product name="prod"/>
        ...
      </action>
    </successor>
  </rule>
  ...
</XMLshape>
```

Un exemple per a identificar els elements:

1: *pred* → *act(p1, p2){prod}*

Implementació

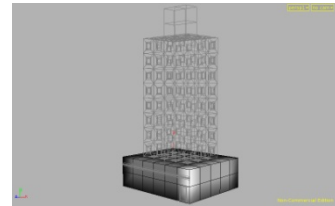
Un dels primers problemes que van aparèixer va ser l'emmagatzematge de les dades que s'anaven generant. Per solucionar aquest problema, s'ha optat per emmagatzemar els nodes utilitzant una llista de tuples anomenada *objects*. Cada tupla conté el nom de l'objecte com *string*, que és el que es fa servir al fitxer XML, una referència a l'objecte físic i una tupla anomenada *scope* que conté les mides de l'objecte.

Nom	Referència	Scope
Base	<i>&base</i>	[x,y,z]
Window	<i>&window1</i>	[x,y,z]
Facade	<i>&facade</i>	[x,y,z]
Window	<i>&window2</i>	[x,y,z]

Per a executar les regles es fan servir les següents accions:

Acció createBase

Aquesta acció crea la base dels edificis. A partir d'un cub de mida 1 unitat, s'obté un poliedre amb les mides definides per l'usuari.



Acció translate

Aquesta acció serveix per desplaçar un objecte una distància determinada des del seu punt actual. Les unitats que es fan servir per mesurar les distàncies es basen en la mida de les divisions dels objectes. Per exemple, es pot desplaçar una figura N divisions seguint l'eix de les X.

Acció comp

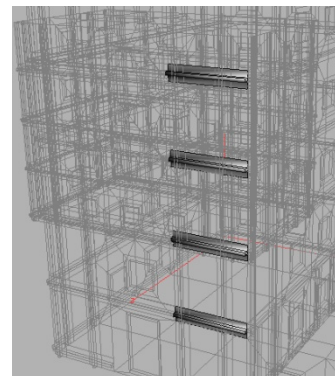
Houdini enumera les divisions d'un objecte seguint sempre un mateix patró. D'aquesta manera es pot aconseguir una fórmula per a determinar quines divisions formen part de cada cara. Aquesta acció aprofita aquest fet per a separar totes les cares d'un poliedre.

Acció subdiv

Seguint el mateix sistema basat en la numeració de les divisions que s'ha vist a l'acció *comp*, amb l'acció *subdiv*, es separen files o columnes d'una mateixa cara.

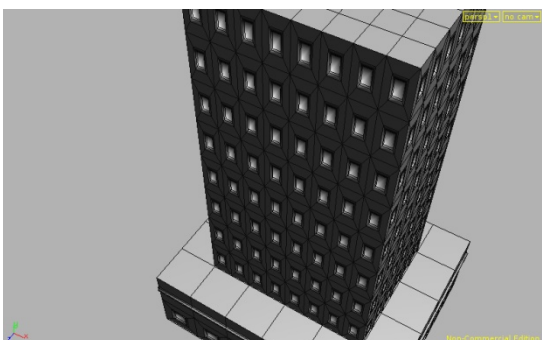
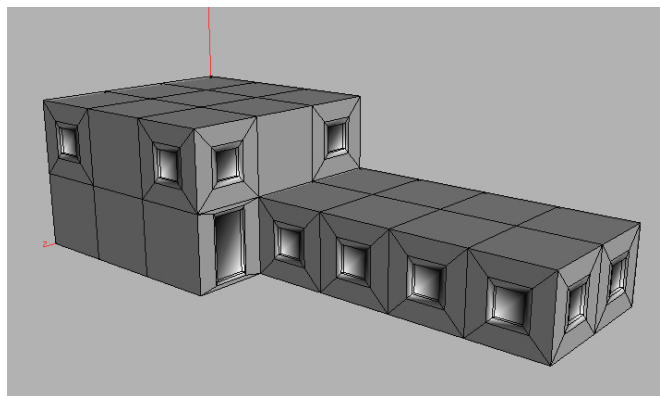
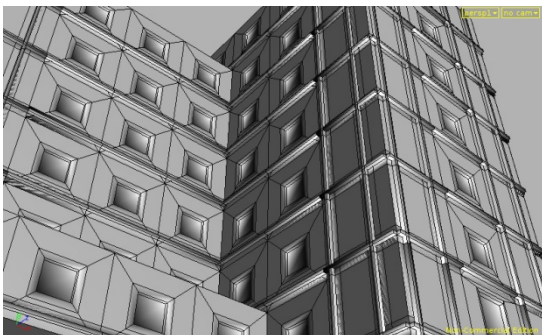
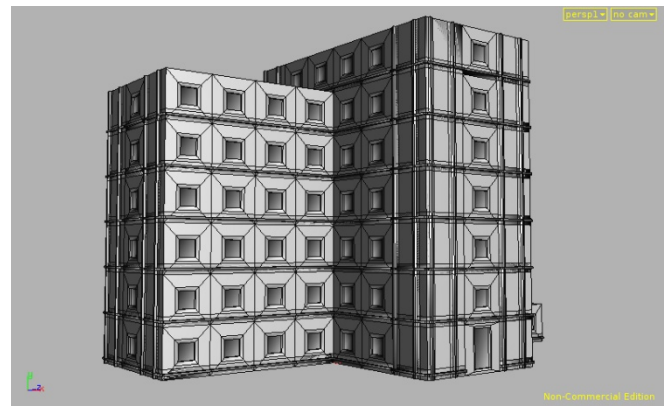
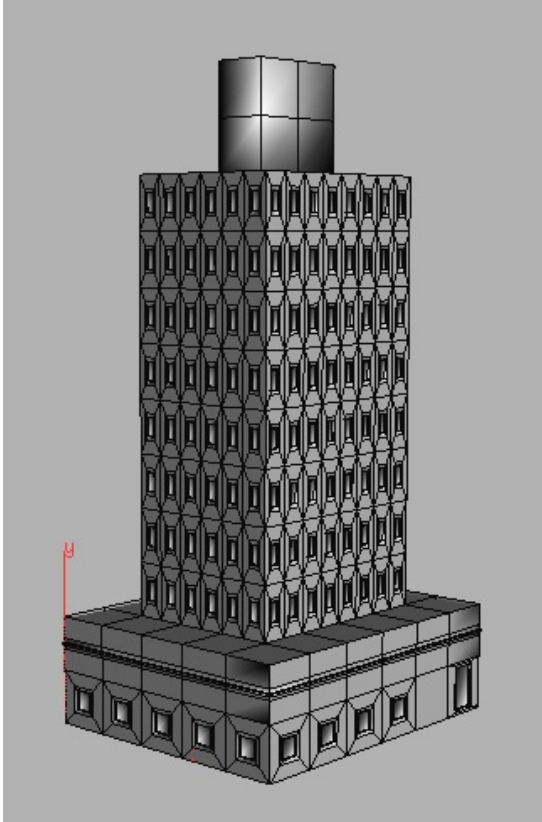
Acció import

Aquest acció serveix per a generar la geometria dels detalls de l'edifici com portes, finestres i altres elements arquitectònics.



Resultats

A continuació es mostren els resultats obtinguts. S'han creat diferents tipus d'edificis. Una casa, un edifici de negocis o un edifici compost per varis blocs intercanviables són un bon exemple:



Conclusions

Un cop valorats els resultats obtinguts cal estudiar si s'ha aconseguit allò que es pretenia a l'inici. L'objectiu bàsic d'aquest projecte era el desenvolupament d'una eina de fàcil ús que permetés a l'usuari obtenir de manera ràpida l'estructura bàsica d'un edifici. A priori l'objectiu s'ha complert, tot i que encara queda molt per a afegir i retocar, com a tots els projectes. En un principi, el camí a seguir era el marcat per Peter Wonka i Pascal Müller als seus articles. S'ha aconseguit un sistema per a crear edificis bastant semblant al de Müller i Wonka. La generació de la geometria bàsica ha quedat bastant completa i el fet que, un cop executat l'algorisme, es pugui seguir manipulant l'objecte amb el Houdini fa que sigui una bona eina a l'hora de crear estructures arquitectòniques de manera ràpida i senzilla. S'ha aconseguit crear un conjunt de regles de fàcil comprensió, però a la vegada molt potents a l'hora de crear geometria. El sistema tan és vàlid per un edifici senzill, d'un sol bloc, sense molts detalls, com per edificis de gran complexitat com poden ser, blocs d'oficines, amb moltes finestres i complements. També s'ha aconseguit crear un model de casa unifamiliar fent els blocs més baixos i senzills. En definitiva, s'ha creat una eina polivalent i útil que avarca multitud de solucions geomètriques.

Treball futur

Una dels temes que queda per tractar és la implementació de figures arrodonides. Seria bo poder afegir cilindres i esferes a l'eina per tal de poder construir torres, columnes, cúpules, etc. L'aplicació de textures als edificis també és una opció que representaria, segurament, el tema principal d'un altre projecte de final de carrera. Un altre front per desenvolupar seria la generació de teulades per a complementar els edificis.