

# Capítol 12

## Annexes

### 12.1 VR Juggler

VRJuggler és un conjunt de llibreries que permeten escriure aplicacions de realitat virtual d'una forma senzilla, sense tenir en compte el hardware d'on s'executarà el software, ni tampoc cal conèixer els perifèrics que es poden utilitzar des d'un punt de vista físic, ja que VRJuggler s'encarrega de tota la gestió. En altres paraules, VRJuggler no és més que un embolcall del hardware de la màquina vist des del punt de vista del software a desenvolupar.

VRJuggler té la peculiaritat de que cada aplicació és considerada com si fos un objecte i reben nom d'*application objects* (*aplicació-objecte*). VRJuggler utilitza l'aplicació-objecte per crear un ambient de realitat virtual que interactuarà amb l'usuari. A més a més, l'aplicació a desenvolupar, caldrà que implementi funcions predefinides per VRJuggler que seran necessàries per a que l'execució es desenvolupi correctament.

#### Avantatges de VRJuggler

VRJuggler permet canvis dinàmics en temps d'execució, és a dir, vrjuggler, permet l'intercanvi d'aplicacions, configurar dispositius en temps real, reconfigurar dispositius que s'estan utilitzant i enviar la reconfiguració a l'aplicació per tal de mantindre la coordinació entre aplicació i dispositiu.

Una aplicació-objecte pot estar desenvolupada en diferents llenguatges de programació, com Python, C#, o VB.NET encara que el nucli de vrjuggler estigui escrit en C++.

El fet de que existeixin el concepte d'aplicació-objecte, permet escriure extensions a vrjuggler sense que aquest pateixi cap canvi.

### 12.1.1 Estructura Bàsica

En aquest apartat es presenten les diferents funcions bàsiques per a que vrjuggler pugui fer funcionar l'aplicació que es vol desenvolupar. El nucli de vrjuggler és qui diu en tot moment què ha de fer l'aplicació-objecte. Constantment, el nucli es troba en un bucle infinit fent crides a les funcions predeterminades. Aquest bucle, no és del tot infinit, donat que quan l'aplicació de l'usuari acaba, aquest bucle, també acaba.

VRJuggler presenta com a primera interfície *vrj::App*, que és pot dir que és la interfície base. Per tant, és per aquí on es comença la programació d'una aplicació de realitat virtual amb aquesta llibreria.

Les funcions que es necessiten implementar són:

- `init()`
- `apiinit()`

#### *init()*

Qualsevol tipus de dada de l'aplicació s'ha d'inicialitzar en aquest punt. Apart d'aquesta inicialització, també ha de ser on l'aplicació fagi aquelles accions que només farà un cop durant la seva execució, com carregar dades d'arxius o fer *back-up* d'algún fitxer, entre d'altres.

#### *apiinit()*

Aquesta funció és cridada per a una inicialització concreta de la API gràfica. En el cas d'OpenGL és on es faria un *glClearColor*, per exemple, encara que no és necessari realitzar la crida en aquesta funció. A part, aquí és on també s'inicialitzen membres de l'aplicació, però, a diferència de l'anterior funció, és en aquest punt on s'inicialitzen les variables que no poden prendre valor fins que l'API gràfica, no s'hagi inicialitzat. A més, el nucli de vrjuggler la crida, després de la inicialització gràfica, però abans, de que es comencin a crear els primers *frames* de l'aplicació.

Aquestes dues funcions anteriors, són les més bàsiques amb les que vrjuggler treballa. Ara bé, vrjuggler per tal de crear fotogrames, internament crea *frames*, és a dir, utilitza unes funcions específiques que es criden en la creació, execució i postexecució de cada *frame*. Com a interfície base, *vrj::App* també té les següents funcions:

- `preFrame()`
- `intraFrame()`
- `postFrame()`

### ***preFrame()***

Aquesta funció és cridada pel nucli just abans de pintar l'escena. És aquí on s'han de fer les últimes actualitzacions de l'aplicació, com pot ser actualitzar les posicions dels diferents dispositius, entre d'altres.

### ***intraFrame()***

Aquest mètode es crida paral·lelament a la renderització de l'escena, és a dir, s'executa mentre un frame és generat. En aquest punt, és on es poden avançar càlculs de cara al següent frame que s'ha de generar. S'ha de tenir en compte, que les accions que es volen generar dintre d'aquesta funció no tinguin res a veure amb el renderitzat, donat que pot donar lloc a resultats no desitjables.

### ***postFrame()***

Finalment aquesta funció està disponible al final del processament del frame, és a dir, quan el bucle del nucli arriba al final de la volta. Aquest és un bon punt per actualitzar aquella informació que no depèn necessàriament dels dispositius ni de informació que canvia en el processat. Com funció específica, es pot utilitzar per a netejar informació no necessària després del frame.

## 12.1.2 Aplicacions OpenGL

Un cop vista la interfície base, fa falta veure com s'organitza `vrjuggler` per tal de gestionar els gràfics. Per tal de fer això, hi ha la interfície `vrj::GLApp` que gestiona les crides de OpenGL. Aquesta interfície hereta de `vrj::App` i té els següents mètodes necessaris d'implementació:

- `draw()`

## Sistema d'inspecció interactiva del procés de construcció d'una catedral

- `getDrawScaleFactor()`
- `contextInit()`
- `bufferPreDraw()`

La més important de les funcions és *draw()* ja que qualsevol cosa que es vulgui pintar en OpenGL s'haurà d'indicar en aquesta funció.

També s'ha de tenir en compte la funció *getDrawScaleFactor()* perquè s'ha d'estipular quina és la unitat de mesura a l'hora de renderitzar. Per defecte, la unitat de mesura és el metre i si no és perquè es desitja treballar amb altres unitats, no cal reimplementar la funció.

### ***Utilització de dispositius***

Un cop que s'ha vist el funcionament de vrjuggler, parlant sempre des de la vessant de mínims que cal seguir per a fer una aplicació que funcioni, cal veure com es poden utilitzar dispositius.

A l'hora d'escriure aplicacions en vrjuggler s'ha de tenir en compte que no es pot accedir directament al hardware dels dispositius. Aquest és el mecanisme pel qual vrjuggler permet desenvolupar aplicacions que són portables cap a qualsevol sistema. La manera de poder utilitzar, doncs, un dispositiu és connectar-se a ell via *proxy*. Un *proxy* no és més que un intermediari entre dos parts. En aquest cas, una de les parts és el dispositiu i l'aplicació vrjuggler. Per tant, l'aplicació fa peticions al dispositiu mitjançant el *proxy*.

Qui gestiona la part de dispositius dintre de vrjuggler és el mòdul Gadgeteer Input Manager per a informació d'entrada del dispositiu cap a l'aplicació. Per assistir a l'ús de *proxies* per dispositius, Gadgeteer proveeix del que es coneix com interfícies de dispositius.

Per a cada tipus de dispositiu, ja sigui digital, analògic, teclat o ratolí existeix una classe de *proxy*. Conèixer una classe *proxy* per tal de gestionar un dispositiu no requereix gran esforç. Només cal conèixer la interfície per al tipus de dispositiu que es vulgui gestionar. Normalment, hi ha prou en conèixer el mètode *getData()* per tal d'obtenir la informació del dispositiu.

Un cop es determina el dispositiu que s'utilitzarà en la interacció amb el món virtual, cal especificar amb quin tipus de dispositiu es treballarà. Per tal de determinar el dispositiu que es farà i *vrjuggler* pugui treballar amb comoditat, cal utilitzar una de les interfícies destinades per a tal finalitat. En concret, cal inicialitzar els dispositius en la funció *init()* de la interfície de l'aplicació, és a dir, *vrj::GLApp* en el cas de que s'utilitzi OpenGL, com en aquest projecte.

Dins d'aquesta funció cal declarar la classe del tipus de dispositiu i inicialitzar el dispositiu amb un text descriptiu, com més descriptiu sigui el text més senzilla és la programació, donat que serà fàcil identificar el text amb el dispositiu que s'està treballant.

Per exemple, es pot suposar el cas d'utilitzar un dispositiu per a realitzar *head-tracking*, un dispositiu que es pot utilitzar perfectament en aquest projecte. Caldria, doncs, inicialitzar la interfície *gadget::PositionInterface* amb les següents sentències:

```
gadget::PositionInterface    head;
head.init( "VJHead" );
```

Cal fer èmfasi en el fet de que la variable *head* ha de ser una variable membre de l'aplicació-objecte. Un cop el dispositiu està inicialitzat *vrjuggler* i el seu mòdul *gadgeteer* estan preparats per treballar en la rereguarda tot actualitzant els *proxies*, i donant lliure accés a la aplicació cap a ells.

A l'hora d'actualitzar informació des de dispositius és millor fer aquesta tasca en la funció *preFrame()*, encara que no és del tot necessari, donat que aquesta funció s'executa abans de començar a realitzar el *frame*. Continuant amb l'exemple anterior del *head-tracking*, per tal de llegir la informació del *proxy* corresponent a la matriu de transformació del cap de l'usuari, s'hauria de fer quelcom com:

```
gtml::Matrix44f head_mat = head->getData();
```

*gtml::Matrix44f* és una matriu de 4 files per 4 columnes del tipus float que el mateix *vrjuggler* proporciona i se sol utilitzar per a calcular transformacions geomètriques donat que aquesta classe, proporciona tot un seguit de funcions que fan que aquesta matriu sigui molt operable.

### *Configuració de VRJuggler*

Per tal d'entendre més i millor com funciona vrjuggler cal veure què són i com es generen els arxius de configuració.

VRJuggler està dissenyat per a ser altament configurable per tal de poder mantenir l'alt grau de portabilitat amb el qual ha estat dissenyat. La unitat més bàsica de configuració per a vrjuggler és el que s'anomena *config-element* (*element de configuració*).

Cada element de configuració està compost per una o més propietats que s'identifiquen amb un nom donat per l'usuari. Cada propietat és d'un tipus com pot ser, string, integer, pointer, entre d'altres, i poden tenir diferents valors (zero o més).

Element:	
display_window	
<b>Origin:</b> <i>int</i>	X Coordinate
	Y Coordinate
<b>Size:</b> <i>int</i>	Width
	Height
<b>Stereo:</b> <i>bool</i>	In stereo?

Figura X Esquema d'un element de configuració

En la figura es pot apreciar com la propietat Size té dos variables: alt i ample. El tipus vàlid per als dos valors és un enter. En general, tots els valors de les variables d'una propietat tenen el mateix tipus, encara que com tot hi ha excepcions. El codi que manega l'element de configuració pot interpretar de diferents maneres el valor de la variable d'una propietat. Per exemple, si una propietat té una variable el valor de la qual és 0, aquest es pot interpretar com a real, enter o booleà.

Un arxiu de configuració és aquell arxiu que té la extensió *.jconf*. Cada arxiu de configuració conté zero o més elements i alhora pot ser que facin referència a altres arxius de configuració. A més, un arxiu d'aquestes característiques està en el format *XML*, un format que permet i molt la portabilitat.

Donat que els arxius de configuració estan elaborats seguint les pautes del format *XML*, vol dir que un arxiu de configuració es pot editar amb qualsevol editor de textos que hom pot tenir en la màquina en la que treballa. VRJuggler a part de proporcionar tots els arxius necessaris per al desenvolupament i execució d'aplicacions, també proporciona una eina, VRConfig, basada en *Java* i amb interfície gràfica, la qual edita automàticament l'arxiu tot indicant els elements que es volen configurar.

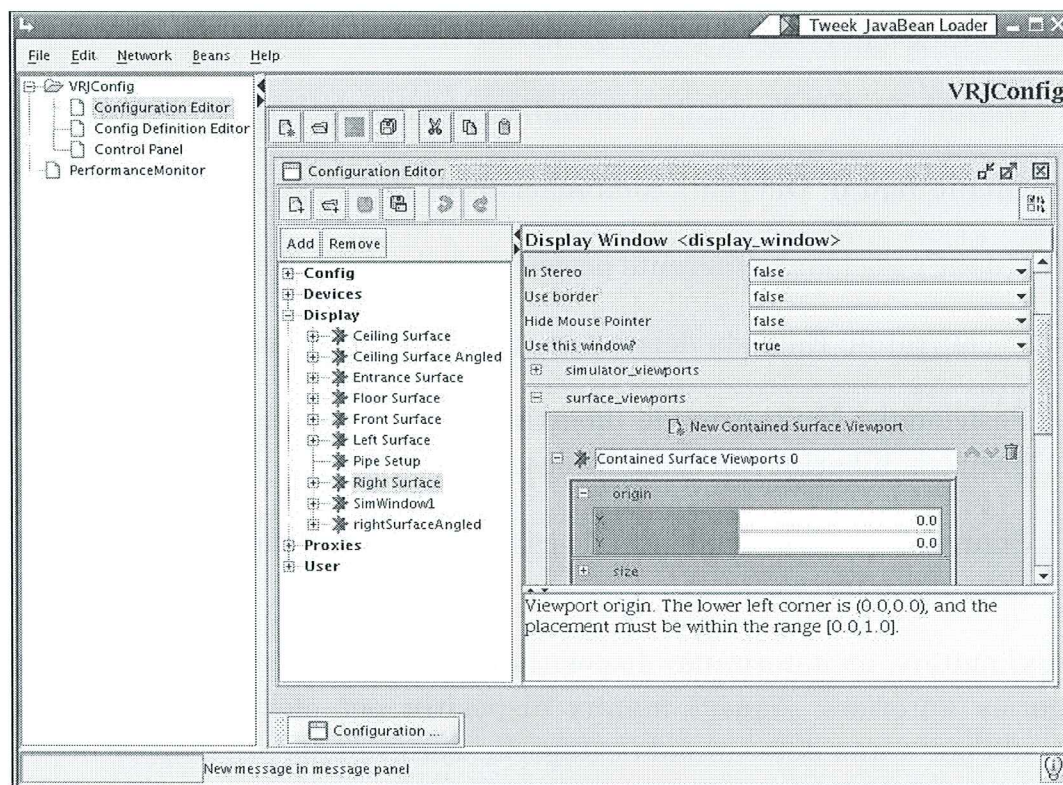


Figura X Captura de pantalla de VRConfig

### **Configuració de dispositius**

El procés de configuració de dispositius d'entrada depèn del dispositiu específic que s'estigui configurant. En concret, Gadgeteer té dues categories de dispositius: dispositiu físic i simulador de dispositius. El simulador de dispositius té com a principal objectiu imitar el comportament de dispositius físics de sis graus de llibertat utilitzant l'aportació d'un teclat i/o un ratolí. La configuració del simulador de dispositius és més complexa que la de dispositius físics.

## Sistema d'inspecció interactiva del procés de construcció d'una catedral

Gadgeteer ofereix també una classificació sobre els diferents dispositius que es poden configurar i que per tant són afegibles a l'arxiu de configuració. Les diferents categories són:

- Analògica: Les dades d'entrada són un rang continu amb dos valors mínim i màxim ben definits.
- Command: Comandes d'entrada discreta, normalment en forma de comandes parlades o gestos predefinitos.
- Digital: On/Off d'entrada, normalment corresponent a pitjar un botó.
- Guant digital: Diferents combinacions de dits.
- Guant posicional: Diferents posicions de la mà de l'usuari.
- Teclat/ratolí: Entrada dels events nadius del sistema de finestres.
- Posicionals: Multi-grau de llibertat de seguiment de dades. Fins a sis graus de llibertat són compatibles.
- String: Arbitrària seqüència de caràcters, generalment corresponent a paraules o frases.

Així mateix, un determinat dispositiu físic o simulat, caurà en una o més categories anteriors. Aquest mateix dispositiu pot oferir múltiples fonts d'entrada per a una determinada categoria. Per exemple, un dispositiu per a jocs que tingui vuit botons i dues palanques. Aquest dispositiu ofereix vuit fonts d'entrada digital, que es corresponen amb els botons, i quatre d'entrada analògica, dels dos eixos del joystick.

### ***Proxies***

Els dispositius proxies introdueixen una capa d'abstracció per garantir que les aplicacions vrjuggler no es converteixin només en aplicacions per a dispositius específics. El *proxy* actua com a un punter a una sola font d'informació des del dispositiu actual.

Com s'ha vist anteriorment, cada dispositiu d'entrada està dintre d'una classificació. Aquesta classificació també serveix per als dispositius *proxies*:

- Proxy analògic



- Proxy de comandes
- Proxy digital
- Proxy digital de guant
- Proxy de gest
- Proxy de teclat/ratolí
- Proxy posicional
- Proxy string

L'element de configuració del proxy per a cada tipus té una propietat que apunta a un dispositiu d'entrada de configuració de l'element corresponent al tipus de dispositiu.

La segona propietat per a cada tipus de proxy és una unitat d'identificació. Com s'ha dit anteriorment, cada dispositiu pot pertànyer a més d'una categoria. L'identificador del dispositiu proxy actua com a índex en base zero dintre de les diferents fonts del dispositiu.

### ***Visualitzadors***

A l'hora de configurar els visualitzadors, el *Display Manager* o manegador de visualització ha de ser un dels primers elements a configurar. El manegador de visualització és configurat per l'element "*Display System*" o "sistema de visualització".

Un altre element a configurar dintre dels visualitzadors són les finestres destinades a la renderització de gràfics, conegudes com *display windows*. Cada *display window* tenen la capacitat per visualitzar gràfics d'OpenGL, i internament, utilitzen una API segons la plataforma on són per tal d'obrir i manegar finestres OpenGL. Els *displays windows* tenen bastants elements a configurar, com per exemple, l'origen de la finestra dins del monitor, l'alçada, si la visualització serà estereoscòpica o no, o si sempre ha d'estar la finestra sempre visible, entre d'altres.

## 12.2 Les columnes

### 12.2.1 Alguns detalls de la construcció de la seu gòtica de Barcelona

La major part dels edificis monumentals actuals, si no tots, han estat erigits sobre edificis anteriors que han quedat soterrats en els fonaments, o que s'han incorporat a les noves estructures. El més corrent és que s'hi trobin elements de recuperació com a reble, etc. però també pot aparèixer el reaprofitament in situ de fragments valuosos o parts senceres.

La motivació per a aquestes actuacions sol ser el caràcter catastròfic de la fi dels monuments precedents, o aspectes econòmics i arquitectònics. Però també hi ha raons socials. En el cas d'esglésies i edificis religiosos hi havia raons litúrgiques que forçaven la conservació de parts sagrades o intocables. De vegades el desig de mantenir la continuïtat del culte o el respecte a tradicions arrelades podien condicionar notablement el disseny dels nous edificis i la salvaguarda de certs elements preexistents. En bastants casos ens trobem davant d'una **ampliació de l'església precedent** més que una **edificació** pròpiament dita. Elías Serra i Ràfols <sup>1</sup> va dir-ho ben clarament:

Com es feren les catedrals gòtiques? No es tracta d'edificis aixecats de planta damunt d'un pati ras, sinó ben al contrari, de la transformació paulatina d'un edifici anterior romànic que hom va enderrocant només a mesura que van restant acabades les parts o trams de la nova construcció. L'aspecte que devien presentar aquestes obres degué ésser més el d'un edifici en reparació que el d'una construcció nova... i així ... no suposava de cap manera la interrupció dels serveis de culte a què estava consagrat l'edifici.

Vegem com s'apliquen aquestes consideracions generals a tres importants catedrals gòtiques de Catalunya, començades coetàniament el segle XIV a partir de catedrals romàniques del segle XI: Santa Eulàlia i Santa Júlia d'Elna (1311), Santa Maria de Girona (1312) i Santa Creu i Santa Eulàlia de Barcelona (1298). Les tres (i tal vegada també Sant Just de Narbona) foren projectades amb una absoluta fidelitat a l'estructura romànica, a la que es van adaptar com el guant a la mà. L'edifici vell va ser literalment

---

<sup>1</sup> SERRA i RAFOLS, ELIAS, *La nau de la seu de Girona* Miscel·lània Puig i Cadafalch. Societat catalana d'estudis històrics. 1943-1951, p.185 i ss.

folrat amb noves capelles, en íntim contacte i comunicació amb l'àmbit romànic, sota l'empara de les ogives.

Per a la seu d'**Elna** tenim la sort de disposar d'evidència arqueològica perquè el projecte va quedar interromput i congelat el 1602, i ha quedat a la vista la part baixa de l'absis gòtic (*les remparts*), que embolca el cap romànic. També les sis capelles gòtiques del costat de l'epístola obrades entre 1341 i 1448, estan perfectament integrades en la nau romànica; mostrant claríssimament com es va resoldre el problema de la destrucció del mur romànic del SE, mantenint la possibilitat del funcionament simultani dels àmbits vells i els nous.

La documentació de la Seu de **Girona** ens fa saber que el 1312 el capítol gironí

statuit, voluit et ordinavit quod caput istius ecclesie de novo construeretur et hedificaretur, et circum ipsum capud novem capelle fierent,...

*Des de 1995 s'està aprofundint intensament en la informació arqueològica, havent-se excavat els peus de la nau i posat al descobert unes escales d'accés i d'altres elements d'interès, que clarificaran el procés constructiu tant de la seu gironina com de les seves germanes.*

Un primer punt fix del projecte de la seu de **Barcelona** va ser la necessitat de preservar el sepulcre de Santa Eulàlia de Barcelona, el cos de la qual està sepultat a la cripta de la seu des del 23 d'octubre de 877. Un segon, no menys important, va ser permetre la substitució del vell temple pel nou amb un mínim de canvis litúrgics i funcionals, mantenint el funcionament de culte durant la construcció. Els altars es varen traslladar mil·limètricament de les crugies romàniques a les capelles gòtiques contigües, sense canvis importants de les seves dedicacions.

L'ampliació de la seu de Barcelona es va començar el 1298 i la destrucció dels elements romànics fou tan intensa que es va perdre pràcticament tot rastre de la seu

## Sistema d'inspecció interactiva del procés de construcció d'una catedral

vella, fins al punt que l'any 1760 Antoni Campillo en la seva *Disquisitio methodi*<sup>2</sup> es preguntava: *On era el temple antic? Quan es va canviar?* Les seves respostes van orientar l'opinió dels historiadors fins a finals del segle XIX en què van començar a obtenir-se evidències arqueològiques, degut a les actuacions arquitectòniques d'August Font (1898), d'Adolf Florensa (1963) i de les troballes del període paleocristià (1965).

Devem també a Campillo (ibidem) la informació que *hi ha la tradició dels ancians que el Temple antic serví de bastida per edificar el Temple modern...* Aquesta tradició ens arriba a través d'un testimoni gairebé quatre segles posterior al procés de construcció, però no deixa de ser fiable, perquè una tradició de tipus tan tècnic difícilment és concebible com fruit de la mera imaginació popular. D'altra banda les paraules de Campillo es quedaven curtes per reflectir la realitat del procés constructiu, com es veu en la proposta de reconstrucció que avui guia als investigadors, establerta el 1978 mercès als treballs documentals de Vergés i Vinyoles.<sup>3</sup>

### **La construcció dels pilars gòtics a Barcelona.**

Els trets essencials de la reconstrucció indicada de la seu de Barcelona, que es poden veure amb detall a les referències esmentades són que la seu romànica era un edifici de tres naus, amb pilars rectangulars de 2,80 m aproximadament. Els trams de la nau central són quadrats, de dimensions de 9.49 m en quadre, aproximadament. La llum dels col·laterals és de 5,75 m, amidats entre l'eix dels pilars interiors i el centre del pilar virtual definit per la paret exterior. En línies generals, i en infinitat de detalls constructius, guarda una extraordinària semblança amb l'església de Sant Vicenç de Cardona, el principal paradigma de les grans esglésies romàniques del segle XI. Entre les plantes de Cardona i Barcelona hi ha una proporcionalitat força precisa (la seu de

---

<sup>2</sup> CAMPILLO, A. *Disquisitio methodi consignando annos aerae Christianae*, Barcelona, 1766. Pàg. 98, Nota 1, títol XXVIII. [Original en llatí que caldria reproduir aquí]

<sup>3</sup> VERGÉS, MARTÍ; VINYOLES, TERESA :

- Arxiu de la Catedral de Barcelona (ACB) *La seu romànica de Barcelona*, 1977, (inèdit).
- Article sobre la *Catedral Romànica de Barcelona*, a : **Santa Creu i Santa Eulàlia de Barcelona**, *Catalunya Romànica*, XX, *El Barcelonès*, *Enciclopèdia Catalana*, 1992, pàg. 154-165.
- *La seu romànica de Barcelona*, **LAMBARD**, *Estudis d'art medieval*, III, 1983-85 (1987), pàg. 97-102.

Barcelona és un 11% més gran en planta) si bé els col·laterals barcelonins són més amples, circumstància que es pot atribuir a l'estretor dels col·laterals de Cardona, derivada de la manca d'amplitud del solar cardonenc.

Les voltes col·laterals de Cardona són de creueria i en canvi les d'Elna són de canó seguit. A Barcelona la solució adoptada podria haver estat una o l'altra, i fins i tot la de mig canó.

Els pilars gòtics barcelonins queden tangents a la cara exterior dels pilars quadrats romànics. Els pilars gòtics tenen una amplada de 1,75 m i perforen els col·laterals que, com s'ha dit, tenen una llum de 5,75 m. El pla sagital en què se situen els pilars gòtics està reforçat per un arc toral, que sobresurt uns 20 cm. El gruix de la volta és de 25 cm. en mitjana; i arriba a 45 cm. amb el regruix de l'arc toral.

Un gran avantatge de l'esquema triat és que permet utilitzar al màxim la nau romànica *com a bastida de la gòtica*, amb notable estalvi i seguretat. Estem convençuts que aquest avantatge constructiu va ser conscientment provocat i aprofitat per l'autor gòtic del projecte de la seu gòtica de Barcelona (sembla que Enric de Narbona, també conegut com Enric Faveran i germà de l'escultor Jaume de Narbona).

El principal problema tècnic d'aquest plantejament és el següent: ¿És possible travessar les voltes romàniques amb pilars de les dimensions indicades sense provocar l'ensorrament de les voltes?

Alguns autors com Juan Bassegoda <sup>4</sup> es pronuncien per la negativa: “*Sería indispensable que se derribasen las naves laterales románicas para hacer los pilares góticos*”. Nosaltres dubtem que aquest solució dràstica fos indispensable i per tant, que l'ús del *Temple antic com a bastida per edificar el Temple modern* fos limitat.

Mancats d'informació directa, i d'evidències arqueològiques (que algun dia poden aparèixer) ens limitem a proposar algunes solucions constructives per permetre travessar

---

<sup>4</sup> BASSEGODA NONELL, JUAN (EI VII CENTENARIO DE LA CATEDRAL DE BARCELONA (1298-1998). *De la seo romànica a la gòtica*, Boletín de la Real Academia de Bellas Artes de San Fernando, núm. 87, Segundo Semestre de 1998, Madrid 1998)

## Sistema d'inspecció interactiva del procés de construcció d'una catedral

la volta romànica sense enderrocs parcials i sense l'obertura de grans esvorancs que posin en perill l'estabilitat de l'estructura romànica.

Com ja s'ha dit, s'haurien de considerar com a mínim tres casos de volta (canó, mig punt, i creuria); en presència o no d'arcs torals. Una primera estratègia, que ens sembla suficientment general per a abastar tots els casos, seria reforçar la volta romànica (en el punt de pas del pilar) amb un bordó de cert gruix, resistent, i fortament agermanat amb la volta. Si el pilar travessa la volta en punts allunyats dels estreps el bordó seria de planta ovalada. En punts pròxims a parets, arcs formers, o pilars romànics, el bordó podria ser obert, per exemple de forma catenària. El buidat o obertura de l'ull del bordó podria resultar innocu mercès a l'efecte arc del bordó que absorbeix les càrregues tangencials de vora.

El procés tindria certa complexitat, però sembla suficientment segur, senzill i econòmic; oi més per a constructors com els medievals, profunds coneixedors de la construcció en pedra i avesats a l'erecció de les immenses naus gòtiques. D'altra banda cal tenir present que els pilars gòtics no són mai de secció transversal entera sinó que estan constituïts per carreus situats perifèricament al voltant d'un nucli buit, reomplible amb reble. Els carreus es disposen en filades verticals, (que a Barcelona són d'uns 28 cm d'alçada), trencant o no juntes, dibuixant les típiques nervacions o columnetes. Per aquest motiu la construcció d'aquest tipus de pilar podria dividir-se en fases (dues per exemple), travessant la volta en dues o tres operacions paral·leles successives. D'aquesta manera el diàmetre de l'esvoranc obert en cada moment podria quedar substancialment reduït. No caldria tampoc eliminar la part més axial de l'arc toral i de la volta romànica, que podrien quedar incorporats al nucli del pilar, com eficient reble. És cert que l'indispensable *pas d'home* limita l'abast de la subdivisió de l'orifici, però el notable diàmetre (1,75 m) dels pilars és prou important com per permetre un mínim de dues fases. Acabada la col·locació dels carreus de la primera fase d'un pis, abans de destruir la segona part es podria en part recuperar la continuïtat de la volta, fent servir la capacitat resistent dels elements gòtics introduïts, amb algun apuntalament o faixat interior. Amb això es podrien millorar les condicions de seguretat de l'obertura de la segona fase o sigui la segona part de l'orifici.

Diguem finalment que el procés en alçat seria forçosament més complex, amb orificis guerxos que afectarien a més d'una filada horitzontal. Els pilars romànics (de 2,80 m de quadre) de la seu de Barcelona quedaven en contacte amb els gòtics, fet que permetia basar-s'hi per a petits treballs auxiliars de suport (bastides locals, apuntalaments, ancoratge d'hargues,

etc.). El cimbori romànic seria un cas especial, que no sembla que pogués presentar problemes. En conjunt, l'estalvi en bastides sembla impressionant i honora els seus inventors. Caldria, admetem, un estudi detallat del procés, cas per cas, per determinar la qualitat de les solucions exposades o els retocs convenients.

## 12.2.2 Les tres naus

Aclarit el problema de l'erecció dels pilars gòtics diguem que, malgrat l'evidència arqueològica de la catedral de Santa Eulàlia d'Elna, també ha preocupat a alguns autors la mera possibilitat de l'erecció d'una nau gòtica sobre una nau romànica de tres naus. L'arquitecte Joan Bassegoda en la mateixa referència esmentada anteriorment ho creu *difícil* i proposa un edifici romànic d'una sola nau:

*Sería relativamente sencillo construir una seo gótica de tres naves sobre una seo románica de nave única; así lo hicieron los constructores de las seos góticas de Vic y de Gerona (1312)... Pero hubiese sido muy difícil con un edificio de tres naves...*

Les raons d'aquesta dificultat es deuen al funcionament de les voltes de mig punt que han de transmetre als estreps els esforços transversals (o coces) generats per les càrregues. En el cas de voltes de tres naus la nau central necessita el contrafort de les col·laterals i recíprocament. Per aquest motiu l'enderroc d'una qualsevol de les naus posa en greu perill la veïna, i en definitiva l'estabilitat de tot el conjunt. Fins i tot un apuntalament previ a l'enderroc pot ser conflictiu per la dificultat de compassar el progrés dels puntals, i mantenir l'equilibri de les transferències d'esforç.

Ara bé el problema té fàcil solució. Perquè l'enderroc es pot efectuar per **llesques transversals**. La plasticitat i adaptabilitat de les voltes romàniques i la conseqüent col·laboració entre els trams contigus de la volta permet progressar axialment per passes comptades, sense dificultat ni riscos.

Joaquim Agulló i Batlle

Martí Vergés i Trias

Barcelona març de 2002

## 12.3 Manual d'usuari

### *Instal·lació i execució*

El primer pas que ha de realitzar l'usuari per tal de fer funcionar el programari és obtenir una còpia del mateix. Un cop obtinguda la còpia cal col·locar-la en qualsevol carpeta de la màquina on es vol executar l'aplicació.

Un cop realitzat el pas anterior, per tal d'executar l'aplicació cal fer *double click* a l'arxiu *VRJSimpleApp.exe* i l'usuari pot apreciar com l'aplicació s'inicia.

### *Funcionament*

Un cop executant l'aplicació, aquesta demana que s'esculli un arxiu de configuració corresponent per a la llibreria VRJuggler. Aquests arxius de configuració es poden trobar dintre de la carpeta de configuració anomenada *config*. Aquests arxius de configuració són diferents configuracions que es poden triar o no segons les necessitats. Dintre d'aquesta carpeta es poden trobar arxius amb extensió *.conf*, necessaris per al correcte funcionament de dispositius i visualització de l'aplicació.

Després de carregar l'arxiu de configuració es pot apreciar l'escena buida, amb un color de fons i amb la finestra principal. Per tal de carregar un model, s'ha de pitjar el botó amb text *Load Project*. Seguidament, apareix una finestra que permet escollir l'arxiu que conté el model de l'escena. El model de l'escena ha de ser un arxiu que té per extensió *.3ds*. Segons la complexitat del model, l'aplicació trigarà més o menys temps, però cal tenir paciència.

Una vegada s'ha carregat el model es pot; si anteriorment s'ha carregat un arxiu de configuració adequat, col·locar les ulleres polaritzades per tal de poder tenir sensació de que el que s'està veient es real. A més es pot navegar, és a dir, avançar, retrocedir, etc, tot utilitzant un dispositiu *wanda*.

Per tal d'activar la línia del temps, hi ha prou en pitjar el botó que conté el text *Open Time Line File*, la qual cosa permetrà seleccionar un arxiu que contingui les dades per a la línia de temps. Cal tenir en compte, que si el fitxer seleccionat no respecta el format de la línia de temps, l'aplicació no funcionarà correctament.



Si el format de l'arxiu seleccionat respecta el format que s'especifica sobre la línia del temps, és a dir, l'aplicació ha carregat correctament la informació, s'activarà el mode de la línia del temps i per tant hi ha la oportunitat de seleccionar l'any de construcció. Per tal de seleccionar l'any de la obra, es pot modificar la barra de desplaçament que apareix quan es carrega l'arxiu de la línia del temps. Igualment com abans, encara que estigui activada la línia de temps, es pot realitzar moviments per l'escena mitjançant el dispositiu wanda. Si es desitja desactivar el mode de línia del temps només caldrà tancar el diàleg que incorpora la barra de desplaçament.

Per tal de navegar per l'escena, només cal utilitzar un dispositiu *wanda* i apuntar en la direcció cap a on es vol navegar.

Per finalitzar l'aplicació, només cal pitjar el botó amb el text *Exit*.

## 12.4 Glossari

**Absidiola:** *f* Absis petit o secundari.

**Absis:** *m* 1 ARQUIT Part d'una església que s'obre normalment al fons de la nau central, on sol haver-hi el presbiteri i l'altar.

**Ara:** *f* 1 RELIG Altar per als sacrificis.

**Arc toral:** *m* l'arc transversal a la nau i que sustenta la volta.

**Atri:** *m* ARQUIT 1 Pati interior, generalment voltat de pòrtics.

**Cimbori:** *m* ARQUIT Construcció de forma poligonal vuitavada o cilíndrica que serveix de base a una cúpula i descansa damunt els arcs construïts en la intersecció de dues naus.

**Creuer:** ARQUIT 1 En algunes esglésies, nau transversal interposada entre la nau central i l'absis a fi d'augmentar, prop de l'altar, l'espai destinat als fidels.

**Deambulatori:** *m* ARQUIT Passadís que encercla el presbiteri d'algunes esglésies pel darrere.

**Fornícula:** *f* ART /CONSTR Buit deixat en el gruix d'una paret per a col·locar-hi una estàtua, un altar, etc.

**Intercolumni:** *m* ARQUIT Espai que hi ha entre dues columnes.

**Nàrtex:** *m* ARQUIT Vestíbul, pòrtic situat davant la porta d'una església.

**Presbiteri:** *m* Part de l'església, anomenada també santuari, situada al fons de la nau central, on hi ha l'altar major, i és reservada als clergues.

**Refectori:** *m* Refetor.

**Refetor:** *m* Menjador comú dels membres d'una col·lectivitat estable, generalment de monestirs o convents.

**Sagristia:** *f* 1 LITÚRG /ARQUIT A les esglésies, local annex, generalment tocant al presbiteri, on són custodiats els ornaments i els vasos sagrats i on

es revesteixen els sacerdots i ministres sagrats per a les funcions litúrgiques.

**Transsepte:** *m* ARQUIT Nau transversal d'una església; creuer.





