

UMEÅ UNIVERSITET
Institutionen för Datavetenskap
Laborationsrapport

11 april 2005

Virtuell Verklighet D

Laboration 1

”Första inlämningen”

Namn Patrik Axelsson
E-mail c00pan@cs.umu.se
Sökväg ~/edu/vr1/lab1

Handledare
Daniel Sjölie, Andreas Burman

Innehåll

1	Problemspecifikation	1
2	Åtkomst och användarhandledning	1
3	Systembeskrivning	1
4	Lösningens begränsningar	1
5	Problem och reflektioner	2
6	Testkörningar	2
A	Källkod	3
A.1	Geometry.cpp	3
A.2	Geometry.h	4
A.3	Group.cpp	5
A.4	Group.h	6
A.5	Light.cpp	7
A.6	Light.h	8
A.7	Material.cpp	9
A.8	Material.h	10
A.9	Node.cpp	11
A.10	Node.h	12
A.11	NodeCallback.h	13
A.12	NodeVisitor.cpp	14
A.13	NodeVisitor.h	15
A.14	RenderVisitor.cpp	16
A.15	RenderVisitor.h	17
A.16	Scene.cpp	18
A.17	Scene.h	19
A.18	SceneGraph.cpp	20
A.19	State.cpp	21
A.20	State.h	22
A.21	Texture.cpp	23
A.22	Texture.h	24
A.23	Transform.cpp	25
A.24	Transform.h	26
A.25	UpdateVisitor.cpp	27
A.26	UpdateVisitor.h	28

1 Problemspecifikation

Denna laboration gick ut på att skapa en scengraf med en objektorienterad uppbyggnad. Den skulle sedan användas för att rendera en egenhändigt skapad scen. Se originalspecifikationen för mer ingående detaljer:
<http://www.cs.umu.se/kurser/TDBD12/VT05/lab/lab1.html>

2 Åtkomst och användarhandledning

I katalogen för detta projekt finns en makefil som möjliggör kompilering genom att där exekvera kommandot `make`. Värt att nämna är att vid länkningen, länkas en statisk version av `vrlib` med, vilken finns i katalogen `~/edu/vr1/vrlib`.

När scenrafen är kompilerad har den exekverbara filen `SceneGraph` skapats i samma katalog, exekvering av den startar scenrafen.

Efter scenrafen startats kommer den att rotera runt scenen om och om igen, vilket ej går att styra manuellt. Man kan byta mellan fönster och fullskärmsläge med `f`-tangenter och avsluta med `q` eller `escape`-tangenter.

För bästa resultat - både sett till jämna rörelser och visuell kvalitet, rekommenderas det att aktivera synkronisering av bufferbyten i OpenGL till skärmens vertikala uppdateringsfrekvens. Tyvärr verkar det inte finnas något standardiserat sätt att åstadkomma detta. Körs scenrafen lokalt på en Linux-dator med `nVidia`-grafikkort och deras drivrutiner aktiverar man detta genom att sätta miljövariabeln `_GL_SYNC_TO_VBLANK` till `1`. I annat fall får manualen för grafikkortets drivrutin konsulteras.

3 Systembeskrivning

Hur den här lösningen skulle vara uppbyggd var relativt hårt specat i labspecifikationen. De tillägg som är gjorda är en `UpdateVisitor`-klass som ärver av `NodeVisitor`-klassen och en `NodeCallback`-klass.

Varje nod i scenrafen kan ha ett `NodeCallback`-objekt. När sedan `UpdateVisitor`n traverserar scenrafen anropar den de `NodeCallback`-objekt som noderna möjligen har. Dessa traverseringar av scenrafen sker med relativt konstant mellanrum, ej knutet direkt till frekvensen av renderingen. Det är på detta sätt som de olika animeringar i scenen åstadkoms, men möjligheterna för vad `NodeCallback`sen kan användas till är i stort sett obegränsade.

`NodeCallback`-klassen är egentligen bara en tom basklass som man ärver ifrån och skapar sen klasser som bestämmer vad som ska ske när `NodeCallback`-objekten anropas.

4 Lösningens begränsningar

Som det är nu är det lite bökigt att skapa en scen. Det vore fördelaktigt om man hade skapat ett filformat för att beskriva hela scenens uppbyggnad. Nu laddas

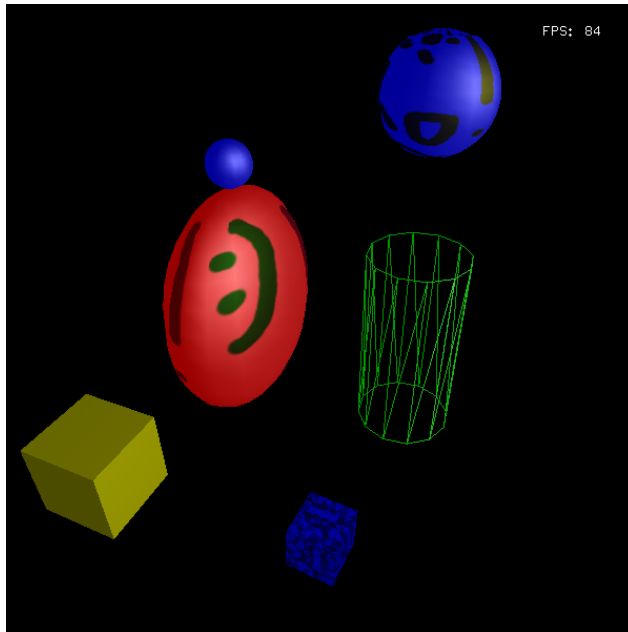
istället scenens objekt in från en fil i AC3D-format och sen krävs handpåläggning i koden för att strukturera upp scenen.

5 Problem och reflektioner

Största problemet har varit dålig planering. Började för sent med laborationerna och var inte färdig med dom när projektet drog igång. Sen under projekt-tiden blev det inte något mer gjort då vi la ner i princip alla vakna timmar på det.

Annars måste jag säga att det har varit en väldigt rolig och lärorik laboration. Måste även ge en eloge till den objektorienterade uppbyggnaden av scengrafan i labspecifikationen - logisk och effektivt, som det ska vara.

6 Testkörningar



Figur 1: scenen

På den här bilden syns den scen som skapats med scengrafan. Sex objekt är synliga, men egentligen består scenen endast av fyra objekt. De två blå bollarna är samma objekt, men med olika states och transformationer ovanför sig. Samma sak gäller för de två kuberna - de är samma objekt med olika states och transformationer ovanför sig.

Den röda bollen, den större blåa bollen och den undre kuberna har material och texturer. Den mindre blåa bollen och den gröna cylindern har enbart ett material, men den senare visas som en wireframe. Den gula kuberna har varken material eller textur utan enbart en färg.