Laboration 4 – OpenGL

Syfte

Erbjuder studenterna en möjlighet att lära sig grunderna i OpenGL.

Genomförande

Genomförs *individuellt* eller i grupp om 2 personer.

Uppskattad tidsåtgång

10-25 timmar effektiv arbetstid per gruppmedlem.

Informations- och bakgrundsmaterial

- En kort introduktion till OpenGL.
- Kapitel i kursboken om transformationer.
- Översikt över OpenGL-kommandon (kursbunten).
- *C för den som kan Java.*
- Utdelat material om JPEG.
- Utdelat material om hur man kompilerar OpenGL-program på PC.
- Artiklar om animation (kursbunten)
- Föreläsnings- och övningsanteckningar, ev. annat utdelat material.
- <u>http://www.opengl.org</u>
- Google!

Uppgift

Animera en 3D-häst i realtid.

Redovisning

Redovisas per e-post till <u>gustavt@csc.kth.se</u> senast **fredag 20 april kl. 23.59**. Följande ska finnas med i redovisningen:

- 1. Studentens(ernas) namn och personnummer.
- 2. C- eller C++-källkoden för *deluppgift* 2 *och* 3 *separat*. Skicka gärna med en *färdigkompilerad version* också!
- 3. *En kopia av den textur* du/ni använt!

Förslag på uppdelning (om ni jobbar i par)

Den här labben har ingen naturlig uppdelning – det bästa är om ni arbetar sida vid sida och hjälper varandra. *Dela inte upp deluppgifterna mellan er och genomför dem ensamma – det betraktas som fusk!* Gör labben individuellt om ni inte har tid att sitta tillsammans.

Detaljbeskrivning

Den här laborationen går ut på att bygga och animera en hierarkisk modell. Slutresultatet är ett "hippokvarium" som visar en animerad häst (eller annat fyrfota djur). Via menyer ska man kunna starta och stoppa animationen och visa den från minst två olika kamerapositioner i 3D.

Laborationen är uppdelad i två delar. Den första delen fokuserar på GLUT (fönster och händelsebaserad hantering) och den andra delen fokuserar på OpenGL.

Deluppgift 1: Få igång Visual Studio

Börja med att kopiera den katalog i kurskatalogen som heter OpenGL till din arbetsdator. Du behöver ungefär 15Mb ledigt utrymme. Använd programmet *Start*→*SSH Secure Shell*→*Secure File Transfer Client* för att komma åt kurskatalogen /afs/nada.kth.se/info/grip07/.

Dubbelklicka på OpenGL.sln för att öppna Visual Studio. Om du blir frågad vilka inställningar du vill ha, välj "C++". Se till att fliken "Solution Explorer" är vald i rutan till vänster och klicka på +-ikonen invid "shaders"-raden. Öppna upp de tre foldrarna "Source Files", "Header Files" och "Resources". Du bör nu kunna se de filer som ingår i projektet. Dubbelklicka på main.c för att öppna den i editorn.

<u>Viktigt</u>: Nästa sak du måste göra är att ange vilken katalog programmet ska startas i när det körs. Högerklicka på OpenGL-projektet i "Solution Explorer" och välj *Properties...* Klicka på *Debugging* i rutan till vänster och skriv in katalogen Debug som *Working Directory* i rutan till höger:

	H 🔛	Build)3		
	E 🗁 B	Rebuild			
		Clean	ır		
	🖻 🗁 S	Project Only	•		
		Profile Guided Optimi	zation 🕨 e		
		Deploy	la		
		Custom Build Rules	n		
		Tool Build Order	1		
		Add	•		
		References			
		Add Web Reference			
		Set as StartUp Projec	ct h		
		Debug	•		
	1	Add Solution to Sour	ce Control		
	×	Cut			
	1	Paste			
		Remove			
	Solutio	Rename	-		
	Output				
	Show output fr	Properties			
		i iopenies			
shaders Property Pages					?×
shaders Property Pages Configuration: Active(Debug)	Platform:	Active(Win32)		Configuration I	? × Manager
shaders Property Pages Configuration: Active(Debug)	Platform: Debugger to launch:	Active(Win32)		Configuration	?≍ Manager
shaders Property Pages Configuration Active[Debug] Common Properties Configuration Properties	Platform: Debugger to launch: Local Windows Debu	Active(Win32) gger		Configuration	?≍ Manager
shaders Property Pages Configuration Active[Debug] Common Properties Configuration Properties Configuration Properties Configuration	Platform: Debugger to launch: Local Windows Debu	Active(Win32) gger	#IT avail Path)	Configuration I	? ≍ Manager
shaders Property Pages Configuration Active[Debug] E: Common Properties Configuration Propert	Platform: Debugger to launch: Local Windows Debu Command Command	Active(Win32) gger	\$(TargetPath)	Configuration	₹ × Manager
shaders Property Pages Configuration Active[Debug] E: Common Properties Configuration Propert	Platform: Debugger to launch: Local Windows Debu Command Working Directory	Active(Win32) gger	\$(TargetPath)	Configuration	z x
shaders Property Pages Configuration Active[Debug] Configuration Properties Configuration Prope	Platform [Debugger to launch: Local Windows Debu Command Working Directory Debugger Type	Active(Win32) gger	\$(TargetPath) No Auto	Configuration	z x
shaders Property Pages Configuration Active[Debug] Configuration Properties Configuration Prope	Platform Debugger to launch: Local Windows Debu Command Working Directory Debugger Type Erwironment Debugger Type Erwironment	Active(Win32) gger	\$(TargetPath) No Auto	Configuration	z z Manager
shaders Property Pages Configuration Active[Debug] Configuration Properties Configuration Pro	Platform Debugger to launch: Local Windows Debu Command Command Working Directory Morking Directory Environment Merge Environment Merge Environment SQL Debugging	Active(Win32) gger No	\$(TargetPath) No. Auto Yes No	Configuration	عام معام معام معام معام معام معام معام
shaders Property Pages Contiguration: Active(Debug)	Platform [Debugger to launch: Local Windows Debu Command Council and Council Working Directory Debugger Type Environment Merge Environment SQL Debugging	Active(Win32) gger	\$(TargetPath) Mo Auto Yes No	Configuration	₹≍ Manager • • • • • • • • • • • • • • • • •
shaders Property Pages Contiguration: Active(Debug)	Platform [Debugger to launch: Local Windows Debu Command Council and Council Working Directory Debugger Type Environment Merge Environment SQL Debugging	Active(Win32) gger	\$(TargetPath) Mo Auto Yes No	Skriv De	₹≍ Manager • • • • • • • • • • • • • • • • • •
shaders Property Pages Configuration: Active(Debug)	Platform Debugger to launch: Local Windows Debu Command Command Debugger Type Debugger Type Environment Merge Environment Merge Environment SQL Debugging	Active(Win32) gger	\$(TargetPath) No Auto Yes No	Skriv De	₹≍ Manager • • • • • • • • • • • • • • • • • •
shaders Property Pages Configuration Active(Debug) Common Properties Configuration Properties	Platform Debugger to launch: Local Windows Debu Command Command Debugger Type Environment Merge Environment Merge Environment SQL Debugging	Active(Win32) gger	\$(TargetPath) Mo Auto Yes No	Skriv De	₹≍ Manager > bug här!
shaders Property Pages Configuration Active(Debug) Common Properties Configuration Properties C	Platform: Debugger to launch: Local Windows Debu Command Debugger Type Debugger Type Environment Merge Environment Merge Environment SQL Debugging	Active(Win32) gger	\$(TargetPath) No Auto Yes No	Skriv De	₹ Manager • • • • • • • • • • • • • • • • • •
shaders Property Pages Configuration Active(Debug) Configuration Properties Configuration Pro	Platform: Debugger to launch: Local Windows Debu Command Command Debugger Type Ervironment Merge Ervironment Merge Ervironment SQL Debugging	Active(Win32) gger	\$(TargetPath) No Auto Yes No	Skriv De	₹ Manager • • • • • • • • • • • • • • • • • •
shaders Property Pages Configuration Active(Debug) Configuration Properties Configuration Pro	Platform: Debugger to launch: Local Windows Debu Command Command Debugger Type Erwironment Megge Erwironment Megge Erwironment SQL Debugging	Active(Win32) gger	\$(TargetPath) No Auto Yes No	Skriv De	₹ Manager bug här!
shaders Property Pages Configuration Active(Debug) Configuration Properties Configuration Pro	Platform. Debugger to launch: Local Windows Debu Command Command Command Debugger Type Environment Merge Environment Merge Environment SQL Debugging	Active(Win32) gger	\$(TargetPath) No Auto Yes No	Skriv De	₹ Manager bug här!
shaders Property Pages Configuration Active(Debug) Configuration Properties Configuration Pro	Platform Debugger to launch Local Windows Debu Command Command Debugger Type Environment Merge Environment SQL Debugging	Active(Win32) gger	\$(TargetPath) No Yes No	Skriv De	₹ Manager bug här!
shaders Property Pages Configuration: Active(Debug) Configuration Properties Configuration Pr	Platiom: Debugger to launch: Local Windows Debu Command Command Debugger Type Environment Merge Environment SGL Debugging Command The debug command	Active(Win32) gger M	\$(TargetPath) No Yes No	Skriv De	IX

Pröva nu att kompilera! Välj *Build*→*Build Solution* i menyn (eller tryck <Ctrl>+<Shift>+). Du bör inte få några kompileringsfel. Om du får det och inte själv kan klura ut vad det är som är fel, kontakta en handledare. För att köra programmet, välj *Debug*→*Start* (eller tryck <Ctrl>+<F5>). Du bör få ett blått, tomt fönster. Tryck <q> eller <Esc> för att stänga det.

Deluppgift 2: GLUT

Den första deluppgiften är att skriva en GLUT-stomme som hanterar det fönster och de menyer som behövs för ditt program.

För att bli godkänd krävs att din stomme har möjlighet att i en meny välja animation på/av, två olika gångarter (t.ex. trav och galopp), och två olika kamerapositioner. Valalternativen ska ligga i submenyer så att inte gränssnittet blir för "plottrigt".

Din stomme ska kunna visa vilket menyalternativ som valts. Detta görs på följande sätt: i fönstret ska två geometriska föremål, indikatorer, visas. Det första föremålet representerar gångart och det andra animation på/av. Föremålens färg ska sedan avspegla vilket menyalternativ som är valt. Säg t.ex. att vi valt en sfär för att representera gångart och en kub för att representera animering på/av. Då kan t.ex. sfären vara blå för trav och gul för galopp

och kuben vara grön för på och röd för av. Använd gärna glColor-funktionen för att sätta färgen. Byte av kameraposition ska fungera och du måste använda perspektivprojektion.

Deluppgift 3: OpenGL

Den andra deluppgiften går ut på att bygga och animera hästen. Du ska utgå från "hippokvariestommen" du skrev i första deluppgiften och ersätta indikatorerna med en 3Dscen som visar hästen. Val i menyerna ska fungera som förväntat – väljer man animation på, t.ex., ska animationen starta. Du måste använda perspektivprojektion, annars går det inte att se "djup" i bilden.

Hästmodellen ska vara hierarkisk i minst tre nivåer, d.v.s. benen ska ha knäled.

Modellen ska använda minst två olika OpenGL-material (det duger inte att sätta färgen med glColor) och ska vara belyst av minst en OpenGL-ljuskälla.

Modellen ska animeras i minst två distinkta gångstilar (att öka hastigheten på en gångstil räknas inte som en ny gångstil). Hästen ska röra sig runt i en cirkel (det räcker inte att hästen "går på stället"). Gångart ska styras från menyn du gjorde i stommen.

Hästen ska ha ett golv eller en gräsmatta att gå på. Denna ska ha en textur. Texturen kan antingen laddas från en JPEG-fil (se JPEG-dokumentationen) eller skapas enl. texturexemplet i OpenGL-kompendiet.

Tips: Kommandot glGetError() kan användas för att ta reda på om man skickat felaktiga parametrar till OpenGL. Lägg gärna till ett sådant test precis före du gör glutSwapBuffers(). För information om glGetError(), se <u>www.opengl.org</u> eller Googla.

Förslag på arbetsgång

- 1. Börja med att skapa hästens kropp och se till att den rör sig i en cirkel när man sätter på animeringen. Kontrollera att du sätter PROJECTION-matrisen till en perspektivprojektion och välj sedan lämpliga kamerapositioner genom att modifiera MODELVIEW-matrisen.
- 2. Lägg till "överben" och se till att de följer med kroppen runt i cirkeln. Kontrollera att benen inte "flyger iväg" vid animation (låt dem t.ex. rotera kontinuerligt).
- 3. Lägg till resten av hästens ben och hals/huvud.
- 4. Skapa de två gångstilarna.
- 5. Lägg till ljuskällor och sätt material på de olika hästdelarna.
- 6. Lägg till en markpolygon och se till att den sitter på rätt ställe.
- 7. Lägg till texturen på markpolygonen.

Om du har tid och lust...

- Leta reda på ett filmklipp som visar ett fyrfota djur i rörelse och försök få dina gångstilar att bli så realistiska som möjligt!
- Om du ritat hästen med glutSolid()-anrop, rita den istället med OpenGLprimitiver och se till att ge hästen en textur också! (glutSolid-anropen skickar inga texturkoordinater till OpenGL.)
- Pröva att ge hästen en skugga på marken genom att "platta till" den med ett glScale-anrop och rita ut den med svart färg i höjd med marken!