Introduktion till 3dsMax

TNM061/TNGD25 Lab 3: Ljussättning och rendering

Contents

1 3dsMax och renderare

3dsMax levereras med flera olika alternativa renderare, som har olika tillämpningsområden och även i grunden helt olika funktionssätt. För här laborationen kommer vi att använda 3dsMax version 2019, men växla mellan två olika renderare för att visa skillnaderna mellan dem och för att förklara grunderna tydligare. De renderare som gör beräkningarna på traditionellt sätt med den lokala datorns CPU är Scanline Renderer, ART (Autodesk Raytracer) och Arnold. Dessutom finns en molntjänst, Autodesk 360 Cloud Rendering, där bilderna renderas av en server på nätet i stället för på den lokala datorn, och renderaren Quiksilver som använder grafikkortet för att snabba upp vissa av beräkningarna. Vi kommer att använda *Scanline Renderer* och *Arnold* i den här laborationen. Scanline Renderer är en gammaldags renderare som inte ger så otroligt realistisk ljussättning, men den är snabb, och dess renderingsmetoder är fortfarande relevanta att förklara eftersom de liknar dem man i dag oftast använder i realtidsrendering för t ex spel. ART och Arnold är betydligt modernare men mer beräkningskrävande så kallade *path tracers*. Path tracing är en utveckling av ray tracing som är förhållandevis beräkningstung, men som blivit användbar på senare år. Nackdelen, förutom att renderingen tar längre tid att beräkna, är att man får brus i bilderna. Rendering med path tracing är fortfarande en avvägning mellan hur snabbt man vill ha fram bilden och hur mycket brus man är beredd att tåla.

Vilken renderare du ska använda väljer du i *Render Setup*, antingen via menyn eller via knappen med en tekanna och ett kugghjul, längst till vänster i *Toolbar* av dem som har tekannor i ikonen, under *Production Rendering Mode*.

2 Ljuskällor

Ljussättning är en viktig del av datorgrafik, såväl estetiskt som tekniskt. En bra ljussättning är väldigt viktig för helheten, och en dålig ljussättning kan förstöra en i övrigt välgjord scen. Ljussättning i verkligheten för fotografi, film och scenbruk är en svår konst som det tar lång tid att lära sig, och för att göra det bra krävs det övning, talang, tålamod och ett kritiskt öga för resultatet. Detsamma gäller för ljussättning i datorgrafik, och det här är inte alls någon introduktion till estetiken i ljussättning. Det är däremot en introduktion till tekniken bakom ljusberäkningarna som sker i en modern renderare, och även om du inte blir så vidare bra på att göra en snygg ljussättning efter bara ett fyratimmars labpass så får du åtminstone erfarenhet av att hantera de grundläggande verktygen, och förhoppningsvis en insikt i vilka metoder som finns.

Vi börjar med att titta på den enklare renderaren *Scanline Renderer*, så välj den i *Render Setup*.

Vi behöver förstås också något att ljussätta. Skapa en enkel scen av boxar som ser ut ungefär som i figuren nedan. Skapa ett material av typen *Standard* med enbart diffus reflexion (*Specular Level* = 0), och sätt dess diffusa färg till ljust grått. Applicera materialet på samtliga objekt i din scen.

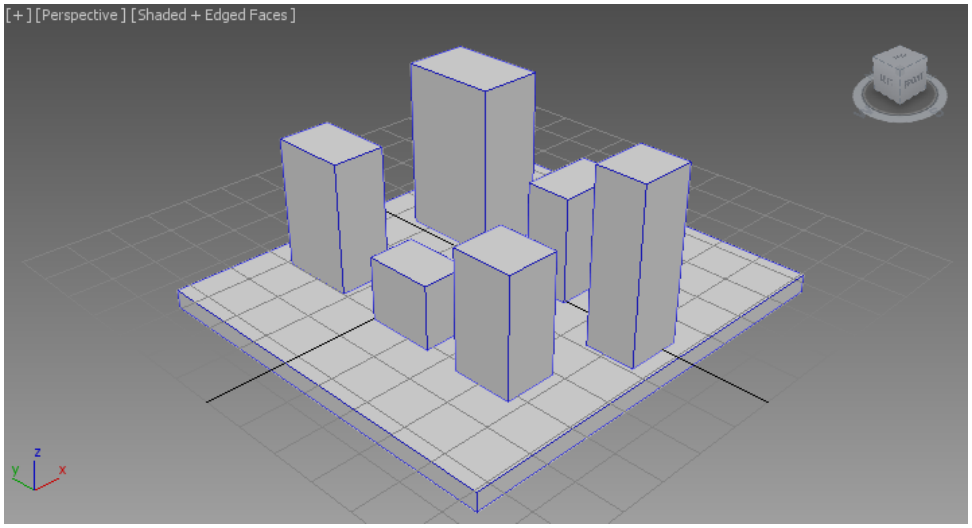

Figure 1: En enkel scen för experiment med ljussättning

Rendera. Resultatet ser platt ut, eller hur? Om du inte skapar någon ljuskälla i scenen så använder 3dsMax en standardljuskälla som är väldigt enkel. Den kastar till exempel inga skuggor. Dess syfte är bara att lysa upp scenen så att bilderna du renderar inte blir helt svarta innan du lagt in några egna ljuskällor. När du lägger in en egen ljuskälla i scenen så avaktiveras den här trista ljuskällan.

2.1 Enkla ljuskällor

Skapa nu en ljuskälla av typen *Target Direct*. ”Direct” betyder i detta fall inte ”direkt”, utan det är en litet olämplig förkortning av ordet ”directional”, vilket avser en riktad ljuskälla med parallella strålar.

Rikta in ljuskällan så att den lyser mot din scen snett uppifrån, och ställ in storleken på ”ljuskonen”, det som kallas *hotspot* och *falloff* i *Directional parameters* under *Modify*-fliken, så att alla objekt i din scen är belysta, eventuellt med undantag för delar av golvet/marken. För att enkelt se vartåt ljuskällan är riktad och hur stort område den belyser så kan du klicka på titeltexten högst upp till vänster i en viewport, där det exempelvis står ”right”, ”front” eller ”perspective”, och välja att i den viewporten titta i ljuskällans riktning från dess aktuella position. Kontrollerna längst ner till höger i programmets huvudfönster, de som används för att rotera och zooma i vyn, kommer då att flytta på ljuskällan och ändra dess spridning. Testa gärna! Det här är ofta ett väldigt smidigt sätt att rikta in en ljuskälla, även om det kan verka litet bakvänt.

När du riktat in ljuskällan, rendera scenen igen. Notera att inte heller den nya ljuskällan kastar skuggor. Du måste nämligen gå in i *Modify*-fliken och slå på dem. Under egenskaperna för ljuskällan finns en rubrik *Shadows*. Kryssa i rutan *On* för att slå på skuggorna, och rendera igen. Skuggorna är kanske litet väl hårda och mörka, och objekten kanske skuggar varandra på ett fult sätt, men skuggorna finns där, och det är en väsentlig förbättring.

Den sorts skuggor som används om man inte ber om annat är av typen *Shadow map*. Det är ett litet gammaldags men snabbt och enkelt sätt att göra skuggor, och det kan i många fall vara tilräckligt.

En skuggmapp *(shadow map)* är en avståndsbild (en bild över z-djupet för varje pixel i xy-planet) som renderas från *ljuskällans* position i stället för från kameran, och som sedan används för att avgöra om en viss yta är belyst eller ligger i skugga. Den yta som är belyst är helt enkelt den som ligger närmast ljuskällan, vilket är den som syns när man renderar en bild från ljuskällans position. Ytor som inte syns från ljuskällan ligger däremot i skugga. (För detaljer om hur *shadow mapping* fungerar hänvisar vi till andra källor, t ex någon bok om datorgrafik eller kursens föreläsningsanteckningar. Wikipedia har också en bra artikel om ämnet.) Avståndsbilden som renderas har en begränsad upplösning, och det syns när du renderar en närbild av skuggans kant. För att få ett bra resultat med *shadow maps* behöver du ställa in upplösningen på dina skuggmappar så att de är tillräckligt detaljerade för de vyer du vill rendera. Inställningen för skuggmappens upplösning ligger litet halvt gömd under *Shadow Map Params*, men den finns där och heter *Size*. Standardinställningen 500 innebär att den förberäknade skuggmappen är 500x500 pixels stor. Det är fullt tillräckligt i många fall, men inte alltid.

Shadow mapping är användbart, men numera har det sin främsta tillämpning i realtidsrendering, där man måste använda enkla metoder för att det ska gå tillräckligt snabbt. Ett modernare sätt att göra skuggor, som också liknar det sätt som används i renderaren som vi skall använda senare (*Arnold*), är att använda ray tracing. Under *General Parameters*, *Shadows*, byt typ på skuggan till *Ray Traced Shadows* och rendera igen. Ser du någon skillnad? Om inte, zooma in litet på scenen så att du ser kanten på en skugga i närbild.

En fördel med shadow maps, förutom att de går snabbt att beräkna, att det är lättare att göra suddiga skuggor med den metoden. Raytrace-skuggor blir skarpa, så skarpa att de lätt uppfattas som hårda och orealistiska. Vi går inte närmare in på här hur man suddar till skuggor med shadow maps. Det rör sig om ganska enkla bildbehandlingsoperationer i skuggmappen, men att förklara detaljerna går litet utanför den här kursens ramar. Vi beskriver i stället hur man med extra beräkningar kan använda raytracing för att beräkna även suddiga skuggor. Moderna datorer är snabba, och det är inte alltid klokt att välja den snabbaste renderingsmetoden. Raytracing av skuggor är mer generellt än skuggmappar, och kan ge ett bättre och mer fysikaliskt korrekt resultat utan att man lägger ner en massa manuellt arbete. För rendering där man inte kan använda raytracing, till exempel vid realtidsrendering i datorspel, så är däremot skuggmappar fortfarande en vanlig och viktig metod.

En riktad ljuskälla simulerar ljus som kommer långt bortifrån, så långt bort att strålarna kan anses vara parallella. En ljuskälla som beter sig så är solen, som är så långt bort att alla punkter på jordytan har i stort sett samma riktning på solljuset i varje givet ögonblick. (Ljuset har naturligtvis inte samma riktning relativt jordytan, men det beror på att jordytan är krökt.) Andra ljuskällor än solen ligger däremot ofta så pass nära det de belyser att de lyser i olika riktning i olika delar av scenen. En sådan ljuskälla är *Target Spot*. Skapa en sådan i stället för din *Target Direct* och studera skillnaden. Du behöver inte ens skapa en ny ljuskälla, du kan faktiskt byta typ på den du redan har i en drop-down meny högst upp i Modify-panelen för ljuskällan. Byt typ till *Spot*. Ljuset kommer nu att *divergera*, det utgår från en punkt och sprider sig i scenen. Skuggorna kommer att peka bort från ljuskällan åt litet olika håll. Effekten blir starkast om ljuskällan placeras nära objekten.

Egentligen ska ljuset också avta i intensitet med avståndet, men det finns en del

problem med att räkna på intensiteten om man gör så. Testa med att slå på det som kallas *Decay* för din spotlight! Välj det som enligt fysikens lagar borde vara korrekt, *Inverse square*, alltså att intensiteten minskar med kvadraten på avståndet till ljuskällan. Du kommer förmodligen att se en mycket mörkare scen. Dra upp *Multiplier* för att öka intensiteten tills det ser OK ut. Som du ser blir det en stor skillnad mellan objekt som ligger nära ljuskällan och objekt som ligger längre bort, och det ser litet för dramatiskt ut. Det beror på att den virtuella kameran ger skarpare kontraster jämfört med hur riktiga kameror och mänskliga ögon fungerar. I stället kan man välja "Inverse", ljus som avtar proportionellt med avståndet i stället för avståndet i kvadrat. Det är egentligen helt orealistiskt, men kan se bättre ut i många fall. Testa! Du behöver förmodligen ändra *Multiplier* igen för att få lagom mycket ljus.

2.2 Val av renderare

Vi har hittills använt *Default Scanline Renderer*. Den är litet gammaldags, men väldigt snabb och tillräckligt bra för vissa ändamål. För större delen av den här laborationen kommer vi däremot att använda en modernare renderare: path tracern *Arnold*. Arnold är betydligt långsammare, men den ger i gengäld mycket mer realistiska bilder just när det gäller ljussättning och skuggor. Till skillnad från många andra klassiska ray tracers räknar dessutom Arnold i väldigt stor utsträckning "rätt" i stället för att fuska med diverse specialmetoder som ser OK ut men inte har så mycket med verkligheten att göra. Den approachen har blivit vanligare under de senaste tio åren, allteftersom datorerna blivit snabba nog för att hantera de mer omfattande beräkningarna.

Under *Render Setup* (knappen med en tekanna och ett kugghjul, längst till vänster i *Toolbar* av dem som har tekannor i ikonen), välj *Renderer:* till *Arnold*.

Rendera din bild igen. Nu får du flera varningar som talar om att Arnold inte förstår dina material, och inte heller din ljuskälla, och bilden blir svart. Skapa ett material av en sort som Arnold förstår, förslagsvis av typen *Lambert* (ett enkelt, rent diffus material med väldigt få inställningar), sätt dess *Color* till ljusgrått, och applicera det på alla dina objekt. Skapa sedan en ljuskälla av typen *Arnold Light*, placera den på ungefär samma ställe som din Spotlight, och ta bort alla tidigare ljuskällor. Rendera. Du kan behöva ändra ljuskällans intensitet (inställningen *Intensity*) för att bilden ska bli lagom ljus.

2.3 Arealjuskällor

Verkliga ljuskällor ger inte helt skarpa skuggor. Det beror på att de inte är punktformiga, och därför blir det inte en abrupt övergång från ljus till skugga. I stället försvinner ljuskällan gradvis bakom ett objekt när man går från ljus till skugga. Den region där ljuskällan är delvis skymd går på engelska under det latinska namnet *penumbra*. På svenska säger man helt enkelt halvskugga, vilket är en direkt översättning. Alla ljuskällor har en viss utbredning, även solen, och ger åtminstone en liten region med halvskugga. Det kan vara viktigt att simulera det i renderingen, och det gör man med så kallade *arealjuskällor* (area light sources). En ljuskälla av typen Arnold Light är med automatik en arealjuskälla. Formen och storleken på den lysande ytan ställer du in under *Shape* i Modify-fliken för ljuskällan. Du ser en representation av ljuskällans area i viewporten. Experimentera med form och storlek för att se vilken inverkan det har på den renderade

bilden! Notera att skuggorna blir skarpare där det skuggande objektet ligger nära ytan, men mer suddiga när objektet är längre bort.

Renderingen av halvskugga baserar sig på att man inte bara tittar på en enda punkt mitt på ljuskällan när man beräknar om den syns eller inte, utan i stället tittar på flera punkter, tar flera *sampel*, och räknar hur stor andel av dem som syns. Antalet sampel för en viss arealjuskälla kan styras med inställningen *Samples* under rollouten *Rendering* i Modify-panelen. Generellt sett behövs det fler sampelpunkter för stora ljuskällor och för ljuskällor som ligger väldigt nära objekt, det som brukar kallas "mjukt ljus" (*soft light*).

Antalet sampel för mjuka skuggor kan också ställas in för scenen som helhet under de globala sampelinställningarna för Arnold. Under *Render Setup* → *Arnold Renderer Sampling and Ray Depth* finns en rad inställningar för att tala om för Arnold vad man anser att algoritmen ska lägga krut på att beräkna. Inställningarna är många och inte alltid helt uppenbara, och för att få koll på läget behöver man experimentera en del med mer komplicerade scener, och helst också läsa litet i dokumentationen. Att ställa in Arnold så att bilderna blir bra utan att det tar onödigt lång tid att beräkna dem tar en stund att lära sig, och vad som är en "bra" inställning beror i hög grad på vilka material och vilka ljuskällor man har i scenen. Vi går inte in på de detaljerna här.

2.4 Sekundärt ljus

Du har kanske märkt att de skuggade delarna av din scen ändå blir litet svagt upplysta när du renderar med Arnold. Var kommer det ljuset ifrån egentligen?

Svaret är att det är vad som kallas *sekundärt ljus* eller *indirekt ljus*, ljus som studsar mot omgivande objekt och sedan landar även på ytor som inte är direkt belysta. Det här beskrivs tyvärr inte särskilt noga i grundböcker om datorgrafik, men under de senaste åren har datorer blivit så snabba att man i dag rutinmässigt använder renderingsmetoder som ganska nyligen var omöjliga att använda i produktion. Sekundärt ljus ingår som en integrerad del av hur *Arnold* beräknar bilderna. I enklare renderare kan sekundärt ljus simuleras med två olika metoder som kallas *Final Gather* och *Photon mapping*, metoder som faktiskt inte har så värst många år på nacken.

Final Gather går enkelt uttryckt ut på att man först beräknar en bild med endast direkt ljus, oftast i lägre upplösning, och sedan använder den i ett andra renderingspass för att få med det indirekta ljuset i renderingen.

Photon mapping går enkelt uttryckt ut på att man först skjuter ut "fotoner", ljusstrålar, från alla ljuskällor och håller reda på var de landar i scenen. Vid rendering tittar man sedan dels på direkta ljuskällor, dels på indirekt ljus från fotoner som passerat genomskinliga objekt eller studsat mot blanka objekt.

En litet mer ingående teknisk beskrivning av såväl *Final Gather* som *Photon mapping* hittar du t ex på Wikipedia. Arnold stödjer ingen av dessa metoder eftersom den har som mål att försöka räkna "rätt", och metoderna ifråga är inte fysikaliskt korrekta. De är däremot fortfarande användbara i många sammanhang, så vi vill i alla fall nämna dem här.

2.5 Renderingsinställningar

Nu ska vi studera vad som händer när man ändrar kvalitetsinställningarna för renderingen. Behåll renderingsfönstret öppet, och öppna *Render Setup* → *Arnold Renderer* → *Sampling and Ray Depth* (Se figur 2). Alla inställningarna har det gemensamt att bilden blir sämre när du minskar siffrorna och bättre när du ökar dem. Det tar också längre tid att rendera bilden ju högre kvalitet du vill ha. Nästan inget är gratis vid rendering – det mesta är en kompromiss mellan renderingstid och kvalitet.

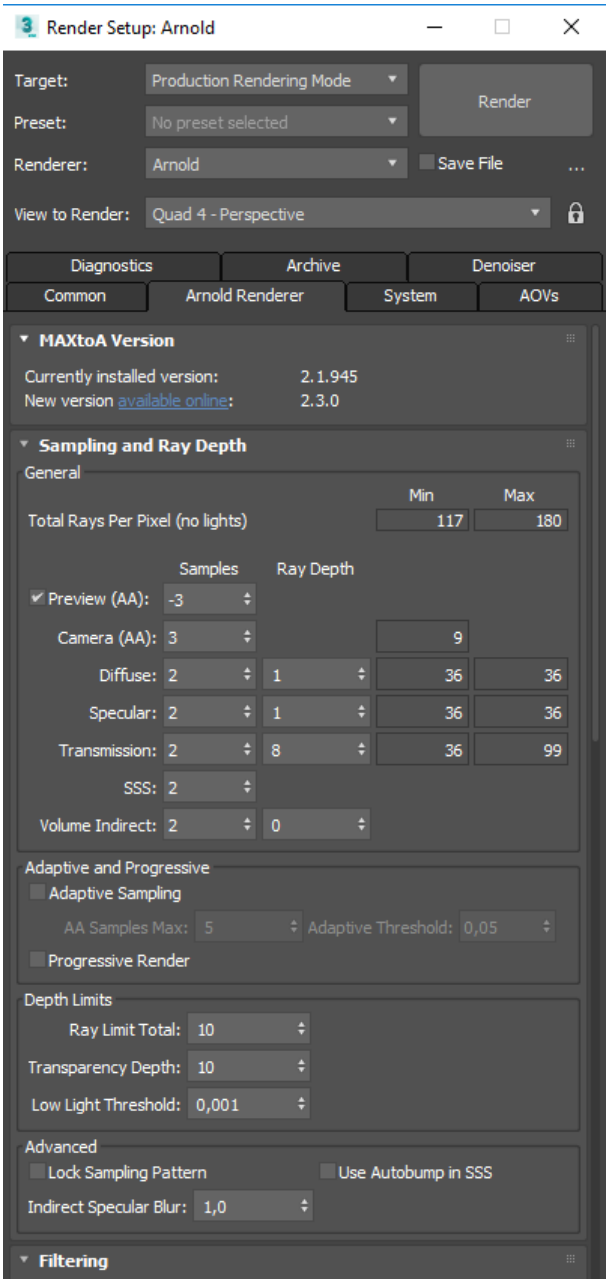

Figure 2: Inställningar för renderingskvalitet i Arnold

Vi ber dig att pilla med några av inställningarna nedan, men kom framför allt ihåg att de finns om du senare vill rendera mer komplicerade scener i Arnold.

Experimentera först med värdet på *Camera (AA)*. Notera hur det värdet påverkar antalet strålar för alla underavdelningar. Det här är den mest allmänna kvalitetsinställningen, men också den som påverkar renderingstiden mest. Med värdet 1 blir bilden brusig, när värdet ökar blir den successivt bättre men tar också längre tid att beräkna. Värden mindre än 1 betyder att man inte ens samplar varje pixel, utan sparar tid genom att minska upplösningen i bilden. Det kan vara praktiskt för snabba previews, men är annars inte att rekommendera. Kom också ihåg att scenen du har är väldigt enkel, och det tar generellt sett längre tid att rendera en mer komplicerad scen. Metoden som används här för att rendera pixels kallas *stokastisk sampling* (stochastic sampling). "Stokastisk" innebär att man samplar varje pixel inte bara med flera strålar, utan också på litet olika ställen för att bryta upp skarpa defekter och ersätta dem med brus, något som ögat oftast är mindre känsligt för.

För arealjuskällor finns det dessutom en separat inställning i ljuskällan för hur noga man ska sampla just den. Prova att ändra inställningen *Samples* under rollouten *Rendering* i Modify-panelen för din ljuskälla, och se vad som händer med bilden.

Om du vill se effekten av indirekt ljus riktigt tydligt, sätt en kraftig, ljus färg som till exempel knallgrön eller skriande orange på någon av dina boxar och se vad som händer med neutralt färgade ytor i närheten av den. Du bör se en åtminstone svagt färgad diffus reflexion från den närliggande färgade ytan, något som kallas för *color bleeding*. Eventuellt kan du behöva experimentera litet med ljuskällans riktning för att se effekten tydligt. Color bleeding syns tydligast i skuggor, där det bara är indirekt ljus som belyser ytan. Det här kan vara förvånansvärt viktigt för realismen i vissa bilder, trots att det är en ganska subtil effekt.

Hur noga man beräknar indirekt ljus bestäms av inställningen *Diffuse: Ray Depth*. Pröva att sätta den till 0. Nu kommer diffusa reflexioner mellan objekt inte att beräknas alls, och alla skuggor blir svarta. Resultatet är en orealistisk och ful bild, men den går snabbt att beräkna. När man stänger av indirekta diffusa reflexioner i Arnold så beter sig programmet i princip som en klassisk ray tracer snarare än en path tracer. I vissa fall kan det vara tillräckligt, men de flesta scener har en ganska stor andel indirekt diffust ljus som oftast behöver tas med i beräkningarna för ett gott resultat.

Kryssrutan *Adaptive Sampling* gör att renderaren försöker lägga extra tid på de delar av bilden som har mest brus, vilket kan vara väldigt användbart. Om du kryssar i *Progressive Render* så kommer bilden att renderas i flera pass med successivt ökande kvalitet i stället för att värdet i varje renderad pixel räknas färdigt direkt.

2.6 Skylight (Skydome)

Det är inte alla scener som har en stark riktad ljuskälla som dominerande belysning. Ofta har man ett stort inslag av ljus som kommer från många olika håll. Inomhus har man ofta ljusa tak och väggar som bidrar starkt till det indirekta ljuset i scenen, och det kan vara bättre att simulera det med en diffus ljuskälla än att lita på att beräkningen av indirekt ljus tar hand om allt. Utomhus har man också en hel del ljus från himlen, inte bara från solen. Om det är mulet ute så kommer ljuset mot en viss punkt från i stort sett alla riktningar där himlen är synlig från den punkten. De här fallen är svåra att hantera med vanliga ljuskällor, och de har därför en egen sorts ljuskälla, ofta kallad *Skylight* eller *Skydome*. Under Modify-fliken för din ljuskälla, ändra *Shape* → *Type:* till

Skydome. Ljuskällans position spelar ingen roll för resultatet i det här fallet – den har en ikon i viewporten för att man ska ha något att klicka på, men ljuset kommer från alla håll.

Rendera din scen med en *Skydome* som enda ljuskälla. Ljuset kommer nu från en väldigt stor yta, så du behöver minska både *Intensity* och förmodligen även *Exposure* för ljuskällan för att bilden inte ska bli helt överexponerad. Se hur mjukt ljuset blir. Du kommer eventuellt också att kunna se en del brus som kan vara störande. Ofta används en Skydome i kombination med en riktad ljuskälla som är starkare, och bruset uppstår då bara i skuggorna, där det sällan syns eftersom de delarna av bilden är mörkare. Om man däremot är intresserad av riktigt noggrann beräkning av en *Skydome*, till exempel för att den är den huvudsakliga ljuskällan i scenen, kan man behöva öka antalet sampel för ljuskällan.

2.7 Exponeringskontroll

Om scenen blir för mörk (eventuellt helt svart) eller för ljus (eventuellt helt vit) medan du experimenterar med olika ljuskällor kan du behöva ändra exponeringsinställningen i renderaren. I menyn *Rendering* → *Exposure Control...* kan du välja hur renderaren ska tolka ljusvärden (som kan variera från 0 till godtyckligt stora tal, beroende på hur många och hur starka ljuskällor du har i scenen) och översätta dem till pixelvärden för utbilden (som behöver ligga mellan 0 och 1 för att visas korrekt på skärmen). Alternativet *Physical Camera Exposure Control* är ett vettigt alternativ som försöker efterlikna hur en kamera skulle bete sig. Du ställer in parametern *EV* (förkortning för *Exposure Value*) för att variera den virtuella exponeringen. Notera att du kan testrendera din scen i låg upplösning i dialogrutan för *Exposure Control*, så att du ser om du träffar rätt när du ställer in exponeringen. Experimentera gärna!

2.8 Daylight System

Utomhus har man ofta en kombination av direkt solljus och diffust ljus från himlen. För att göra det enklare att skapa realistisk ljussättning i utomusscener finns det ett färdigt *Daylight System* i 3dsMax. Tyvärr fungerar inte den sortens ljuskälla med Arnold. Den är gjord för att användas med renderaren ART, som vi inte berör i den här laborationen, men vi vill ändå nämna att den finns. Du hittar *Daylight* i menyn under *Create* → *Systems* → *Daylight System*, eller under *System*-fliken i *Create*-panelen. Det kan vara smidigt att använda om du vill rendera en utomhusscen, eftersom du ställer in konkreta parametrar som plats och tidpunkt snarare än i vilken vinkel solen ska lysa. Det är framför allt väldigt användbart inom byggvisualisering för att visa hur ljus skulle falla i en planerad verklig miljö eller inuti ett hus som inte byggts än. Då vill man nämligen testa hur det ser ut vid olika tidpunkter på dagen och under olika årstider. Vi går inte in på detaljer om hur *Daylight System* fungerar. Det är i princip bara en kombination av en riktad ljuskälla och en *Skylight*, men man ställer in både riktning, intensitet och färg på ljuset genom att ange en geografisk plats på jorden samt datum och tid på dygnet. Detaljer om detta finns i hjälpsystemet, och även i en separat tutorial om du vill experimentera själv.

3 Reflexion och brytning

Renderaren *Arnold* är en så kallad path tracer, en vidareutveckling av ray tracing. Ray tracing och path tracing är mycket lämpliga renderingsmetoder för att rendera blanka och genomskinliga objekt, och eftersom sådana objekt kan vara väldigt viktiga och centrala i många scener ska vi i resten av den här laborationen titta litet närmare på just speglingar och brytningar.

3.1 Testscen

Gör *Reset* på 3dsMax om du inte redan gjort det, för att ta bort eventuella kvarvarande inställningar från föregående uppgift. Ställa in *Render Setup* så att du använder *Scanline Renderer*. Det finns nämligen en gammal och mer begränsad ray tracing-modul i den renderaren som är bra att använda för att förklara principen innan vi så småningom byter tillbaka till Arnold.

För att visa reflexioner och brytningar behöver vi en testscen som kan reflekteras och brytas i objekten vi skapar. I stället för att göra en komplicerad scen med en massa objekt kan vi fuska och göra ett schackrutigt golv. Det är en klassisk bakgrund för att tydligt visa just reflexioner och brytningar, även om det naturligtvis inte är så realistiskt.

Skapa ett *Plane* med storlek 100 × 100 enheter och ganska många segment i båda ledderna så att det går att böja snyggt. Lägg på en *Bend* i X-led med *Angle* satt till -90, samt *Limit Effect* aktiverat med *Upper Limit* 60. Lägg på en *Bend* till i Y-led med *Angle* -75. Klicka ur rutan *Real-World Map Size* i ditt *Plane*, och lägg ett material på objektet som har en diffus map av typen *Checker*. Klicka ur rutan *Use Real World Scale* (om den är ikryssad) och sätt *Tile* till 12 i båda ledderna. Resultatet bör bli något i stil med figur 3.

För ett tydligare visuellt resultat i följande uppgifter, byt färg på rutorna från svart och vitt till något som har litet mindre extrem kontrast, till exempel mellangrått och ljusgrått. På helt svarta ytor syns inte skuggor, och för helt vita ytor är det svårt att ställa in exponeringen rätt. Det blir lätt ”vitare än vitt” utan att man märker det.

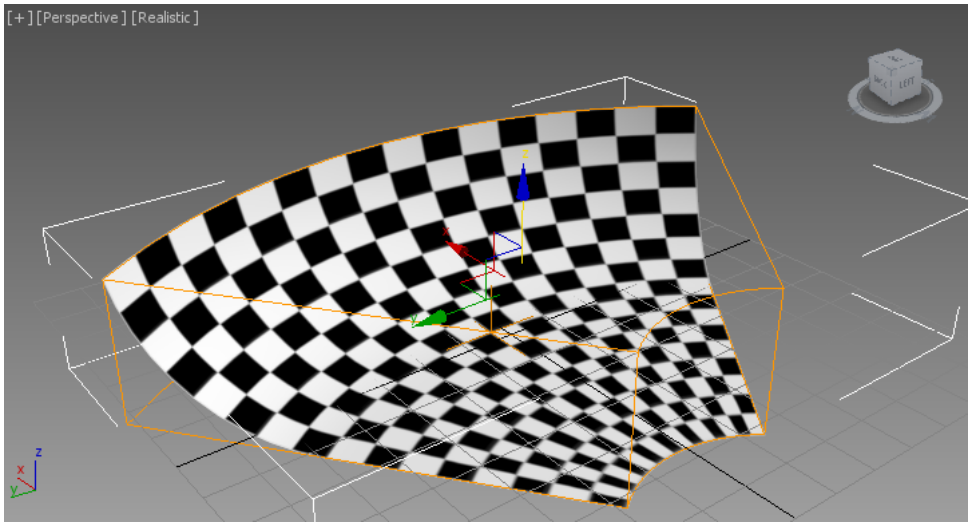

Figure 3: Bakgrund för experiment med raytracing

3.2 Ett vinglas

Nu ska vi göra ett objekt som har både reflexion och transparens. Vi kunde välja en enkel sfär. Många klassiska vetenskapliga artiklar om raytracing har illustrerats i stort sett uteslutande med bilder på sfärer i glas och krom på schackrutiga golv, men för att göra det litet intressantare ska vi göra ett vinglas med hjälp av modifiern *Lathe*. Ordet "lathe" är engelska för "svarv", och modifiern används för att skapa rotationssymmetriska objekt. Rita först en *Line* så att den ser ut åtminstone ungefär som figur 4. Rita i Top-vyn så att du jobbar i xy-planet. Börja med att placera ut hörnpunkterna. Högerklicka när du satt ut den sista punkten. Linjera sedan upp de två ändpunkterna så att de ligger på samma x-koordinat. Det kan du göra på flera sätt, men först och främst måste du komma åt att flytta på enstaka hörnpunkter. I *Modify*-panelen för din *Line*, under rollouten *Selection*, välj *Vertex* (ikonen med fyra röda prickar) för att manipulera enstaka punkter längs linjen. Markera sedan en av ändpunkterna, välj *Align* (alt-A) och klicka på linjen självt som *Target*-objekt. Linjera i X-led mot *Maximum* (förutsatt att du ritat med samma koordinatriktningar som i figur 4. Upprepa för den andra hörnpunkten. Nu ligger båda på exakt samma X-koordinat. Det är viktigt för att få ett snyggt resultat i nästa steg. Verktyget *Align* är tyvärr litet krångligt att använda för enstaka punkter på det här sättet. Om du vill kan du i stället klicka på hörnpunkterna du vill flytta och knappa in deras exakta koordinater direkt i textrutorna längst ner i programfönstret. Huvudsaken är att de två ändpunkterna på din *Line* har *exakt* samma x-koordinat innan du går vidare.

För att göra linjen mjukt krökt, högerklicka på hörnpunkterna en efter en så att du får upp en meny. Ändra punkternas typ till *Béziér*. För punkten på ovansidan av foten där du ska ha ett skarpt hörn väljer du i stället *Béziér Corner*. Pilla med handtagen på tangenterna en stund och flytta hörnpunkterna tills du är nöjd med resultatet. Det lönar sig att vara noggrann här, men du kan naturligtvis gå tillbaka och ändra senare om du inte blir nöjd. Tänk på att ett verkligt vinglas är väldigt tunt, bara någon millimeter, så gör helst inte väggen för tjock. Då kommer ditt glas att se ut som en tjock kristallvas snarare än ett tunt vinglas.

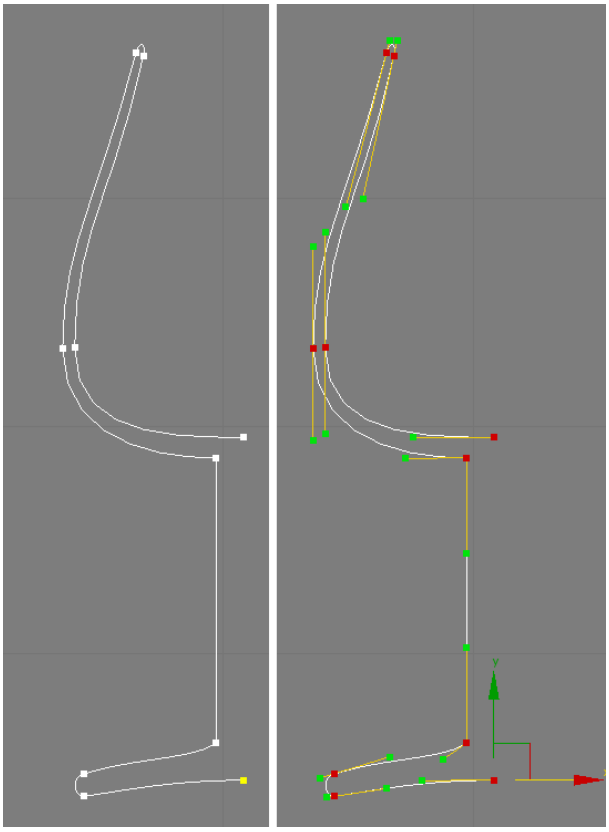

Figure 4: Profilen för ett vinglas

Lägg sedan modifiern *Lathe* på din *Line*. För *Direction*, välj *Y*, och för *Align*, välj *Max*. Välj *Mesh* under *Output* för att skapa en vanlig polygonmodell. Resultatet bör bli något i stil med vänstra halvan av figur 5. Om det ser fult ut, försök gärna att fixa till proportionerna, men det estetiska är inte huvudsaken här. Om det ser ut som till höger i figuren har du råkat få normalerna åt fel håll. Det beror på åt vilket håll du ritade din *Line* tidigare, och det är svårt att från början veta vad som är ”rätt” håll. Fixa problemet, om det uppstår, genom att kryssa i rutan *Flip Normals* i *Lathe*. Felvända normaler är ett vanligt problem medan man modellerar, speciellt när man skapar olika slags svepta ytor. Kryssa även i rutan *Weld Core*, så blir glaset snyggare precis kring rotationsaxeln.

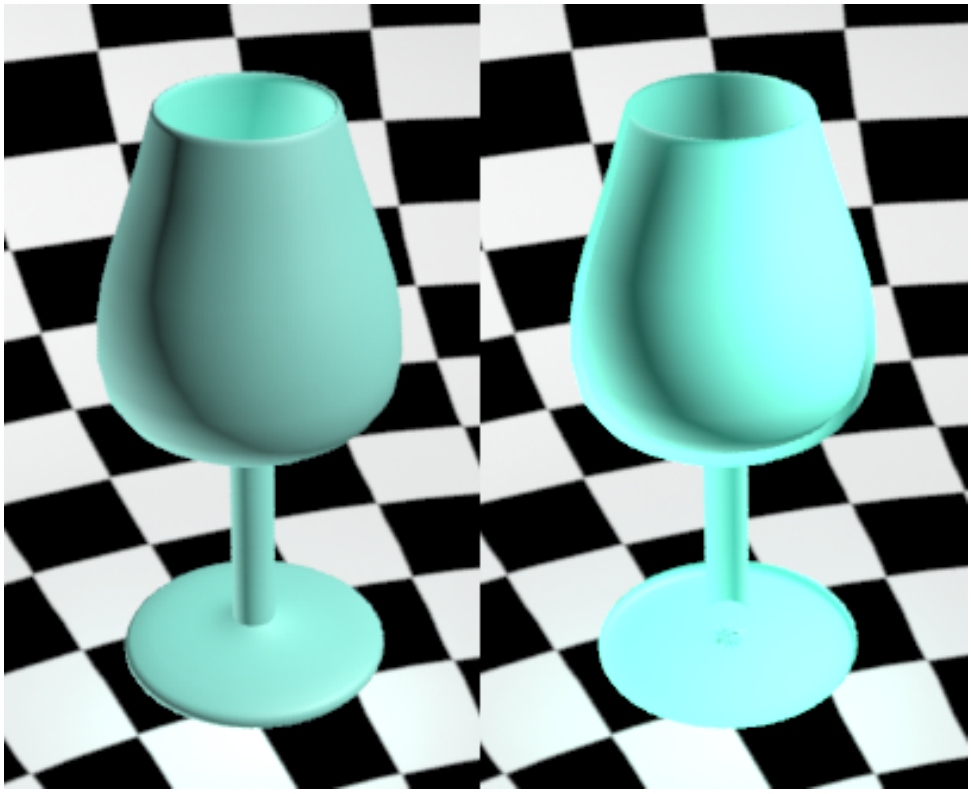

Figure 5: Det färdiga vinglaset, rättvänt (till vänster) och felvänt (till höger)

3.3 Ett glasmaterial

Nu ska vi göra ett glasmaterial till vinglaset, och vi ska först göra det med ett *Standard*-material för att förstå vad det är som behövs.

Lägg ett material på glaset. Sätt typen till *Standard*. Välj en diffus färg som är helt svart. Blankt och rent glas har i princip ingen diffus reflexion alls. Sätt däremot *Specular*-färgen till vitt. Den färgen påverkar av någon outgrundlig anledning även reflexioner som görs med en *Reflection map*.

Glas är genomskinligt, men det reflekterar också en del i ytan. Vi börjar med att göra reflexionen, inte för att det är den viktigaste egenskapen hos glas, utan för att den är enklast att göra och enklast att förstå sig på. Under *Maps* i materialet, välj *Reflection* och sätt typen till *Raytrace*. Rendera. Ditt glas ser ut att vara gjort av blankpolerad metall, men det är i alla fall en korrekt reflexion.

Glas är som sagt genomskinligt. Man kan göra enkel genomskinlighet via *Opacity* för materialet. Testa det. Det blir inte snyggt. Glas har som framträdande egenskap att det bryter ljus som passerar igenom det, och *Opacity* fixar inte att simulera den effekten. Sätt tillbaka *Opacity* till 100 och välj i stället under *Maps* en *Refraction* av typen *Raytrace*.

Stäng dessutom av *Reflection* genom att bocka ur kryssrutan. Vi kommer strax till varför. Rendera. Nu tar det längre tid att beräkna bilden, men på en modern dator bör du ändå ha rimligt korta väntetider.

Om du tittar noga på ditt renderade glas ser du att det faktiskt sker både en reflexion i ytan och en brytning genom ytan, trots att du stängt av reflexionsmappen. Anledningen till det ligger i inställningarna i rollouten för din *Raytrace*. Längst ner under *Refractive*

Material Extensions finns en kryssruta *Treat Refractions as Glass (Fresnel Effect)*. Bocka ur den och rendera igen. Nu bör det gå fortare att rendera, men resultatet ser inte längre riktigt ut som glas. Det beror på att reflexionerna är helt borta. Du kan få tillbaka dem med en separat Reflection map, men det finns ett problem med det: glas reflekterar olika mycket i olika vinklar, och hur mycket som bryts beror på hur mycket som reflekteras. (Det är detta som kallas Fresnel-effekt.) Reflexion och transmission i ett material måste tillsammans vara mindre än 1 för att energi inte ska skapas ur intet, och det är litet trixigt att få en *refraction map* och en *reflection map* att variera sina intensiteter tillsammans på det sätt som behövs. Det är inte omöjligt, men det blir en del pill som involverar en mapp som heter Falloff. Därför finns den här praktiska kryssrutan som gör det enklare att skapa just glasliknande material (vilket inkluderar även klara vätskor, till exempel vatten).

3.4 Renderingsinställningar

I renderingsinställningarna under Render Setup → Raytracer finns en inställning som heter *Maximum Depth*, som anger hur många reflexioner och brytningar strålarna ska tillåtas göra innan man slutar räkna. Värdet 0 betyder att man räknar endast på den första reflexionen och helt struntar i transmission. Minska värdet från standardinställningen 9. Vid för låga värden slutar glaset att vara genomskinligt, och redan vid värden strax under 9 ser du skillnader i punkter där ljuset studsar i interna reflexioner inuti glaset. Testa att sätta *Color to use at Max Depth* till något iögonenfallande, exempelvis rött, så ser du tydligt i vilka pixels som strålföljningen avbröts för att den nådde maximalt djup innan den träffade på en diffus yta. Om du minskar Max Refractions till 0 så kommer transmissionen att försvinna helt och glaset blir en blank svart yta.

3.5 Bättre glas med Arnold

Om du nu sätter in en ljuskälla i scenen och slår på skuggor så ser du ett allvarligt fel med vår bild: Glaset kastar en kompakt skugga. Det går att ange att skuggan som kastas inte ska vara helt svart, men det är inte så som glas beter sig. Ljus bryts genom glaset och kan landa delvis i den region som skulle ha varit i skugga om glaset hade varit genomskinligt, men omfördelat på grund av ljusets brytning. På liknande sätt kan ett speglande objekt med en krökt yta kasta kraftigt deformerade reflexer som lyser upp dess omgivning. Sådana effekter kallas *caustics* på engelska, och de är tyvärr krångliga att rendera även med raytracing, eftersom ljus först går från ljuskällan, studsar i ett speglande eller transparent objekt och sedan landar på en diffus yta. Raytracing i sin grundläggande form (det som kallas *invers raytracing* eller *Whitted raytracing* efter upphovsmannen Turner Whitted) klarar bara av det omvända fallet: ljus landar på en diffus yta, och den diffusa ytan betraktas sedan genom en eller flera reflexioner och brytningar.

För att rendera skarpa och detaljerade *caustics* i en ray tracer behöver vi göra ett extra pass av förberäkningar med en metod som kallas *Photon mapping*. Tyvärr går det inte att göra detta med Arnold, eftersom det är en metod som gör alla möjliga sorters fusk och därför inte passar in i Arnolds ”renderingsfilosofi”, om man får kalla det så. (Programmerarna bakom Arnold anser att i alla fall att det är en viktig principfråga, om än inte en filosofi.) En path tracer kan visserligen i teorin simulera caustics på ett korrekt

sätt, men det kräver enorma mängder sampel för att räkna ut skarpa caustics utan en massa brus. Vi håller oss därför till att belysa vinglaset med en rimligt stor arealjuskälla i övningarna med Arnold nedan. Caustics-effekterna blir då suddigare och inte alls lika känsliga för brus, och man kan komma undan utan att beräkna dem särskilt noggrant.

Byt renderare till *Arnold*. Skapa ett nytt material av typen *Standard Surface*. Detta material är liksom *Lambert* specialgjort för Arnold, men det kan ha såväl diffus och speglande reflexion som transparens och emission, och är därför ett mycket generellt och flexibelt material. De allra flesta ytegenskaper som finns i verkligheten kan faktiskt beskrivas med parametrarna i en *Standard Surface*.

Byt material på ditt vinglas till *Standard Surface*. Grundinställningarna i materialet är en matt yta, men du kan enkelt ändra det till ett glasliknande material bara genom att minska den diffusa reflexionen (*Base*) till 0, och i stället öka *Transmission* till ett värde något mindre än 1. Styrkan på dem speglande reflexionen sätter du under *Specular*. För en glasyta är denna reflexion ganska liten, bara några procent, men du kan ändå låta Specular vara 1.0. Anrold är en fysikaliskt korrekt renderare som inte tillåter att ljus uppstår ur ingenting, och inställningarna i det här materialet ger prioritet till transparensen. Reflexionen blir alltså det som är kvar efter att transmissionen tagit sin del av den simulerade strålen. Sätt däremot gärna *Roughness* till 0 för att få blankt glas i stället för en litet lätt frostad yta. Övriga inställningar kan du lämna orörda.

Nu behöver vi bara en ljuskälla. Skapa en sådan av typen *Arnold Light* i din scen, placera den så att den lyser litet snett uppifrån mot ditt objekt, och sätt dess *Shape* till att vara en *Quad* med en lagom storlek för att ge ett mjukt ljus som ändå kastar tydliga skuggor. Du behöver också öka intensiteten på din enda ljuskälla, eller ställa in *Exposure Control*, för att din bild inte ska bli för mörk.

Rendera. Notera att det tar en stund, och att bilden blir något grynig. Båda delarna är en oundviklig bieffekt av att scenen nu innehåller ett transparent och brytande material. Det är de pixels där man ser glaset som tar längst tid att beräkna. För att bilden ska bli bra behöver du förmodligen öka antalet sampel något under rubriken *Transmission* i *Render Setup* (Figur 2). Arnold kan hantera de flesta sorters reflexion, transmission och ljuskällor rakt av, utan krav på särbehandling, men i många fall behöver man öka antalet sampel i någon eller några kategorier för att få en tillräckligt bra bild. Tyvärr finns det inget sätt att helt bli av med bruset, men det går att minska det tills det inte längre är märkbart, eller åtminstone inte störande.

3.6 Caustics

Det finns fortfarande en brist i bilden: den saknar *caustics*. Bristen blir tydligast om du minskar storleken på ljuskällan så att skuggan blir skarp. I materialet *Standard Surface* finns en kryssruta *Enable caustics* under avsnittet *Advanced*, men den gör tyvärr ingen större skillnad i vårt fall. Subtila, reflektiva caustics går hyfsat bra att rendera med Arnold, men skarpa och intensiva transmissiva caustics av det slag som vi skulle behöva här är helt enkelt inte möjliga att beräkna med programmet. (Det går strängt taget att få till det genom att använda ett emissivt objekt som ljuskälla i stället för en Arnold Light, men det blir helt ohanterligt beräkningstungt.) Photon mapping, som används av många andra renderare, är en ”fuskmetod” som utvecklarna av Arnold har valt att inte blanda in i sin fysikaliskt korrekta algoritm. Det finns flera goda skäl till det, men

effekten blir att Arnold egentligen inte är en särskilt lämplig renderare för att rendera scener med framträdande caustics. Traditionell ray tracing med diverse fusk inslängt i efterhand har fortfarande sina fördelar.

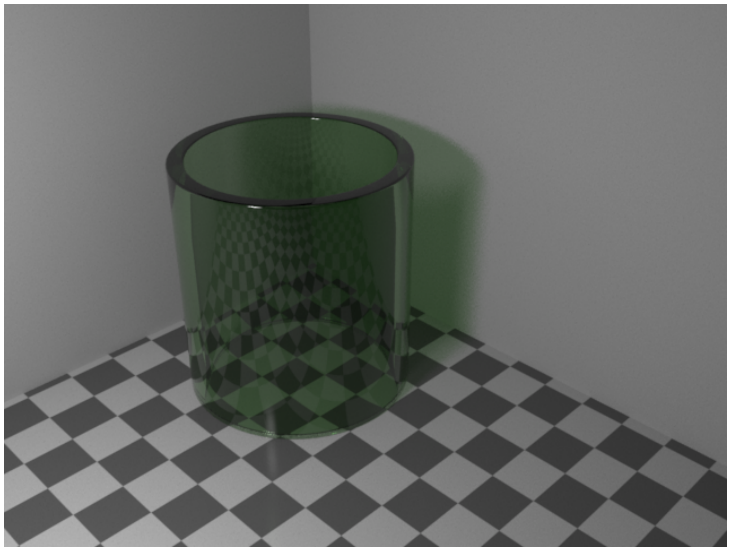

Figure 6: Ett enkelt glasobjekt renderat med Arnold

4 Uppgifter att redovisa

- Studera avsnittet om ljus och skuggor, och rendera en bild av din testscen med diffust reflekterande objekt så att den får mjuka skuggor från en arealjuskälla. Rendera också med en Skydome, gärna i kombination med en riktad ljuskälla för att efterlikna dagsljus utomhus eller en inomhusscen i ett rum med ljusa väggar.

- Rendera bilder som du är nöjd med av glaset du skapade, dels med ditt egna Standard-material i Scanline Renderer, dels med Arnold. Bilden ska vara rimligt realistisk både vad gäller reflexion och brytning, men den ska inte behöva ta en timme att rendera. Använd en enda riktad ljuskälla för belysningen i båda fallen.

Bilderna redovisar du under laborationen till labassistenterna på labtillfället. Lycka till, och fråga gärna om du vill veta mer, men glöm inte heller bort att du kan läsa mer på nätet och i programmets omfattande hjälpfiler och tutorials. Och inte minst: var inte rädd för att experimentera på egen hand!