

# Bakningslösningar för nästa generationens texturer

David Rothelius

Luleå tekniska universitet

C- uppsats  
Datorgrafik

Institutionen för LTU Skellefteå  
Avdelningen för Fritid och Underhållning

## **Förord**

Det här examensarbetet utfördes av David Rothelius under 10 veckor på det Uppsalabaserade företaget Starbreeze. Examensarbetet är en avslutning på en treårig datorgrafikutbildning på Luleå Tekniska Universitet.

Under min tid på utbildningen i Skellefteå har jag vid ett flertal tillfällen stött på onödigt komplexa tekniska verktyg och extremt höga inlärningskurvor för dessa. Ofta har problemen visat sig så stora att mycket tid har lagts åt att få teknikerna att fungera smidigt i en produktion.

När jag kom till Starbreeze märkte jag redan efter någon vecka att arbetsflödet och bakning av texturer inte kändes helt optimalt. Visserligen berodde det till viss del på ovana, men jag upplevde att utnyttjandet av den nya tekniken kunde skötas bättre. Jag bestämde mig för att göra texturbakning till mitt examensarbete.

Jag skulle vilja tacka min handledare Nicholas Sirén som hjälpte mig igång med Starbreeze egna verktyg och som kontinuerligt har gett mig bra feedback på mitt grafiska arbete.

Slutligen bör det också nämnas att jag under dessa veckor har producerat material som faller under ett så kallat Non Disclosure Agreement, vilket betyder att materialet är hemligstämplat. Detta material kan således inte visas upp för allmänheten och därför blir också det visuella resultatet och den upplevda arbetsmängden i rapporten något lidande.

## ***Sammanfattning***

Nya tekniker och bättre hårdvara ger spelutvecklare möjlighet att skapa titlar med högre realism än någonsin förut. Detaljrikedomen och den generella kvaliteten förväntas öka varje år samtidigt som utvecklingstiden ligger still eller ibland till och med blir kortare. Detta ställer så klart oerhörda krav inte bara på de individuella utvecklarna, utan även på verktygen och arbetsflödet i en produktion.

Den här rapporten tar upp ett ämnesområde som i princip alla stora spelutvecklare rör sig i och där det finns många förbättringar kvar att göra för att förenkla och öka produktionshastigheten.

Texturbakning är olika tekniker för att spara undan information som är för tung för att behandla i realtid i bilder som sedan laddas in i spelet. Rapportens fokus ligger på normalmappning och ambient occlusion och vad man kan göra för att förbättra arbetsflödet kring dessa tekniker.

## ***Abstract***

New techniques and better hardware gives developers possibilities to create games with unprecedented visual quality and realism. The level of detail and the overall quality is expected to rise each year while development time is more or less the same as before. This of course puts extreme demand on not only the developers making the game but also the tools and the pipeline in the production.

This thesis project deals with a subject that virtually every major game developer utilizes, but where there's also a lot of room for improvement and speed increases.

Texture baking are different techniques for storing information too heavy for realtime usage in images. The focus will be on normal mapping and ambient occlusion, and what tools you can use to speed up the usage of these techniques.

The thesis project is carried out at Starbreeze Studios in Uppsala and my name is David Rothelius.

# Innehållsförteckning

<b><i>Inledning</i></b>	<b>1</b>
<b>Ämnesval</b>	<b>1</b>
<b>Syfte</b>	<b>1</b>
<b>Termer och förtydliganden</b>	<b>1</b>
Renderare	1
Mental Ray	1
Normalmappning	2
Ambient Occlusion	3
Starbreeze Studios	3
Ogier	4
xNormal	4
Illuminatelabs Turtle	5
<b>Fallstudier</b>	<b>6</b>
Metallsköld	6
Ansikte	6
<b><i>Arbetsflöde</i></b>	<b>7</b>
<b>Ogier / Mental Ray</b>	<b>7</b>
<b>xNormal</b>	<b>9</b>
<b>Turtle</b>	<b>11</b>
<b><i>Prestanda och resultat</i></b>	<b>12</b>
<b>Testmetod</b>	<b>12</b>
<b>Tidsprestanda</b>	<b>13</b>
Metallsköld	13
Ansikte	13
<b>Visuella resultat</b>	<b>14</b>
Metallsköld	14
Ansikte	14
<b><i>Diskussion</i></b>	<b>15</b>
<b>Arbetsflöde</b>	<b>15</b>
Ogier	15
xNormal	16
Turtle	17
<b>Resultat</b>	<b>18</b>
Prestanda	18
Visuellt resultat	18
<b>Slutsats</b>	<b>19</b>
<b>Fortsatt arbete</b>	<b>19</b>
<b>Referenser</b>	<b>20</b>

# Inledning

## Ämnesval

Allt eftersom komplexiteten i spelproduktioner ökar höjs också kraven på verktygen som spelutvecklare använder. Effektivt arbete och optimerat arbetsflöde är av extrem vikt för en lyckad spelproduktion. Metoder som användes för fem år sen håller inte måttet längre och nya lösningar måste tas fram för att skapa mer innehåll på mindre tid.

Tredjepartslösningar är en växande marknad och många spelutvecklare väljer att koppla in externt utvecklad programvara för att kapa både utvecklings och produktionstid. Frågan är om det alltid är en bra idé?

## Syfte

Detta examensarbete har som syfte att med hjälp av olika empiriska metoder jämföra lösningar för att generera bakade texturer för nästa generations spel. Faktorer som kommer att vägas in är användarvänlighet, tidsåtgång och visuellt resultat.

Eftersom examensarbetet utförs på spelföretaget Starbreeze så kommer deras interna verktyg Ogier att vara en central del i jämförelsen. Utöver Ogier så kommer även programvarorna xNormal och Turtle att användas.

## Termer och förtydliganden

### Renderare

Rendering är processen som gör om informationen från en tredimensionell modell till en bild. En renderare är helt enkelt programkod som kör denna process. <sup>[1]</sup>

### Mental Ray

Mental Ray är en fristående renderare designad för att så smidigt som möjligt kunna integreras i olika grafiklösningar. För närvarande stöds Mental Ray av Autodesk Maya, Autodesk 3ds Max, Softimage|XSI och ett flertal andra stora grafikprogram. <sup>[2]</sup>

Arbetet på Mental Ray påbörjades 1986 av det tyska företaget Mental Images. Företaget köptes tillsammans med Mental Ray upp år 2007 av grafikortstillverkaren NVIDIA. <sup>[3]</sup>

Mental Ray har använts på ett flertal större film och spelproduktioner och är industristandard främst inom simulering och visualisering.

## Normalmappning

Normalmappning är en teknik som möjliggör hög detaljnivå på enklare modeller. Detta sker genom att man förflyttar geometrisk information från en högupplöst till en lågupplöst modell med ungefär samma form. Till detta används en så kallad normalmapp.



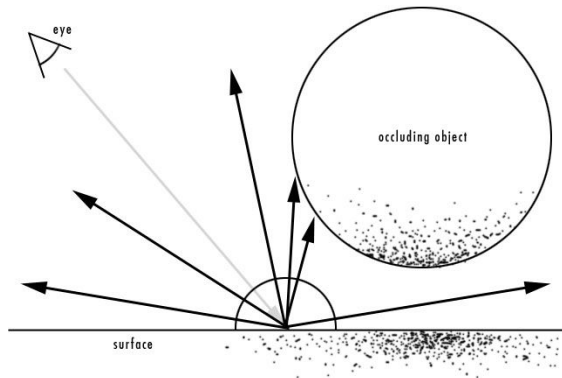
Rent tekniskt så är normalmappen en bild som beskriver hur ljus ska bete sig när det träffar en tredimensionell yta. Detta sker genom att informationen om hur ytorna ska peka i ljusberäkningarna lagras i de tre färgkanalerna på en textur. Normalmappar får därför ett regnbågsutseende som drar mot ljusblå. Genom att låta en enklare modell jämföra sin yta mot en mer avancerad kan man ta till vara på detaljerna i den högupplösta ytan utan att faktiskt lägga till mer geometrisk detalj. Detta sparar mycket prestanda jämfört med att använda den högupplösta modellen från början.

Användandet av normalmappar i spelindustrin slog igenom på bred front i och med lanseringen av ID Softwares storsäljare Doom 3. Tekniken sågs som revolutionerande och används nu för tiden i stort sett i varenda titel på marknaden. Detta gäller så klart även på Starbreeze som använder tekniken för i stort sett allt grafiskt material i sina spel.

## Ambient Occlusion

Ambient Occlusion är en teknik som simulerar de skuggor som uppstår när geometriska objekt ligger emot varandra. Tekniken är ännu inte användbar i realtid och måste därför bakas in i texturer.

Genom att mäta avståndet för varje yta gentemot de andra ytorna på en modell så skapar ambient occlusion en sorts global illuminering. Ju närmare varandra två ytor befinner sig desto mörkare kommer de att bli i uträkningen. <sup>[5]</sup>



Även om detta inte är en korrekt simulering av faktiskt ljus så anses den fungera tillräckligt bra för att användas så väl i spelindustrin som i hollywoodproduktioner, detta främst på grund av dess effektivitet.

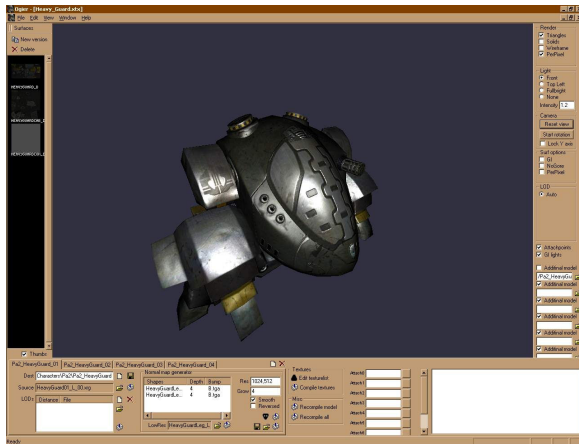
Genom att kombinera tekniken med normalmappning kan man ta till vara på och framhäva ännu fler detaljer från en högupplöst modell till en lågupplöst. Användandet av Ambient Occlusion är också praxis på Starbreeze.

## Starbreeze Studios

Starbreeze är ett Uppsalabaserat spelföretag som grundades 1998 av medlemmar ur demogruppen Triton. Företaget har sedan dess utvecklat ett antal titlar både till PC och till olika spelkonsoller. Chronicles of Riddick och The Darkness är företagets mest kända produktioner. <sup>[6]</sup>

Starbreeze har för närvarande ungefär 80 anställda som jobbar på två olika projekt. Mitt arbete kommer på grund av sekretesskäl inte att beröra dessa produktioner specifikt även om de tekniker och verktyg jag använder i viss mån är de samma som för dessa titlar.

## Ogier



Ogier är Starbreeze interna utvecklingsverktyg som används för nästan allt material som läggs in i de egna produktionerna. Ogier har varit i utveckling sedan företaget grundades och har använts i samtliga titlar som Starbreeze har släppt.

Det har grundläggande stöd för baneditering i Quake och är sen 2004 tillgängligt för allmänheten som vill skapa sitt eget material till Starbreeze titlar. [7]

Den här rapporten kommer att fokusera på Ogiers skapande av normalmappar, men verktyget kommer att vara slutpunkt oavsett vilket program som används för texturbakningen eftersom det representerar det visuella resultat som sedan hamnar i produktionen.

## xNormal



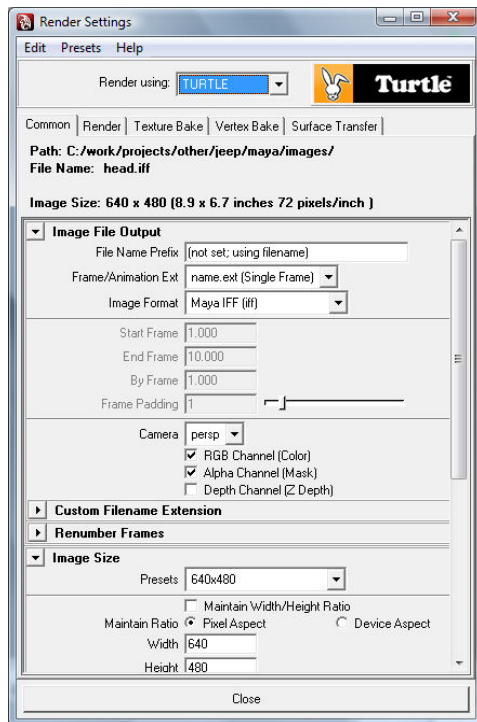
xNormal är en privatutvecklad programvara för rendering av exempelvis normalmappar och ambient occlusion. [8] Det är helt gratis även för kommersiellt bruk, något som förstås gör det intressant för spelföretag som Starbreeze.

Programmet kommer utrustat med ett visualiseringsläge för att visa resultatet av användarens texturbakningar men förlitar sig i övrigt på ett ickegrafiskt interface.



LTU Skellefteå

## Illuminatelabs Turtle



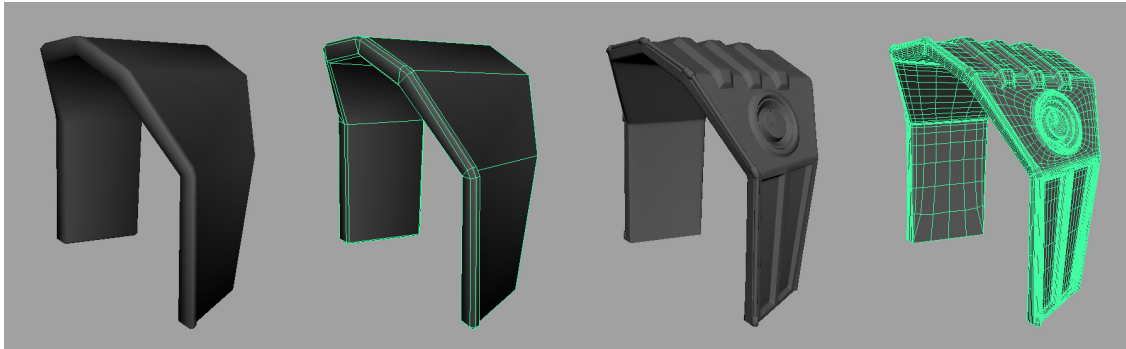
Turtle är ett renderingsverktyg utvecklat av Göteborgsbaserade Illuminate Labs. Programmet är framtaget med spelproduktion i åtanke och det är helt integrerat med Maya. Det finns möjlighet till så väl ljusbakning för spel som konventionell rendering.<sup>[9]</sup> Den här rapporten kommer dock enbart att fokusera på dess förmåga att baka ut normalmappar och ambient occlusion.

Illuminate Labs säljer utöver Turtle den plattformsoberoende bakningslösningen Beast som är helt anpassad för integration i existerande interna verktyg som exempelvis Ogier. Fullt stöd för exempelvis Unreal Engine 3.0 är redan implementerat.<sup>[10]</sup> Beast kommer inte att beröras i den här rapporten.

## Fallstudier

För att undvika eventuella NDA-överträdelser har följande exempelfall tagits fram helt oberoende av Starbreeze och deras projekt. De representerar två vanliga men ganska skilda fall för bakning av texturmappar i spelsyften.

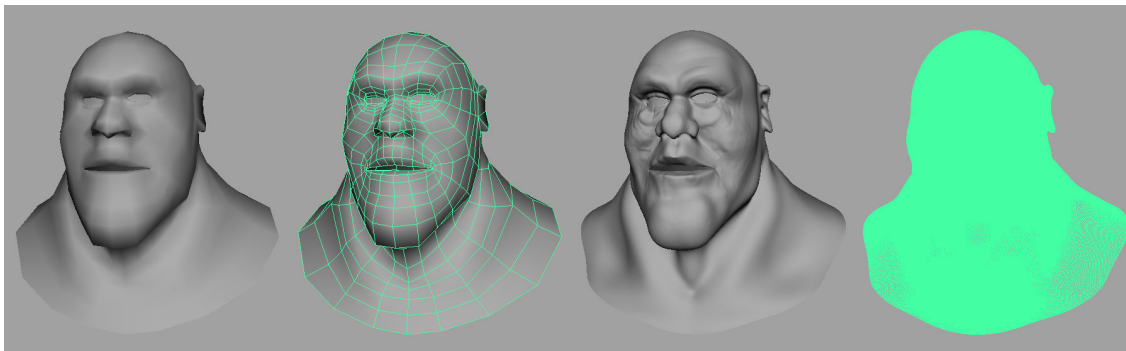
### Metallsköld



De två bilderna till vänster visar den lågdetaljerade modellen som ligger på 194 polygoner. Till höger är den högdetaljerade modellen som består av 30888 polygoner.

Objektet är förhållandevis simpelt, även i sin högdetaljerade form. Som ett enklare exempelfall och som komplement till det mer avancerade ansiktet fungerar det utmärkt.

### Ansikte



Till vänster visas den lågdetaljerade modellen som består av 930 polygoner. Den högdetaljerade modellen till höger består av 953 344 polygoner.

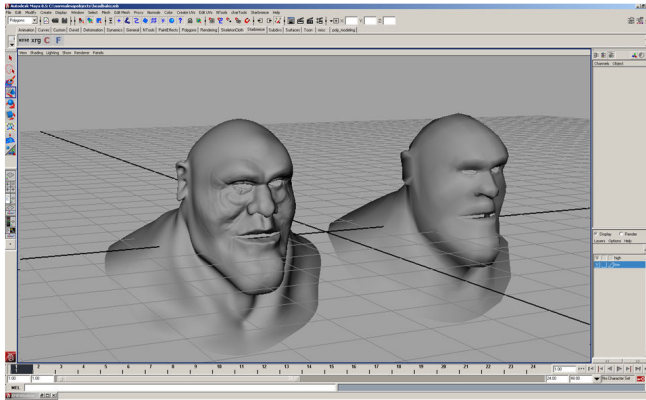
Digitalt återskapade ansikten kräver extrem detaljrikedom för att ge ett realistiskt intryck. Många speltillverkare, till exempel Starbreeze, väljer att viga lika mycket eller till och med mer resurser åt ansiktet än hela resten av kroppen för att fånga alla detaljer och skapa trovärdighet. I det här fallet är ansiktet av det lite mer karikerade slaget, men med hjälp av programvaran Zbrush har avancerade detaljer som rynkor och ärr lagts till för att öka realismen och detaljrikedomen.

LTU Skellefteå

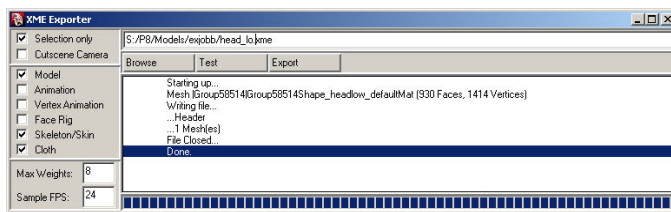
## Arbetsflöde

Arbetsflödet, eller snarare det engelska begreppet pipeline används flitigt inom spelindustrin. I den här delen kommer jag att visa arbetsprocessen för de tre olika bakningslösningarna.

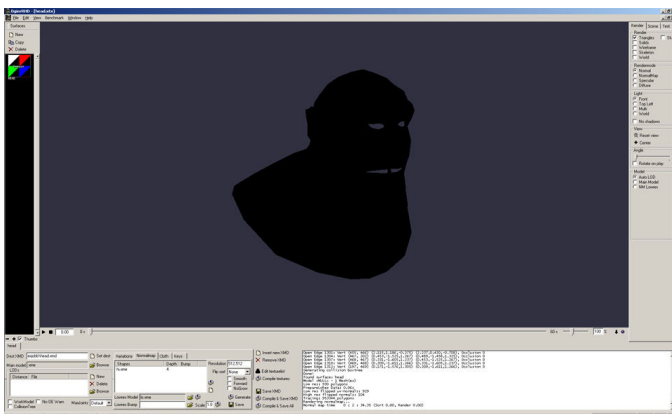
### Ogier / Mental Ray



Samtliga arbetsflöden med de olika programmen utgår ifrån samma maya-fil där den högdetaljerade och den lågdetaljerade filen ligger.

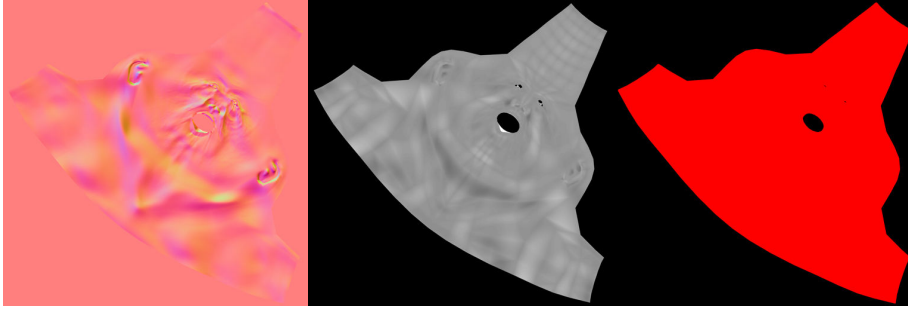


De båda modellerna exporteras via Starbreeze egna maya-exporterare. Filerna sparas i XME-formatet, vilket är ett format för modeller som är specifikt för Ogier.

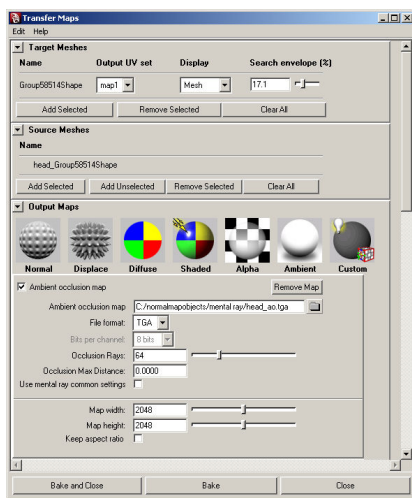


XME-filerna importeras in i Ogier där det också definieras vilken som är den högdetaljerade respektive lågdetaljerade modellen. Inställningar för bakning och förhandsvisning av de olika modellerna kan göras i det här skedet. När man är nöjd så trycker man på Generate.

## LTU Skellefteå

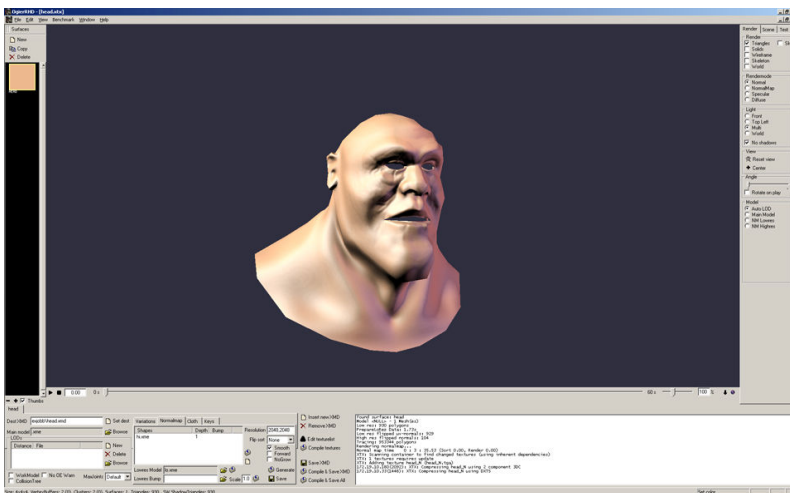


Ogier genererar inte bara en normalmapp (längst till vänster) utan även ett par andra texturer som kan användas som hjälpmedel i det fortsatta arbetet med texturen. Höjdmappen beskriver som namnet antyder höjdvärden och meshcolor färgar geometriska detaljer i olika färger. I fallet med ansiktet fanns det bara ett geometriskt objekt i form av ansiktet så därför blir meshcolorn också helt enfärgad.



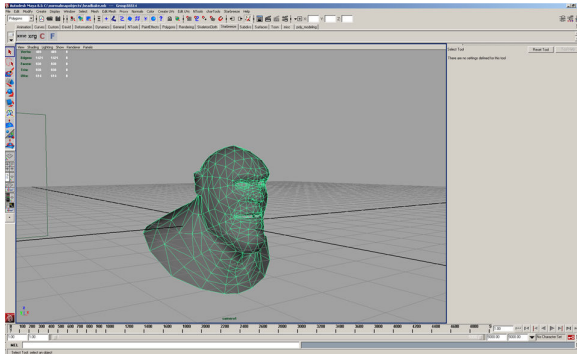
Eftersom Ogier inte har någon metod för att generera ambient occlusion så görs detta vanligtvis genom Maya och mer specifikt renderaren Mental Ray.

Det finns några olika metoder för att baka texturer med nyss nämnda renderare. I mitt fall använder jag Transfer Maps.

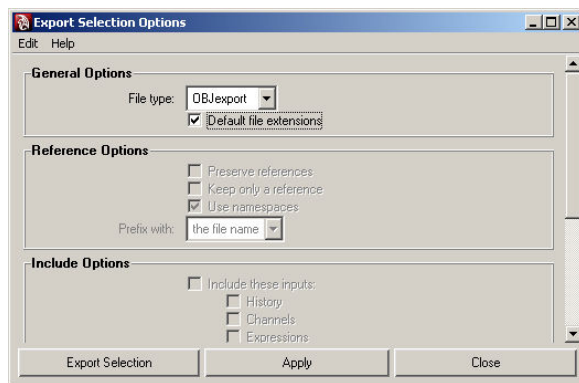


Den bakade occlusionmappen multipliceras på diffustexturen och kompletterar den bakade normalmappen i Ogier och ger det färdiga resultatet.

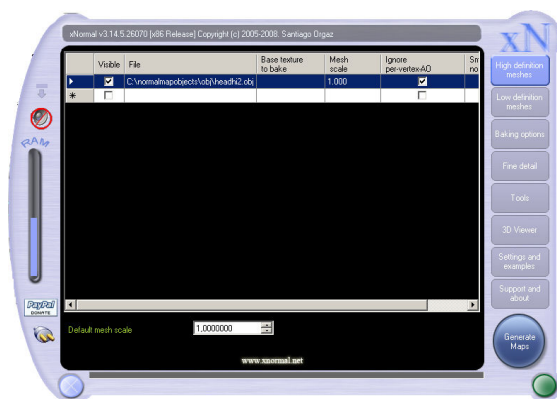
## xNormal



Precis som i det föregående arbetsflödet så utgår arbetet från Mayafilen med de två modellerna. Eftersom xNormal föredrar att modellerna enbart består av trianglar så måste modellerna trianguleras innan de skickas vidare.

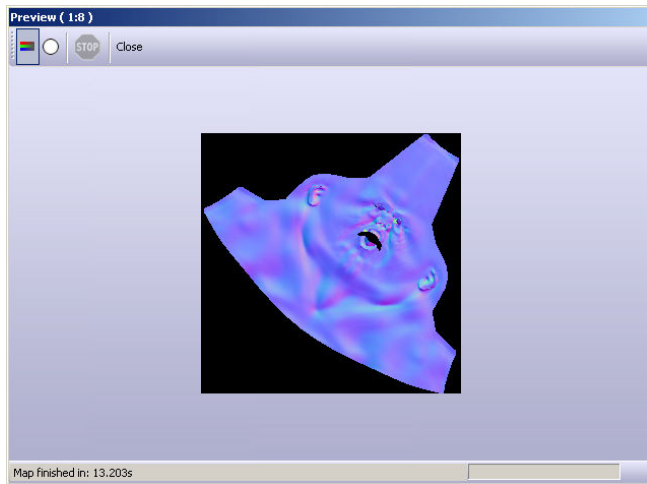


Modellerna exporteras till standardformatet OBJ som Maya kan exportera till direkt från installation.

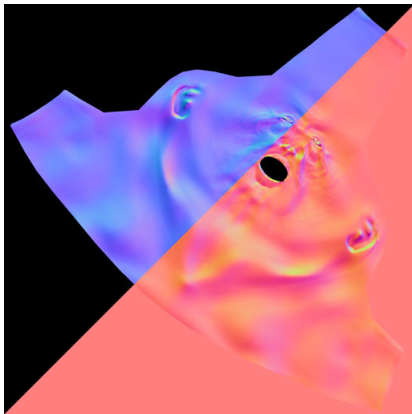


OBJ-filerna importeras i xNormals dialogbaserade interface. Typen av information som ska bakas och inställningar för dessa definieras. I det här fallet betyder det normalmapp och ambient occlusion.

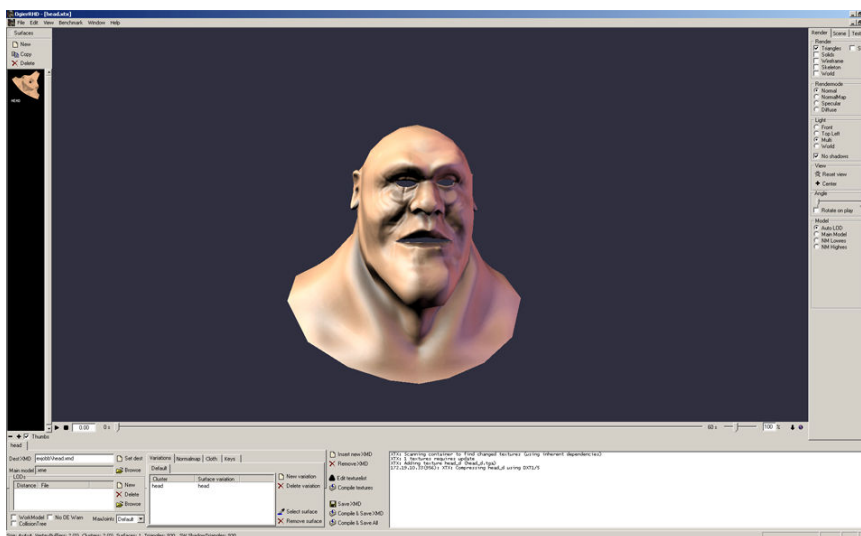
## LTU Skellefteå



Resultatet förhandsgranskas i ett fönster som dyker upp vid renderingen.



Eftersom Ogier använder ett eget format för sina normalmappar måste den genererade normalmappen från xNormal konverteras så att den tolkas korrekt av motorn. Starbreeze har utvecklat ett script i Photoshop som automatiserar detta.

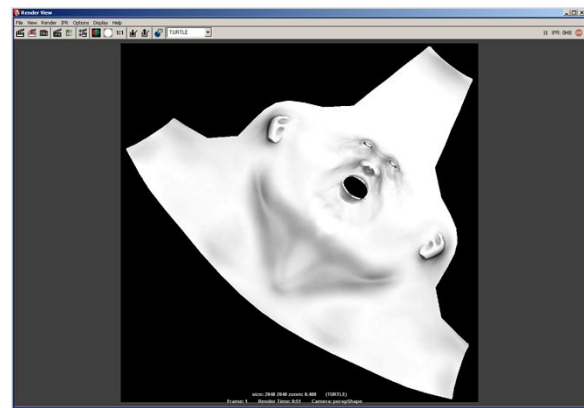
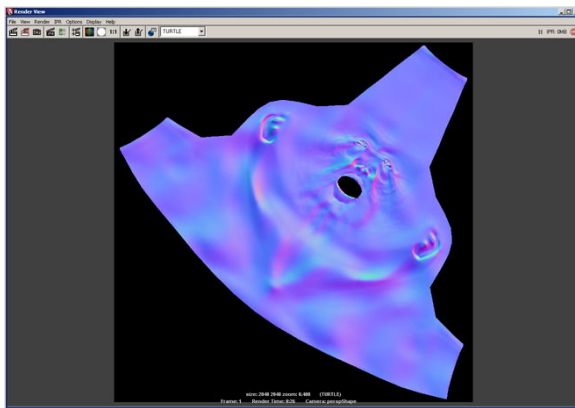
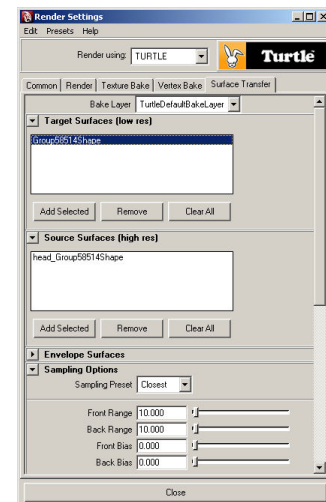


De bakade texturmapparna importeras i Ogiers materialeditor och appliceras på modellen.

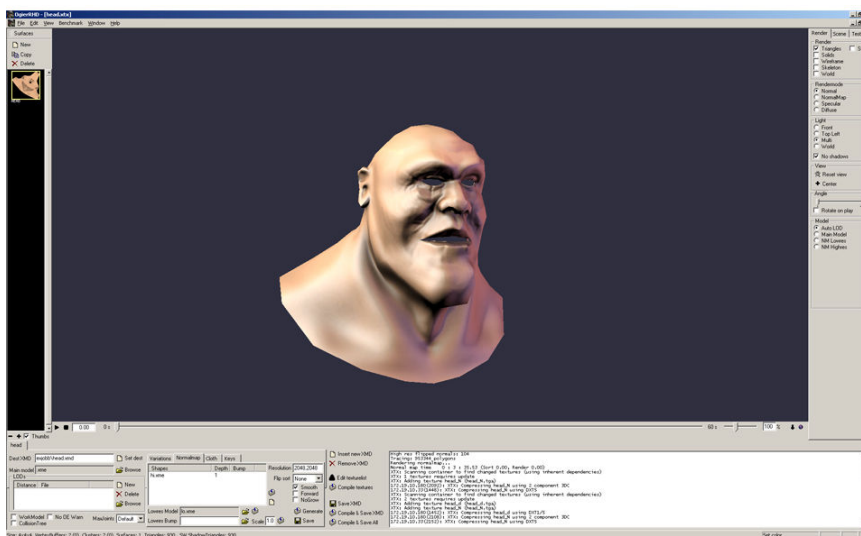
## Turtle

Turtle laddas som tidigare nämnts in som en renderare i Maya. Någon exportering är därför inte nödvändig.

Användaren definierar vilken av modellerna som är lågupplöst respektive högupplöst genom funktionen Surface Transfer. När inställningarna för bakningen är gjorda använder man Mayas vanliga render-kommando.



Bakningsresultaten visas i "Render View" där vanliga renderingar visas i normala fall. Precis som i fallet med xNormal måste normalmappen konverteras till Ogiers format.



Texturmapparna appliceras på modellen i Ogiers via materialeditorn.

## **Prestanda och resultat**

### ***Testmetod***

Samtliga tester utgår från samma grundmaterial och samma fil i Autodesk Maya. De tidtagningar som görs startar när kommandot för att påbörja bakningen ges och stoppas när programmet rapporterar att bakningen är slutförd.

De tider som presenteras ger ingen exakt bild men en god fingervisning om hur de olika programmen presterar gentemot varandra. De slutgiltiga resultatbilderna tas när modellerna och de tillhörande texturen har laddats in och applicerats i Starbreeze motor Ogier.

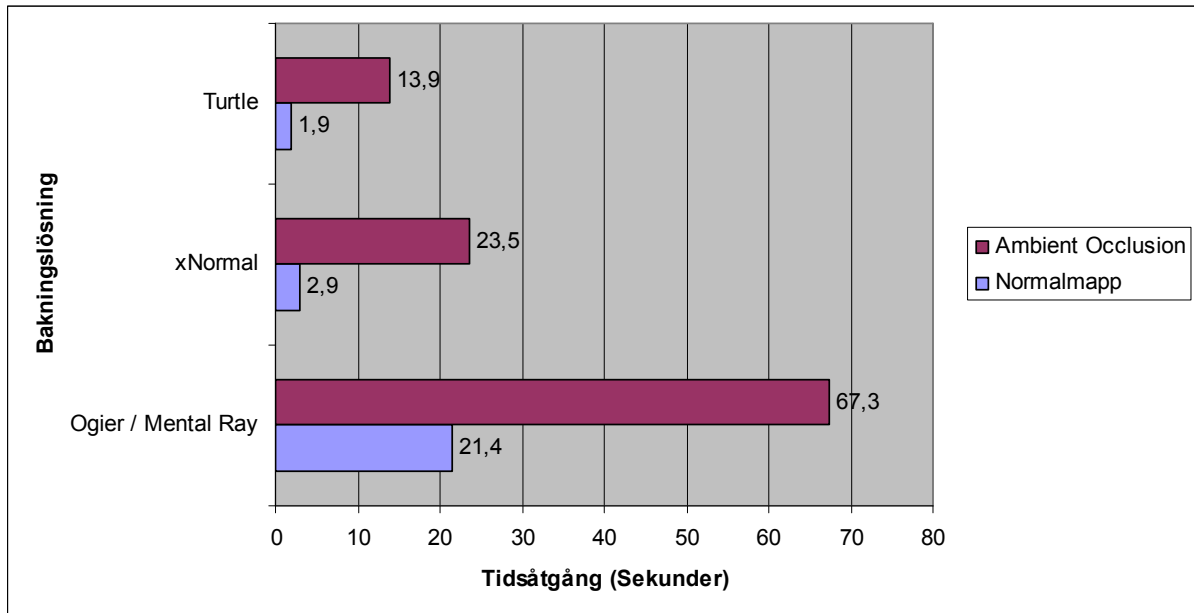
För att ge en rättvis jämförelse har programmen ställts in för att använda så likvärdiga inställningar som möjligt men samtidigt ge ett jämförbart visuellt resultat. I de fall kvalitén har blivit för dålig i ett specifikt program har inställningarna höjts tills ett jämförbart resultat har uppnåtts. Detta främst för att fasa fram vilken eller vilka lösningar som fungerar snabbast med krav som känns trovärdiga i spelindustrin.



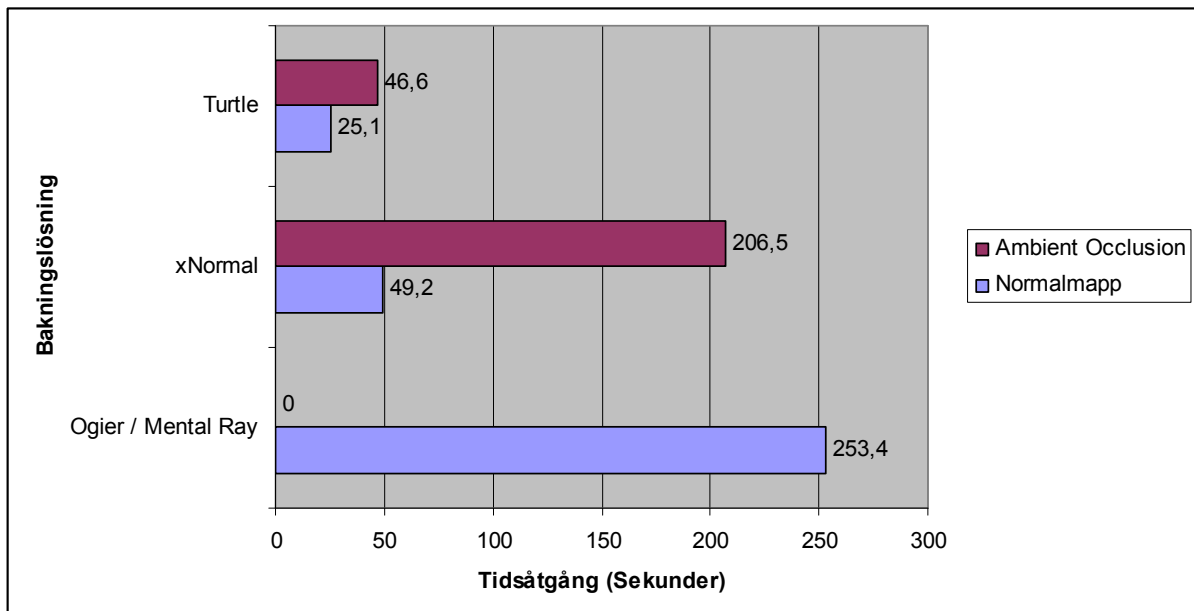
LTU Skellefteå

## Tidsprestanda

### Metallsköld



### Ansikte



## Visuella resultat

### Metallsköld



### Ansikte



## Diskussion

### Arbetsflöde

#### Ogier

Arbetsflödet med Ogier kan te sig en aning förvirrande till en början, och det är en förhållandevis lång process innan man kommer igång med genereringen av sina mappar. De till synes omständiga stegen ser dock till att man sparar sitt arbete korrekt och ger rätt sökvägar för att materialen inte ska hamna på fel plats.

Resultatmässigt finns det ett par saker värda att notera. Ogier har en högre feltolerans gentemot de andra verktygen vilket betyder att färre artefakter och spårningsmissar uppstår i renderingarna. Detta sparar förstås tid eftersom man slipper ändra inställningarna för att uppnå önskat resultat.

Ett mindre smickrande faktum är den totala avsaknaden av kantutjämning. Det finns visserligen ett alternativ som är döpt till "smooth", men i själva verket innebär inställningen att normalmappen genereras i dubbel upplösning för att sedan skalas ner till den valda storleken. Detta har som följd att minnet kan ta slut när man renderar tunga objekt i hög upplösning. Användaren måste då starta om programmet och stänga ner alla andra program som används samtidigt.

Som tidigare har nämnts så saknar Ogier stöd för att generera Ambient Occlusion på egen hand. Därför används gratisrenderaren Mental Ray i den här undersökningen men också generellt på Starbreeze. Mental Ray är erkänt svårarbetat och svårt att hålla nere renderingstiden med. I det enklare fallet med metallskölden fungerar det visserligen men resultatet är inte speciellt imponerande. Med ansiktet tar det tvärstopp. Programmet svarar inte på kommandon och arbetar i över 30 minuter utan resultat. Om detta beror på programmet eller på felaktiga inställningar får vara osagt, men klart är att Mental Ray inte är bland de mest intuitiva verktyg som har tagits fram.

Sammanfattningsvis så fungerar arbetsflödet relativt bra med undantag för bakningen av ambient occlusion och några irriterande brister som avsaknaden av kantutjämning. Resultatet blir bra och implementationen i motorn är förståeligt nog fullkomlig.

LTU Skellefteå

## xNormal

xNormal kan importera ett imponerande antal format in i programmet, men av de som erbjuds kan endast OBJ exporteras från Maya. Programmet kräver också att modellerna trianguleras men detta är inte något större hinder. Tyvärr lämnar Mayas OBJ-exporterare en del att önska. De enklare modellerna går bra att exportera men det högdetaljerade ansiktet orsakar flera programkrascher vid exportering. Det finns inget direkt hinder i filformatet för så avancerade modeller så problemet ligger helt enkelt i Mayas exportering.

xNormal har ett lite speciellt gränssnitt som inte känns speciellt intuitivt. Det finns inget 3d-läge utöver en otymplig extern förhandsgranskare. Man får således gissa sig fram i jakten på bra värden och inställningar.

Under genereringens gång kan man förhandsgranska resultatet vilket är mycket smidigt. Med bra förhandsvisning tar det oftast inte många sekunder att upptäcka fel, något som är omöjligt med till exempel Mental Ray. Renderingtiden är också föredömligt kort i jämförelse med Ogier och kanske främst nyss nämnda Mental Ray.

Om man ser till resultat så är det överlag bra men det finns en del kvar att önska, kanske framförallt när det gäller Occlusion-bakningen. Tydliga nivåer syns ofta i gradienterna vilket drar ner resultatet en hel del. Normalmapparna blir dock oftast bra. xNormal genererar också många andra sorters mappar, vissa automatiskt och andra efter önskemål. Det är förstås aldrig en nackdel.

Konverteringen till Ogiers normalmappsformat är lättare sagt än gjort. Visserligen finns det ett script som automatiserar det hela, men olika program har olika praxis på vilken information som representerar vad i normalmappen. Följaktligen kan det vara lite klurigt att få alla färgkanaler på rätt plats.

Sammanfattningsvis så kan man säga att xNormal besitter en enorm potential i sin råprestanda och det faktum att programmet är gratis. Tyvärr är gränssnittet ett problem som är svårt att bortse från.

LTU Skellefteå

## Turtle

Eftersom Turtle är integrerat i Maya så blir arbetsflödet rakare. Man slipper struliga exporteringsprocesser och kan jobba mer direkt med sitt material. Inställningarna och gränssnitt för Turtle är välbekanta för dem som har använt någon av Mayas renderare men för övriga kan det förmodligen te sig en smula förvirrande.

Resultatmässigt presterar Turtle väldigt bra, både tids och kvalitetsmässigt. Förhandsgranskningen gör det precis som med xNormal enkelt att avbryta och ändra i sina inställningar. Med standardinställningar producerar Turtle mer artefakter och spårningsfel än till exempel Ogier. Inställningarna behöver finjusteras för att uppnå önskat resultat.

Turtle hanterar höga texturupplösningar bra till skillnad från exempelvis Ogier som får problem med minneshantering vid för stora operationer.

Även om Ambient Occlusion-bakningen är imponerande snabb så finns det lite smolk i glädjebägaren i form av att själva funktionen är så pass undandömd i programmet. Användaren själv får koppla ihop materialet med en speciell materialnod i Mayas materialeditor. Det skapade materialet måste sedan appliceras på det högdetaljerade objektet för att bakas ut. Visserligen inget större problem för en van användare, men en så pass grundläggande funktion skulle man vilja ha en lätthittad förinställning för.

Ett annat problem är Turtles förmåga att skräpa ner arbetsfiler med specifika noder som sedan ger problem för användare utan programvaran. Tack vare att renderaren inte laddas ur när nya scener laddas in så sprids lätt dessa noder till alla ens arbetsfiler, oavsett om man faktiskt använder Turtle i dessa eller inte. Detta är ett ganska stort irritationsmoment, men Illuminatelabs verkar medvetna om det och presenterar en metod för att lösa problemet på sin hemsida.

Om man ska se på hur Turtle fungerar tillsammans med Ogier så är det främst de olika normalmappsformaten som gör det svårt och omständigt att flytta normalmappar mellan programmen.

## **Resultat**

### **Prestanda**

Specialiserade bakningsprogram som xNormal och Turtle har en fördel av att de är specifikt designade för ändamålet. Det syns också i resultatet. Framförallt Mental Ray är känt som ett svårtyglat program för att hålla nere renderingstid och följaktligen är det också de renderingarna som tar längst tid.

Vid den mer tungjobbade renderingen med ansiktet skenar Mental Rays rejält och resultat för ansiktet uteblir på grund av att det helt enkelt inte gick att utröna huruvida programmet faktiskt jobbade eller om processen har stannat helt. Detta är förstås inte helt optimalt och kan utgöra mycket frustration vid uteblivna resultat.

Turtle presterar bäst av de tre lösningarna, tätt följt av xNormal även om skillnaderna ökar vid höga upplösningar.

### **Visuellt resultat**

Det kan vara svårt att se någon större skillnad i jämförelsebilderna och det var också avsikten enligt testmetodiken. Vissa artefakter kan spåras på exempelvis Mentalray-renderingen av Ambient Occlusion som uppvisar en ganska tydlig grynighet trots att inställningarna var satta till relativt hård filtrering.

Om man endast ser till normalmappsrendering presterar samtliga bakningslösningar ganska likvärdiga visuella resultat. En viss fördel har Turtle av att ha bra filtreringsinställningar utan att renderingstiden skenar. xNormal klarar sig också bra i sammanhanget men uppvisar som tidigare nämnts vissa artefakter i Ambient Occlusion.

LTU Skellefteå

## **Slutsats**

Så vad kan man då dra för slutsatser av det här? Är en optimerad tredjepartsprogramvara att föredra för att öka hastigheten på produktionen i ett kommersiellt projekt?

Först och främst så måste man se till de unika förutsättningarna för det egna arbetet. I Starbreeze fall finns det ett antal faktorer som talar emot användningen av extern programvara. För det första ställer de egna standarderna till problem, även om det förmodligen är relativt enkelt avhjälpt. Man kan också tänka sig är att det i framtiden kommer att behöva bakas ytterligare information som endast kan tas fram genom motorn. Om ett tredjepartsprogram redan används för bakning så kan det bli problematiskt att utvinna den önskade informationen.

En annan faktor att ta i beaktande är priset. Turtle väger i dagsläget in på dryga 1500 USD per licens <sup>[11]</sup>. Detta att jämföra med den egenproducerade lösningen som förmodligen inte kostar lika mycket i arbetstimmar att lägga till den specifika funktionaliteten som efterfrågas.

Den gyllene medelvägen kan tyckas heta xNormal, men tyvärr känns inte programmet tillräckligt moget för att fungera i ett kommersiellt arbetsflöde. Gränssnittet är som tidigare nämnts komplicerat och hela processen känns för omständlig. Prestandan är dock imponerande, och för mindre projekt och hobbymaterial borde det dock fungera alldeles utmärkt.

## **Fortsatt arbete**

Det här arbetet har egentligen bara skrapat på ytan av den bakningsteknologi som i allt högre utsträckning nyttjas av spelindustrin. Nya tekniker som parallax mapping skapar nya utmaningar och ställer hårdare krav för både grafiker och programvaran de jobbar med. Även äldre tekniker som exempelvis ljusbakning utvecklas ständigt och möjlighet till att fördjupa sig i dessa ämnen ytterligare saknas inte.

Vidare finns det ytterligare programvara att testa och jämföra med för att skapa en mer komplett bild av de konkurrerande produkterna än den som ges här. Ett mer komplext testfall som ställer högre krav på programvaran kan också vara intressant för att komma ännu närmare de oerhörda krav som modern spelproduktion ställer.

Slutligen skulle någon form av tidtagning på hela arbetsprocessen för de olika lösningarna ge en bättre bild av hur effektivt programvaran fungerar i en produktion. Risken i ett sådant fall är att resultatet kan bli helt beroende av testarens kompetens och erfarenheter av de olika programmen sen tidigare.

## **Referenser**

[1] Shannon Kietzman. What is Digital Rendering?

URL: <http://www.wisegeek.com/what-is-digital-rendering.htm>

[2] Mental Images. About mental ray.

URL: [http://www.mentalimages.com/2\\_1\\_0\\_mentalray/index.html](http://www.mentalimages.com/2_1_0_mentalray/index.html)

[3] Mental Images. Company History.

URL: [http://www.mentalimages.com/1\\_4\\_history/index.html](http://www.mentalimages.com/1_4_history/index.html)

[4] Cignoni (1998). A general method for recovering attribute values on simplified meshes

URL: <http://vcg.isti.cnr.it/publications/papers/rocchini.pdf>

[5] Bob Moyer (2002). Ambient Occlusion – a presentation by Bob Moyer

URL: <http://www-viz.tamu.edu/students/bmoyer/617/ambocc/>

[6] Starbreeze Studios. Company information

URL: <http://www.starbreeze.com/company.jsp>

[7] Starbreeze Studios. Ogier - Starbreeze Leveleditor and Multitool

URL: <http://www.starbreeze.com/ogier.jsp>

[8] xNormal. xNormal - bakes your maps

URL: <http://www.xnormal.net/>

[9] Illuminate Labs. Turtle

URL: <http://www.illuminate labs.com/turtle/the-brand-new-turtle-3>

[10] Illuminate Labs. Beast

URL: <http://www.illuminate labs.com/turtle/beast-info>

[11] Illuminate Labs. Purchasing Turtle.

URL: <http://www.illuminate labs.com/shop>