

Oppgavetekst:

Med utgangspunkt i den eksisterende modellen av Sverresborgen, er oppgaven å finne og vurdere alternativer for visualisering av livet på borgen i middelalderen. Det er ønskelig at resultatet skal kunne vises ved Sverresborg Trøndelag Folkemuseums sommerutstilling 2005. Ønsker og råd fra museet tas hensyn til under utviklingen.

Forord

Denne rapporten er resultatet av Espen Almdahl sin masteroppgave ved NTNU, Institutt for Datateknikk og Informasjonsvitenskap, linje for Algoritmekonstruksjon og Visualisering

Tidligere prosjekt ved instituttet, deriblant høstprosjektet undertegnede gjennomførte med Bård Terje Fallan høsten 2004, har resultert i en tredimensjonal virtuell modell av middelalderborgen Sion (Kong Sverres borg) i Trondheim. Det eksisterende systemet var i stand til å vise modellen med animasjon i sanntid på NTNUs RAVE-installasjon, dog med relativt lav oppdateringsfrekvens, slik at bildet ”hakker”. Målet med denne oppgaven har vært å vurdere metoder og finne best mulig løsning for en virtuell guidet tur gjennom modellen, tilpasset Sverresborg Trøndelag Folkemuseums ønsker og behov. Denne guidede turen ble utarbeidet i samarbeid med museet, og resultatet ble en sentral del av museets sommerutstilling 2005.

Oppgaven ble gjennomført våren 2005, og store deler av den ble gjort i samarbeid med Bård Terje Fallan. Det har vært en utfordrende oppgave, spesielt med tanke på å balansere det tekniske og ingeniørfaglige mot det estetiske og rent produksjonsmessige. Det ble kanskje noe mer produksjonsarbeid enn man hadde tenkt seg på forhånd, men siden resultatet tross alt skulle vises for et stort publikum hos en ekstern instans, følte vi det var viktig å levere produkt så nært opp til museets ønsker som mulig. Vi har hele veien benyttet ingeniørfaglige metoder og vår kunnskap om visualisering i våre valg, og det har vært veldig spennende å se teori bli overført til praksis.

Jeg ønsker å takke Bård Terje Fallan for et godt samarbeid, og veileder Torbjørn Hallgren for hjelp med å skaffe til veie utstyr for visning på museet, samt gode innspill og tilbakemeldinger underveis. Ønsker også å takke Sverresborg Trøndelag Folkemuseum, spesielt Regin Meyer, for godt samarbeid og positive tilbakemeldinger.

Til slutt ønsker jeg å takke Christoffer B. Jensen i Enigmatic Studios for demonstrering av mappingteknikker i praksis, og for hjelp og gode tips i forbindelse med modelleringen og tekstureringen av menneskene som nå befolker borgmodellen, samt Karstein Kristiansen ved IDI for hjelp med teknisk utstyr for testing av resultatene.

NTNU, Trondheim 16.06.2005.



Espen Almdahl

Sammendrag

Dette dokumentet er sluttrapporten til Espen Almdahls masteroppgave ved NTNU, våren 2005, ved Institutt for Datateknikk og Informasjonsvitenskap, NTNU.

Oppgaven er en videreføring av arbeidet undertegnede gjennomførte i samarbeid med Bård Terje Fallan høsten 2004 – ”Integrasjon av animerte, menneskelige figurer i Sverresborgmodellen” [Prosj04].

Målet med oppgaven har vært å finne best mulig løsning for en virtuell guidet tur gjennom modellen, tilpasset Sverresborg Trøndelag Folkemuseums ønsker og behov. Et annet delmål var å konvertere det eksisterende systemet fra IRIX til Windows, for å undersøke muligheter for parallellutvikling av IRIX- og Windowsversjoner, samt teste Bård Terje Fallans lydssystem som ble utviklet parallelt med denne oppgaven.

Det ble tatt utgangspunkt i koden fra [Prosj04], som igjen bygget på prosjektet ”Virtual Guide for a Virtual Heritage Environment” av Jorge Ordóñez Serrano våren 2004 [Serrano04]. Denne viste seg imidlertid å være veldig knyttet mot det CAVE-spesifikke API’et CAVELib. Av lisensmessige grunner var det ønskelig å løsrive Windows-versjonen fra CAVELib, noe som ikke lot seg gjøre uten dyptgående endringer i det eksisterende systemet. Underveis i oppgaven ble det derfor – i samråd med veileder Torbjørn Hallgren – besluttet å lage en prototyp på et Windows-system, basert på koden fra ”Virtuell rekonstruksjon av Sverresborg” av Kundestyrte Prosjekt gruppe 13 høsten 2003 [KPro03], slik at vi fikk testet resultatene fra Bård Terje Fallans prosjekt angående posisjonering av lyd i Sverresborgmodellen. Denne prototypen er ikke spesielt egnet for direkte videre arbeid, men vår anbefaling er uansett å bygge opp et Windowssystem fra bunnen av, eventuelt bytte til et bedre dokumentert og mer brukt API enn OpenGL Performer.

For den guidede turen til Folkemuseet ble det vurdert flere løsninger, men det endelige resultatet ble en animasjonsfilm rendret i stereovisjon direkte fra 3D Studio MAX. Filmen ble en sentral del av museets sommerutstilling 2005, og tilbakemeldingene fra museet har vært overveldende positive.

Innholdsfortegnelse

1	Innledning.....	11
1.1	Historikk.....	11
1.2	Avgrensning	12
1.3	Om oppbygningen av rapporten.....	12
2	Metodegrunnlag	13
2.1	Modelleringsteknikker	13
2.2	Mappingteknikker	16
2.3	Animasjonsteknikker.....	22
2.4	Menneskelig syn.....	23
2.5	Stereoprojeksjon.....	24
3	Krav til systemet.....	25
4	Løsningsalternativer	26
4.1	Gjenbruk og videreutvikling av eksisterende system.....	26
4.2	Bruk av annet sanntidssystem	28
4.3	Bruk av prerendret film	29
4.4	Valg av metode.....	30
5	Implementering	33
5.1	Implementering av MAX-versjonen	33
5.2	Implementering av Performer-systemet.....	40
6	Evaluering	43
6.1	Evaluering av arbeidets gang	43
6.2	Evaluering av sluttresultatet - Konklusjon	44
7	Videre arbeid	45
8	Referanseliste	47
9	Appendix	49
9.1	Appendix A - Installering av programvare	49
9.2	Appendix B - Kravspesifikasjon	50
9.3	Appendix C – Manuskript.....	52
9.4	Appendix D – Timingdiagram for animering	56
9.5	Appendix E – Referanseillustrasjoner.....	59

Figurliste

Figur 2-1 : Ekstrudering av flater	13
Figur 2-2 : Eksempler på maps	19
Figur 2-3 : Modell med og uten ekstra maps	20
Figur 2-4 : En repeterbar metalltekstur	21
Figur 2-5 : Repetert tekstur	21
Figur 2-6 : Tilpasning av envelope	22
Figur 5-1 : Utgangsskisse for modellering av mennesker	33
Figur 5-2 : Symmetrisk boksmodellering	34
Figur 5-3 : Ferdig modellert mann og kvinne	34
Figur 5-4 : UV-map av kvinnelig modell	35
Figur 5-5 : Eksempel på endringer ved bruk av Layer-funksjonen i Photoshop	36
Figur 5-6 : Eksempel på UV-mapping av hode	36
Figur 5-7 : De fire modelltypene	37
Figur 5-8 : De røde hjørnene bindes til beinet i underarmen	38

1 Innledning

Denne oppgaven er en videreføring av høstprosjektet undertegnede gjennomførte i samarbeid med Bård Terje Fallan høsten 2004 – ”Integrasjon av animerte, menneskelige figurer i Sverresborgmodellen” [Prosj04]. En stor del av grunnarbeidet for denne oppgaven ble således gjort og beskrevet i detalj i forbindelse med høstprosjektet. En del temaer vil derfor kun gjennomgå i sammendragsform i denne rapporten. For dypere beskrivelser vil det henvises til [Prosj04].

Utgangspunktet for denne oppgaven er at Sverresborg Trøndelag Folkemuseum denne sommeren (2005) skal ha en utstilling om Kong Sverre og hans borg, Sion. Det er ønsket både fra museet og instituttet at man bruker tilgjengelig teknologi for å visualisere livet på borgen rundt Kong Sverres tid på best mulig måte, gjerne i form av en virtuell guidet tur gjennom borgen.

Fagfeltet denne oppgaven faller under, kalles ”Virtual Heritage”, eller *virtuell kulturell arv*, og er et fagfelt som slår sammen arkeologi, historie og samfunnskunnskap med datateknologi, i den hensikt å gjenoppbygge og/eller bevare verdens kulturarv i virtuelle omgivelser, og formidle informasjon om kulturarven til publikum på best mulig måte.

1.1 Historikk

Prosjektet med å visualisere Sverresborg og omgivelsene rundt borgplatået har pågått ved NTNU, IDI, over en periode på flere år. Det har resultert i en relativt nøyaktig tredimensjonal rekonstruksjon av borgplatået, ringmuren, porttårnet og noe bebyggelse inne i selve borgen. Et sanntidssystem for navigering i modellen er også utviklet, men fungerer pr. i dag kun via NTNUs RAVE-laboratorium.

Dagens modell og navigeringssystem er i stor grad resultat av arbeidet til Kundestyrte Prosjekt gruppe 13, høsten 2003 [KPro03], som skrev om store deler av systemkoden for å gjøre den mer gjenbrukbar, samt gjorde store endringer i borgmodellen etter anvisning fra hovedfagsstudent i arkeologi Regin Meyer, hvis hovedfagsoppgave omhandler Sverresborgen og borgplatået. Det bør likevel nevnes at bare deler av borgplatået har vært gravd ut, slik at man har vært nødt til å ta seg visse friheter og gjøre kvalifisert gjetting basert på andre borganlegg fra samme tidsepoke, samt Sagalitteraturen.

Også diplomarbeidet til Jose Ordóñez Serrano bør nevnes i denne sammenhengen. Hans oppgave ”Virtual Guide for a Virtual Heritage Environment” [Serrano04] ser på muligheter for å implementere en virtuell guide i systemet, og en prototyp på et slikt system dannet grunnlaget for det nevnte høstprosjektet [Prosj04], hvor et animasjonssystem ble implementert og en menneskelig, animert karaktermodell satt inn for Serranos ”plassholder-guide”.

Systemet er skrevet i C++ og bygger på API’et OpenGL Performer, som er et høynivå-språk utviklet av Silicon Graphics. Det er også nært knyttet til biblioteket CAVELib, som håndterer problematikk rundt posisjonering og navigering i RAVEn samt antall projeksjonsflater og stereoprojeksjon.

1.2 Avgrensning

Siden det ses på som et absolutt krav at borgen skal være befolket, vil det fokuseres på modelleringsteknikker og modelleringsarbeid. Oppgaven vil angripes fra et ingeniørståsted, og vurderinger av forskjellige metoder vil legge grunnlag for valg av teknikker og gjennomføringen av disse.

Nye elementer vil bli prioritert fremfor endringer i den eksisterende modellen.

Når det gjelder programmering som må utføres, vil det i stor grad ses på som prototyping for å få testprogram opp å kjøre for å teste hva som er mulig, og det vil dermed ikke prioriteres å gjøre koden gjenbrukbar.

1.3 Om oppbygningen av rapporten

Kapittel 2 går gjennom bakgrunnsteori knyttet til prosjektet. Siden en stor del av oppgaven dreier seg om modellering, vil det være fokus på en del modelleringsprinsipper, mappingteknikker og et sammendrag av animasjonskapittelet fra [Prosj04]. Det vil også bli sett nærmere på menneskelig syn og hva som skal til for å gi inntrykk av 3D dybdefølelse.

Kapittel 3 går gjennom museets og instituttets ønsker, behov og krav i forbindelse med systemet.

I kapittel 4 drøftes forskjellige scenarier og løsningsmetoder for systemet, og basert på dette vil beslutningen om endelig metode tas.

Kapittel 5 beskriver implementeringen av valgene som ble tatt i kapittel 4, altså hva som ble gjort, og hvorfor det ble gjort slik.

I kapittel 6 evalueres arbeidet og resultatene.

Kapittel 7 diskuterer tanker om videre utvidelser og arbeid i tilknytning til Sverresborgmodellen.

2 Metodegrunnlag

Det finnes en rekke forskjellige metoder og modelleringsteknikker for generering av tredimensjonale objekter. Dette kapitlet vil ta for seg noen av de mest brukte metodene når det gjelder modellering, teksturering/mapping og animasjon av menneskelige karakterer. Flere bøker er brukt som referansemateriale for dette kapitlet, og det henvises til [Giambruno02], [Watt00], [Lammers02], [Finney04] og [Prosj04]. Disse teknikkene er alle aktuelle å bruke i forbindelse med videreutvikling av ”Virtual Sverresborg”, og danner grunnlaget for valgene som må tas i forbindelse med modelleringen. Kapittel 5 beskriver hvilke løsninger som ble valgt, og hvorfor nettopp de valgene ble tatt.

2.1 Modelleringsteknikker

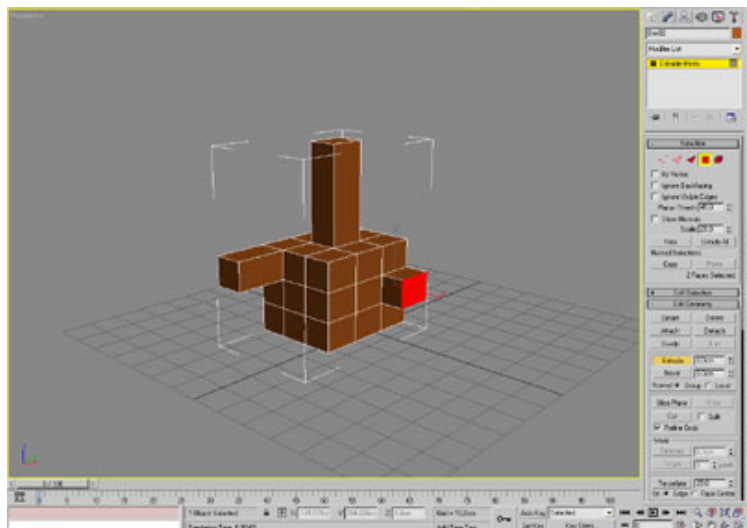
2.1.1 Polygonmodellering

Et polygon defineres av 3 eller flere hjørnepunkter definert i et tredimensjonalt rom. I prinsippet kan et polygon bestå av så mange hjørner som ønskelig, men det vanligste er 3 hjørner, slik at det danner et triangel. Ved bruk av flere hjørner for å definere ett polygon, risikerer man at ikke alle hjørnene ligger i samme plan. Dette kan medføre problemer ved *rendering* (visning av objektet på skjermen), siden man tar utgangspunkt i polygonets flatenormal for en rekke operasjoner i forbindelse med renderingprosessen..

Polygonmodellering er den eldste formen for 3D-modellering, og stammer fra tiden hvor man måtte skrive inn koordinatene til alle hjørner for hånd. Heldigvis har man i dag programvare som er litt mer brukervennlig enn som så. Man har i de fleste 3D-modelleringsprogram en rekke basisobjekter eller *primitiver* (typisk punkt, linje, plan, boks, kule, kjegle, torus og lignende) som utgjør grunnblokkene i polygonmodelleringen.

Polygonmodellering foregår i hovedsak på to måter, enten ved å manipulere flere forskjellige basisobjekter for så å sette dem sammen til ett objekt, eller ved såkalt boksmodellering, hvor man tar utgangspunkt i ett basisobjekt, som regel en boks, som man så manipulerer, deler opp og *ekstruderer* (se Figur 2-1) for å tilpasse referanseobjektet.

Et ankepunkt mot polygonmodellering er at siden modellen blir bygget opp av flate trekantede, må disse trekantene være relativt små og mange (høy polygontetthet) for at modellen ikke skal se kantete ut, spesielt i silhuett. Fordelen er at



Figur 2-1 : Ekstrudering av flater

det i prinsippet ikke er noen begrensning på hva man kan modellere ved bruk av polygonmodellering, bare polygontettheten er høy nok.

Et annet problem er at det er vanskelig å rette opp feil som ble gjort tidligere i modelleringen. Man kan derfor risikere å måtte gå mange steg tilbake for å rette feilen, og så modellere på nytt igjen fra det punktet.

Det er også mulig å generere polygonobjekter ved hjelp av mer automatiske teknikker, som

- Sveiping – I grunnen en form for ekstrudering. Tverrsnittet av det aktuelle objektet føres langs en kurve, og polygoner genereres på veien.
- Dreining – Samme som sveip, bare at tverrsnittet dreies om en akse istedenfor å føres langs en kurve.
- Skinning – Forskjellige tverrsnitt knyttes sammen av automatisk genererte polygoner
- Scanning – Man bruker eksternt utstyr, for eksempel en laserscanner, for å scanne det aktuelle objektet som skal modelleres. Man får da en punktsky som resultat, og polygoner genereres mellom dem, for eksempel ved Delauney-triangulering. Det er ofte en del etterarbeid med scannede modeller, såkalt *cleanup*, for å fjerne uønskede småfeil (for eksempel hår ved scanning av hode).

2.1.2 Splines

En *spline* er en (som regel) kurvete linje definert av *kontrollpunkter*. Siden splines er fundert på matematiske kurver, er de såkalt oppløsningsuavhengige, hvilket betyr at man i prinsippet kan zoome inn så mye man vil uten at modellen vil se kantete ut. Det eksisterer flere typer splines, men de mest brukte er Béziersplines, B-Splines, og NURBS (Non-Uniform Rational B-Splines). De er alle like på den måten at de bruker kontrollpunkter for å definere formen på kurver og flater, men måten kontrollpunktene benyttes er litt forskjellig.

2.1.2.1 Béziersplines

Opprinnelig var ideen bak Béziersplines denne: En romlig kurve er inneholdt i en kube, hvor kurvens endepunkter er definert av to hjørner diagonalt motstående i kubens. Hvis man så manipulerer kubens til å bli et parallelepiped, vil også kurven inni endres. Parallelepipedet, og dermed kurven, kan således defineres av fire kontrollpunkter. Følgende egenskaper gjelder for en Bézierkurve:

- Endepunktene er gitt av motstående hjørner i parallelepipedet.
- I startpunktet er kurven tangentiell med x-aksen i epipedet.
- I endepunktet er kurven tangentiell med z-aksen i epipedet.

Med dette som utgangspunkt kan kurven defineres mellom endepunktene på parametrisk form

$$Q(u) = \sum_{i=0}^3 P_i B_i(u), \quad 0 \leq u \leq 1$$

Hvor hvert punkt på kurven finnes ved å skalere hvert kontrollpunkts innflytelse med et polynom kjent som en basis- eller blandefunksjon. For Bézierkurver er blandefunksjonene identisk med Bernsteinpolynomene:

$$\begin{aligned}
 B_0(u) &= (1-u)^3 \\
 B_1(u) &= 3u(1-u)^2 \\
 B_2(u) &= 3u^2(1-u) \\
 B_3(u) &= u^3
 \end{aligned}$$

På matriseform kan man derfor skrive det hele slik:

$$\mathbf{Q}(u) = [u^3 \quad u^2 \quad u \quad 1] \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} [\mathbf{P}_0 \quad \mathbf{P}_1 \quad \mathbf{P}_2 \quad \mathbf{P}_3]^T$$

I praksis manipulerer man kurven ved å manipulere kontrollpunktene. Dette er en meget intuitiv måte å arbeide på.

2.1.2.2 B-Splines

Bézierkurver har to ugunstige egenskaper som B-Splines unngår. Den første, ikke-lokalitet, betyr at manipulasjon av et kontrollpunkt i større eller mindre grad har innvirkning på *hele* kurven, fordi blandefunksjonene er forskjellig fra 0 over hele u . Den andre ulempen er forholdet mellom kurvens grad og antall kontrollpunkter. Det er ikke mulig å modellere n punkter uten å bruke flere kurvesegmenter.

En B-spline er et stykkevis kubisk polynom bestående av ett eller flere kurvesegmenter. Det er et kubisk kurvesegment over et gitt intervall, og koeffisientene forandres når man beveger seg fra et kurvesegment til et annet. Et kurvesegment kan skrives på matriseform

$$\mathbf{Q}_i(u) = [u^3 \quad u^2 \quad u \quad 1] \frac{1}{6} \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{vmatrix} [\mathbf{P}_{i-3} \quad \mathbf{P}_{i-2} \quad \mathbf{P}_{i-1} \quad \mathbf{P}_i]^T$$

Hvert kurvesegment defineres av 4 kontrollpunkt, og hvert kontrollpunkt influerer 4 (og bare 4) kurvesegmenter, derfor sier man at B-splines gir lokal kontroll.

Dersom de parametriske intervallene mellom knutepunktene er like store over hele kurven, kalles det en *uniform* B-spline, mens *ikke-uniforme* B-splines tillater forskjellige intervaller.

2.1.2.3 NURBS

NURBS står for Non-Uniform Rational B-Splines, og er sammen med polygonmodellering den mest populære formen for representasjon av 3D-objekter (for eksempel er modelleringspakken Maya i stor grad basert på NURBS-modellering). NURBS tillater følgende:

- Interaktiv plassering og manipulering av kontrollpunkt
- Interaktiv plassering og manipulering av knuter
- Interaktiv kontroll over vekten av kontrollpunktene
- Interaktiv kontroll på oppløsning

Fordelen, og samtidig ulempen med NURBS, er at det er matematisk basert, og styres av en rekke parametere. Dette gjør at resultatene blir veldig nøyaktige, men samtidig kan det være en uintuitiv måte å modellere komplekse objekter på.

2.1.3 Deformasjonsbur og Subdivision Surfaces

En modelleringsteknikk som blir stadig mer populær, og som kan ses på som en slags hybrid mellom tradisjonell polygonmodellering og NURBS, er bruk av såkalte *deformasjonsbur* (Deformation cage) og subdivision surfaces. Subdivision surfaces er i utgangspunktet bare en iterativ teknikk for å øke polygonantallet i en modell for jevne ut kantene i ønskede områder, mens deformasjonsbur er en mer interaktiv måte å modellere på for å oppnå samme resultat.

I praksis går det ut på å binde en høyoppløselig polygonmodell (high-poly) til en omsluttende lavoppløselig (low-poly) modell. Man kan dermed arbeide med en low-poly modell med lav kompleksitet, mens resultatet også affekterer den bundne high-poly-modellen. Alle oppdelinger/raffineringer (subdivisions) i deformasjonsburet vil også raffinere den underliggende modellen og dermed fungere som en slags brukerstyrt subdivision-prosess.

Subdivision surfaces ble først tatt i bruk av Pixar Studios for å modellere hendene og ansiktet til Geri i kortfilmen *Geri's Game*, og var en så stor suksess at de senere har benyttet teknikken i alle sine utgivelser.

2.2 Mappingteknikker

For å øke graden av realisme i en modell, fins det en rekke *mappingteknikker*. En mapping er kort fortalt et todimensjonalt bilde projisert inn på et tredimensjonalt objekt. Dette kapitlet tar for seg de vanligste mappingtypene (diffuse, specular, bump, opacity), samt noen metoder som brukes i forbindelse med mapping. Til slutt vises praktiske eksempler på maptyper.

2.2.1 Diffuse maps

Kalles også color maps (fargekart), selv om det teknisk sett ikke er det samme (modelleringspakken LightWave skiller også mellom color og diffuse maps, men man bruker de i fellesskap) og er den formen man først og fremst forbinder med teksturmapping. Diffuse maps er ofte basert på fotografier eller innskannede bilder, men kan også være håndtegnet i et tegneprogram. Hvis man for eksempel vil modellere en treplanke vil det ikke se spesielt

realistisk ut om man lager en avlang boks og farger den lysebrun, men dersom man legger på et fotografi av en treplanke med kvistmarkeringer og andre typisk trekk, vil man umiddelbart gjenkjenne objektet som en treplanke.

2.2.2 Specularity maps

Specularity maps viser hvordan lyset reflekteres på et objekt. De kan brukes både gråtoneform for å regulere lysintensiteten på høylysflekkene, eller de kan brukes i farger for å spesifisere hvilke farger som reflekteres mest, for eksempel om man vil ha en gull-aktig refleksjon.

2.2.3 Opacity maps

Kalles også alpha maps, fordi det kan inkluderes som en såkalt alpha-kanal i enkelte bildeformat, for eksempel .TGA (Targa)-formatet. Det bestemmer graden av gjennomsiktighet på et objekt. Ved hjelp av et slikt map kan man for eksempel modellere et jerngitter ved å bruke en flate med diffusemap av jerngitter og opacity map som gir gjennomsiktighet mellom gitterstengene. Man trenger da kun ett eller to polygoner for å modellere noe som ellers ville måtte modelleres med en mengde sylindere.

2.2.4 Bumpmapping/normalmapping

Bumpmaps er gråtonebilder som forteller noe om variasjoner i overflatestrukturen på et objekt. Gråtonene perturberer normalene på flatene under renderingprosessen, slik at det gir inntrykk av små forhøyninger og forsenkninger i overflaten. Dette er meget effektivt for små detaljer som hadde gitt unødvendig mye merarbeid å modellere fysisk, som for eksempel ruheten på et appelsinskall.

Normalmapping er en relativt ny teknikk som er utviklet av spillindustrien, som har et kronisk problem med polygonbudsjetter. Man ønsker å ha så detaljerte modeller som mulig med så få polygoner som mulig for å kunne vise modellene i sanntid. Teknikken bygger på mye av det samme som bumpmapping, men brukes på mange måter omvendt. Man tar utgangspunkt i en høyoppløst polygonmodell, og bruker så en algoritme som mapper alle modellens normaler til et 2D-map. I motsetning til bumpmap som er i gråtoner og kun forteller om forholdet forhøyning/forsenkning, er normalmaps i RGB-format og forteller nøyaktig hvilken retning normalen har på dette punktet, og interaksjonen med lyskilder blir derfor enda bedre enn med bumpmaps. Teknikken gjør det mulig å vise low-poly modeller som ser ut som såkalte "ultra-high-poly"-modeller i sanntid.

Selv om bruksområdene er relativt like for de to teknikkene, kan man si at man bruker bumpmaps for små detaljer som går fortest å lage direkte i et 2D-map, som ruhet/støy, små riper og kanter, mens større detaljer som muskler, rynker på klær og lignende gjerne må modelleres fysisk for best resultat, og dermed egner normalmaps seg best. Merk også at det gir mening å bruke bumpmaps i en high-poly modell, mens normalmaps kun gir mening for low-poly-modeller.

Felles for både bump- og normalmapping er at de ikke endrer den fysiske geometrien, noe som blir tydelig om man zoomer nært nok inn på modellen, eller ser på den i silhuett.

2.2.5 Mappingkoordinater

I de aller fleste tilfeller trenger man en måte å tilordne plassering av et map på 3D-objektet man modellerer. For dette trenger man mappingkoordinater, en måte å fortelle 3D-programmet hvilken koordinat i 3D-rommet som hører til hvilken koordinat på 2D-map'et. Dette kan man bl.a. gjøre ved planar, sylindrisk eller sfærisk projeksjon.

2.2.5.1 Planare koordinater

Ved planar mapping projiseres det todimensjonale bildet rett på modellen. Dette fører til at såkalt *smøring* oppstår på flater som ikke står normalt på projeksjonsflaten, og man vil tydelig se at teksturen er ”dratt” utover flaten. Denne mappingmetoden egner seg derfor bare til plane flater eller til nød svakt kurvede flater.

2.2.5.2 Sylindriske koordinater

Sylindriske koordinater bretter bildet rundt en av objektets akser slik at det møter seg selv. Dette fører gjerne til en synlig søm, med mindre spesielle forholdsregler er tatt under generering av map'et. Denne formen for projeksjon egner seg godt til for eksempel etiketter på flasker, dekorerte søyler og hermetikkbokser.

2.2.5.3 Sfæriske koordinater

Sfærisk projeksjon bretter bildet rundt objektet som om det var en kule, og man får derfor en forvrengning av bildet rundt ”polpunktene”. På samme måte som med sylindriske koordinater, får man en tydelig søm der endene av bildet møtes, om man ikke tar forhåndsregler.

2.2.5.4 Tiling

Dersom man har en overflate med repeterende utseende, for eksempel en mursteinvegg, kan man bruke en minnebesparende teknikk kalt tiling, som vil si å bruke samme tekstur flere ganger på samme flate. Dette forutsetter at teksturbildet er tilpasset slik at overgangen mellom bildeinstansene blir sømløs.

2.2.5.5 UV-maps/skinning

De foregående metodene egner seg best på objekter som har en generell form som passer en av mappingteknikkene. I tilfellet karaktermodellering (modellering av menneskelige modeller/karakterer), har man ofte mange forskjellige former og spesielle krav til detaljer i teksturen som vanskelig lar seg løse med disse standard mappingmetodene. Her kommer UV-mapping, også kalt skinning, inn i bildet. Denne teknikken går motsatt vei av de tidligere, og innebærer at man ”bretter ut” 3D-modellen sin og legger den utover i et todimensjonalt plan. Hvert polygon vil da bli representert med nøyaktige koordinater i 2D-planet, og man kan bruke denne utbrettede modellen til å fargelegge, bumpmappe og lignende som man ønsker. Metoden vil bli nærmere beskrevet i Kapittel 5.

2.2.6 Prosedyrebaserte teksturer

Et problem med alle mappingtyper og –metoder beskrevet ovenfor, er at utstrakt bruk av mange og store maps tar opp veldig mye internminne. Et annet problem er at vanlige teksturer *piksellerer* om man kommer nært nok og teksturen ikke er av høy nok oppløsning. En mulighet for å bote på disse problemene, er å benytte prosedyrebaserte teksturer (procedural textures). Slike teksturer er matematisk definerte teksturer, altså matematiske funksjoner som tar inn gitte parametere og simulerer forskjellige overflatestrukturer som for eksempel marmor, tre eller metall.

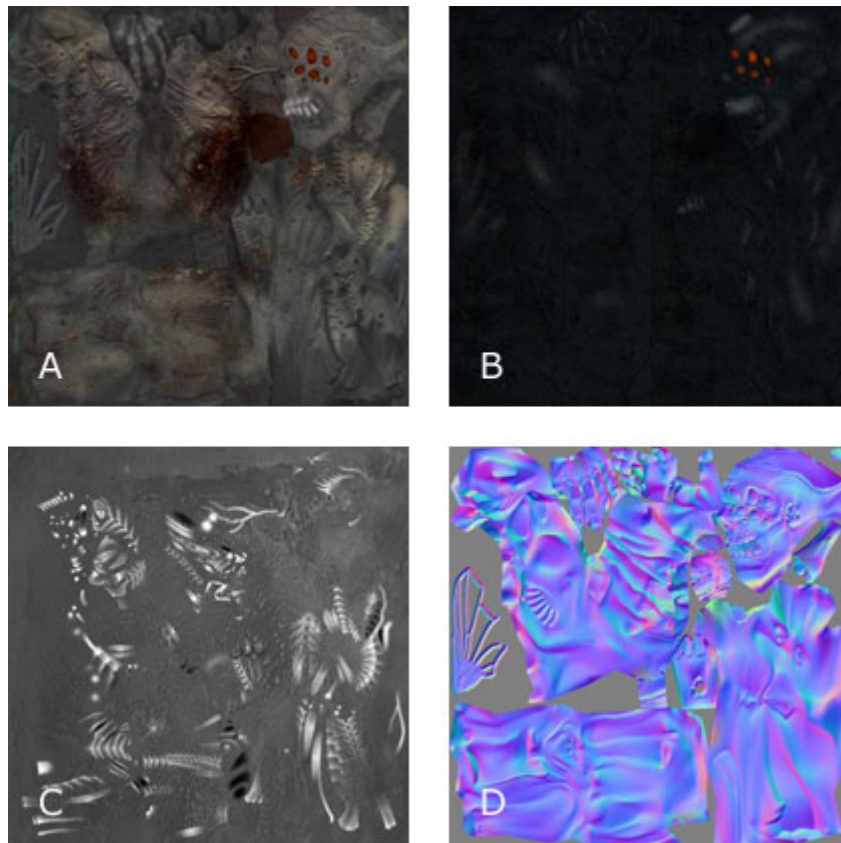
Prosedyrebasert teksturer har flere fordeler enn mindre minnebruk; de trenger ikke noen spesifikke mappingkoordinater, og de er pr. definisjon tredimensjonale, hvilket betyr at endringer i geometrien etter at teksturen er lagt til ikke får noen konsekvenser for utseendet. En tredje fordel er at slike teksturer er enkle å animere, det bare å legge til en tidsparameter i den matematiske funksjonen. Dette kan brukes for å simulere røyk, brann, vann eller lignende.

2.2.7 Billboards

Billboards er en teknikk som gjerne brukes for å ”modellere” objekter på litt avstand, og er et polygon eller plan med et eller annet bilde mappet til seg (gjerne diffuse med opacity map) som roterer om sin egen vertikale akse, slik at normalen hele tiden står rettet mot kameraet. Dette er ofte tilstrekkelig til å gi en illusjon av et 3D-objekt, og man sparer mye tid både på modellering og rendering.

2.2.8 Eksempler

I dette avsnittet vises en del figurer som illustrerer hvordan de ulike mappingteknikkene fungerer i praksis. Dersom ikke annet oppgis er bildene tatt fra spillet Doom 3 eller spilletts innebygde level editor.



Figur 2-2 : Eksempler på maps (© id Software)

I Figur 2-2 ser man eksempler på fire forskjellige typer maps, alt basert på samme UV-utbredning. Bilde A viser et diffuse/color map, bilde B er specular map, bilde C er bumpmap og bilde D er normalmap

Figur 2-3 viser effekten av bruk av flere typer maps samtidig. Det øverste bildet viser en figur og omgivelse fra Doom 3 hvor kun diffuse map er i bruk, mens det nederste bildet viser samme bilde med bruk av alle fire map-typene fra Figur 2-2, og man kan se en dramatisk økning i detaljgrad til tross for at polygonantallet er nøyaktig det samme. Dessverre kan ikke et stillbilde gjengi den fulle effekten av mappingen, siden enten objektet eller lyskilden må være i bevegelse for å illustrere forskjellen i lys- og skyggespill med og uten slike maps.

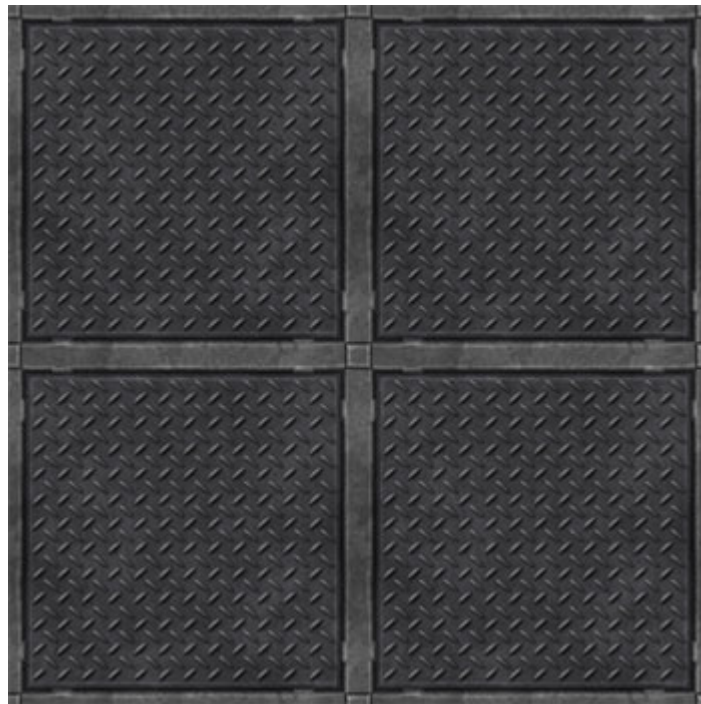


Figur 2-3 : Modell med og uten ekstra ekstra maps (© id Software)

Figur 2-4 viser en repeterbar ("tilable") tekstur laget for et nivå i spillet Quake 3 Arena. Det kan noteres at siden Quake 3 (i motsetning til Doom 3) ikke har støtte for verken dynamisk lys, specular eller bumpmaps, må alle detaljer bakes inn i diffuse map'et så godt det lar seg gjøre. Figur 2-5 viser samme tekstur repetert på alle kanter, og man kan se at overgangen mellom instansene er sømløs.



Figur 2-4 : En repeterbar ("tilable") metalltekstur (© Christoffer B. Jensen)



Figur 2-5 : Repetert tekstur

2.3 Animasjonsteknikker

Animasjonsteknikker ble grundig gjennomgått i [Prosj04], men her følger likevel en kort gjennomgang av teknikkene.

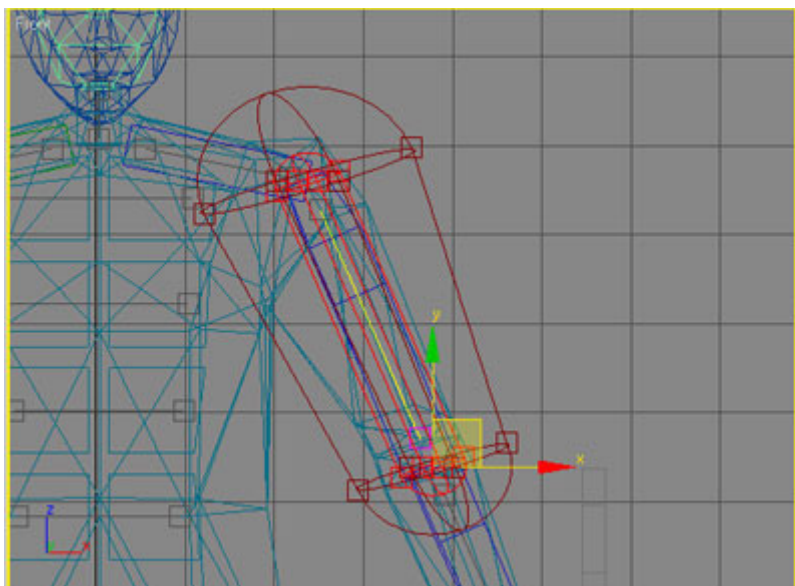
2.3.1 Keyframe-animasjon

Keyframing er en metode som har sin opprinnelse i håndanimasjon. I tradisjonell animasjon tegner hovedanimatøren nøkkelpositurer på figurene, og såkalte *inbetweenere*, eller bare *tweenere*, tegnet så alle bildene mellom utgangsposituren og sluttposituren. Overført til dataanimasjon betyr dette at man gir objektet som skal animeres en utgangsposisjon/-form og en sluttposisjon/-form, og lar datamaskinen ta tweening-jobben. Dette gjør den ved å interpolere keyframene, og denne interpoleringen kan være enten være lineær, eller styres av en kurve/spline som defineres av animatøren. For å rotere objektet under interpolering, brukes som regel quaternioner.

2.3.2 Hierarkiske modeller og skjelettanimasjon

Komplekse objekter som for eksempel karaktermodeller er vanskelig å animere ved bare å flytte polygonhjørner, og dette løses ved bruk av hierarkiske modeller og skjelettanimasjon. Man lager en rotnode som man så bygger et hierarki av *ledd* og *bein* (joints og bones), også kalt skjelett, for objektet ut ifra. På dette skjelettet kan man legge bevegelsesbegrensninger (kinematic constraints), slik at det beveger seg naturlig når man skal animere det.

Skjelettstrukturen bindes så til modellen, i en prosess kalt *rigging*. Dette kan gjøres på flere måter, den mest effektive er ved bruk av såkalte *envelopes* som omslutter beina og geometrien i nærheten av et gitt bein i større eller mindre grad (se Figur 2-6). Et hjørne kan altså være bundet til flere bein, og det benyttes da en vekting av innflytelsesgradene under animering. Man animerer altså kun skjelettet, og modellen deformeres i forhold til vektingen mellom beina i skjelettet.



Figur 2-6 : Tilpasning av envelope

2.3.3 Motion capture

Keyframeanimasjon egner seg godt til relativt enkle bevegelser og for karakterer som skal ha et tegneserieaktig bevegelsesmønster. For mer realistiske, komplekse bevegelser, er motion capture bedre egnet. En skuespiller fester da referansepunkter (magnetiske, akustiske eller reflekse), på ledd og eventuelt muskulatur (for eksempel ved ansiktsanimasjon), og bevegelsene til disse punktene mates så inn i datamaskinen og mappes til en skjelettrepresentasjon som beskrevet i avsnittet ovenfor.

2.4 Menneskelig syn

Menneskelig syn er et uhyre merkelig og komplekst fenomen, og til tross for at man vet mye om hvordan øyet og signalstrømmen til hjernen fungerer, er fenomenet *syn*, altså det faktum at vi kan se, fortsatt et mysterium [Vince95].

Menneskets syn er opprinnelig tilpasset jakt og samling av mat, og til slik bruk er *dybdesynet* det viktigste. Vi oppfatter dybde på flere måter. Noen slike tegn kan oppfattes selv om man bare ser med ett øye, disse kalles ”monokulære dybdetegn”, og er typisk basert på erfaringer med forhold mellom objekter. Tegn som krever to øyne kalles ”binokulære dybdetegn” og baserer seg på hvordan øynene fysisk er plassert i forhold til hverandre og hvordan de samhandler.

2.4.1 Monokulære dybdetegn

- *Relativ størrelse*: Store objekter virker å være nærmere enn små objekter.
- *Overlapping*: Objekter som overlapper andre, tolkes slik at det som overlapper er nærmest observatøren.
- *Relativ høyde*: Eller relativ avstand fra horisonten, jo større avstanden (høyden) mellom et objekt og horisontlinjen er, jo nærmere observatøren er objektet.
- *Lineært perspektiv*: Parallele linjer samles i horisonten
- *Skygge*: Et objekt som kaster skygge på et annet står nærmere observatøren
- *Bevegelsesparallakse*: Relativ endring av posisjon i forhold til observatøren er større for nære objekter.
- *Atmosfærisk diffusjon*: Objekter som er nært observatøren vil pga elementer i lufta (støv og lignende) være klarere enn objekter som befinner seg langt unna.

2.4.2 Binokulære dybdetegn

Øyekonvergens: Øyekonvergens er et mål på vinkelen mellom øynenes optiske akser når man fokuserer på et punkt. Liten vinkel betyr at fokuspunktet er langt unna. Man kjenner som regel ikke at man bruker musklene som styrer fokuseringen (bortsett fra når man prøver å fokusere på veldig nære objekter, skjeling), men informasjonen blir likevel brukt av hjernen for å bedømme avstand.

Stereopsi/stereoskopisk syn: Stereoskopisk syn oppnås ved at to distinkte bilder - ett bilde for hvert øye - av samme objekt (stereopsi) slås sammen til ett 3-dimensjonalt bilde. I de to bildene vil alle objekter i scenen ha *korresponderende punkter*, punkter som ved bruk av øyekonvergens lar oss fokusere på objektene slik at de (i disse punktene) virker todimensjonale. Områdene på objektet som ikke har korresponderende punkter oppfattes som overlappende dobbeltbilder. Forskjellen på disse bildene kalles *binokulær disparitet*, og brukes av hjernen for å beregne dybde.

2.5 Stereoprojeksjon

Kapittel 2.4 tok for seg hvordan menneskelig syn fungerer i det daglige liv. Hvordan kan man få til en lignende effekt med datagrafikk som i utgangspunktet vises på en todimensjonal skjerm? Dette kapitlet beskriver prinsippene bak metoden som er mest aktuell for dette prosjektet, nemlig bruk av et lerret, to projektorer og polariserende filter.

I det virkelige liv sitter øynene plassert et stykke fra hverandre (ca 6,5 cm i de fleste tilfeller), og de ser derfor ikke nøyaktig det samme, og det er dette som gjør at vi opplever verden som tredimensjonal. For å simulere dette i datagrafikk, er vi derfor avhengig av å ha to virtuelle kameraer som fungerer som øynene våre. De må være plassert i scenen i samme plan, men med en viss avstand mellom seg. Hvor stor denne avstanden må være, varierer med bruksområde og graden av 3D man vil oppnå.

Videre trenger man to projektorer som viser overlappende bilder av det høyre og venstre kamera fanger opp i scenen. Ved bruk av horisontale og vertikale polariseringsfilter foran projektorerne, og tilsvarende filter i et brillepar foran øynene, samt et spesielt lerret som ikke bryter polariseringen, oppnår man at høyre og venstre øye ser forskjellige bilder, akkurat som i virkeligheten. Bildet vil da se ut som det ligger utenpå (og inni) lerretet.

De vanligste måtene å sende videosignalene til projektorerne på, er 3D signalstrøm og side-ved-side-strøm. Avhengig av tilgjengelig utstyr vil en av metodene bli brukt i forbindelse med dette prosjektet.

2.5.1 3D signalstrøm

Denne metoden krever at videosignalet er multiplekset slik at bilder for høyre og venstre øye kommer annenhver gang. En demultiplekser fordeler signalstrømmen på de to projektorerne, slik at de viser henholdsvis høyre og venstre bilde. Denne metoden brukes også til spesialutstyr som aktive LCD-lukker-briller og VR-hjelmer (Head Mounted Display), og krever en spesiell 3-pins utgang på skjermkortet.

2.5.2 Side-ved-side-strøm

Denne metoden krever et skjermkort med to VGA/DVI-utganger og kapasitet til å vise to uavhengige skjermbilder samtidig. Videostrømmen være av en slik art at høyre og venstre bilde vises samtidig, side ved side. Når dette så vises i fullskjerm-modus, vil venstre bildehalvdel, og dermed venstre bilde, sendes ut på den ene utgangen og vises på den ene projektoren, mens det høyre bildet vil sendes ut og vises på den andre.

3 Krav til systemet

Både Sverresborg Trøndelag Folkemuseum og Institutt for Datateknikk og Informasjonsvitenskap (IDI) har en del krav og ønsker for prosjektet, og dette vil diskuteres i dette kapitlet. En fullstendig kravspesifikasjon er inkludert i Appendix B.

Det er noen absolutte krav som gjør seg gjeldende for å gjøre "Virtual Sverresborg" så levende som mulig; inkludering av menneskelige karaktermodeller og lyd. Begge disse to elementene er helt essensielle for å gjøre modellen troverdig.

Andre absolutte krav er for museets del at sluttresultatet må kunne oppleves av mer enn 1 person av gangen hvis det skal ha noen hensikt å vise det på en utstilling. Fra IDI er det et absolutt krav at resultatet skal kunne kjøres på en Windowsbasert pc, slik at man ikke er avhengig av tilgang til RAVEn for å kjøre en demonstrasjon av modellen.

Elementer som har *høy* prioritet, er blant annet diversitet på modellene, altså at det skal se ut som borgen er befolket av forskjellige mennesker, ikke bare en mengde kopier av samme modell. Det er ifølge museet historisk belegg for å tro at både kvinner, barn og dyr var å finne i borgen i det aktuelle tidsrommet, i tillegg til soldater og håndverkere.

Både museet og IDI ser også gjerne at man implementerer systemet som en virtuell guidet rundtur gjennom modellen, og at denne rundturen er satt til en bestemt historisk setting. Manuskript for dette vil bli utarbeidet av museet.

Andre høyt prioritert krav, er bruk av stereoskopisk projeksjon for å gi en økt 3D-følelse, og at systemet kjører med bildeoppfriskningsfrekvens (framerate) som gjør at bildet oppleves som flytende (dvs. en framerate på 24 bilder pr. sekund eller høyere).

Lavt prioriterte krav er modellering av dyr, og utbedringer av den eksisterende modellen. Se for øvrig Appendix B for en fullstendig og nummerert kravspesifikasjon.

4 Løsningsalternativer

I dette kapittelet vil alternative løsningsmetoder for problemet vurderes ut fra oppfylleelsesgrad i forhold til kravspesifikasjonen, risiko forbundet med metodene, arbeidsmengde og gjennomførbarhet.

Utgangspunktet for oppgaven er dagens Sverresborg-applikasjon. Dette er som tidligere nevnt et sanntidssystem utelukkende for bruk på NTNUs RAVE-installasjon, noe som ikke er spesielt bruker- og demonstrasjonsvennlig. Applikasjonen kjører i sanntid, men med relativt lav *framerate* (bilder/sekund), noe som gjør opplevelsen lite tilfredsstillende. Det er heller ikke implementert noen mulighet for bruk av lyd i applikasjonen. For å kjøre på RAVEn, som er en IRIX-basert maskin, benytter systemet APIet OpenGL Performer og et eksternt bibliotek kalt CAVELib. CAVELib tar seg bl.a. av håndteringen av flere projeksjonsflater og stereoskopisk projeksjon, samt rutiner for bruk i navigering og posisjonering. Dette biblioteket eksisterer også for Windows, men av lisensmessige grunner er det ønskelig å løsrive en Windowsversjon fra CAVELib. En Windowsversjon vil uansett ikke trenge annen CAVELib-funksjonalitet enn stereoskopisk projeksjon, og dette skal være mulig å få til på andre måter.

De alternative løsningsmetodene som vil bli vurdert i denne rapporten er som følger:

- Gjenbruk og videreutvikling av eksisterende system
- Benytte et annet grafikkssystem/grafikkmotor
- Bruk av prerendret film

Felles for alle metodene er at menneskelige modeller og lyd må lages eller skaffes til veie på annen måte.

4.1 Gjenbruk og videreutvikling av eksisterende system

Første mulighet, og den som er sterkest ønsket fra IDIs side, er gjenbruk av kode fra det eksisterende systemet, og videreutvikle dette slik at det kan kjøres på Windowsmaskiner. Ønsket fra IDI er å ende opp med ett portabelt system, slik at eventuelle senere utvidelser av systemet bare trenger å implementeres ett sted, for så å kompileres på ønsket plattform. For å få til dette, må man først og fremst isolere alle kall til UNIX-spesifikke bibliotek, isolere kall til CAVELib, og man må lage et nytt navigasjonssystem tilpasset mus og tastatur istedenfor RAVens posisjoneringssystem og *wand*. [KPro03] påbegynte et slikt arbeid, men kom ikke helt i mål. Imidlertid hevder de at et slikt navigasjonssystem kan integreres i koden ved enkle grep. Man må lage en egen klasse som implementerer interfacet *UInterface*, og isolere CAVELib-kall i klassen *Navigator*. I utgangspunktet skal ingenting annet måtte endres.

4.1.1 Fordeler med gjenbruk og videreutvikling

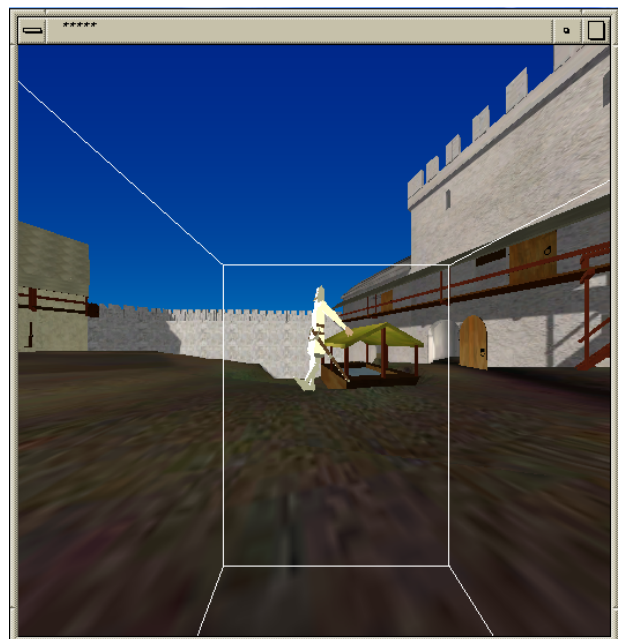
- Samme system og kode ble benyttet under arbeidet med [Prosj04]. Å bruke dette kan være tidsbesparende i forhold til å sette seg inn i ny kode.
- All kode er tilgjengelig her og nå, med relativt god dokumentasjon, jfr [KPro03] og [Serrano04].
- Alt er bygget på prosjekter og konsepter utviklet ved IDI.
- Basert på dokumentasjonen og påstander lagt fram i [Kpro03], kan man anta at en konvertering til Windows vil være gjennomførbart uten store endringer i koden. Nærmere bestemt må man lage en grensesnittklasse og fjerne kall til CAVELib i Navigator-klassen.
- Ved fjerning av CAVELib-kall vil forhåpentligvis framerate forbedres, siden det da bare blir 1 projeksjonsflate å jobbe med.

4.1.2 Ulemper med gjenbruk og videreutvikling

- API'et som blir benyttet i det eksisterende systemet, OpenGL Performer, er lite benyttet på verdensbasis, og relativt dårlig dokumentert. Det kan derfor være vanskelig å finne gode løsninger på eventuelle problemer som kan dukke opp.
- Animasjonssystemet som ble implementert i [Prosj04] er ikke spesielt avansert, og krever mye systemressurser; hvert frame i animasjonen lagres som en separat modell som lastes inn i scenegrafen.
- Støtte for lyd må implementeres.
- Støtte for stereoskopisk projeksjon må legges til, ettersom det er CAVELib som tar seg av stereoprojeksjon i det nåværende systemet.
- Det er usikkert om framerate på sluttproduktet vil bli tilfredsstillende (> 24 bilder pr. sekund).
- Det er ikke noe godt utviklet interaksjonssystem utover det å navigere i modellen.

4.1.3 Risikovurdering og konklusjon

Det eksisterende systemet er velkjent og var relativt enkelt å utvide under [Prosj04], og kildekoden er relativt godt dokumentert. Undertegnedes erfaring med C++ er begrenset til det som ble gjort i [Prosj04], og det er vanskelig å vurdere hvor stor arbeidsmengde som er påkrevd for en konvertering til Windows. Noen problemer må påregnes, men gitt at påstandene fra [KPro03] medfører riktighet, bør det være gjennomførbart. Det er imidlertid tvilsomt om det blir tid til å legge til ny funksjonalitet eller optimalisering av kode. Den største risikoen knyttes derfor til hvorvidt sluttresultatet vil gi et visuelt tilfredsstillende resultat, både med tanke på bilde kvalitet og framerate. Figur 4- viser et bilde



Figur 4-1 : Bilde fra det eksisterende systemet

generert av det eksisterende systemet. Programmet er kjørt gjennom VizServer, og perspektivet virker ”dratt” fordi de fire projeksjonsflatene blir emulert. Likevel ser man at bildekvaliteten, teksturer og skygger (bakt inn i teksturen som i Figur 2-4), er grei, men ikke noe mer enn det.

Konklusjonen blir at en konvertering til Windows sannsynligvis er gjennomførbart, men med begrensninger både med tanke på videre utvidbarhet og bildekvalitet.

4.2 Bruk av annet sanntidssystem

En annen mulighet er å benytte et annet sanntidssystem/grafikkmotor som allerede er bevist fungerende på Windowsbaserte maskiner, for så å tilpasse en slik motor etter tiltenkt bruk. Det eksisterer en rekke alternativer, både kommersielle motorer brukt i spill og open source-systemer. Prisene varierer fra gratis til millioner av kroner.

4.2.1 Fordeler med annet sanntidssystem

Fordeler og ulemper vil naturlig nok variere alt etter hvilke systemer man ser på, og det er et prosjekt i seg selv å vurdere et representativt utvalg i dybden. Likevel kan man med relativt stor sikkerhet si at systemet er garantert å fungere i en Windowsomgivelse, og en del systemer er også klare for andre plattformer som Linux, Mac og IRIX. Man vil også mest sannsynlig ha støtte for avansert animasjon og stereo/surroundlyd. Enkelte systemer har i tillegg innebygd støtte for stereoskopisk projeksjon. Mange systemer, spesielt spillmotorer, har innebygde *scriptsystem* (ofte med syntaks som tilsvarer subsett av språk som C, Java eller Python) som gjør det mulig å endre modellenes oppførsel, utseende, kamerabaner, triggering av hendelser og lignende, etter at de er plassert i scenen. Slike systemer er i så måte godt egnet for interaktivitet.

4.2.2 Ulemper med annet sanntidssystem

En stor ulempe ved enkelte slike systemer/motorer, er at de gjerne krever at den virtuelle verdenen er bygget i en proprietær editor, tilpasset systemets spesifikasjoner. Man kan derfor risikere at hele eller deler av Sverresborgmodellen må, eller i alle fall *bør* bygges på nytt i en slik editor. Det vil dessuten være tidkrevende å sette seg inn i systemkoden hvis man vil gjøre endringer i den.

4.2.3 Risikovurdering og konklusjon

Risikoen ved denne løsningen ligger i tidsbruken. Det vil ta lang tid å vurdere flere systemer opp mot hverandre, sette seg inn i bruk av verktøy og systemkode, eventuelt bygge opp ”virtual Sverresborg” på nytt i en editor. I tillegg må man lage modeller og alt som hører til i forbindelse med utvidelse av dagens eksisterende Performer-system. Resultatet vil sannsynligvis bli et bedre optimalisert og mer bruker- og utviklervennlig system med flere funksjoner og bedre bildekvalitet enn om man skal fortsette med Performer-systemet, men det vurderes likevel dithen at et slikt arbeid neppe er gjennomførbart i det tidsvinduet dette prosjektet er berammet til.

4.3 Bruk av prerendret film

En tredje mulighet er bruk av prerendret film, altså at det som vises ikke skjer i sanntid, men bare er visning av en film som er generert på forhånd. Denne genereringen kan igjen skje på to måter; framegrabbing eller direkte rendring fra et 3D modelleringsprogram (i vårt tilfelle 3D Studio MAX).

4.3.1 Framegrabbing

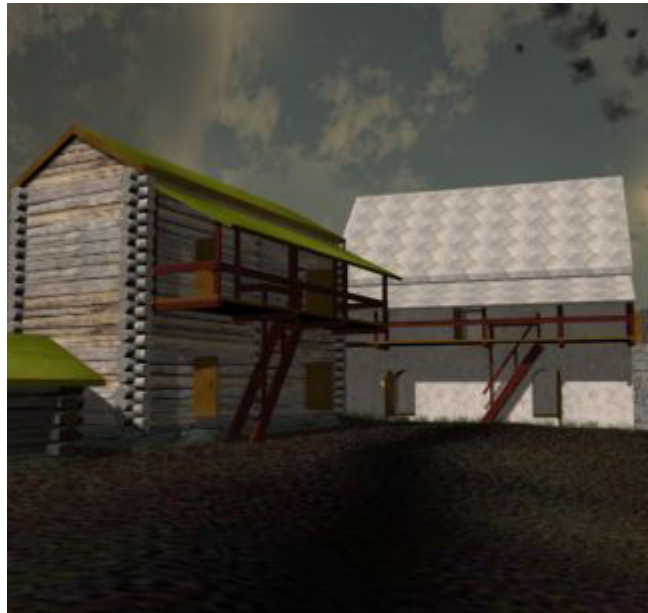
En framegrabber er et program som tar opp det som skjer på skjermen til enhver tid, omtrent som man tar opp tv-program på en videospiller. Man kan dermed "ta opp" mens man beveger seg rundt i modellen i et sanntidssystem, for så å vise denne filmen for publikum ved utstilling. Fordelen med dette er at resultatet blir seende likt ut både i filmversjon og i sanntid.

Ulemper er at eventuelle *framerate drops* (endringer i oppfriskningshastigheten, "hacking" i bildet) som forekommer i sanntidssystemet også vil bli festet til filmen, og at det ikke er gitt at framegrabberen (enten den utvikles selv eller man bruker en ferdiglaget) er i stand til å ta opp stereoprojeksjon og lyd. Dette er da elementer som må legges til i ettertid via et videoredigeringsprogram.

4.3.2 Direkte rendring

Istedenfor å bruke framegrabbing kan man benytte seg av den innebygde rendereren i 3D Studio MAX, hvor man kan lagre direkte i et filmformat som for eksempel AVI. Dette vil gi en bedre visuell kvalitet på filmen, og garanterer også at bildefrekvensen holdes konstant på ønsket hastighet. Det har også fordelen med at alle modeller og animasjoner uansett må lages og tilpasses i MAX før de eksporteres til andre systemer, og man har derfor "halve jobben gjort" i utgangspunktet.

Ulempen, i den grad det er en ulempe, er at bildene vil være av mye høyere kvalitet enn i de fleste sanntidssystemer, da renderingalgoritmene i MAX bl.a. bruker *raytracing* i sine utregninger av farge, lys og skygge (se Figur 4-2). Dette er for kostbart for sanntidsrendering. Det er også uvisst om det lar seg gjøre å rendere ut filmen i stereovisjon uten store komplikasjoner. MAX har innebygd støtte for monolyd, men stereo/surround kan legges på i ettertid gjennom et videoredigeringsprogram.



Figur 4-2 : Bilde av modellen rendret i 3DS MAX

4.3.3 Risikovurdering og konklusjon

Risikoen ved bruk av framegrabbing er nært knyttet mot hvordan sanntidssystemet oppfører seg om konverteringen blir vellykket, og er derfor et meget usikkert kort. Risikoen ved rendring fra MAX vurderes å være minimal, ettersom den eneste tilleggsjobben som må gjøres etter modelleringen er å rendre ut filmen og redigere den. Det er dog noe usikkerhet forbundet med stereorendring og hvor lang tid det vil ta å rendre ut selve filmen når modellering og animasjon er ferdig.

4.4 Valg av metode

Basert på foregående risikovurderinger, har man i samråd med Sverresborg Trøndelag Folkemuseum og IDI kommet fram til følgende:

Det er enighet om at sluttresultatet skal vises vha. projektor(er) på et lerret under sommerutstillingen 2005. Det vil for museets del være utelukket å ha en ansatt til å styre navigasjonen i en sanntidsvisning av modellen, og det er heller ikke aktuelt å la publikum selv styre navigasjonen. Dette medfører at man enten må ha en scriptet eller forhåndsprogrammert kamerabane som viser publikum rundt i borgen.

Siden det ikke vil være noen interaktivitet mellom publikum og modellen, faller de sterkeste argumentene for å skifte til et annet grafikkssystem/spillmotor bort. Dette, i tillegg til tidsaspektet beskrevet tidligere, gjør at alternativet med nytt system kan utelukkes.

IDI ser gjerne at det eksisterende systemet benyttes så langt det er mulig, men er enig i at usikkerhetsmomenter rundt konverteringen, implementering av lyd, framerate, animasjon, bildekvalitet, hardwarekrav og lignende, gjør det larest i å basere seg på at en film rendret direkte fra 3D Studio MAX vil utgjøre resultatet som blir vist på utstillingen. I tillegg må som nevnt alle nye modeller og animasjoner lages og tilpasses modellen i 3D Studio, så mye av jobben må gjøres der uansett.

Konklusjonen blir derfor at en animasjonsfilm rendret direkte fra MAX vurderes som det beste alternativet for denne oppgaven, og vil få prioritet i det videre arbeidet.

I tillegg til dette vil det eksisterende systemet bli forsøkt konvertert til Windows, for å gjøre en vurdering om det er mulig å ha ett system som kan utvikles videre og fungere på både Windows og IRIX. Et annet poeng med windowskonvertering er å få integrert og testet et lydsystem som blir utviklet som en del av Bård Terje Fallans masteroppgave parallelt med denne oppgaven.

4.4.1 Konsekvenser av metodevalget

Basert på kravspesifikasjonen og valget av løsningsmetode, må følgende gjøres videre:

- For å tilfredsstillere krav 1 – Menneskelige modeller, må en menneskelig figur modelleres og animeres. Denne figuren kan så brukes direkte i modellen som skal til museet (heretter kalt MAX-versjonen), men også i sanntidssystemet (heretter kalt Performer-versjonen) ved hjelp av metodene beskrevet i [Prosj04].
- Krav 2 – Lyd vil måtte implementeres i MAX-versjonen vha. et videoredigeringsprogram etter at filmen er ferdig rendret. For Performer-versjonen vil lyd bli integrert når konverteringen og lydssystemet er ferdig.
- Krav 3 – Visning for flere personer gjelder kun for museet, og er tatt høyde for i metodevalget ved at resultatet blir en film som skal vises på lerret.
- Krav 4 – Kjøring på Windowsmaskin gjelder i utgangspunktet kun for IDI. I MAX-versjonen blir dette implisitt ivaretatt ved at filmen kan kjøres på hvilken som helst plattform, så lenge det eksisterer avspillingsprogram for video på den. Performer-versjonen ivaretar kravet eksplisitt ved at systemet skal konverteres til Windows.

De respektive absolutte kravene vil altså bli ivaretatt i begge versjoner. Når det gjelder de resterende kravene, vil de høyt prioriterte naturlig nok søkes oppfylt i størst mulig grad. Eneste unntak er krav 11 – Interaktivitet. Siden dette bare er relevant for Performer-versjonen, og det er såpass liten støtte for interaktivitet i den opprinnelige koden, ses dette på som ikke gjennomførbart med tanke på tidsbruk. Dette kravet vil derfor bli ignorert videre i rapporten.

Lavt prioriterte krav vil bli forsøkt oppfylt dersom tiden tillater det.

All modellering vil gjøres med tanke på mest mulig kompatibilitet med Performersystemet og eventuelle andre sanntidssystemer.

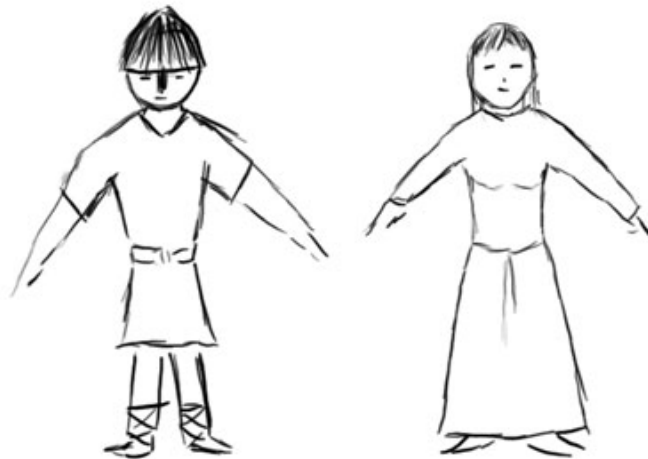
5 Implementering

Dette kapitlet tar for seg hva og hvordan ting ble gjort for å oppfylle de respektive kravene, både for MAX- og Performer-versjonen. Et manus for en guidet tur ble utarbeidet av museet (se Appendix C), og dette manuset utgjorde sammen med kravspesifikasjonen basisen for det videre arbeidet.

5.1 Implementering av MAX-versjonen

5.1.1 Modellering av mennesker

Kravene 1, 5 og 6 slås her sammen. For å oppfylle alle disse ønskene uten å bruke altfor mye tid på rent produksjonsarbeid, ble det besluttet å lage 1 mannlig modell og 1 kvinnelig modell (i praksis å regne som en modifikasjon av den mannlige). Med utgangspunkt i illustrasjonene i Appendix E, ble følgende primitive arbeidsskisse laget:

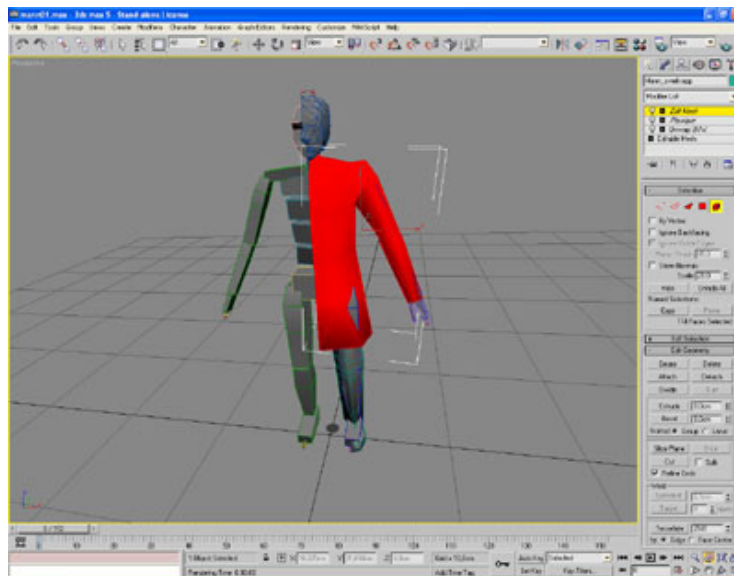


Figur 5-1 : Utgangsskisse for modellering av mennesker

Tanken var her at "kjortelen" på den mannlige modellen uten problemer kunne bekles med ringbrynje- og sivilisttekstur (heretter kalt *skin*) uten endringer i selve modellen, noe illustrasjonene i Appendix E støtter. Man kunne dermed ha mange mennesker med forskjellig utseende uten bare ved å spesifisere hvilket skin modellen skulle bruke. Hjelmen ble modellert som eget objekt, og kunne dermed fjernes fra sivilistmodellene. For å modellere kvinnen var det bare å strekke kjortelen ut til en kjole og legge til kvinnelige former.

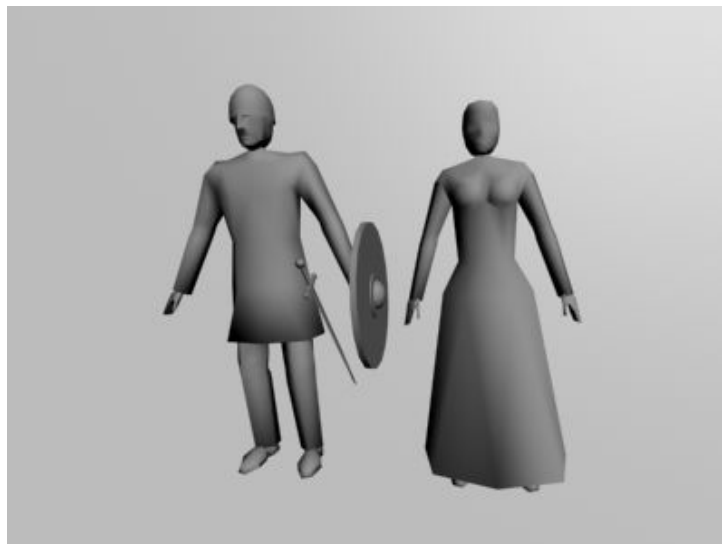
Ved selve modelleringen ble det tatt utgangspunkt i 3D Studio MAX sin innebygde *biped* (jfr. [Prosj04]), for å få proporsjoner og positur riktig, og fordi denne senere kunne brukes i animasjonsøyemed. Siden modellene er symmetriske, holdt det å modellere bare høyre eller venstre halvdel (se Figur 5-2), for så å speile denne om den vertikale akse og smelte de to halvdelene samme (MAX har en innebygd funksjon for dette som heter *weld*). Siden ingen av de involverte i denne prosessen hadde særlig mye erfaring med 3D-modellering generelt, og spesielt ikke karaktermodellering, ble det prioritert å få opp en figur som så passelig bra ut fremfor å følge et bestemt modelleringsparadigme slavisk. Siden tanken var at modellen også skulle kunne brukes i Performer-systemet, kunne den ikke være NURBS-basert og burde ha

så få polygoner som mulig. Teknikken som ble brukt under modelleringen, var derfor en form for hybrid mellom boksmodellering og sammensetting av basisobjekter (se kap. 2). Det ble i alt modellert 8 objekter; fot, legg, kjortel/kjole, hånd, hode, hjelm, skjold og sverd.



Figur 5-2 : Symmetrisk boksmodellering

For å spare litt tid fikk kvinnemodellen håret satt i en liten hestehale fremfor langt hår og skaut. Historisk sett var visst dette en litt tvilsom måte å dandere hår på, men man har heller ikke bevis for at ingen gjorde slik, så det ble godtatt. De mannlige og kvinnelige modellene ble derfor seende ut som i Figur 5-3



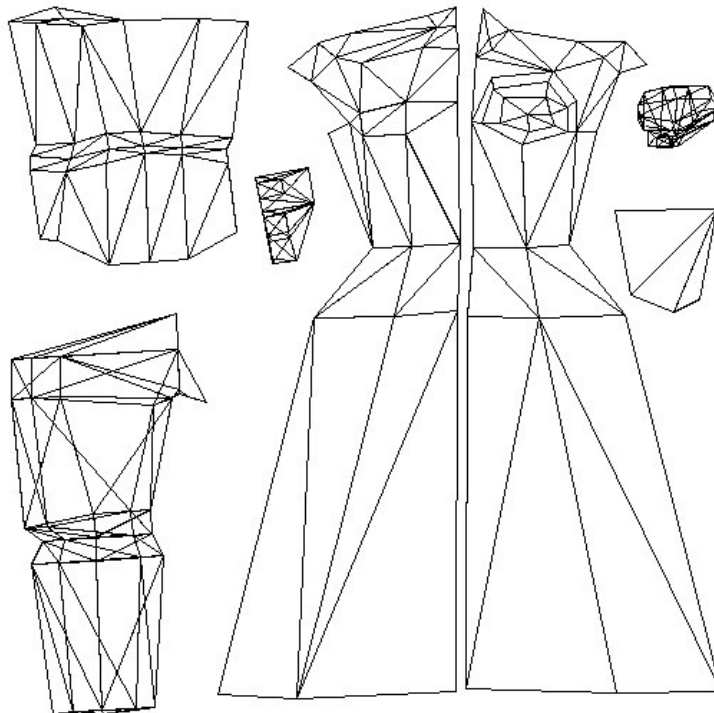
Figur 5-3 : Ferdig modellert mann og kvinne

Som barnemodeller ble skalerte utgaver av den mannlige modellen brukt.

5.1.2 Bekledning av modellene

For bekledning/teksturering/skinning av modellen sto det mellom to valg; den innebygde materialeditoren i MAX, eller teknikken som gjerne benyttes i spill, såkalt UV-mapping. Det første alternativet er best egnet for enhetlige objekter som trenger en spesifikk struktur eller er laget av et bestemt materiale, eventuelt modeller av såpass høy tetthet av polygoner at detaljer i strukturen faktisk er modellert. Man bruker da mappingprosjeksjonene beskrevet i kapittel 2 (plan-, sylinder- eller kuleprosjeksjon) på isolerte deler av modellen, og bruker projeksjonsmetoden som passer de respektive delene best.

For sammensatte modeller med lavt polygonantall er UV-mapping bedre egnet. Denne metoden er også beskrevet i kapittel 2, men gjennomgås i mer detalj her. Metoden går i praksis ut på å ”brette ut” hele 3D-modellen i et todimensjonalt plan (gjøres vha. *Unwrap UVW*-funksjonen i MAX). Dette resulterer i et nett som Figur 5-4, og man kan tydelig se polygonstrukturen overført til 2D. Hvert polygon i modellen kan finnes igjen i dette nettet, og MAX holder selv styr på hvilke UV-koordinater som hører til hvilke polygoner i modellen. Det er verdt å merke seg symmetrien i modellen kan benyttes også i UV-mapping. Som man ser av Figur 5-4, er bare den høyre halvdel av den kvinnelige modellen UV-mappet.



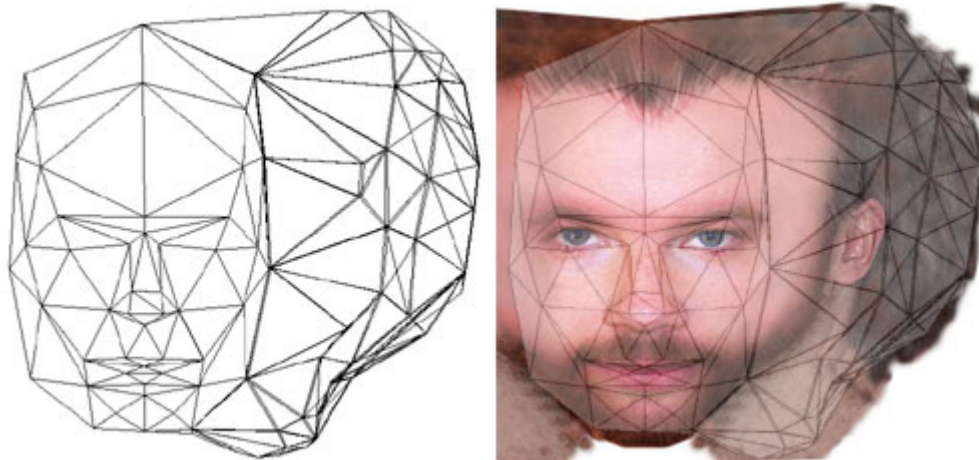
Figur 5-4 : UV-map av kvinnelig modell

Når man har fått UV-mappet klart, laster man det inn i et bildebehandlingsprogram (i dette tilfellet ble Adobe Photoshop 7.0 brukt), hvor man så fargelegger ”polygonene”. Det kan være en fordel å bruke et digitalt tegnebrett i denne prosessen (i dette prosjektet ble et Wacom Intuos2 benyttet), spesielt ved detaljtegning og skyggelegging. Man kan også bruke fotografier som grunnlag for denne fargeleggingen. Ved å bruke Layer-funksjonaliteten i Photoshop, kan man effektivt lage flere skins ved bare å skru layers av og på og endre fyllfarge. Se Figur 5-5 for eksempel på dette.



Figur 5-5 : Eksempel på endringer ved bruk av Layer-funksjonen i Photoshop

Ansiktene, sverd og skjold har egne UV-maps. For den kvinnelige modellen ble det først tegnet et ansikt, men dette tok uforholdsmessig lang tid i forhold til resten av skinningen, så for resten av ansiktene ble det prioritert å bruke fotografier. Figur 5-6 viser hvordan UV-mappingen av hodet foregår. Et foto rett forfra smeltes sammen med profilfoto og skaleres og tilpasses rutenettet som representerer hodemodellen (merk at for hodet er symmetri kun benyttet fra tinningen og til bakhodet).



Figur 5-6 : Eksempel på UV-mapping av hode (bildet er et manipulert foto av undertegnede og ble brukt som ansiktet til Torstein Kugad i animasjonsfilmen).

Figur 5-7 viser de fire modelltypene som ble modellert for Sverresborg; kvinne, soldat, sivilist/håndverker og barn. Figuren viser samtidig hvordan man effektivt kan generere vidt forskjellig utseende på mennesker basert på samme grunnmodell ved å bruke UV-mapping og fornuftig bruk av layers i Photoshop.



Figur 5-7 : De fire modelltypene

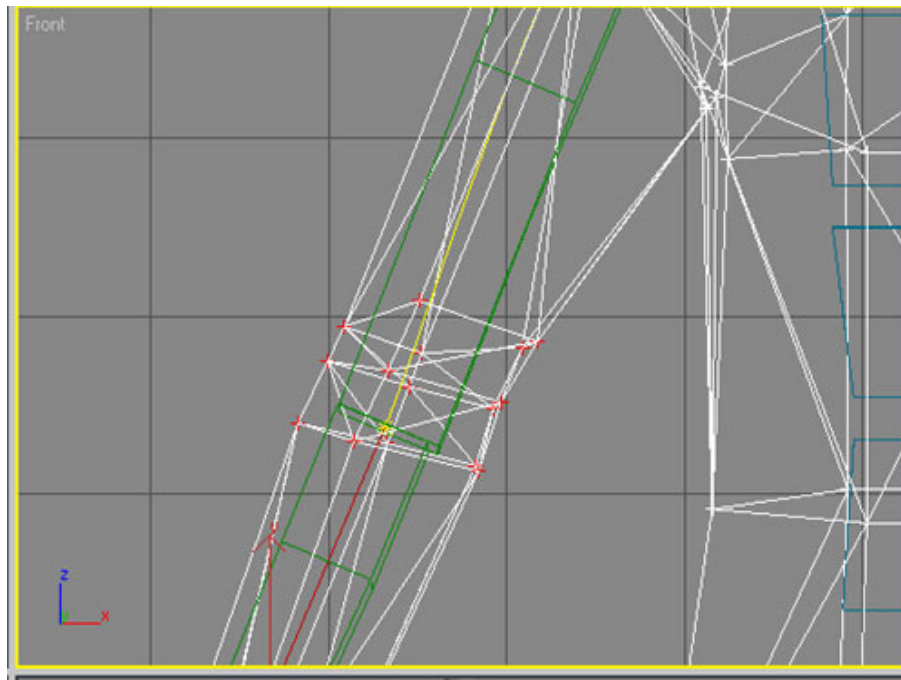
Siden Performer-systemet ikke har støtte for andre typer maps enn diffuse map, ble det besluttet å ikke benytte slike maps i MAX-versjonen, men heller bruke tiden på andre forbedringer.

5.1.3 Animering av modellene

Som nevnt ble modellene bygget rundt en biped, en ferdig skjelettstruktur innebygd i MAX. For å knytte geometri til skjelett (rigge modellen, jfr. Kapittel 2) i MAX, kan både *Physique*- og *Skin*-modifikatoren benyttes, men synet på hvilken metode som er best er i høyeste grad delt. *Skin*-modifikatoren bruker envelopes, som ble beskrevet i kapittel 2, mens *Physique* lar

brukeren spesifisere bindinger hjørne for hjørne. Ifølge [Web02] er Physique-metoden mest vanlig ved animasjon som skal eksporteres til spillmotorer, og en del motorer *krever* at figurene er rigget med Physique, så for å ta høyde for at mulige konverteringer til andre systemer skal fungere uten å måtte rigge og animere på nytt, ble Physique-metoden benyttet i dette prosjektet.

Metoden er såkalt *rigid*, og man tilordner manuelt hvert polygonhjørne til et bein i skjelettstrukturen, som vist i Figur 5-8. Ved flytting av dette beinet vil de bundne hjørnene også flyttes slik at de relative avstandene til beinet forblir de samme. Når alle hjørner er bundet, kan man sette i gang med animeringsarbeidet.



Figur 5-8 : De røde hjørnene bindes til beinet i underarmen

I mangel på utstyr til motion capture-animering, ble keyframe-metoden benyttet til generelle bevegelser, mens den innebygde Footstep-funksjonen i MAX ble benyttet for guidens gange. En tentativ timing-plan ble utarbeidet for hele filmen, basert på manuset (se appendix), og denne la grunnlaget for kamerabanen og hvor mange forskjellige modeller som skulle plasseres og animeres i scenen. Den ga i tillegg en pekepinn på hvor lang filmen ville ende opp med å bli.

Det ble prioritert å modellere karakterene fra manuskriptet (Appendix C), hvilket innebar en guide (i praksis en soldat), soldater i passiar på vaktrommet, soldater som spiller hnefatafl, en smed og en assistent som sliper pilspisser, Torstein Kugad og familien hans (tjenestefolket ble droppet). I tillegg ble noen få anonyme, forholdsvis statiske sivilister og soldater puttet inn i scenen for at borgen skulle virke mer befolket (guiden sier jo at de er over 80 personer i beredskap der).

5.1.4 Modellering av dyr

Krav 7 – Modellering av dyr hadde lav prioritet, men som det kan sees i manuset, var tanken at man skulle passere en gretten hund på rundturen i borgen. Det ble ikke prioritert å

modellere noen dyr, men en modell av en hund ble lastet ned fra [Web04] for bruk i modellen. Modellen var ikke rigget, og det ble heller ikke prioritert å bruke tid på dette. Det var bare geometrien på hodet og halsen som ble festet til bein, slik at det var mulig å animere kast på hodet ved bjeffing.

5.1.5 Andre forbedringer

En del andre forbedringer ble også gjort i henhold til kravspesifikasjonens krav 8. Siden dette var et lavt prioritert krav, ble kun de minst tidkrevende punktene oppfylt. Punkt for punkt ble følgende gjort:

- Flere bygninger langs ringmuren: Kopier av eksisterende husmodeller ble plassert langs ringmuren.
- Endring i krenelering på porttårnet: Kreneleringen på tårnkronen ble skalert ut slik at den så mer naturlig ut.
- Korreksjon av vinduer: Et vindu på porttårnet var for bredt og plassert feil i forhold til vektergangen gjennom tårnet. Dette ble fikset ved å gruppere polygonhjørnene som utgjorde vinduet, skalere dem for å gjøre vinduet smalere. De ble også flyttet til riktig posisjon. Dette førte til en forvrengning av teksturen, men denne var såpass lite synlig at publikum som ikke vet om den sannsynligvis ikke får den med seg.
- Korreksjon av vektergang i porttårnet: Meningen med vektergangen er at vaktene skulle kunne gå hele veien rundt borgen. Slik vektergangen opprinnelig var modellert, kunne ikke vaktene ha gått gjennom porttårnet. Dette ble fikset med en enkel skalering, slik at vektergangen og tårnet ble liggende inntil hverandre.
- Endring av bakketeksturen: Den opprinnelige teksturen som dekket terrenget i og rundt borgen hadde en sti som gikk langt på siden av vindebroen, dette ble korrigeret slik at stien opp til borgen nå treffer vindebroen naturlig. Det ble også lagt på farger og høylys på den delen av teksturen som dekker selve berget, slik at dette nå ser mer ut som stein.
- Vindebro: Endringer i vindebroen ble gjort slik at den kunne heves og senkes, og at den dekket inngangspartiet når den var hevet.
- Fallgitter ved inngangen: Et fallgitter innenfor vindebroen ble modellert og lagt til.
- Gresstuster ved grunnmur: For å gjøre overgangen mellom terreng og bygninger mykere, ble billboard-teknikken (jfr. Kapittel 2) med å mappe et bilde og opacity map til et polygon brukt, bare at billboardene ble låst i posisjoner parallelt med murene fremfor å rotere mot kameraet.
- Himmeltekstur ble lagt til, mappet til innsiden av en kule som omslutter hele modellen.
- Kvaderstein på porttårnet ble forsøkt modellert ved å legge på ekstra teksturmaps, men resultatet ble ikke bra, og de ble derfor fjernet igjen.
- Møblering av rom og endring av krenelering på ringmuren ble vurdert å være for tidkrevende i forhold hva man ville tjene på det rent visuelt.

I tillegg til disse punktene, ble et par elementer lagt til på bakgrunn av manuskriptet. Turen går forbi en smie, og det ble derfor laget et ildsted og en ambolt, og røykeeffekter med prosedyrebaserte teksturer (jfr. Kapittel 2) ble lagt til på husene som representerer smia og kjøkkenet.

5.1.6 Bruk av stereoskopisk projeksjon

Krav 9 – bruk av stereoskopisk projeksjon hadde høy prioritet, og den umiddelbare tanken for løsning der, var å rendre filmen med to forskjellige kameraer på samme kamerabane, ett for hvert øye, og så redigere filmene sammen til 1 film i side-ved-side-format (se kap. 2). Man kunne frykte problemer med å holde kameraene parallelle ved rotering langs kamerabanen, men plugin-modulen Stereographer tok seg av denne problemstillingen. Denne modulen genererer et kamerapar som hele tiden går parallelt med hovedkameraet. En brukerstyrt parameter bestemmer avstanden mellom kameraene, og dette representerer øyeavstanden på den virtuelle observatøren. Ved litt prøving og feiling, ble det endelige resultatet en øyeavstand som tilsvarte 1-1,5 meter i modellens målestokk. Denne øyeavstanden ga en relativt behagelig opplevelse med god dybdefølelse, men uten særlig grad av dobbeltbilde eller gjennomsiktede objekter, typiske bieffekter som gjerne oppleves med stor øyeavstand.

5.1.7 Rending og videoredigering

Hele den animerte sekvensen ble på 9700 bilder (frames). Et overslag på renderingtid tilsa at 1 datamaskin ville bruke ca. 14 dager på å rendre hele filmen med to kameraer (overslaget var et snitt på 1 minutt renderingtid pr bilde: $9700 \text{ bilder} * 2 \text{ kameraer} = 19400 \text{ minutter} = 13,5 \text{ døgn}$). Ved å dele opp filmen i bolker à 200-1500 frames, ble renderingtiden maksimalt ett døgn pr. bolk, og flere maskiner kunne benyttes samtidig. Dette resulterte i at den totale renderingtiden ble på ca. 5 døgn, inkludert retting av småfeil og re-rendering av mindre bolker.

Når filmene så var ferdig rendret, ble hver bolk lastet inn i videoredigeringsprogrammet Adobe Premiere Pro 1.5 (prøveversjonen, inkludert på DVD'en som følger med denne rapporten), hvor de ble satt sammen til en sammenhengende film, og lydeffekter og dialog ble lagt til og *pannet* til høyre og venstre lydkanal i forhold til bevegelsene på skjermen. Alle lydeffekter ble lastet ned fra [Web03], og dialogen sto museet for. Høyre og venstre film ble så mikset sammen til en stereoskopisk side-ved-side-film ved hjelp av sharewareprogrammet Stereo Movie Maker (vedlagt på DVD). For å vise filmen i stereo, brukes et annet sharewareprogram; Stereoscopic Player (også vedlagt på DVD).

5.2 Implementering av Performer-systemet

Det ble tatt utgangspunkt i koden fra [Prosj04], og det henvises til den rapporten for detaljer om alle klassene og deres funksjoner. Det er i grunnen bare følgende tre klasser som er relevante for denne oppgaven:

5.2.1 UInterface

En abstrakt klasse som definerer en del konstanter og metoder som arvende klasser må overlage. Tanken er at man skal kunne bruke et hvilket som helst hardwaregrensesnitt så lenge klassen som implementerer det arver fra UInterface.

5.2.2 Navigator

Navigator-klassen er brukerens *avatar* i den virtuelle omgivelsen. Det er denne klassen som sørger for bevegelse i scenen. Det er to måter å navigere på; flymodus og gåmodus. Gåmodus er bundet til bakken og har en rutine for kollisjonsdeteksjon, mens flymodus ikke har noen fysiske restriksjoner.

5.2.3 UIWin

Klassen UIWin ble laget for dette prosjektet, og inneholder grensesnittet mot Windowsomgivelsen. Klassen arver fra UInterface for å kunne knyttes sammen med resten av systemet. Det er i denne klassen den nye mainmetoden og renderingløkka befinner seg.

5.2.4 Hva ble gjort?

Ifølge [KPro03] skulle alle CAVELib-kall være isolert i to klasser; UICaveLib og Navigator. UICaveLib ble erstattet med klassen UIWin og alle referanser til CAVELib ble fjernet fra Navigator-klassen.

Som sagt ble det tatt utgangspunkt i siste tilgjengelige versjon av koden, og med dette fulgte en del uforutsette problemer. Det viste seg at kodetillegget fra [Serrano04] var sterkt avhengig av CAVELib, så det lot seg ikke gjøre å kompilere og linke koden med funksjonaliteten fra [Serrano04] intakt når CAVELib ble fjernet fra systemet. Det ble derfor besluttet å gå tilbake til koden fra [KPro03] og starte på nytt med den. Dette førte naturlig nok også til at tillegget fra [Prosj04] ble ubrukelig.

Men også denne tilnærmelsen førte til uforutsette problemer. Det viste seg at påstanden fra [KPro03] om at alle CAVELib-kall var isolert til klassene UICavelib og Navigator var feilaktige, bl.a. var hele menysystemet også nært knyttet mot dette biblioteket, i tillegg til at filformatet det bygget på (Inventor - .iv) ikke var støttet i Performer for Windows. Og siden menysystemet måtte fjernes (og det ikke var tid til å bygge nytt), satt man igjen med et system som var relativt strippet for funksjonalitet.

Likevel var det som nevnt ønskelig å få en prototyp opp og kjøre for å få testet det nyutviklede lydsystemet fra Fallans masteroppgave i sin planlagte omgivelse. Etter at alle referanser til CAVELib og menysystemet var fjernet, var det endelig mulig å kompilere, linke og kjøre programmet, men siden det eksisterende navigasjonssystemet var knyttet mot CAVELib, var det ikke mulig å bevege seg i modellen på dette tidspunktet. Det skulle også vise seg å by på større utfordringer enn ventet å legge til støtte for navigering med mus og tastatur. Metodene beskrevet i [Perf01] og [Perf02] hadde ingen effekt, sannsynligvis fordi de baserer seg på andre renderingmetoder enn dette system. Eksempelet i [Perf01] går ut på å ta imot en hendelsesstrøm (`extern pfuEventStream`) fra vinduet, men dette viste seg å ikke fungere i praksis, da Visual Studio ikke ville godta kodeordet `extern`. Metoden i [Perf02] er modulbasert (hvor mus- og keyboardbehandlig lastes inn som en egen modul) og krever at systemet er bygget opp rundt et `pViewer`-objekt, noe dette systemet ikke er. Det lot seg heller ikke gjøre å benytte rene OpenGL-kall direkte, selv om det ifølge [Perf02] skal være mulig. Resultatet ble derfor å bruke C++ sitt `stdin`-bibliotek, altså vanlig keyboard input. Det kinkige med dette er at `stdin` bindes til vinduet hvor programmet startes, og ikke vinduet som tegner grafikken på skjermen, og man kan derfor ikke kjøre applikasjonen i fullskjerm-modus, men må ha et lite DOS-vindu aktivt ved siden av modellvinduet.

Endringer gjort i systemet:

Klasse	Endring
Navigator	<ul style="list-style-type: none"> • Fjernet (kommentert ut) referanser og kall til CAVELib og CAVELib-funksjoner. • Lagt til metoder for navigasjon i forbindelse med keyboard input; moveForward(), moveBackward(), rotateLeft() og rotateRight(). Dette ble vurdert å være adekvat bevegelsesmulighet for å teste om lyden fungerte.
PfMenu	<ul style="list-style-type: none"> • Fjernet referanser og kall til CAVELib
PfMenuItem	<ul style="list-style-type: none"> • Fjernet referanser og kall til CAVELib
Sverresborg	<ul style="list-style-type: none"> • Fjernet referanser og kall til CAVELib • Fjernet kall til pfdInitConverter("iv"), siden denne konverteren ikke eksisterer for Windowsversjonen av OpenGL Performer. • Fjernet alle referanser til menysystemet.
UIWin	<ul style="list-style-type: none"> • Denne klassen er ny for dette prosjektet og knytter systemet til input fra tastatur istedenfor hardwaren i RAVEn, men det er basert på klassen UICaveLib (alle kall til CAVELib er naturlig nok fjernet). • Lagt til handleEvents(), som kalles fra renderingløkka og sjekker for keyboard input. Dersom bokstavene w, a, s eller d trykkes, vil korresponderende metode kalles i Navigator. Hvis 0 (null) trykkes, vil programmet avsluttes. • Det er lagt til en peker til pfChannel-objektet knyttet til visningsvinduet, slik at metoden getChannel() har noe å returnere (getChannel returnerte opprinnelig et kall til CAVELib).

6 Evaluering

6.1 Evaluering av arbeidets gang

I prosjektets startfase var det enda usikkert hvor involvert museet ville være, og hvilke muligheter de hadde til å vise sluttresultatet under sommerens utstilling. Oppgaven fikk derfor ikke noe særlig fokus før noen uker uti semesteret, da det ble klart at løsningen med lerret og projektorer var endelig og ble inkludert i utstillingsarkitektens planer.

I den tidligste fasen var altså mye av jobben å vurdere alternativer for en mengde forskjellige scenarier, så det var godt å få fokusert og avgrenset oppgaven og ta fatt på mer praktiske vurderinger. Av tidshensyn måtte noen billige løsninger velges, spesielt i modelleringsfasen. Det hadde vært spennende å kunne bruke mer tid og lære seg å bruke ulike modelleringssteknikker fremfor å måtte gå for den enkleste, og kanskje øke diversiteten på borgens beboere enda mer. Ikke minst hadde vært gøy å kunne bruke litt mer tid på lage gode animasjoner, men dette er en meget tidkrevende prosess, spesielt når man ikke sitter på noen særlig grad av erfaring.

Programmeringsarbeidet som måtte gjøres viste seg å være en mye større utfordring enn først antatt. For det første har undertegnede svært begrenset erfaring med C++ generelt, og Windowsomgivelser spesielt (arbeidet med høstprosjektet foregikk på Unix, og vi kunne bruke rammeverket fra tidligere prosjekt). Bare det å sette opp Visual Studio med referanser til kataloger, biblioteker og .dll-filer var en prøvelse i seg selv. Mye tid ble brukt på å tolke ukjente feilmeldinger og å debugge den eksisterende koden. Bl.a. viste det seg at "near" og "far", som var to variabelnavn i den opprinnelige koden som kompilerte fint på Unix, forårsaket kompileringfeil i Visual Studio fordi disse to ordene er reserverte ord i Microsofts kompilatorer.

I tillegg er som nevnt dokumentasjonen til OpenGL Performer relativt dårlig, og brukermassen er også ganske begrenset. Det har derfor vært vanskelig å finne løsninger på problemer som oppsto under konverteringsarbeidet (at mange av problemene oppsto som følge av CAVELib-problematikk gjorde ikke saken bedre), og mye tid har blitt brukt på prøving og feiling, og resultatet ble i det minste greit nok for målet det etter hvert fikk, og man fikk i alle fall et godt bevis for at parallellutvikling for IRIX og Windows på ett og samme system kan bli meget vanskelig, og sannsynlig ikke verdt arbeidet. En tommelfingerregel innen systemutvikling er at hvis mer enn 30% av et system må skrives om, er det bedre å bygge opp systemet på nytt fra bunnen av, og det kunne fastslås at noen enhetlig, utvidbart system for både Windows og IRIX ikke var mulig med den eksisterende koden pga litt for mange avhengigheter mot CAVELib og manglende støtte for filformatet brukt i menyen.

Når det gjelder samarbeidet med Sverresborg Trøndelag Folkemuseum, så har det vært meget fruktbart etter at den endelige avgjørelsen om å vise filmen på utstillingen ble tatt. Det hadde nok vært greit og fått endelig manus og ikke minst lydfilene med den innspilte dialogen noe tidligere (de ble klare en liten uke før utstillingen åpnet), men det holdt akkurat.

Det har vært spennende og motiverende å få jobbe med en såpass praktisk rettet oppgave, og det å få være med på hele prosessen fra planlegging til sluttresultat har vært veldig lærerikt.

6.2 Evaluering av sluttresultatet - Konklusjon

Basert på oppfylleelsesgraden av kravspesifikasjonen, det visuelle resultatet og tilbakemeldingene både fra museet og IDI, er det liten tvil om at det beste valget ble tatt i forhold til valg av visningsmetode og -teknologi. Alle krav unntatt kravet om interaksjon ble oppfylt ved fornuftig bruk av tilgjengelig teknologi og metoder. Resultatet ble en unik blanding av visualiseringsteknologi, historie og arkeologi, som norsk museumspublikum ikke har opplevd tidligere, og er definitivt en meget god formidlingsteknikk.

Man må også kunne si seg fornøyd med den visuelle kvaliteten på modellene, og det er vanskelig å si om valg av andre modelleringsmåter ville gitt bedre resultat med det tidsvinduet prosjektet var berammet til. Det måtte eventuelt vært ved bruk av laserscanning og motion capture-utstyr, noe som ikke er tilgjengelig ved NTNU pr. i dag. I ettertid kan man kanskje si at inkludering av bump- og specularmaps på deler av modellene, spesielt ringbrynjer, hjelmer, sverd og skjold kunne gitt et bedre resultat, men det ville hatt konsekvenser for tidsbruken både for å lage slike maps og for renderingtiden.

Når det gjelder konverteringen til Windows, var den som nevnt mindre vellykket enn man kunne ønske seg. Men den fungerte likevel til sitt formål, og man fikk svar på de spørsmålene man hadde angående videreutvikling av systemet med tanke på Windows- og IRIX-kompatibilitet: det er såpass store forskjeller i hardwarebehov og biblioteker at det vil være mer fornuftig å utvikle Windows- og IRIX-systemene hver for seg. Det anbefales for Windows-systemet sin del å vurdere et bedre dokumentert system med større brukermasse enn Performer.

7 Videre arbeid

Selv om resultatet av dette prosjektet ble tilfredsstillende, vil det ikke si at det ikke er rom for forbedringer, både på selve 3D-modellen og ikke minst i forbindelse med sanntidsvisning av den. Følgende punkter foreslås som mulige videre arbeider med Virtual Sverresborg:

- Skille mellom Windows og RAVE-system, og bygge opp Windowssystemet fra grunnen av. En del elementer, som for eksempel klassene som laster inn og bygger opp scenegrafen (SceneGraphLoader, SceneGraphBuilder), kan sannsynligvis brukes om igjen akkurat som de er, men hele renderingprosessen og hardwaregrensesnittet bør bygges opp med fokus på at systemet skal brukes i Windows, og kontrolleres med mus og tastatur, ikke romlig posisjon og wand.

Systemet bør også implementere en mer dynamisk måte å plassere objekter i scenen på. Slik det er i dag må objektene tilpasses i MAX og eksporteres derfra med offset-informasjon.

Det vil også være naturlig å ta høyde for mer interaktivitet, lyd og et mer robust animasjonssystem i et slikt nytt system.

- Dersom fokus for fremtidige utvidelser skal være økt interaktivitet, anbefales det å se alternative systemer. Et greit prosjekt kan for eksempel være å undersøke et bredt utvalg av kommersielle og open source spillmotorer og grafikk-system og bygge et mer interaktivt Virtual Sverresborg på det som passer best. Disse systemene har som oftest de fleste eller alle elementene beskrevet i forrige punkt ferdig implementert, og man slipper da å finne opp hjulet på nytt, og kan heller konsentrere seg om å gjøre selve modellen så detaljert som mulig og legge til interessante interaktivitetsmomenter.
- Modellene som ble laget for dette prosjektet er relativt enkelt animert, og det kan kanskje være en passende oppgave å lage mer detaljerte og menneskelige bevegelser. Det kunne kanskje være aktuelt å se på muligheten for bruk av motion capture-data, og på denne måten gjøre borgens innbyggere virkelig levende.
- En stor utfordring i forhold til animasjon, er ansiktsanimasjon og leppesynkronisering. Dette temaet har ikke nødvendigvis noe med selve Sverresborg å gjøre, men innlevelsen hadde nok blitt enda større i dagens modell dersom figurene hadde beveget leppene når de snakket.

8 Referanseliste

[Finney04]	<p>“3D Game Programming All in One” Kenneth C. Finney Premier Press 2004 ISBN: 1-59200-136-X</p>
[Giambruno02]	<p>“3D Graphics & Animation” 2nd Edition Mark Giambruno New Riders 2002 ISBN: 0-7357-1243-3</p>
[KPro03]	<p>”Virtuell rekonstruksjon av Sverresborg” Kundestyrt Prosjekt gruppe 13 IDI, NTNU, høsten 2003</p>
[Lammers02]	<p>“Maya 4 Fundamentals” Jim Lammers with Lee Gooding New Riders 2002 ISBN: 0-7357-1189-5</p>
[L&N03]	<p>“Medieval Scandinavian Armies (1) 1100-1300” D. Lindholm & D. Nicolle Osprey Publishing 2003 ISBN: 1-84176-505-8</p>
[Perf01]	<p>“OpenGL Performer Getting Started Guide” SGI Følger med distribusjonen av OpenGL Performer. Også tilgjengelig på web: http://techpubs.sgi.com/library/tpl/cgi-bin/browse.cgi?coll=0620&db=bks&cmd=toc&pth=/SGI_Developer/Perf_PG</p>
[Perf02]	<p>“OpenGL Performer Programming Guide” SGI Følger med distribusjonen av OpenGL Performer. Også tilgjengelig på web: http://techpubs.sgi.com/library/tpl/cgi-bin/browse.cgi?coll=0620&db=bks&cmd=toc&pth=/SGI_Developer/Perf_GetStarted</p>
[Prosj04]	<p>”Integrasjon av animerte, menneskelige figurer i Sverresborgmodellen” Espen Almdahl og Bård Terje Fallan Høstprosjekt ved IDI, NTNU 2004</p>
[Serrano04]	<p>”Virtual Guide for a Virtual Heritage Environment” Jorge Ordóñez Serrano Hovedoppgave ved IDI, NTNU 2004</p>

[Vince95]	"Virtual Reality Systems" John Vince Addison Wesley 1995 ISBN: 0-201-87687-6
[Watt00]	"3D Computer Graphics" 3 rd Edition Alan Watt Addison Wesley 2000 ISBN: 0-201-39855-9
[Web01]	"How to set up a biped in 3D Studio MAX" http://drhugo.sites.uol.com.br/tutorials/BIPED.htm
[Web02]	"Fitting a mesh with Biped and Physique" http://www.ogle.com/Classes/IVC/biped_setup.pdf
[Web03]	Partners in Rhyme http://www.partnersinrhyme.com/pir/PIRsfx.html
[Web04]	3D Café http://www.3dcafe.com/index.php?option=com_wrapper&Itemid=38

9 Appendix

9.1 Appendix A - Installering av programvare

All nødvendig programvare og alt kildemateriale (kode, 3d-modeller, maps etc.) ligger vedlagt på DVD. Det gjør også filmen som ble levert til Sverresborg, både i stereo- og monoformat. Hvis man har utstyr til å vise filmen i stereo, må programmet Stereoscopic Player installeres. For visning av monofilmen kan hvilken som helst avspiller brukes, så lenge den tar .AVI-formatet. Det anbefales uansett at filmen spilles av på et system som er i stand til å gjengi stereolyd, da lyden er en sentral del av opplevelsen.

For å kjøre sanntidssystemet, må OpenGL Performer først installeres på maskinen. Deretter skal sverresborg.exe kunne kjøres direkte fra DVD'en.

9.2 Appendix B - Kravspesifikasjon

1. Menneskelige modeller	
Prioritet: <i>Absolutt</i>	Krav gitt av: Begge
Det ses på som et absolutt krav både fra IDI og museet at borgmodellen må befolkes av menneskemodeller, fortrinnsvis animerte, for å øke graden av realisme og innlevelse.	
2. Lyd	
Prioritet: <i>Absolutt</i>	Krav gitt av: Begge
Det ses på som et absolutt krav både fra IDI og museet at lyd må implementeres i modellen for å øke graden av realisme og innlevelse.	
3. Sluttresultatet må kunne oppleves av mer enn 1 person av gangen	
Prioritet: <i>Absolutt</i>	Krav gitt av: Sverresborg
Siden sluttresultatet skal være en del av museets utstilling, må det være mulig for mer enn 1 person å oppleve modellen av gangen.	
4. Sluttresultatet må kunne kjøres på en Windowsbasert pc	
Prioritet: <i>Absolutt</i>	Krav gitt av: IDI
Det er et absolutt krav fra IDI at resultatet skal kunne kjøres på en vanlig Windowsbasert pc, slik at man ikke er avhengig av RAVE-installasjonen for å bruke modellen.	
5. Virtuell guidet tur	
Prioritet: <i>Høy</i>	Krav gitt av: Begge
Det er ønsket fra begge hold at resultatet skal oppleves som en <i>virtuell guidet tur</i> gjennom borgmodellen. I dette ligger at man blir vist rundt av en guide, som så forteller om utvalgte elementer i modellen på en eller annen måte.	
6. Diverse menneskemodeller	
Prioritet: <i>Høy</i>	Krav gitt av: Sverresborg
Sverresborg ser på det som meget viktig å vise at borgen ikke utelukkende var befolket av menn i rustning, men også sivilister, håndverkere, kvinner og barn.	

7. Modeller av dyr**Prioritet:** *Lav***Krav gitt av:** Sverresborg

Museet ønsker å vise at også dyr, deriblant hunder, hester og griser bodde på borgen.

8. Endringer i den eksisterende modellen**Prioritet:** *Lav***Krav gitt av:** Sverresborg

En del endringer i den eksisterende modellen ønskes utført, deriblant:

- Flere bygninger langs ringmuren
- Endring i kreneleringen på toppen av porttårnet
- Korreksjon av vinduer
- Korreksjon av vektergang ved porttårnet
- Endring av bakketeksturen slik at den passer inngangpartiet og ser mer ut som fjell
- Fallgitter ved inngangen
- Vindebro
- Møblering av rom
- Kvaderstein på porttårnets hjørner
- Endring av krenelering på ringmuren
- Gresstuster ved grunnmur på husene
- Gjøre himmelen mer interessant

9. Bruk av stereovisjon**Prioritet:** *Høy***Krav gitt av:** IDI

Det er sterkt ønsket fra instituttet at sluttresultatet skal benytte stereoskopisk projeksjon slik at graden av 3D og dermed opplevelsen blir best mulig.

10. Oppdateringsfrekvens**Prioritet:** *Høy***Krav gitt av:** IDI

Det er sterkt ønsket at sluttproduktet skal kjøre med en bilder-pr-sekund-frekvens som gjør at skjermbildene oppleves som flytende, dvs mer enn 24 bilder/sekund.

11. Interaktivitet**Prioritet:** *Høy***Krav gitt av:** IDI

Fra instituttets side er det ønsket en interaktiv modell, altså at man beveger seg fritt rundt i borgen og kan *trigge* gitte hendelser, som for eksempel snakke med mennesker, plukke opp objekter og lignende.

9.3 Appendix C – Manuskript

Forslag til 3D-guide, Sverresborgutstillingen 1. Juni 2005

av Regin Meyer

-Intro.

Beskrivelse:

Middelaldermusikk, stemningsmusikk eller lydeffekter spilles i bakgrunnen mens det kommer en kort innføring i den historiske konteksten og Sverresborgs opprinnelse. Dette blir enten lest opp av en fortellerstemme eller en rulletekst vises på skjermen slik som ”på film”.

Forslaget under er inspirert av historiske filmer:

”...Norge, Herrens år 1197... utbrudd av borgerkrig har herjet landet i mer enn 60 år. Mange blodige lidelser har tynget uskyldige kvinner, menn og barn, såvel byfolk som bønder. Gårder har blitt brent, hus har blitt plyndret, Nidaros, Oslos og Børgvins gater stormet i blod... Rotløse opprørere kjemper mot hirdmenn, lendmenn og stormennenes hærskarer. Birkebeinere og Baglere måler kreftene på slagmarken.....Det er mange om beinet i denne striden om hvem som skal bli landets enerådende konge. En av dem, Sverre Sigurdsson fra Færøyene har bygget to borger som folk knapt har sett maken til, en ved Nidaros og en ved Bjørgvin. Sverre legger solide fundamenter når han først skal grunnlegge landets nye kongedynasti. Borgen på Steinberget ved Nidaros blir trolig Norges første borg bygget i stein. Etter at Kong Magnus falt i slaget ved Fimreite i 1184 hylles Sverre som enerådende konge. Men misnøyen er fremdeles stor blant restene til de gamle makthaverne...

Kong Sverres borg på Steinberget ved Nidaros.....

Beskrivelse:

Tilskuerne får nå se borgen i fugleperspektiv. Perspektivet går en runde rundt borgen og ender til slutt opp et stykke nedenfor borgen i skråningen i sydvest. Her står guiden som tar i mot gjestene.

-Velkomst

Beskrivelse:

Guiden vår er en birkebeinersoldat som tilhører mannskapet på borgen. Han er ikledd ringbrynje, hjelm, våpen etc.. Velkomsten foregår nede ved området hvor Lo kirka står i dag noe som vil gi tilskuerne/gjestene anledning til å se borganlegget på avstand.

Han tar vel imot gjestene på ekte sagamaner. Velkomstfrase sies.

Replikk:

”Det er godt å se at fremmedfolk fremdeles våger seg ut på landeveien i slike urolige tider. Hvem er så dere? Dere ser vennlige ut og ligner slettes ikke på noen farlige baglere. Dere har vel ikke sett noe til det fandenske pakket på veien hit? Hmmm....? Vel, dere får følge etter så skal jeg vise dere hvordan vi har det her på kong Sverres borg...

-På vei opp til borgen

Beskrivelse:

Guiden med gjestene går så opp bakken i retning mot porttårnet.

På vei opp forteller guiden raskt om situasjonen på borgen: Beredskap, antall menn og kvinner..., kong Sverres fravær ettersom han er nede i Viken, Torstein Kugad er derfor fungerende borgherre i kongens fravær.... mye bygningsarbeid pågår inne og utenpå borgen, birkebeinerne har fått god hjelp av håndverkere fra Kristkirken...

Bakgrunnslyder: Fotsteg mens guiden og gjestene går opp bakken.

Replikk:

"...pust...pes.....nå er kongen selv dessverre ikke til stedet i borga, han er nede på Austlandet for å få skikk på de opprørske vikværingene... Det er derfor Torstein som er sjef her nå, Torstein Kugad fra Gaustad på Byneset, ellers har vi ganske høy beredskap på borgen for tiden med vel 80 mann, alle godt bevæpnet, de fleste trøndere, og noen av dem har til og med tatt med seg familien..... Det kommer godt med i det daglige virket....."

-Foran porttårnet

Beskrivelse:

Til slutt kommer følget opp til porttårnet og står like foran vindebroen som er hevet.

Guiden forteller om ryktene som sier at Baglerne er på vei opp Gudbrandsdalen mot Nidaros, Guiden roper til vaktmann opp i tårnet om at vindebroen skal senkes.

Replikk:

"Som dere ser så holder vi for tiden vindebrua hevet. Ryktene sier nemlig at krigslystne Balgerene er på vei opp Gudbrandsdalen med en stor hærflokk og kan komme når som helst. Slik som de brant og herjet sist de var i Nidaros så er det lurt å ikke ta noen sjanser... "

Guiden roper til vaken oppe i porttårnet:

"Senk vindebroa!"

-Gjennom passasjen i porttårnet

Beskrivelse:

Gjestene går rolig gjennom passasjen og hører et lite smell bak seg når vindebrua går opp. Passasjen er noe mørkere enn lysforholdene utendørs. Til venstre er døren åpen inn til vaktrommet hvor gjestene ser vaktmenn som prater. (Er det her også mulig å få laget et fallgitter i passasjen som heves ?) Foran seg ser gjestene lyset som kommer inn fra borggården.

Bakgrunnslyder (med ekstra romklang ettersom scenen foregår inne i en lukket passasje):

-Vindebrua som heves og svakt smeller når den treffer veggen.

-Fotstegene til gjestene når de går.

-Bakgrunnslyder fra vaktrommet i: *..latter, prating i bakgrunnen..*

-Inne i borggården

Beskrivelse:

Gjestene møter synet at et yrende liv inne på borggården. Kvinner som henter vann i brønnen, soldater som prater og diskuterer, håndverkere som reparerer murverk, tjenestefolk som bærer proviant og varer, et par guttunger som leker, hund som bjeffer, soldater som patruljerer på ringmuren. Noen av personene inne på borggården snur seg og ser på gjestene som ankommer, andre fortsetter med sitt daglige virke.

Guiden gir en rask beskrivelse av borgplataet. Perspektivet/synsfeltet fokuserer på de forskjellige bygningene og stedene etter hvert som han prater og beskriver. Guiden og gjestene snur seg for å se på bygningsfløyene på hver side av porttårnet. Guiden forteller og viser hvor kapellet ligger, boligkvarter ligger, lagerrom, fengsel...etc..

Bakgrunnslyder: Lydeffekter som gir inntrykk av et mangfold av aktiviteter.

Replikk:

”Sånn, endelig fremme i borggården. Her bor og lever vi altså, inne på borggården godt beskyttet bak murene. Et rett så livlig og trivelig sted nå når det er ro og fred, men en heller dystert og alvorlig plass når fienden truer, det kan jeg love dere...

Vi har forskjellige hus og bygninger til mangt slags bruk.

I den andre enden ser dere kong Sverres praktfulle herberge, men det er dessverre ikke så ofte han bor der nå for tiden, langs ringmuren ligger verksteder, soldatenes forlegningshus, stall, lagerhus, og på den andre siden står kjøkkenet. Og like foran oss, kanskje det viktigste av alt, nemlig brønnen vår. Hvis baglere eller andre fiender skulle være frekke nok til å beleire oss må vi passe på at de ikke sulter oss ut, ja som dere kanskje vet: uten brønn ingen borg.... og bak oss...(gjestene snur seg)

...har vi øverst til venstre kappellet vårt, i etasjen under ligger boligkvarter for gjester, fintfolk og hirdmenn, på den andre siden gildehallen til møter, fester og høytidelige begivenheter, og under det et godt bevoktet fengsel. Der har vi fremdeles et par kulvunger lagt i fotlenker til skrekk og advarsel, etter forræderi og mannedrap.....

-På vandring mot den kongelige residensbygningen (langs ringmuren)

På vei mot den kongelige residensbygningen som ligger lengst nord på plataet passerer gjestene flere verksteder og lagerbygninger i lafta tømmer. Ved et av verkstedene får gjestene hilse på en av håndverkerne, for eksempel en smed eller en snekker? Under vandringen ser mann soldater som går oppe på ringmuren. Kanskje kommer det en hund som bjeffer etter gjestene?

Replikk:

Følg på, så skal jeg vise dere litt omkring her borte. Pass dere for den fillebikkja, den er litt gretten i dag og kan bite!

Det er nemlig slik at så lenge ikke fienden lusker omkring må vi bruke all den tiden vi har til forberedelser. I fredstid er derfor kanskje håndverkere de viktigste på borgen for å gjøre den sterk og stødig mot beleiringer. Mange dyktige håndtverkerne har vi hentet fra byen og noen utenlandske speisalister kommer til og med fra Kristkirka (Bakgrunnslyd: noen som snakker engelsk el. et fremmed språk ?) Deler av den gamle ringmuren må repareres, veggene skal kalkes, nye laftahus skal settes opp, tømmer som skal skiftes ut, proviantlagrene må fylles opp, reservedeler, pilspisser og bolter må lages, nye våpen skal smies..... ja, sukk, mye slit når mann er i kongens tjeneste, også for de som ikke er stridførne menn...

Underveis..

Underveis passerer guiden og gjestene to soldater som sitter ved et bord og spiller brettspill ”hnefatafl” (uttales: nævatafl). Guiden vår kjefter sint på dem:

Sa jeg ikke at dere to skulle ut og kalke murene? I stedet sitter dere her og spiller hnefatafl..... Av gårde før jeg setter Borghild på dere !!... ..spillegale vestlendinger...

Underveis..

Like ved passerer følget en bygning som er smia på borgen. Det kommer røyk ut en åpning i taket. Innenfra hører vi ”slag og hamrelyder”.

Guiden:

....det er bra å høre at hvertfall noen tar arbeidet på alvor, her inne holder smedene våre til, Sjur og Eirik Stål,

Og litt ved siden av smia sitter en kar på en stubbe og sliper spydspisser,

Guiden:

...og her sitter han Brynjolv og sliper våpen.....bra Brynjolv!... sørg for at de spydene biter godt fra seg hvis vi kommer i kamp....

Og bortenfor han igjen står et par snekkere som hamrer på et skur...

-Ved den kongelige residensbygningen

Her får gjestene hilse på Torstein Kugad, Sverresborg borgherre og kongens stedfortreder. Torstein forteller at han har tatt med seg kona, barna og tjenestefolket fra storgården hans på Byneset. Gjestene blir introdusert for familien.

Replikk fra Torstein:

...velkommen hit til Sion, kongens borg, jeg heter Torstein Kugad og bestemmer her på borgen når kongen selv ikke er til stede, som dere ser har jeg med meg tjenestefolket og familien fra garden min på Byneset....

...dette er eldstekaren..... han blir nok en stor birkebeiner en dag.....

...her har vi jentene.....

...og kona mi, Borghild hun er et eller annet sted....

Kona, Borghild kommer plutselig, hun har et stort nøkkelknippe festet i beltet (lyd av nøkler som rasler). Hun er lettere irritert, noen har forsynt seg av forrådslagrene. Hun begynner å kjeft på Torstein:

...så der er du!! Det er fælt så mye denne borgrampen driver og forsyner seg av forrådslagrene våre når mann ikke passer på... Nå må du få skikk på dem Torstein!! Vi har sikkert gått tom for mat lenge før vikkværingene kommer og beleirer oss..... Og Torstein, ditt surrehode, hvor mange ganger har jeg ikke sagt at du må huske å lukke latrinedøra etter deg.....den glemskheten din kan komme til å koste oss dyrt en dag!!

Borghild henvender seg til gjestene:

....så? Har vi fått fremmedfolk på besøk? Jeg pleier vanligvis ikke å vise fremmede rundt i kongens herberge... det ville Sverre nok ikke like.... men jeg kan ta dere med inn hit litt så kan dere få nyte utsikten....

-Inne i et rom i den kongelige residensbygningen

Kona til Torstein tar gjestene med inn i et lite rom i residensbygningen. De stopper ved et vindu mot fjorden og nyter utsikten i noen sekunder.

Plutselig hører de noen som blåser i lur og folk som roper:

”BAGLERNE KOMMER!!.....”

9.4 Appendix D – Timingdiagram for animering

Denne delen er ment som en oppsummering av animeringen av Sverresborgen i 3D Studio MAX. Til grunn ligger det tekstlige manuskriptet fra Regin Meyer, og hovedhensikten har vært å skape en oversikt slik at eventuelle fremtidige endringer av animasjonen/modellen blir enklere.

Tid (s)	Frames	Lyd	Beskrivelse
-30-0	X	Bakgrunnsmusikk.	Introduksjonstekst eller stemme.
0.0-0.30	0-720	Bakgrunnsmusikk?	Fugleperspektiv, sirkuler rundt borgen.
0.30-0.34	720-816	Fade bakgrunnsmusikken bort.	Ned fra fugleperspektiv til guiden (hjelm, våpen, ringbrynje) på forsiden av borgen.
0.34-0.36	816-864		2 sek pause.
0.36-0.56	864-1344	Replikk.	Velkomstreplikk av guiden.
0.56-1.21	1344-1944	Fotsteg, pust, pes, replikk.	På vei opp til borgen og porttårnet. Guiden snakker.
1.21-1.36	1944-2304	Replikk.	Foran hevet vindebro og senket gitter. Guiden snakker.
1.36-1.38	2304-2352	Replikk.	Guiden roper om senking av vindebroen.
1.38-1.43	2352-2472	Vindebrokjetting.	Senking av vindebro.
1.43-1.44	2472-2496		1 sek pause.
1.44-1.48	2496-2592	Jernlyder av gitter.	Gitteret heves.
1.48-1.54-1.58	2592-2736-2832	Lyd av kjetting pga. vindebro + smell av vindebro som går opp.	Gjennom passasjen i porttårnet.
1.58-2.01-2.04	2832-2904-2976	Bakgrunssnakk, latter fra soldater, fotsteg.	Snur venstre mot vaktrommet, der soldater sitter og snakker.
2.04-2.06-2.25	2976-3024-3480	Bakgrunnslyder: smed, snakking, bjeff ++.	Inne i borggården. "Yrende liv": se * under. Observerer en liten stund før guiden begynner å snakke.
2.25-2.34	3480-3696	Replikk, bakgrunnslyder som over.	Ser mot kongeboligen på andre siden (følg replikk fra guide på de ulike bygningene). Går mot midten av borggården og blir stående der.
2.34-2.41	3696-3864	Replikk, bakgrunnslyder som over.	Ser mot venstre side langs ringmuren på de ulike bygningene / verkstedene.
2.41-2.49	3864-4056	Replikk, bakgrunnslyder som over.	Ser mot kjøkkenet.
2.49-	4056-	Replikk, bakgrunnslyder	Ser mot brønnen.

2.52- 3.05	4128- 4440	som over.	
3.05- 3.09	4440- 4536	Replikk, bakgrunnslyder som over.	(med retning mot hovedbygningen). Fokuserer på øvre, venstre rom (kapell).
3.09- 3.13	4536- 4632	Replikk, bakgrunnslyder som over.	Fokuser på nederste etasje.
3.13- 3.18	4632- 4752	Replikk, bakgrunnslyder som over.	Fokuser mot den andre øvre delen.
3.18- 3.28	4752- 4992	Replikk, bakgrunnslyder som over.	Fokuser mot den andre nedre delen.
3.28- 3.38	4992- 5232	Replikk, bakgrunnslyder som over.	Snu mot kongeboligen.
3.38- 3.43- 3.53- 4.03- 4.14	5232- 5352- 5592- 5832- 6096	Replikk, bakgrunnslyder som over + engelsktalende håndverkere?	Går mot kongeboligen, langs rekken av verksteder langs ringmuren.
		Replikk, bakgrunnslyder som over + bjeffende hund.	På veien kommer det bort en hund som bjeffer.
		Replikk, bakgrunnslyder som over.	Se opp mot ringmuren der soldater patruljerer.
4.14- 4.26	6096- 6384	Replikk, bakgrunnslyder som over.	Passerer to soldater som spiller brettspill.
4.26- 4.34	6384- 6576	Replikk, bakgrunnslyder som over + slag og hamrelyder.	Hos smeden.
4.34- 4.43	6576- 6792	Replikk, bakgrunnslyder som over + slag og hamrelyder.	Hos smeden. Ser en person som sitter og spiser et spyd utenfor smia.
4.43- 4.47- 4.58	6792- 6888- 7152	Bakgrunnslyder som over + slag og hammerlyder.	På vei bort fra smeden. To personer som snekrer på et skur.
4.58- 5.10	7152- 7440	Replikk, bakgrunnslyder som over.	Ved kongeboligen. Hilser på kongen.
5.10- 5.14	7440- 7536	Replikk, bakgrunnslyder som over.	Hilser/ser på "eldstekaren".
5.14- 5.17	7536- 7608	Replikk, bakgrunnslyder som over.	Hilser/ser på to jenter.
5.17- 5.22	7608- 7728	Replikk, bakgrunnslyder som over.	Ser mot kongen (forteller om kona, vet ikke hvor hun er).
		Bakgrunnslyder som over + lyder av nøkkelknippe.	Ser mot kona som kommer gående med nøkkelknippe.
5.22- 5.43	7728- 8232	Replikk, bakgrunnslyder som over.	Kona står foran kongen og kjefter på han.
5.43- 5.59	8232- 8616	Replikk, bakgrunnslyder som over.	Kona snur seg mot gjestene.

5.59- 6.04- 6.08- 6.10- 6.12- 6.15	8616- 8736- 8832- 8880- 8928- 9000	Bakgrunnslyder som over.	Kona går oppover trappa utenfor kongeboligen, og gjestene følger etter.
6.15- 6.20- 6.23	9000- 9120- 9192	Fotsteg, diffuse lyder?	I 2.etg går veien gjennom huset og mot vinduet på andre siden av huset.
6.23- 6.27.5	9192- 9300	Diffuse lyder?	Ser ut av vinduet en stund mot Munkholmen. Kameraet zoomer inn mot havet.
		Replikk (rop om at fienden angriper) + lur (varselhorn).	Panorer kameraet bort fra vinduet og fade til svart eller hvitt.

* Animasjoner inne i borggården:

- løpende og bjeffende hund
- kvinner som henter vann fra brønnen
- soldater som prater og diskuterer
- håndverkere som reparerer murverk
- tjenestefolk som bærer proviant og varer
- et par guttunger som leker
- soldater som patruljerer på ringmuren
- enkelte personer inne i borggården snur seg mot gjesten, de fleste ikke

9.5 Appendix E – Referanseillustrasjoner



Tatt fra 'Men-at-Arms 396 'Medieval Scandinavian Armies (1) 1100-1300' illustrated by Angus McBride. © Osprey Publishing Ltd. (brukt med tillatelse)