

Sammendrag av Bacheloroppgaven

| | | |
|---|------------------------------------|--|
| Tittel: | 3D Visualisering på Web | Nr. : 4 |
| | | Dato : 20.05.09 |
| Deltaker(e): | Jon Espen Kvisler | |
| | Aleksander Stalsberg | |
| Veileder(e): | Anne Kristin Kvitle | |
| Oppdragsgiver: | Visualisere.no | |
| Kontaktperson: | Kristian Botnen | |
| Stikkord (4 stk) | 3d, brukergrensesnitt, design, web | |
| Antall sider: 158 | Antall bilag: 14 | Tilgjengelighet (åpen/konfidensiell): åpen |
| Kort beskrivelse av bacheloroppgaven: | | |
| <p>Prosjektet har gått ut på å lage en portal for opplastning og visning av 3D-modeller på nett, samt design og utforming av det grafiske brukergrensesnittet til 3D-applikasjonen utviklet av oppdragsgiver.</p> <p>Oppgaven har ikke bestått i å utvikle den applikasjonen som skal vise selve 3D-modellen.</p> | | |

Forord

Denne rapporten er en del av bacheloroppgaven ”3D Visualisering på Web”, og markerer avslutningen av en 3-årig bachelorutdanning i Medieteknologi ved Høgskolen i Gjøvik. Prosjektet er blitt gjennomført våren 2009.

Vi valgte et prosjekt fra visualisere.no, der oppgavene var å utvikle en webportal med mulighet for opplastning av 3D-modeller, samt design og utforming av det grafiske brukergrensesnittet til en 3D-applikasjon utviklet av oppdragsgiver.

Oppgaven tar for seg teknologiske og estetiske utfordringer ved det å utvikle en webside tilgjengelig for alle.

I tillegg skal et brukergrensesnitt tilpasset en målgruppe med varierende IT-kompetanse utvikles.

Vi vil rette en takk til:

- Anne Kristin Kvitle, for god veiledning, og hjelp til å holde oss på sporet.
- Kristian Botnen, for god hjelp og bra tilbakemelding som oppdragsgiver.

Gjøvik, 20. mai 2009

Jon Espen Kvisler

Aleksander Stalsberg

Innholdsfortegnelse

| | | |
|----------|---------------------------------|-----------|
| 1 | Innledning | 9 |
| 1.1 | Problemområde og avgrensning | 9 |
| 1.1.1 | Effekt- og resultatmål | 10 |
| 1.2 | Bakgrunn | 10 |
| 1.2.1 | Gruppens bakgrunn og kompetanse | 10 |
| 1.3 | Målgruppe | 11 |
| 1.4 | Prosjektorganisering | 11 |
| 1.4.1 | Ansvarsfordeling | 11 |
| 1.5 | Rammer | 12 |
| 1.5.1 | Arbeidsmengde | 12 |
| 1.6 | Arbeidsmetoder | 12 |
| 1.6.1 | Utviklingsmodell | 12 |
| 1.6.2 | Statusmøter | 13 |
| 1.7 | Organisering av rapport | 14 |
| 1.7.1 | Målgruppe for rapporten | 14 |
| 1.7.2 | Terminologi | 14 |
| 2 | Kravspesifikasjon | 17 |
| 2.1 | Overordende funksjonelle krav | 17 |
| 2.2 | Funksjonelle krav | 17 |
| 2.2.1 | Grafisk profil | 17 |
| 2.2.2 | Websiden | 17 |
| 2.2.3 | 3D applikasjon og GUI | 20 |
| 2.3 | Operasjonelle krav | 20 |
| 2.4 | Krav til brukeren | 20 |
| 2.5 | Use cases | 21 |
| 3 | Utstyr og ressurser | 23 |
| 3.1 | Programvare | 23 |
| 3.2 | Datateknisk utstyr | 23 |

| | | |
|----------|---|-----------|
| 4 | Design | 25 |
| 4.1 | Systemdiagram | 25 |
| 4.2 | Grafisk profil | 26 |
| 4.2.1 | Farger | 26 |
| 4.2.2 | Logo | 26 |
| 4.3 | Webside | 28 |
| 4.3.1 | Klient | 28 |
| 4.3.2 | Tjener | 32 |
| 4.3.3 | Databasestruktur | 35 |
| 4.4 | GUI, 3D-applikasjon | 37 |
| 4.4.1 | Oppgaven fra arbeidsgiver | 37 |
| 4.4.2 | Inspirasjon | 37 |
| 5 | Utvikling | 41 |
| 5.1 | Klient | 41 |
| 5.1.1 | JavaScript og jQuery | 41 |
| 5.1.2 | 960 Grid System | 43 |
| 5.2 | Tjener | 43 |
| 5.2.1 | Rammeverk | 44 |
| 5.2.2 | Kontrolløren | 45 |
| 5.2.3 | Brukersystem | 46 |
| 5.2.4 | Søkefunksjon | 48 |
| 5.2.5 | Databasetilkobling | 48 |
| 5.2.6 | Opplasting av 3D-modell | 49 |
| 5.2.7 | Eksterne skript | 52 |
| 5.3 | GUI | 52 |
| 5.3.1 | Forarbeid og menneskelig interaksjon | 52 |
| 5.3.2 | De første utkastene | 56 |
| 5.3.3 | Forandringer etter hvert | 58 |
| 5.3.4 | Sammensetning av GUI | 59 |
| 6 | Testing | 63 |
| 6.1 | Brukertest | 63 |
| 6.2 | Erfaringer fra brukertesten | 64 |
| 6.3 | Endringer etter test | 64 |
| 6.3.1 | Websiden | 64 |
| 6.3.2 | GUIen | 65 |
| 7 | Endelig produkt | 67 |
| 7.1 | Websiden | 67 |
| 7.2 | GUI og 3D applikasjon | 68 |
| 7.2.1 | Farger | 68 |
| 7.2.2 | GUI for visning av hus og liknende modeller | 68 |
| 7.2.3 | GUI for produktvisning | 70 |
| 7.3 | Applikasjonen | 71 |

| | | |
|-----------|--|-----------|
| 7.4 | Implementering av GUI hos oppdragsgiver..... | 71 |
| 8 | Drøftinger..... | 73 |
| 8.1 | Resultater..... | 73 |
| 8.2 | Erfaringer | 73 |
| 8.3 | Videre arbeid..... | 74 |
| 8.3.1 | Websiden..... | 74 |
| 8.3.2 | GUI..... | 75 |
| 8.3.3 | Tjenesten | 75 |
| 8.4 | Evaluering av gruppa..... | 75 |
| 9 | Konklusjon | 77 |
| 10 | Bibliografi | 79 |
| 11 | Vedlegg | 81 |

1 Innledning

1.1 Problemområde og avgrensning

Det finnes per i dag ingen metode for å laste opp egenproduserte 3D-modeller fra en klientmaskin og vise disse i et interaktivt 3D-miljø via nettleseren.

Oppdragsgiver har gjennomført undersøkelser som viser at det er et behov i markedet for en slik tjeneste, og vi har valgt å være med på utviklingen av dette, som kan kalles et prøveprosjekt.

Etter gjennomført prosjekt skal vi ha produsert en webside som skal fungere som en portal for applikasjonen som viser 3D-modeller, hvor registrerte brukere kan laste opp modeller fra sin klient, og administrere sine modeller. Brukere med høyere tilgangsnivå, administratorer, skal ha tilgang til alle modeller og brukere, samt kunne legge ut nyheter på websiden. Under utviklingen av websiden må det hele tiden tilrettelegges for sikkerhet og rask responstid, da brukermassen er potensielt stor.

Det skal også utvikles et grafisk brukergrensesnitt (heretter kalt GUI (Graphical User Interface)) for denne applikasjonen. GUIen skal gi brukeren en brukervennlig og enkel måte å navigere i et 3D-miljø, samtidig som den må ha samme grafiske profil som websiden.

Dermed kan oppgaven deles i to; webside og GUI for 3D-applikasjonen.

Vi skal ikke utvikle verken selve 3D-fremviseren eller applikasjonen som konverter modellen, da oppdragsgiver står for dette. Prosjektet vårt er avhengig denne 3D-applikasjonen, i den forstand at formålet med websiden er å kunne starte applikasjonen som da skal inneholde GUIen vi skal utvikle. Både websiden og GUIen *må* utvikles etter de retningslinjer oppdragsgiver har satt, for å kunne kobles til 3D-applikasjonen.

Det kunne vært hensiktsmessig å bygge opp websiden mer som et sosialt nettsted, med mulighet for kommentering og diskusjon av 3D-modeller, men dette var aldri målet for oppdragsgiver. Det finnes nok av fora og nettsteder som fyller dette målet. I tillegg hadde vi som prosjektgruppe sannsynligvis ikke klart å utvikle en fullverdig tjeneste i prosjektperioden hvis det var krav til dette.

1.1.1 Effekt- og resultatmål

Effektmål

- Arkitekter og andre som benytter seg av 3D-visualisering kan spare tid og kostnader ved å vise frem sine 3D-modeller over internett til sine klienter.
- Alle kan se sine 3D-modeller i et 3D-miljø i sin nettleser, gratis.
- Visualisere.no blir synligere i markedet.

Resultatmål

- Lage et fungerende nettbasert system åpent for alle som gjør det mulig for registrerte brukere å laste opp 3D-modeller i forskjellige størrelser og beskrive disse, og lagre disse på websiden. Systemet skal være ferdigstilt til prosjektets avslutning.
- Opplastede 3D-modeller må sendes til en Java-applikasjon som konverterer modellen til et format som 3D-fremviseren forstår. Vi må så koble konverterte modeller til websiden, slik at modellene kan vises fra nettleseren til brukeren.
- Tjenesten skal være enkel og brukervennlig, slik at alle i målgruppen kan ta den i bruk uten nevneverdig opplæring. Da tjenesten skal være åpen for alle, må filopplastingen og databasen må sikres mot eventuelle angrep.
- Websiden må være fremtidsrettet og skalerbart, så andre programmerere skal ha mulighet til å sette seg inn i all kode og utvide funksjonalitet.
- Utvikle en GUI som kan tilpasses 3D-applikasjonen fra oppdragsgiver.
- Gi alle brukere en enkel måte å navigere i et interaktivt 3D-miljø.

1.2 Bakgrunn

Vi skal i forbindelse med avslutningen på vår bachelorgrad i medieteknologi utføre et prosjekt i faget *IMT3912 – Bacheloroppgave*, som består av 20 studiepoeng. Etter litt frem og tilbake angående valg av prosjekt endte vi opp med et prosjekt fra Visualisere.no.

Målet til oppdragsgiver er å utvikle et rammeverk for visning av 3D-modeller via internett på tvers av flere disipliner, hvorav de har valgt visualisering av arkitektur og produkter som test-case. Videre kan antall kategorier utvides, og rammeverket for websiden kan gjenbrukes til andre prosjekter, for eksempel et 3D-visningsprosjekt av golfbaner (dette er et annet prosjekt oppdragsgiver holder på med, les på under 4.4.2.1 på side 37) eller en maritim utgave, med visning av båter som hovedoppgave.

3D-applikasjonen er utviklet i Java (lwjgl.org, 2009) webstart, og er en ekstern applikasjon som kjører på topp av nettleser, men startes fra nettleser.

1.2.1 Gruppens bakgrunn og kompetanse

Gruppen består av to studenter som fullfører sin bachelor-grad i medieteknologi ved Høgskolen i Gjøvik (HiG). Det er kunnskapen gruppen har tilegnet seg fra dette studiet som ligger til grunn for denne oppgaven.

Gruppen har noe erfaring med PHP- og JavaScript-programmering, MySQL-databaser, brukertesting, objekt-orientert programmering og 3D-utvikling, da gjennom diverse fag ved HiG og egne erfaringer, men mesteparten av denne kunnskapen var grunnleggende. Vi har også kunnskaper med Adobe Photoshop, HTML og CSS.

Utover dette hadde vi lite erfaring med systemutvikling av et system i denne skalaen og den prosjektorganiseringen som trengs. Vi hadde ikke benyttet oss av en MVC-tankegang tidligere, og hadde heller ingen erfaring med sikring av et offentlig tilgjengelig nettbasert system som tillater filopplasting. På grunn av krav og arbeidsmetoder har vi opparbeidet oss erfaring innenfor PHPDoc, Adobe Illustrator og serverkonfigurasjon.

1.3 Målgruppe

Målgruppen for dette prosjektet har forandret seg igjennom prosjekt perioden. Dette kom frem da oppdragsgiver kom med ønske om at 3D-applikasjonen også skulle dekke produktvisning (se punkt 4.4.1 på side 37). Dette førte til at målgruppen ble utvidet til følgende:

Mottakerne: De som bruker websiden og 3D-applikasjonen for å se på modeller. Dette blir da basis brukeren som kan ha hva som helst av teknisk bakgrunn.

Brukere: De som da tar i bruk systemet og laster opp modeller for å kunne vise disse på nett til sine kunder, og *mottakerne*. Disse brukerne innehar da gjerne en større kompetanse innen IT og 3D enn hva *mottakerne* har.

1.4 Prosjektorganisering

Studenter

Jon Espen Kvisler (06HBMEDTEA)

Aleksander Nielsen Stalsberg (06HBMEDTEA)

Prosjektveileder

Anne Kristin Kvitle.

Oppdragsgiver

www.visualisere.no v/Kristian Botnen

Visualisere.no / Visualisere.org består av Kristian Botnen (Botnen 3D¹), Stian R og Jonas D.

Gruppen er satt sammen rundt nettstedet visualisere.no som kom online i 2008, med mål å bli en ressurs innenfor fagfeltet visualisering. Grunnet gruppens sammensetning har det vært mye hus / eiendom og terreng som har vært i fokus, både på nettstedet og nå også i 3D-applikasjonen som er under utvikling. Kristian er programmerer og driver utvikling og web, Stian er teknisk tegner av profesjon og jobber med 3D visualisering samt CAD 2D for arkitektfirmaer. Jonas D er student på siste året ved Høgskolen i Ålesund. Kristian B og Jonas D har tidligere vært involvert i wsgolfer, interaktive golfbaner i 3D.

1.4.1 Ansvarsfordeling

I begynnelsen av prosjektet fordelte vi arbeidsoppgaver og ansvarsområder til hvert gruppemedlem. Siden prosjektet vårt hadde to klare oppgaver; websiden og GUI-delen, bestemte vi oss for å dele disse to mellom oss. Dette gjorde vi på grunn av interesser, og for å oppfylle de krav stilt av oppgaven innenfor prosjektperioden. Arbeidsoppgaver og arbeidsområder ble da som følger:

Aleksander Stalsberg: Gruppeleder, ansvar for GUI.

Jon Espen Kvisler: Sekretær, ansvar for websiden og grafisk profil

¹ <http://www.botnen.org/>

1.5 Rammer

1.5.1 Arbeidsmengde

En bacheloroppgave har ifølge emnebeskrivelsen² en arbeidsmengde på 20 studiepoeng, som tilsvarer en arbeidsmengde på rundt 30 timer i uka for en student. Dette tilsvarer da prosjektets arbeidsmengde.

1.6 Arbeidsmetoder

1.6.1 Utviklingsmodell

Prosjektet har hatt to hovedmoduler.

- **Modul 1:** Websiden og det tekniske rundt den, samt den grafiske profilen.
- **Modul 2:** Det grafiske brukergrensesnittet, og tilpassning av dette opp mot 3D-applikasjonen utviklet av oppdragsgiver.

Arbeidsfordelingen av dette kan du se ovenfor i punkt 1.4.1. På grunn av at disse to modulene gikk samtidig har det blitt brukt to forskjellige utviklingsmodeller her.

1.6.1.1 Modul 1

Systemet rundt websiden besto av klare delsystemer (grafisk profil, tjener, klient) var det naturlig å dele opp i en inkrementell utviklingsmetode. I begynnelsen av prosjektet ble det utarbeidet en generell kravspesifikasjon, og denne ble mer detaljert beskrevet i begynnelsen av hvert inkrement.

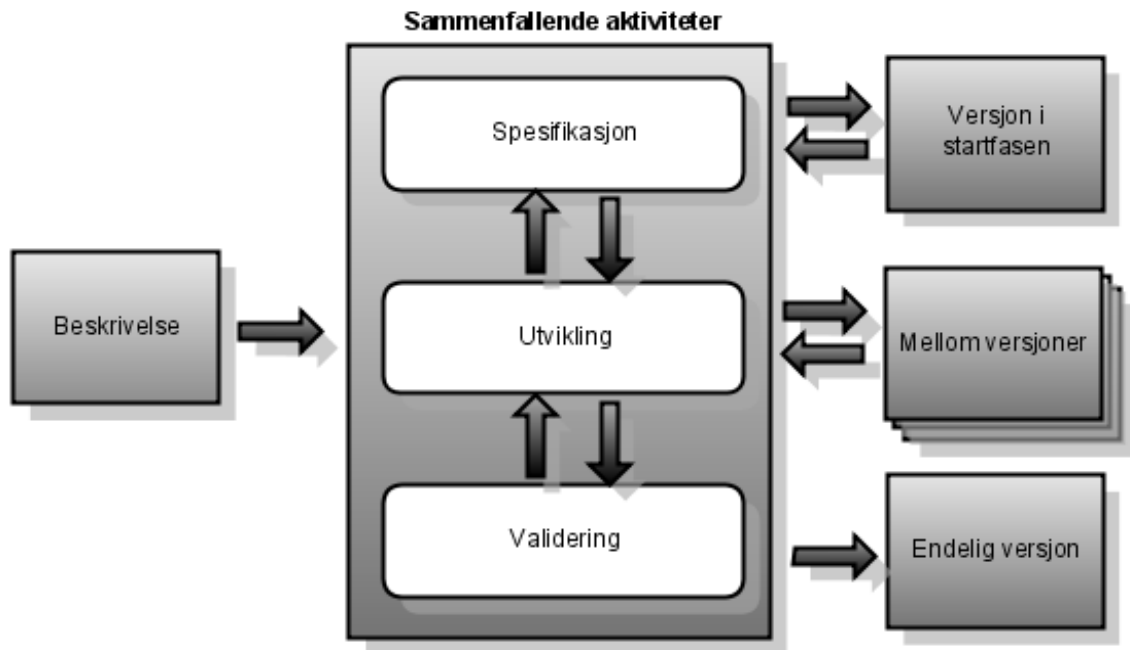
Dermed hadde vi en fleksibel utviklingsmetode som tillot oss å legge til og endre krav underveis, etter hvert som vi tok lærdom av hvert inkrement.

1.6.1.2 Modul 2

Utviklingsprosessen for det grafiske brukergrensesnittet ville ikke fungere på samme måte som i modul 1. Her ville det hele tiden foregå forandringer basert på det som vi hadde tidligere. Den beste utviklingsmodellen vi fant her var den evolusjonære utviklingsmodellen.

2

http://www.hig.no/studiehaandbok/studiehaandboeker/2008_2009/emner/avdeling_for_teknologi_oekonomi_og_ledelse/ing3901_bacheloroppgave_20



Figur 1.1 Evolusjonær utviklingsmodell (oversatt fra Tom Røises engelske versjon).

Som modellen ovenfor viser gikk vi ut ifra en generell beskrivelse (beskrevet i vedlegg B, forprosjektrapporten på side 87). Videre gikk vi da i gang med å utvikle det første utkastet. Dette ble da vår ”versjon i startfasen”. Denne ble så tatt tilbake til utviklingsfasen der vi så over spesifikasjoner, arbeidet videre med den, og prøvde den ut for validering.

Ved flere tilfeller ble ikke utkastet godkjent, eller det passet ikke med de krav som var stilt. Så vi går tilbake til utviklingsstadiet, og så over hva som kunne gjøres annerledes eller bedre. Dette blir fortsatt og testet til vi til slutt finner en versjon vi er fornøyd med og som oppfyller kravene til kravspesifikasjonen.

Denne utviklingsmodellen passet veldig godt til denne modulen, da vi kontinuerlig kom med skisser og utkast til hvordan en GUI kunne se ut, for så å ta disse tilbake til videre utvikling og tilpassning.

1.6.2 Statusmøter

Veileder

Til å begynne var det statusmøter annenhver uke sammen med veileder.

Det som ble gjennomgått på disse møtene var:

- Hva som hadde blitt gjort siden forrige statusmøte.
- Daværende status på prosjektet.
- Og vi skulle redegjøre for hva som var planen fremover for prosjektet.

Vi har valgt og ikke levere en statusrapport hver uke, da gruppen var så liten og kommunikasjon ikke ble sett på som noe problem.

De eneste innleveringene underveis har vært de obligatoriske statusrapportene hver måned til veileder.

Oppdragsgiver

Siden oppdragsgiver befinner seg på andre siden av landet, Bergen, har alle statusmøter foregått via Skype.

Vi har ikke hatt noen faste tidspunkter for statusmøter med oppdragsgiver, da dette har blitt avtalt fortløpende.

1.7 Organisering av rapport

1.7.1 Målgruppe for rapporten

Målgruppen for rapporten er i hovedsak veileder, oppdragsgiver og den eksterne sensoren som skal evaluere prosjektet. Rapporten bygger på at de som skal lese rapporten har en viss innsikt i de teknologiene som er blitt tatt i bruk i prosjektet, og vil derfor være til tider relativt teknisk.

Sensor skal kunne bruke rapporten med vedlegg for å kunne evaluere prosjektets omfang og kvalitet.

All kode er markert med `courier` new i skriftstørrelse 10.

1.7.2 Terminologi

I rapporten blir det brukt en del ord og begreper, og vi har valgt å definere de vi mener kan oppfattes som uklare her.

- Gjennom prosjektet omtales programvaren som leveres oppdragsgiveren, altså applikasjonen som skal brukes for å vise 3D-modellene og applikasjonen som konverterer opplastede modeller til riktig format som **3D-applikasjonen** (eventuelt applikasjonen hvis dette er underforstått).
- 3D-modellen blir noen steder omtalt som **modellen**.
- Med **websiden** menes hele det nettbaserte systemet, da både klientside, serverside og database.
- Når det snakkes om **klient** og **tjener** i sammenheng med websiden, er dette henholdsvis forforskninger av ordene *frontend* og *backend*. Under klient faller alt som prosesseres av klienten, altså på brukerens lokale datamaskin. Dette er da hovedsakelig HTML, CSS og JavaScript, og dette er kode som alle brukere får full tilgang til. Med tjener menes da all kode som prosesseres på tjeneren og sendes til klienten. Dette er hovedsakelig PHP.
- Selv om all HTML-kode skrives med XHTML, beskriver vi denne som **HTML**.
- Med **ikke-påtrengende JavaScript** menes såkalt "*unobtrusive*" JavaScript. Dette er "et sett med retningslinjer hvis mål er samvittighetsfull bruk av JavaScript" (Johansen, Ikke-påtrengende Javascript, 2007). Hovedsakelig betyr det at JavaScript aldri skal legges rett i dokumentet, og skal være et tillegg, og ikke funksjonalitet man er avhengig av.
- "**Semantisk HTML** handler om å bruke HTML-taggene for den meningen, eller den semantikken, de gir innholdet" (Johansen, Semantisk HTML, 2009). For eksempel skal alle overskrifter plasseres i `h2`-tagger.
- **Iden/iden/id** brukes som en forkortelse for identifikasjon/identifikasjonen.
- Med **detaljvisning av modell** mener vi visning av modellen på websiden, hvor man har mulighet til å starte 3D-applikasjonen, og ikke selve 3D-visningen.
- **Minimap** (minikart), ett lite kart som oftest brukes i applikasjoner der man beveger seg rundt og man samtidig bør ha oversikt over hvor man befinner seg.

- **Touchpad** er den musen som følger med en bærbar PC. Ett lite område gjerne rett nedenfor tastaturet som er trykkfølsomt og vil da styre musa på skjermen.
- **Joystick**, en styrespak for å kunne navigere 360 grader.
- **Superbruker** brukes her i sammenheng med GUI-utviklingen, og beskriver en bruker som har brukt applikasjonen før, og vet om det som er av hurtigtaster.
- **Viewport**, hovedvindu av applikasjonen som da viser selve 3D-modellen. Det er her selve visningen av 3D-modellene vil foregå.

2 Kravspesifikasjon

For å klargjøre hva som måtte gjøres tidlig i prosjektet, utviklet vi *Use cases*. Vi så på dette som et godt verktøy for å kartlegge hva som måtte gjøres og konkretisere kravene til prosjektet.

2.1 Overordnede funksjonelle krav

Vi skal lage en dynamisk webside hvor brukere skal kunne utføre forskjellige handlinger basert på tilgangsnivå. Siden dette systemet blir åpent tilgjengelig for alle, må sikkerhet være et viktig punkt.

Alle skal ha tilgang til visning av 3D-modeller, uansett kunnskapsnivå. Derfor må websiden være tilgjengelig og brukervennlig. 3D-applikasjonen skal også være enkel i bruk, derfor må GUIen være enkel og lett å forstå, selv for de uten nevneverdig erfaring fra 3D-modellering. Denne GUIen skal være selvforklarende og intuitiv, og også hele tiden fortelle brukeren hvor han befinner seg i en 3D modell. Siden 3D-applikasjonen åpnes i et nytt vindu, er det viktig at GUIens farger minner om de på websiden. Dermed må det utvikles en grafisk profil, som blir brukt som et tema både i websiden og i GUIen.

Registrerte brukere kan laste opp 3D-modeller fra sin klientmaskin, og vi må dermed lage en funksjon hos tjeneren som tar i mot disse filene og sender til 3D-applikasjonen for konvertering, og legger de i korrekt mappe. Det er viktig at brukeren får tilbakemeldinger hele tiden om hva som skjer under opplastingsprosessen, for ikke å skape forvirring.

Videre skal administratorer ha tilgang til å slette alle modeller, legge inn nyheter og slette brukere.

All kode skal dokumenteres for videre utvikling, og slik at andre skal ha en mulighet til å sette seg inn i systemet uten opplæring.

2.2 Funksjonelle krav

2.2.1 Grafisk profil

I forhold til den grafiske profilen har vi ingen krav bortsett fra at den skal inneholde elementer som viser at tjenesten omhandler 3D-modellering. I tillegg må profilen bli brukt konsekvent i både GUIen og websiden, slik at det hersker ingen tvil for brukeren at vinduet som kommer opp ved visning av en 3D-modell tilhører tjenesten.

Logo som utvikles skal produseres i vektorformat, for skalering uten kvalitetstap. Den skal også fungere både i fargeversjon og i svart-hvitt, og følge fargene satt i den grafiske profilen.

2.2.2 Websiden

Hele websiden skal være klar for lansering ved avslutningen av prosjektet, og oppfylle de krav listet opp nedenfor.

2.2.2.1 Rammeverk

Websiden skal ha et rammeverk for inkludering av sider. Dette vil bli en blanding av dynamiske og statiske sider. Selve koden bør struktureres slik at presentasjonsfilene inneholder et minimum av PHP-kode, og holdes adskilt fra klasser og funksjoner.

All tekst brukt på de dynamiske sidene skal lagres i en ekstern språkfil, slik at språk for websiden kan byttes ut for fremtiden. I tillegg skal alle globale variabler lagres i en konfigurasjonsfil.

2.2.2.2 Tilgjengelighet og brukervennlighet

All HTML-kode skal kodes så semantisk som mulig, og innhold skal skilles fra utseende.

Vi skal benytte oss av dokumenttypen XHTML 1.0 Transitional, og alle sider skal validere i W3Cs HTML- og CSS-valideringstjeneste³.

Websiden skal være så brukervennlig som mulig, da målgruppen består blant annet av mennesker uten tung IT-kompetanse.

JavaScript skal, til den grad det er mulig, være ikke-påtrengende og implementeres slik at det ikke hindrer brukere uten JavaScript.

Websiden skal være tilgjengelig i nettleserne listet opp i Yahoo!s A Grade-liste⁴. Siden 3D-applikasjonen per i dag kun støtter Windows-plattformen, er det ikke nødvendig at plattformene Mac og Linux er støttet, selv om de sannsynligvis blir det da vi må skrive kode basert på W3Cs standarder.

Logo (med "Hjem"-link), hovednavigasjon og søk skal alltid ligge på samme plass, slik at brukeren har et referansepunkt og dermed minske sjansene for at han roter seg bort.

I og med at vi skal ha et CAPTCHA-script under brukerregistreringen må dette gjøres så brukervennlig som mulig, og brukere må få mulighet til å be om et nytt bilde hvis de ikke forstår innholdet.

2.2.2.3 Brukersystem

Systemet skal ha tre forskjellige tilgangsnivåer:

1. Uregistrerte brukere
2. Registrerte brukere
3. Administratorer

Dermed må vi ha et brukersystem, hvor brukere kan logge seg sikkert inn og ut. For at brukere skal slippe å logge seg inn hver gang, skal det lages en "Husk meg"-funksjon som lagrer brukernavnet og passordet på brukerens klient.

Registrerte brukere skal ha mulighet til å laste opp 3D-modeller, endre informasjon om seg selv, og slette sine opplastede modeller. Administratorer skal i tillegg ha mulighet til å slette alle modeller lastet opp, legge til og slette nyheter og slette brukere.

³ <http://validator.w3.org/>

⁴ Yahoo!s A Grade-liste: <http://developer.yahoo.com/yui/articles/gbs/index.html>

Ved registrering av konto skal alle passord krypteres ved lagring. For å verifisere at brukere som registrerer seg faktisk er mennesker og ikke automatiserte spamroboter skal kontoregistreringen ha et såkalt CAPTCHA-script. Her får brukeren opp et bilde med forvrengt tekst, som må gjengis i et tekstfelt.

2.2.2.4 Opplasting

Opplasting av en 3D-modell skal skje gjennom flere steg, hvor brukeren skal ha mulighet til å bevege seg frem og tilbake uten å miste informasjon tastet inn. Modellen skal ikke publiseres før brukeren har fått en mulighet til å lese igjennom det han har tastet inn, og eksplisitt klikket ”Publiser”.

I og med at en modell ikke trenger å ha teksturer, men kan ha flere, må det legges opp til at brukeren kan laste opp et x antall teksturer. For at det skal være mulig å sortere modeller, må det lagres tidspunktet for når de ble lastet opp, hvilken bruker som lastet den opp og hvilken kategori den tilhører. I tillegg må man kunne beskrive sin modell med en tittel, en beskrivende tekst, tags og et miniatyrbilde. Siden noen brukere ikke har interesse av å dele modellen sin med resten av verden, skal det være mulig å passordbeskytte og velge hvilke brukere som skal ha tilgang til modellene. I tillegg skal man ha mulighet til å angi en annen opphavsmann hvis man selv ikke sitter med opphavsrettighetene til modellen man laster opp.

Alt informasjon om modellene skal lagres i databasen. Selve modellene skal lagres utenfor www-mappen, for å forhindre direkte tilgang modellene. Det må også sjekkes at filer som lastes opp er av korrekt type, har riktig filendelse, ikke er større enn lovlig filstørrelse og ikke inneholder ulovlige tegn i filnavnet. Alle tegn utenom det engelske alfabetet (26 tegn), tall, binde- og understreker eller to punktum eller flere på rad er ulovlige, da det kan skape problemer for 3D-applikasjonen. Det skal heller ikke være lov med doble under- eller bindestreker. Eventuelle mellomrom skal erstattes med understrek.

2.2.2.5 Visning av modeller

Med visning av modeller, menes visning av modeller på websiden, og ikke i 3D-applikasjonen.

For at brukere skal ha en mulighet til å finne frem til sine egne og andres modeller, må det være mulig å bla igjennom modeller allerede opplastet. Disse modellene skal kunne sorteres på kategorier, antall visninger og når de er lastet opp. Brukere skal kunne vise alle modeller lastet opp siden sist besøk. I visningen av flere modeller skal modellens tittel, miniatyrbilde og navnet til brukeren som lastet opp modellen vises. I tillegg kan modellene merkes med ingen eller flere tags, som igjen danner såkalte ”tag clouds”.

Inne på detaljvisningen av en modell skal modellens tittel, beskrivelse, miniatyrbilde, tags, antall visninger, opplastingsdato og brukeren tilknyttet modellen vises. I tillegg må det være mulig å starte selve visningen av modellen i 3D-applikasjonen. Hvis modellen er passordbeskyttet eller man ikke har tilgang til modellen, skal man ikke få opp denne detaljvisningen.

2.2.2.6 Database

Databasearkitekturen skal være normalisert og inneholde så lite redundante data som mulig. Primær- og fremmednøkler skal benyttes.

For å gjøre det mulig å bytte databasesystem senere skal vi benytte oss av et såkalt databaseabstraksjonslag, som tar seg av tilkobling og spørringer mot databasen.

Alle databasespørringer med data manipulert av bruker skal gjøres klar og fjerne ondsinnet kode av dette abstraksjonslaget for å forhindre tilfeller av SQL-injection⁵.

2.2.3 3D-applikasjon og GUI

Selve applikasjonen og GUIen må oppfylle følgende krav:

- Tilby muligheten til å bevege seg fritt i ett 360 graders 3D-miljø.
- Inneholde ett grafisk brukergrensesnitt for visning av hus eller liknende.
- Gi mulighet for å bytte mellom wireframe-visning og vanlig visning av 3D-modeller.
- Gi en mulighet for brukere å ta såkalte *screenshots* av 3D-modellen for å kunne lagre disse i ett galleri for senere bruk.
- Gi nok informasjon til brukeren om hvor man befinner seg i en 3D-modell, både når det gjelder en 3D-modell av et hus, og under en produktvisning.
- Alt skal utformes og implementeres inn i et vindu på 640 x 480 piksler.

I tillegg til dette var det noen krav fra oppdragsgiver som dukket opp litt senere i arbeidsprosessen, og derfor ble med som et alternativt krav.

- Utforme et grafisk brukergrensesnitt for en produktorientert visning, der modellen beveges, og ikke kamera.

For å løse de kravene som nevnt ovenfor må GUIen tilrettelegges for å gi brukeren full mulighet til å bevege seg i alle retninger. Dette løses ved å utvikle knapper som skal kunne illustrere dette uten at brukeren skal måtte behøve å lese en brukermanual.

Det samme gjelder både wireframe-visning, og knappen for screenshots. Disse skal være selvforklarende, og brukeren skal kunne ta i bruk disse relativt lett uten å måtte bruke tid på å sette seg inn i en brukermanual.

For å kunne oppfylle kravet om og hele tiden kunne informere bruker om hvor man er, må vi gi brukeren ett slags minimap, eller en informerende modell om hvor han/hun befinner seg.

2.3 Operasjonelle krav

Tjeneren hvor websiden skal ligge må ha støtte for et dynamisk programmeringsspråk, og ha en database installert, da helst PHP5 og MySQL5.

2.4 Krav til brukeren

For å ta i bruk vår tjeneste i sin helhet må brukeren ha en internettkobling og en nettleser som er listet under Yahoo!s A Grade-liste. Videre kreves det Flash, JavaScript og Java for visse funksjoner.

⁵ http://en.wikipedia.org/wiki/SQL_injection

2.5 Use cases

Ut i fra kravene ovenfor utviklet vi use cases. Disse er vist i Figur 2.1 på side 21. Vi utviklet overordnede use case-beskrivelser for utvalg av disse, og en detaljert use case-beskrivelse. Disse finnes i vedlegg C på side 97. Dette gjorde vi for å se systemet fra brukerens synsvinkel, og som et supplement til de funksjonelle kravene.

Merk at alle use cases ble utviklet forholdsvis tidlig i prosessen, og variasjoner i forhold til endelig implementering kan forekomme.



Figur 2.1: Use case-diagram

3 Utstyr og ressurser

Nedenfor følger en liste over programvare og teknisk utstyr vi har tatt i bruk i løpet av prosjektperioden.

3.1 Programvare

| | |
|-----------------------|---|
| Adobe Illustrator | Vektorbasert illustrasjonsprogram. http://adobe.com/illustrator |
| Adobe Photoshop | Bilderedigering. http://adobe.com/photoshop |
| Color Scheme Designer | Interaktivt fargehjul. http://colorshemedesigner.com/ |
| DropBox | Online backup av systemet. https://www.getdropbox.com/ |
| Firebug | Mozilla Firefox-tillegg, debugging av HTML, CSS, JS. http://getfirebug.com/ |
| Gliffy | Online program for diagrammer og lignende. http://www.gliffy.com |
| Google Docs | Samskriving og deling av dokumenter underveis i prosjektet. http://docs.google.com |
| IETester | Testing av websiden i Internet Explorer 6. http://www.my-debugbar.com/wiki/IETester/HomePage |
| MS PowerPoint | Prototype av 3D-applikasjon, brukertesting. http://microsoft.com |
| MS Word | Rapportskriving, http://microsoft.com |
| phpMyAdmin | MySQL i et grafisk brukergrensesnitt. http://www.phpmyadmin.net |
| Skype | Muntlig kommunikasjon over internett. http://skype.com |
| Web-developer toolbar | Ytelsesmåling, generell webutvikling. https://addons.mozilla.org/en-US/firefox/addon/60 |
| YSlow | Utvidelse av Firebug, ytelsesmåling. http://developer.yahoo.com/yslow/ |
| Zend Studio | PHP-programmering og FTP-tilkobling. http://zend.com |

3.2 Datateknisk utstyr

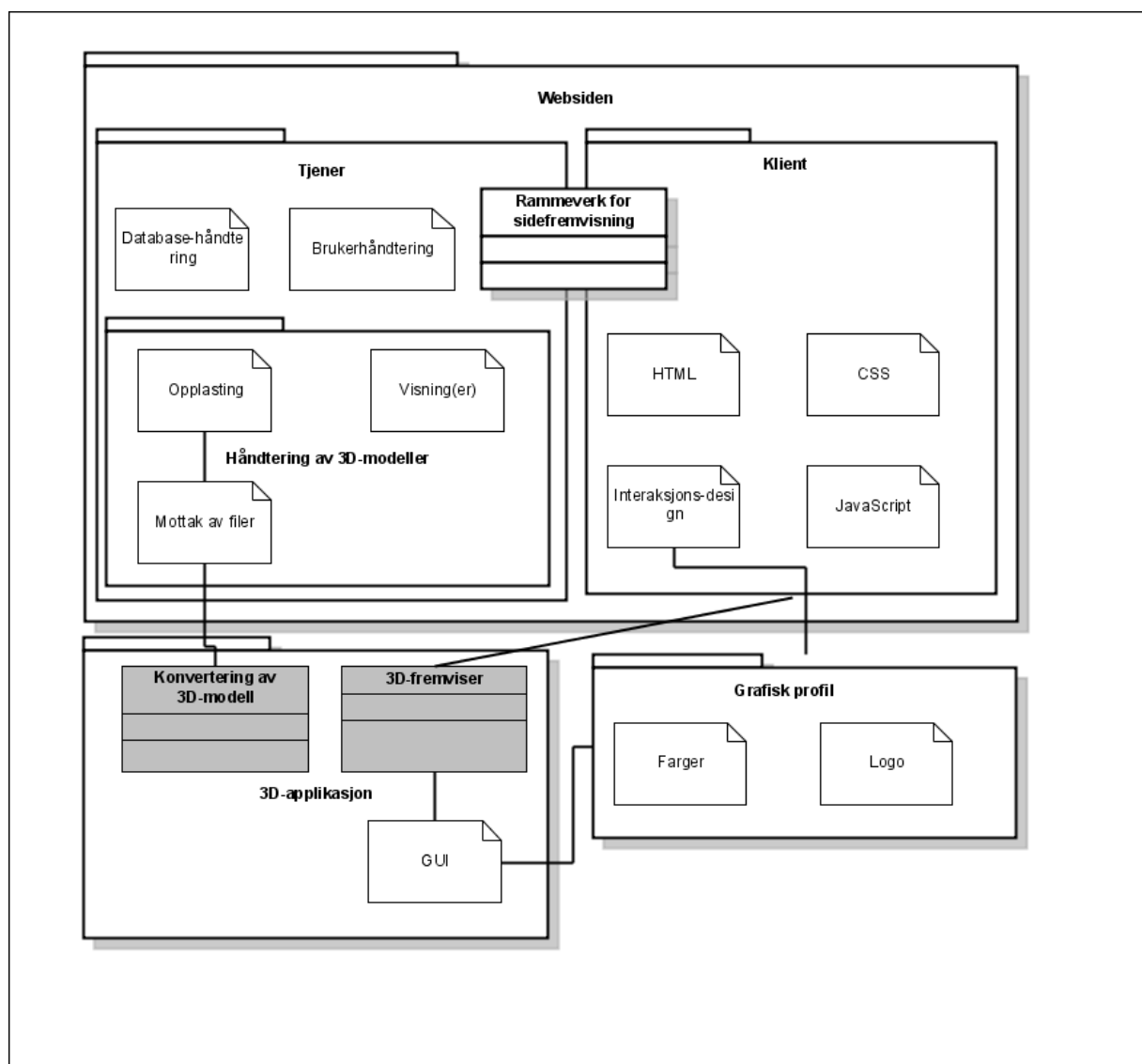
- 2 bærbare og en stasjonær datamaskin
- Arbeidsstasjoner
- Skanner
- Webhotell fra Domeneshop

4 Design

4.1 Systemdiagram

For å vise hele systemet, og vise hva vi skulle utvikle og hva oppdragsgiver skulle utvikle, laget vi et grovt systemdiagram. Selv om det finnes flere relasjoner, og alle elementene sannsynligvis er linket til hverandre på et eller annet hvis, valgte vi kun å lage relasjoner mellom de elementene som var veldig tett linket, og var avhengig av hverandre.

Elementene som er grå skal da oppdragsgiver produsere.



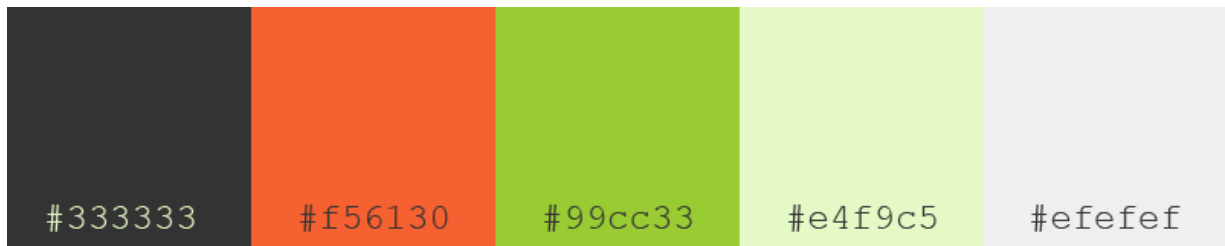
Figur 4.1: Grovt systemdiagram

Systemet er da delt i tre, websiden, 3D-applikasjonen og den grafiske profilen. Websiden er igjen delt i to, tjener og klient.

4.2 Grafisk profil

Det var viktig å ha en grafisk profil, både på websiden og i brukergrensesnittet for 3D-applikasjonen. Siden 3D-applikasjonen åpnes i et nytt vindu ved visning av 3D-modeller, må det komme tydelig frem at 3D-applikasjonen er tilknyttet websiden. En gjennomført grafisk profil med en forhåndsbestemt fargepalett, en gjenkjennbar logo og typografi er dermed et godt virkemiddel for å oppnå dette.

4.2.1 Farger



Figur 4.2: Farger for den grafiske profilen

Vi fikk ingen spesielle føringer på fargevalg for brukergrensesnittet, verken på web eller i 3D-applikasjonen, bortsett fra at de skulle være like. Dermed satte vi i gang, og det eneste målet vårt var at vi ikke skulle ende opp med en grå og kjedelig palett, for dette synes vi (ofte) er det motsatte av hva man ser i 3D-modeller. Dagens 3D-modeller utnytter skjermer som kan vise millioner av farger og nyanser, med nok prosessorkraft til å rendere dem. Vi valgte derfor sterke, livlige farger som kontrastfarger, mens lyse og duse farger blir brukt som bakgrunner. Oransje- og grønnfargen er (så å si) komplementærfarger og utfyller hverandre og skaper harmoni mellom hverandre.

4.2.2 Logo

Vi ville skape en enkel logo som symboliserte og gjenspeilet produktet til oppdragsgiver. Den måtte være gjenkjennbar og enkel, men samtidig original. For å søke inspirasjon ble ulike kilder oppsøkt, blant annet Logopond⁶ og Autodesk⁷. Her så vi etter ulike former og konsepter, og tegneblokken ble funnet frem.

Skisseringen var en viktig del av prosessen, hvor vi fikk rablet ned tanker og ideer på en kjapp måte, og ikke var bundet av teknologiske rammer. Selv om tegneferdighetene ikke var helt på topp var det heller ikke poenget i denne prosessen, da vi kun skulle finne et konsept. Det ble tegnet tre-fire forskjellige konsepter, og til slutt valgte vi et som vi følte representerte produktet godt. Dette var en kube i tre dimensjoner, med en V (for Visualisere) plassert inni. Vi vurderte å plassere et annet objekt i kubens, blant annet et hus eller et annet 3D-objekt (for eksempel den berømte ”Teapot”⁸), men valgte altså ”V” til slutt. Grunnen til dette var at logoen må være like tydelig ved store som små størrelser, og med andre objekter kunne det bli vanskelig å tyde logoen.

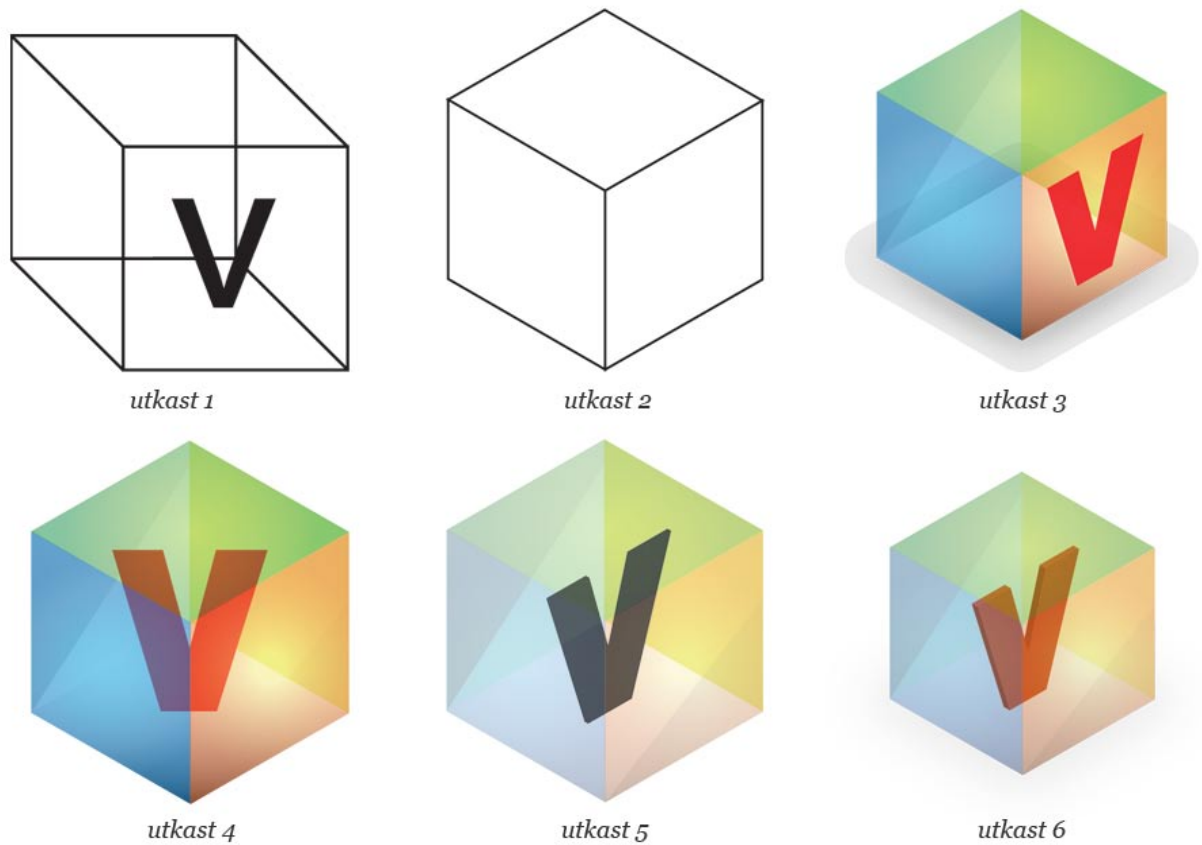
Det neste steget var å lage en wireframe-modell av konseptet, slik at vi kunne presentere det for oppdragsgiver og høste tilbakemeldinger. Dette ble gjort Adobe Illustrator. Vi valgte dette programmet fordi støtten for vektorgrafikk var større enn i Adobe Photoshop, selv om vi hadde betydelig mer erfaring med sistnevnte. Det første utkastet av logoen ble utarbeidet, og oversendt til oppdragsgiver sammen med våre tanker rundt logoen. Vi følte at logoen, i tillegg til å representere

⁶ <http://www.logopond.com> – galleri/showcase for logoer

⁷ <http://www.autodesk.com> – Utvikleren bak 3ds Max, et av de mest brukte 3D-programmene

⁸ http://en.wikipedia.org/wiki/Utah_teapot – En 3D-modell som ofte blir brukt som referanse

3D-modellering, kunne presenteres på flere måter. Den kunne for eksempel animeres ved å rotere 360° rundt sin egen akse, og dermed hadde man en spinnende kube som lett kunne animeres ved behov. Et annet poeng var at siden vi jobbet ut i fra at "Visualisere" kun var en arbeidstittel så måtte det være enkelt å tilpasse logoen til det eventuelle nye navnet, noe man enkelt kan gjøre ved å bytte ut bokstaven "V" med en annen.



Figur 4.3: Logoutkast

Etter positiv tilbakemelding satte vi i gang med det videre arbeidet, altså typografi og fargelegging. Det skal også nevnes at det første utkastet besto av en kube sett forfra, og så veldig klumsete ut. Derfor valgte vi å tegne neste utkast i det isometriske perspektivet. Dette er et perspektiv som blir mye brukt i tekniske tegninger, og betyr kort fortalt å vise et tredimensjonalt objekt i to dimensjoner, hvor de tre koordinataksene er i samme skala og vinklene mellom alle aksene er 120°. Et problem som dukket opp med denne tilnærmingen var at kuben kunne oppfattes som en sekskant, og for å forhindre dette lekte vi oss mye med skygger, lyssetting og fargelegging. Derfor la vi inn en tredje farge, nemlig blå (som ikke er en del av fargepaletten i den grafiske profilen), for å markere de tre sidene av kuben som synes med hver sin farge.

Vi hadde egentlig bestemt oss for utkast 4, men etter at oppdragsgiver meldte at "V"-en hadde lite kontrast, endte vi opp med utkast 6. Her la vi også på en subtil skygge under kuben for å gi et inntrykk av at det var en kube som sto plassert på noe, og ikke svevde i luften, og "V"-en ble satt i isometrisk format.

Typografimessig falt valget ganske kjapt på Neue Helvetica, en grotesk skrifttype (altså uten serifer), som er en veldig utbredt skrifttype. Denne ble lettere modifisert for å skape et uttrykk og skille seg ut, ved at tverrstreken i A-en ble fjernet og den dermed minnet om V-en i logoen satt opp ned. Alle bokstavene er satt til versaler, med en høyere sperring (tracking). Dette skaper et ”teknologisk” og futurisk inntrykk, som samtidig ser seriøst ut.



VISUALISERE

Figur 4.4: Endelig utkast

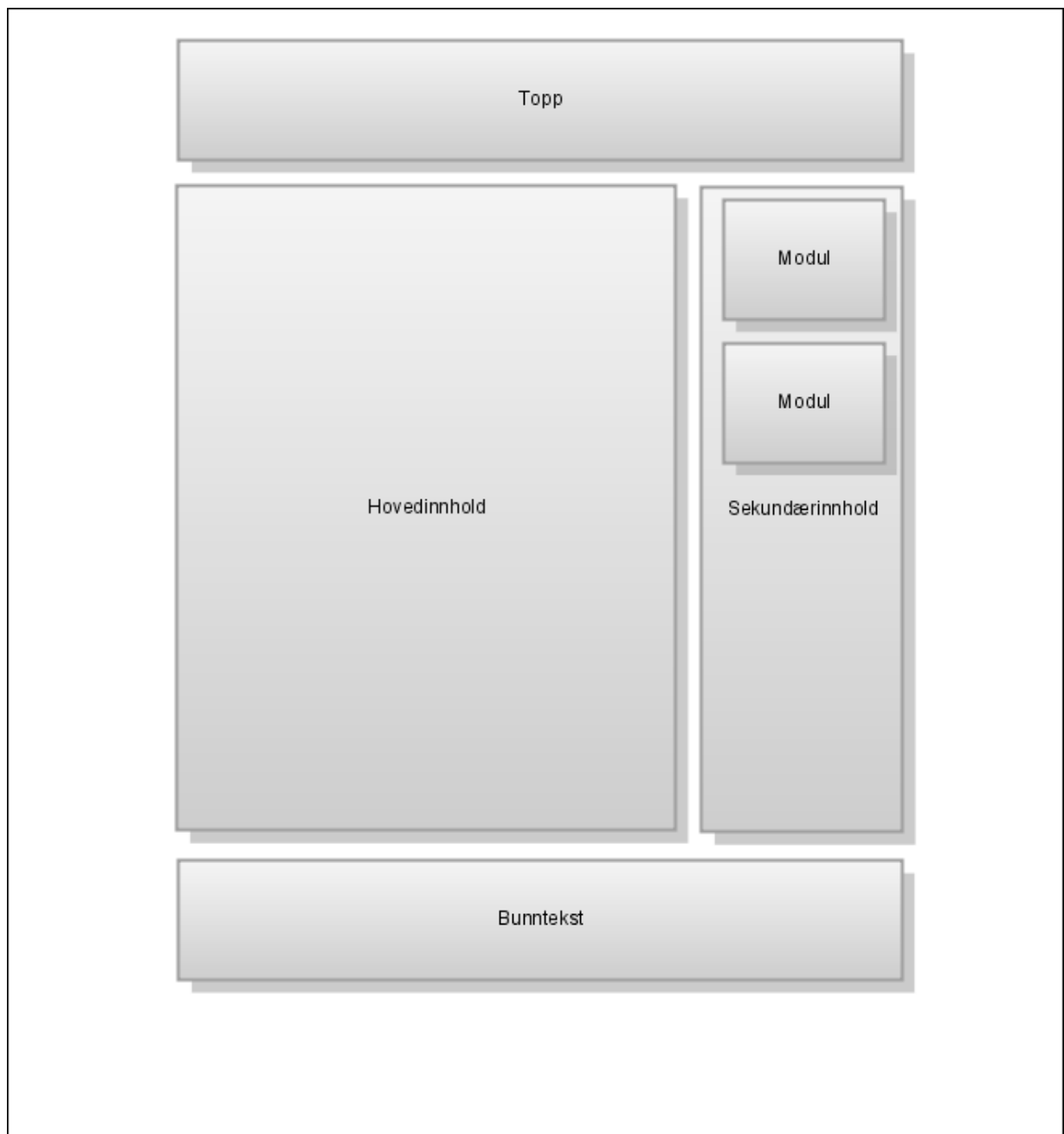
4.3 Webside

4.3.1 Klient

Det første vi gjorde var å tegne opp utkast til sidestruktur, som ivaretok kravene fra oppdragsgiver og presenterte informasjonen på en oversiktlig måte. Vi valgte en to kolonnens layout, med en header og bunntekst (Figur 4.5 nedenfor). Toppen, eller *headeren*, inneholder da en logo, hovednavigasjon, navigasjon for brukerkonto og søkefunksjon. Dette er da innhold som holder seg statisk fra alle sider, slik at brukeren alltid skal kjenne seg igjen, og enkelt navigere mellom de ulike undersidene.

Hovedinnholdet viser innholdet på den siden man befinner seg på, mens sekundærinnholdet typisk inneholder en eller flere moduler. Dette kan være generelle moduler som går igjen på flere sider eller spesielle moduler tilknyttet hovedinnholdet på den nåværende siden.

Bunnteksten er som toppen statisk, og går igjen på alle sider. Her skal det være en såkalt supernavigasjon, altså linker til alle undersider på websiden, også de som ikke finnes i toppnavigasjonen.



Figur 4.5: Struktur for layout

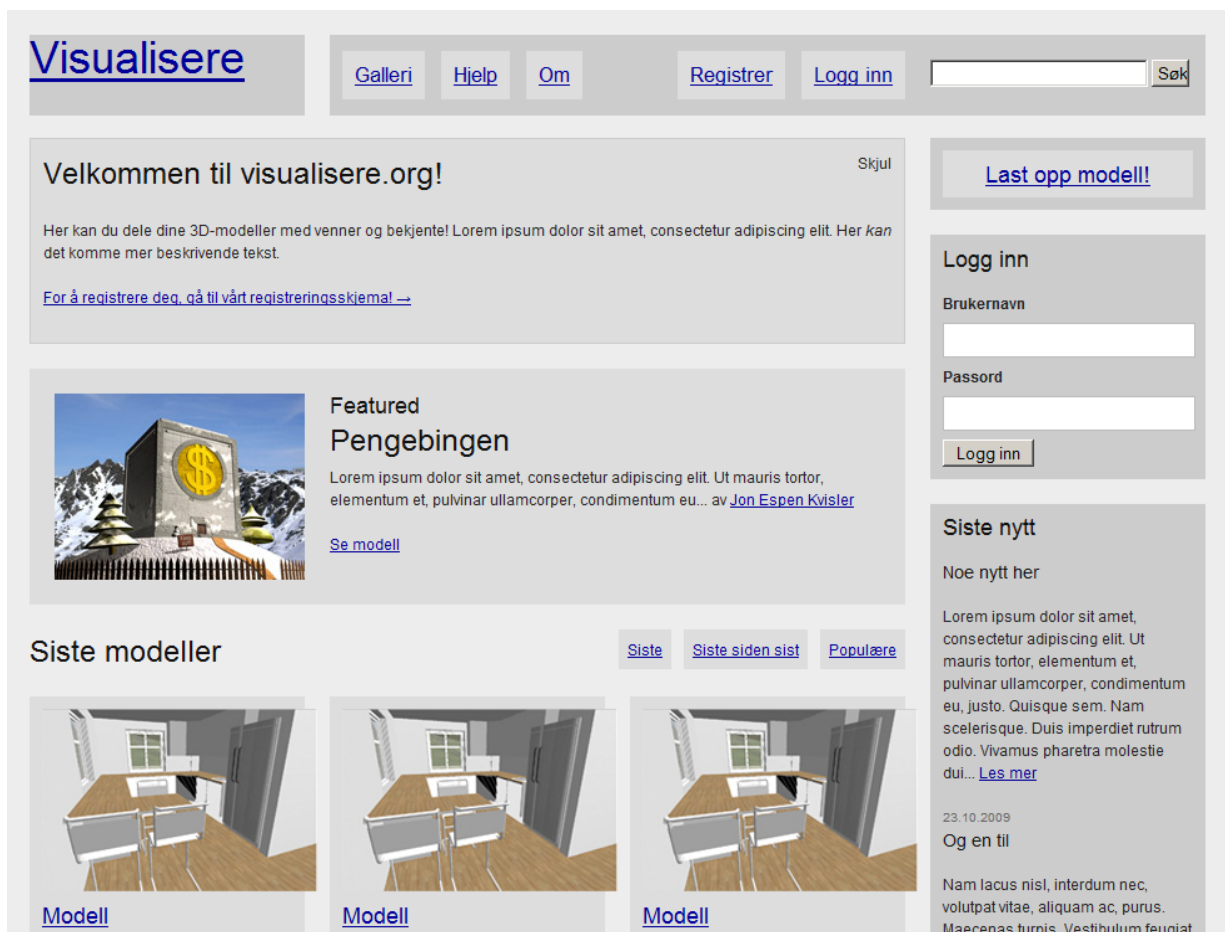
Etter at standardlayoutet var satt, satt vi i gang med å tegne opp en generell mal, og de mest spesielle sidene som skiller seg ut fra den generelle. Disse spesielle sidene inkluderte forsiden, gallerivisning for modeller, detaljvisning for modell, opplasting av modell og registrering av brukerkonto. Den viktigste av disse var forsiden, siden denne sannsynligvis får flest visninger. Dermed ville vi vise det viktigste siden har å tilby, nemlig hvordan man laster opp modeller og en liten introduksjon til tjenesten, og modeller som allerede er lastet opp.

Da de fleste sidene var grovt skissert, gikk vi over til utvikling av en HTML/CSS wireframe og den første prototypen. For å skape fortløp i denne prosessen, bestemte vi oss for å benytte oss av et ferdigutviklet grid-system, også kalt CSS-rammeverk, ved navn 960 Grid System (960.gs) Les mer om dette under 5.1.2 på side 43. Eksempel på bruk av grid-systemet kan sees i vedlegg I på side 141. Her

har vi lagt et gjennomsiktig rutenett over det endelige designet, og man kan enkelt se hvilke elementer som er på linje.

Siden vi hadde valgt en luftig og lett fargeprofil, var det viktig for oss å skape en layout med nok whitespace mellom elementene i komposisjonen. Bruken av et CSS-rammeverk forenklet denne prosessen, da det allerede er definert et passe mellomrom mellom elementene vertikalt. Men horisontalt må vi selv definere mellomrom, og siden det var brukt 20 piksler vertikalt i CSS-rammeverket ble det 20 piksler horisontalt.

Figur 4.6 viser den første prototypen (bildet er kuttet slik at det ikke tar opp for mye plass, HTML-versjonen finnes på cd) hvor farger ikke var prioritert. Dette er forsiden, og viser header, en velkomstbeskjed som kan skjules, en utvalgt modell, og de siste modellene lastet opp. I sekundærinneholdet kan man laste opp modell, logge seg inn og de siste nyhetene fra utviklerne vises.

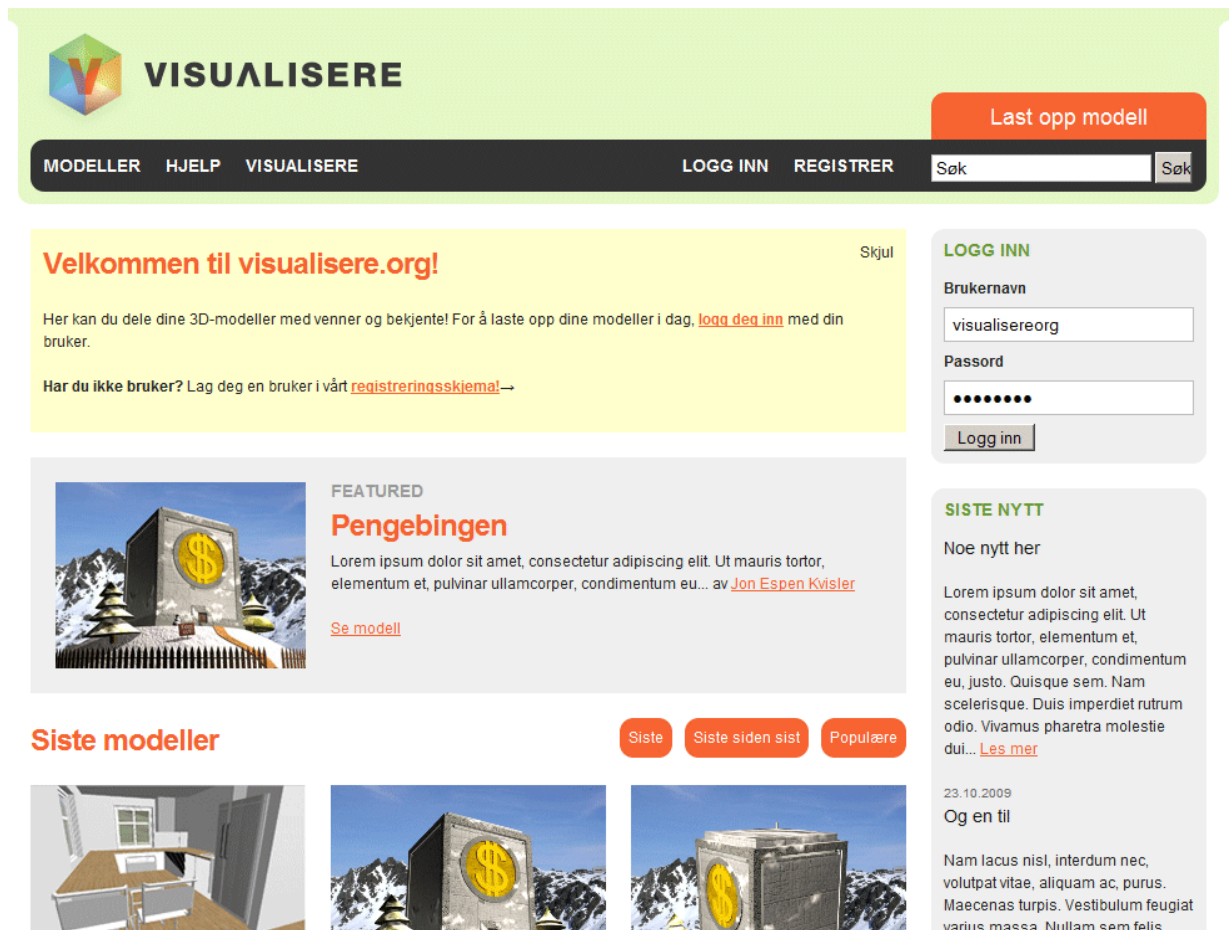


Figur 4.6: Wireframe av prototype

Til den neste prototypen la vi på farger og utviklet en del undersider, slik at en brukertest kunne gjennomføres. Den generelle typografien ble bestemt, da skrifter, skriftstørrelser og linjehøyde for de ulike HTML-elementene. Det ble valgt Helvetica og Arial som standardskrift, med en standard sans-serif-skrift som sistevalg hvis ingen av de andre skriftene fantes. Alle skriftstørrelser og linjehøyder er angitt i em, slik at de alltid er i forhold til skriftstørrelsen i body. Dermed kan skriftstørrelsen enkelt endres med å angi font-size i body.

Stilsettet er delt opp i forskjellige filer, og den første som inkluderes er et `reset-stilsett`⁹ hvis formål er å redusere ulikhetene mellom de forskjellige nettleserne ved å fjerne alle standard marger, linjehøyder og skriftstørrelsene. Dette må da defineres igjen, og filene `text.css`, `forms.css`, og `colors.css` brukes til å angi tekst-, skjema- og fargeformatering. `960.css` er filen for CSS-rammeverket, mens `css.css` er stilsettet hvor all formatering i forhold til layout skjer.

Det ble gjort noen mindre endringer under utviklingen av prototypen på plasseringen av elementer, men det samme oppsettet ble beholdt. Her merket vi at det å lage den første prototypen i HTML og 960.gs var et smart trekk, da vi kunne beholde mye av den samme koden, og bygge på HTML- og CSS-koden fra prototypen. For å gjøre siden litt mer behagelig visuelt og fjerne litt av ”boks”-utseende, laget vi avrundede hjørner på alle boksene i headeren og sekundærinnholdet. Dette gjorde vi ved hjelp av CSS3-funksjonen `border-radius`. Ulempen ved denne implementeringen av avrundede hjørner er at den kun fungerer i Mozilla Firefox og Apple Safari, men vi ser ikke på denne funksjonen som kritisk, da den kun forbedrer det estetiske ved siden, og ikke tillegger noen ekstra funksjonalitet.



Figur 4.7: Prototype

Figur 4.7 viser et skjermbilde av prototypen som ble brukt i brukertesting. I denne testen så vi blant annet etter svakheter med brukervennligheten, og måter brukergrensesnittet kunne bli forbedret. Resultatene fra denne brukertesten er mer beskrevet i vedlegg D på side 105, men de største svakheterne var opplastingsknappen og synlighet av visse elementer i opplastingsprosessen. Hele prototypen er inkludert på cd.

⁹ <http://meyerweb.com/eric/tools/css/reset/> – CSS Tools: Reset CSS. (Gratis til alle formål)

I det endelige resultatet gjorde vi en del endringer basert på erfaringer fra brukertesten. Blant annet ble ”last opp”-knappen flyttet ned for å skille seg ut fra headeren, og alle knapper fikk et gjenskinn slik at de minner mer om ordentlige knapper. I opplastingsprosessen ble ”synlighet”-listen gjort mer synlig ved å legge på en boks som beskriver hvert punkt.

4.3.1.1 Bruk av designmønstre

Gjennom hele designprosessen har vi sett på kjente løsninger av problemer knyttet til brukergrensesnitt på web, og hvordan andre websider har laget sine brukergrensesnitt. En viktig kilde har vært websiden www.ui-patterns.com, som samler og kategoriserer slike mønstre (patterns). Ved å hente inspirasjon fra kjente mønstre blir det enklere for brukeren å kjenne seg igjen, og vi slipper å finne opp hjulet på nytt.

For eksempel er det vanlig å bruke en liste over alle stegene i en prosess, slik vi har i opplastingsprosessen. Da får brukeren tilbakemelding om hvor han befinner seg, og hva som gjenstår.

4.3.1.2 Ikonbruk

For å gjøre brukermenyen på toppen av siden mer synlig valgte vi å bruke, i tillegg til beskrivende tekst, ikoner som fungerer som metaforer. For å logge seg inn valgte vi en hengelås som symboliserer at man låser seg inn, mens en liten person ble brukt for å vise kontoreregistrering. Etter at man har logget seg inn får man mulighet til å se sin offentlige profil, gå til sitt kontrollpanel og logge seg ut. Igjen brukes den lille personen for å symbolisere profilen sin, mens en skiftenøkkel brukes for å symbolisere kontrollpanelet.

Alle ikoner er hentet fra famfamfams Silk-pakke¹⁰. Silk-pakken er lisensiert under Creative Commons 2.5 Da opphavspersonen bak Silk krever en link til sin side ved bruk av ikonsettet, er denne plassert under ”Om” på websiden.

Det har nærmest har blitt en de facto standard å tilby brukeren et såkalt *favicon*, altså et lite ikon som legger seg til venstre i adresselinjen i nettleseren. Dette hjelper brukere å raskt identifisere siden hvis man har mange faner i nettleseren, og gir et eventuelt bokmerke til websiden et ikon. Som ikon var det da naturlig å velge logoen.

4.3.2 Tjener

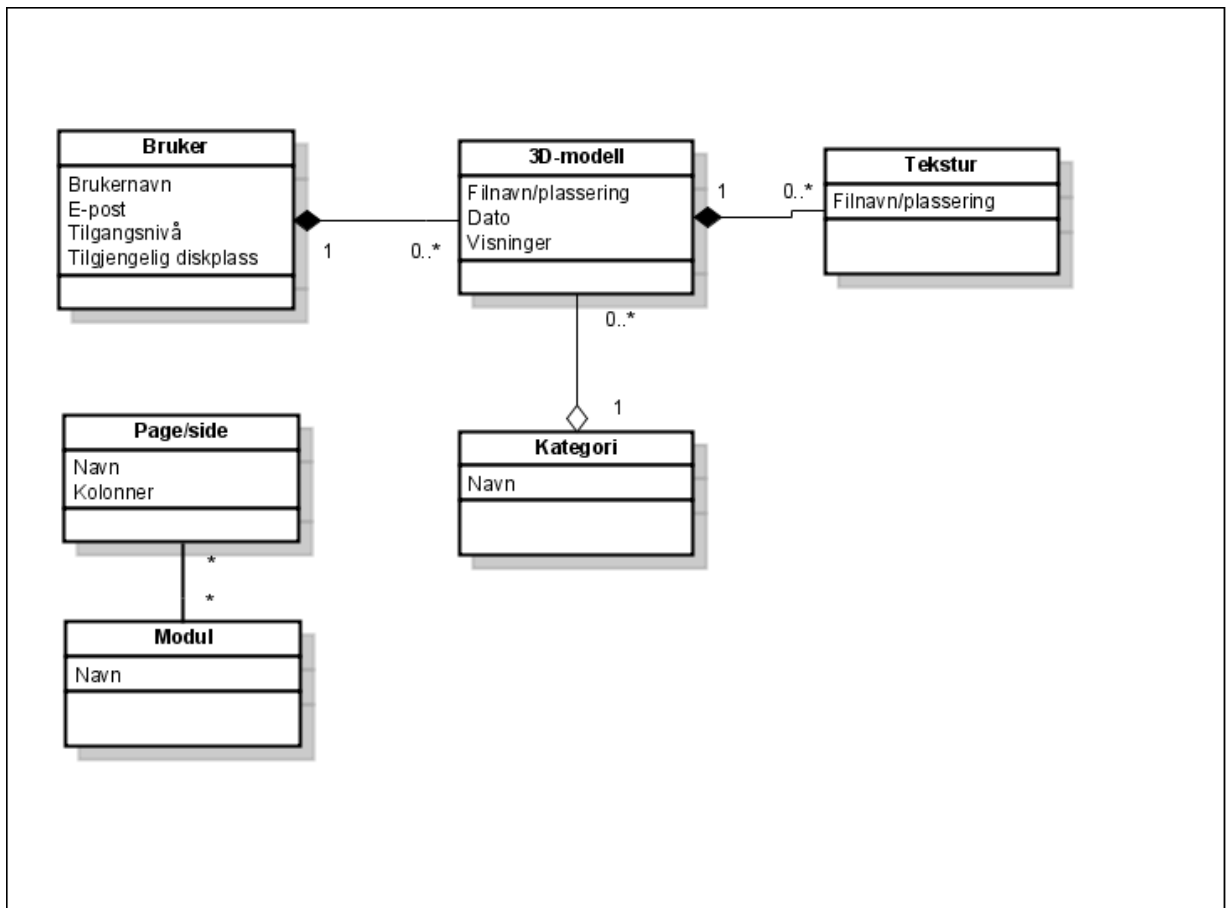
Grovt sett er systemet delt opp i tre: brukerhåndtering, maler og modeller. I Figur 4.8 nedenfor er dette illustrert i et enkelt klassediagram med de mest nødvendige attributtene.

Her ser vi at en bruker kan være tilknyttet ingen eller flere 3D-modeller, mens hver 3D-modell kun kan være tilknyttet en bruker, og ikke kan eksistere uten en bruker. Det samme gjelder for teksturene, en tekstur kan kun være tilknyttet en 3D-modell, mens 3D-modellen kan ha enten ingen eller flere teksturer. Uten en 3D-modell kan ikke en tekstur eksistere.

Hver kategori kan inneholde ingen eller flere 3D-modeller, mens hver 3D-modell kun er tilknyttet en kategori. Her er derimot ikke kategoriene avhengig av 3D-modellene, og kan eksistere uten 3D-modeller.

Malene er bygd opp av hovedsider, hvor hver side kan inneholde ingen eller flere moduler, og hver modul kan være tilknyttet ingen eller flere sider.

¹⁰ <http://www.famfamfam.com/lab/icons/silk/>

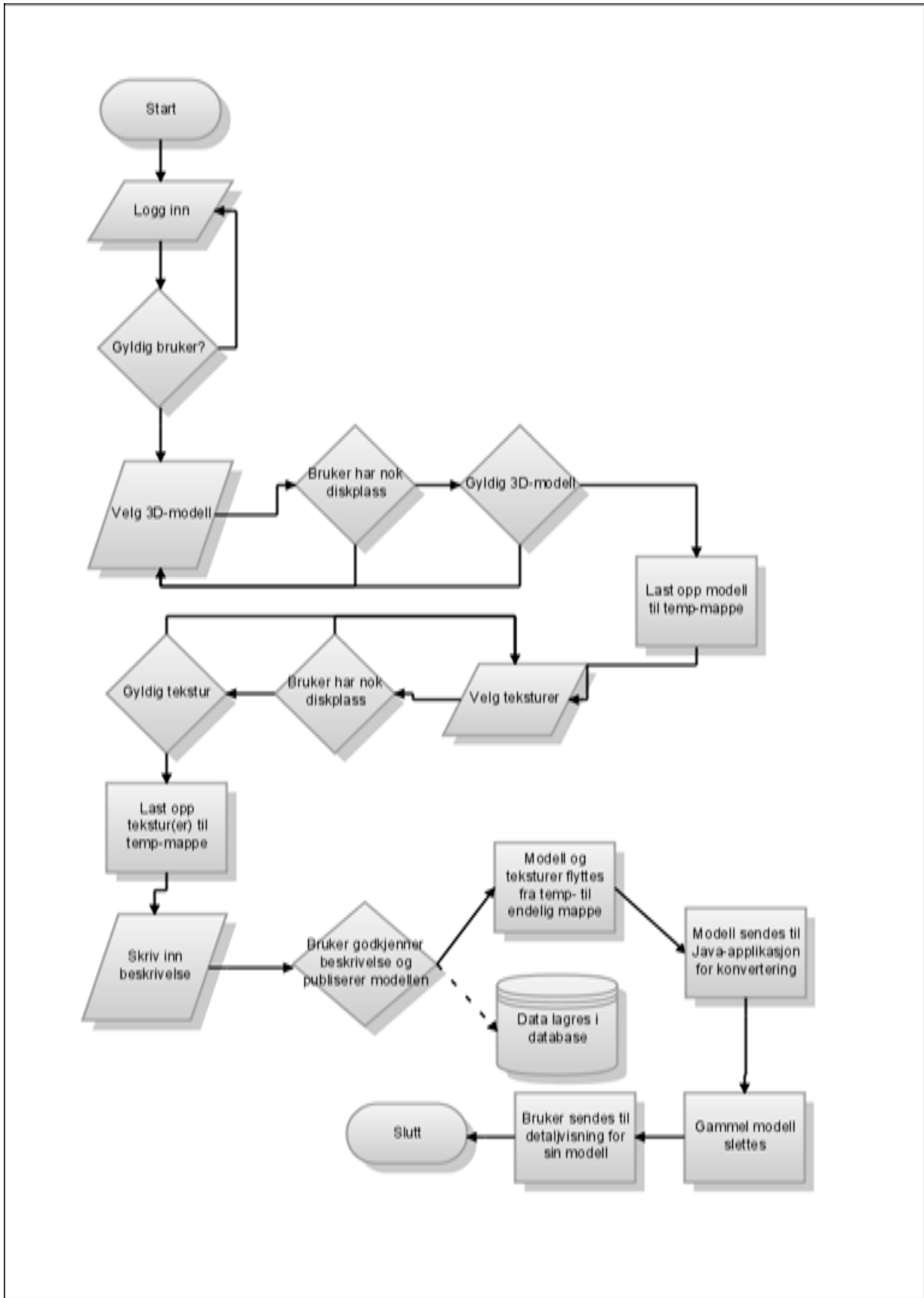


Figur 4.8: ER-klassediagram

For å holde prosjektet organisert, laget vi en mappestruktur.

- `config/` inneholder alle konfigurasjonsfiler.
- `lib/` inneholder alle klasser og funksjoner.
- `scripts/` inneholder alle eksterne klasser og funksjoner som ikke er utviklet av oss.
- `template/` inneholder alle CSS- og JavaScript-filer, og alle bilder brukt til layout. I tillegg ligger alle maler brukt for å generere HTML-filer.
 - I og med det kan bli forholdsvis mange HTML-maler, bruker vi understreker først i filnavnet for å beskrive hvor de skal plasseres på siden. Dobbel understrek () betyr at den beskriver en av de fire hovedstrukturene på siden. Enkel understrek () betyr at den er en modul og hører til i sekundærinholdet, mens ingen understrek betyr at den er en egen side og hører til i hovedinnholdet.

En mer utfyllende mappestruktur finnes i vedlegg L på side 149.



Figur 4.9: Flytkartdiagram for opplasting av 3D-modell

Den største prosessen i systemet er uten tvil opplasting av 3D-modell, og er illustrert i Figur 4.9 ovenfor. Dette diagrammet forutsetter at bruker har opprettet seg en brukerkonto på websiden. Prosessen består av fire steg:

1. Opplasting av 3D-modell
2. Opplasting av tekstur(er)
3. Skriv inn beskrivelse
4. Verifiser/publiser

Den starter med at en bruker vil laste opp en 3D-modell i steg en, og må derfor logge seg inn. Bruker angir brukernavn og passord, og systemet sjekker at dataene stemmer overens med de lagret i databasen. Deretter velger bruker én 3D-modell, hvor systemet sjekker at bruker har nok diskplass til denne modellen, og verifiserer at 3D-modellen er av riktig filtype. Modellen legges i en midlertidig mappe på serveren, og bruker føres til steg to.

Her velger brukeren de teksturer som eventuelt er knyttet til modellen, og igjen kjøres sjekk for diskplass og gyldige filer. Hvis dette går igjennom, lastes disse opp til den samme midlertidige mappen som 3D-modellen.

I det tredje steget skal brukeren skrive inn beskrivende tekst for sin 3D-modell, altså en tittel, beskrivelse og valg av kategori. Brukeren kan også skrive inn eventuelle tags, laste opp et beskrivende bilde, velge hvem som skal kunne se 3D-modellen og angi eventuell opphavsmann.

Det fjerde og siste steget leser brukeren igjennom sine data fra det forrige steget, og publiserer sin 3D-modell. Ved publisering flyttes 3D-modellen og alle teksturer fra den midlertidige mappen til den endelige mappen, og all data om modellen lagres i databasen. Videre sendes 3D-modellen til en Java-applikasjon som konverterer modellen til et format som 3D-applikasjonen kan lese. Deretter slettes den originale modellen som brukeren lastet opp, og brukeren blir sendt til detaljvisningen for modellen. Med dette avsluttes prosessen.

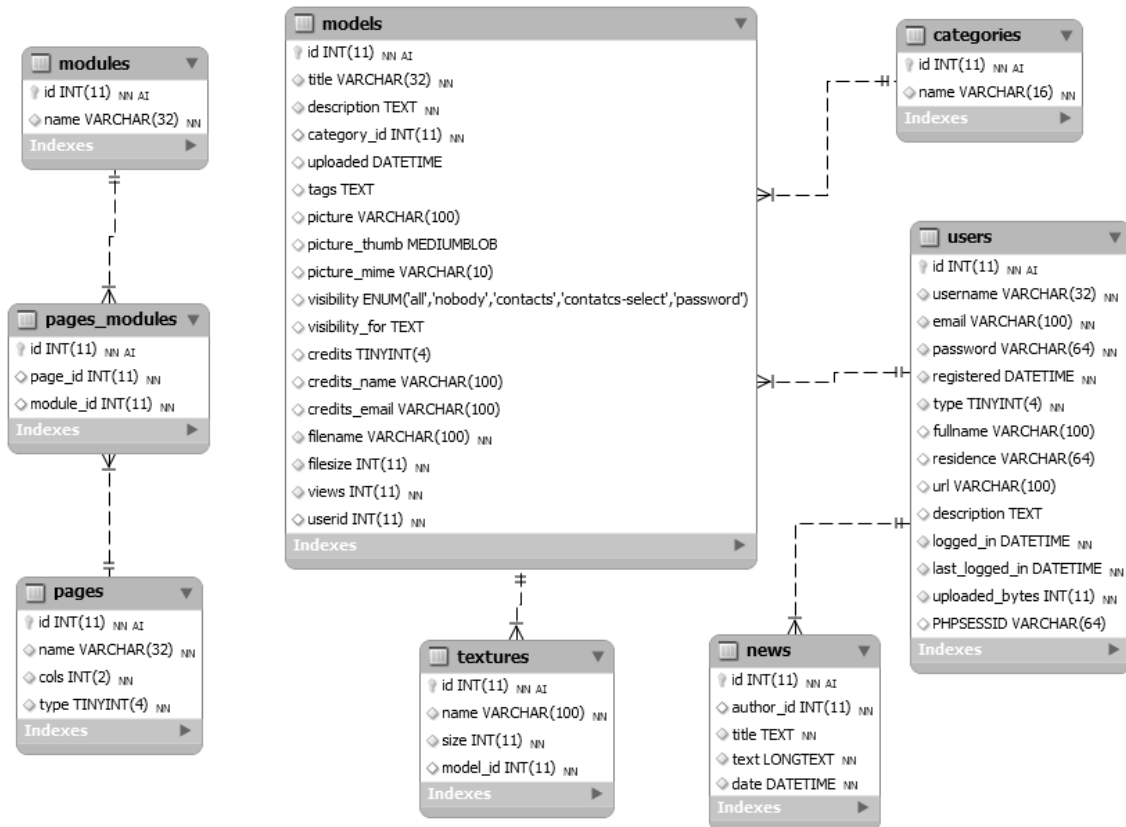
4.3.3 Databasestruktur

Databasens struktur ble utviklet tidlig i prosessen, og ble utvidet med hvert steg i prosessen for å lagre alle data som var nødvendige. For å hindre at data ble lagret flere steder ble det laget flere tabeller, hvor alle skulle ha et identifiserende felt, også kalt primærnøkkel. Hvis data skulle linkes sammen, laget vi fremmednøkler mellom disse.

Alle fremmednøkler har spesielle regler om hva som skal skje hvis den ene fremmednøkkelen slettes. Et eksempel på dette er i `model_s`-tabellen, som inneholder en fremmednøkkel til brukers id. Her er det laget en regel som sier at hele modellen skal slettes hvis brukeren slettes:

```
ADD CONSTRAINT `model_user_id` FOREIGN KEY (`userid`) REFERENCES `users`  
(`id`) ON DELETE CASCADE ON UPDATE NO ACTION;
```

Cascade betyr da at den skal slettes.



4.10: EER-diagram

Models: Inneholder all data om hver modell. Det lagres iden til brukeren og kategorien, som er koblet til henholdsvis bruker- og kategoritabellen via en fremmednøkkel. Hvis kategorien som modellen er knyttet til settes kategori til NULL, og hvis brukeren modellen er knyttet til slettes, slettes modellen.

Textures: For hver tekstur som lastes opp lagres filnavnet, filstørrelsen og iden til modellen som er tilknyttet teksturen. Slettes denne modellen, slettes også teksturen.

Users: Tar vare på informasjonen om hver bruker i systemet, og inneholder blant annet brukernavnet, e-post, kryptert passord og type. Sistnevnte bestemmer hvilket tilgangsnivå brukeren har (0 er vanlig bruker, 2 er administrator).

Categories: Inneholder navnet til alle kategoriene.

Pages: En tabell som inneholder filnavn, antall kolonner og type til alle undersider. Type bestemmer hvilke menyer siden skal havne i.

Modules: Inneholder filnavnet til malen for hver modul.

Modules_pages: Siden hver modul kan tilhøre flere sider, og hver side kan ha flere moduler, måtte vi ha en ekstra tabell som linker modulene og sidene sammen.

News: Inneholder tittelen, artikkelteksten, datoen for publisering og iden til brukeren som opprettet nyheten. Hvis brukeren slettes, slettes nyheten.

4.4 GUI, 3D-applikasjon

4.4.1 Oppgaven fra arbeidsgiver

Selve GUIen har som hovedoppgave å gi brukeren av denne applikasjonen muligheten til å navigere seg rundt i en 3D verden der brukerne selv skulle stå for 3D modellene.

I tillegg til den applikasjonen som skulle la brukeren bevege seg rundt i en 3D verden, skulle oppdragsgiveren også lage en applikasjon som skulle ta for seg produkter, og produktvisning. Da for eksempel en bil, en bok, eller noe liknende. Her ville det da bli aktuelt å gå bort i fra ett system der man styrer ett kamera, men istedenfor styre objektet i viewporten.

Dette ville selvsagt kreve en helt annen form for GUI. Det vi kom frem til var jo at det kunne vært greit, både for brukeren, og for den totale helheten til websiden og applikasjon(e) at det fantes en GUI for produktvisningen som knyttet de to måtene å vise frem 3D på sammen.

Så da for å kunne holde en helhet gjennom begge måtene å legge frem en 3D modell på, så gikk vi løs på en ekstra GUI.

4.4.2 Inspirasjon

4.4.2.1 3D Golfbaner

Til inspirasjon fikk vi tilsendt noen av de tidligere versjonene av en lignende applikasjon oppdragsgiver har stått for, en 3D-visning av en golfbane (Figur 4.11).

Du har to måter å kunne bevege deg rundt på i denne golf banen. Den første måten er ved hjelp av de grønne knappene du kan se nedover på venstre side på bildet. Disse knappene er markert med «hull 1, hull 2, og videre», og vil da flytte deg umiddelbart til det aktuelle hullet på golfbanen slik at du slipper å måtte lete deg frem.



Figur 4.11: 3D applikasjonen til visualisere.no som viser en golfbane fra fugleperspektiv.

I tillegg til denne måten å bevege seg rundt på har man da det vi til slutt kalte «navigasjons pod'en» (Figur 4.12). Grunnen for dette kallenavnet var ganske enkelt fordi hele navigasjonskontrollen var basert på en iPod Nano, der man da hadde byttet ut de vanlige knappene på en iPod med sine egne til bruk for navigasjon i en 3D verden. I tillegg til dette har de også lagt til kontroller for tilt av kamera, samt elevasjon, eller bevegelse horisontalt.

Denne kontrollen har også ett «vindu», for å kunne gi beskjeder som brukeren kunne ha nytte av. Det eneste problemet med denne skjermen, var at den ble borte i forhold til resten av applikasjonen.

Disse knappene er også ganske selvforklarende, og man skjønner fort hva de gjør ved å trykke på dem og prøve seg frem. Det negative med denne er at det er litt for mye, på for liten plass, vi fant ved flere tilfeller ut at ved den minste forandring på musepekeren kunne fort få deg til å fare av sted i feil retning.



Figur 4.12: Ett utsnitt av navigasjons pod'en som brukes for å styre kamera rundt på golfbanen.

4.4.2.2 Diverse spill

Så klart når man skal lage en applikasjon der i første omgang hele oppgaven går ut på å bevege seg rundt i ett hus eller liknende, så faller tanken fort på diverse spill, alt fra rollespill til skytespill.

I ett typisk skytespill der man beveger seg rundt i ett hus eller liknende er det som regel noen kontroller som går igjen. W, A, S og D er nok de mest kjente kontrollene for da frem, venstre, bak og høyre.

I tillegg til disse fire tastene har man også da gjerne musen som det viktigste verktøyet. Denne er den som styrer selve kameraet. Opp vil bety at kameraet vil tilte bakover, og peke oppover. Venstre eller høyre vil gjøre at kamera vil panorere tilsvarende. Denne måten å kontrollere kamera og bevegelse på kan man spore helt tilbake til DOOM (Figur 4.13) og har siden det vært en de facto standard



Figur 4.13: Ett skjermbilde fra det første DOOM spillet. Her beveget man personen med tastaturet, og pekte rundt seg med musen.

blant spill av denne typen.

Dette var da selvsagt noe vi også vurderte å bruke da i vår GUI. Dette er en vel utprøvd, og fungerende måte å bevege seg i et 3D-rom i såkalt «*first-person-view*», og ville gitt oss det vi trenger av mobilitet og bevegelighet.

Det kunne fungert, og vi har gitt uttrykk til oppdragsgiver som står ansvarlig for selve applikasjonen at vi vil gjerne ha disse tastene aktive i applikasjonen, selv om det da også kommer til å være en GUI med i tillegg. Dette for å gi de såkalte superbrukerne en mulighet til å ta i bruk applikasjonen uten å måtte ty til bruk av mus hele tiden.

En ting som kunne gjøre dette vanskelig derimot er hvis brukeren ikke har tilgang til enten mus eller tastatur. Dette ville da skape problemer og kan lamme bruken av applikasjonen kraftig.

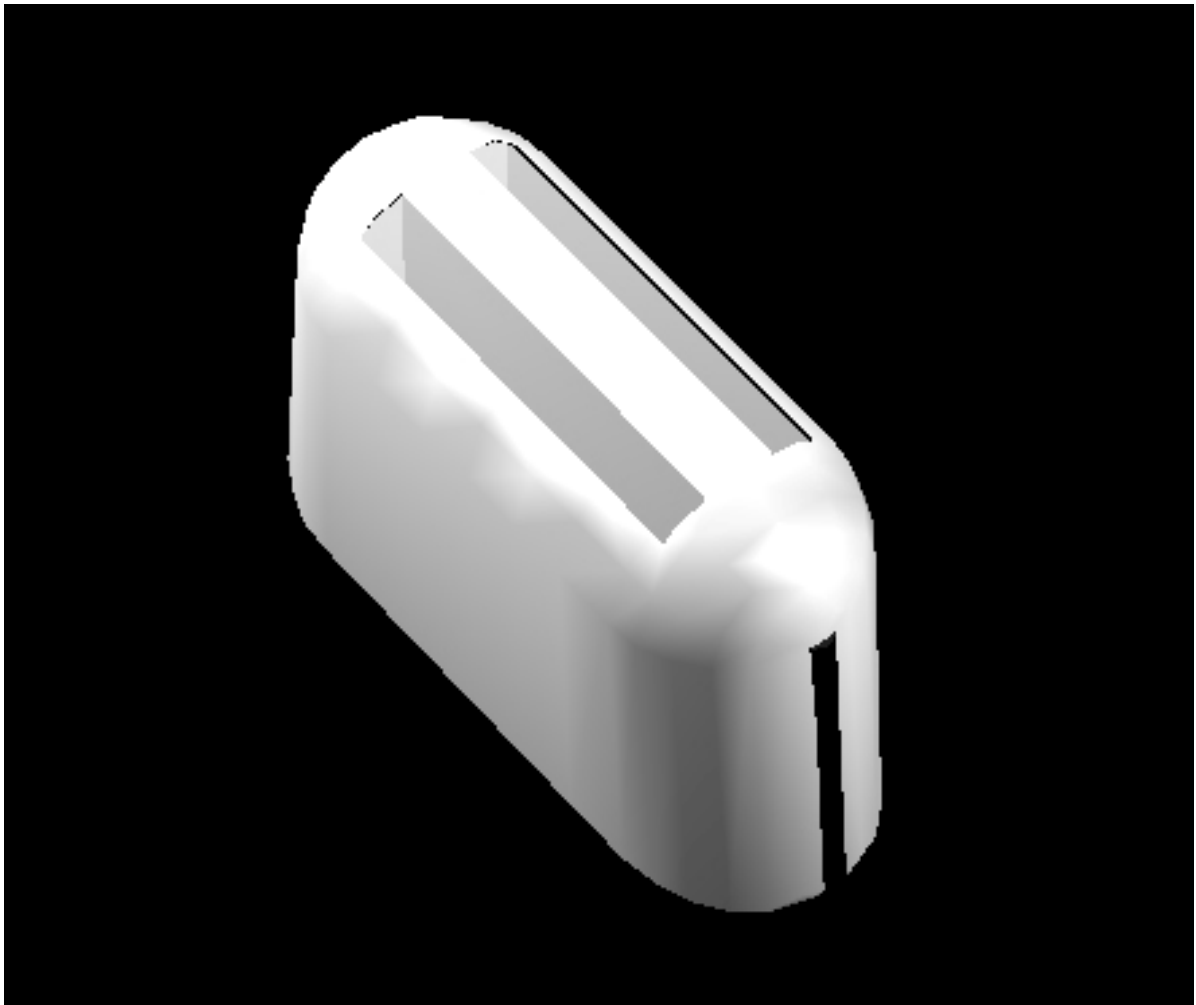
Et annet scenario er ved fremvisning, og alt man har tilgjengelig er en bærbar PC. Dette vil nok fungere, men vil skape problemer ettersom det ikke er alltid man er like stødig på touchpadden, og spesielt da om man skulle overlape med ett sett fingre på bevegelse, og et annet sett på touchpadden for å kunne se rundt seg. Dette var faktorer som alle spilte sin rolle i utviklingen av en GUI som kunne brukes i den sammenhengen vi var ute etter.

4.4.2.3 Produktvisning og en annen vri på 3D verden

Som nevnt ovenfor i punkt 4.4.1, så ville oppdragsgiver også ha mulighet til å kunne bruke denne applikasjonen til å kunne vise frem enkle produkter. I en slik visning vil det ikke lenger være aktuelt å gå rundt objektet, men heller å ha et statisk kamera, og la brukeren rotere objektet slik han vil.

Denne måten å bevege objekter på kan løses på flere måter. Den mest vanlige måten er å kun gi brukeren ett vindu med objektet plassert i midten, for så å muligens ha noe tekst forklarende hvordan brukeren kan snu på objektet. Den mest vanlige måten å gjøre dette på er ved å la brukeren klikke på objektet, og deretter dra musen rundt for å rotere objektet rundt seg selv. Det vil da gi brukeren en mulighet til å se objektet eller produktet fra alle vinkler og kanter.

Kristian Botnen, vår kontaktperson hos visualisere.org, har laget en tidlig versjon av produktvisningsapplikasjonen og tanken bak den er vist i Figur 4.14 nedenfor.



Figur 4.14 En tidlig versjon av produktvisningsapplikasjonen, da med en brødrister som modell.

4.4.2.4 Minimap

I tillegg til funksjonaliteten nevnt over kom vi med et forslag om et såkalt minimap til 3D-applikasjonen, som skulle kunne hjelpe brukeren med navigasjon i store modeller. Dette er noe som manglet på golfbane programmet nevnt ovenfor (se punkt 4.4.2.1), og kunne vært veldig nyttig når det kommer til navigering rundt på banen.

Et minimap er et ekstra vindu plassert et eller annet sted på applikasjonen som har som hovedoppgave å vise hele 3D-objektet rett ovenifra slik at brukeren kan se hvor på modellen han befinner seg. Minimap er en utbredt måte å gi generell oversikt i mange medier i dag. Et godt eksempel på dette er Google Maps¹¹, der hovedvindu gjerne er zoomet inn, og man kan se et lite minimap nede i høyre hjørnet som et oversiktskart for å gi brukeren en mulighet til å se hvor han befinner seg i et større perspektiv, da gjerne med gatenavn. De fleste minimaps i applikasjoner har gjerne har en annen topografi enn hva selve hovedkartet da har. Et minimap er ofte veldig enkelt sett grafisk, da med tydelige ikoner som skal vise brukeren hvor han befinner seg, og eventuelle interessepunkter i nærheten.



Figur 4.15: Eksempel på et minimap i Google Maps

4.4.2.5 Klar for videre arbeid

Etter å ha sett på alle eksemplene ovenfor følte vi at vi hadde relativt god oversikt over hvordan de forskjellige problemstillingene kunne løses. Det finnes mange løsninger på de forskjellige problemene, og det vi nå måtte gjøre var å se over de valgene vi hadde, og finne ut hvilke løsninger som passet applikasjonen vår best.

Les mer om våre valg, teoretisk bakgrunn og løsninger under punkt 5.3 på side 52.

¹¹ <http://maps.google.no/> - Google Maps Norge.

5 Utvikling

5.1 Klient

Vi har benyttet oss av to rammeverk, jQuery og 960 Grid System, for henholdsvis JavaScript og CSS. Fordelen med å bruke rammeverk er rask utvikling og god nettlestøtte. Man sikrer også bruk av standardisert kode. I tillegg får man tilgang på mange utvidelsespakker og god dokumentasjon.

Ulempene er mye unødvendig kode, og inkludering av mange filer, som kan skape unødvendig mange HTTP-forespørsler mot serveren og går utover ytelsen.

Både jQuery og 960 Grid System, samt alle utvidelsespakker er lisensiert under GNU GPL- og MIT-lisensen.

5.1.1 JavaScript og jQuery

Vi har benyttet oss av JavaScript (JS) og jQuery for diverse Ajax-funksjoner og for å forbedre brukervennligheten. Ulempen med JS er at noen brukere har det avskrudd på grunn av sikkerhetshensyn og at ikke alle nettlesere støtter JS. Dette er ikke veldig mange, men i følge W3Schools sine statistikker¹² var det i januar 2008 5 % av alle brukere av internett som hadde JS deaktivert. Derfor er det viktig at all *nødvendig* informasjon og funksjonalitet er tilgjengelig for alle, uansett om JS er aktivert eller deaktivert.

5.1.1.1 Velkomstboks

På forsiden får alle nye besøkende opp en velkomstboks med generell informasjon om tjenesten. Da denne er ganske dominerende, må man ha mulighet til å fjerne denne. Dette gjøres ved å trykke det røde krysset i høyre hjørne, da skjules boksen ved hjelp av en jQuery-animasjon. Samtidig settes en informasjonskapsel (cookie) hos brukeren som angir at brukeren har skjult denne beskjeden, og som varer i ett år. Neste gang brukeren oppdater forsiden, og så lenge informasjonskapselen finnes eller brukeren er logget inn, filtreres beskjeden vekk ved hjelp av PHP.

5.1.1.2 Informasjonsvindu

Ved opplasting av modell og registrering av bruker krever vi at brukeren har lest retningslinjene, og for at brukeren ikke skal bli ledet vekk fra siden når linken til retningslinjene klikkes på, åpner vi disse i et eget JS-vindu, som legger seg over siden. Dermed mister ikke brukeren noe av informasjonen på den nåværende siden, og kan lukke vinduet når han har lest retningslinjene. Denne funksjonen utføres av en jQuery-plugin kalt Impromptu¹³. Det samme informasjonsvinduet blir brukt når man skal godkjenne sletting av en 3D-modell, nyhet eller bruker.

¹² W3Schools statistikker: http://www.w3schools.com/browsers/browsers_stats.asp

¹³ jQuery Impromptu: <http://www.trentrichardson.com/Impromptu/index.php>

Her har vi brukt ikke-påtrengende JS, altså at de med JS avslått får tilgang til akkurat samme funksjon. Linken til retningslinjene peker egentlig til retningslinjer-siden, men JS fjerner denne linken og lar heller brukeren få opp et eget vindu. Dermed går ikke de uten JS glipp av noe.

5.1.1.3 Opplasting av teksturer

Brukere skal ha mulighet til å laste opp ingen eller flere teksturer til sin 3D-modell.

Standardfilvelgeren i HTML (`<input type="file" .../>`) støtter kun opplasting av en fil om gangen. Vi hadde fire løsninger på dette problemet;

1. La brukeren laste opp en tekstur, og la han velge om han vil laste opp flere teksturer.
2. Bruker velger hvor mange teksturer han vil laste opp i et `input`-felt, og det genereres via PHP så mange filvelgere som han trenger. Man må deretter velge en og en tekstur, men kan laste opp alle i en bolk.
3. Utvikle en Java-applet som tar seg av filopplasting.
4. Vise en filvelger, men generere denne via JS/Flash, og la bruker velge flere filer i samme velger.

Alle hadde sine fordeler og ulemper, men vi valgte å fokusere på brukervennlighet og valgte dermed det siste punktet. Vi ser at det tredje punktet kunne gitt en like god brukeropplevelse, og bedre håndtering av store filer, men i og med at vi hadde lite erfaring med Java-programmering, valgte vi bort denne løsningen. Den største ulempen med en slik løsning av at brukeren må ha JS og Flash aktivert/installert. Allikevel er dette et krav vi er villig til å sette, i og med at antallet hjemme PC-er med Flash tilgjengelig ligger på 99 % ¹⁴.

Løsningen vi valgte var utvidelsen Uploadify til jQuery, som dekket våre behov. Dette er en jQuery-plugin lisensiert under GPLv3 og MIT, for filopplasting med mange valgmuligheter.

Vi får da en visuell fremstilling av hvor mye av hver fil som er lastet opp, og mulighet til å angi hvilke filtyper man kan velge og maksimumsstørrelse på hver fil. Hver fil blir sendt til en egen PHP-funksjon som tar i mot informasjon om filen fra den globale `$_FILES`-variabelen, og flytter filen til en midlertidig mappe. Her blir den også validert en gang til, men denne gangen på tjeneren, siden man aldri kan stole på data validert på klientsiden.

5.1.1.4 Sletting av modeller

Alle brukere har mulighet til å slette sine egne modeller via sitt kontrollpanel, som gjøres ved hjelp av en Ajax-funksjon. Dermed slipper brukeren å laste siden på nytt, og slipper eventuelle feilmeldinger om at modellen allerede er slettet.

Alle modellene til brukeren blir vist i en tabell, og hver rad har id `itemid_<modell_id>`. For å forhindre at noe blir slettet ved et uhell, får man opp et bekreftelsesvindu som man må trykke "Ok" for å slette, eller "Avbryt" for å gå tilbake. Her hentes modellens id nummer ut, og sendes til en PHP-funksjon som sletter den aktuelle modellen.

¹⁴ Adobes statistikker: http://www.adobe.com/products/player_census/flashplayer/

5.1.1.5 Gjennomsiktige PNG-filer i IE6

Internet Explorer 6 (IE6) støtter ikke gjennomsiktige PNG-filer, og da vi bruker noen gjennomsiktige bilder (logo, ikoner), inkluderte vi et JavaScript som skrur på dette i IE6. Skriptet heter DD_belatedPNG¹⁵, og inkluderes kun i denne nettleseren, via *conditional comments*. Conditional comments brukes for å inkludere HTML-kode til visse nettlesere, ved en if-blokk:

```
<!--[if IE 6]>  
Instruksjoner for IE6 her  
<![endif]-->
```

5.1.2 960 Grid System

Rammeverket er et gridsystem som setter tekst og bilder i et rutenett, og baserer seg på, som navnet tilsier, en bredde på 960 piksler. Dette er en bredde som blir mye brukt, da den passer for oppløsninger fra 1024x768 og oppover (forutsatt at nettleservinduet er maksimert).

Ifølge Market share¹⁶ er det kun rundt 5 % av alle brukere som benytter seg av en oppløsning lavere enn dette. Vi fikk også statistikk på dette området fra oppdragsgiver (vedlegg J, side 139), men denne var ufullstendig da 45 % av alle skjermoppløsninger var ukjent. Vi ser allikevel at det kun er 0,8 % som bruker oppløsningen 800 x 600, og kan anta at det ikke finnes noen med lavere oppløsning. Vi har ikke tatt hensyn til nettleservinduer som ikke er maksimert.

Det finnes tolv og 16 kolumner hvor vi valgte 16 kolumner, og i den generelle layouten blir tolv av dem blir brukt på hovedinnholdet, mens de resterende fire blir brukt på sekundærinholdet.

Noe av koden kan bli usemantisk, for eksempel klassenavn som `grid_4` sier lite om innholdets. Vi anser likevel fordelene som større enn ulempe.

5.2 Tjener

I og med at oppdragsgiver hadde satt det som en føring at vi skulle utvikle et dynamisk nettsted, med PHP versjon 5.2 og MySQL 5 installert på webhotellet. Dermed hadde vi ganske stor frihet, og sto med tre valg i forhold til utviklingsplattform

1. Utvide et eksisterende CMS-system (Content Management System, et publiseringssystem)
2. Ta i bruk et rammeverk
3. Utvikle selv

Vi luket kjøpt ut det første punktet, da det kunne blitt veldig tidkrevende å sette seg inn i et CMS-system og utvikle egne utvidelser, og så at vi like gjerne kunne bruke en rammeverk hvis vi skulle benytte oss av ferdige systemer.

Men mellom det å bruke et rammeverk og utvikle selv var det nesten dødt løp, da begge metodene hadde fordeler og ulemper. Et rammeverk vil være mer skalerbart og dokumentert, men vi ville ikke hatt like mye læringsutbytte, samt er de fleste rammeverk forholdsvis bratt læringskurve. De rammeverkene vi så på var CakePHP¹⁷ og CodeIgniter¹⁸, begge open-source rammeverk.

¹⁵ http://www.dillerdesign.com/experiment/DD_belatedPNG/

¹⁶ <http://marketshare.hitslink.com/report.aspx?qprid=17>

¹⁷ <http://cakephp.org/>

¹⁸ <http://codeigniter.com/>

Ved å utvikle systemet selv får vi akkurat de funksjonene vi trenger og et system som er tilpasset våre behov. Da slipper å forholde oss til funksjoner vi ikke trenger. Ulempene er at det kun er vi som virkelig kjenner systemet, og kan være vanskelig å utvide i fremtiden.

Vi valgte til slutt å utvikle systemet vårt selv. Dermed var det veldig viktig at alle funksjoner og klasser ble dokumentert grundig, kodekonvensjoner ble fulgt og at all kode ble skrevet på en oversiktlig måte. For å holde koden oversiktlig bestemte vi å bruke et designmønster basert på Model-view-controller (MVC) prinsippet. Her skilles logikk (*model*) fra presentasjon (*view*). Presentasjonen skal ikke kunne endre noe data, og kun brukes som en ren mal som viser HTML kode. På den andre siden skal ikke logikken (altså hvor all data er lagret) presentere noe for brukeren, kun hente ut data og ha funksjoner som kan endre egne data. I mellom disse har man kontrolløren (*controller*), som brukes for å kommunisere mellom logikken og presentasjonen. kontrolløren mottar kommandoer fra brukeren via presentasjonen, og utfører disse kommandoene på logikken. Dermed har man en lagdelt applikasjon, hvor de to lagene logikk og presentasjon skal fungere uavhengig av hverandre.

Vi valgte også å bruke objektorientert programmering. Dette er forholdsvis nytt i PHP, og dokumentasjonen er til tider litt tynn. Vi hadde erfaring fra objektorientering i programmeringsspråk som C++ og Java, men webapplikasjoner skiller seg fra disse språkene i den forstand at en webapplikasjon opprettes og avsluttes alle objekter ved hver sidevisning, mens i en C++-applikasjon lever objekter helt til applikasjonen avsluttes.

Backup har blitt utført ved jevne mellomrom, og ved at alt innholdet på webhotellet og i databasen har blitt hentet ned og plassert på Jon Espens utviklingsstasjon. De har også blitt lagt i en Dropbox-konto. Vi har forsøkt å ta minst to backup i uken, og mot slutten ble dette gjort hver dag.

5.2.1 Rammeverk

Vi bestemte oss for å lage et rammeverk for fremvisning av sider, og la en fil inkludere alle sidene. Dette er da `index.php`-filen i rotmappen, som også inkluderer og *instansierer* (opprettet et objekt av en klasse) de nødvendige klassene, altså `user`-, `error`-, `helper`-, og `template`-klassen.

Da utviklingsversjonen av websiden ble lagt i en annen mappe enn der den endelige versjonen, ble det definert en global `ROOT`-variabel som viser til denne utviklingsmappen. Dermed må kun denne variabelen endres ved lansering. For å legge til rette for et eventuelt serverbytte senere er alle mappeseparatører (*directory separators*) byttet ut med PHPs `DIRECTORY_SEPARATOR` (og forkortet til `DS`), da noen Microsoft-baserte servere bruker `"\"` som mappeseparatør, mens unix-baserte servere bruker `"/`.

Da vi vil følge MVC-tankegangen inkluderes en egen fil, `controller.php`, som gjør all kommuniserer mellom logikk og presentasjon.

For å starte selve sideinkluderingen hentes først toppen på siden via `getHeader()`, som inkluderer malen for toppen, hvor 960.gs systemet startes.

Så inkluderes selve siden, som henter hovedinnhold og sekundærinnhold, via `getPage()`. Her hentes iden til siden som skal inkluderes via `GET`-variabelen `p`, og hvis ingen er angitt settes filnavnet til å være `"front"`, iden til 1 og antall kolonner til 12. Iden brukes i en spørring mot databasen, som henter ut filnavnet og antallet kolonner siden består av. For å vise siden har vi en metode som heter `displayPage()`, som inkluderer `__page.php` fra HTML-malen. I denne blir igjen riktig grid-system generert fra antall kolonner, og riktig mal blir inkludert fra filnavnet.

Deretter opprettes det som skal bli sekundærinneholdet, ved at et objekt av `Sidebar` instansieres i `template`-instansen. Konstruktøren til `Sidebar` tar i mot iden og (16 - antall kolonner) til den nåværende siden som argument. Da vet sekundærinneholdet bredden sin, og har iden til den nåværende siden.

Det neste steget er å hente ut modulene for siden, som er lagret i tabellen `modules` i databasen. Da hver modul kan høre til flere sider, og en side kan ha flere moduler, måtte vi opprette en ekstra tabell som binder disse sammen, `pages_modules`. Her er modulens id og sidens id lagret. `Sidebar` har metoden `addModules()`, som tilkalles i `__sidebar.php`. Her hentes hver modul for den aktuelle siden.

Nå er toppen, hovedinneholdet og sekundærinneholdet vist, og mangler kun bunnteksten, som hentes fra metoden `getFooter()` i `template`-instansen. Da er en side vist.

5.2.1.1 Konfigurasjon av `.htaccess`

`.htaccess` er en konfigurasjonsfil som kan plasseres i alle mapper for å manipulere serverens konfigurasjon.

For å forhindre tilgang til andre mapper, benyttet vi oss av en `.htaccess`-fil i rotmappen, som rekursivt skrur av mappevisning i alle undermapper. For å spare brukeren og serveren for unødvendig bruk av båndbredde, komprimerer vi alle `php`-, `css`- og `js`-filer med `gzip` (BetterExplained, 2007). Kort fortalt betyr dette at hvis klienten godtar komprimering, komprimeres alle filer sendt av tjeneren, som igjen pakker ut filene automatisk. Dette gjøres via en tre linjer i `.htaccess`-filen:

```
<FilesMatch "\.(php|css|js)$">
    SetOutputFilter DEFLATE
</FilesMatch>
```

Alle `php`-, `css`- og `js`-filer matches via et regulært uttrykk, som blir komprimert via metoden `deflate`. I følge Web Developer-toolbaren i Firefox komprimeres forsiden fra 121 kb til 59 kb, noe som tilsvarer en reduksjon på 48 %. I seg selv er ikke 62 kb mye, men hva brukermassen blir i fremtiden vet vi lite om, og hvis tjenesten blir populær i fremtiden er en reduksjon på nærmere 50 % av båndbredde noe alle tjener på, både brukere og eiere av tjenesten. En ulempe med denne typen kompresjon er at den kan skape problemer i eldre nettlesere¹⁹, men ut i fra statistikkene (vedlegg J) er denne andelen forsvinnende liten.

5.2.2 Kontrolløren

Kontrolløren har som oppgave å binde sammen data og presentasjon.

Denne er bygd opp ved hjelp av to `switch`-blokker, en som evaluerer `POST`-variabelen `do` og en annen som evaluerer `GET`-variabelen `do`. Det som blir sendt med i disse variablene bestemmer hvilken handling som skal utføres. Her utføres all feilhåndtering, og sender eventuelle feil til `errorHandler`-klassen.

¹⁹ http://schroepl.net/projekte/mod_gzip/browser.htm

Et eksempel på dette er en slik handling er visning av bilder:

```
case 'getimage' : {
    if (isset ($_GET ['id']) && is_numeric ($_GET ['id'])) {
        $model = new Model ($_GET ['id']);
        $modelData = $model->getModelData (array (
            "models.picture",
            "models.picture_thumb",
            "models.picture_mime"
        ));

        if (! $modelData) { // if no returned row, set error
            $error->setError ($msg ["no-image-could-be-provided"]);
            break;
        } else {
            header ("Content-type: {$modelData ['picture_mime']}");
            header ("Content-disposition:
filename=thumb_{$modelData ['picture']}");
            echo $modelData ['picture_thumb'];
            exit ();
        }
    } else {
        $error->setError ($msg ['error-id']);
        break;
    }
}
```

Denne kalles ved URLen <rot-adresse>?do=getimage&id=<id>. Først sjekkes det om den medsendte iden er satt, og at den er numerisk. Hvis ikke, avsluttes løkken med en feilmelding. Det instansieres så en Model-klasse til objektet \$model, og bildenavnet, selve bildet og MIME-typen (altså filtypen) hentes. Hvis ingen data returneres, har noe gått feil, da denne grenen av switch-blokken aldri skal kalles hvis et bilde ikke finnes. Dermed vises en feilmelding, og grenen avsluttes. Hvis det derimot finnes et bilde, settes header-typen til en type som passer bildet, slik at bildet kan vises i nettleseren.

5.2.3 Brukersystem

Ved hver sidevisning instansieres user-klassen, til objektet \$user. I klassens konstruktør sjekkes det om en session er startet, hvis ikke opprettes en via den globale PHP-funksjonen session_start(). Deretter sjekkes det om session-variabelen userid er satt. Hvis brukeren er innlogget, inneholder denne brukerens id fra databasen, hvis ikke finnes ikke variabelen. Sist sjekkes det om en informasjonskapsel med navnet "userSavePass" finnes, som inneholder brukernavnet og det krypterte passordet serialisert i base64-format. Serialiseringen lager brukernavnet og passordet i en streng, som gjør det enklere å lagre verdiene. Base64 lager en ASCII-string av binære data, slik at det kan lagres i en variabel.

Hvis brukeren er innlogget, hentes all informasjonen om denne brukeren fra databasen, og lagres i `$data` som et assosiativt array, med navnet på feltene fra databasen som nøkler. Vi har så laget en hjelpemetode, `getUserInfo()`, som tar i mot navnet på feltet man vil ha som argument, og returnerer verdien. Denne metoden er mye brukt over hele systemet, og brukes ofte for å returnere iden til brukeren som er logget inn, så vi valgte å legge den ved:

```
public function getUserInfo($prop){
    if (empty ($this->id)) { // if no user is loaded, return false
        return false;
    }

    return $this->data [$prop];
}
```

5.2.3.1 Innlogging

Innlogging skjer via metoden `login()`, som tar tre argumenter: brukernavnet, passordet (i klartekst) og en boolsk variabel som definerer om brukeren skal lagre innlogginformasjonen sin ("Husk meg"-funksjonen). Det utføres en spørring mot databasen, som sjekker om brukernavnet finnes og det medsendte passordet (kryptert i md5) stemmer overens med det lagret i databasen. Hvis det returneres data, stemmer brukernavnet og passordet, og brukeren settes som innlogget. Valgte brukeren "Husk meg", opprettes en informasjonskapsel, med PHP-funksjonen `setcookie()`. Brukeren er da innlogget i to uker, eller til informasjonskapselen i nettleseren slettes. Hvis ikke en kapsel er opprettet, er brukeren innlogget så lenge som `session`-en varer, altså 24 minutter.

Ved innlogging kjøres en spørring mot databasen, som setter inn brukerens session-id, tidspunktet for innlogging, og tidspunktet for forrige innlogging. SQL-spørringen er gjengitt under:

```
UPDATE users
SET PHPSESSID = '{$_COOKIE['PHPSESSID']}',
    logged_in = now(),
    last_logged_in = '{$this->data['logged_in']}'
WHERE id = {$this->id}
```

For økt lesbarhet legges alle PHP-variabler i krøllparentes.

Grunnen til at vi må ha tidspunktet for forrige innlogging er "Vis modeller siden sist"-funksjonen, da vi må ha en referanse til når brukeren var logget inn sist.

5.2.3.2 Registrering av konto

For å laste opp modeller, trenger man en brukerkonto, som kan opprettes i registreringskjemaet. Da det finnes en del automatiserte roboter som registrerer seg ved offentlige tilgjengelige skjemaer, implementerte vi et CAPTCHA-script som minsker sjansene for at en slik robot registrerer seg. Ved registrering angir man et brukernavn, en e-postadresse, og et passord, som kamufleres. Dette passordet må skrives to ganger for verifikasjon. Man må også huke av for at man har lest retningslinjene.

Alle data blir validert, hvor det blant annet sjekkes at e-postadressen har riktig syntaks, og at lengden på brukernavnet ikke er for langt/kort.

5.2.3.3 Utlogging

Vi har laget en funksjon så brukere manuelt kan logge ut, og slette den gjeldende session-variabelen med brukeriden, `$data`-arrayen og den eventuelle informasjonskapselen. For å slette kapselen settes varigheten på den til `time() - 3600`, altså det nåværende tidspunktet minus en time. Da sletter nettleseren den selv med neste sidevisning, siden den er utløpt.

5.2.4 Søkefunksjon

Ved et søk instansieres `Search`-klassen til et `$search`-objekt. Dette er en forholdsvis enkel søkefunksjon, som benytter seg av sammenlignings-operasjonen `LIKE` og `WILDCARD`-funksjonen i MySQL, altså at søkestrengen pakkes inn i mellom to prosenttegn. I søkestrengen fjernes alle eventuelle mellomrom før og etter ved hjelp av `trim()`. Denne søker i alle titler og beskrivelser til 3D-modellene, og brukernavnene.

Alt arbeidet gjøres stort sett av denne SQL-spørringen:

```
$sql = "SELECT models.title,
        models.id as model_id,
        models.uploaded,
        users.username,
        users.id as user_id
FROM models, users
WHERE (models.title LIKE ?
OR models.description LIKE ?
OR users.username LIKE ?)
AND models.userid = users.id";
```

Da søkestrengen kommer direkte fra brukeren, benytter vi bruker PDOs `prepare()`- og `execute()`-metode for å fylle inn søkestrengen (byttes ut med spørsmålsteget) for sikkerhet.

Deretter presenteres søkeresultatet i malen `search.php`, hvor brukeren får opp søkestrengen sin i tillegg.

5.2.5 Databasetilkobling

Som databaseabstraksjonslag har vi valgt PHPs innebygde PDO. Dermed slipper vi å inkludere eller installere eksterne pakker på serveren. For å administrere databasen i et brukergrensesnitt installerte vi `phpMyAdmin`.

I starten av prosjektet brukte vi en klasse for databasetilkobling som ble opprettet hver gang vi skulle gjøre operasjoner mot databasen. Dette medførte opp mot 5-10 tilkoblinger per sidevisning, noe som medfører mye unødvendig belastning mot serveren, og øker lastningstiden for klienten. Etter litt søking på internett fant vi en metode kalt *singleton*²⁰, som betyr at en klasse kun opprettes en gang, og er tilgjengelig gjennom hele livssyklusen til applikasjonen.

Vi laget en singletonklasse, `DBHandler`, og en som kobler til databasen, `DBConnect`. Her bruker man `DBConnect` for å koble til databasen, og metoden `get()` for å returnere databasetilkoblingen. Dermed slipper vi å koble til databasen hver gang en operasjon skal utføres, og sparer databasen for unødvendige tilkoblinger.

²⁰ <http://php.no/node/10>

I starten slet vi med å vise norske bokstaver korrekt fra databasen, da tegnene som ble returnert fra databasen var av typen ISO-8859-1, mens vi brukte tegnsettet UTF-8. For å konvertere til UTF-8 må spørringen `SET NAMES utf8` kjøres mot databasen før noen operasjoner blir kjørt. Vi fant ut at PDOs konstruktør kan ta i mot et argument, `PDO::MYSQL_ATTR_INIT_COMMAND`, hvor denne kommandoen kan kjøres. Da slipper vi å kjøre en ekstra spørring mot databasen.

Vi skulle gjerne kunne dokumentert ytelsesforbedringer etter at vi byttet til singleton, men hadde dessverre ingen nøyaktige verktøy for å måle dette på tjeneren. Men hele websiden ble mer responsiv etter utbyggingen, så vi så ingen grunn til å gå tilbake til den gamle metoden. I tillegg er det logisk at færre databasetilkoblinger øker ytelsen.

5.2.6 Opplasting av 3D-modell

Det var mye frem og tilbake hvordan opplastingsprosessen skulle håndteres. Siden vi ville ha den delt opp i forskjellige steg, og at ingenting skulle settes inn i databasen eller flyttes til brukerens mappe før modellen var publisert bød på mange problemer. Samtidig skulle brukeren ha mulighet til å gå tilbake i prosessen og endre data, og få feilmeldinger uten å miste data hvis han gjorde noe feil.

For å beholde informasjon bruker har gitt igjennom hvert steg, lagres alle data i en session-variabel igjennom prosessen. På serveren vår er varigheten på en session satt til 1440 sekunder, altså 24 minutter, som da gir brukeren 24 minutter på seg å gjennomføre en opplasting av modell. Denne session-variabelen blir tømt ved publisering, eller hvis bruker beveger seg vekk fra opplastingsprosessen før publisering.

Steg 1 – Last opp modell

Vi valgte å bruke PHP til opplasting av modellene. Dette er ikke optimalt for store filer, men i og med at vi har satt en grense på 10 mb for hver modell burde ikke dette by på noen problemer. Når modellen lastes opp, kjøres modellen igjennom fem sjekker:

1. At brukeren faktisk har valgt en fil
2. At filen har en gyldig filendelse
3. At filnavnet ikke inneholder ugyldige tegn (alt utenom det engelske alfabetet (26 tegn), tall, binde- og understreker er ulovlige) eller to punktum på rad. Konverterer eventuelle mellomrom til understreker.
4. At filen ikke er større enn maksimum størrelse for modeller
5. At brukeren faktisk har nok diskplass til filen

Hvis ingen av disse sjekkene returnerer falskt, flyttes den til en midlertidig mappe, og modellens filnavn og størrelse lagres i `$_SESSION` variabelen `uploadData`.

Steg 2 – Last opp teksturer

Det første som skjer er at det sjekkes etter data i `$_SESSION` variabelen `uploadData`, som ble angitt på det forrige steget. Hvis denne er tom, er det ikke lastet opp noen modell, og meningsløst å fortsette. Mye av dette steget skjer ved hjelp av Ajax (altså asynkront), og det viste seg å være en liten nøtt. Siden Uploadify, scriptet som laster opp filene, benytter seg av Flash for opplasting av filer, er det umulig å verifisere brukeren på grunn av en bug²¹ hvor Flash ignorerer alle `SESSION`- og `COOKIE`-variabler, og vi har ingen måte å finne ut hvilken bruker som er logget inn. Dermed får vi heller ikke sendt informasjon om de opplastede filene tilbake til det nåværende HTML-objektet. Det

²¹ <https://bugs.adobe.com/jira/browse/FP-1044>

vi gjorde var da å sende med filnavnet på 3D-modellen lastet opp i det første steget med hver tekstur, og lagre alle teksturer i den midlertidige mappen sammen med 3D-modellen.

Siden vi ikke vet hvor mange teksturer som er lastet opp, eller størrelsen på dem, må dette kalkuleres for å finne ut om brukeren har brukt opp sin diskplass. Det eneste vi vet om teksturene er hvor de plassert (/models/tmp/<modelname>/maps/ (\$dir)), og må da gå igjennom alle filer i denne mappen og få ut filstørrelsen for hver. For å få ut alle filene i mappen, bruker vi PHP-funksjonene glob, array_map og array_sum.

```
array_sum (array_map ('filesize', glob ($dir . '*')));
```

Kort fortalt henter glob filinformasjon om alle filene i mappen ved hjelp av wildcard uttrykket *, og array_map kjører filesize (en glob funksjon) på hver fil, som legger filstørrelsen for hver fil i et assosiativt array. Sist kjører array_sum igjennom det assosiative arrayet og legger sammen alle verdier.

Hvis bruker har nok plass, legges alle filnavnene og filstørrelsene for teksturene i SESSION variabelen textureData.

Steg 3 – Skriv inn beskrivelse

Alle modeller trenger en tittel og beskrivelse, samt en kategori. Brukeren fyller dette ut i et skjema, og kan i tillegg laste opp et beskrivende bilde (thumbnail) av sin modell.

Nok en gang sjekkes det etter at en 3D-modell faktisk er lastet opp, og siden feltene tittel, beskrivelse og kategori må være fylt ut, sjekkes det at disse feltene blir fylt. Hvis en thumbnail er lastet opp, må dette verifiseres. Siden dette er så å si de samme sjekkene som blir brukt om en 3D-modell (filnavn, filendelse, filstørrelse), blir den samme klassen brukt. Siden bildet som lastes opp hovedsakelig skal bli brukt som en beskrivelse av modellen i modellgalleriet og detaljvisningen for en modell, må det skaleres til korrekt størrelse. Les mer om dette i Skalering av miniatyrbilde på side 52. Av sikkerhetshensyn vil bildet bli lagret i databasen, men vi tar også vare på originalbildet av hensyn til videre utvidelser. Dette lagres da i

/www/uploads/<brukernavn>/<bildenavn>/<bildenavn>.ext. For å ivareta sikkerheten forhindrer vi opplisting av filer i undermappene her, og hindrer at kode på tjeneren kan kjøres hvis noen klarer å laste opp en fil som inneholder ondsinnet kode. Dette gjør vi via en .htaccess-fil, som ser slik ut:

```
# disable directory browsing
Options All -Indexes
# secure htaccess file
<Files .htaccess>
  order allow,deny
  deny from all
</Files>
# secure directory by disabling script execution
AddHandler cgi-script .php .pl .py .jsp .asp .htm .shtml .sh .cgi
Options -ExecCGI
```

Vi hindrer også tilgang til selve .htaccess-filen.

Som sagt skal miniatyrbildet lagres i databasen, sammen med selve filnavnet på bildet, og bildets MIME-type, men dette skal ikke skje før publisering. Derfor lagres alle verdier som er fylt ut i SESSION-variabelen `uploadData`.

Steg 4

Når modellen skal publiseres, tas all informasjon fra session-variabelen og settes inn i tabellene `models` og eventuelt `textures` (hvis det er lastet opp teksturer). Vi har som mål at alle data som tas i mot fra bruker og skal inn i databasen blir klargjort av PDO, og dermed eliminerer ondsinnet kode og forhindre SQL injections. Dette medførte visse problemer i og med at vi kun vil sette inn data som var fylt ut (for eksempel hvis et miniatyrbilde ikke ble lastet opp, er det ikke vits å oppdatere feltene for miniatyrbilder i databasen). Men, siden alle data er lagret i det assosiative arrayet `$uploadData`, som er konstruert på denne måten: `databaseFelt => verdi`, har vi mulighet til å splitte alle verdiene og bruke de i den forberedte spørringen.

Vi tok i bruk PHP-funksjonen `implode()`, som splitter opp et array, og lar oss angi ”limet” som skal være i mellom hver del. For å hente feltene hvor dataene skal inn i, deler vi opp nøklene til det assosiative array-et, som settes inn i spørringen ved hjelp av `array_keys()`. Deretter blir like mange spørsmålstegn plassert i `VALUES`, ved bruk av `array_fill()`, som teller alle verdiene i `$uploadData`, og plasserer et spørsmålstegn for hver verdi. Da vet PDO hvor mange verdier som skal settes inn. Så klargjøres spørringen, og PDO trenger kun verdiene som skal settes inn. Da bruker vi `array_values()`, som henter ut alle verdiene i array-et, og spørringen er kjørt mot databasen.

```
$sql = "INSERT INTO models
(
  " . implode(' ', array_keys($_SESSION ['uploadData'])) . " ...
)
VALUES
(
  " .
    implode(", ", array_fill(0, count($_SESSION ['uploadData']), '?'))
    ...
  " ,
)";
$stmt = $db->prepare($sql);
$stmt->execute(array_values($_SESSION ['uploadData']));
```

Hvis spørringen var vellykket, flyttes et eventuelt bilde (altså den originale versjonen til miniatyrbildet) og selve 3D-modellen med alle teksturer til den endelige mappen. Filnavnene og filstørrelsene til eventuelle teksturer settes så inn i databasen, og brukerens diskplass oppdateres. Så slettes `$uploadData` og `$textureData`, da alle data er lagret i databasen.

Det siste som skal skje er at 3D-applikasjonen kalles, og 3D-modellen og teksturene konverteres til et format 3D-applikasjonen forstår, og brukeren forflyttes til detaljvisningen for modellen.

5.2.7 Eksterne skript

5.2.7.1 Skalering av miniatyrbilde

Denne funksjonen kommer fra Øivind Kolloen, lærer ved HiG, som ble brukt i faget WWW-teknologi. Vi har innhentet Kolloens tillatelse til bruk av denne funksjonen.

Opprinnelig fungerte funksjonen på den måten at den tok i mot to argumenter, selve bildet og ønsket bredde (x-verdien). Deretter skalerte funksjonen bildet til x-verdien, og returnerte et nytt bilde. Vi ønsket en funksjon som gjorde dette, men i tillegg tok i mot et argument til, ønsket høyde (y-verdien). Dermed la vi til funksjonalitet, så hvis bildet hadde høyere x-verdi enn y-verdi, ble bildet skalert til ønsket bredde, og visa versa. Samtidig ble det opprinnelige størrelsesforholdet beholdes, slik at bilde ikke ble forvrengt.

5.2.7.2 Generering av CAPTCHA

CAPTCHA-skriptet er utviklet av Jose Rodriguez, og er lisensiert under GPLv3. Dette er ikke modifisert på noen måte.

Hjemmeside for CAPTCHA-skriptet: <http://code.google.com/p/cool-php-captcha/>

5.2.7.3 Formatering i nyhetsartikler

For å gi mulighet til å formatere tekst i nyhetsartiklene, la vi inn `Textile`. Denne produserer validert XHTML-kode av ren tekst med enkel formatering. For eksempel konverteres `*uthevet*` til `uthevet`.

Hjemmeside for `Textile`: <http://textile.thresholdstate.com/>

5.3 GUI

5.3.1 Forarbeid og menneskelig interaksjon

5.3.1.1 De første tankene

Design av en GUI kan variere i det uendelige. Det finnes derimot noen måter å identifisere de på. Man har i følge Nigel og Jenny Chapman²² to ekstremer av en GUI.

- **Tradisjonell stil:** På den ene siden har man den tradisjonelle stilen. Dette er den stilen du møter mest. Lett gjenkjennelig fordi den tar i bruk knapper, menyer, stiler og valg som er typiske for det systemet du sitter på. Disse vil du også finne mye av på nett. Et typisk internett skjema som du fyller ut på nett er et av disse eksemplene på en GUI som er av denne typen. Men som med alt annet har denne typen GUI både positive og negative sider:
 - **Positive:** Brukere kjenner denne typen GUI siden den er tradisjonell, og brukt. En bruker som er erfaren på det systemet du utvikler en GUI for kan sannsynligvis ta i bruk programvaren uten å måtte sette seg inn i systemet og lese eventuelle manualer.

²² Chapman Nigel, Chapman Jenny (2004). *Digital Multimedia*. West Sussex, John Wiley & Sons Ltd.

- **Negative:** Det negative ved denne typen GUI er det at man ikke kan tilpasse den slik man vil. Det finnes tilfeller der man gjerne vil tilpasse GUIen slik man vil, dette gjelder spesielt hvis du vil prøve å tilpasse GUIen i henhold til den grafiske profilen for bedriften eller kunden du utvikler applikasjonen for.
En kunde som ikke kjenner til plattformen og systemet du utvikler for kan ha problemer, hvor han/hun kanskje heller hadde gjort det bedre med en spesialtilpasset GUI for den aktuelle applikasjonen.
- **Utradisjonell stil:** Ved bruk av denne stilen går utvikleren for å gi GUIen et personlig preg, og mer tilpasser den bruken den er laget for. Denne typen GUI er det du vil se i mange online applikasjoner, og da gjerne spesielt i programmet som nettopp det vi skal utvikle. Selvsagt har jo også denne typen GUI sine positive og negative sider.
 - **Positive:** Utviklere kan tilpasse GUIen og den grafiske profilen til å matche det en eventuell oppdragsgiver ønsker.
En annen positiv side er at hvis dette blir gjort riktig kan en slik GUI være bedre enn en GUI basert på den tradisjonelle stilen. Dette fordi disse knappene kan være bedre tilpasset den bruken de er utviklet for enn de standard knappene som er tilgjengelig.
 - **Negative:** Hvis det ikke blir gjort riktig kan en spesialtilpasset GUI gjøre en bruker mer forvirret enn å hjelpe. Det vanskelige med denne type GUI er å få til GUIen til å bli brukevennlig, og samtidig selvforklarende.

For å lage en GUI som skulle fungere, måtte vi først finne ut hva slags kontroller vi egentlig trengte. For å kartlegge dette gikk vi tilbake til kravspesifikasjonen på side 20. Vi startet så på det som da skulle bli de første utkastene av hva slags kontroller og knapper man kunne trenge på selve GUIen. I første omgang så vi på navigasjons-pod'en (Figur 4.12 side 38), på denne kontrollen er det 8 forskjellige valg for hvilken retning man vil bevege seg. Da illustrert ved piler som er snudd og plassert rundt den sirkelen man har å bruke til navigasjon. For at en bruker skal kunne bruke disse må brukeren skjønne at men går ut i fra at midten på denne navigasjonskontrollen representerer det aktuelle punktet du er på nå for å kunne bevege deg fremover eller bakover.

5.3.1.2 Kompleksitet og menneskers oppfattning

For at et produkt i dag skal oppfattes som ett bra produkt, må man hele tiden ta hensyn til de forskjellige nivåene av oppfattning, og i hvilke sammenhenger de skal brukes.

Design og utvikling av et objekt, en stol, et dataprogram, eller et sosialt nettverk kan selvsagt ikke angripes på samme måte. Det er helt andre faktorer som melder seg, og spiller inn på de forskjellige nivåene av design.

Totalt er det kjent 6 nivåer av kompleksitet for design og utvikling av ett produkt (se Figur 5.1), de følgende nivåene er:

- **Ecology:** Gjensidig avhengighet av levende ting, for bærekraftig design.
- **Anthropology:** Den menneskelige tilstanden, brukes til global design.
- **Sociology:** Måten mennesker forholder seg til hverandre, brukes til design av nettverk.
- **Psychology:** Måten mennesket tenker. Brukes for design av interaksjon mellom menneske og datasystemer.
- **Physiology:** Måten kroppen fungerer. For design av fysisk interaksjon mellom menneske og maskin.
- **Anthropometrics:** Størrelsen på mennesker, brukes da til design av fysiske objekter.

Modellen over er hentet fra *Moggridge, 2007, side 652*.

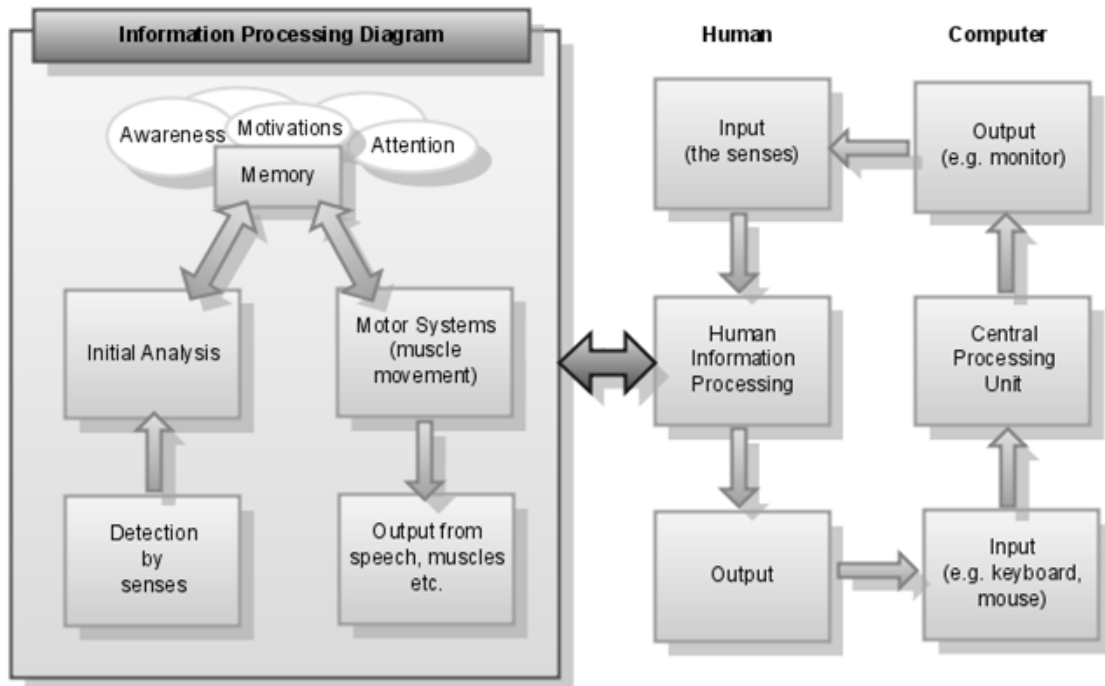
Så for vår del så vil vi ligge på nivå tre av denne stigen. Nemlig under ”psychology”, eller da også kjent som psykologi, med andre ord er dette noe vi måtte ta hensyn til under design av GUIen for 3D applikasjonen.

Dette vil med andre ord si at vi som utviklere og designere av GUIen må ta hensyn til menneskelige faktorer for hvordan de vil oppfatte GUIen i en applikasjon. Dette er også kjent som kognitiv psykologi. Så lenge man nå skal designe funksjonene som ett program skal gjennomføre har man ikke lengre kun det fysiske og menneskets biologiske forutsetninger å ta hensyn til, men også det psykiske og hvordan hjernen fungerer når den er satt ovenfor de valgene som kan introduseres i ett dataprogram.



Figur 5.1: Illustrasjon av de forskjellige nivåene av design, basert på de forskjellige nivåene av menneskelige faktorer.

Figuren nedenfor er hentet fra *Benyon, Turner, & Turner, side100-101*.



Figur 5.2: Ett diagram som viser hvordan informasjon blir behandlet når det sendes mellom menneske og datasystemer. Det viser også hvordan mennesket oppfatter og behandler informasjon før det gir ett svar.

Som man kan se fungerer en datamaskin og mennesket ganske så likt med hverandre, begge får inputs, og begge prosesserer dette, får så å komme med et output.

Den store forskjellen ligger under det punktet hvor man prosesserer informasjonen som man får igjennom input. I en data er det relativt rett frem, der den behandler fakta og programinformasjon. Hos oss mennesker er denne prosessen litt vanskeligere, og det er denne prosessen som gjør det hele mer innviklet for oss da vi skulle lage en GUI.

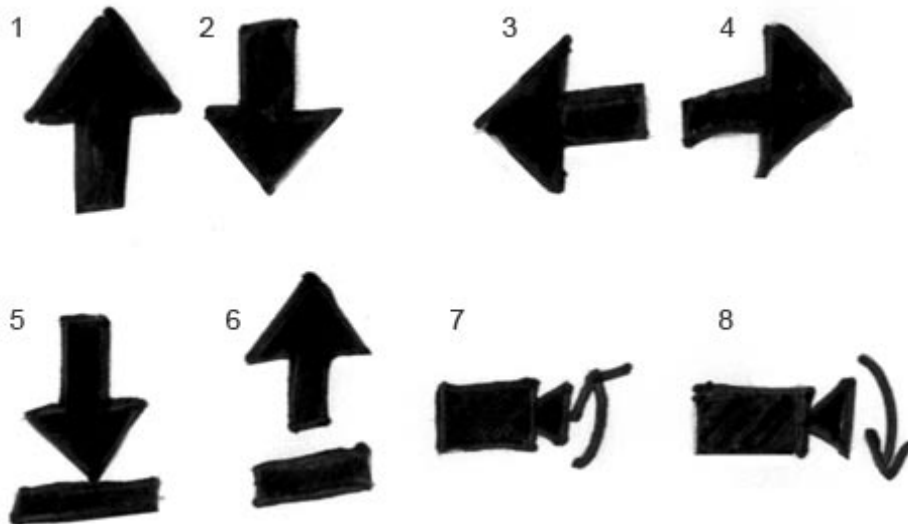
I tillegg til det estetiske må vi også ta hensyn til målgruppen, og hva slags forventninger og kunnskaper de sitter inne med fra før av. Ifølge diagrammet (Figur 5.2) sanser man først en input, og deretter begynner behandlingen.

- **Detection by senses:** Mennesket oppfatter noe enten det er ved syn eller en annen sans.
- **Initial Analysis:** Det er her hvor man begynner arbeidet med å analysere det man ser (eller på annen måte oppfatter), og man begynner så å gå for neste steg av informasjonsbehandlingen.
- **Memory:** Det er her det begynner å bli interessant. Det er her den menneskelige sansen begynner å gjøre seg gjeldene. Her begynner brukeren av systemet å hente opp minner fra tidligere erfaring av samme, eller liknende program, henter opp informasjon og vaner, og brukeren vil også begynne å bruke såkalt konsekvens orientert tankegang. Det er på dette punktet at GUIen kan enten feile, eller skåre mange poeng.
- **Motor Systems:** Fysiske bevegelser som da begås av brukeren for å ta det valget han kom frem til under det forrige punktet.
- **Output from speech, muscles, etc:** Resultatet av bevegelsen. Da gjerne tastetrykk eller annen form for input på PCen.

Som utvikler av ett program eller en GUI må man spesielt ta hensyn til steg nummer tre ovenfor. En person som har mye erfaring innen 3D og navigasjon fra før av vil mest sannsynlig mestre dette fortere enn en som åpner programmet for første gang. Derfor må en GUI både være brukervennlig og selvforklarende uten at man må behøve å levere en stor brukermanual ved siden av for at en bruker skal kunne ta i bruk applikasjonen.

5.3.2 De første utkastene

Under utviklingen av GUIen bestemte vi oss først for hvilke kontrollen man kunne trenge for å bevege seg i ett 3D-miljø. Dette her er de første skissene vi kom frem til.

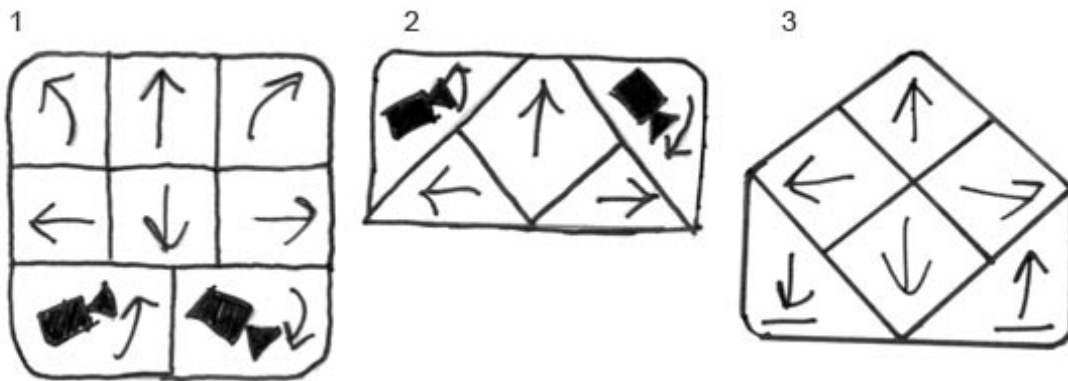


Figur 5.3 De første utkastene til kontrollene for GUIen.

Som du kan se på Figur 5.3 har vi 8 forskjellige symboler. Hver med sin egen funksjon. Nedenfor ser du en liste som beskriver hva hver enkelt av disse knappene gjør:

1. **Fremover:** Dette vil ta brukeren fremover applikasjonen.
2. **Bakover:** Det motsatte av fremover, dette vil føre brukeren bakover.
3. **Venstre:** Skal snu brukerens kamera til venstre.
4. **Høyre:** Skal snu brukerens kamera til høyre.
5. **Vertikal bevegelse ned:** Skal flytte kamera rett ned.
6. **Vertikal bevegelse opp:** Skal flytte kamera rett opp.
7. **Tilt kamera bak:** Tilter kamera bakover, og gjør at brukeren kan se oppover.
8. **Tilt kamera opp:** Tilter kamera opp, og gjør at brukere kan se nedover.

Dette var da de første elementene av en eventuell GUI. Deretter gikk vi videre med disse, og begynte å sette de sammen for en helhet. Dette var for å skaffe en samlet kontroll for bevegelse av kamera. Nedenfor har vi da plukket ut 3 tidlige utkast for design av GUI.



Figur 5.4 Tre tidlige skisser av kontroll panelet for GUIen til 3D applikasjonen.

1. **Det første utkastet:** Her er de fleste knappene med, inkludert 2 nye knapper som bare ble kastet på. Disse skulle forestille diagonal bevegelse, inspirert av lignende knapper som navigasjonspoden (Figur 4.12 side 38). Denne utgaven hadde ett problem vi la merke til ganske tidlig, den tok opp for stor plass i selve applikasjonen.

2. **Det andre utkastet:** Etter å se på hvordan det første utkastet var følte vi den var litt for ”firkantet”, og litt for stor. Som ett resultat av dette kom vi med utkast nummer to. Denne var en minimalisert versjon av en større GUI. Denne versjonen var den første der vi tok i bruk diagonale linjer eller samtale med oppdragsgiver der vi fikk klarert hva vi hadde av begrensninger. Den tidligere informasjonen vi fikk var at alle elementer måtte være firkantede. Men da vi senere snakket med oppdragsgiver fikk vi vite at dette ikke lenger var nødvendig, og vi fikk friere tøyler når det gjaldt design av GUI.

I denne GUIen var ideen at vi ikke trengte bakover, og heller ikke diagonal bevegelse. Man kunne ganske enkelt snu seg rundt, og bevege seg dit man ville ved de vanlige knappene. Problemet med denne GUIen var at det var for lite valgmuligheter, noe som igjen hindret brukeren i den friheten vi ville ha.

3. **Tredje utkastet:** Dette utkastet kom så å si samtidig med det andre utkastet og skulle være ett alternativ til dette. Forskjellen er her som du ser at vi har tatt med bevegelse bakover, og fjerne kontrollene som skulle styre kamera. Dette fordi at vi her tenkte oss at så lenge en bruker kan bruke en mus til å trykke på knappene kan man også bruke musen til å ganske enkelt trykke på vinduet der selve modellen vises, og deretter snu kamera slik man vil der. Denne typen hadde også dessverre noen negative sider, da denne typen navigasjon og rotasjon av kamera kunne bli litt vanskelig hvis brukeren satt med en bærbar PC, og en touchpad.

I tillegg til denne forandringen har vi også lagt til knappene for vertikal bevegelse her. Disse ble da plassert under venstre og høyre, av den enkle grunn at det var her de passet best. Vi kunne ikke plassere de ovenfor på grunn av det ekstra rommet som ble skapt under på grunn av pilen bakover.

5.3.3 Forandringer etter hvert

5.3.3.1 Uforutsett problem

Etter hvert som vi satte sammen forskjellige GUI elementer la vi jo merke til hva som fungerte og ikke. I ett møte med veileder kom det blant annet opp at veileder trodde pilene som skulle vise at man snudde seg til venstre og høyre (se Figur 5.4) egentlig var knapper for å bevege seg rett til venstre og høyre. For å gjøre det tydelig at det ikke var dette som var planen var det tydelig at vi måtte forandre disse knappene. For å kompensere for dette kom vi da opp med de følgende pilene på

kontroll panelet.

Disse pilene var de som da tok over for høyre og venstre. Som man klart kan se er det ikke lenger noen tvil om hva disse pilene betyr.

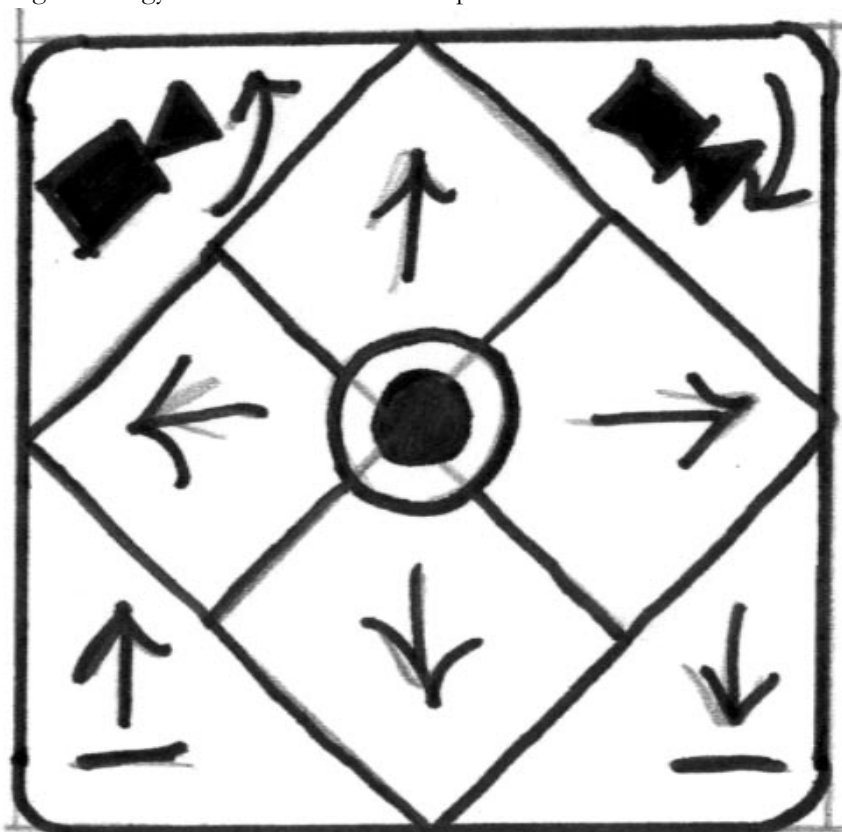


Figur 5.5 Skissene for knappene som tok over for høyre og venstre.

5.3.3.2 Tilbake til start

Etter vi hadde gått over kravspesifikasjonen over hvilke kontroller som måtte være med i GUIen så fant vi ut at vi måtte ha minimum 8 forskjellige knapper dersom alt skulle kunne styres ved hjelp av kontrollpanelet.

For å fikse dette så vi over de tidligere skissene og fant flere typer som vi hadde regnet som for store og ubrukelige til å begynne med. En av de vi la spesielt merke til var denne:



Figur 5.6 En tidligere skisse som vi hadde forkastet på grunn av størrelsen.

Denne skissen var en av de vi lagde for å kompensere for det at vi ikke kunne trykke på selve hovedvindu får å styre kamera rundt. Det vi gjorde her var å ta utkast nummer tre i Figur 5.4, for

deretter å legge på kamera kontroll på toppen av dette. Dette ga oss en grei kontroll, men likevel ikke helt den friheten vi ville ha av kamerakontroll.

Hvis du da ser på midten av Figur 5.6 har vi lagt til noe som skal illustrere en slags slags joystick, denne var ment å styre selve kamera i alle retninger.

Tanken bak denne joysticken var at vi skulle la brukeren få full frihet til å kunne kontrollere kamera dit man ville, da ved hjelp av denne joysticken. Dette kunne nok fungere det, og kunne nok vært en god ide, hadde det ikke vært for at vi da kunne tatt i bruk noe så enkelt som det systemet tiltenkt i det tredje utkastet i Figur 5.4, der brukeren kan trykke hvor han/hun vil i applikasjonen for så å styre kameraet dit man vil.

En annen negativ side ved denne joysticken var ganske enkelt, den var for liten.

Vi bestemte oss for å fjerne denne joysticken og da sitter vi igjen med ett kontroll panel som oppfyller de kravene definert i 2.2.3 på side 20. Så det var da denne vi baserte oss på.

Det resultatet vi endte opp på kan du se her i figur Figur 5.7. Dette kontrollpanelet har muligheten til å bevege seg frem og tilbake, samt snu seg til venstre og høyre. I tillegg har man også muligheten til å tilte kamera opp og ned, samt bevege seg vertikalt opp og ned. Dette ville gi brukeren en mulighet til å fritt kontrollere kamera med ett kontrollpanel.



Figur 5.7 Det endelig kontrollpanelet vi endte opp med.

5.3.4 Sammensetting av GUI

5.3.4.1 Begrensninger i størrelse

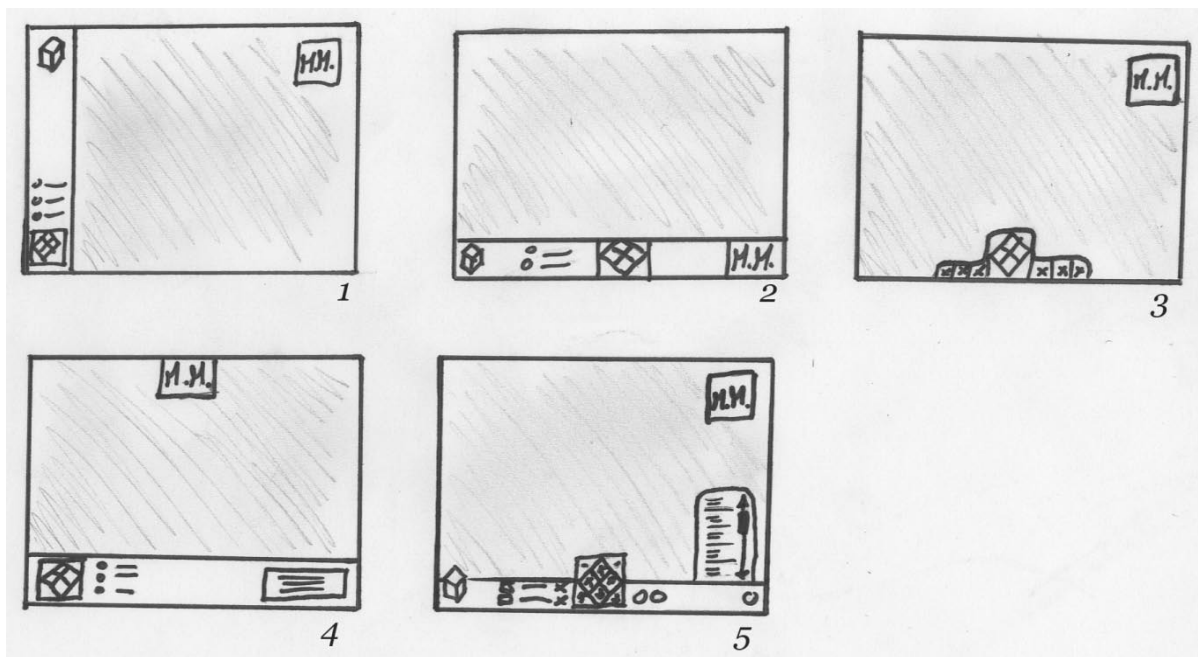
Selve GUIen er jo mer enn bare kontroll panelet, derfor måtte vi også stå for utformingen av denne. En ting vi måtte ta med i beregningen og utformingen av en GUI er størrelsen på selve applikasjonen. Selve vinduet som applikasjonen skal åpnes i er på 640 x 480 pixler. Dette vil med andre ord si at alt vi skal implementere må passe i forhold til størrelsen vi har tilgjengelig.

5.3.4.2 Grafiske elementer i applikasjonen

Elementer som skal legges inn i GUIen er følgende:

- Kontrollpanel (se Figur 5.7 ovenfor).
- Logoen (se Figur 4.4 side 28).
- Minimap
- Knapper for hjelp, snapshot og filmstripe.

For å tilpasse dette laget vi skisser for å prøve ut dette, og finne ut hvor vi kan plassere disse elementene. Vi har skannet inn noen av disse skissene og man kan tydelig se forandringen som har skjedd underveis.



Figur 5.8 Skisser av hele GUIen.

1. Første utkast, da inspirert av golfbane visningen til oppdragsgiver som du kan se på Figur 4.11. Med område for navigering og logo på venstre side av GUIen. Mens Selve 3D visningen ligger på høyre side av applikasjonen. Mens du kan se minimapet plassert øverst i høyre hjørne, dette for å få det litt vekk fra resten, men remdesles synlig.
2. Her har vi flyttet alt ned til bunnen av vinduet i applikasjonen. Logo plassert synlig i venstre hjørne, med kontrollpanel i midten, og minimap til høyre.
3. En tidlig komprimert utgave av GUIen. Dette utkastet ble til fordi vi følte at utkast nr. 2 tok for stor plass, spesielt fordi den stripa vi måtte avse i bunn av applikasjonen ble for dominerende og høy. Enda ett tiltak for å spare litt plass nederst i applikasjonen var å plassere minimapet tilbake oppe i høyre hjørne.
4. Under samtale med oppdragsgiver kom det frem at oppdragsgiver også ønsket ett felt der informasjon om modellen, eller annen info kunne vises. Dette utkastet ble da altså til ved å gå ett skritt tilbake, ved å flytte det meste ned i bunen av applikasjonen, for så å plassere en info boks nederst i høyre hjørne, og heller flytte minimapet til midt på i toppen av skjermen. Dette ville heller ikke fungere slik vi ønsket, og dermed ble dette også forkastet.
5. Igjen tok vi utgangspunkt i utkast nr. 2. Alle elementer er da flyttet ned igjen, med unntak av minimapet, som nå er på plass oppe i høyre hjørne, der det skal kunne minimaliseres for å ta mindre plass.

Forskjellen mellom dette utkastet og utkast nr. 2 er at her er den stripen vi har avsatt på bunnen av skjermen senket, og ligger på halveis opp på kontroll panelet, samt logoen ligger lengre opp slik at de øvre hjørnene blir da på linje med selve linjen som deler applikasjonen. For å gi oppdragsgiver muligheten til å vise den informasjonen han/hun vil har vi da valgt å løse dette ved å da la ett tekst felt ligge usynlig under det hele inntil det er kalt på av brukeren av applikasjonen. Da det ikke er nødvendig med denne informasjonen hele tiden. Dette var da den skissen vi gikk for, og under kan du se resultatet av første utkast.



Figur 5.9 Tidlig utgave av GUIen satt sammen i Photoshop. Dette var det som skulle bli malen for det endelige oppsettet.

Dette var da det første vi satte sammen for å kunne bruke dette til en brukertest. Denne GUIen oppfyller nesten alle kravene definert i kravspesifikasjon i punkt 2.2.3 på side 20. Det som mangler her er en mulighet til å ta bilder og legge disse i ett album for senere bruk.

6 Testing

6.1 Brukertest

Vi valgte å gjennomføre en brukertest av vårt system. Brukertestingen besto av fire steg, hvor det første var forberedelser.

Forberedelser

Vi bestemte oss ganske tidlig for å gjennomføre en såkalt "Ansikt til ansikt"-test, hvor testpersonene gjennomførte en eller flere forhåndslagde oppgaver, også kalt scenarioer, på vår teststasjon. Vi utarbeidet tre slike scenarioer, som vi skrevet ut før brukertesten. Disse ble utviklet for å teste mest mulig av systemet, slik at vi fikk tilbakemeldinger på funksjonalitet. Vi satte som mål at alle personene skulle gå igjennom minst to scenarioer, men på grunn av tidsmangel ble det noen som kun gjennomførte et scenario.

Rekruttering

Vi måtte skaffe mennesker som kunne gjennomføre brukertesten, og som ikke hadde (for) stor kjennskap til vårt system. Siden målgruppen vår består av forholdsvis unge personer med gode datakunnskaper, så vi ikke på det som et problem å bruke studenter som testpersoner. Vi valgte å teste seks mer eller mindre tilfeldige personer (også kalt "Hallway testing"²³). Grunnen til det lave antallet er at man ifølge brukervennlighetseksperten Jacob Nielsen får så og si all den tilbakemeldingen man trenger fra fem personer²⁴, og at vi rett og slett ikke hadde tid eller resurser til å ha flere testpersoner. Vi burde heller brukerteste flere ganger i løpet av utviklingsprosessen. I løpet av brukertesten var dette noe vi merket oss også, etter 4-5 testpersoner kom det frem lite nytt, og de samme problemene ble rapportert mer og mer.

Gjennomføring av test

Alle testpersonene ble før brukertesten fortalt at det er produktet som skal testes, og ikke personen. Dette for å gjøre personen mer sikker, og ikke være redd for og "dumme" seg ut. De ble også fortalt at de skulle tenke høyt under hele prosessen, slik at vi hele tiden forsto hva personen foretok seg. Hvis det ble stille over en lengre periode under testen, ble testpersonen minnet på dette.

Jon Espen var sekretær, og noterte reaksjoner og viktige handlinger fra testpersonen. Testen foregikk på vårt kontor ved HiG, A036, og laptopen til Jon Espen ble benyttet som testmaskin. Før hver test ble PC-en "nullstilt", det vil si at alle vinduer på PC-en ble lukket, bortsett fra Microsoft PowerPoint, nettleseren Google Chrome ble åpnet uten noen faner, og alle data i nettleseren ble slettet. Dette ble gjort for å sikre at alle testene ble gjort uten forstyrrelser og at testene ble så like som mulig.

Etterarbeid

Etter gjennomført test diskuterte gruppen resultatene, og alle steg fra hver testperson ble dokumentert. Erfaringene ble benyttet i videre designutvikling.

²³ http://en.wikipedia.org/wiki/Usability_testing

²⁴ "Why You Only Need to Test with 5 Users", Jacob Nielsen: <http://www.useit.com/alertbox/20000319.html>

6.2 Erfaringer fra brukertesten

Vi fikk mange gode tilbakemeldinger på prototypene våre, og lærte mye om brukertesting. En ting vi kunne gjort annerledes var å legge mer tid og arbeid i scenarioene. Å liste opp hvert steg som testpersonen skulle gjøre ble for enkelt, og ga generelt liten utfordring. Vi kunne heller kuttet ut stegene, og kun hatt en beskrivende tekst for hvert scenario. Allikevel kunne dette by på problemer i og med at vi kun hadde prototyper som ikke var veldig funksjonelle, og hvor ting måtte gjøres i en rekkefølge.

Det var vanskelig å få med seg alle tilbakemeldingene fra en bruker, og vi kunne med fordel ha montert et kamera som observerte alle testpersonene. Vi skulle gjerne gjennomført flere brukertester etter endringene som ble gjort basert på erfaringene fra denne testen, men dette rakk vi dessverre ikke.

6.3 Endringer etter test

6.3.1 Websiden

Mange av testpersonene slet med å finne opplastningsknappen, og trodde den kun var en del av grafikken i toppen (headeren). Mange klaget også på plasseringen, da de mente den befant seg i "blindsonen" på websiden. Vi gjorde litt undersøkelser på området, og fant flere artikler som konkluderte med nogenlunde det samme, blant annet en fra Jacob Nielsen²⁵. Et av stedene som fikk minst oppmerksomhet var øverst til høyre, med mindre det inneholdt elementer som tiltrakk oppmerksomheten, for eksempel ikoner. Dermed kan vi konkludere med at både plasseringen og utformingen på opplastningsknappen var dårlig utført. For å endre på dette flyttet vi opplastningsknappen ned under toppgrafikken, slik at den havnet i dette synsmønsteret. Vi laget også en CSS-stil som ga knappen et mer "knappelignende" utseende, ved å legge på en gjenskinns- og skyggeeffekt. I følge Stephen P. Anderson i artikkelen *In Defense of Eye Candy*²⁶ hjelper det hvis elementer minner om fysiske elementer fra den ordentlige verdenen. Alle knapper på websiden fikk da denne knappestilen, for å skape en lik stil og gjøre knappene gjenkjennbare.

"Frem"- og "Tilbake"-knappene i opplastingsskjemaet ble plassert lengre i fra hverandre, og fikk en pil for å indikere hvilken vei man gikk. Høyre er da fremover, mens venstre er bakover.

Etter at vi flyttet denne knappen ble det veldig tomt, i headeren, og vi bestemte oss for å gå tilbake til en av de tidligste skissene og flytte brukermenyen helt til toppen. For å ikke gå i samme fella igjen la vi på et ikon ved hvert element, samt ga den en annen bakgrunn.

I opplastingprosessen hvor man angir hvem som skal ha tilgang til modellen var det mange som slet med å finne det siste punktet; passordbeskytt. Mange savnet også en beskrivelse av hvert punkt. Derfor laget vi ved hjelp av jQuery en funksjon som viste en informasjonsboks ved hvert element, og endret seg etter hva man valgte.

Noen slet med punktet hvor man velger hvilke brukere som skal ha tilgang til modellen, da det var tungvindt å velge med CTRL-tasten. Vi fikk ikke tid til å utbedre denne funksjonen, men kom med skisser om hvordan det kunne utbedres. Les mer i vedlegg D, Rapport, brukertesting side 105.

Tekststørrelsen i alle skjemaer ble økt, sammen den røde stjernen som indikerer nødvendige felt.

²⁵ http://www.useit.com/alertbox/reading_pattern.html

²⁶ <http://www.alistapart.com/articles/indefenseofeyecandy/>

Alle skjemaer hvor man må godkjenne retningslinjene før innsendelse inneholder nå en link til retningslinjene som åpnes i et eget JavaScript-vindu, for å forhindre at data brukeren har skrevet inn går.

En link til registrering av konto ble plassert under "Logg inn"-skjemaet i sekundærinnholdet, da mange forventet dette.

Noen ville ha mulighet til å bruke e-postadressen sin som innloggingsnavn, men det ble ikke prioritert da noen ville ha mulighet til å registrere et brukernavn. Allikevel er dette noe vi skal vurdere i fremtiden.

6.3.2 GUIen

For å teste GUIen til applikasjonen brukte vi en 3D-modell av ett hus, der vi så rendret ett bilde, satte dette inn i PowerPoint, og lagde ved hjelp av dette ett step-by-step scenario for testpersonene å følge i PowerPoint.

6.3.2.1 "Clipping" og "Wireframe"

Testpersonene visste ikke, eller forstod ikke hva "clipping" og "wireframe" var. Det ble også nevnt at eventuelle tooltips kanskje kunne løse dette problemet.

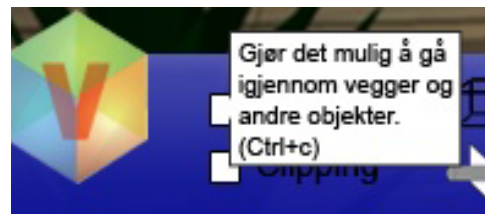
Dette var ett problem vi nesten forutså selv, men vi ville teste dette med folk, for å se om vi måtte erstatte de, eller om det fungerte. Det viste seg under testen at selv om testpersonene ikke visste i forkant hva dette var skjønnte de det relativt fort etter å ha prøvd knappene. "Clipping" var fremdeles litt uklart, mens "wireframe" ble tydelig med en gang de testet dette.

For å rette på dette problemet tok vi kontakt med oppdragsgiver, som var enig med at å legge til eventuelle tooltips på disse.

Figur 6.1 viser ett eksempel på hvordan dette kan løses. Det som må gjøres er å legge til en funksjon eller klasse i Java Programeringen som lager ett tooltip når musepekeren er over dette området over lengre tid.

6.3.2.2 GUI og størrelsen

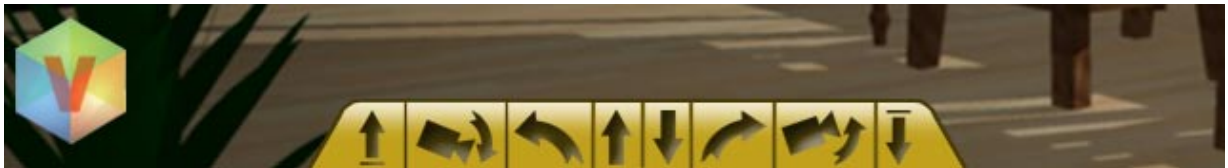
Testpersonene ønsket også om mulig en mindre GUI, da de følte at elementer av GUIen tok opp for stor plass på skjermen.



Figur 6.1 Eksempel på hvordan man kan la tooltips hjelpe brukeren.

Som nevnt over i punkt 5.2.3 side 46 følte vi at denne typen oppsett av GUI kunne ta opp for stor plass av selve applikasjonen. Dette var da noe vi fikk bekreftet av brukertesting, selv om vi da hadde gjort den betraktelig mindre enn det første utkastet. Ett forslag fra en av testpersonene var å la brukeren kunne bytte til en mindre GUI. Dette ville da være en minimal GUI med kun navigasjonsknappene tilgjengelig, mest til bruk for den såkalte "superbrukeren".

For å tilrettelegge for dette laget vi en GUI som da bestod kun av navigasjonspanelets knapper, men da lagt i en linje i bunnen av applikasjonen for å frigjøre så mye som mulig av selve viewporten til applikasjonen. Dette ville gi en bruker mere plass til å se selve applikasjonen, men fremdeles la brukeren styre 3D applikasjonen med musen. Figuren nedenfor viser hvordan vi har valgt å implementere dette i GUIen.



Figur 6.2 Viser hvordan en minimalisert versjon av kontroll panelet ser ut. Denne ble laget på bakgrunn av tilbakemeldinger under brukertesting.

7 Endelig produkt

Etter at vi gjennomførte brukertesting gikk vi over resultatene og kom med forandringer til både websiden og GUIen til applikasjonen. Nedenfor kan du se forandringer vi kom frem til.

7.1 Websiden

The screenshot shows the Visualisere website interface. At the top, there is a green header with the logo and the word "VISUALISERE". To the right of the header, there are links for "Hei jonespen", "Kontrollpanel", "Adminpanel", and "Logg ut". Below the header is a dark navigation bar with links for "HJEM", "MODELLER", "HJELP", and "OM", along with a search bar. The main content area is divided into several sections: a "FEATURED" section with a post titled "Pengebingen" by Jon Espen Kvisler, a "LAST OPP MODELL" button, a "NYHETER" section with a news item dated 17.05.2009, and a "Modeller" section. The "Modeller" section displays a grid of model thumbnails, each with a placeholder image and the text "INTET BILDE". Below each thumbnail, there is a link to "Noe" and the text "av jonespen". At the bottom of the page, there is a dark footer with the text "Retningslinjer" and "© Visualisere 2009".

Her er et screenshot av websiden i skrivende stund (20.mai 2009). Utviklingsversjonen kan sees på <http://www.visualisere.org/dev/php/> (og forhåpentligvis <http://www.visualisere.org/> etter presentasjonen av prosjektet).

7.2 GUI og 3D applikasjon

7.2.1 Farger

Den endelige versjonen har da også fått farger. Disse fargene skulle i utgangspunktet være de samme som vi kom frem til i den grafiske profilen (se punkt 4.2 på side 26). Disse fargene fungerer på websiden, men vi kom frem til at disse fargene ville bli for blasse og utydelige i sammenheng med applikasjon og de 3D elementene som vil være i viewporten.

Det som ble gjort for å løse dette var at vi tok fargene fra den grafiske profilen. Deretter senket vi lysintensiteten på disse med ca. 30 %, dette gjorde fargene betydelig mørkere, mens de ble mer fremtredende på selve GUIen.

7.2.2 GUI for visning av hus og liknende modeller

7.2.2.1 Den vanlige versjonen

Dette er det som da ble den ferdige GUIen til den delen av applikasjonen som skal vise modeller som hus og liknende. Denne versjonen har da fått satt på plass knappen for ”snapshot”, og har også knappen for å tilkalle galleri, her visualisert av en filmstripe.

Knappene er også blitt forandret litt i den forstand at de er blitt runde, dette for å gi ett mykere bilde, og bryte litt med den store firkanten som er kontrollpanelet.

Den endelige versjonen vi har sendt til oppdragsgiver ser slik ut:



Figur 7.1 Den originale GUIen vi sendte til oppdragsgiver for implementering i deres applikasjon.

7.2.2.2 Minimalisert versjon for en "superbruker"

Som nevnt over i punkt 6.3.2.2 på side 65 følte brukere til tider at selve GUIen tok opp for stor plass av selve viewporten.

Figuren nedenfor viser hele skjermbildet for hvordan en minimalisert GUI ser ut:



Figur 7.2 Den minimaliserte GUIen som ble sendt til visualisere.no for implementering i tillegg til den normale GUIen.

7.2.3 GUI for produktvisning

Som i kravene i punkt 2.2.3 på side 20, skulle vi også utforme en GUI for produktvisning i 3D. Denne typen visning burde da ha samme generelle utseende som GUIen til den originale applikasjonen, men også være tilpasset produktvisning. Det ble klart at vi ikke kunne ha samme form for minimap her. Dette ble da byttet ut med en navigasjons kube inspirert fra Autodesk sin "Nav-Cube" i deres 3D Studio Max 2009²⁷ Nedenfor kan du se hvordan denne versjonen av en GUI ble seende ut.



Figur 7.3 Viser tiltenkt GUI for bruk i produktvisning. Fremdeles usikkert om oppdragsgiver vil ta i bruk denne.

²⁷<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659302>

7.3 Applikasjonen

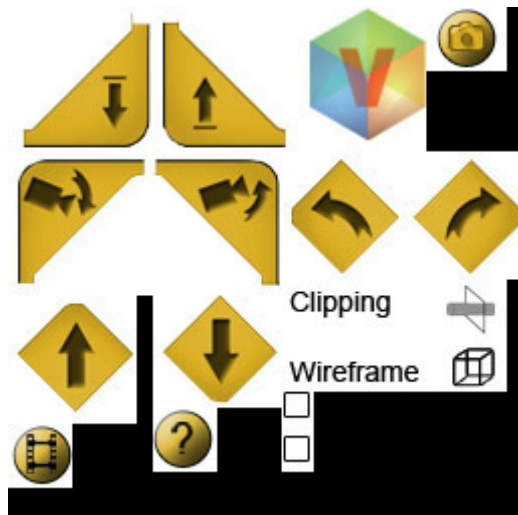
Selve applikasjonen er det ikke vi som står for utviklingen av, den oppgaven er det oppdragsgiver som står for.

Hele applikasjonen er basert på LWJGL²⁸ (LightWeight Java Game Library). Dette er ett alternativ med både profesjonelle og amatører som målgruppe. LWJGL gir brukere tilgang til kraftige verktøy som OpenGL²⁹ (Open Graphics Library), og OpenAL³⁰ (Open Audio Library), noe som gjør LWJGL til ett relativt kraftig verktøy.

LWJGL er ikke laget for å gjøre spill utvikling direkte lett, men det gir tilgang til resurser som ellers enten er dårlig implementert eller ikke tilgjengelig i andre eksisterende alternativer.

7.4 Implementering av GUI hos oppdragsgiver

Dette var da de bildene vi sendte i som .psd (Photoshop dokument) filer til oppdragsgiver. I disse filene var da alle de grafiske elementene av brukergrensesnittet delt opp i forskjellige lag. Dette skulle senere bli brukt av oppdragsgiver i ett såkalt "spritemap", for å implementere dette i Java-koden i selve applikasjonen. Dette er det "spritemapet" vi ble tilsendt av oppdragsgiver:



Figur 7.4 Spritemap oppdragsgiver benytter for implementering av GUI i applikasjonen.

Dette bilde brukes så sammen med Java-koden, der man henter ut elementer i dette spritemapet ved å angi koordinatene til øverste venstre hjørne, og deretter nedre høyre hjørne. Dette er grunnen til at alle elementene er satt sammen i ett rutenett uten overlappende elementer i deres ytre kanter.

Som du ser mangler den store blå linjen på bunnen av Figur 7.1. Den ble utelatt her på grunn av at dette elementet ville ta for stor plass på spritemapet, og kan legges til uten større problem i OpenGL som applikasjonen er bygd på.

Etter at elementene er blitt markert i spritemapet vil en annen klasse tegne selve GUIen i applikasjonen. Dette gjøres da vel å kalle på de forskjellige elementene definert tidligere, ved å så gi de

koordinater om hvor de skal tegnes opp i applikasjonen.

²⁸ <http://www.lwjgl.org/>

²⁹ <http://www.opengl.org/>

³⁰ <http://connect.creativelabs.com/openal/default.aspx>

8 Drøftinger

I dette kapittelet ytrer vi våre subjektive meninger om og drøfter hvordan vi synes prosjektet og det endelige resultatet har blitt.

8.1 Resultater

Vi mener vi har innfridd målene beskrevet i underkapittel 0 på side 10. Prosjektet har resultert i et ferdig produkt, og de fleste kravene i spesifikasjonen er ivaretatt. Merkingen av modeller med tags, har vi dessverre ikke kommet i mål med. Dermed er heller ikke visningen av tag cloud implementert. Vi vurderte i samarbeid med oppdragsgiver hva som var prioritert i forhold til kravspesifikasjonen, og da kom tags ganske langt ned. I tillegg til dette føler vi at vi har tilfredstilt det vi satte av krav til en GUI. Vi har laget en enkel, men samtidig utfyllende og selvforklarende GUI til visning av ett 3D miljø. Vi har også, på kort varsel, laget en ekstra GUI for produktvisning, da dette ble aktuelt halvveis i prosjekt perioden.

Opplastingen av 3D-modell kunne vært bedre gjennomført, da spesielt opplastingen av teksturene. Når den innebygde filvelgeren til nettleseren for 3D-modellene, mens en blanding av Flash og JavaScript brukes for teksturene. Grunnen til dette er at vi fikk beskjed ganske sent av oppdragsgiver at teksturer måtte lastes opp separat. Den midlertidige lagringen av modellen og teksturene før de publiseres er ganske klønete, da de lagres i en midlertidig mappe med filnavnet til modellen. Hvis noen forsøker å laste opp en modell med samme navn til samme tid, vi det sannsynligvis skje en feil. Vi har heller ikke (i skrivende stund) fått på plass en løsning for automatisk fjerning av disse midlertidige filene hvis en bruker ikke publiserer modellen sin.

I forhold til videre utvikling forsøkte vi å implementere websystemet i henhold til en objektorientert- og MVC-tankegang. Vi synes det har vært lærerikt og gitt systemet en god oversiktighet og senket terskelen for andres innsyn og forståelse for systemet. Vi er ikke helt fornøyde med MVC-implementeringen, da spesielt kontrolløren, som kan oppfattes som ustrukturert og uorganisert. Noen av mal-filene inneholder også for mye logikk. I tillegg har vi lært mye om MVC-tankegangen underveis, men kunne på grunn av tidsbegrensinger ikke skrive om systemet hver gang vi lærte noe nytt.

I forhold til videre utvikling av GUIen er det satt av plass til å utvide denne med flere funksjoner dersom dette skulle være nødvendig (lys, teksturer). I tillegg har vi også sendt en .psd fil til oppdragsgiver, slik at denne lett kan tas i bruk for videre design og utvikling av GUIen. Dette gjør at oppdragsgiver, eller en fremtidig utvikler kan se tilbake på hvilke farger og løsninger vi har valgt å bruke.

Systemet, så vel som GUIen, skulle være *enkel* og *brukervennlig*. Dette er noe luftige begreper som er vanskelig å måle, men vi valgte å gjennomføre en brukertest for å se om dette var oppnådd. Dette var en prosess vi lærte mye av, og er fornøyd med gjennomføringen. Vi ser at vi helst skulle gjennomført flere slike tester for å se om våre forbedringer hjalp, men tiden strakk dessverre ikke til.

8.2 Erfaringer

Vi har gjennom prosjektet jobbet med en oppdragsgiver og kontaktperson som befinner seg i Bergen, og som vi har aldri har møtt. Dette har vært en interessant opplevelse, hvor all korrespondanse har foregått over e-post, lynmeldingsstjenester og Skype. Dette er en form for kommunikasjon som vi tror ofte skjer i arbeidslivet, og vi har fått en nyttig erfaring.

Da prosjektet har vært et samarbeid med oppdragsgiver, var det viktig at alt ble utviklet i henhold til krav og status hos oppdragsgiver. Dette er også en erfaring som kan dras inn i en arbeidssituasjon, hvor man ofte jobber tett med en ekstern oppdragsgiver.

Vi har også fått erfaring med å bruke ferdigutviklede funksjoner og skript, og implementering av disse i vår eksisterende kode. Som et resultat av dette har vi også fått mye erfaring med lisenser og rettigheter, og åpen kildekode.

Ved å gjennomføre et prosjekt av denne størrelsen har vi lært veldig mye om utvikling av et større system. Viktige momenter har vært oversiktlig kode- og filstruktur, objektorientering og adskillelse av presentasjon og logikk. Vi har også høstet erfaringer med sikring av et nettbasert system, samtidig som ytelse og brukervennlighet er ivaretatt.

Brukertesting er vi jevnt over fornøyd med, og lærte mye om selve prosessen og fikk konstruktive tilbakemeldinger på våre prototyper.

En annen side av prosjektet har vært utviklingen av det grafiske brukergrensesnittet til 3D applikasjonen. Prosjektet har lært oss mye om hvordan man må tenke om man skal kunne utforme og utvikle et grafisk brukergrensesnitt. Vi har lært at det finnes flere viktige faktorer som spiller inn i utviklingen av et grafisk brukergrensesnitt enn det vi var klar over i begynnelsen. Den menneskelige faktoren har vist seg å være vanskelig å definere under planleggingen, men har også vist seg å være uvurderlig i evaluering av produktet.

8.3 Videre arbeid

8.3.1 Websiden

”Siste siden sist”

”Siste modeller siden sist”-funksjonen fungerer ikke helt optimalt hvis brukeren logger inn ved bruk av ”Husk meg”-funksjonen. ”Siste modeller” henter alle modeller fra databasen som er nyere enn sist brukeren var logget inn, og denne datoen oppdateres kun hvis brukeren logger inn via ”Logg inn”-skjemaet. Dermed får brukere som lagret innloggingsinfoen sin ikke oppdatert denne datoen, og får alle modeller siden de logget inn manuelt. For å få denne funksjonen til å fungere må vi oppdatere datoen når brukere logger inn med informasjonskapsel. Problemet er at dette skjer hver gang brukeren oppdaterer websiden, eller går til en underside, og de eventuelle modellene som er lastet opp i mellomtiden vil ikke vises. En løsning kan være og kun oppdatere datoen hvis en session ikke er satt, da dette må bety at brukeren ikke har vært på websiden på minst 24 minutter.

Søk

Søkefunksjonen er ikke særlig optimalisert, og *bør* byttes ut før eller siden. Den sorterer ikke etter hvor relevant hvert resultat er, og kan bli en flaskehals i forhold til ytelse hvis antallet modeller blir stort i fremtiden, da kombinasjonen LIKE og WILDCARD i MySQL kan være treg. En løsning er å ta i bruk et tillegg som benytter seg av fulltekst-søk, og er optimalisert for ytelse og sortering av resultater. Et eksempel er Sphinx³¹, som er open-source.

Ytelse

Vi har gjort mye for å forbedre ytelsen (gzippe alle CSS-, JS- og PHP-filer, cache CSS- og JS-filer, bruke et minimum av databasetilkoblinger), men allikevel er det mye mer som kan gjøres. For

³¹ <http://sphinxsearch.com/index.html>

eksempel kan *memcache*³²-systemet implementeres for og mellomlagre alle data fra databasespørringene på serveren, og dermed slipper man å spørre databasen hver gang. I tillegg kan alle CSS- og JS-filer slås sammen, og komprimeres/minimeres. Da sparer vi HTTP-forespørsler og minsker filstørrelsen ytterligere.

RSS

En RSS-funksjon som oppdateres hver gang en ny modell legges ut, burde absolutt implementeres.

Generelt

I begynnelsen sto vi mellom valget mellom å benytte et PHP-rammeverk, eller utvikle alt selv. Vi valgte som kjent det siste valget, og er fornøyd med det. Allikevel ser vi nå, for systemets og fremtidig utvikling, at det kunne vært fordelaktig å benyttet et ferdigutviklet rammeverk. Vi vurderer derfor å omstrukturere koden (*refactor*) og lage hele websiden på nytt i et rammeverk, men dette er ikke fastslått.

En 404-side, altså en ”Siden finnes ikke”-mal som vises hver gang en bruker går til en side som ikke finnes, bør implementeres. Her kan det komme nyttige tips til hva brukeren kan gjøre for å gå til korrekt side.

8.3.2 GUI

For å kunne gjøre arbeidet lettere for en videre utvikling av GUIen er det blitt sendt, og vedlagt, en .psd fil som inneholder selve GUIen. Dette vil gjøre en eventuell redesign av GUIen lettere ettersom en senere utvikler kan ta for seg denne, og forandre på det han/hun vil.

Dette vil gjøre det lettere å eventuelt tilpasse 3D applikasjonen til visning av andre elementer, eller å forandre farger skulle den grafiske profilen endre seg. Her vil også en fremtidig utvikler kunne se hvordan de forskjellige elementene er satt sammen i en helhet.

8.3.3 Tjenesten

Per i dag er ikke utvikling av tjenesten kjent, bortsett fra hos oppdragsgiveren og oss. For at den skal bli tatt i bruk må den dermed markedsføres. Oppdragsgiver har ikke tenkt til å legge penger i markedsføringen, men benytte seg av diverse fora på internett og sitt nettverk innenfor 3D-miljøet i Norge.

Konverteringen av 3D-modellene tar forholdsvis mye prosessorkraft, og hvis tjenesten blir populær må sannsynligvis tjenesten flyttes til en dedikert serverløsning. Det vil sannsynligvis også bli problemer med diskplass og båndbredde, da modellene hver bruker får 50 mb til rådighet.

8.4 Evaluering av gruppa

Vi føler at samarbeidet i gruppa har fungert relativt greit. I og med at vi delte oppgaven i to deler (webside og GUI), har vi ikke jobbet så tett i gruppe som en slik oppgave tilsier. Dermed har vi ikke en fullstendig og dyptgående innsikt i hva den andre har jobbet med, men føler vi har en god oversikt. Kommunikasjonen mellom oss har fungert godt, men i og med at prosjektet ble delt, har vi ikke kunne diskutert faglige og tekniske problemer som har dukket opp underveis med hverandre på et nivå som var ønskelig. Oppdragsgiveren, da ved Botnen, har her vært en god ressurs som begge

³² <http://pureform.wordpress.com/2008/05/21/using-memcache-with-mysql-and-php/>

gruppemedlemmene kunne diskutere sine utfordringer med og fått en annen synsvinkel. Veilederen har også vært til god hjelp på dette punktet.

Vi er delvis fornøyde med innsatsen som er lagt ned i prosjektet. Vi kom forholdsvis sent i gang med prosjektet, og i løpet av januar og februar ble det ikke arbeidet like mye som arbeidsmengden (1.5.1 side 12) tilsa. Etter hvert som vi kom i gang ble det lagt ned vesentlig mer arbeid, og mye ble hentet inn. Allikevel ble det travelt mot slutten av perioden, og da vi ville levere et fullstendig produkt til oppdragsgiver ble nok dette prioritert over rapportskriving. Det har vært enklere å finne motivasjon til å arbeide med selve utviklingen av systemet og GUIen enn til rapportskriving, og vi ser nå at mer tid til rapportskrivingen burde vært satt av.

9 Konklusjon

Hovedprosjektet har for oss vært en ny måte å jobbe på, og vi har ikke gjennomført noe slik tidligere. Vi har jobbet tett med en ekstern oppdragsgiver, og samtidig har vi hatt en forholdsvis fri måte å jobbe på, hvor vi selv bestemte mye.

I løpet av prosjektet har brukt mye av kunnskapene opparbeidet her på HiG, og fått en mulighet til å jobbe med det vi har stor interesse for, noe vi er fornøyde med. Selve grunnideen bak prosjektet er noe vi har stor tro på, og ser for oss at dette kan bli populært i fremtiden. Disse faktorene har vært en ekstra motivasjon for å utvikle et godt produkt. Vi har også fått muligheten av oppdragsgiver til å arbeide videre på prosjektet, og ser frem til lansering av et produkt vi er stolte av.

Prosjektet har vært en fin avslutning på vårt bachelorstudie ved HiG, og vi føler vi kan dra nytte av de kunnskaper og erfaringer oppnådd i arbeidslivet.

10 Bibliografi

- Autodesk. (2009). *Autodesk 3ds Max*. Hentet 2009 fra <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659302>
- Benyon, D., Turner, P., & Turner, S. (2005). *Designing Interactive Systems*. Harlow, England: Addison-Wesley.
- BetterExplained. (2007). *How To Optimize Your Site With GZIP Compression*. Hentet 2009 fra BetterExplained: <http://betterexplained.com/articles/how-to-optimize-your-site-with-gzip-compression/>
- Chapman Nigel, C. J. (2004). *Digital Multimedia*. West Sussex: John Wiley & Sons Ltd.
- Glarner, H. (2007, Mars 19). *Isometric Projection*. Hentet Mars 2009 fra <http://herbert.gandraxa.com/herbert/imp.asp>
- Johansen, C. (2007, September). *Ikke-påtrengende Javascript*. Hentet 2009 fra [cjohansen.no](http://www.cjohansen.no): http://www.cjohansen.no/javascript/ikke_paatrengende_javascript
- Johansen, C. (2009, Oktober). *Semantisk HTML*. Hentet Mai 2009 fra [cjohansen.no](http://www.cjohansen.no): http://www.cjohansen.no/html/semantisk_html
- lwjgl.org. (2009). *Lightweight Java Game Library*. Hentet fra <http://lwjgl.org/about.php>
- Moggridge, B. (2007). *Designing Interactions*. London: The MIT Press.
- Rannem, Ø. (2005). *Typografi og skrift*. Oslo: Abstrakt.
- Starr, J. (2006, Januar). *Stupid htaccess Tricks*. Hentet 2009 fra Perishable Press: <http://perishablepress.com/press/2006/01/10/stupid-htaccess-tricks/>
- The PHP Group. (2009, Mai). *PHP Manual*. Hentet 2009 fra <http://www.php.net/manual/en/>
- UI-Patterns.com. (2009). *UI-patterns.com*. Hentet 2009 fra <http://ui-patterns.com/>
- Wikimedia. (2009, Mai). *Model-view-controller*. Hentet 2009 fra Wikipedia: <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

11 Vedlegg

| | |
|-------------------------------|-------|
| A. Definisjoner | s 83 |
| B. Forprosjektrapport | s 87 |
| C. Use cases | s 95 |
| D. Rapport, brukertesting | s 103 |
| E. Gantt-diagram | s 115 |
| F. Statusrapporter | s 119 |
| G. Møtereferater | s 127 |
| H. Logg | s 135 |
| I. Eksempel på bruk av 960.gs | s 139 |
| J. Statistikk, webside | s 143 |
| K. Kodekonvensjon | s 147 |
| L. Mappediagram | s 151 |
| M. Gruppeavtale | s 155 |

Vedlegg A:

Definisjoner

| | |
|----------------------|---|
| (X)HTML | (Exensible) Hypertext markup language |
| Ajax | Asynkron JavaScript og XML. En webutviklingsteknikk for å utvikle interaktive nettsider. Data utveksler i bakgrunnen, istedenfor at siden lastes på nytt. |
| Array | Brukes for å lage lister med nøkler og tilhørende verdier. Et assosiativt array er da et array med egendefinerte nøkler. |
| CAPTCHA | ” Completely Automated Public Turing test to tell Computers and Humans Apart”. Brukes for å finne ut om en bruker er menneske eller maskin, ved å vise frem noe som ikke en masin kan tolke, for eksempel forvrent tekst. |
| Creative Commons 2.5 | Gir rettigheter til å bruke produktet i til akkurat hva man vil, og endre det så mye man vil. Det eneste opphavspersonen krever er at han/hun blir navngitt på den måte de angir, og verkene skal deles videre med samme lisens. Les mer hos Creative Commons: http://creativecommons.org/licenses/by-sa/2.5/deed.no . |
| CSS | Cascading style sheets |
| First-Person-View | Kan enklest beskrives som at bildet på skjermen etterligner det øynene dine ville sett i de samme omgivelsene. Du ser ut gjennom øynene på skjermen. |
| Flash | Adobe Flash, verktøy for å publisere animasjoner og multimedia på web. |
| GNU GPL | Lisens for publisering og distribuering av fri programvare. |
| MD5 | Message-Digest algorithm 5, en sjekksumalgoritme som lager en 128-bits sjekksum, som oppgives i et 32-tegns heksadesimalt tall. |
| MIT | Lisens for publisering og distribuering av fri programvare. |
| MySQL | En SQL-basert databasetjener lisensiert under GPL. |
| PDO | PHP5 Data Object, et databaseabstraksjonslag innebygget i PHP5, som kan gjøre det lettere å implementere støtte for flere databasesystemer samt å bytte fra et databasesystem til et annet. Øker også sikkerheten ved hjelp av forberedte spørringer. |
| PHP | PHP: Hypertext Preprocessor, et dynamisk og tolket programmeringsspråk brukt for å utvikle dynamiske websider. |
| Session | PHPs måte for å overføre informasjon mellom sider. |
| Superbruker | En bruker av en applikasjon som er godt kjent i applikasjonen og tar i bruk de mer avanserte funksjonene som finnes tilgjengelig. |
| Tag cloud | For å visualisere data på en webside vises alle tags, hvor de mest brukte er mer fremtredende (for eksempel har større tekststørrelse) |
| Tags | Brukes for å organisere og sortere data, da data kan inneholde en eller flere tags. |
| Use cases | Verktøy for å spesifisere krav og deler av et system |
| Vertex | Punktet der linjene møtes i en 3D modell. |
| Whitespace | Et viktig grafisk element i design. Kan brukes til å samle et design, men også til å dele opp tekst. Et generelt viktig estetisk element i design. |
| Wireframe | En visuell visning av en 3D modell, der kun linjene fra vertex til vertex vises. |

Vedlegg B:

Forprosjektrapport

1 Bakgrunn

Vi skal i forbindelse med avslutningen på vår bachelorgrad i medieteknologi utføre et prosjekt i faget IMT3912 – Bacheloroppgave, som består av 20 studiepoeng. Etter litt frem og tilbake angående valg av prosjekt endte vi opp med et prosjekt fra Visualisere.no.

Målet til Visualisere.no er å utvikle et rammeverk på tvers av flere disipliner, hvorav de er valgt visualisering av hus, bygninger og arkitektur er valgt som test-case. Videre ser de for seg å flytte 3D-golf-prosjektet over på dette rammeverket, samt utvide det til andre områder. Her mulighetene mange, for eksempel en maritim utgave, hvor visualisering av båter er produktet.

3D-applikasjonen skal utvikles i enten Director Shockwave 3D (kjører i html som plugin ala flash) eller Java (JOGL, LWJG eller jMonkey Edition) webstart (ekstern applikasjon som kjører på topp av nettleser, men startes fra nettleser).

1.1 Visualisere.no

Visualisere har tidligere utviklet en 3D-visning for to golfbaner³³ og en for et hus³⁴, med henholdsvis Adobe Shockwave 3D³⁵ og jMonkey Edition³⁶. 3D-motoren skal bygges videre på erfaringene med disse prosjektene.

<http://www.visualisere.no>

2 Rammer

Vi er gitt visse rammer av skolen i form av tidsfrister. De viktigste er:

- Oppgaven skal leveres til studenttorget den 25.05.09 innen kl. 12.00
- En A3-plakat som presenterer vårt prosjekt skal leveres til laminering 28.05.09 innen kl. 12.00, og skal leveres til studenttorget 02.06.09 innen kl. 12.00.
- Muntlig presentasjon av prosjektet skjer den 03., 04. eller 05.06.09.

3 Oppgavebeskrivelse

Vi har som mål å utvikle en dynamisk webside og GUI for Visualisere.no, som skal utvikle en 3D-applikasjon for web. Vi har ansvaret for å utvikle websiden for denne applikasjonen, med de funksjoner som applikasjonen og prosjektet krever. Dokumentasjon for vår del skal også utarbeides.

Navigering i 3 dimensjoner er en utfordring da man har seks grader av frihet. Hvordan løse dette på en bra måte? Skal man bruke en HUD, menyer eller noe annet? Viewcube, remote-controller og lignende er aktuelle stikkord. Det er også viktig å fokusere på brukervennlighet, da sluttbruker av produktet ikke nødvendigvis har kunnskaper om hvordan man beveger seg og bruker en 3D-

³³ http://www.wsgolfer.com/dswmedia/volda08_public_beta_a.htm

http://www.wsgolfer.com/dswmedia/solnoer07_public_beta_a.htm

³⁴ <http://www.botnen.org/webstart> visualisere:prosjekt2009

³⁵ <http://www.adobe.com/products/director/>

³⁶ <http://www.jmonkeyengine.com/>

applikasjon. Se på brukerscenarioet lengre ned. Brukerinteraksjon er da et viktig tema. For å sikre god brukerinteraksjon skal vi utføre brukertesting på reelle brukere, og vi skal utføre dette på en gruppe mennesker innenfor målgruppen for produktet. Hvor mange vi skal utføre brukertesten på blir bestemt senere, men vi planlegger å teste ved hjelp av et eller flere use cases, som testpersonen går igjennom mens et av gruppemedlemmene observerer og noterer brukerens hendelsesforløp.

Websiden skal være som sagt være dynamisk, dette vil vi løse ved hjelp av programmeringsspråket PHP og det SQL-baserte databaseadministrasjonssystemet MySQL. Funksjoner oppdragsgiver blant annet trenger er

- Brukerhåndtering
- Menysystem
- Gallerisystem
- Opplastingsfunksjon
- Forskjellige views, basert på blant annet tid og popularitet
- Sitemapfunksjon.

Dette må selvfølgelig integreres og samkjøres med 3D-applikasjonen, med tanke på både det grafiske og backend. Brukergrensesnittet på applikasjonen ser man for seg har gjenkjennelige elementer fra hjemmesiden, for eksempel farger og former som går igjen på begge steder.

Vi skal underveis vurdere bruken av ferdiglagede løsninger kontra egenutvikling, men hvis vi tar i bruk ferdige løsninger er det viktig at vi undersøker nøyte lisenser og rettigheter i forhold til kommersiell bruk.

Vi skal også utarbeide en enkel dokumentasjon for prosjektet, "Brukerveiledning" til websiden og applikasjonen, det vil si en forklaring for brukeren, da sluttbruker ikke nødvendigvis innehar stor IT-kompetanse.

4 Problemstilling

Måten arkitekter i dag viser frem sine konstruksjoner online består som regel av bilder av plantegninger, samt et eller to statiske 3D-bilder av konstruksjonen (ref. Norgeshus³⁷ og BoligPartner³⁸). Dette gir klienten en viss innsikt i konstruksjonen, men en komplett 3D-visning i en 3D-motor gir et mer realistisk innblikk.

Vi skal i samarbeid med Visualisere.no utvikle en webside ved hjelp av PHP og MySQL, som har som jobb å fungere som en hovedside og rammeverk for en 3D-applikasjon. Her skal man få informasjon om verktøyet, og hvordan man bruker det.

- Hvordan kan 3D-materiell innsendt av brukere kvalitetssikres, sjekkes for korrekt format og lagres for bruk av 3D-applikasjonen?
- Hvordan skape en API (Application Programming Interface)/rammeverk slik at 3D-motor kan tilkobles?

³⁷ <http://www.norgeshus.no/article.php?id=1429>

³⁸ <http://www.boligpartner.no/ferdighus/Nostalgi/Alsgaard/>

Vi skal også utvikle en GUI for 3D-motoren, som skal gi brukeren en brukervennlig og enkel måte å navigere i ett 3D-miljø. Hovedutfordringen blir å skape en enkel og logisk arbeidsflyt mellom webside og 3D-applikasjon.

- Hvordan kan en god arbeidsflyt mellom webside og 3D-applikasjon sikres?
- Hvilke grep trengs for å skape en god brukerinteraksjon i et 3D-miljø, selv for brukere uten kunnskaper på dette feltet?
- Hvordan skal GUI samarbeide med 3D-motoren til Visualisere.no?

5 Avgrensning

Vi skal ikke utvikle selve 3D-applikasjonen, da Visualisere.no selv skal utvikle denne.

Siden skal ikke nødvendigvis integreres med den allerede eksisterende visualisere.no siden så her er vi helt fri til å designe fra grunn og opp.

Vårt prosjekt er avhengig av 3D-applikasjonen Visualisere.no utvikler.

6 Prosjektets mål

6.1 Hovedmål

Vi har som mål å utvikle en dynamisk webside basert på PHP og MySQL for Visualisere.no med et rammeverk og API.

Vi skal også utvikle en GUI for 3D-applikasjonen laget av Visualisere.no, samt utvikle dokumentasjon.

6.2 Effektmål

- Visualisere.no vil spare kostnader for sine kunder ved å gjøre modellene lettere tilgjengelig for sine klienter, da over internett.
- De vil også spare tid, ettersom modellene ligger ute på nett klare til gjennomsyn av kunder og klienter.
- Arbeidsgiver har ingen definerte effektmål per dags dato, men kun ideer og visjoner om hvor prosjektet skal bære hen når det er ferdig. Siden dette prosjektet er et test-case for videre utvikling av 3D-rammeverket, vil dette prosjektet gi nyttig informasjon og være en ressurs i den videre utviklingen.

6.3 Læringsmål

- Vi skal i løpet av prosjektperioden oppnå bedre kunnskaper om hvordan man setter opp en webside for en kunde, og hvordan en GUI utvikles og overleveres i riktig format.
- Vi skal også få bedre innsikt i hvordan et prosjekt av denne typen utføres for en ekstern arbeidsgiver. Tidsfrister må også overholdes, arbeid skal bli likt fordelt og dokumentasjon og rapporter skal utarbeides.

- Vi skal lære å jobbe mot et konkret mål i et prosjekt. Vi skal også lære å dele prosjektet inn i prosesser, for å kunne styre prosjektet bedre.

6.4 Brukerscenario

Under følger et veldig enkelt brukerscenario, som viser hvordan oppdragsgiver ser for seg generelt bruk av 3D-applikasjonen. Dette er altså en overordnet forklaring av systemet.

”Arkitektkontoret bezy i Bodø ønsker å vise en modell under bygning og laster den derfor opp på systemet så klienten i Kristiansand kan logge seg på og se den”.

Dette skal altså gi brukeren, her arkitektene ved bezy, en mulighet til å vise sine kunder sine arkitektmodeller, visualisert i en 3D-motor. Et eksempel på hvordan dette gjøres i dag, er hos Norgeshus³⁹. Her vises i 3D fra en vinkel, mens man kan se bilder i 2D av huset forfra, bakfra og fra sidene. Med vårt prosjekt skal dette gjøres sømløst i et 3D-miljø.

Fordelene for arkitektene gir dette produktet en merverdi, da det tar seg bedre ut for kunden. For kunden gir det en utvidet mulighet til å se det fremtidige huset i alle vinkler, og satt i et naturlig miljø.

Ulempene ved denne fremstillingen er at man må lage 3D-modeller av objektene som skal fremstilles, mens det kan være vanskelig og unaturlig for en kunde uten særlig datakunnskaper og benytte seg av en slik tjeneste.

6.5 Målgruppe

Målgruppen for dette prosjektet er hovedsakelig arkitekter og andre som arbeider med arkitektur, som vil vise frem sine verk til sine kunder. Målgruppen er dermed relativt bred, med brukere med forskjellig teknisk innsikt innenfor IT.

6.6 Teknologiske krav

3D-applikasjonen skal fungere på de mest brukte operativsystemene (Windows Vista, Windows XP og Mac OS X 10.5).

3D-applikasjonen kommer til å kreve at bruker installerer tredje-parts programvare, mest sannsynligvis enten Java eller Shockwave, men skal ikke kreve noe spesiell nettleser. Selve websiden skal heller ikke kreve noen spesiell nettleser, men vi skal ikke spesielt utvikle for nettlesere som ikke er på Yahoo!s A-Grade liste⁴⁰.

7 Deltakere

7.1 Gruppemedlemmer

7.1.1 Aleksander Stalsberg

Østre Totenveg 137
2816 Gjøvik
408 68 004
aleks@stalsberg.net

³⁹ <http://www.norgeshus.no/article.php?id=1179>

⁴⁰ <http://developer.yahoo.com/yui/articles/gbs/>

7.1.2 Jon Espen Kvisler

Østre Totenveg 116

2816 Gjøvik

480 47 590

jonespen@gmail.com

Kontaktperson, oppdragsgiver

Kristian Botnen

952 18 147

kristian@visualisere.no

8 Ressursbehov

Websiden og tekniske løsninger: Øyvind Kolloen ved HiG.

GUI (brukerinteraksjon og brukervennlighet): Frode Volden ved HiG.

GUI (grafisk): Ole Lund ved HiG.

GUI: 3D UI Best Practice⁴¹

Prosjektveileder: Anne Kvitle ved HiG.

9 Fremdriftsplan

9.1 Milepæler

1. 11.02.09 – Forprosjekt.
2. 20.02.09 – Analyse og undersøkelser av aktuelle teknologier.
3. 27.03.09 – Første utkast, webside og GUI. Grafisk.
4. 07.05.09 – Ferdigutvikling av GUI og webløsning, med rammeverk.
5. 25.05.09 – Ferdigstilling av rapport og dokumentasjon.
6. 03., 04. eller 05.06.09 – Fremføring av prosjekt.

9.2 Faser

9.2.1 Forprosjekt

- Fastsette mål og problemstilling, samt gi en generell beskrivelse av prosjektet og organiseringen av dette.
- Dette er starten på prosjektet hvor vi beskriver prosjektet og hva vi skal gjøre fremover.

9.2.2 Research

- Se på aktuelle teknologier for webside, og snakke med ressurspersoner angående GUI.
- Søke etter lignende nettsider og programvare, og se på hvordan ting er lagt opp for å se om vi kan lære noe.
- Vi må også avklare med oppdragsgiver hvordan et rammeverk skal settes opp.
- En målgruppe må også defineres.

⁴¹ <http://3dui.manchesterdda.com/>

9.2.3 Utforming av utkast og prototype

- Her skal vi først utvikle en generell grafisk profil som blir en slags mal for det grafiske utseende både for web og GUI.
- Deretter skal vi utvikle en prototype både for web (ferdige HTML og CSS filer) og GUI.
- Brukertestning skal så utføres på personer innenfor den definerte målgruppen. Dette for å få et inntrykk av hvordan den jevne bruker oppfatter våre prototyper. Planen for brukertestning vil bli satt opp på et senere tidspunkt.

9.2.4 Utvikling

- Koding av webside og utvikling av GUI. I denne perioden skal webside, GUI og dokumentasjon for bruker ferdigstilles.

9.2.5 Rapport

- Ferdigstille rapport og ombreкке denne, slik at den er klar for korrektur.

Vedlegg C:

Use cases

1 Overordnede use cases

Det forutsettes at aktørene ”Registrert bruker” og ”Administrator” er logget inn.

1.1.1 Registrere seg

Mål: Lage en konto hos Visualisere.org.

Aktør: Bruker.

Beskrivelse: En bruker oppretter en ny konto på websiden. Alle data lagres i databasen, sammen med når brukeren ble opprettet.

Pre-betingelser

1. Verken brukernavnet eller e-postadressen er registrert fra før.
2. Brukeren kan lese CAPTCHA-koden.
3. Brukeren godkjenner retningslinjene for Visualisere.org.

Post-betingelser

1. En ny bruker opprettet i databasen.
2. Bruker har mulighet til å logge seg inn.

1.1.2 Logge inn

Mål: Logge seg inn.

Aktør: Bruker.

Beskrivelse: Aktør logger seg inn med sitt brukernavn og passord. Passordet skal kamufleres ved inntasting slik at ingen andre ser det.

Pre-betingelser

1. Aktør har en brukerkonto.
2. Aktør er ikke logget inn.

Post-betingelser

1. Aktør har økt tilgangsnivå.
2. Databasen oppdateres med tidspunkt for når brukeren logget seg inn.

1.1.3 Se 3D-modell

Mål: Vise 3D-modell.

Aktør: Bruker.

Beskrivelse: Aktøren navigerer seg til aktuell 3D-modell på websiden, og starter 3D-applikasjonen. Informasjon om plasseringen til 3D-modellen er lagret i databasen, og må dermed hentes ut og sendes med til 3D-applikasjonen.

Pre-betingelser

1. Aktøren har en nettleser som er kompatibel med nettstedet.
2. Aktøren har en nettleser kompatibel med Java versjon 6.
3. Korrekte parametere blir sendt med til 3D-applikasjonen for å fortelle hvilken modell som skal lastes.

Post-betingelser

Antall visninger av modell blir oppdatert i databasen.

1.1.4 Laste opp 3D-modell

Mål: Bruker laster opp en 3D-modell fra sin klient og inn i systemet på tjenersiden.

Aktør: Registrert bruker.

Beskrivelse: På nettstedets opplastingsside, velger aktøren en 3D-modell fra sin klient, og velger "Last opp". Filen blir så sendt til en funksjon som sjekker filstørrelse, format og ser etter om filen kan inneholde usikker kode. Deretter skal aktøren beskrive sin modell med en tittel, beskrivelse og kategori. I tillegg kan aktøren velge hvem som skal ha tilgang til modellen, og beskrive den ytterligere med tags. Hvis aktøren ikke har laget modellen selv, eller ikke har opphavsretten til den, kan han angi navnet og e-postadressen til opphavspersonen.

Pre-betingelser

1. Aktøren har en 3D-modell på sin klient i et kompatibelt format.
2. Aktøren må ha en brukerkonto hos Visualisere.org, og være logget inn.

Post-betingelser

1. Modell er tilgjengelig for allmennheten.
2. Informasjon om modellen er lagret i databasen.
3. Modellen er lagret på serveren.

1.1.5 Endre modell

Mål: Bruker kan endre informasjon om sin egen modell.

Aktør: Registrert bruker.

Beskrivelse: Aktøren får listet opp alle modeller som han selv har lastet opp, og velger ønsket modell herfra. Her kan man endre tittel, beskrivelse, kategori, tilgang til modellen, tags, og endre opphavsmann. Det skal også være mulig å laste opp et nytt miniatyrbilde til modellen.

Pre-betingelser

Aktøren har lastet opp en 3D-modell.

Post-betingelser

1. Informasjon som er endret blir oppdatert i databasen.
2. Eventuelt bilde blir lastet opp, og konvertert til riktig størrelsesformat for miniatyrbilder.

1.1.6 Endre brukerdata

Mål: Bruker skal kunne endre data om seg selv.

Aktør: Registrert bruker.

Beskrivelse: Aktøren skal kunne endre e-post, passord, fullt navn, bosted, hjemmeside og beskrivelse. Det skal ikke være mulig å endre passord uten å oppgi det gamle passordet først.

Post-betingelser

De data som aktøren endret skal oppdateres i databasen, og dermed for alle som ser profilen til aktøren.

1.1.7 Lage nyhetsartikkel

Mål: Administrator oppretter en nyhetsartikkel.

Aktør: Administrator.

Beskrivelse: På websiden skal det vises nyhetsartikler, hvor det er meningen at oppdateringer om websiden kan postes. Disse artiklene må ha en tittel og dato for publisering i tillegg til selve artikkelen, sorteres slik at den nyeste artikkelen legges først.

Post-betingelser

1. Artikkelen lagres i databasen.
2. En ny artikkel er tilgjengelig for alle brukere.

1.1.8 Slette nyhetsartikkel

Mål: Administrator sletter en nyhetsartikkel

Aktør: Administrator.

Beskrivelse: Aktør finner en nyhetsartikkel i en liste over alle artikler, og sletter den gitte artikkelen herfra.

Pre-betingelser

Det finnes en artikkel.

Post-betingelser

Den valgte artikkelen slettes fra databasen.

1.2 Detaljert use case

1.2.1 Laste opp modell

Mål: Bruker laster opp en 3D-modell fra sin klient og inn i systemet på tjenersiden.

Aktør: Registrert bruker.

Pre-betingelser

1. Aktøren har en 3D-modell på sin klient i et kompatibelt format.
2. Aktøren må ha en brukerkonto hos Visualisere.org, og være logget inn.

Post-betingelser

1. Modell er tilgjengelig for allmennheten.
2. Informasjon om modellen er lagret i databasen.
3. Modellen er lagret på serveren.

Varianter

6a Flere teksturer valgt

1. Gjør punkt 7 og 8 for hver tekstur

Unntakshåndtering

3a modellen er større enn lovlig filstørrelse,

3b modellen er ikke en gyldig 3D-modell,

3c aktøren har ikke mer diskplass,

7a teksturen overgår maksimum filstørrelse

7b teksturen er ikke et gyldig bilde

7c teksturene tar større plass enn bruker har plass til

1. Viser aktuell feilmelding og lar aktør velge en ny fil.

14a modellen finnes ikke

15a teksturene som skal ligge ved modellen finnes ikke

1. Vis feilmelding og be aktøren starte prosessen på nytt.

| | Akør | Klient | Tjener |
|----|--|-------------------------------------|--|
| 1 | | Viser opplastingsfelt for 3D-modell | |
| 2 | Velger modell | | |
| 3 | | | Validerer fil |
| 4 | | | Legger filen i den midlertidige mappen |
| 5 | | Viser opplastingsfelt for teksturer | |
| 6 | Velger tekstur | | |
| 7 | | | Validerer fil |
| 8 | | | Legger filen i den midlertidige mappen |
| 9 | | Viser beskrivelsesfelt for modellen | |
| 10 | Skriver inn tittel, beskrivelse og velger kategori | | |
| 11 | | | Validerer data fra brukeren |
| 12 | | Viser data bruker tastet inn | |
| 13 | Bruker bekrefter at data og publiserer modell | | |
| 14 | | | Finner 3D-modellen i den midlertidige mappen |

| | | | |
|----|--|--|---|
| 15 | | | Finner teksturer i den midlertidige mappen |
| 16 | | | Legger data om modellen i databasen |
| 17 | | | Legger inn data om teksturene i databasen |
| 18 | | | Legger 3D-modellen i den endelige mappen, hvis navn består av brukernavnet til aktøren, og ID-en til modellen fra databasen |
| 19 | | | Sender 3D-modellen og teksturene til 3D-applikasjon for konvertering |

Vedlegg D:

Rapport, brukertesting

1 Rapport, brukertesting visualisere.org

Onsdag 22. april gjennomførte vi en brukertest av prototype for websiden og 3D-applikasjonen, for å avdekke eventuelle brukervennlighetsproblemer og/eller andre problemer med våre prototyper. Prototypen av websiden var utviklet i HTML/CSS, og ble kjørt rett i nettleseren, mens prototypen av 3D-applikasjonen ble utviklet i Microsoft PowerPoint, og krevde dermed PowerPoint for å kjøres.

1.1 Brukertest

Brukertesting besto av fire steg, hvor det første var rekruttering.

1.1.1 Rekruttering

Vi måtte skaffe mennesker som kunne gjennomføre brukertesten, og som ikke hadde (for) stor kjennskap til vårt system. Siden målgruppen vår består av forholdsvis unge personer med gode datakunnskaper, så vi ikke på det som et problem å bruke studenter som testpersoner. Vi valgte å teste seks mer eller mindre tilfeldige personer (også kalt "Hallway testing"⁴²). Grunnen til det lave antallet er at man ifølge brukervennlighetseksperten Jacob Nielsen får så og si all den tilbakemeldingen man trenger fra fem personer⁴³, og at vi rett og slett ikke hadde tid eller resurser til å ha flere testpersoner. Vi burde heller brukerteste flere ganger i løpet av utviklingsprosessen. I løpet av brukertesten var dette noe vi merket oss også, etter 4-5 testpersoner kom det frem lite nytt, og de samme problemene ble rapportert mer og mer.

1.1.2 Forberedelser

Vi bestemte oss ganske tidlig for å gjennomføre en såkalt "Ansikt til ansikt"-test, hvor testpersonene gjennomførte en eller flere forhåndslagede oppgaver, også kalt scenarioer, på vår teststasjon. Vi utarbeidet tre slike scenarioer, som vi printet ut før brukertesten. Disse ble utviklet for å teste mest mulig av systemet, slik at vi fikk tilbakemeldinger på funksjonalitet. Vi satte som mål at alle personene skulle gå igjennom minst to scenarioer, men på grunn av tidsmangel ble det noen som kun kjørte et scenario.

1.1.3 Gjennomføring av test

Alle testpersonene ble før brukertesten fortalt at det er produktet som skal testes, og ikke personen. Dette for å gjøre personen mer sikker, og ikke være redd for å "dumme" seg ut. De ble også fortalt at de skulle tenke høyt under hele prosessen, slik at vi hele tiden forsto hva personen foretok seg. Hvis det ble stille over en lengre periode under testen, ble testpersonen minnet på dette.

Jon Espen var sekretær, og noterte reaksjoner og viktige handlinger fra testpersonen. Testen foregikk på vårt kontor ved HiG, A036, og laptopen til Jon Espen ble benyttet som testmaskin. Før hver test ble PC-en "nullstilt", det vil si at alle vinduer på PC-en ble lukket, bortsett fra Microsoft PowerPoint, nettleseren Google Chrome ble åpnet uten noen faner, og alle data i nettleseren ble slettet. Dette ble gjort for å sikre at alle testene ble gjort uten forstyrrelser og at testene ble så like som mulig.

1.1.4 Etterarbeid

Etter gjennomført test diskuterte gruppen resultatene, og alle steg fra hver testperson ble dokumentert. Erfaringene ble benyttet i videre designutvikling.

⁴² http://en.wikipedia.org/wiki/Usability_testing

⁴³ <http://www.useit.com/alertbox/20000319.html>

1.2 Scenarioer

Alle filer blir lagt i mappen "3D-mod" på skrivebordet på teststasjonen.

1.2.1 Scenario #1

Du er arkitekten **Per Olsen**, ansatt hos **BYZ AS**, et arkitektkontor. Du har nettopp ferdigstilt en 3D-modell (*modell-zx.mod*) av et hus for en klient, og nå vil klienten se modellen. Klienten, **Reodor Felgen**, er bosatt på andre siden av landet, derfor vil du vise den via internett. Visualisere.org tilbyr dette, så du bestemmer deg for å prøve dette. Modellen viser (det fremtidige) huset til Reodor, og tanken er at Reodor kan se huset i et interaktiv 3D-miljø.

Registrer deg på Visualisere.org (<http://www.visualisere.org/dev/brukertesting>), og last opp modellen. Du er litt usikker på hva som skjer med din opphavsrett ved opplasting av modell til nettstedet, derfor vil du lese Visualisering.orgs **retningslinjer** for opplasting. Du har laget en beskrivende bilde, *modell-zx-img.jpg*, som ligger i samme mappe som modellfilen. Siden det kun er Reodor som skal se denne modellen, og han ikke er registrert på nettstedet, kan du passordbeskytte filen med passordet "**reodor123**".

1. Naviger deg til <http://www.visualisere.org/dev/brukertesting> i en nettleser
2. Lag en konto (valgfritt brukernavn/passord)
3. Last opp en ny modell med følgende informasjon
 1. Modellen: "modell-zx.mod"
 2. Tittel: Modell ZX
 3. Beskrivelse: Dette er en modell av huset til Reodor Felgen
 4. Kategori: Arkitektur
 5. Tags: "Hus", "BYZ AS"
 6. Velg bildet "modell-zx.jpg" fra skrivebordet
 7. Passordbeskytt filen med passordet "**reodor123**"
 8. Gå videre
4. Hvis alt data stemmer, publiser modellen

1.2.2 Scenario #2

Du er fortsatt arkitekten Per Olsen, men nå skal du laste opp en modell av en golfbanen Gjøvik Golfbane, laget av modellereren **Kurt Kristiansen** (kurt@kristiansen.no). Denne skal kun være tilgjengelig for **Kurt Nilsen** og **Gøran Sørloth**. 3D-modellen heter *gjovik-golf.mod*, mens bildefilen heter *gjovik-golf-img.jpg*.

1.2.3 Scenario #3

Finn modellen *Arkitekt_hus_1 - revidert*, og åpne denne i 3D applikasjonen.

Inne på selve modellen skal du gjøre følgende:

1. Ved hjelp av navigerings verktøyet skal du bevege deg frem til trappa. Først fremover, for så å snu deg 90-grader, slik at du ser trappa
2. Igjen ved hjelp av navigerings verktøyet skal du nå gå helt bort til trappa. Gå ett skritt frem, og deretter snu kamera 45-grader, slik at du ser opp trappa.
3. Gå opp trappen, snu kamera ned igjen, og snu deg 45-grader til høyre for å se på snømannen.
4. Snu deg tilbake til venstre, gå frem til vinduet, og skru på "clipping".
5. Gå deretter fremover, igjennom vinduet til innsiden av rommet. Inne i rommet, snu deg 90-grader til høyre.
6. Skru nå på "minimap"et slik at du kan navigere deg igjennom rommet uten å se på selve

modellen. Gå nå frem og snu til venstre, så går du frem til sofaen.

7. Snu deg så til venstre til du ser på tre-bordet. Skru så på "wireframe" for å se hvordan den er bygd opp.

1.3 "Magnus", 23 år

1.3.1.1 Scenario #1

1. Navigerte seg til nettsiden uten problemer.
2. Fant linken til registreringssiden med en gang.
 1. Ville helst registrert seg kun med epost-adresse, og lurte på hvor man kunne skrive inn fullt navn.
3. Logge seg inn gikk fint.
4. Hadde store problemer med å finne "Last opp modell"-linken, tenkte at linken var en del av headeren og ikke en klikkbar link, dermed leste han aldri innholdet i linken. Letet etter en "last opp modell"-link under "Modeller", og måtte til slutt bli veiledet til linken.
 1. Klarte alt under opplasting av modell uten problemer, men syntes "Gå tilbake"- og "Gå videre"-knappene var for like og tett plassert. Syntes det var fort gjort å trykke feil og dermed gå feil vei.
 2. Klaget på stavingen av "Passordbeskytt".
5. Publisering ok.

1.4 "Anita", 23 år

1.4.1.1 Scenario #1

Gjorde ingen spesielle feil gjennom scenarioet. Eneste tilbakemeldingen var at hun helst ville ha epost-adressen sin som brukernavn, og bruke denne ved innlogging.

1.4.1.2 Scenario #3

1. Fant modellen relativt kjapp, gikk via "Modeller" for å finne den.
2. 3D applikasjonen
 1. Navigerte bra.
 2. Usikker på hva clipping og minimap var/gjorde.
 3. Kommenterte at hun var usikker på hva tilteknappene av kameraet gjorde, og forsto ikke "gå opp"- og "gå ned"-knappene.

1.5 "Tommy", 25 år

1.5.1.1 Scenario #1

1. Navigerte seg til siden uten problemer.
2. Brukte linken i velkomstbeskjeden for komme til registreringssiden. Ellers gikk registreringen av konto fint.
3. Innlogging gikk fint.
4. Fant ikke linken til "Last opp modeller", gikk til modeller og forventet å se den der, og så etter den i hovedmenyen. Måtte til slutt veiledes til linken.
 1. Fant ikke "Passordbeskytt". Syntes den burde gjøres mer tydelig, han leste starten på listen under "Synlighet", men hoppet over det siste punktet (Passordbeskytt).

1.6 "Erling", 27 år

1.6.1.1 Scenario #1

1. Navigerte seg til siden uten problemer.
2. Fant frem til registreringssiden, kommenterte at han like registreringsskjemaet.
 1. Vil helst registrere seg med både brukernavn og e-post.
3. Av en eller annen grunn reagerte ikke enter-tasten når han skrev inn brukernavn og passord for å logge seg inn, men kom seg inn ved hjelp av musen.
4. Slet med å finne siden med retningslinjene, men gikk til slutt til "Hjelp" hvor de var plassert.
 1. Kommenterte at han syntes det burde være en "Hjem"-link på hovedmenyen.
5. Brukte lang tid på å finne "Last opp modell"-linken.
 1. Han synes ikke det så ut som linken/knappen var en del av siden, trodde det kun var en del av designet.
 2. Ville helst hatt linken på menylinjen, eventuelt plassert knappen under headeren, da det så ut som linken var en del av headeren og ikke en del av "siden".
 3. Forsto ikke helt hva man skulle skrive beskrivelse av i steg 2 av modellopplasting.
 4. Mente passordet man skrev inn for å passordbeskytte modellen burde være i formen ***, og ikke synlig.

1.7 "Hilde", 24 år

1.7.1.1 Scenario #1

1. Navigerte seg til siden uten problemer.
2. Ville helst hatt en link til registreringsskjemaet under "Logg inn"-modulen i sidebaren, men fant registreringsskjemaet.
 1. Ellers gikk registreringen ok.
3. Logget seg inn uten problemer.
4. Fant retningslinjene med en gang, under hjelp. Synes allikevel det kunne vært en mer synlig link, og at den kunne bli kalt "Rettigheter" eller lignende.
5. Brukte over 5 sekunder på å finne "Last opp modell"-linken, og synes den var dumt plassert. Lå på det siste stedet hun så på en nettside, og synes den var alt for gjemt bort til å være en så viktig funksjon.
 1. Fikk lastet opp modellen, men synes de rød stjernene som markerte at et felt er nødvendig var alt for små.
 2. Mente teksten i <label>-taggene burde være større, da hun synes det var litt vanskelig å lese så liten fet tekst.
 3. Når hun skulle skrive inn tags, ble hun forvirret av hjelpeteksten (*Skilles med komma: "bus, hytte"*), da hun skrev inn anførselstegnene og ble usikker på om disse var nødvendige.
 4. Synes det var vanskelig å se "Passordbeskytt"-knappen
 1. Generelt synes hun alle punktene under "Synlighet" burde beskrives bedre, og markeres bedre.
 5. Lurte på hva "Credits" var, synes navnet burde endres til "Rettigheter". Synes også at man ikke kunne gå videre før man hadde valgt at man hadde laget modellen, ikke at den burde være automatisk valgt.
 1. For eksempel kunne en liste med radio-knapper vært brukt, med ingen av valgene valgt som standard. Denne listen kunne inneholde to punkter; "Jeg har laget modellen selv", og "Noen andre har laget modellen".

1.7.1.2 Scenario #2

1. Klarte alt frem til punktet hvor man skulle velge flere kontakter. Hun fant ut hvordan man fikk frem sine kontakter, men ble veldig usikker på hvordan man kunne velge mer enn én kontakt.
 1. Til slutt måtte hun veiledes til å bruke CTRL-knappen for å velge flere kontakter. Dette synes hun var veldig tungvindt, og kom med forslag på hvordan hun synes dette burde vært løst: At man hadde en liste med alle sine kontakter (som det er i dag), og en liste til høyre med valgte kontakter. I mellom kan det være en slags funksjon (for eksempel en pil begge veier) hvor man kan føre over kontakter som man vil skal se modellen i den "valgte" listen.
 2. Under "Credits" synes hun det burde vært mer beskrivende tekst. Hun lurte mest på hva som skjedde hvis man valgte en annen opphavsmann.
 - Blir denne personen kontaktet via e-post og gjort oppmerksom på at en annen bruker lastet opp en modell som var hans/hennes? Og blir det opplyst om *hvem* som har lastet opp modellen?
 - Må opphavsmannen godkjenne at modellen skal publiseres?
 - Generelt ville hun ha mer informasjon om hva som skjedde ved bruk av denne funksjonen.

1.7.1.3 Scenario #3

1. Fant frem til den aktuelle 3D-modellen (gikk via "Modeller")
2. Synes 3D-applikasjonen var enkel og intuitiv, men forsto ikke helt begreper som "Clipping" og "Wireframe".
3. Synes "Hjelp" burde markeres bedre
 - o Foreslo bruk av tooltip/at enkel hjelp/beskrivelse poppet opp når man holdt musen over elementer
4. Trodde "vertikal bevegelse"-knappene var hoppeknapper.

1.8 "Nina", 22 år

1.8.1.1 Scenario #1

1. Navigerte seg til siden uten problemer.
2. Fant registreringssiden og laget konto uten problemer.
3. Logget inn uten problemer.
4. Gikk rett til footeren for å finne link til retningslinjene.
5. Fant "Last opp modell"-linken uten problemer.
 - o Ville helst at linken til retningslinjene på opplastingssiden skulle åpnes i et nytt vindu, slik at hun ikke måtte velge fil på nytt.
 - o Skrev tags i fnutter, ble forvirret av hjelpeteksten.
 - o Fant ikke "Passordbeskytt"-linken, synes den var for lite markert. Stusset også på stavelsen, mener det burde vært "Passordbeskyttelse".

1.8.1.2 Scenario #2

1. Mente at det under retningslinjene burde vært mer informert om at man kan velge en annen person som opphavsmann til opplastede modeller, og hvordan.
2. Greide å velge flere kontakter, men synes det var lite brukervennlig.

1.8.1.3 Scenario #3

1. Fant modellen, og synes det var fint at man kunne trykke på bildet av modellen for å komme inn på detaljvisningen. Likte den store klikeflate.
2. Greide fint å navigere i 3D-applikasjonen
3. Forsto ikke helt hva clipping var, trodde det kunne bety "Gå igjennom glass".
4. Ville helst ha linken til minimap nede sammen med de andre lenkene.
 - o Trodde at man kunne navigere via minimappet.

1.9 "Ingrid", 22 år

1.9.1.1 Scenario #1

1. Navigerte seg til siden uten problemer.
2. Fant registreringssiden med en gang.
 1. Registrerte et brukernavn med mellomrom.
3. Logget seg inn uten problemer.
4. Slet med å finne retningslinjene, men havnet til slutt på "Hjelp"-siden.
 1. Ville helst hatt en "Hjem"-link i menyen.
5. Fant "Last opp modell"-linken fort.
 1. Skjønte alt, men synes passordlinken var lite synlig.

1.9.1.2 Scenario #2

1. Klarte fint å velge flere brukere med CTRL-knappen, men synes det var en lite brukervennlige metode.
2. Synes måten "Credits" var satt opp på var *idiotisk*, vanskelig å se feltet i og med at det er huket av til vanlig, ville heller hatt at den ikke var huket av til vanlig, slik at man aktivt måtte velge om man har laget modellen eller om noen andre har laget den.

1.9.1.3 Scenario #3

1. Fant modellen, gikk til "Modeller" for å finne den.
2. Navigerte uten problemer i 3D-applikasjonen.
3. Ville helst hatt et annet ord enn "Clipping". Hun ser effekten og skjønner hva den gjør, men synes ordvalget var dårlig.
4. Forsto vertikal bevegelse.
5. Ville hatt en mulighet til å minimere GUI, da hun synes den tok for stor plass av synsfeltet. Når brukere blir kjent med applikasjonen, tror hun at man klarer seg med mindre knapper.

1.10 Erfaringer

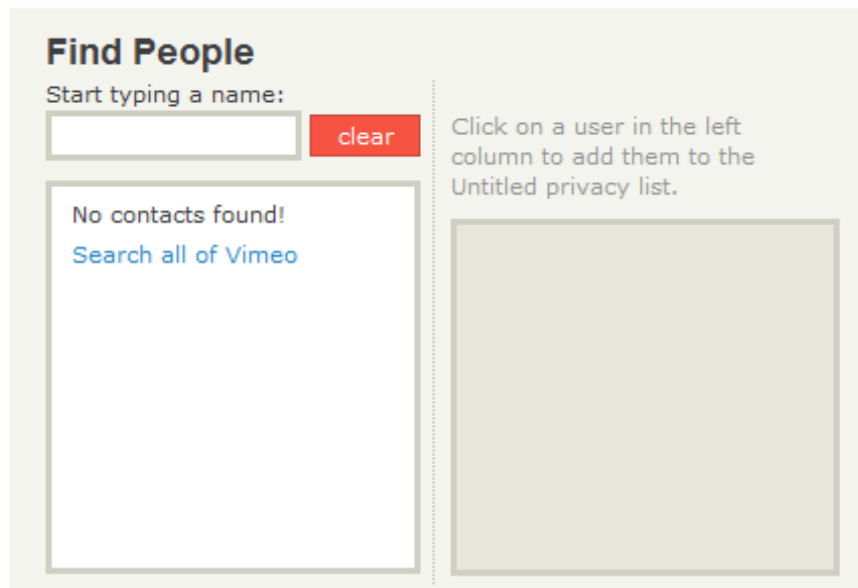
1.10.1 Websiden

- "Last opp modell"-linken
 - o Det største problemet med websiden var linken til opplasting av modeller. Dette er en av de viktigste funksjonene på websiden, og må være enkel å finne. Denne brukertesten viste det motsatte, da fire av seks (66 %) av testpersonene slet med å finne den, og noen fant den ikke uten hjelp (!). De fleste trodde enten at linkknappen ikke var en del av siden, at knappen var en del av headeren eller at knappen var kun en estetisk del av siden. Det ble også klaget på plasseringen, da de fleste opplyste at dette (øverst til høyre) var det siste stedet de så på en webside. Når

de ikke fant linken, gikk mange til undersiden "Modeller" for å se om man kunne laste opp modeller her.

- For å korrigere dette problemet bør "Last opp modeller"-linken gjøres mer synlig. Et enkelt grep for gjøre dette er å gi den et ikon. Den bør også flyttes ut fra headeren (kan plasseres under), og en link til opplastingen bør definitivt plasseres på siden "Modeller".
- "Synlighet" av opplastede modeller
 - Det kom mye kritikk mot "Synlighet"-delen av opplastingen, altså hvor man velger hvem skal kunne se modellen. De fleste synes den var for dårlig forklart, og ville ha en grundigere forklaring på hvert punkt. I tillegg ble menypunktene for gjemt bort, så og si alle slet med å finne ut hvordan man passordbeskyttet en modell. Noen ville også at passordet man skrev inn her skulle bli sensurert.
 - Det ble også mye forvirring rundt "Velg kontakter", altså der man kan velge hvilke kontakter som skal se modellen. Slik det var lagt opp nå var at man fikk alle kontaktene i en "multiple select"-liste, hvor man velger de kontaktene man vil med bruk av CTRL-knappen. Dette viste seg å være lite brukervennlig, og det kom tips om at det heller burde være to lister, en som viser alle dine kontakter, og en som viser valgte kontakter. Da kan man enkelt velge flere brukere, og man får god oversikt over hvilke brukere som er valgt/ikke valgt.
 - Under er et eksempel fra Vimeo (<http://www.vimeo.com>) som viser hvordan dette kan gjøres (på en litt annen måte, men samme poeng)

• People you choose



- Det ble kommentert på stavelen av "Passordbeskytt", og et nytt forslag var "Passordbeskyttelse". Bør vurderes endret.
- Noen ble forvirret av hjelpeteksten under "Tags", og ble usikre på om de skulle ha med anførselstegn eller ikke. Denne bør derfor endres, slik at det ikke er tvil om at man kun trenger å skrive: tag1, tag2.
- "Credits"
 - Forslag til navneendring: "Rettigheter".
 - Under testen var denne automatisk valgt til "Jeg har laget denne modellen", noe som førte til at mange overså den. Den burde heller gi brukeren to valg; "Jeg har laget denne modellen, og har opphavsrett til denne", og "Noen andre har laget modellen", og ingen av dem skal være valgt på standard. Dette får brukeren til å måtte ta et valg.
 - Det ble satt spørsmålsteget ved hva som faktisk skjedde når man valgte en annen opphavsmann:
 - Får opphavsmannen beskjed om at en av hans modeller er lastet opp, og av hvem?

- Må opphavsmannen godkjenne at modellen lastes opp før publisering?
 - Generelt burde det være mer beskrivende tekst rundt denne funksjonen.
- "Gå bakover"- og "Gå fremover"-knappene i opplasting av modeller er for like og er plassert for nærme hverandre. Disse bør dermed skille fra hverandre.
- Alle felt som må være fylt ut blir merket med en rød stjerne, men det kom tilbakemelding på at denne var for liten. Denne bør derfor gjøre større.
- "Retningslinjer" var vanskelig å finne, men de fleste gikk til slutt til "Hjelp" for å finne den.
 - Det kom et forslag om å endre navnet til "Rettigheter".
 - Akkurat hvor ellers denne kan plasseres er usikkert, men det kan muligens plasseres mer synlig link på forsiden.
 - Alle lenker til retningslinjene som er plassert i skjemaene (registrering av konto, opplasting), bør åpnes i nytt/i en JavaScript-popup.
- Registrering av konto
 - En link til registrering av konto bør plasseres i "logg inn"-modulen på siden, da flere forventer dette.
 - Noen lurte på hvor man skulle skrive inn sitt fulle og hele navn. Vi bør opplyse om at man kan fylle inn slik info (fullt navn, bosted, webside osv) senere.
 - Det kom frem at de fleste foretrakk å kun registrere seg med sin e-post adresse, og bruke denne som innloggingsnavn. Allikevel er det noen som vil ha begge deler. Derfor kan vi ha registreringen slik den er i dag, men gjøre det mulig å bruke e-post adressen sin og brukernavnet sitt som innloggingsnavn.

1.10.2 GUIen

- Navigasjonsknappene ser ut til å fungere akkurat som de skal, uten spesielle problemer for brukerne.
 - Fremover, bakover, høyre og venstre. Disse knappene fungerte uten problemer for alle brukerne. De skjønnte med en gang hva disse knappene gjorde og ser derfor ingen spesiell grunn til å forandre på disse. Oppdragsgiver virket også fornøyd med disse knappene.
 - Kamera tilt opp eller ned fungerte også. Var kun en av testpersonene som var litt i tvil om hva disse knappene gjorde. Men med en gang de prøvde knappene var det veldig tydelig hva disse gjorde, og det virket ikke som om noen hadde noen direkte problemer med å finne disse knappene heller.
 - Vertikal bevegelse opp og ned. Her gjorde jeg en stor feil når det gjaldt scenario, jeg glemte å legge til dette blandt en av testene. Men vi spurte testpersonene under testen hva de trodde disse knappene betydde. Og det det viste seg at 3 av de skjønnte hva det var uten større problemer. Siste testperson skjønnte til og med hva det var uten å bli spurt, og spurte heller oss om det var nettopp vertikal bevegelse disse knappene var til for. Så de fungerer virker det som.
- Clipping & Wireframe var vel egentlig den største svakheten i GUIen vår. Problemer her er ikke at brukeren ikke finner knappene, eller checkboksen, men det går mer på det at bruker ikke vet hva clipping & wireframe faktisk er.
 - Clipping er det at du nå kan "clippe" gjennom objekter i 3D verden, slik at du ikke lenger er en del av det "fysiske" i 3D modellen. Det var ingen av test personene som visste hva dette var da de begynte, men de skjønnte det fort når det ble testet i 3D modellen. Nå kunne man gå igjennom vegger og ting.
 - Wireframe er litt mer utbredt, og de som hadde noen erfaring med 3D generelt visste hva dette var, men igjen, andre visste ikke hva det var. Men også som nevnt over, med en gang det ble prøvd ut viste det seg fort hva de gjorde, og test personen skjønnte hva det var.
- Minimap skal fungere som ett lite kart der du kan se hvor du befinner deg i 3D modellen rett oven ifra slik at du kan se overlayen av 3D modellen. I testen fikk vi også forslag om at man kunne aktivt benytte seg av minimapet til å navigere seg rundt. Joda, ideen kan nok hende at

- er god den, men ser ikke for oss at dette skal hjelpe så mye i selva applikasjonen da dette kan bli kronglete og en unøyaktig måte å bevege seg rundt på.
- Tooltip vil være noe som dukker opp dersom man har musepekeren over en knapp litt lengre.
 - Vi fikk forslag fra en av testpersonene at en måte å kunne løse problemet med "clipping" og "wireframe" som det vi har nevnt over kunne være å ha tooltips til å dukke opp dersom man holder musepekeren over dette valget over litt lengre tid. Slik at man da skal kunne få forklart hva denne knappen gjør inne i selve applikasjonen. Ser ikke for oss at dette skal fungere på knapper som f.eks. navigasjon frem og tilbake, da disse knappene er så å si selvforklarende uten store problemer. Men dette kan jo løse problemet med å forklare terminologi som nevnt over.
 - GUI minimalisering, vi fikk også tilbakemelding om at GUIen kanskje tok opp litt mye plass i forhold til vinduet satt av til selve 3D visningen.
Her kom vi opp med 2 ideer:
 - Enten kan vi legge til en tastekombinasjon på tastaturet der man kan gjemme hele GUIen, og man må benytte seg av såkalte "hotkeys" for å navigere seg rundt, og skru av og på ting som f.eks. "clipping".
 - Det andre forslaget var da å lage en mini versjon av selve navigasjons panelet, slik at man minimalisere selve GUIen og man kan fokusere på selve 3D modellen, men at man fremdeles kan bruke kun musen for navigasjon.
 - GUIen vi brukte for brukertesting var vel det man kan kalle fargeløs og grå. Dette skal vi nå forandre på. Slik at den blir mer riktig i forhold til fargene brukt på selve websiden, og logoen. Dette vil også gjøre at GUI elementer blir lettere å finne, siden de da gjerne vil stå litt mer i kontrast i forhold til resten. Dette kommer jo også ann på selve 3D modellen så klart. Men det burde hjelpe uansett.

1.11 Konklusjon

Vi har fått mange gode tilbakemeldinger og lært mye av denne testen, og det største problemet med prototypen var nok plasseringen av "Last opp modell"-linken og usikkeren rundt "Synlighet" og "Credits" under opplasting av modeller. I tillegg fikk vi veldig mye annen god feedback, som vi sannsynligvis aldri selv hadde tenkt på. Antallet testpersoner var også greit, da vi etter 4-5 personer begynte å få mye av de samme tilbakemeldingene.

Vi har også lært å lage mer grundige scenarioer, da noen av scenarioene var utydelige, og skapte litt trøbbel for noen av testpersonene. I våre scenarioer benyttet vi oss av en liste over steg som brukeren skulle følge, noe som guidet og hjalp brukeren for mye.

Det kunne være vanskelig til tider som testleder og holde seg nøytral når testpersonene satt seg fast, men dette var noe som bedret seg etter hvert som vi testet.

1.12 Kilder

Perifer brukertesting, Eidar, <http://www.iallenkelhet.no/perifer-brukertesting>

Usability testing, http://en.wikipedia.org/wiki/Usability_testing

Vedlegg E:

Gantt-diagram



Vedlegg F:

Statusrapporter

2 Statusrapport 1, 3D-visualisering

28.02.09

3D-Visualisering

Jon Espen Kvisler

Aleksander Stalsberg

2.1 Tilstand nå:

Tilstanden nå er at oppdragsgiver har bestemt seg for hvilket rammeverk de har tenkt til å utvikle deres 3D motor i. Og derfor hvilket rammeverk vi som gruppe også må sette oss inn i. Det rammeverket som visualisere.no har valgt heter LWJGL (Lightweight Java Game Library), og er ett relativt enkelt system med støtte for 3D.

I tillegg til dette har vi over mail med oppdragsgiver nå bestemt oss for hvordan man skal kunne manøvrere rundt i denne 3D modellen. Siden LWJGL er utviklet som en spill motor har oppdragsgiver og vi funnet ut at for å gjøre dette lettere baserer vi oss på at vi "beveger" oss rundt i 3D verdenen som du vanligvis ser i såkalt first-person-view. Det vil med andre ord si at det vi styrer inne i 3D verdenen er selve kameraet som beveger seg rundt. Det vi har tenkt oss ut er å bruke både knapper i en GUI og taster på tastaturet til å navigere dette kameraet rundt. Vi har sett i eksempler på kode og kilde for LWJGL at det er ingen veldig stor oppgave å knytte knapper og ikke minst musebevegelse til dette. Det vil med andre ord gi oss en større mulighet for bevegelsesfrihet når du kan bruke musa og for eksempel WASD som i de fleste andre spill for navigering og bevegelse.

Vi har bestemt oss for å utvikle et rammeverk for websiden selv, altså et slags CMS-system. Ulempen med å utvikle dette selv er at risikofaktoren for prosjektet øker, da vi kan ende opp med å ta vann over hodet, men dette ser vi på som lite sannsynlig. En annen ulempe er dokumentasjon og videre utvikling for oppdragsgiver, da et ferdigutviklet system (ofte) har mange brukere, god dokumentasjon og fremtidige oppdateringer. Fordelene med å utvikle et eget system er at vi får akkurat det vi vil ha, og slipper å forholde oss til mange funksjoner vi ikke har bruk for. Vi kunne selvfølgelig tatt et eksisterende system og tilpasset dette til våre behov, men vi tror dette hadde vært minst like tidkrevende. Derfor er det veldig viktig at det vi utvikler er skalerbart for fremtiden, enkelt i bruk og at koden er godt kommentert. Vi skal benytte oss av, i den grad det er mulig, objekt-orientert programmering som PHP5 støtter, og data skal lagres i en relasjonsdatabase, med MySQL.

For å hente inspirasjon til oppsett, systemdiagrammer osv har vi lånt et par tidligere bacheloroppgaver fra biblioteket.

2.2 Problemer og løsninger:

Problem:

Vanskeligheter for å kommunisere med oppdragsgiver da han befinner seg på andre siden av landet. Og dro også på ferie for øyeblikket.

Løsning:

Få til ett, eller flere, SKYPE møter med oppdragsgiver for tettere kommunikasjon og samarbeid.

Problem:

Brudd i forhold til tidsplanen

Løsning:

Jobbe ekstra timer

2.3 Tidsplan:

I forhold til GANT-diagrammet skulle vi nå vært ferdig med research på aktuelle teknologier, og sett på GUI-konvensjoner. På grunn av at vi måtte jobbe mer med forprosjektrapporten enn vi hadde planlagt, ligger vi noe etter tidsplanen. Men vi har sett en del på teknologier og løsninger, og gjort en del research på blant annet LWJGL.

2.4 Planen fremover:

For det første skal vi ta ett møte med oppdragsgiver så fort han er tilbake fra ferien for å kartlegge hva som må gjøres fremover, og også hvordan det går med deres del av prosjektet ettersom vi også er avhengige av deres del for å begynne å lage noe direkte konkret. Men vi har begynt med utkast til diverse former for GUI og løsninger på GUI.

Det neste vi skal gjøre er å tegne opp et systemdiagram, usecase-diagram, operasjonelle og funksjonelle krav, altså en kravspesifikasjon. Vi må også komme ajour med tidsplanen, altså bli ferdig med alle research, noe vi snart er i mål med. Vi må fortsatt se på mulighetene for å ta i bruk et PHP rammeverk, som for eksempel CakePHP⁴⁴ eller CodeIgniter⁴⁵, som begge er gratis.

3 Statusrapport 2, 3D-visualisering

30.04.05

3.1 Tilstand

- Overordnet design ferdig, og alle handlinger bruker skal kunne gjennomføre på websiden er kodet i HTML/CSS (og noe JavaScript-funksjonalitet). Interaksjonsdesign og brukergrensesnitt er laget basert på forskjellige designpatterns (ui-patters.com, pattertap.com), og inspirasjon er hentet fra lignende sider (flickr, youtube, vimeo, facebook). Vi vil lage designet så brukervennlig og robust som mulig, og med vår tidsramme er dette et viktig grep. Utviklingen kan følges her: <http://visualisere.org/dev/brukertesting/> (work in progress)
- Planleggig av funksjon som tar i mot fil (3D-modell) fra bruker er i gang. Stikkordene her er "stol aldri på data fra bruker", altså at alt som kommer fra userinput må sjekkes, dobbeltsjekkes og sjekkes igjen. En slik opplasting- og kjøring av fil kan være farlig hvis fil inneholder et skjult PHP-script med ondsinnet kode. Handlinger for å forhindre dette er: sjekke filendelse, gi fil et random filnavn (som holdes styr på i databasen), legge filen utenfor /www-mappen (slik at filen blir utilgjengelig via nettleseren).
- Forslag til logo er ferdigstillt og vektorisert. I og med at vi kun jobber ut i fra en arbeidstittel (Visualisere), kan det komme forandringer på denne. Men den logoens konsept (en 3D-boks med en V for "Visualisere") er godkjent av oppdragsgiver, og kan enkelt endres ved navnenring. Oppdaterte versjoner av logo legges her: <http://visualisere.org/dev/logo/>
- GUI skisser er i ferd med å bli digitalisert. Venter fremdeles svar fra Christian anngående ett spm her. Men dette burde være på plass innen onsdag. Legg merke til at disse kan jo bli voldsomt forandret etter brukertesting dersom det viser seg at det ikke funker.

⁴⁴ <http://cakephp.org/>

⁴⁵ <http://codeigniter.com/>

- Jobber også med å gjøre klar en prototyp av selve 3D applikasjonen for utforming av GUI for å bli mest mulig brukervennlig. Denne vil bli laget i powerpoint, og vil, etter planen være klar til testing, seneste over påske.

3.2 Problemer og løsninger

3.3 Tidsplan

- Vi skulle vært ferdig med den grafiske profilen, noe vi ikke er. Har laget en logo og fargepalett, som så og si er godkjent av arbeidsgiver (logo trenger noen justeringer).
- Brukertesting er ikke gjennomført. Websiden er klar til å brukertestes, vi mangler kun scenarioer.
- GUI ferdig digitalisert i løpet av onsdag den 1/4. Der vi da også kan sende forslaget til oppdragsgiver for at han skal se over det.
- Powerpoint-prototyp under utvikling hos Aleks. Burde være ferdig enten rett før, eller klar til testing etter påske.

3.4 Planen fremover

- Lage scenarioer for brukertesting
- Sette opp selve prototypen for brukertesting
- Teste GUIen til 3D applikasjonen og se om man kan få innført denne i 3D applikasjonen vi har fått tilsendt av oppdragsgiver.
- Lage utseende til websiden
- Begynne kode funksjonalitet inn i websiden
- Jobbe med funksjon som tar i mot korrekt fil fra bruker (en 3D-modell som er av "lovlig" filtype), sjekker for farlig kode, og legger filen på et sikkert sted.

4 Statusrapport 3, 3D-visualisering

30.04.09

4.1 Tilstand

- Vi har gjennomført brukertesting, med gode tilbakemeldinger. [Rapport](#) .
- Den grafiske profilen er så og si ferdig, og oppdragsgiver har godkjent logoen.
- Templatesystem for websiden er ferdig, samt at brukersystemet
- Vi hadde et kort statusmøte med oppdragsgiver den 22. april; [referat](#) .
- Vi har begynt med oppsettet til den endelige rapporten.

4.2 Tidsplan

- I følge GANT-diagrammet skal vi være ferdig med all utvikling/design fredag 8. mai, og begynne på rapporten da. Dette tror vi skal gå, men det kommer til å kreve mye jobbing. Vi har publisert en [prioriteringsliste](#) (til oppdragsgiveren), som viser hvordan vi ligger an i forhold til kravspesifikasjonen. Vi håper på å i alle fall få ferdig alle punktene under prioritet 1 og 2.

4.3 Planen fremover

4.3.1 Webside

- Gjøre forandringer i designet basert på tilbakemeldinger fra brukertesten.
- Fikk en forespørsel fra oppdragsgiver om vi kunne kjøre den opplastede filen igjennom et javaprogram, men avventer mer informasjon (skulle komme i løpet av uka). Har gjort litt research, og dette skal være mulig å gjøre fra de innebygde PHP-funksjonene `exec()` eller `system()` som kan brukes til å kjøre kommandolinje-operasjoner direkte på server. Allikevel kan dette by på problemer, da det kan kreve en del serverkonfigurasjon. Men som sagt, oppdragsgiver skulle sjekke dette litt ut, og komme med mer informasjon denne uka.
- Fortsette å jobbe nedover kravspesifikasjonen, holder nå på med visning av modeller, sortering av visningen, og visning av én modell.

4.3.2 GUI

GUI skal gjøres ferdig og forandres etter ønske fra oppdragsgiver etter møte. Forandringene som skal skje er:

- Oppdragsgiver vil ha en "snapshot" knapp for at brukere av 3D applikasjonen skal kunne lagre posisjoner og ett bilde av denne posisjonen. Dette skal fungere som en måte å kunne navigere seg rundt til "keypoints" i 3D modellen. Da f.eks. døren til ett hus, eller ett overblikk.
- For at funksjonen over skal fungere vil også oppdragsgiver ha en mulighet til å bringe opp en samling av det som er av "snapshots" i en 3D modell for så å kunne navigere seg til det stedet.

Vi hadde også en brukertesting av websiden, og GUIen der vi fikk veldig mye bra og positiv tilbakemeldinger, samt vi fikk også gode innspill over ting som kanskje burde enten forandres eller forbedres. Forandringer vil være:

- GUIen skal såklart få farge. Denne skal gjøres så lik som mulig logoen vi har, men med de justeringer som kreves for at elementer av GUIen skal være synlige.
- Under brukertesting fikk vi også høre at enkelte følte at selve GUIen tok opp for stor plass i selve 3D applikasjonen. Derfor skal det også lages en liten versjon av GUIen der du da legger hele GUIen ned, for å ha en liten "nav-bar" på bunnen av applikasjons vinduet. Dette vlr da mest for de såkalte "superbrukerne" der de da gjerne bruker hurtigtaster istedenfor å hele tiden navigere seg rundt ved hjelp av navigasjons knappene i den vanlige GUIen.
- En annen ting vi lærte under brukertesting var at det var veldig få som skjønnte hva "clipping" og "wireframe". Det vil si, de visste ikke hva det var, men da de under scenario fikk beskjed om å bruke disse funksjonene hadde de ingen problemer med å verken finne frem, eller skjønne hva denne funksjonen gjorde etter at de hadde prøvd den. Ett tiltak vi skal gjøre for å forbedre dette er å la det dukke opp ett "tooltip" så lenge man har musepekeren over dette området over en viss tid. Dette er er noe som skal være der for knapper som "clipping" og "wireframe", men ikke for de store navigasjons knappene ettersom disse var tydelige nok i seg selv ifølge brukertesting.

4.3.3 Prosjektet

- Pusse litt på rapporten fra brukertesten.
- Utføre det som er av forbedring av GUI.
- Gjøre ferdig webside.

- Begynne med utforming av rapport, og bestemme oss for hvilken tekstbehandler vi skal bruke (Word, InDesign eller LATEX).

Vedlegg G:

Møtereferater

1 Referat, veiledning 11. mars

1.1 Tilstede

- Aleksander
- Jon Espen
- Anne

1.2 Saker

- Gå over UseCase-diagrammet, fjern unødvendige relasjoner
- Vi skal ha møte med oppdragsgiver fredag 13. mars kl 13:30
 - Føye på spørsmål til møte med oppdragsgiver.
 - Hvor stor er vår arbeidsmengde?
 - Målestokk for 3D-visning
 - Kanskje velge to punkter, og få opp avstanden mellom disse
 - Mulighet for visning av wireframe av modeller?
 - Fysikken i visning, skal det være mulig å gå igjennom vegger?
- Testing
 - Viktig med testing av design, da vi fort kan se oss blinde, og gå glipp av viktige feil/mangler.
 - Lage en prototype av design
 - Vurdere å sette filme testingen, for å ikke gå glipp av viktige observasjoner
 - Mulig testperson: Frode Volden

Neste møte blir om to uker til samme tid, altså onsdag 25. mars kl 13:00.

2 Møte over Skype 13.mars

2.1 Deltakere

Jon Espen Kvisler (sekretær)
Aleksander Stalsberg
Visualisere.no v/Kristian botnen

2.2 3D-applikasjon

- Brukergrensesnitt må på plass først, slik at man kan bevege seg rundt i 3D-rommet.
- Ellers er Visualisere.no i rute med 3D-applikasjonen.
- Grafiske filer til GUI kan leveres i hvilket som helst format (som er Photoshop-kompatibelt), og vil bli konvertert til det formatet som gir best kvalitet i forhold til filstørrelse. PNG og JPG er da et aktuelle formater.
- Visualisering.no tar seg av selve kodingen, men vi ble enige om at vi kunne få kodeoppgaver som knyttet seg til 3D-applikasjonen.
- Applikasjonen tar pr i dag i mot filer av typen .3ds og .obj, som begge er godt støttede formater. Skal i fremtiden utvides til flere formater.
- Alle modeller må være modellert i korrekt målestokk (1 meter er 1 meter)
 - Viktig å informere brukere om dette ved opplasting av modell

- Det mulig å vise modellens wireframe, men skal være avskrudd som standard. Lage en knapp som skrur denne funksjonen av og på.
 - Ikke veldig nyttig ved store og komplekse modeller (feks et hus), men kan være nyttig ved visning av mindre modeller (feks prototyper)
- Fysiske begrensninger
 - Akkurat nå kan man fly akkurat hvor man vil i modellen (altså igjennom vegger o.l.), men dette er ikke optimalt da det kan forvirre brukere.
 - Bør være av som default, men med mulighet til å skru på (samme som wireframe)
 - Egentlig burde man aldri kunne fly igjennom "bakken", men det er vanskelig å definere hva som er "bakken" når tjenesten skal kunne ta i mot alle slags modeller.
- Minimap
 - Samme som wireframe, av som default, men mulighet til å skru på.

2.3 Web-siden

- Opplasting av filer
 - Kjøre fil igjennom en PHP-funksjon som sjekker at filen har en filendelse som er lovlig, samt sjekker at filen ikke inneholder ulovlige tegn/tegnkombinasjoner
 - Opplastede filer skal publiseres med en gang, og trenger **ikke** godkjenning av administrator/moderator.
 - Om en bruker laster opp en fil uten opphavsrett, eller usømmelig innhold, kan dette fjernes av moderator/administrator senere.
- 3(4?) brukernivåer
 - Uregistrert bruker
 - Registrert bruker
 - Moderator
 - Site administrator
 - I tillegg vil selvfølgelig de med servertilgang sitte med høyeste tilgang, med direkte tilgang til database og opplastede 3D-modeller
- Den som laster opp en modell, bør selv laste opp thumbnails/forhåndsvisningsbilder
 - Vi kan ikke definere vinkler som en modell skal avbildes i. Blir litt som at Finn.no skulle tatt bilder av din leilighet.
- Litt usikkert om vi skal benytte oss av kategorisering eller tags på modeller, men vi ble enige om vi kunne legge til rette for bruk av tags i alle fall.
- Kommentarer av modeller blir overflødig, Visualisere.no tror ikke dette kan bidra til noe særlig konstruktivt.
 - Samme med karaktersetting på modeller
- Lagring av favorittmodeller i sin bruker bør implementeres.
- Visning av 3D-modellen må foregå i et pop-up-vindu
 - Bedre tilgang til systemressurser
 - Derfor er det viktig å samkjøre grafisk profil både på web og i 3D-applikasjon slik at brukere kjenner seg igjen.
- Bruk av CAPTCHA er ønskelig ved brukerregistrering
 - Hindrer spam
 - Bidrar til økt sikkerhet
- Utseende
 - Samme look n feel før følge nettstedet
 - Vi skal lage et generisk utseende, med "3D-visualisering" i fokus. Mulig Visualisere.no vil definere andre grafiske profiler senere
- Dokumentasjon!
 - **Ekstremt** viktig med kommentering og god kodestruktur, da det er meget sannsynlig at andre enn oss må inn i koden og gjøre endringer/utvidelser.

2.4 Diverse

- Selve tjenesten
 - Poenget med denne tjenesten er at alle skal kunne laste opp modeller når de vil, og at de blir publisert med det samme. Visualisering.no skal ikke legge føringer på hva som kan og ikke kan lastes opp.
 - Retter seg mot arkitektur i begynnelsen.
- Kravspesifikasjon
 - Sende denne over til Visualisere.no så fort vi kunne
 - Sette opp en prioriteringsliste over funksjonalitet
- Eksisterende løsninger
 - Mye som har skjedd de siste 6 månedene, og Visualisere.no tror det vil skje mye fremover i forhold til 3D-visualisering på nett.
 - Pr i dag finnes det ingen løsninger hvor man kan laste opp modeller gratis, og vise disse i en web-side.
 - Unity3d.com
 - Ble for to måneder siden tilgjengelig for andre plattformer enn Mac.
 - Koster mye penger
 - Finnes diverse shockwave løsninger for visning av 3D på web
 - <http://showcase.directorforum.com/>
- Visualisere.no kunne trenge flere test-modeller, så vi ble enige om at vi kunne sende over våre modeller fra kurset IMT2471 -- 3D-modellering til Visualisere.no

3 Referat, veiledning 27. april

3.1 Tilstede

Aleksander (Kom etter 15 min)

Jon Espen (Sekretær)

Anne Kristin Kvitle

3.2 Saker

- Brukertesting
 - Vi burde ha med mer i brukertestrapporten om hvordan selve brukertesten gikk, og hva vi lærte av prosessen, og hva som kunne vært gjort anderledes.
 - Må også legge ved scenarioene i rapporten.
- Forslag til rapport av endelig innlevering
 - Veldig viktig at vi beskriver problemstilling og mål, samt at vi drøfter og evaluerer oss selv.
 - Design/utvikling er mer en beskrivelse av hva vi har gjort.
 - Vi burde ha med hva våre forutsetninger var før prosjektet, slik at vi kan beskrive hva vi har lært av prosjektet.
- Diverse
 - Den muntlige presentasjonen av prosjektet er på 30 min, og dette inkluderer opprigg og spørsmål og svar.
 - Innlevering av rapport den 20. mai til kopimannen, og innlevering til studenttorget den 25. mai.

Foreslår å legge neste veiledningsmøte til **torsdag 8. mai, kl 13:00.**

4 Referat, skypemøte 06.05.09

4.1 Tilstede

- Kristian (oppdragsgiver)
- Jon Espen (sekretær)
- Aleks

4.2 Hva som gjenstår, som må bli ferdig

- Opplastede filer skal automatisk mates til et javaprogram som konverterer modellen til ønsket format for visning.
 - Den opplastede .obj-filen blir konvertert til en fil ved type ".serialize" (for øyeblikket, men denne kan endres), og får navnet <filnavn>.obj.serialize. Etter denne konverteringen trengs ikke den originale filen, og denne kan slettes. Mulig det blir to filer, en for 3d-modellen og en for teksturene.
- Kommentere ferdig kode
- Er en bug med opplastingsknappen i Opera, slik at den legger seg over logoen. Dette må fikses, og siden vi allikevel må flytte knappen (siden mange slet med å finne den i brukertesten), skal vi teste den i Opera når den er flyttet. (Til nå har vi testet i Firefox, Internet Explorer 6 og 7, Safari (Win) og Chrome)
- Innlegging av nyheter dynamisk.
- Ved sletting av modeller skal alt om modellen slettes (mappestruktur, databaseinfo, thumbnail og teksturer). Det som gjenstår er sletting av teksturer.
- All debug-informasjon fjernes før lansering.

4.3 Søk

- Vi regner ikke med å bli ferdig med søkefunksjonen til fristen, dette er noe vi bør legge til senere da dette er en veldig nyttig funksjon.

4.4 Thumbnails

- Det burde være mulig å endre thumbnail for en modell etter at den er opplastet for brukeren. Derfor er dette noe vi skal fokusere på *hvis* vi får tid til overs.

4.5 Telling av antall visninger av modell

- Denne skal oppdateres via PHP ved visning av modellen.

4.6 GUI

- Ferdiggjøres og sendes til oppdragsgiver

4.7 Logo

- Logo skal legges ut i vektorformat og høyoppløselig format, slik at oppdragsgiver har tilgang på denne. Lagt ut her: <http://visualisere.org/dev/logo/fin/>

4.8 Neste møte

Neste møte blir onsdag 13.mai kl 13:30 (via Skype)

5 Referat skypemøte, 15.mai 2009

Tilstede

Aleks

Jon Espen (sekretær)

Visualisere v/ Kristian

Siden Aleks og Jon Espen har vært opptatt med rapportskrivning siden sist statusmøte, har det ikke skjedd nevneverdig verken på GUI eller websiden. Det skal heller ikke skje store forandringer frem til innlevering (bortsett fra tilkobling av 3D-applikasjonen).

Vi ble enige om å sende en foreløpig rapport til Kristian, så han kan lese igjennom før innlevering.

5.1 3D-applikasjon

- Status
 - GUI er under implementering.
 - Litt problemer med teksturer og plassering av disse.
 - Ble ikke noe av snapshots til innlevering, på grunn av problemer med lesing fra klienttjener og rettigheter rundt dette.
Vil ikke snoke i filene til folk. Mulig dette ble implementert senere.
 - Websiden: Del 1, altså den som konverterer opplastede modeller til riktig format for 3D-applikasjonen, er ferdig. Del 2, som starter visningen av modellen via websiden er ikke helt i boks enda.
 -
- Krav til GUI
 - Finnes her: <http://www.visualisere.no/main/nb/viz-prosjekt-2009/3d-interface.html>
- Målestokk er ikke ferdig, og kommer ikke til å bli det til innlevering. Vanskelig å implementere i og med at modeller kan komme i feil skala, og dermed umulig å måle. Men på prosjekter som omhandler feks golfbaner er det mulig, og nødvendig.
- I fremtiden kunne det bli lagt opp til at man skal kunne endre farger på flater i 3D-modeller, men dette er det lengste man skal kunne endre modeller. 3D-applikasjonen skal ikke være et verktøy for redigering, men kun visning av modeller for å holde applikasjonen så liten og rask som mulig.

5.2 Selve tjenesten

- I starten vil tjenesten være gratis for alle brukere, men i fremtiden, hvis det blir en betydelig brukerbases, kan det være mulig å legge inn en betalingsversjon, hvor man for eksempel betaler for ekstra funksjoner.
- Det skal heller ikke legges inn reklame på nettstedet før det eventuelt er bygd opp en brukerbases.
- Det finnes ingen andre tjenester som tilbyr akkurat det Visualisere.org skal tilby, men i det siste (de siste 6 månedene) har det kommet ulike tjenester som kan minne om vår tjeneste.

- Visualiseres fordel er at alle brukere som har et 3D-modelleringsprogram kan lagre sin modell i et format som Visualisere.org godtar, og laste opp sin modell. Denne kan så vises til alle som har en PC med internetttilgang og Java installert.
- Google API, Unity 3D, Autodesk DWG. Alle disse er mye tyngre, og alle tilbyr kun deler av det Visualisere tilbyr.
- Markedsføring
 - Ingen planer om å bruke penger på finansiering
 - Kun markedsføre via diverse forum og utnytte nettverket Visualisere har.

5.3 Websiden

- Bruke StatCounter for å måle statistikk.

Neste møte blir: **Torsdag 21. mai kl 11:00**

Vedlegg H:

Logg

6 Logg

6.1 Uke 3-7

13.01.09 -- 11.02.09

Begynnelsen av prosjektet, og vi utarbeidet en forprosjektrapport hvor vi beskrev oppgaven, avgrensinger og formulerte mål. Tidsskjema ble utarbeidet, og tegnet opp i et gantt-diagram. Prosjektavtale ble sendt til oppdragsgiver, som tok et utskrift, skrev under, og skannet inn, slik at vi fikk hans underskrift.

Arbeid og ansvar ble fordelt, og vi bestemte oss for å dele arbeidsoppgavene i to.

6.2 Uke 6-8

02.02.09 -- 27.02.09

Vi satte i gang med research, i forhold til web så vi på aktuelle teknologier og systemer og vurderte hva som passet best til vårt prosjekt. Kom frem til å utvikle et selv
Når det gjaldt GUIen så vi over tidligere prosjekter fra oppdragsgiver, og andre systemer som omhandlet 3D.

6.3 Uke 9-11

27.02.09 -- 13.03.09

Fargevalg som skulle brukes i den grafiske profilen ble bestemt, og research på logo ble gjennomført. Skisserte flere utkast, og valgte til slutt et konsept. Denne ble oversendt oppdragsiver for godkjenning. Etter godkjenning ble logoen vektorisert og fargelagt.
For bedre forståelse av utviklingen av GUI ble det fokusert på tidligere pensum i Ergonomi i Digitale Medier og andre bøker som omhandlet liknende temaer.

6.4 Uke 12 - 17

16.03.09 -- 03.04.09

Layout for websiden ble planlagt og skissert, og en wireframe ble utviklet i HTML og CSS. Denne ble grunnlaget for resten av siden.
De første skissene av GUIen ble tegnet opp. Begynte også å lage noen av disse i Photoshop, men ingen av skissene er valgt enda.

15.04.09 -- 22.04.09

Utvikling av prototyper. I denne perioden var det utvikling og oppsett av prototype som var i hovedfokus.
Forberedelse til brukertest, utvikling av scenarioer.
Samtidig begynte vi utvikling av websiden, med oppsett av databasestruktur og malsystemet.

22.04.09

Brukertest ble gjennomført med 7 testpersoner.
Fikk en del bra tilbakemeldinger underveis, og satte oss så ned for å fikse disse eventuelle problemområdene både på websiden og på GUIen.

6.5 Uke 18 - 21

23.04.09 -- 30.04.09

Fortsatte utvikling av websiden. Visning og sortering av modeller, detaljvisning for en modell. Brukersystem med innlogging og utlogging.

Skrev om models-klassen, slik at alle endringer på en modell skjer i model-klassen.

Laget "Modeller siden sist"-visningen av modeller. Laget en funksjon som gir mulighet til å bla mellom modellene (pagination-klassen), ved nummerering i bunn.

Fortsatt med utvikling og sammensetting av GUIen. Det meste begynner å komme på plass, og kontrollpanelet er satt sammen.

05.05.09

Fikk beskjed av Kristian om at teksturer skulle lastes opp sammen med modellen. Måtte da lage et ekstra seg i opplastingsprosessen, og en ny tabell i databasen.

GUI ferdig satt opp, godkjent av oppdragsgiver (Kristian), og fikk nå beskjed om å dele opp elementene hver for seg.

Dette ble gjort samme dag og sendt til Kristian.

06.06.09 -- 08.06.09

Laget nyhetsfunksjon og søkefunksjon.

09.05.09 -- 20.05.09

Rapportskriving.

Mye bugfiksing og småpuss på websiden

Vedlegg I:

Eksempel på bruk av 960.gs



FEATURED

Pengebingen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut mauris tortor, elementum et, pulvinar ullamcorper, condimentum eu... av [Jon Espen Kvistler](#)

[Se modell](#)

[LAST OPP MODELL →](#)

NYHETER

Dette er en nyhet 17.05.2009

I dag har vi feiret 17. mai og spist is.
Det var fint.

Modeller

Sortering [Siste](#) [Nye siden sist](#) [Populære](#)



INTET BILDE

Noe
av jonespen



INTET BILDE

Noe
av jonespen



INTET BILDE

Noe
av jonespen



INTET BILDE

Noe
av jonespen



INTET BILDE

Noe
av jonespen



INTET BILDE

Noe
av jonespen

Vedlegg J:

Statistikk, webside

Alle tallene er tatt ifra visualisere.no, men det er grunn til å tro at tallene vil være representative for visualisere.org når den kommer online.

Screen resolution: visualisere.no

45.30% Unknown
 31.07% 1280x1024
 19.16% 1024x768
 2.31% 1600x1200
 1.76% 1152x864
 0.40% 800x600

OS: visualisere.no

| Prosentandel | Operativsystem |
|--------------|----------------|
| 67.32 | WinXP |
| 20.48 | WinVista |
| 5.83 | Win2003 |
| 3.90 | MacOSX |
| 0.85 | unknown |
| 0.56 | Win98 |
| 0.44 | Linux |
| 0.31 | Win7 |
| 0.21 | Win2000 |
| 0.06 | iPhoneOS |
| 0.04 | SymbianOS |

Nettlesere: visualisere.no

| # | Treff | Prosentandel | Nettleser |
|---|--------|--------------|-----------------------|
| 1 | 691629 | 78,55 % | Firefox |
| | 482553 | 54,80 % | Firefox/2 |
| | 207935 | 23,62 % | Firefox/3 |
| | 1116 | 0,13 % | Firefox/1 |
| 2 | 121438 | 13,79 % | MSIE |
| | 61822 | 7,02 % | MSIE/7 |
| | 56774 | 6,45 % | MSIE/6 |
| | 2637 | 0,30 % | MSIE/8 |
| 3 | 20819 | 2,36 % | Opera |
| | 20362 | 2,31 % | Opera/9 |
| 4 | 19739 | 2,24 % | Netscape (compatible) |
| 5 | 9439 | 1,07 % | Safari |
| | 8622 | 0,98% | Safari/525 |

Vedlegg K:

Kodekonvensjon

1 PHP og JavaScript

- Konstanter skal skrives med store bokstaver: CONSTANT.

Funksjoner

```
func(argX, argY) {  
    ...  
}
```

Klasser

```
class ExampleClass {  
    private $var;  
  
    public function setVar($var){  
        $this->var = $var;  
    }  
}
```

2 HTML-maler

Lesbarhet

I HTML-maler skal så lite PHP-kode som mulig brukes, og det skal legges vekt på oversiktlig og lesbarhet. Derfor skal den alternative syntaksen for if/else, while, foreach og så videre benyttes⁴⁶ og PHP-blokker skal ikke inneholde HTML eller gå over flere linjer (til den grad dette er mulig).

Grunnen til dette er at malene *ikke* skal inneholde logikk, så for eksempel designere kan lese koden. Nedenfor er et eksempel på denne strukturen:

```
<h1><?php echo $title?></h1>  
<li>  
<?php foreach($elements as $element) : ?>  
    <ul><?php echo $element ?></ul>  
<?php endforeach; ?>  
</li>
```

Semantikk

Alle navn som gis til id- og class-attributter skal beskrive innholdet i elementet:

```
<ul class="products">  
    <li>Product #1</li>  
    <li>Product #2</li>  
</ul>
```

⁴⁶ <http://no2.php.net/manual/en/control-structures.alternative-syntax.php>

Vedlegg L:

Mappestruktur

```

/
models/          * inneholder alle modeller lastet opp
  <brukernavn>/    * alle modeller til en bruker
    <filnavn>_<id>/  * en modell
      maps/         * alle teksturene til modellen

tmp/            * midlertidig mappe for modeller

www/
config/
  config.php     * inneholder de viktigste
                 konfigurasjonsvariablene

lib/
  lang/
    no-nb.lang.php * språkvariabler
    _ajax.funcs.php * all ajax funksjonalitet
    _init.php      * fil som inkluderer nødvendige
                   klassebibliotek og filer
    _shared.php    * diverse mindre funksjoner som blir
                   brukt av flere klasser
    categories.class.php * Henting av kategorier
    db.class.php   * databasetilkobling og singleton
                   databaseklasse
    error.class.php * klasse for error-håndtering
    helper.class.php * generell helper-klasse
    model.class.php * klasse for håndtering av én modell
    models.class.php * klasse for håndtering av flere
                   modeller
    news.class.php * håndtering av nyhetsartikler
    paginator.class.php * singleton klasse, genererer
                   nummerering på opplisting av
                   blant annet modeller
    search.class.php * søkefunksjonalitet
    sidebar.class.php * klasse som generer sekundærinnhold
                   og henter alle moduler
    sidemodule.class.php * klasse som generer en modul, og
                   inkluderer korrekt mal
    template.class.php * klasse som genererer riktig side
                   ut i fra hva brukeren navigerer seg
                   til, og inkluderer riktig
                   sekunderinnhold. Brukes også til å
                   hente header og footer
    upload.class.php * stor klasse som tar seg av
                   opplasting av 3D-modeller,
                   modellbilder og teksturer og
                   inneholder alle
                   filsjekkingsfunksjoner

    user.class.php * oppretting av nye brukere,
                   innlogging, all logikk som omhandler
                   brukere

scripts/
  captcha/      * alle filer brukt av captcha-scriptet

```

```

uploadify/ * alle filer brukt av uploadify
classTextile.php      * Textile som brukes til formatering
                        av tekst (i nyhetene)
makeThumbnail.php * script som forminsker bilde
template/
css/
    960.css      * grid system
    colors.css  * alle farger i stilsettet
    css.css     * formatering av utseende til layout
    forms.css   * formatering av skjemaer
    reset.css   * tilbakestilling av standardutseende i
                        nettleser
    text.css    * formatering av all typografi
html/
    * filer som begynner med __ : de fire filene
    som utgjør basisstrukturen for siden (topp,
    hovedinnhold, sekundærinnhold, bunn)
    * filer som begynner med _ : alle html-filer som
hører til i sekundærinnholdet
    * resten : hver fil tilhører en egen side
img/      * alle bilder brukt til layout
icons/    * ikoner
js/
    DD_belatedPNG.js      * aktiverer png-filer med
                        gjennomiktig bakgrunn
    jquery-impromptu.2.5.min.js * informasjonsvindu, js
    jquery.js             * rammeverk for javascript
    jquery.uploadify.js   * opplastingsscript
    vis.js                * javascript-kode spesielt for nettstedet
uploads/
    <brukernavn>/
        <bildenavn>_<modellid>.ext * originalbilde
                                    lastet opp med
                                    modellen
    .htaccess * hindrer at script-filer kan bli kjørt i denne
                mappen
index.php * inkluderer alle filer, og setter opp template-
            klassen
controller.php * gjør alle handlinger som modifierer data,
                basert på hva som kommer fra GET-variabelen
.htaccess * hindrer visning av innholdet i mappene ovenfor

```

Vedlegg M:

Gruppeavtale



Prosjektavtale

mellom Høgskolen i Gjøvik (HiG) (utdanningsinstitusjon),
_Kristian Botnen ved Visualisere.no (oppdragsgiver), og
Jon Espen Kvisler og Aleksander Stalsberg (student(er))

Avtalen angir avtalepartenes plikter vedrørende gjennomføring av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer:

1. Studenten(e) skal gjennomføre prosjektet i perioden fra 1. Februar til _25. Mai 2009.

Studentene skal i denne perioden følge en oppsatt fremdriftsplan der HiG yter veiledning.

Oppdragsgiver yter avtalt prosjektbistand til fastsatte tider. Oppdragsgiver stiller til rådighet kunnskap og materiale som er nødvendig for å få gjennomført prosjektet. Det forutsettes at de gitte problemstillinger det arbeides med er aktuelle og på et nivå tilpasset studentenes faglige kunnskaper. Oppdragsgiver plikter på forespørsel fra HiG å gi en vurdering av prosjektet vederlagsfritt.

2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:

- Oppdragsgiver dekker selv gjennomføring av prosjektet når det gjelder f.eks. materiell, telefon/fax, reiser og nødvendig overnatting på steder langt fra HiG. Studentene dekker utgifter for trykking og ferdigstillelse av den skriftlige besvarelsen vedrørende prosjektet.
- Eiendomsretten til eventuell prototyp tilfaller den som har betalt komponenter og materiell mv. som er brukt til prototypen. Dersom det er nødvendig med større og/eller spesielle investeringer for å få gjennomført prosjektet, må det gjøres en egen avtale mellom partene om eventuell kostnadsfordeling og eiendomsrett.

3. HiG står ikke som garantist for at det oppdragsgiver har bestilt fungerer etter hensikten, ei heller at prosjektet blir fullført. Prosjektet må anses som en eksamensrelatert oppgave som blir bedømt av faglærer/veileder og sensor. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.

4. Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode, disketter, taper mv. som inngår som del av eller vedlegg til besvarelsen, gis det en kopi av til HiG, som vederlagsfritt kan benyttes til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av HiG til andre formål, og ikke overlates til utenforstående uten etter avtale med de øvrige parter i denne avtalen. Dette gjelder også firmaer hvor ansatte ved HiG og/eller studenter har interesser.

Besvarelser med karakter C eller bedre registreres og plasseres i skolens bibliotek. Det legges også ut en elektronisk prosjektbesvarelse uten vedlegg på bibliotekets del av skolens Internett-sider. Dette avhenger av at studentene skriver under på en egen avtale hvor de gir biblioteket tillatelse til at deres hovedprosjekt blir gjort tilgjengelig i papir og netttutgave (jfr. Lov om opphavsrett). Oppdragsgiver og veileder godtar slik

offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og dekan om de i løpet av prosjektet endrer syn på slik offentliggjøring.

5. Besvarelsens spesifikasjoner og resultat kan anvendes i oppdragsgivers egen virksomhet. Gjør studenten(e) i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom oppdragsgiver og student(er) bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.
6. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten(e) ikke rett til å publisere sin besvarelse, det være seg helt eller delvis eller som del i annet arbeide, uten samtykke fra oppdragsgiver. Tilsvarende samtykke må foreligge i forholdet mellom student(er) og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.
7. Studenten(e) leverer 3 - tre - eksemplarer av oppgavebesvarelsen med vedlegg til Studenttorget. I tillegg leveres et eksemplar til oppdragsgiver. HiG kan stille til disposisjon ytterligere eksemplar(er) for oppdragsgiver mot at denne godtgjør produksjonskostnadene.
8. Denne avtalen utferdiges med et eksemplar til hver av partene. På vegne av HiG er det dekan som godkjenner avtalen.
9. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og HiG som nærmere regulerer forhold vedrørende bl.a. eiendomsrett, videre bruk, konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene.

Dersom oppdragsgiver og student(er) ønsker en videre eller ny avtale, skjer dette uten HiG som partner.
10. Når HiG også opptrer som oppdragsgiver trer HiG inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.
11. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene i mellom. Dersom det ikke oppnås enighet, er partene enige om at tvisten løses av voldgift, etter bestemmelsene i tvistemålsloven av 13.8.1915 nr. 6, kapittel 32.

12. Deltakende personer ved prosjektgjennomføringen:

HiGs veileder (navn): Anne Kristin Kvite _____

Oppdragsgivers
kontaktperson (navn): Kristian Botnen _____

Student(er) (signatur): *Olav Arvid Stalsby* dato 29/1-09

Jon Erik Kvern dato 29/1-09

_____ dato _____

_____ dato _____

Oppdragsgiver (signatur): *Kristian Botnen* dato 28/01-09

Dekan (signatur): *Janette Sædd* dato 29/1-09

Revidert 11.10.07, Ivar Moe