

PROSJEKT HESSDALEN

EN 3D-REISE

Knut Martin Fjelldal, Tor Even Pettersen, Kjell-Magne Larsen og Jon-Patrick Henriksen
B019-G17 Høgskolen i Østfold



HØGSKOLEN I ØSTFOLD

Avdeling for Informasjonsteknologi

Remmen

1757 Halden

Telefon: 69 21 50 00

www.hiof.no

BACHELOROPPGAVE

Prosjektkategori: 3D design og programering	<input checked="" type="checkbox"/>	Fritt tilgjengelig
Omfang i studiepoeng: 20	<input type="checkbox"/>	Fritt tilgjengelig etter:
Fagområde: 3D, design, spillprogramering	<input type="checkbox"/>	Tilgjengelig etter avtale med arbeidsgiver

Rapporttittel: Prosjekt Hessdalen, en 3D reise	Dato: 16. Mai 2019 Antall sider: 168 Antall vedlegg: 2
Forfattere: Knut Martin Fjelldal Tor Even Pettersen Kjell-Magne Larsen Jon-Patrick Henriksen	Veileder: Gunnar Misund
Avdeling / linje: Avdeling for Informasjonsteknologi (alle)	Gruppenummer: BO19-G17

Utført i samarbeid med: HiØ/IT	Kontaktperson: Gunnar Misund
-----------------------------------	---------------------------------

Ekstrakt:

3D opplevelser er et veldig interessant hjelpemiddel til bruk for læring og utforskning. Å bruke 3D kart som et redskap i å utforske et land område er et veldig spennende prosjekt som vi er veldig interessert for.

Denne rapporten handler om et prosjekt for prosjekt Hessdalen hvor vi skal lage et 3D kart av kartdata og lage en bruker opplevelse som skal lokke gamle, unge og nye besøkende til dalen og dens kultur. I denne rapporten så vil du finne introduksjon til gruppen, analyse av oppgaven og arbeidsgiver, brukertest implementasjon og 3D modellering av kjente bygg fra Hessdalen.

3 emneord:

Hessdalen	3D	Kart
-----------	----	------

Sammendrag

På oppdrag fra Erling Petter Strand som er en høyskoleforeleser på Høgskolen i Østfold og en leder av “Project Hessdalen”, har vi fått tildelt oppgaven om å lage en visuell opplevelse av Hessdalen. Prosjekt Hessdalen ble opprettet i sommeren 1983, etter at det ble gjort observasjoner av et lysfenomen i området. Det gjøres fortsatt observasjoner i Hessdalen, men nå er antallet på ca. 20 i året. En automatisk målestasjon ble satt opp i Hessdalen i august 1998.

Oppgaven går ut på å lage et interaktivt 3D generert kart med hjelp av kartdata for å visualisere Hessdalen i sin helhet. Dette kartet skal ha strukturen basert på riktig plassering av dalen i seg selv, bygda i dalen samt hus posisjoner og vei oversikt. Hensikten med dette er å lage en visuell reise for besøkende på nettsiden til Prosjekt Hessdalen. Arbeidsgiver vil at fokuset på oppdraget skal være dalen i seg selv, og elementene som gjør dette til et interessant kulturpunkt, og ikke bare fenomenet som har satt Hessdalen på kartet.

Arbeidsgiver ville at vi skulle utføre brukertester underveis for å skreddersy opplevelsen for brukeren. Vi bestemte oss for å ta to brukertester, en for å teste kontrollene, og en for å teste kartet i sin helhet. Noen av brukertestresultatene har blitt implementert, mens andre er mulig å utforske ved videre arbeid.

Implementasjon av kartet er blitt oppnådd ved bruk av kartdata og 3D modellering av bygninger. Vi brukte Unity for å sette sammen kart og funksjonalitet, slik som bevegelse og grafikk, så brukeren får en interaktiv opplevelse. Vi brukte Maya for modellering av 3D bygninger, for å gi brukeren en opplevelse av at bygningene som ligger i dalen. Disse bygningene ble modellert i riktig skala i henhold til virkelighetsvariantene.

Takk Til

Vi vil takke alle de som har vært med under prosjekt tiden med veiledning, støtte og tilgjengelighet. Vi vil først takke Erling Petter Strand for å ha tildelt denne oppgaven, og gjort det mulig for oss å jobbe med et så spennende tema. Hans ønsker for kartet har gjort det mulig for oss å utføre ulike funksjoner gjennom en prioritert rekkefølge.

Vi vil takke veilederen vår Gunnar Misund for hans biståelse, korrigering, hjelp og anbefaling under prosjektiden. Med hans hjelp så kunne vi ta tak på prosjektet og hvordan vi skulle implementere kartet på en god måte. Dette førte til en bedre flyt gjennom prosjekt tiden.

Vi vil gjerne takke alle som har hjulpet med brukertesting av produktet fra familie og venner av prosjektgruppe medlemmene. med deres hjelp så har vi klart å teste produktet som vi har produsert. Uten dere, så ville vi ikke vite hva som måtte videre utvikles, og hvordan vi kunne løse ikke fult så åpenbare feil.

Innhold

Sammendrag.....	3
Takk Til.....	4
Innhold	5
1 Introduksjon.....	10
1.1. Prosjektgruppen.....	10
1.2. Oppdragsgiver	11
1.3. Oppdraget	12
1.4. Hvorfor, hva og hvordan: Formål, leveranser og metode.....	12
Hovedmål:	12
Leveranse:	12
Metode:	13
1.5. Rapportstruktur.....	13
1.6. Risikoanalyse.....	14
1.6.1. Risiko 1.....	14
1.6.2. Risiko 2.....	14
1.6.3. Risiko 3.....	15
1.6.4. Risiko 4.....	15
1.6.5. Risiko 5.....	16
2. Analyse.....	17
2.1. Prosjekt Hessdalen.....	17
Project Hessdalen	17
Lysfenomenet	17
2.2. Oppgaven.....	17

2.3.	Lignende prosjekter	18
	Google Earth	18
	Google Maps	19
	NASA World Wind.....	20
	Earth 3D map	21
	3D kommunekart.....	21
	Meteoearth.....	22
	Electric oyster.....	22
	Mapme.....	22
2.4.	Om Kartdata	23
	2.4.1. Overflatemodell.....	23
	2.4.2. Tekstur.....	24
	2.4.3. Heightmap	25
	2.4.4. Koordinater.....	26
2.5.	Alternativer for å hente inn data	27
	2.5.1. Høydedata.....	27
	2.5.2. Unity World Composer	28
	2.5.3. 3D Map Generator.....	29
	2.5.4. NASA Earth Data.....	29
	2.5.5. Terrain Party.....	29
2.6.	Innhente kartdata	29
	2.6.1. Terreng	29
	2.6.2. Tekstur.....	30
2.7.	Behandling av kartdata.....	30
	GeoTiff.....	34
	Photoshop RAW.....	34

2.8.	Valg av programvare	34
2.9.	Innhenting av informasjon.....	38
2.9.1.	Viktige Bygninger	39
2.10.	BuildR 2	39
3.	Begreper og definisjoner	42
3.1.	3D Grafikk.....	42
3.1.1.	Modellering	42
3.1.2.	Polygon mesh	43
	Edge.....	43
	Vertex.....	43
	Face	43
3.1.3.	Teksturer og materialer.....	44
3.1.4.	Rendering	44
3.1.5.	Low and high polygons	45
3.2.	Filformater.....	45
3.2.1.	GeoTIFF/TIFF.....	45
3.2.2.	RAW.....	46
3.2.3.	PNG	46
3.3.	Web	46
3.3.1.	HTML5.....	46
3.4.	GDAL.....	47
3.4.1.	GDAL_translate/warp	47
3.4.2.	Ot.....	47
3.4.3.	Of.....	47
3.4.4.	Scale	47
4.	Design.....	47

4.1 Metode 1 – 3D Modellering ut i fra bildereferanser.....	47
4.2 Metode 2 – 3D Modellering uten å ha nøyaktige mål av bygget	49
4.2.1 Finn mål av design objekt	49
4.2.2 Finne bilder av design objekt	52
4.2.3 Utregning.....	53
5. Implementasjon	54
5.1. Brukerkontroller	54
5.2. Navigasjon.....	56
5.2.1. Minikart	56
5.2.2. Kompass	57
5.2.3. Stedsnavn.....	58
5.2.4. Teleportering	60
5.3. Nedtrekksmeny.....	61
5.4. Overflate.....	61
5.4.1. Klippe terreng.....	61
5.4.2. Importere/eksportere.....	62
Forsøk 1:.....	63
Forsøk 2.....	64
Forsøk 3.....	65
Forsøk 4.....	67
5.4.3. Terrain Stitches	68
5.4.4. Område	68
5.5. Skybox.....	69
5.6. Importere 3D modeller	70
6. Brukertesten	70
6.1. Problemstilling	70

6.2.	Metode.....	71
6.3.	Brukerprofil.....	71
6.4.	Miljø og krav.....	71
6.5.	Brukertest 1.....	72
6.6.	Brukertest 2.....	73
7.	Diskusjon.....	75
7.1.	Oppgave mål.....	75
7.2.	Levering.....	76
7.3.	Tekniske utfordringer.....	76
7.4.	Videre utvikling.....	77
	Teleport-klikk funksjon.....	77
	Instillings meny.....	77
	Begynner tutorial/guide.....	78
	Større og mer levende verden.....	78
	Lysfenomen.....	78
	Mer informasjon.....	78
7.5.	Feil og mangler.....	79
8.	Konklusjon.....	79
9.	Referanser.....	Error! Bookmark not defined.
10.	Vedlegg.....	80
10.1.	3D modellering.....	80
10.1.1.	Hessdalen UFO camp.....	80
10.1.2	Røde kors hytten.....	83
10.1.3	Bekken Gård.....	98
10.1.4	Samfunnshuset.....	108
10.1.5	Publ.....	110

10.1.6 Blue Box.....	124
10.2. Brukertest resultater.....	163
11. Bibliotekliste	174

1 Introduksjon

1.1. Prosjektgruppen

Prosjektgruppens medlemmer er: Knut Martin Fjelldal, Kjell-Magne Larsen, Jon-Patrick Henriksen and Tor Even Pettersen. Knut Martin og Kjell-Magne har jobbet på prosjekter sammen fra før, mens Jon-Patrik og Tor Even har jobbet utenom gruppen i tidligere situasjoner. Gruppen spesialiserer seg i Medieproduksjon, 3D modellering og webutvikling. Alle i gruppen har erfaring innenfor flere programvarer, og prosjektarbeid fra før.

Knut Martin Fjelldal:

Studerer: Digitale Medier

Relevant fagkompetanse: 3D modellering, Grafisk design, Spill design, Webutvikling, Spillutvikling.

Interesser: Data, musikk, film, Dataspill.

Knut Martin er 24 år og kommer fra Sarpsborg i Østfold fylke hvor han er bosatt. Ved siden av studiet så jobber han på en Deli de Luca deltidsjobb. Han brenner for spill og utviklingsprosessen av dette, og holder seg regelmessig oppdatert fra bransjen.

Kjell-Magne Larsen:

Studerer: Digitale Medier

Relevant fagkompetanse: 3D-modellering, grafisk design, spilldesign, webutvikling

Interesser: Data, film, fotball, innebandy

Kjell-Magne er 21 år og bosatt i Fredrikstad. Ved siden av studiene jobber han som Taxisjåfør og webutvikler. Han har flere aktive verv som LAN crew og koordinator for festivaler. På CVn har han også flere kortfilmer som lyd, og produksjonsassistent.

Jon-Patrick Henriksen:

Studerer: Digitale Medier

Relevant fagkompetanse: 3D-modellering, koding, design, spilldesign, spillutvikling

Interesser: Data, fly-spotting, lyd og lys

Jon-Patrick er 25 år og bor i Rygge. Han jobber som freelance rigger og stagehand og sikkerhetsvakt. Han trives også med frivillig arbeid, gjerne som crew på flere LAN og festivaler.

Tor Even Pettersen:

Studerer: Digitale Medier

Relevant fagkompetanse: 3D-modellering, spilldesign, spillutvikling, videoproduksjon, grafisk design

Interesser: Dataspill, fotball, musikk, TV-serier og film

Tor Even er 24 år og kommer fra Langhus i Akershus fylke, men bor og studerer i nå i Halden. Før studiet jobbet han på Meny Langhus i tre år.

1.2. Oppdragsgiver

Oppdragsgiveren til dette prosjektet er Erling Petter Strand som er en høgskoleforeleser på Høgskolen i Østfold og en leder av "Project Hessdalen", et forskningsprosjekt om Hessdalen fra 1981. Han har en sivilingeniør utdanning fra NTNU fra universitetet i Trondheim tilbake i slutten av 70 tallet. Erling har kunnskaper og erfaring fra forskningsavdelingen til Standard Telefon og kabelfabrikk på Økern Oslo, EDAS målesystemer som lå i eidsvoll, og EDB og automatisering avdelingen på østfold ingeniørhøgskole i sarpsborg.

1.3. Oppdraget

Oppdraget i dette prosjektet er å lage et interaktivt kart med hjelp av kartdata for å gi besøkende, og personer som er interessert i å besøke dalen, en bedre forståelse av hvordan den ser ut og er utformet. Dette oppdraget har en hensikt med å tiltrekke flere besøkende og inspirere de som ikke kan besøke dalen. Arbeidsgiver vil at fokuset på oppdraget skal være dalen i seg selv, og elementene som gjør dette til et interessant kulturpunkt, og ikke bare fenomenet som har satt Hessdalen på kartet. Mens dette fenomenet er viktig når dalen blir nevnt, så er det mye mer til dalen enn fenomenet, og dette vil være en liten del av oppdraget.

1.4. Hvorfor, hva og hvordan: Formål, leveranser og metode

Hovedmål:

Lage et interaktivt 3D generert kart med hjelp av kartdata for å visualisere Hessdalen i sin helhet. Dette kartet skal ha strukturen basert på riktig plassering av dalen i seg selv, bygda i dalen samt hus posisjoner og vei oversikt. Hensikten med dette er å lage en visuell reise for besøkende på nettsiden til Prosjekt Hessdalen.

Delmål 1:

Lage en skalert korrekt miniatyr versjon av Hessdalen.

Delmål 2:

Lage en visuell opplevelse av hvordan Hessdalen ser ut, som kan oppleves ved å styre et kamera gjennom området.

Delmål 3:

Lage informasjonsbokser som brukeren kan velge å se i full størrelse når man klikker på ikoner som er tilgjengelige på siden, som representerer spesifikke steder.

Leveranse:

Leveransen gjennom prosjektet vil være hovedsakelig hoveddokumentet med produktet under fullføring av prosjektet, men vi vil også ha del leveranse i forhold til studiets veiledning. Dette vil si at vi vil ha tre versjoner av hoveddokumentet som skal bli levert. Første versjon skal bli levert innen 8 mars, andre versjon skal bli levert innen 23 april, og tredje, og det fullførte

dokumentet med vedlegg skal bli levert innen 16 mai. Prosjektet vil også ha en plakatlevering som vil bli ferdig og hengt opp på skolen innen 27 mai. Prosjektet vil også ha leveranser for arbeidsgiveren som vil hjelpe han med å implementere produktet til bruk. En kort oppsummering av de forskjellige leveransepunktene.

- Hjemmeside for gruppen
- Forprosjektrapport
- Hoveddokument v1
- Hoveddokument v2
- Refleksjonsnotat
- Hoveddokument
- Prosjektplakat
- Brukertest av produktet
- Produktet(3D modell med funksjoner)
- Nettsideløsning for å implementere produktet.

Metode:

For å oppnå dette vil gruppa gjennomføre en grundig gjennomgang av tidligere prosjekter og lignende produkter. Samtidig vil det holdes møter med arbeidsgiver for å kartlegge hva som ønskes i produktet. Ved å lage en prototype kan vi få den brukertestet før prosjektet leveres. Ved å 3D-modellere i Maya så har vi mulighet til å gjenskape miljø og bygninger i Hessdalen slik at mennesker som ikke har vært i Hessdalen før kan gjøre seg kjent i området. I tillegg så benytter vi kartdata fra Det Norske Kartverket som gjør at dalen blir gjenskapt på en troverdig måte.

1.5. Rapportstruktur

I kapittel 2 foretar vi en bakgrunns analyse som starter med en fyldigere beskrivelse av selve oppgaven. Videre ser vi på grafikk, og dens historie og verktøy for å behandle det. Deretter analyserer vi Statens Kartverk, og går dypere inn i Norge Digitalt. Så ser vi på innhenting av teksturer og terreng via kartdata, før vi går over til behandling av kartdata. Vi går også gjennom valg av programvare, samt hvorfor vi har valgt det vi bruker. Deretter tar vi for oss innhenting av informasjon om Hessdalen, med en oversikt over kjente bygninger. Kapittelet ender med en analyse av lignende prosjekter som finnes fritt tilgjengelig på internett.

Videre, i kapittel 3 forklarer vi begreper og definisjoner. I kapittel 4 presenterer vi en overordnet design av hvordan vi har tenkt til å forme løsningen. Deretter tar vi for oss hvordan vi

produserte resultatene av prosjektet og beskriver selve produktet i kapittel 5. I kapittel 6 ser vi på brukertester, før vi diskuterer resultatet av prosjektet. Rapporten avsluttes med en konklusjon i kapittel 8.

1.6. Risikoanalyse

1.6.1. Risiko 1

Prosjekt Hessdalen, risikoanalyse			
Risk id: 1	Dato: 21.01.2019	Sannsynlighet: 30%	Innvirkning: Høy
Beskrivelse: Manglende deltagelse under arbeid på prosjektet Langvarig sykdom som påvirker muligheten til å delta på møter og arbeid			
Skadebegrensning og/eller overvåkning: Skadebegrensning: Kjapt overføre arbeid til medlemmer som ikke er blitt tilsidesatt av latskap eller sykdom Overvåkning: Passe på at medlemmer gjør jobben sin til oppsatt tid			
Ledelse / Beredskapsplan / Utløser Ledelse: Beredskapsplan: Utløser:			
Opphavsmann:		Tildelt:	

1.6.2. Risiko 2

Prosjekt Hessdalen, risikoanalyse			
Risk id: 2	Dato: 27.01.2019	Sannsynlighet: 30%	Innvirkning: Veldig høy
Beskrivelse:			

Ved ingen tilgang til resurser			
Ved mangelfull tilgang til resurser			
Skadebegrensning og/eller overvåkning:			
Påse når tilgang til resurser blir åpnet, at tilgangen fungerer			
Sjekk ved jevne mellomrom at tilganger fortsatt er tilgjengelig			
Ledelse / Beredskapsplan / Utløser Ledelse:			
Ledelse: Ved utløst krise, opprett forbindelse med ressurseier via mail eller møte snarest			
Beredskapsplan: Prøv å få resurser så for som mulig og ha offline back-up			
Utløser: Når det blir oppdaget at tilgang har blitt tilbake eller ikke blitt gitt fra før			
Opphavsmann:		Tildelt:	

1.6.3. Risiko 3

Prosjekt Hessdalen, risikoanalyse			
Risk id:	Dato:	Sannsynlighet:	Innvirkning:
3	21.01.2019	25%	Middels
Beskrivelse:			
Ikke gjennomført brukertester			
Ikke tilfredsstillende sluttresultat			
Skadebegrensning og/eller overvåkning:			
Påse at brukertest blir gjennomført			
Ved ikke tilfredsstillende sluttresultat, gjør testen om igjen			
Ledelse / Beredskapsplan / Utløser Ledelse:			
Planlegge brukertester, slik at det er god til å gjennomføre test og gjennomgå resultat			
Opphavsmann:		Tildelt:	

1.6.4. Risiko 4

Prosjekt Hessdalen, risikoanalyse			
Risk id:	Dato:	Sannsynlighet:	Innvirkning:
4	21.01.2019	10%	Veldig Høy
Beskrivelse:			

<p>Hardware kan regnes som enkeltkomponenter i en pv eller som hele pcen.</p> <p>Ved tap av data som følge av:</p> <p>Software krasjer før du får lagret</p> <p>Hardware som krasjer</p> <p>Hardware som blir korrumpert</p> <p>Hardware som blir ødelagt</p> <p>Hardware som blir tapt som følge av tyveri</p>			
<p>Skadebegrensning og/eller overvåkning:</p> <p>Påse at samtlige jobber i Google disk, hvor det en skriver automatisk blir lagret</p> <p>Påse at de som jobber offline eller i et annet program tar å lagrer hypping og tar sikkerhetskopier.</p> <p>Påse at sikkerhetskopi-løsninger ikke bare er en annen plass på samme pc, men gjerne i en skyløsning eller lignende</p>			
<p>Ledelse / Beredskapsplan / Utløser Ledelse:</p> <p>Tap av hardware eller korrumpert hardware: få en lånemaskin og innhent sikkerhetskopi og fortsett arbeid</p> <p>Ved manglende sikkerhetskopi: Forhør seg med de andre på gruppa om de har sikkerhetskopier.</p> <p>Hvis ingen sikkerhetskopi er tilgjengelig, påbegynn arbeid fra starten av</p>			
Opphavsmann:		Tildelt:	

1.6.5. Risiko 5

Prosjekt Hessdalen, risikoanalyse			
Risk id:	Dato:	Sannsynlighet:	Innvirkning:
5	21.01.2019	20%	Veldig Høy
<p>Beskrivelse:</p> <p>Ved en hendelse hvor et eller flere medlem av gruppa mister internett-tilkoblingen hjemme</p>			
<p>Skadebegrensning og/eller overvåkning:</p> <p>Prøv å feilsøk problemer og kartlegg hvor problemet ligger.</p> <p>Hvis der er feil hos medlemmet, prøv å restarte routeren og kjør en DNS-flush.</p> <p>Hvis det er hos ISP, Internet service provider, kontakt de og søk assistanse der</p>			
<p>Ledelse / Beredskapsplan / Utløser Ledelse:</p> <p>Benytt 3G/4G/5G via telefon</p>			
Opphavsmann:		Tildelt:	

2. Analyse

Dette kapittelet består av en bakgrunnsanalyse av hvordan vi skal samle relevant informasjon til prosjektet. Denne delen inneholder også hvordan vi kommer fram til valg av teknologi og programvare, samt en fyldigere beskrivelse av oppgaven.

2.1. Prosjekt Hessdalen

Project Hessdalen

Project Hessdalen ble opprettet i sommeren 1983. Den første feltaksjonen ble gjennomført fra 21. januar til 26. februar i 1984. Under denne felt aksjonen ble det gjort 53 visuelle observasjoner, og flere instrument avlesninger. Det gjøres fortsatt observasjoner i Hessdalen, men nå er antallet på ca. 20 i året. En automatisk målestasjon ble satt opp i Hessdalen i august 1998.

Lysfenomenet

I desember 1981 ble det først oppdaget ukjente lysfenomener i Hessdalen. Disse lysfenomenene ble ofte beskrevet som store lysende kuler som beveget seg rundt i dalen. Fram til slutten av 1984 ble flere hundre observasjoner gjort i dalen. Under den tiden hvor det var mest aktivitet der, ble det registrert opp til 20 observasjoner i uka. Ettersom fenomenet ble mer kjent, kom det mange fra andre steder som reiste til Hessdalen for å se fenomenet selv. Dette fenomenet har blitt kalt "Hessdalsfenomenet" av prosjekt hessdalen. bakgrunnen rundt dette fenomenet er fremdeles ukjent, men det finnes noen konkrete teorier om hva som kan være årsaken. Mulighetene rundt dette fenomenet er håpefullt stort, blant annet innenfor energi virksomheten på grunn av hvor mye energi disse fenomenene har.

2.2. Oppgaven

Oppgaven omhandler å lage et verktøy for Prosjekt Hessdalen med hensikt om å skape mer interesse om dalen og dens kultur. Verktøyet som vi skal produsere for prosjektet er et 3D kart, tilgjengelig på nettsiden deres hvor besøkende og interesserte kan ta en titt inn i hvordan dalen er, viktige lokasjoner og interessepunkter. Hensikten med dette er å skape mer interesse for besøkende som planlegger å ta reisen og vil forberede seg på turen, eller for reisende som er villige men ikke har muligheten eller under andre omstendigheter ikke kan besøke dalen.

Kartet vi skal lage for prosjektet er et 3D kart laget med hjelp av kartdata. På denne måten lager vi en skalert versjon av dalen med så mye detaljer som er tilgjengelig. Med hjelp av kartdata og teksturdata, vil vi med 3D programmet maya, og programmotoren Unity, utvikle et interaktivt kart som har funksjoner som et lite “mini” spill. Med dette så mener vi at funksjonaliteten på programmet vil inkludere en kontrollmekanisme som gjør at brukeren selv kan bevege seg over 3D området, og vil ha muligheter til å utforske de forskjellige delene av kartet, og også lese om interessepunkter fra Hessdalen som vil være involvert. Kartet vil inneholde informasjon om Dalen, lokalbefolkningen og viktige plasser som f.eks bygninger. Kartet vil også inneholde historie om landsbyen i dalen, og relevant informasjon om lysfenomenet som har satt dalen på kartet, samtidig som dette ikke er et fokus.

For å kunne lage et kart med relativt mye informasjon, så vil vi samle informasjon fra forskjellige kilder angående historiene rundt dalen, kjente monumenter, bygninger og områder, og lage en enkel, men godt fyldig måte for besøkende å lese denne informasjonen på. Med dette så vil vi lage en funksjon for brukeren som lar dem lese om forskjellige lokasjoner på kartet med hjelp av klikkbare ikoner. Når brukeren klikker på disse ikonene vil de få muligheten til å lese fra en tekstboks, om informasjonen.

2.3. Lignende prosjekter

I denne delen gjøres det undersøkelse på lignende plattformer og applikasjoner som ligger fritt tilgjengelig på internett. Sammenligningen tar hovedsakelig for seg hensikten med plattformen, hvorvidt det er i 3D og brukervennligheten.

Google Earth

Google Earth(tidligere Keyhole Earth Viewer) er et interaktivt 3D globus program fra selskapet Google lansert i 2004. Programmet er tilgjengelig på desktop, nettbrett og smarttelefoner, men kan også kjøres rett i nettleseren(Kun Google Chrome). Eksportering av kartdata er også fritt tilgjengelig i desktop versjonen. Google bruker satellittbilder som er satt sammen for å skape større bilder, bildene i seg selv varierer i kvalitet, men er stadig under oppdateringer. I motsetning til mange andre kartverk kan brukere zoome inn på høyoppløselige bilder i 3D av større befolkede områder og landemerker. Plattformen bruker kun datamusen til å bevege seg rundt kartet, som gjør det hurtig og effektivt å navigere. Det finnes også

1 https://en.wikipedia.org/wiki/Google_Earth

tilleggsfunksjoner som lar brukere endre syns perspektiv, måle avstander, og navigere seg rundt i gater ved hjelp av tilleggstjenesten Google Street View. Google Earth har sin hensikt for å utforske verden². Figur 1 viser datamuskontroller for Google Earth.



Figur 1: Datamus Kontroller for Google Earth

Google Maps

Google Maps(lansert 2005) er en annen karttjeneste levert av selskapet Google, med mange av de samme funksjonene som Google Earth, se Figur 2 for mer informasjon. På samme måte som Earth benytter Maps seg av satellittbilder satt sammen for å skape illusjon av et større bilde³. Maps samler informasjon om bedrifter, som brukere kan dra nytte av når de bruker plattformen. Bilder, anmeldelser, historie og priser er noen av dataene brukere kan finne om bedrifter. Maps tilbyr flere typer underlag som satellitt og terreng for å dekke ulike behov, i tillegg har plattformen ekstra trafikk funksjoner som lar brukere finne kollektivtransport, følge varsler om kø og sykkelruter. Maps sin hensikt er mest for navigasjon og avstand, brukere kan også laste ned kart for bruk uten internett.

² <https://www.google.com/intl/no/earth/>

³ <https://www.google.com/maps>



Figur 2: Datamus Kontroller for Google Maps

NASA World Wind

Et open-source virtuell globus utviklet av NASA i 2003 og ble lansert 6 måneder før Google Earth⁴. En nettleser versjon ble lansert i 2015 og er fortsatt i en demo fase. Brukeren kan fritt utforske jorda ved å navigere seg rundt, zoome inn på ulike destinasjoner og velge ulike bakgrunnslag. Nettleserversjonen er ikke like stabil som Google sine karttjenester og har en lang nedlastingstid, mens nedlastbar utgave(Java) ikke er bygd sammen til en applikasjon og er mest brukt av myndigheter og kommersielle utviklere rundt om i verden. Figur 3 viser datamus kontroller for NASA World Wind.



Figur 3: Datamus Kontroller for NASA World Wind

⁴ https://en.wikipedia.org/wiki/NASA_WorldWind

Earth 3D map

En nettside som bruker Google Maps sin tjeneste. På langt hold virker det som et 2D kart, men på nært hold viser kartet oversiktlige 3D bilder av byer og større befolkede områder. I likhet med Maps er det fortsatt mange områder som kun vises i 2D. Nettsiden har en meny som lar brukeren velge ulike mål å utforske.

3D kommunekart

Et 3D kart som dekker hele Norge, tjenesten er levert av Norkart og har som hensikt å samle alle norske kommuner og la kommunene selv være med å tilpasse informasjon for sine innbyggere. Tjenesten lar brukere velge tider på døgnet og legge på skygger for å se hvordan det påvirker lyset. Det er også mulig å søke på kommune eller spesifisert adresse for å navigere. Manøvrering gjøres kun med datamusen ved enten å velge kommune, eller å dra rundt på kartet. Det virker på en litt annen måte enn Google, se Figur 4. Terreng og overflaten på bygninger er tydelig gjenskapt i 3D, men terrenget har kun 2D teksturer og bygninger har kun en standard farge. Kartet viser stort sett alle bygninger i 3D og det er mulig å navigere mer detaljert i byer enn i Google sine tjenester⁵.



Figur 4: Datamus Kontroller for 3D kommunekart

⁵ <https://www.norkart.no/2016/03/09/ny-3d-kartlosning/>

Meteoearth

Et interaktivt 2D kart levert av MeteoGroup som lar brukere utforske det globale været i sanntid. Tjenesten har en type “timelapse” som viser været for hver dag. Ved å bruke en sidemeny med tydelige ikoner kan brukere selv skru av og på ulike kriterier som regn, skyer og vind i timelapsen. Kartet viser kun terrenget med veinett, men har et par tilleggsfunksjoner som 3D og lys for å vise hvor det er tettbebygde områder. 3D funksjonen strekker bare kartet lenger ut, men gir ikke noen følelse av 3D.

Electric oyster

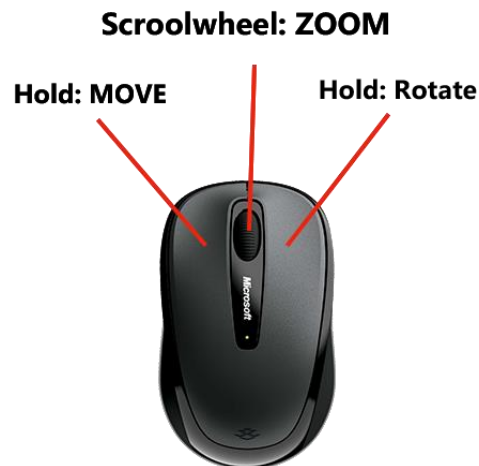
Oppgaveeksempel og inspirasjon til prosjektet. En interaktiv reise som tar brukeren gjennom et terreng av fjell med tekstur av snø. Brukeren har lite kontroll på navigasjon, men styrer retning med datamusen. Kamera tilter i den retningen som styres, som gir brukeren en følelse av å “fly” gjennom fjellkjeden. Programmet har en lyssetting som reflekterer skygge over fjellene og gir en slags dybde i terrenget. Se Figur 5 for datamus kontrollen til Electric Oyster.



Figur 5: Datamus Kontroller for Electric Oyster

Mapme

Mapme er en tjeneste som gir brukere muligheten til å lage egne interaktive 3D kart basert på ulike behov. Bygninger og landemerker kan utforskes i 3D, men selve terrenget er i 2D. Tjenesten er betalt og summen avhenger av funksjonene brukeren vil ha. Mapme har en eksempelside med ulike typer kart, en av de er en side over kjente katedraler i Europa. Grensesnittet viser informasjonbokser med bilder, tekst og videoer over katedralene. På kartet er generelle bygninger stort sett bare kubeformet i en grå tekstur, mens katedralene skiller seg ut med detaljerte polygoner. Brukere kan zoome nærmere og få et inntrykk av hvordan de ser ut. Mapme beskriver på sin nettside de ulike hensiktene og brukermassene kartet kan ha, for eksempel utdanning, reise og kultur. Se Figur 6 for datamus kontrollen til Mapme.



Figur 6: Datamus Kontrolller for MapMe

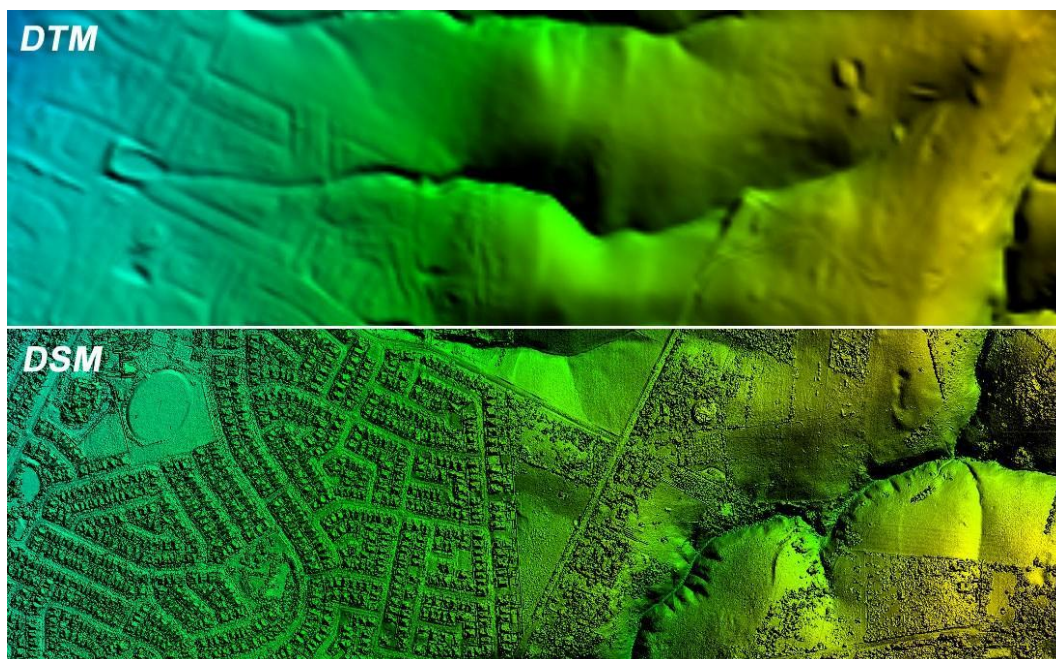
2.4. Om Kartdata

2.4.1. Overflatemodell

En overflatemodell er 3D representasjon av et terrengs overflate. Overflate modeller brukes ofte i geografiske informasjonssystemer og er det vanligste grunnlaget for digitalt produserte kart. Det finnes flere typer modeller, de mest populære er DEM(Digital Elevation Model), DTM(Digital Terrain Model) og DSM(Digital Surface Model), men DEM er mye brukt som en generisk betegnelse for både DTM og DSM.

DTM er en digital terrengmodell som inneholder høydeforskjeller av naturlige terrengtrekk, som åser, elver og daler. ForhøyeIser av vegetasjon og kulturelle egenskaper, som bygninger og andre objekter er ikke inkludert. En DSM fanger opp både de naturlige og

6 <https://gisgeography.com/dem-dsm-dtm-differences/>

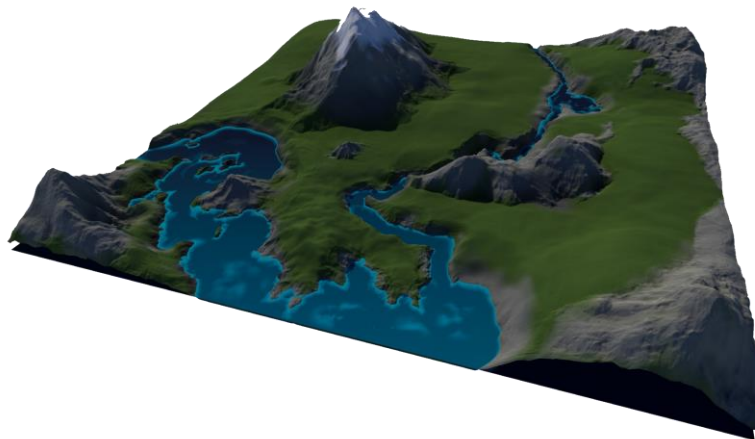


Figur 7: forskjellen på en DTM og DSM modell konstruerte egenskapene på jordens overflate. Figur 7 viser forskjellen på en DTM og DSM modell.

2.4.2. Tekstur

En tekstur kan refereres til det menneske observerer som stimuli av en overflate sin farge, orientering og intensitet i et objekt⁷. I 3D sammenheng er “texture mapping” en metode for å definere høyfrekvent detalj, overflate tekstur eller fargeinformasjon på et 3D-objekt. I kart sammenheng er teksturer flyfoto(ortofoto) tatt av landskapet. Et ortofoto har samme geometriske egenskaper som et kart og kan knyttes til et referansesystem, og kan koples sammen med en

⁷<https://www.cs.auckland.ac.nz/~georgy/research/texture/thesis-html/node5.html#fig:TextureExampleArtificial>



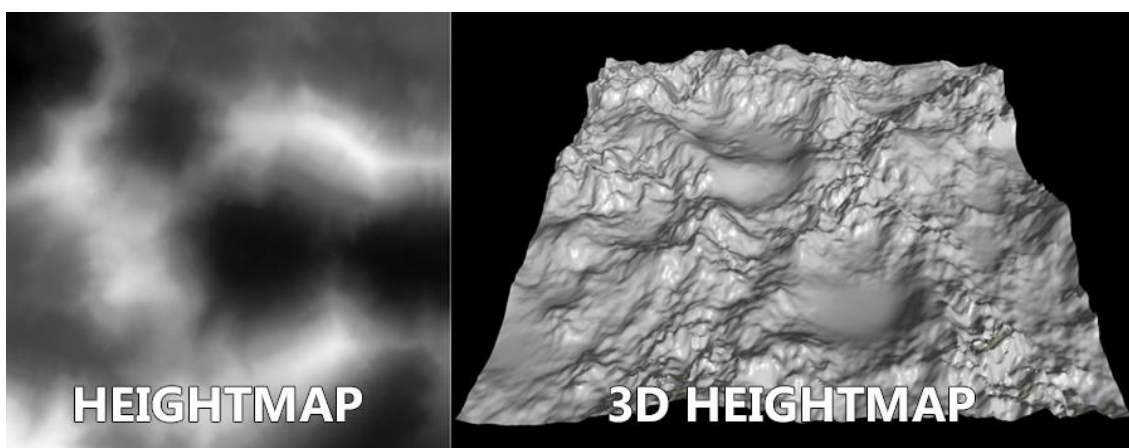
Figur 8: Terreng med tekstur på overflaten
terrengmodell med samme utgangspunkt i et 3D program. Figur 8 viser en terrengmodell med
tekstur lagt over⁸.

2.4.3. Heightmap

Heightmap er et rasterbilde som ofte er brukt i høyde modellering. Et høydekart bruker en gråtone fargekanal for å beskrive vertikal distanse på et terreng, hvor hvit definerer “høyt” og svart “lavt”⁹. Figur 9 viser forskjellen på et høydekart og hvordan det vil se ut hvis et 3D program renderer bilde basert fargen.

⁸ <https://kartverket.no/geodataarbeid/Flyfoto/Ortofoto/>

⁹ <https://en.wikipedia.org/wiki/Heightmap>



Figur 9: Heightmap før og etter det er konvertert til en 3D modell

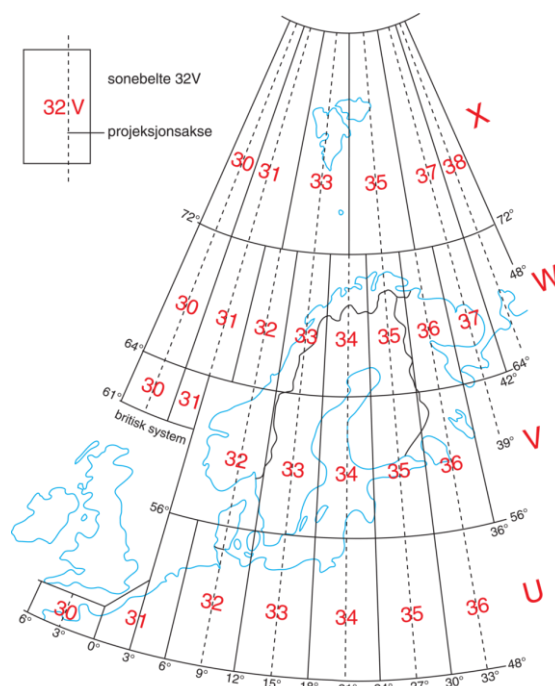
2.4.4. Koordinater

Geodetisk Datum er en referanseramme som ligger til grunn for all geografisk informasjon som kart, posisjonering, jordobservasjon og måling av klimaendringer¹⁰. I Norge bruker vi referanserammene EUREF89, ITRF, WGS84 og NN2000. EUREF89 ble etablert for å erstatte NGO1948 og ED5011, som var tidligere standarder i Norge. Koordinatene blir uttrykt som kartplankoordinater, N(nord) og E(øst), i en projeksjon som kalles UTM(Universal Transversal Mercator)¹². Norges hovedkartserie i målestokk 1:500000, N50, bruker EUREF89 som referanseramme. UTM projeksjonen er i bruk over hele Europa i ulike nummererte soner. Sonene er i stigende rekkefølge mot øst, se Figur 10 for en oversikt av soneinndelingen. I Norge brukes UTM 32, 33 og 35.

¹⁰ <https://snl.no/EUREF89>

¹¹ <https://www.kartverket.no/Posisjonstjenester/bruke-referanserammer/Referanserammer-for-Noreg/>

¹² <https://snl.no/UTM>



Figur 10: Kart over UTM soner

2.5. Alternativer for å hente inn data

Det finnes flere gratis og betalte aktører som tilbyr eksportering av kartunderlag. I denne seksjonen tar vi utgangspunkt i de mest aktuelle tjenestene, både betalte og gratis, men som ikke koster for mye. I likhet med Kartverket sine tjenester i Norge, har Finland og Nederland egne aktører som tilbyr eksport av kartunderlag, men kun av sitt respektive land.

2.5.1. Høydedata

Høydedata er en tjeneste levert av Kartverket(Statens Kartverk) og tilbyr et presist kartunderlag, men kun over Norge. Filer kan eksporteres i både GeoTiff og DEM med ulike kompresjoner, i tillegg til en oppløsning ned til 1x1 meter. Kartunderlag kan eksporteres i både DTM og DOM(DSM)¹³, se 2.4.1 for beskrivelse. Høydedata tilbyr ikke eksport av tekstur, men det kan hentes ut av Norgebilder, en tjeneste Kartverket samarbeider med.

13 <https://hoydedata.no/LaserInnsyn/>

2.5.2. Unity World Composer

WorldComposer er en Unity plugin som lar en hente ut terrengmodeller og teksturer og importere dette rett i Unity. Helt enkelt minner designet om Google Maps, med like kontrollere for navigere rundt på kartet. WorldComposer dokumenterer på sin nettside at det er enkelt å tegne et utsnitt og importere til Unity. Man får også flere valg for å tilpasse dataene og i den siste oppdateringen skal det være mulig å importere terreng i 10x10 meter per pixel¹⁴. Dette er ikke testet da pluginen koster 25\$ for å laste ned fra Unity asset store, det er heller ikke dokumentert hvor WorldComposer henter sine høydedata fra. Pluginen har stort sett fått gode vurderinger av brukere på asset store og sitt eget forum, men etter at Unity rullet ut sin 2018.3 oppdatering desember 2018 har WorldComposer bare fått dårlig kritikk. For de fleste fungerer ikke pluginen i det hele tatt og utviklerne svarer ikke på forespørsel.

I begynnelsen av mars 2019, så ble utgiver av WorldComposer aktiv igjen og begynte å svare på forum. Han sa da at det ville komme en oppdatering som ville fikse problemet mange hadde. Vi bestemte oss for å kjøpe WorldComposer for å teste dette. Til å begynne med så fungerte programmet. Det ble åpnet i Unity for så krasje uten å gi en feilkode. Etter denne krasjen så har ikke WorldComposer ville virke som normalt og samme problem oppleves som mange andre på forumet til WorldComposer. Noen dager etter at gruppen hadde kjøpt programmet og opplevde problemene, så kom det en ny oppdatering til WorldComposer, men dette fikset ikke problemene heller.

Gruppen har gått så langt å avinstallere Unity og WorldComposer, og fjerne alle spor på dataen, for også gjeninstallere alt sammen. Men samme historikk. Det starter og vi får ut et terreng kart. Legger dette inn i Unity. Også krasjer Unity uten noen feilmelding og får ikke fortsette å arbeidet. Det er ukjent om problemet ligger i Unity eller i WorldComposer, men det kan virke som Unity blir “ødelagt” etter å ha lagt inn WorldComposer.

På bakgrunn av dette så vil ikke prosjektgruppen anbefale å benytte WorldComposer. Programmet TerrainComposer ble også underøkt, laget av samme utvikler. Det oppfattes at WorldComposer benyttes når en ønsker et kart fra en virkelig lokasjon og TerrainComposer benyttes når en ønsker et terreng som er programmert på forhånd og vil gi et kart basert på fiksjon.

¹⁴ <http://www.terraincomposer.com/worldcomposer/>

2.5.3. 3D Map Generator

Dette er en betalt tjeneste som tilbyr eksport av kartunderlag som en plugin til Adobe Photoshop. Tjenesten koster 21\$ og har generelt fått gode tilbakemeldinger (Envatomarket, 2019). Det er en oversiktig og forklarende dokumentasjon på deres egen nettside, med video tutorials og eksempler på hvordan lage egne kart. Map Generator bruker Google maps for å hente ut høydedata og satellittbilder som brukes til tekstur, hvor dette settes sammen i Photoshop og rendres til et 3D kart. Map Generator har også et eget bibliotek med presets for teksturer, dersom man ikke vil bruke satellittbilder. Baksiden er at det ikke følger med noen metadata, og det er ikke dokumentert noen form for oppløsning eller koordinatsystem.

2.5.4. NASA Earth Data

Denne tjenesten fungerer som et 2D kart der brukeren kan navigere seg rundt og markere opp ønsket felt. Earth Data tilbyr terrengmodell over hele kloden, men Det krever en profil hos NASA sine systemer for å kunne laste ned¹⁵. Filer kommer i GeoTIFF format og har en varierende oppløsning der beste er 30x30 meter.

2.5.5. Terrain Party

Terrain Party er en åpen tjeneste basert på OpenStreetMap, en tjeneste med en samling av kartdata for hele kloden satt sammen av et åpent samfunn av utviklere. Terrain Party er gratis å bruke og . Data kommer i gråtone PNG format med et varierende underlag fra 10x10 meter til og med 900x900 meter, men i Norge er beste oppløsning 30x30 meter.

2.6. Innhente kartdata

2.6.1. Terreng

Kartunderlag hentes ut fra Høydedata.no, som er en tjeneste levert av Kartverket. Tjenesten leverer et presist terreng over Norge, som fritt kan lastes ned og behandles på ulike måter. I spillsammenheng er kartverket sine data benyttet i blant annet Battlefield 5, som er satt i

¹⁵ <https://earthdata.nasa.gov/>

2.verdenskrig, Kartverket nevner dette på sine egne sider¹⁶. I spillet er terreng over Larvik benyttet, basert på virkelige hendelser fra krigen.

Til prosjektet benyttes DTM-1(1x1 meter) som kartunderlag for å oppnå høy kvalitet på terrenget. Det samme kartunderlaget blir også lagt inn som DTM-10(10x10 meter) for å sammenligne terrenget. I tillegg benyttes koordinatsystemet EUREF89 UTM 33 som er landsdekkende.

Dataene kommer i GeoTiff format, som behandles i Adobe Photoshop. Til slutt konverteres dataene til RAW format så det kan leses i Unity. Ved importering bygger Unity automatisk sammen terrenget.

2.6.2. Tekstur

Høyoppløselige bilder for teksturer hentes fra Norgebilder.no. Ved å benytte samme koordinater og underlag får vi teksturer som passer i forhold til terrenget. Når teksturene er lastet ned slipper vi å konvertere filene til et annet format siden Unity kan importere en rekke filformater som kartverket støtter. Teksturene må uansett skaleres ned for å passe terrengmodellen som er importert.

2.7. Behandling av kartdata

2.7.1. Fargemodell

Ubehandlet kartdata kommer i en gråtone modell som representerer høydekurvene i terrengmodellen. Gråtoneskalaen er en skala med nyanser fra svart til hvitsom er illustrert i Figur 1117. I raster sammenheng er terrenget høyere desto lysere nyansen er. Unity bruker den informasjonen til å kalkulere høyden på terrenget.

¹⁶ <https://www.kartverket.no/data/Smartkart/>

¹⁷ <https://techterms.com/definition/grayscale>



Figur 11: Gråtoneskalaen er en overgang fra hvit til svart

18

2.7.2. Fargedybde

Unity kan kun importere bilder i 8 og 16-bit dybde, les Figur 14 for mer informasjon om fargedybde. Ubehandlet kartdata fra 2.5.1 Høydedata kommer i et 32 bit format som virker nærmest uleselig for de fleste bildevisnings verktøy. Ved å konvertere bilde til 16 bit får vi et resultat som er likt Figur 9. Ved å konvertere til 8 bit kan man sammenligne kvalitetsforskjellen på terrenget når det er importert i Unity. Utrykket «bits» var mye omtalt om spillteknologi på 80 og 90 tallet, i den perioden var det en drakamp om kundene med å ha flest mulig «bits» i en konsoll¹⁹. I Figur 12 er det en enkel sammenligning av spillet Super Mario Bros til Nintendo Entertainment System(NES) og Super Nintendo Entertainment System(SNES). NES klarte kun å produsere 256 farger gjennom sin 8-bit teknolgi, mens SNES som kom ut noen år senere kunne vise opptill 16-bit.

Fargedybde handler om hvor mange unike farger som er tilgjengelig i et bilde en fargepalette definert i de binære nummerene 0 og 1. Hver pixel i et digitalt bilde er en kombinasjon av rød, grønt og blått(RGB). Anall unike fargekombinasjoner og shaders et bilde har tilgjengelig avhenger av hvor mange bits et bilde har. Et utrykk som benyttes er «bits per pixel» (bpp) som

18

¹⁹ <https://www.youtube.com/watch?v=fb73kHBAirU>

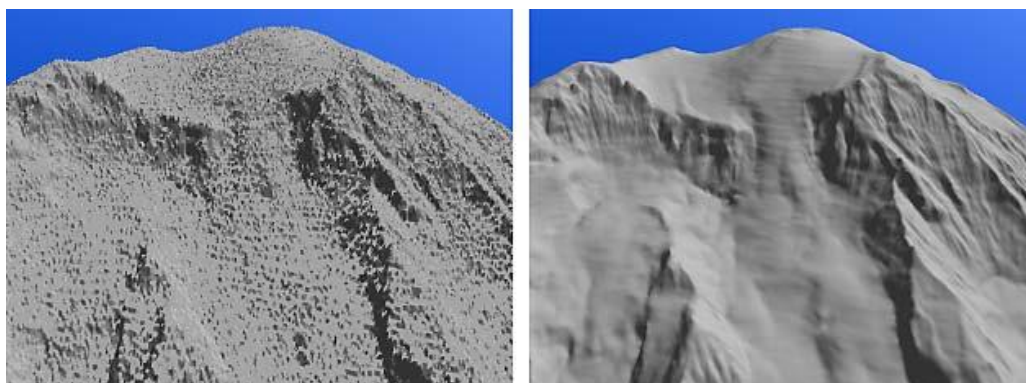
refererer til den totale summen av bits i hver hovedfarge(fargekanal) tilsammen i de ulike fargekanalene og representerer det toale antall farger i hver pixel²⁰.



Figur 12: Sammenligning av Nintendo konsollene NES med 8-bit og SNES med 16-bit grafikk. Spillet er Super Mario Bros

Vi ser en tydelig forskjell på terrenge i Figur 13 der bilde til venstre er i 8-bit, mens det til høyre er i 16-bit. Som Figur 14 viser er det kun 256 farger i et 8-bit bilde, mens 16-bit kan holde opp til 65536 farger.

²⁰ <https://www.cambridgeincolour.com/tutorials/bit-depth.htm>



Figur 13: Bilde til venstre er et 3D terreng av et fjell i 8-bit, mens bilde til høyre er i 16-bit.

Bits Per Pixel	Number of Colors Available	Common Name(s)
1	2	Monochrome
2	4	CGA
4	16	EGA
8	256	VGA
16	65536	XGA, High Color
24	16777216	SVGA, True Color
32	16777216 + Transparency	
48	281 Trillion	

Figur 14: Tabell som sammenligner antall farger per bit (fargedybde)

2.7.3. Konvertering

Unity kalkulerer en terrengmodell ut fra et hightmap som importeres, men tar kun imot RAW format, maks oppløsning på 4096x4096 pixler og 8 eller 16-bit fargedybde²¹. Dataene må konverteres, klippes og slås sammen igjen for få de til å passe. Dataene kommer i GeoTIFF format, siden det benyttes DTM1(1x1px = 1x1m) er filene over Hessdalen over 15000x15000px og kommer i en dybde på 32-bit. Filene må konverteres til 16-bit, klippes opp i mindre terreng slik at det kan lastes inn i Unity, og konverteres til RAW format.

Vi fant QGIS og GDAL til å være de beste løsningene til vår hensikt. QGIS et åpent program for å vise, analysere og redigere geografiske data egnet seg best til vår hensikt²². GDAL(Geospatial Data Abstraction Library), et åpent bibliotek for å lese å skrive rasterfiler²³. QGIS og GDAL bygger på mye av det samme, GDAL bruker kun shellscript med kommandoer

²¹ <https://docs.unity3d.com/Manual/terrain-OtherSettings.html>

²² <https://en.wikipedia.org/wiki/QGIS>

²³ <https://www.gdal.org/>

og er kompatibel med programmeringsspråket Python, mens QGIS har et brukergrensesnitt. Oppgavene QGIS gjør derimot er basert på GDAL kommandoer, som gjør det lettere å kryssjobbe med de ulike løsningene. Det positive er at GDAL kan konvertere til RAW format, som er det eneste formatet Unity tar imot, men ulempen er terrenget mister sine høydekurver i konverteringen til RAW. Det betyr at når fila importeres i Unity, vil det ikke lenger se ut som en dal, selv om Hessdalen har fjell opp mot 1300 meter.

Dermed har gruppen sett på Adobe Photoshop som en løsning til å konvertere fra GeoTIFF til RAW. Ulempen med Photoshop er at programmet leser ikke metadata som inneholder viktig informasjon om blant annet høyde og koordinater. Gruppen har blitt frarådet om å bruke programmet, men så langt er Photoshop den eneste løsningen som har gitt positive resultater. Det har blitt gjort flere forsøk på å konvertere mellom de ulike løsningene, som kan leses om under implementasjons kapittelet.

GeoTiff

GeoTIFF(Georeferenced Tagged Image File Format) basert på formatet (.tiff) med innebygd georefereringsinformasjon²⁴.

Photoshop RAW

Photoshop RAW(.raw) er et filformat for overføring av bilder mellom programmer og maskinplattformer²⁵, og er det eneste filformatet Unity tar imot for å generere en terrengmodell. Det må ikke forveksles med Camera RAW som er et format for overføring av bilder fra kamera til Photoshop. Det står lite dokumentasjon på Adobe sine egne sider om eksportering i RAW format, kontra importering i RAW som kan variere ut fra kameraprodusenten.

2.8. Valg av programvare

Unity

Valget av spillmotor faller under grunnlaget, det er tilpasset vårt prosjekt. De mest aktuelle spillmotorene gruppen har sett på er Unity og Unreal Engine 4, hvor valget ble Unity.

²⁴ <https://earthdata.nasa.gov/about/esdis-project/eso/standards-and-references/geotiff>

²⁵ <https://helpx.adobe.com/no/camera-raw/using/introduction-camera-raw.html>

Etter utforskning har gruppen konkludert med at Unity har en større brukermasse tilpasset for begynnere. Det er lettere å feilsøke problemer som oppstår underveis, en lettere læringskurve og finnes et større utvalg av plugins laget av samfunnet til en billig penge²⁶.

Spillmotorene oppgraderes stadig og har idag stort sett mange av de samme funksjonene, med ulike måter å håndtere å håndtere det. Det som setter Unity i skyggen av Unreal er antall kjente spill laget av spillsselskaper i den sistnevnte spillmotoren. Man kan se at AAA selskaper velger Unreal på grunn av blant annet den grafiske funksjonaliteten til å lage fotorealistiske spill²⁷. Sammenlikner man spill laget i Unity og Unreal finner man et større spekter av indie spill hos Unity²⁸, og det motsatte i Unreal²⁹.

Å velge Unity som spillmotor går i stor grad at prosjektgruppen har benyttet dette i tidligere fag på Høgskolen i Østfold som ITM21016 spillutvikling. Figur 15 viser en enkel sammenligning mellom Unity og Unreal og noen av de større forskjellene på spillmotorene.

Unity er også gratis for personlig bruk og så lenge inntekten ikke stiger over 100000 \$ per år, som beskrevet på deres egen nettside³⁰. Unity kan lett importere modeller laget i andre programmer som Maya 3DS max og Blender, i tillegg til å lett importere høydedata. En utfordring var at Unity har en grense på hvor store terreng som kan importeres, dette ble løst ved hjelp av pluginen “terrain stitcher” som setter sammen terreng med enkel funksjonalitet³¹.

26 <https://www.youtube.com/watch?v=baAPN0TCwkc&t=244s>

27 <https://sundaysundae.co/unity-vs-unreal/>

28 https://en.wikipedia.org/wiki/List_of_Unity_games

29 https://en.wikipedia.org/wiki/List_of_Unreal_Engine_games

30 https://store.unity.com/products/unity-personal?_ga=2.55772971.1465292936.1557762061-914599527.1556848038

31 <https://assetstore.unity.com/packages/tools/terrain/terrain-stitcher-42671>

Unity har en innebygd funksjon for å enkelt bygge spillet til å kjøre på web plattformer ved hjelp av WebGL. Ulempen er at det offisielt ikke støtter mobile plattformer i nettleseren enda³². Da måtte prosjektgruppen isåfall bygget en egen versjon for å publiseres på de mobile markedene.

	
Unity	Unreal Engine
<ul style="list-style-type: none"> • Uses JavaScript & C# • Good option for mobile game development • Possibility to build 2D/3D games • Supports 20+ platforms • Asset Store with samples & models • Intuitive interface • Supportive community 	<ul style="list-style-type: none"> • Uses C++ • Beneficial for PC & consoles • Best choice for AAA games • Supports 10 platforms • Next-gen graphic capabilities • Free to use (5% royalty from sales) • Blueprint visual scripting system

© Program-Ace

Figur 15: Sammenligning av Unity og Unreal

Maya

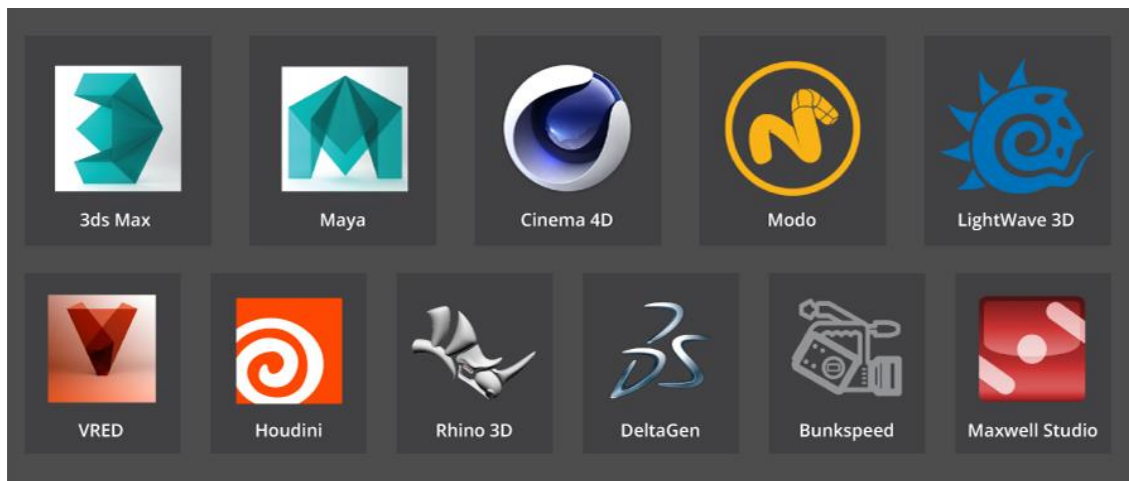
Grunnlaget for valg av 3D program kommer av tidligere erfaring og mulighetene til å kombinere det med Unity. Prosjektgruppen har tidligere benyttet Autodesk Maya i faget ITM30511 3D-modellering og animasjon ved Høgskolen i Østfold³³. Programmet er dessuten

³² <https://docs.unity3d.com/Manual/webgl-browsercompatibility.html>

³³ <https://www.hiof.no/studier/emner/it/2017/host/itm30511.html>

gratis til personlig bruk. Det finner mange alternativer til programmer for å lage 3D grafikk og animasjon, Figur 13 viser en oversikt over flere populære.

Sammenlikner man Maya og søsterprogrammet 3DS Max er Maya bedre for animasjon, mens 3DS Max leverer på modellering. 3DS Max fungerer kun på Windows plattformer som utgjør en av grunnene til å benytte Maya dersom en eller flere i gruppen bruker Mac OS³⁴. Blender, et open source 3D program er også tatt i betraktningen i valget av modelleringsprogram. Blender er gratis og medfører ingen kostnader dersom produktet skulle innhente en sum.



Figur 16: Oversikt over flere populære programmer brukt til å lage og animere 3D objekter

WebGL

WebGL(Web Graphics Library) er et JavaScript API for å rendere interaktiv 2D og 3D grafikk i hvilken som helst kompatibel nettleser uten bruk av plugin-moduler. WebGL gjør det ved å introdusere en API i tråd med OpenGL som kjøres i <canvas> elementet som ble introdusert i HTML5(Hyper Text Markup Language). I tillegg til IE11(Internet Explorer 11) er WebGL kompatibel med de fleste nettlesere som følger dagens standarder³⁵.

³⁴ <https://www.youtube.com/watch?v=8BT4YSOjTas>

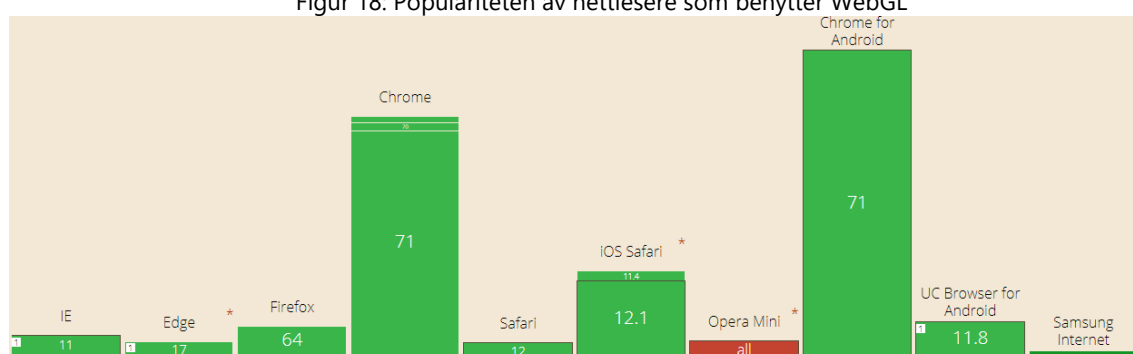
³⁵ https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android
		2-3.6	4-7	3.1-5	10-11.5									
		4-23	8-32	5.1-7.1	12.1-18	3.2-7.1								
6-10	12-17	24-64	33-71	8-11.1	19-56	8-11.4		2.1-4.4.4	7	12-12.1			10	
11	18	65	72	12	57	12.1	all	67	10	46	71	64	11	11.8
		66-67	73-75	12.1-TP		12.2								

Figur 17: Oversikt over nettlesere som støtter WebGL

WebGL støtten avhenger også av enheten sin grafikkprosessor. Det er ikke sikkert WebGL vil fungere optimalt på eldre enheter med utdaterte video drivere, selv om nettleseren er støttet. Figur 1736 gir en oversikt over nettleserversjoner som støtter WebGL, mens Figur 18 viser populariteten av nettlesere der WebGL brukes.

Figur 18: Populariteten av nettlesere som benytter WebGL



C#

Prosjektgruppen valgte å gå for programmeringsspråket C#, selv om Unity også støtter JavaScript. C# er et godt integrert språk og har god dokumentasjon i Unity, i tillegg er det mye mer hjelpemidler og tutorials som gjør det lettere å feilsøke. C# ble også benyttet som standard programmeringsspråk i faget ITM21016 Spillutvikling.

2.9. Innhenting av informasjon

For å samle informasjon om Hessdalen benyttet gruppen seg av et google docs dokument hvor vi samlet all nyttig informasjon vi fant. Vi brukte google for å finne så mye informasjon som mulig, og har også kontaktet kontaktpersoner for spesifikke kjente bygninger på e post, for å finne mer informasjon enn det som står på nett. En av grunnene til at vi gjør dette er for å få en oversikt over hva som skal modelleres, slik at vi kan begynne på det tidlig. Vi gjør det også for å finne informasjon som skal være med på kartet vi lager. Det vil være ikoner på siden som representerer

36 <https://caniuse.com/#feat=webgl>

viktige bygninger, og annet som er interessant i området, som brukeren kan klikke på for å få opp tekstbokser med informasjon.

2.9.1. Viktige Bygninger

Hessdalen kirke

Hessdalen kirke er en laftet langkirke med 186 sitteplasser. Den ble oppført på overgangen mellom 1930- og 1940-tallet og formelt vigslet først i 1949. Den er lokalisert på vestsiden av fylkesvei 576 og parallelt med den. Orienteringen er fra nordøst til sørvest. Kirken har takrytter nær nordøst enden, der inngangen før øvrig er gjennom en dør nær enden av langveggen.

Hessdalen UFO camp

Hessdalen UFO camp er et område med store infotavler og lavvoer laget av tre, hvor det tilbys overnatting i samarbeid med Visit Hessdalen. De kan tilby en helt unik friluftsopplevelse.

Blue Box Observatorium

Hessdalen automatiske målestasjon (AMS) (Blue box) er en observasjonsstasjon i Hessdalen, som brukes for å registrere "Hessdalsfenomenet". Den har vært i bruk siden 7. August 1998, og er utstyrt med en magnetometer, to svart-hvit TV kameraer og et farge TV kamera. Den registrerer ofte lysfenomen, men siden den bare måler noen utvalgte parametere, kan den ikke forklare alle registreringene.

Bekken Gård

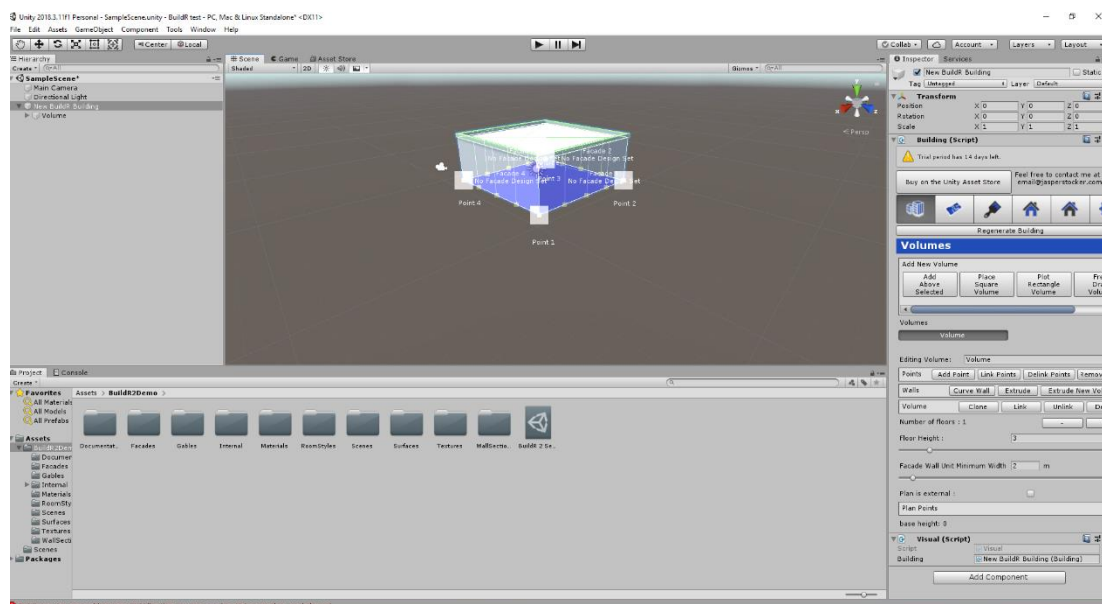
Bekken Gård er en naturlig hestegård med 25 års erfaring. De har 15 islandshester som kan brukes til å dra på fjellturer. Bekken gård ligger lengst sør i Hessdalen, 650 m.o.h. og er opprinnelig en gammel gruvearbeider gård fra midten av 1700-tallet, og fremstår fortsatt slik.

2.10. BuildR 2

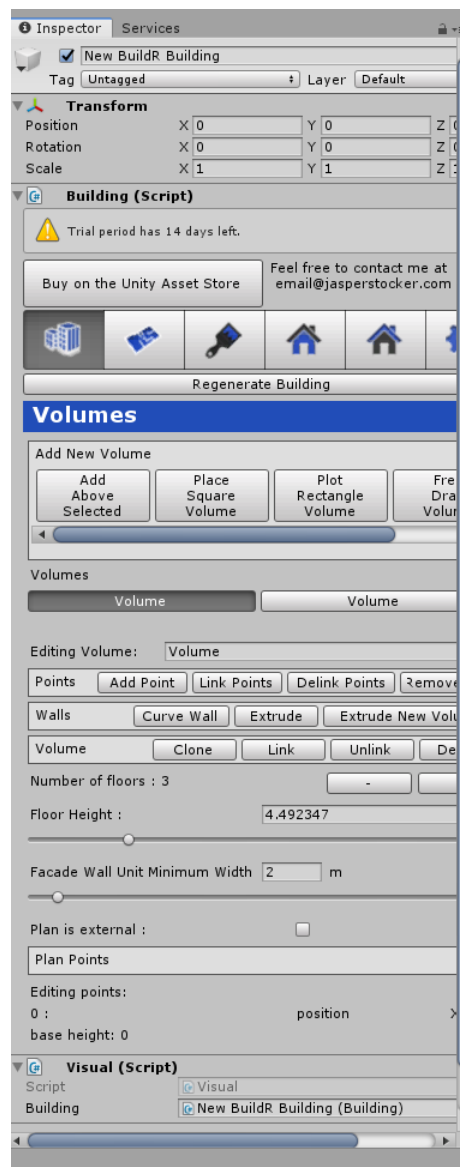
BuildR 2 ble testet av gruppen som et alternativ til å 3D modellere bygninger i Maya. BuildR 2 er tilgjengelig gjennom unity assets store. Det gir brukeren mulighet til å lett generere bygninger i Unity. Det er laget av Jasper Stocker. Han er også den som har lagt ut alt av dokumentasjon og tutorials som gruppen kunne finne.

For å lage en bygning i Unity med BuildR 2, går du på “tools” -> “BuildR” -> “Create new building”. Velg området som bygningen skal dekke ved en av alternativene under “add new volume”, slik som du kan se på Figur 19. Deretter kan du legge til flere lag på bygningen, samt en rekke andre valg i inspector menyen, vist på Figur 20. Legg til teksturer og vinduer osv. ved å gå inn på de i assets foldern, og dra og slipp de på en vegg, Figur 21 og 22 viser dette. Velg tak ved å gå inn på “roof designs”.

Gruppens første tanker er at det kan brukes til å lettere lage store byer, men man har ikke like mye frihet til å lage bygninger med særegne detaljer som ved å 3D modellere i Maya (mulig at dette er fordi gruppen har mer erfaring med å bruke det). Det har også muligheten til å lage interior design, som ville gjort det mye lettere for oss dersom det var nødvendig. Men med tanke på at vi ikke har bruk for interior design, har det ikke noen nytte for oss. Gruppens vurdering er å holde oss til å 3D modellere i Maya, men mindre vi skal masseprodusere generiske hus til kartet.



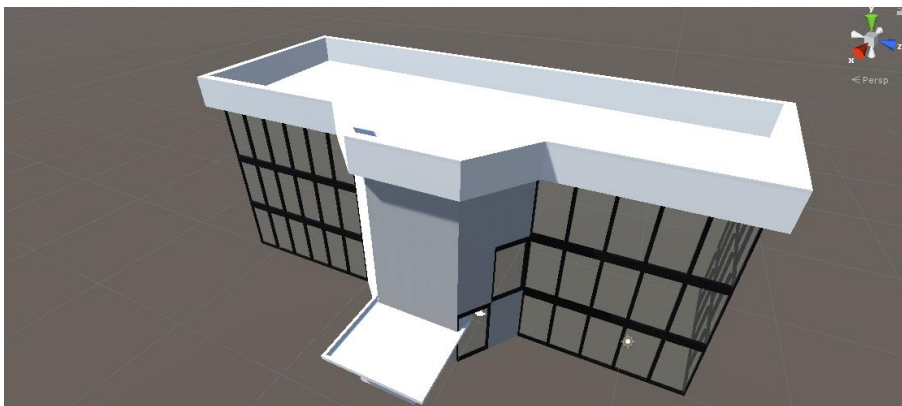
Figur 19: BuildR 2 i Unity



Figur 20: Inspector til bygningen



Figur 21: Materials menu



Figur 22: Test på å lage en bygning

Youtube turtoirals: <https://www.youtube.com/user/irreal1/videos37>

Dokumentasjon om BuildR 2: <http://support.jasperstocker.com/buildr2/38>

3. Begreper og definisjoner

3.1. 3D Grafikk

3.1.1. Modellering

Modellering beskriver prosessen å forme og danne et objekt . De vanligste kildene til 3D-modeller er de som en designer eller ingeniør lager ved hjelp av modelleringsverktøy, og modeller skannet inn i en datamaskin fra virkelige objekter. Tredimensjonale(3D) modeller representerer et fysisk objekt satt sammen av punkter(vertex) i et 3D område, koblet sammen av ulike geometriske enheter som trekanter, linjer og overflater. Det finnes flere måter å representere en modell i et 3D verktøy, de 2 mest populære er Polygonal og Curve modellering(NURBS)³⁹.

Polygonal modellering er satt sammen av minst 3 punkter(vertices) forbundet med linjesegmenter for å danne et polygon Mesh. De fleste 3D-modeller er bygget på teksturerte polygonale modeller, fordi de er fleksible og fordi maskiner kan rendere dem raskt. Imidlertid er polygoner flate og kan kun lage en bue ved hjelp av mange polygoner.

³⁷ <https://www.youtube.com/user/irreal1/videos>

³⁸ <http://support.jasperstocker.com/buildr2/>

³⁹ https://en.wikipedia.org/wiki/3D_modeling

I Curve modellering er overflater definert av kurver, som påvirkes av vektete kontrollpunkter. Kurven følger som regel punktene, ved å øke vekten på et punkt kurven trekke seg nærmere det punktet. Figur 19 viser en visuell forklaring av elementene i et polygon mesh.

3.1.2. Polygon mesh

En samling av vertices, edges and faces som definerer et trekantet objekt i 3D-grafikk og modellering⁴⁰.

Edge

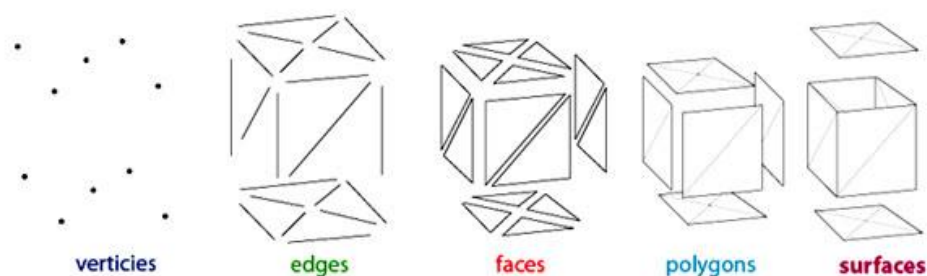
Edge er en linje som hjelper å definere et 3D objekt ved at det kobler sammen to vertices.

Vertex

Et vertex er alene bare et punkt i et 3D rom, men satt sammen for å lage et polygon mesh utgjør vertices(flere vertex) hjørnepunktene.

Face

Face er en overflate som fyller igjen rom laget av et sett med edges, en trekantet face har tre edges, mens firkantet face har fire edges. Face er den synlige delen av et polygon, som viser materialer når det blir truffet av lys.



Figur 19: Forklaring av begreper til en polygon mesh

⁴⁰ https://en.wikipedia.org/wiki/Polygon_mesh

3.1.3. Teksturer og materialer

Materialer og teksturer er egenskaper som rendermotorer(eks: Arnold) bruker til å rendere modellen. Man kan gi en modell materialer for å fortelle hvordan renderermotoren skal behandle lys når det treffer overflaten. Teksturer brukes til å gi materialer som legges på modellen farger eller et albedokart. Teksturer kan også legge til funksjoner som gir en 3D følelse i materialet ved hjelp av “bump” og eller “normal map”.

3.1.4. Rendering

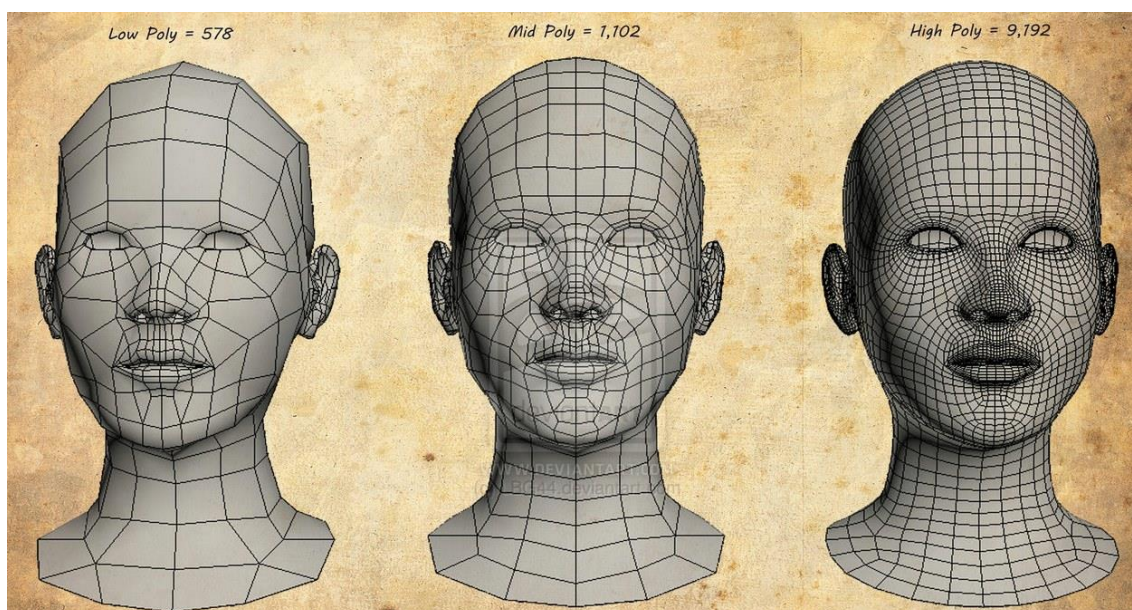
Rendering konverterer en modell til et bilde, de grunnleggende operasjonene i en realistisk rendering er transport(hvor mye lys kommer fra et sted til et annet) og spredning(hvordan overflater interagerer med lys)⁴¹. Renderingen kan gjøres i selve 3D programmet eller i en 3D-grafikk API. Det finnes også egne rendering programvare som gjør jobben, Autodesk Maya har tilleggsprogramvaren Arnold⁴² som renderer modellene som lages i programmet.

41 <https://www.easyrender.com/3d-rendering/3d-rendering-a-brief-explanation>

42 <https://www.arnoldrenderer.com/arnold/arnold-for-maya/>

3.1.5. Low and high polygons

Jo flere polygoner en mesh modell har, desto mer detaljert og realistisk vil modellen se ut. Bakdelen er at større modeller krever en del maskinkraft og går utover programmets ytelse. Det er viktig å balansere antall polygoner med hvordan modeller skal se ut, og fjerne unødvendige punkter for å spare maskinkraft⁴³. Figur 20 viser 3 modellerte hoder som har ulik antall polygoner. Som figuren viser har hodet til venstre 578 polygoner, mens det realistiske hodet til høyre har over 9000.



Figur 20: Ansiktsformet 3D modell i low, medium og high poly

3.2. Filformater

3.2.1. GeoTIFF/TIFF

TIFF (Tagged Image File Format) er format for lagring av bilder og dokumenter. Formatet er populært blant artister og trykk industrien⁴⁴. TIFF formatet er støttet for varierte handlinger som skanning, faksing, ord prosessering, optisk karakter gjenkjenning og skrivebords publisering.

⁴³<https://medium.com/@arjun07/differences-between-high-poly-vs-low-poly-3d-models-348cab56e82e>

⁴⁴ <https://www.lifewire.com/tif-tiff-file-2622393>

TIFF formatet støtter opptil 32-bit fargedybde(16777216 farger + alpha) og støtter størrelse opptil cirka 4 GB. Se Figur 14 for en oversikt over fargedybde.

GeoTIFF er en offentlig metadata standard basert på TIFF formatet. GeoTIFF tillater georeferering innebygd i TIFF filer med mulighet til å legge til informasjon som kartprojeksjoner og koordinatsystemer.

3.2.2. RAW

Formatet RAW har flere betydninger, som Camera RAW og Photoshop RAW. Camera RAW er en fellesbetegnelse for formater som benyttes ved digital fotografering. En Camera RAW-fil inneholder ubehandlede, ukomprimerte gråtone data fra kameraets digitale bildesensor. I tillegg følger det med data som forteller hvor og når bilde ble tatt(metadata) Photoshop kan importere og eksportere RAW filer ved hjelp av en ekstra programvare. Siden metadataene følger med kan bilde farge behandles før det legges inn i selve programmet.

3.2.3. PNG

PNG(Portable Network Graphics) er et bildeformat uten kvalitetstap, som opprinnelig ble laget som en erstatning av GIF(Graphics Interchange Format). PNG formatet har likheter med GIF, men PNG er ikke begrenset til 8-bit(256 farger)⁴⁵. Dette tillater at hver piksel å inneholde grader av gjennomsiktighet(transparent).

3.3. Web

3.3.1. HTML5

HTML5(Hyper Text Markup Language) er den 5 store utgivelsen av oppmarkering språket for å sette opp nettside struktur og fungerer som “skjelettet” til en nettside⁴⁶. HTML5 kom med nye funksjoner for å lage robuste web applikasjoner, bedre mobilvennlighet og semantikk. Et nytt element er <canvas> taggen som lar utviklere tegne grafikk ved bruk av programmeringsspråket JavaScript.

⁴⁵ <https://medium.com/@duhroach/how-png-works-f1174e3cc7b7>

⁴⁶ <https://www.techradar.com/news/internet/web/html5-what-is-it-1047393>

3.4. GDAL

3.4.1. GDAL_translate/warp

Gdal_translate og Gdal_warp er verktøy som benyttes i GDAL for å reprojektere og konvertere rastere. Der Gdal_translate brukes til konvertering, mens gdal_warp er for reprojektering.

3.4.2. Ot

Bestemmer hva slags utgangsbånd som skal følge den angitte datatypen. I prosjektet sin sammenheng kommer dataene fra Høydedata i 32-bit og blir konvertert til UInt16(16-bit).

3.4.3. Of

Velger formatet som filtypen skal konverteres til. Først inntaster en fila som skal konverteres, og deretter et nytt navn på fila pluss en forlengelse(Eks: .png) som fila skal konverteres til.

3.4.4. Scale

Reskalerer pixel verdier fra en minimum til maksimum verdi. Hver pixel har sin egen verdi som går fra hvit til svart. I et 8-bit bilde kan verdiene være mellom 0 til 255, mens UInt16(16-bit) kan verdiene være opptil 65536, som vist på Figur 14.

2122

4. Design

For å lage 3D-modellene av bygninger som skulle være med på kartet tok vi i bruk ulike metoder for å konstruere modellene ut ifra bildereferanser av de ulike bygningene. Dokumentasjonen bak hver enkelt av de 3D-modellerte bygningene er lagt til som vedlegg.

4.1 Metode 1 – 3D Modellering ut i fra bildereferanser

Går ut på å finne flest mulige bilder av bygningene som skal lages ved hjelp av google maps, google bilder og facebook bilder som tilhører organisasjonen bak en av bygningene. Dermed var det viktig å finne flest mulig bilder for å få en god oversikt over alle bygningene som tilhører et område, samt ulike vinkler for å få med så mange detaljer som mulig.

I de tilfellene hvor det finnes dødvinkler hvor vi ikke har noen bilder fra en side av en bygning, er det tatt en vurdering av hva som sannsynligvis finnes der, ut ifra ulike faktorer slik som hva slags bygning det er, hva som er synlig på de sidene som er i bildet, og eventuelt hva som mangler ut i fra det. For eksempel, hvis ikke er noen dør synlig på et hus i bildet, må det ligge i dødvinkelen.

Bekken Gård og Samfunnshuset ble laget uten at det ble tatt screenshots underveis. Men samme metode for å modellere som for Pub1 og Blue Box, ble benyttet for Bekken Gård og Samfunnshuset. Derfor valgte gruppen at det skulle lages en til bygning der det ble tatt screenshots, slik at vi lettere kunne dokumentere den prosessen, i stedet for å reverse jobben som er gjort med Bekken Gård og Samfunnshuset.

Det vil også si at detaljene på hva som er dokumentert er mye nøyere på Pub1 og Blue Box, enn Samfunnshuset og Bekken Gård.

Rødekorshytta og UFO camp har også egen dokumentasjon på vedlegg, mens infoplassen o

Figur 25 og 26 viser et eksempel på en ferdig 3D-modell ut ifra denne metoden. Dette er Pub1.



Figur 23: Pub1 sett fra Google Street view



Figur 24: 3D-modell av Pub1

4.2 Metode 2 – 3D Modellering uten å ha nøyaktige mål av bygget

4.2.1 Finn mål av design objekt

Her prøvde jeg flere metoder. Benyttet først Wikipedia i ⁴⁷. Etter å ha lest igjennom det lille som sto på Wikipedia viste meg videre til *Hessdalen Kyrkje 1949-1989 ii* ⁴⁸ og kapittelet “Hessdalen Kirke” i *Våre kirker. Norsk kirkeleksikon iii* ⁴⁹.

Etter nøye gjennomlesning av de 3 første linkene fikk jeg ikke den informasjonen jeg ønsket. Jeg bestemte meg da for å sende en mail til Ålen Sogn, som Hessdalen Kirke ligger under, hvor jeg etterspurte assistanse om fler nøyaktige bilder, samt om de hadde informasjon på bygget. Etter en god stund fikk jeg ikke svar, så jeg bestemte meg for å gå videre.

Jeg gikk inn på Kartverket⁵⁰ for å finne informasjon om eiendom. Etter å ha tastet inn informasjonen de trengte så fikk jeg dette bildet:

⁴⁷

https://no.wikipedia.org/wiki/Hessdalen_kirke

⁴⁸ https://www.nb.no/items/URN:NBN:no-nb_digibok_2010111510008

⁴⁹ https://www.nb.no/items/URN:NBN:no-nb_digibok_2014121107543#240

⁵⁰ <https://seeiendom.kartverket.no/>

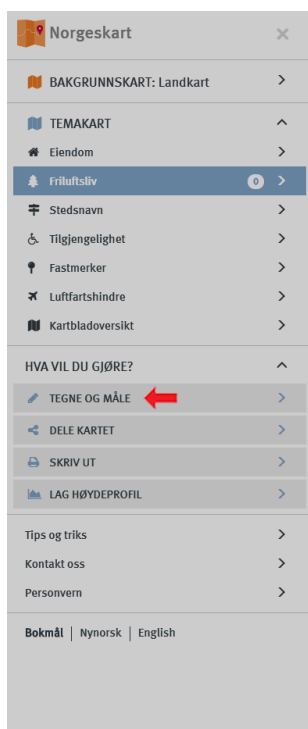


Figur 25: Oversiktsbilde av Hessdalen Kirke

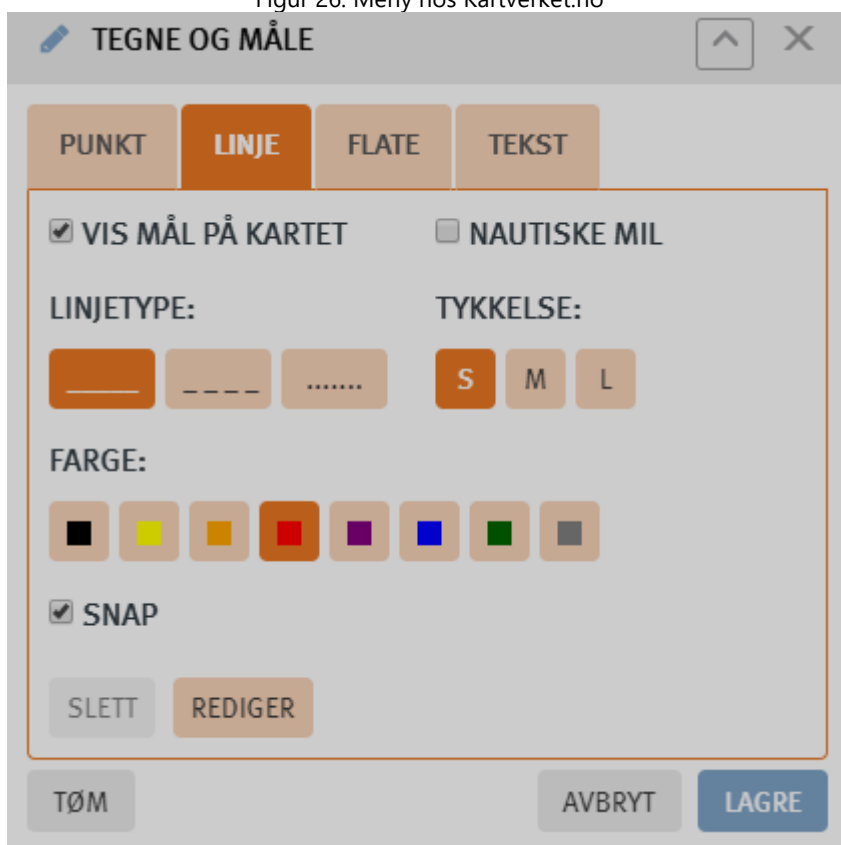
⁵¹ *Oversiktsbilde av Hessdalen Kirke*

Kartverket er veldig greit å benytte fordi de har innebygget verktøy for å måle ca distanse mellom to eller flere punkter.

Oppe i venstre hjørne av vinduet så er det et søkefelt. Til venstre i feltet er det 3 streker, dette er en meny. Ved å trykke på denne så ekspanderer menyen seg. Illustrert med Figur 26. Jeg benyttet funksjonen de har kalt: Tegne og Måle.



Figur 26: Meny hos Kartverket.no



Figur 27 Tegne og måle verktøy

Tegne og måle verktøy

I Figur 26 er det en rød pil som illustrer hvor Tegne og måle verktøyet ligger. Figur 27 viser hvordan tegne og måle verktøyet ser ut.

Ved å trykke ca på veggen som er avbildet og trekke den så godt som 90°.

I mitt eksempel kommer det at langveggen på kirka er 17 meter.

4.2.2 Finne bilder av design objekt

Her kan man gå frem på forskjellige metoder, blant annet kan man bare søke på Google Bilder for å finne eksempler.

Ved å benytte Google Maps sin street-view funksjon⁵² så fant jeg dette bildet.

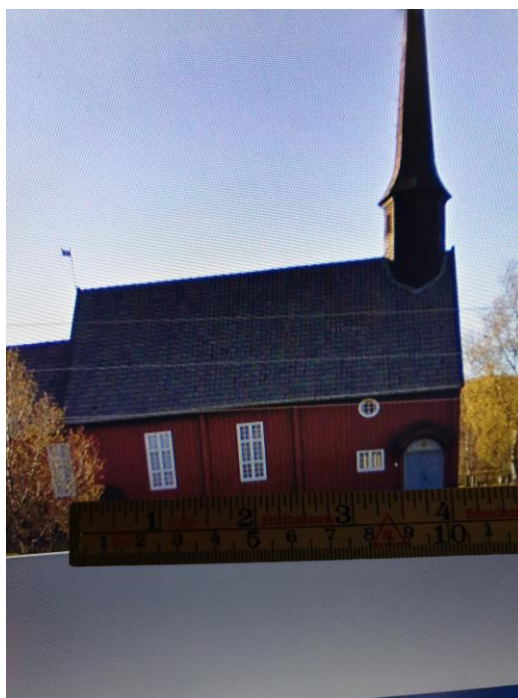


Figur 28: Hessdalen kirke fra Google Maps Street View

Ved å benytte en tommestokk så finner jeg ut at fremste del av kirka er 10 centimeter lang.

⁵²

https://www.google.com/maps/@62.7905832,11.1819215,3a,49y,290.87h,105.46t/data=!3m6!1e1!3m4!1stjeScPkm_9oduoGUi_XZCw!2e0!7i13312!8i6656

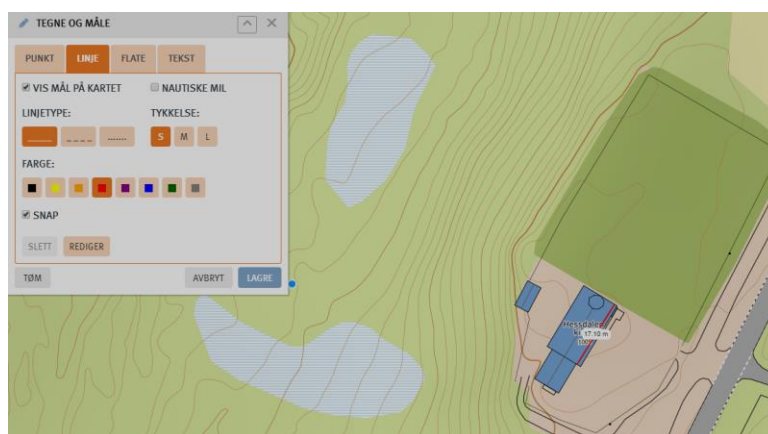


Figur 29: Illustrering av oppmåling med tommestokk

Deretter må jeg konvertere fra enten meter eller centimeter. For enkelheten sin skyld i dette prosjektet velger jeg å konvertere alle mål til meter.

Da får jeg frem at veggen, som er 10 centimeter blir 0.010 meter.

Når jeg da vet at veggen er 17 meter, som illustrert i dette bildet:



Figur 30: Viser at veggen er 17 meter lang

4.2.3 Utregning

Ved da å sette opp følgende regnestykke

$$X = (17 * X) / 0,010$$

Jeg ønsker å finne lengden på den veggen, den er målt til 4.5cm

Regnestykket blir da:

$$X = (18 * 0.045) / 0.12$$

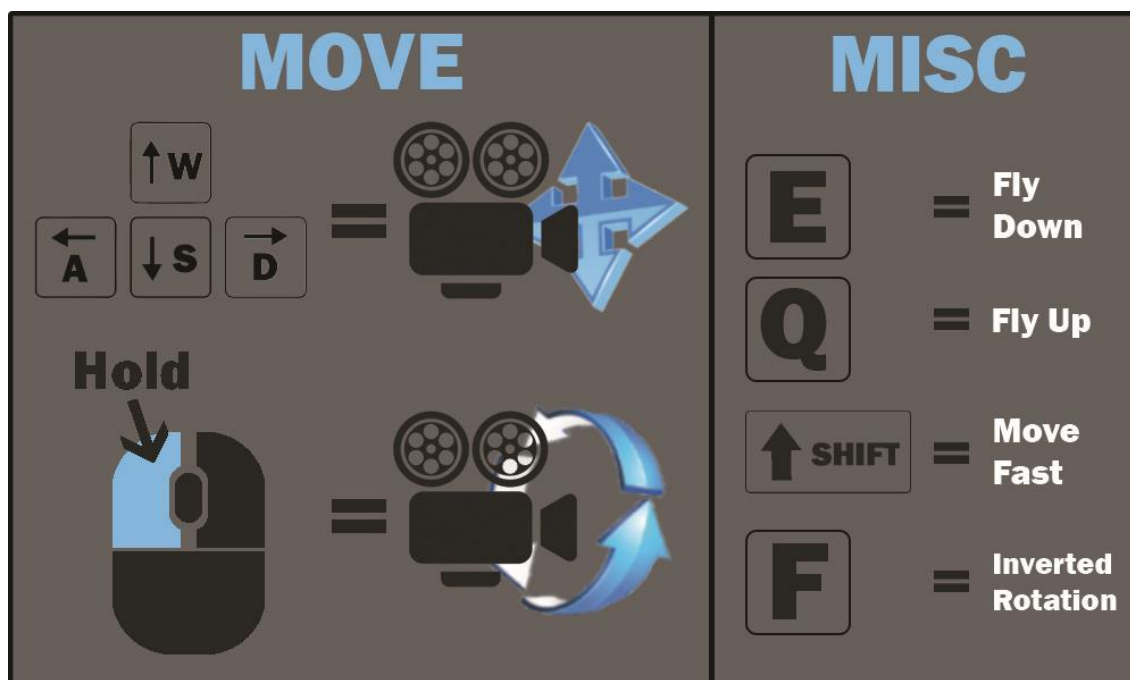
$$X = 6,75\text{m}$$

Veggen som jeg målte til 4.5cm på bildet, er da 6.75m høy.

5. Implementasjon

5.1. Brukerkontroller

Kontrolleren i Unity er et kamera som fritt kan fly rundt på kartet. Som vist på Figur 31 bruker en både tastatur og datamus for å manøvrere kameraet. Det er lagt til tilleggsfunksjoner under “misc” for å blant annet fly fortere og invertere rotasjonen.



Figur 31: Brukerkontroller for å navigere kamnera

Manøvrering og rotasjon er lagt i hvert sitt C# script for å holde det så ryddig som mulig, Figur 32 viser scriptet for manøvrering. Øverst i scriptet er det satt opp variabler som gjør at en kan endre hastighet og avgrense hvor kamnera kan fly i Unity editoren. Skulle kamnera forsøke å fly vekk fra den avgrensede sonen blir det plassert tilbake på den angitte posisjonen under “min and max values”.


```

public class cam_controller : MonoBehaviour {

    [Header("Speed")]
    public float normalMoveSpeed = 10;
    public float fastMove = 3;

    [Header("Min and max values")]
    public Vector3 min;
    public Vector3 max;

    void Update() {
        if (Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.UpArrow)) { Move(Vector3.forward); }
        if (Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.DownArrow)) { Move(Vector3.back); }
        if (Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.LeftArrow)) { Move(Vector3.left); }
        if (Input.GetKey(KeyCode.D) || Input.GetKey(KeyCode.RightArrow)) { Move(Vector3.right); }
        if (Input.GetKey(KeyCode.Q) || Input.GetKey(KeyCode.PageUp)) { Move(Vector3.up); }
        if (Input.GetKey(KeyCode.E) || Input.GetKey(KeyCode.PageDown)) { Move(Vector3.down); }
    }

    private void Move(Vector3 direction) {

        transform.Translate(direction * normalMoveSpeed * Time.deltaTime);

        if (Input.GetKey(KeyCode.LeftShift) || Input.GetKey(KeyCode.RightShift)) {
            transform.Translate(direction * normalMoveSpeed * fastMove * Time.deltaTime);
        }

        Vector3 pos = transform.position;
        pos.x = Mathf.Clamp(pos.x, min.x, max.x);
        pos.y = Mathf.Clamp(pos.y, min.y, max.y);
        pos.z = Mathf.Clamp(pos.z, min.z, max.z);
        transform.position = pos;
    }
}

```

Figur 32: Kamerakontroller script

Rotasjon scriptet som vist på Figur 33 kontrollerer rotasjonen til kamera. Det er lagt inn et alternativ for å skru av og på inverted rotasjon som gjør at kamera roterer motsatt retning brukeren beveger datamusa. Inverted rotasjon er et alternativ fordi de fleste større 3D kart plattformene som er analysert i kapittel 2.3 har denne funksjonen, og vil være tilgjengelig for brukere som er vane med inverted rotasjon.


```

public float speed = 3.5f;
private float X;
private float Y;

public bool inverted = false;
public Texture2D grab;
public CursorMode cursorMode = CursorMode.Auto;
public Vector2 hotSpot = Vector2.zero;

void Update()
{
    if (Input.GetKeyDown("3")) { inverted = !inverted; }

    if (Input.GetMouseButton(0)) {
        Cursor.SetCursor(grab, hotSpot, cursorMode);
        if (inverted) {
            transform.Rotate(new Vector3(Input.GetAxis("Mouse Y") * speed, -Input.GetAxis("Mouse X") * speed, 0));
        }
        else {
            transform.Rotate(new Vector3(-Input.GetAxis("Mouse Y") * speed, Input.GetAxis("Mouse X") * speed, 0));
        }

        X = transform.rotation.eulerAngles.x;
        Y = transform.rotation.eulerAngles.y;
        transform.rotation = Quaternion.Euler(X, Y, 0);
    }
    else {
        Cursor.SetCursor(null, Vector2.zero, cursorMode);
    }
}
}

```

Figur 33: Kamera rotasjon script

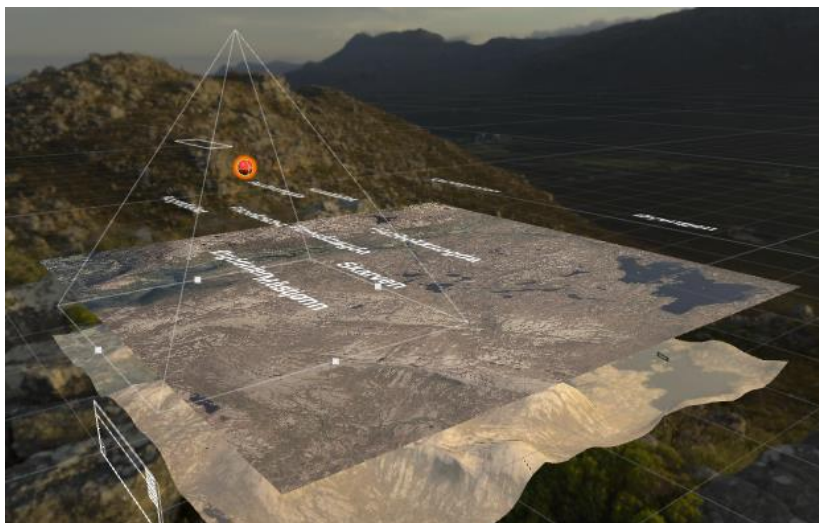
5.2. Navigasjon

5.2.1. Minikart

Et eget kamera(kamera2) er satt opp for å rendre et minikart med stedsnavn i GUI. Det er gjort ved å legge et bilde av teksturen til Hessdalen brettet ut over terrenget. Ved hjelp av scriptet på Figur 35 er kamera2 relativt til spiller kameraet sin posisjon på x- og z-aksen, mens det er rotert 90 grader på x aksen. Kamera2 sin posisjon justeres manuelt på y-aksen for å enkelt finne best mulig høyde i forhold til minikart bilde. Figur 34 og Figur 36 viser minikartet med stedsnavn og et rødt objekt, objektet representerer spiller kameraet sin nøyaktige posisjon på x- og z-aksen og er et child element av kamera2.



Figur 34: Minikart som bruker ser i GUI



Figur 36: Minikartet sett fra perspektiv i Unity editoren. Det ligger over 3D kartet, men spiller kameraet renderer det ikke.

```
public class minimap_Controller : MonoBehaviour {
    public Transform player;

    void LateUpdate () {
        Vector3 newPosition = player.position;
        newPosition.y = transform.position.y;
        transform.position = newPosition;

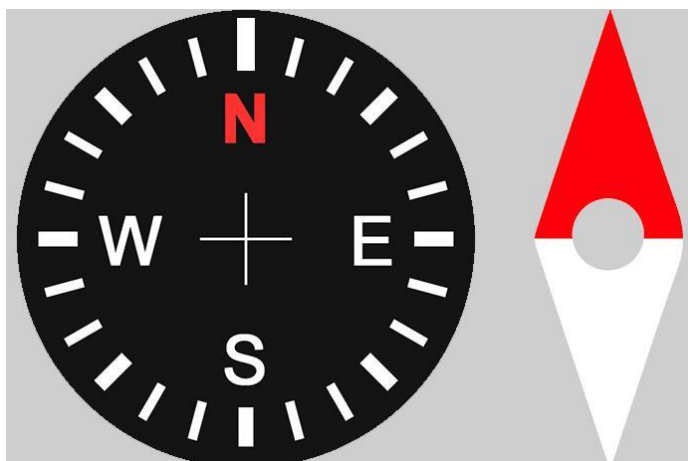
        transform.rotation = Quaternion.Euler(90f, player.eulerAngles.y, 0f);
    }
}
```

Figur 35: Minikart script som får minikart kamera til å følge spiller kamera sin relevante x og z posisjon

Alle objekter kan bli tildelt et layer i Unity sitt eget layer system. I dette tilfellet har minikart med tilhørende objekter fått sitt eget layer ved navn “minimap”, der kamera2 renderer kun objekter som er i dette layeret. Figur 36 viser minikartet brettet ut over terrenget med stedsnavn og aktuell posisjon. Hovedkameraet ser alt annet utenom minikartet, dette sparer blant annet ressurs kraft, slik at ting ikke blir rendret flere ganger. Stedsnavn på minikartet roteres alltid riktig vei i forhold til kamera2 ved hjelp av scriptet vist på Figur 35.

5.2.2. Kompass

Kompasset, som vist i Figur 37 består av nåla som er et statisk bilde som alltid peker den veien brukeren ser, og kompasset som roterer relativt med kameraet.



Figur 37: Kompass og nål som brukes i GUI

Scriptet i Figur 38 viser hvordan kompasset roterer på sin z-akse relativt til kameraet sin y-akse.

```
public class Compass : MonoBehaviour
{
    public Transform player;
    Vector3 vector;

    void Update()
    {
        vector.z = player.eulerAngles.y;
        transform.localEulerAngles = vector;
    }
}
```

Figur 38: Kompass script

5.2.3. Stedsnavn

Tekst komponenter benytter seg av et canvas, som er et område for å vise UI elementer. Stedsnavn vises 3 forskjellige steder, i terrenget, på minikartet og i GUI. Stedsnavn i terrenget som vist på Figur 39 bruker et bilde komponent som bakgrunn med en gjennomsiktig svart farge. Dette er for å gjøre teksten mer leselig i et 3D område. For å differensiere stedsnavn og navn på bygninger er det benyttet forskjellige farger, der grønn tekst er stedsnavn, og hvit er navn på hus/bygninger.

```

public class Face_camera : MonoBehaviour {
    public Camera flyCamera;

    // Update is called once per frame
    void Update () {
        transform.LookAt(transform.position + flyCamera.transform.rotation * Vector3.forward, flyCamera.transform.rotation * Vector3.up);
    }
}

```

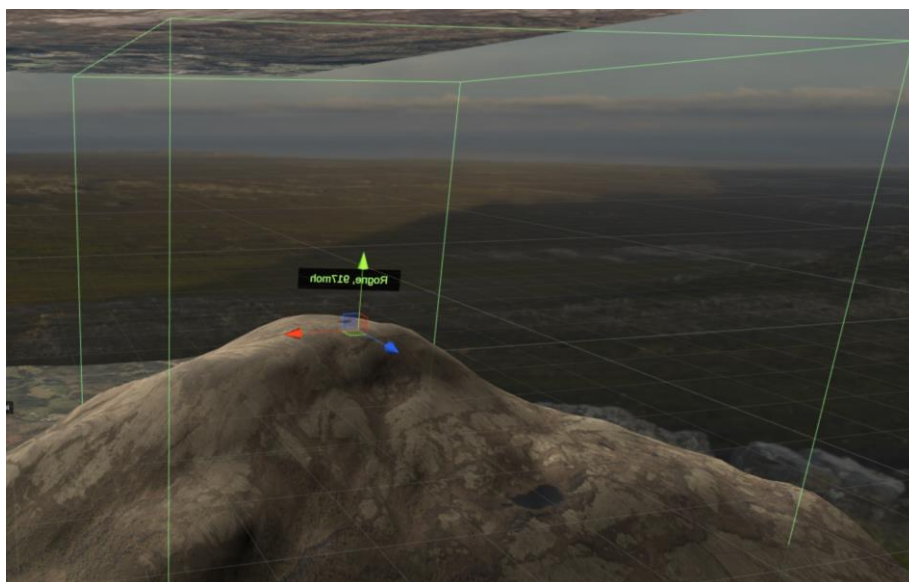
Figur 40: Script som roterer 2D tekst i retning mot kamera

Siden tekst komponentene er 2D elementer benyttes scriptet i Figur 40 som alltid roterer teksten riktig vei mot kamera. UI komponenter med dette scriptet vil alltid vendes mot kamera, uansett hvilken retning det er.



Figur 39: Navn på bygning i terreng

Stedsnavn i GUI viser navnet på et visst område bruker befinner seg i. Som vist i Figur 41 er det et "tomt" 3D objekt med en box collider som definerer det aktuelle område. Kamera har også en collider, så ved hjelp av scriptet i Figur 42 vil navnet på stedet vises i GUI når kamera treffer objektet, og forsvinner når kamera forlater det.



Figur 41: En urendret kube som definerer et bestemt område/sted. Stedsnavn vil vises i GUI når kamera kolliderer med den aktuelle kubens.

```

public class NameAppear : MonoBehaviour {

    public GameObject customText;

    private void Start()
    {
        customText.SetActive(false);
    }
    void OnTriggerEnter(Collider name)
    {
        if(name.gameObject.tag == "MainCamera")
        {
            customText.SetActive(true);
        }
    }

    void OnTriggerExit(Collider name)
    {
        if (name.gameObject.tag == "MainCamera")
        {
            customText.SetActive(false);
        }
    }
}

```

Figur 42: Script som sjekker om «MainCamera» kolliderer med et bestemt objekt

Et objekt kan få tildelt en «tag», i dette tilfellet har kameraet taggen «MainCamera» som blir sjekket av scriptet når noe kolliderer med objektet scriptet er tildelt. Teksten vises når objektet som kolliderer har den riktige taggen, og forsvinner når objektet med den samme taggen flyr ut.

5.2.4. Teleportering

Klikker en på et objekt i nedtrekksmenyen vil kamera bli teleportert til aktuell destinasjon på kartet. Det er gjort ved å plassere tomme 3D objekter ved de aktuelle stedene og bruke posisjonen til objektet som referanse. Scriptet i Figur 43 fungerer som en base for posisjonen til

```

public GameObject target;

public GameObject TPLoc1;
public GameObject TPLoc2;
public GameObject TPLoc3;
public GameObject TPLoc4;
public GameObject TPLoc5;
public GameObject TPLoc6;
public GameObject TPLoc7;
public GameObject TPLoc8;
public GameObject TPLoc9;
public GameObject TPLoc10;

public void TP1() { target.gameObject.transform.position = TPLoc1.gameObject.transform.position;
target.gameObject.transform.eulerAngles = TPLoc1.gameObject.transform.eulerAngles; }

```

Figur 43: Script som holder på koordinatene brukeren kan teleportere til. alle objektene og kameraet. Kamera i dette tilfellet er «target», ved å klikke på knappen i menyen relatert til objektet «TPLoc1» blir kamera teleportert til posisjonen til den destinasjonen.

5.3. Nedtrekksmeny

Det er menyen som holder på navn som bruker kan teleportere til. Menyene er en knapp som automatisk åpner menyen dersom brukeren flytter musepekeren over den. Menyene er et UI element bestående av Canvas med flere knapper under seg. Scriptet i Figur 44 kontrollerer om menyen skal være åpen eller lukket, hastigheten når den åpnes og lukkes og hvordan den er i lukket tilstand.

```
public RectTransform container;
public bool isOpen;

// Use this for initialization
void Start () {
    container = transform.Find("Container").GetComponent<RectTransform>();
    isOpen = false;
}

// Update is called once per frame
void Update () {
    Vector3 scale = container.localScale;
    scale.y = Mathf.Lerp(scale.y, isOpen ? 1 : 0, Time.deltaTime * 12);
    container.localScale = scale;
}

public void OnPointerEnter(PointerEventData eventData) {
    isOpen = true;
}

public void OnPointerExit(PointerEventData eventData) {
    isOpen = false;
}
```

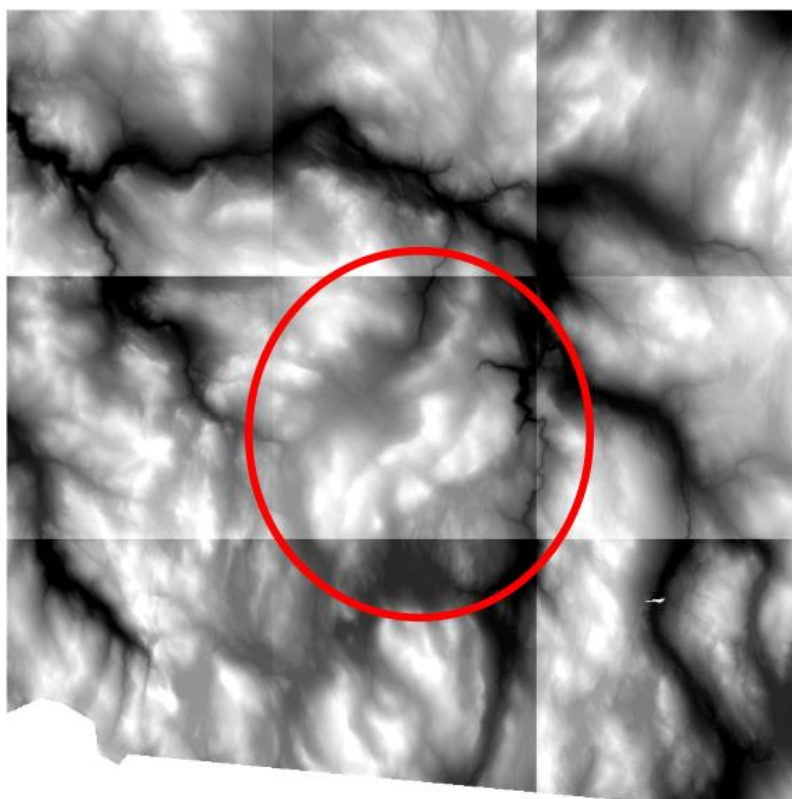
Figur 44: Script som trekker sammen alle underobjekter i et objekt og lager en nedtrekksmeny. Åpnes og lukkes når brukeren holder musepekeren over objektet.

5.4. Overflate

5.4.1. Klippe terreng

Utsnittet over Hessdalen i DTM1(1x1px) er på litt over 15000x15000px. Unity tar et bilde på maks 4096x4096px, men for å spare maskinkraft og størrelse vil vært terreng klippes ned til 2048x2048px. Terrenget kan ikke skaleres, fordi da blir området mindre. Ved å laste ned fra Høydedata kan man få ekstra terreng som ikke skulle vært med. Her er dokumentasjonen til Høydedata litt dårlig, fordi det står ikke hvor mye ekstra man kan få med dersom man velger et utsnitt.

Fremgangsmåten var å lime sammen hele terrenget i Qgis, så man får et stort utsnitt. Siden vært terreng som skal eksporteres til Unity må være 2048x2048px, må det klippes ut deler av terrenget og lagres i nye filer. Figur 45 viser et stort område rundt Hessdalen i DTM1 oppløsning. Det er lagt på en rød ring som viser området som er brukt til prosjektet.



Figur 45: Holtålen kommune og omegn satt sammen av 9 heightmaps i programmet QGIS. Området med ring rundt er Hessdalen.

Qgis har et saks verktøy for å klippe terreng, for å bruke det må en oppgi koordinater som skal klippes ut. Figur 46 viser felter for x og y verdier som må fylles ut. X er i dette tilfellet Øst, mens Y er Nord. Desto høyere verdi på X jo lengre øst, høyere verdi på Y er lenger nord.

Velg utstrekning ved å dra i kartvinduet eller endre utstrekningens koordinater					
1	x	<input type="text" value="303392.255797"/>	x	<input type="text" value="306768.005797"/>	
	y	<input type="text" value="6973498.63959"/>	2	y	<input type="text" value="6968944.05625"/>

Figur 46: Saks verktøy i programmet Qgis for å klippe ut deler av terreng. X defineres som øst, mens Y defineres som nord. Koordinatene i feltene er fra et område i Hessdalen.

5.4.2. Importere/eksportere

Importer av terrengdata fra Kartverket er gjort i flere omganger i ulike prosesser for å sammenligne utfallet. Gruppen ble anbefalt å ikke involvere Adobe Photoshop i prosessen da

dette ikke tar hensyn til filene sine metadata. Etter flere forsøk ved å konvertere til RAW format i GDAL og QGIS fant vi ut at høydekurvene forsvant i prosessen, og må derfor gjøre siste konvertering i Photoshop for å få det til å se ut som det er et terreng. Denne seksjonen viser flere prosessene som er gjort og resultatet av hvert utfall. For en beskrivelse av GDAL kommandoer som er oppgitt i figurene i denne seksjonen, les kapittel 3.4.

Unity har et eget verktøy for å styre oppløsning til terrenger som vist i Figur 47. Terrain width og length styrer hvor stort terrenget er på x og z aksene. Terrain height styrer y aksene, som standard er på 600. Resolution er oppløsningen til selve terrenget som importeres + 1, det vil si hvis terrenget er på 2048x2048px skal resolution være 2049. I dette tilfellet er høyden og bredden på objektet i Unity 2048x2048px, mens terrenget som importeres er på 4096px.

Resolution	
Terrain Width	2048
Terrain Length	2048
Terrain Height	600
Heightmap Resolution	4097
Detail Resolution	1024
Detail Resolution Per P	8
Control Texture Resolu	512
Base Texture Resolutic	1024

Figur 47: Instillinger i Unity for å endre egenskaper til terreng. De viktigste instillingene er width, length, height og heightmap resolution.

Forsøk 1:

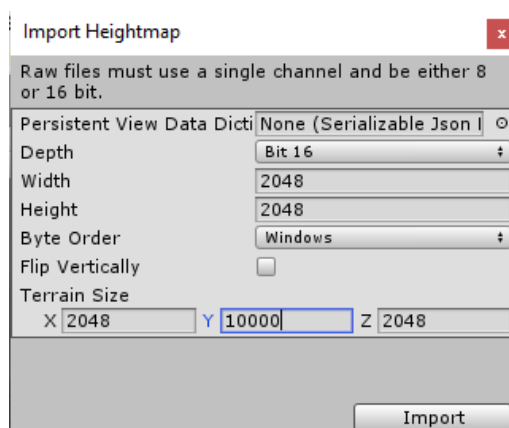
Hele prosessen er kjørt kun i GDAL, Figur 48 viser kommandoen som er kjørt.

Original fila er i 32-bit GeoTIFF format, og konverteres til 16-bit RAW.

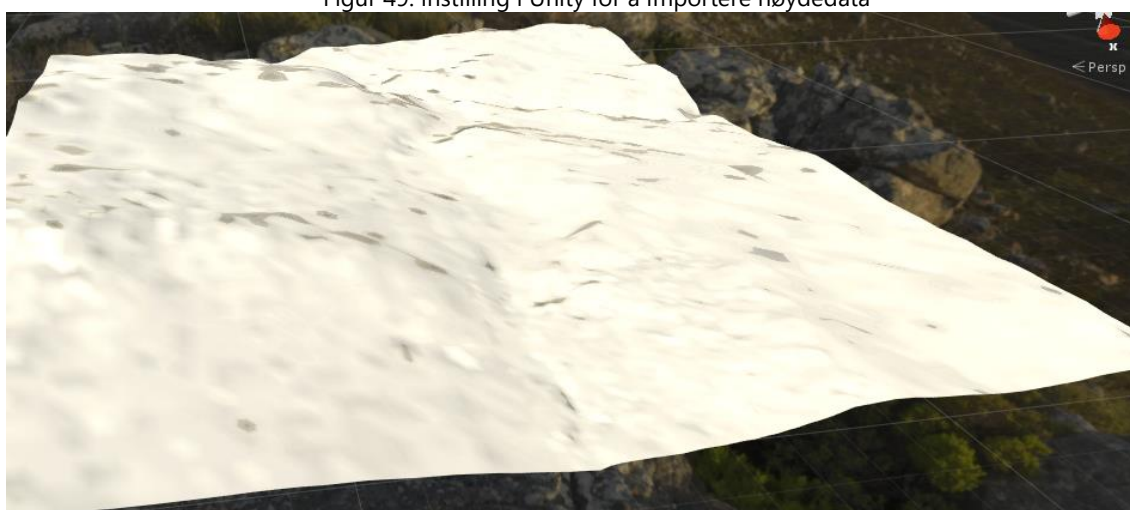
```
B:\bachelor_data\rawkartdata\dtm1\resampled_data>gdal_translate -ot UInt16 -scale 170.819 818.859 -of ENVI_scaled_dtm1_upper_left.tif terrain_upper_left.raw_
```

Figur 48: Kommando i GDAL for å konvertere et 32-bit heightmap i GeoTiff til 16-bit RAW

I importerings vinduet til Unity er Y aksene satt til maks, altså 10000 Unity units, som vist i Figur 49. Resultatet i Figur 50 viser at terrenget såvidt har noen høydekurver, fjellene i Hessdalen kan være opp mot 1300 meter over havet og dette terrenget er ikke i nærheten selv om det er skalert til maks høyde.



Figur 49: Instilling i Unity for å importere høydedata



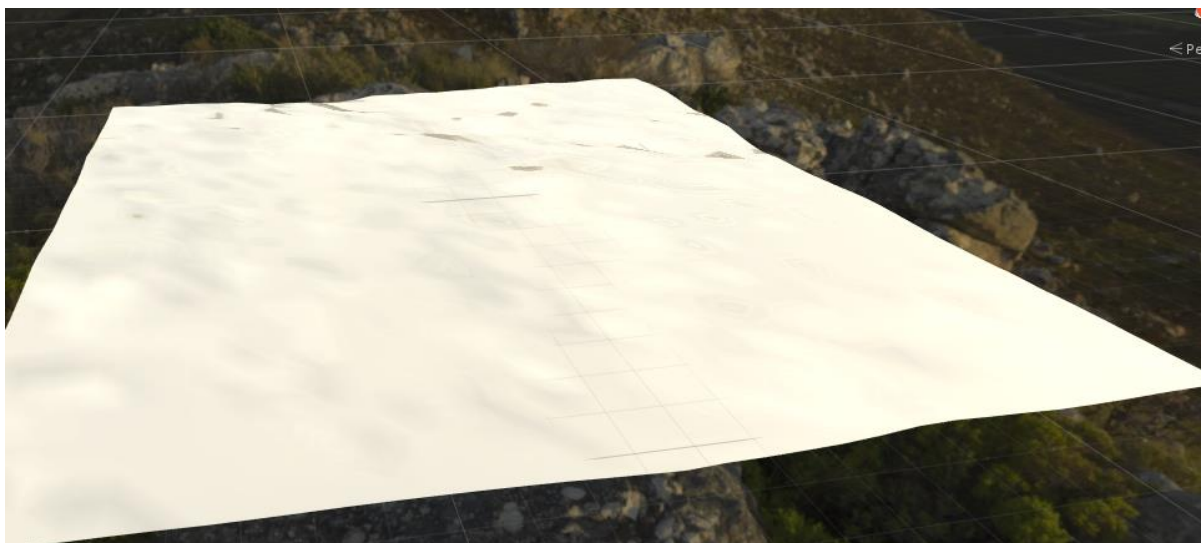
Figur 50: Resultat av første forsøk på å importere heightmap med kun GDAL til Unity. Det er små høydeforskjeller, selv om y-aksen er skalert opp til maks(10000 Unity units)

Forsøk 2

Andre forsøk er GTIFF brukt som filformat istedenfor ENVI, vist i Figur 51 som en kommando i GDAL.

```
B:\bachelor_data\rawkartdata\dtm1\resampled_data>gdal_translate -ot UInt16 -scale 170.819 818.859 -of GTiff scaled_dtm1_upper_left.tif terrain_upper_left1.raw
```

Figur 51: Andre forsøk på å konvertere et 32-bit heigmtap i GeoTIFF formatl til RAW i GDAL



Figur 52: Andre forsøk på å importere heightmap i Unity, resultatet er dårligere enn i første forsøk. I importeringen til Unity for en de samme verdier som i Figur 49, men Figur 52 viser et dårligere resultat enn i forsøk 1.

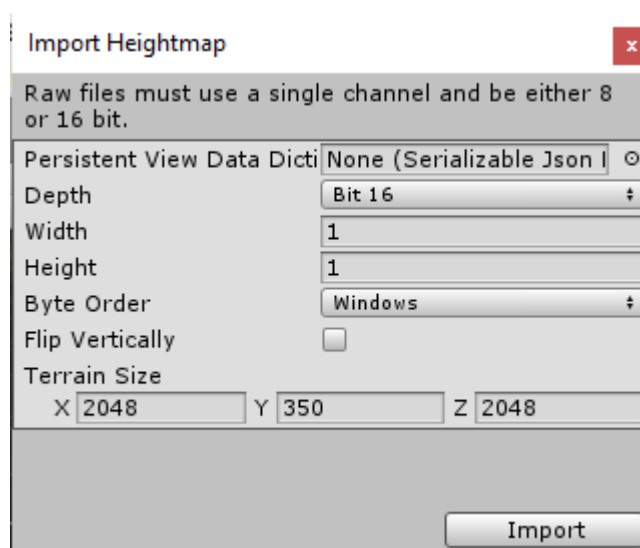
Forsøk 3

I dette forsøket har konverteringen fra 32 til 16-bit blitt gjort separat fra konverteringen til RAW. Det har blitt gjort 2 forsøk, et der konverteringen til 16-bit har skjedd i QGIS, og et i GDAL. Deretter har det blitt konvertert til RAW, som er kjørt i Figur 53.

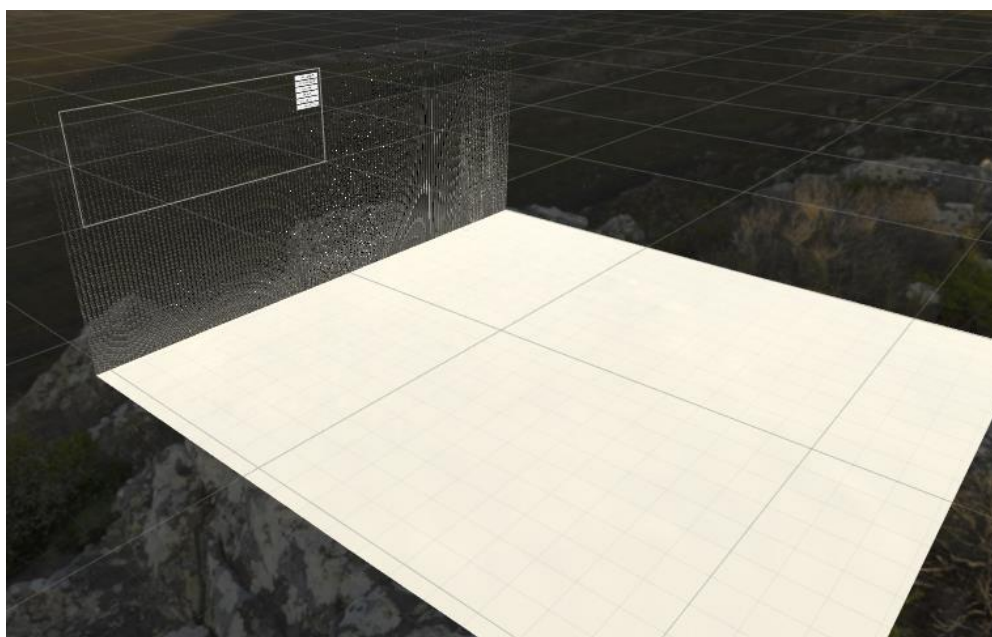
```
B:\bachelor_data\rawkartdata\dtm1\resampled_data>gdal_translate -of ENVI terrain_upper_left3.tif terrain_upper_left2.raw
```

Figur 53: tredje forsøk på å konvertere GeoTIFF fil til RAW i GDAL. I denne kommandoen er filen ikke konvertert til 16-bit.

Verdiene blir forskjellig fra forsøk 1 og 2, men det ser ut som Unity ikke kan registrere RAW filen sin høyde og bredde, som vist i Figur 54. I tillegg er Y-aksen skalert ned til 350 Unity units.



Figur 54: tredje forsøk på å importere høydedata i Unity. Instilling for å importere terreng registrerer ikke RAW filen sine dimensjoner.

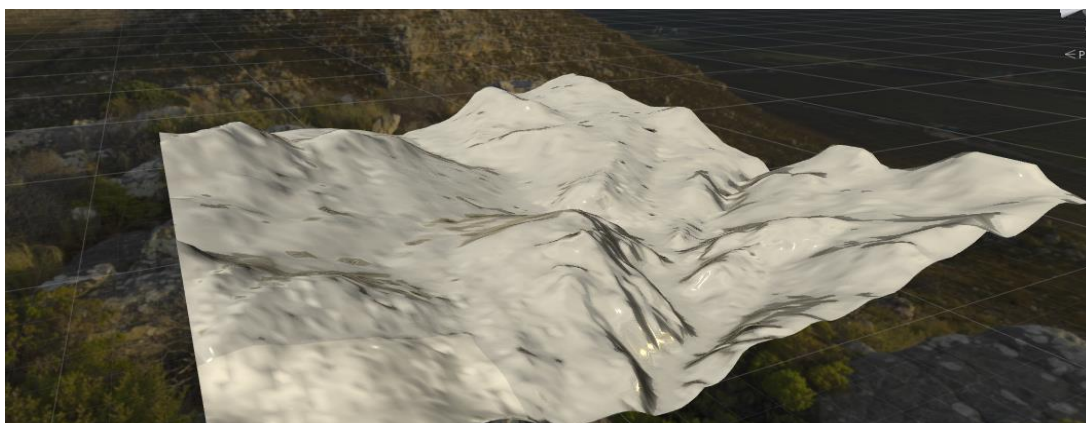


Figur 55: Resultat av tredje forsøk på å importere heightmap i Unity som viser et korrupt terreng.

Resultatet i forsøk 3 dårligere enn forsøk 1 i Figur 50 og 2 i Figur 52. I Figur 55 ser det ut som terrenget har blitt korrupt i konverterings fasen.

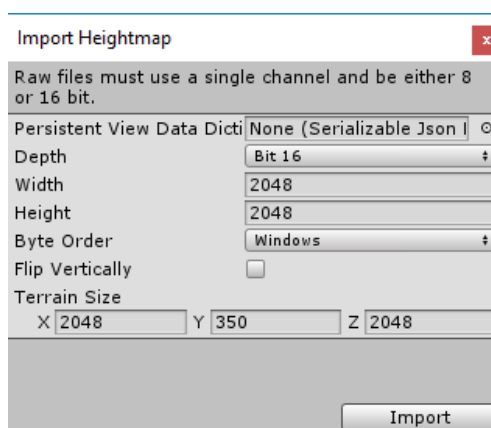
Forsøk 4

I det siste forsøket har GeoTIFF filene blitt konvertert fra 32 til 16-bit både i QGIS og GDAL separat, mens konverteringen til RAW har blitt gjort i Adobe Photoshop. Det har ikke blitt gjort noe med skaleringen av pixel verdiene. Figur 57 viser hvordan Unity reagerer på RAW filen når den blir importert. Y-aksen er satt til 350, kontra 10000 som måtte til i forsøk 1 og 2 for få et resultat. Resultatet i forsøk 4 er mye bedre, i Figur 56 ser det ut som et terreng med ulike høydekurver når det blir importert. Det er forsøk 4 på å importere terreng i Unity, ved bruk av Photoshop har gitt et bra resultat.



Figur 56: fjerde forsøk på å importere heightmap i Unity, ved bruk av Photoshop i siste fase til å konvertere til RAW har gitt det beste resultatet

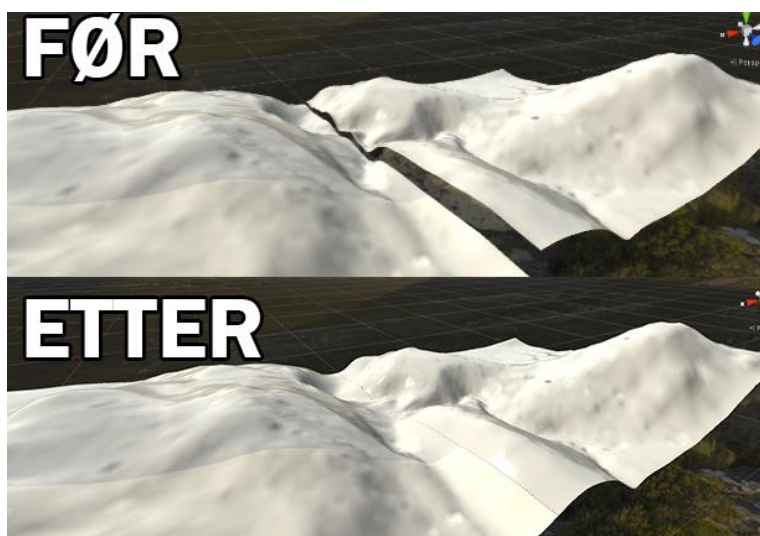
Dette konkluderer med at fremgangsmåten i forsøk 4 gir det beste resultatet. Selv om gruppen har blitt frarådet å bruke Photoshop, er det egnet som gir best resultat i Unity.



Figur 57: fjerde forsøk på importere heightmap i Unity. Konvertering av bits er forsøkt gjort i GDAL og QGIS med samme resultat, mens konvertering til RAW er utført i Photoshop.

5.4.3. Terrain Stitcher

For å sette sammen terrenmodellene i Unity benyttes Terrain Stitcher, en plugin fra Unity Asset store. For at Terrain stitcher skal fungere må terrenmodeller ligge på lik linje på y-aksen, ellers gjør pluginen jobben selv. Det finnes et par alternativer for hvordan terrenget skal slås sammen, for eksempel hvor jevnt sammenslåingen skal være. Figur 58 viser hvordan et resultat av å slå sammen terrenget med Terrain Stitcher.



Figur 58: Terrain Stitcher asset stifter sammen terrenget i Unity

5.4.4. Område

Området til prosjekt Hessdalen er delt inn i koordinatene som vises under. Figur 59 er teksturen som er lagt på terrenget i 3D kartet.

- Øst min – 296571, Øst maks – 308861
- Nord min – 6970689, Nord maks – 6966591

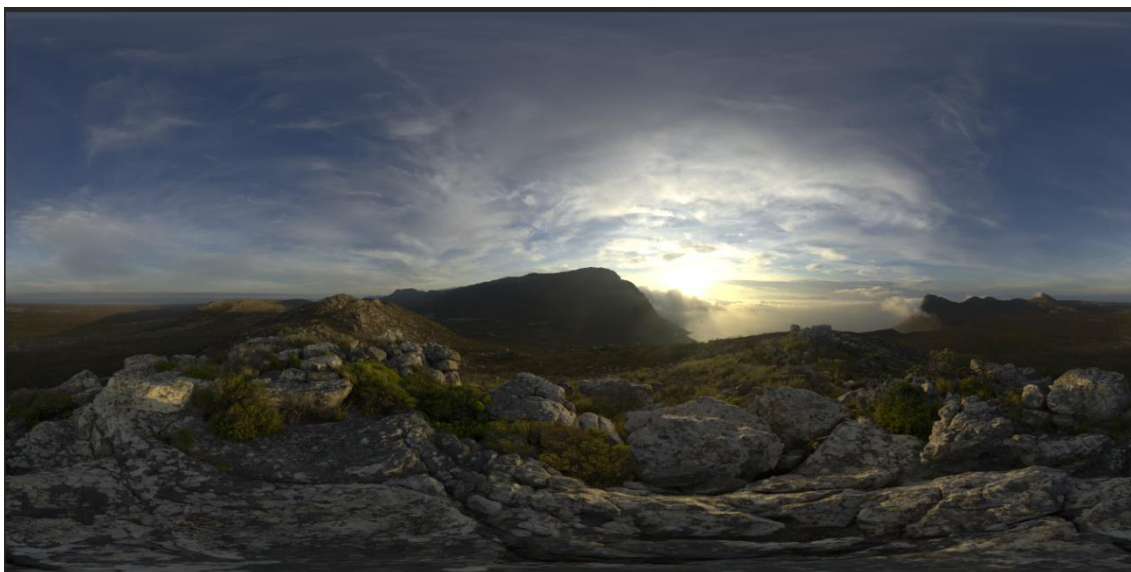


Figur 59: Tekstur og område over Hessdalen som brukes i 3D kartet

5.5. Skybox

Skybox er en metode for å lage bakgrunner som brer seg rundt en scene for å skape en illusjon at en verden er større enn den egentlig er⁵³. Som regel brukes HDR bilder som er formet som en boks. Med en himmel, bygninger og fjell fra lang distanse og noen objekter som ikke kan nåes er prosjektert inn i bildet sine flater benyttes en teknikk som kalles cube mapping. Figur 60 viser bilde som brukes til skybox til dette prosjektet.

⁵³ [https://en.wikipedia.org/wiki/Skybox_\(video_games\)](https://en.wikipedia.org/wiki/Skybox_(video_games))



Figur 60: HDR bilde som benyttes til Skybox i prosjektet

5.6. Importere 3D modeller

6. Brukertester

Formålet for brukertestene i dette prosjektet er å skreddersy hva opplevelsen vil være for brukeren når de bruker 3D kartet over Hessdalen. Med hjelp fra brukeren, får vi muligheten til å utvikle kartet fra et perspektiv som er annerledes enn våre egne. Vi gjennomgikk to omganger med brukertester; den første omgangen dreiet seg om opplevelsen av kontrollen på kameraet og bevegelsene over kartet. Dette var viktig å få på plass tidlig for å vite hvilken retning vi burde fokusere på når vi designet UI elementene og mekanikken på bevegelsene. Den andre brukertesten hadde fokus på innholdet av kartet og helheten på opplevelsen. Dette innebærer plassering av bygninger, navneskilt, informasjonsområder, teksturer og generell opplevelse.

6.1. Problemstilling

Problemstillingen i brukertestene dreier seg om en normal brukers opplevelser med kartet. Fokuset på testene er å finne ut hva brukeren synes er negativt eller positivt med produktet, begrunnelse over hvorfor resultatet ble påpekt i deres øyne, samt forbedringspotensialet. Produktet skal være brukervennlig i et bredt spektrum av alder og IT kunnskaper, noe som gjør at problemstillingen vær vil bli stilt til en bred aldersgruppe.

6.2. Metode

Metoden vi brukte under brukertestene av 3D-kartet var en observatørrolle med en veiledende siderolle. Brukertestene vil ha et kvalitets fokusert tilnærming, hvor testpersonenes personlige meninger om produktet vil bli oppveiet mot hverandre. Vi har valgt denne metoden for prosjektet siden det vil gi oss best mulig tilbakemelding om produktet og hvordan den oppleves hos testpersonene. Prosjekt medlemmene vil være observatører under testingen, Observatørrollen vil innebære dokumentasjon under testingen, hvor prosjektmedlemmene vil dokumentere deres reaksjon, tanker og meninger om produktet, men også om hvordan oppgavene gikk å utføre for testpersonene. Observatører vil også være litt til hjelp under testingen for at testpersonene ikke vil møte på problematikk som stopper progresjonen i for lang tid uten å komme seg videre. Vi valgte denne metoden i stedet for et nettbasert test skjema for å samle mer subjektivt informasjon angående produktet.

6.3. Brukerprofil

Brukergruppen for prosjekt Hessdalen er mangfoldig og utbredt. På grunn av dette så vil brukertestene gå i bredden istede for dybden. Vi bestemte oss for å teste seks personer i både brukertest nummer 1 og nummer 2. Testpersonene er i alder 20 til 80 år og forskjellige nivåer med datakunnskaper. Dette mener vi vil gi oss en god mengde informasjon om hva 3D kartet må inneholde og gruppen representerer brukergruppen som kommer til å bruke den på Prosjektets hjemmeside.

Vår kriterier for brukertestene var:

1. Brukerne som skulle teste produktet burde tilhøre allmenn gruppen av data kunnskaper.
2. Gruppen testere burde være 50/50 menn og kvinner.

6.4. Miljø og krav

Brukertesten må gjennomgås i et kontrollert miljø. Et medlem av gruppen må være tilstede som observatør for å kunne gjøre brukertesten. Miljøet som testene ble gjennomgått må være med minst mulig forstyrrelser. Dette vil si ikke noen eksterne lyder, skjermer, vinduer eller annet i samme retning som brukeren sitter mens de gjør testen. Kravet til brukertestene er at observatøren skal gå gjennom prosessen, og forklare stegene i testen, og etter at testen er ferdig, dokumentere eventuelle tilleggs tanker angående kartet.

6.5. Brukertest 1

6.5.1. Innledningsspørsmål

Vi hadde 3 innledningsspørsmål under brukertesten. Disse spørsmålene var for å definere hvem testpersonene er, og i henhold til brukergruppene, hvem testpersonen hører til.

Disse spørsmålene er:

1. Kjønn?
2. Alder?
3. Datakunnskaper?

6.5.2. Oppgaver

Oppgavene var basert på å teste ut kontrollene av 3D kartet i brukertest nummer en. På grunn av dette så var spørsmålene rettet mot en gjennomgang av bruksområdene på knappene.

Kontroll egenskapene som vi testet i brukertest 1 var:

1. Kontroller kameraet i en sirkel.
2. Beveg kameraet fram og tilbake
3. Se opp
4. Se ned
5. Bruk hastighets knappen
6. Bruk drop down menyen til å teleportere deg.
7. Finn en bygning.

6.5.3. Oppfølgingsspørsmål

1. Hva var bra?
2. Hva var dårlig?
3. Forbedringer

6.5.4. Utførelse

- Observatøren gikk først igjennom hvordan brukertesten skal gjennomføres, de ulike spørsmålene som vil bli stilt og oppgavene som skal bli gjort. Etter at dette ble gjort, så ble brukeren spurt om det var noen spørsmål angående testen og spørsmålene, om det var noen meninger om testen i seg selv, og om de var enige i hvordan testen skulle bli utført.

- Neste steg i brukertesten var at observatøren startet skjermen og gjorde klar programmet.
- Innledning Spørsmålene ble stilt til brukeren.
- Observatøren begynner å gi spørsmålene, etter hvert steg som blir gjort, avkrysser observatøren boksene på skjemaet.
- Etter oppgavene har blitt gjort, blir oppfølgingsspørsmålene stilt. Mens brukeren forklarte både positive og negative ting med kartet, skriver observatøren ned tilbakemeldingen.
- Testen blir avsluttet med at observatøren skrur av skjerm og program.

6.6. Brukertest 2

Innledningsspørsmål

Vi beholdt de første innledende spørsmålene på brukertest versjon 2 som i versjon 1. Vi ville oppnå en lik forståelse basert på alder og datakunnskaper som fra brukertest 1.

Disse spørsmålene er:

1. Kjønn?
2. Alder?
3. Data Kunnskaper?

Oppgaver

Kontroller testen

Oppgavene var basert på å teste ut kontrollene av 3D kartet i brukertest nummer en. På grunn av dette så var spørsmålene rettet mot en gjennomgang av bruksområdene på knappene.

Kontroll egenskapene som vi testet i brukertest 2 var:

1. Kontroller kameraet i en sirkel.
2. Beveg kameraet fram og tilbake
3. Se opp
4. Se ned
5. Bruk hastighets knappen
6. Bruk drop down menyen til å teleportere deg.
7. Finn en bygning.

Laptop test

I denne delen av brukertesten så ville vi finne ut av hvordan brukerens opplevelse ville være om en laptop uten noe ekstra utstyr ville være. Denne funksjonaliteten var veldig viktig å finne ut av på grunn av majoriteten av brukerne som ville bruke dette kartet, ville mest sannsynlig ikke ha en ekstern mus for lettere kontroll.

Vi gjorde en simpel test for å finne ut om kontrollene var brukbare og ba brukeren om å gjøre en simpel gjennomgang av kontrollene til kartet, og finne ut om det var noen problemer med dette.

Spørsmålene var som følger:

1. Beveg kameraet rundt med laptop touch Paden.
2. Fungerer dette?

Dette gav oss litt åpenhet for at brukeren skulle eksperimentere litt. Vi følte at dette ville gi oss det beste svaret på om en tilfeldig bruker kunne teste seg fram til en løsning på en laptop.

UI testen

I denne seksjonen av brukertesten så ville vi finne ut mer spesifikt, hvordan så interfacen som brukeren skulle håndtere til en hver tid ut. Det er viktig at all informasjon de får til seg er tydelig og lett leselig, så vi ville gi de korte spørsmål som var åpne for kommentarer og reaksjoner.

Oppgavene var som følger:

1. Finn kompasset
2. Finn miniatyr kartet.

Kart utforskning

I denne delen av brukertesten så ville vi teste funksjonaliteten med kartet i sin helhet. Vi ville se om brukeren kunne manøvrere seg frem til forskjellige viktige bygninger og områder uten problem.

Spørsmålene vi brukte var:

1. Finn Hessdalen Krike
2. Finn UFO camp
3. Finn Røde kors hytten
4. Finn Bekken Gård
5. Finn det høyeste fjellet

Oppfølgingsoppgaver

Oppfølgingene vi hadde under denne brukertesten var for hver oppgave del. Vi gjorde dette for å få mer presis tilbakemeldinger på de forskjellige seksjonene. Etter hver oppgave seksjon så spurte vi om hvordan brukeren syntes at disse delene var og hvordan prosessen gikk.

I slutfasen av brukertesten så spurte vi om helhetsinntrykket av kartet for brukeren, positive og negative deler med den, og forbedringer som brukeren ville se.

Disse spørsmålene var:

1. Syntes brukeren at kartet var bra eller dårlig?
2. Hvordan var helhetsinntrykket av kartet?
3. Har brukeren noen spesifikke klager?
4. Forbedringer som brukeren synes trengs?

Utførelse

1. Observatøren gikk først igjennom hvordan brukertesten skal gjennomføres, de ulike spørsmålene som vil bli stilt og oppgavene som skal bli gjort. Etter at dette ble gjort, så ble brukeren spurt om det var noen spørsmål angående testen og spørsmålene, om det var noen meninger om testen i seg selv, og om de var enige i hvordan testen skulle bli utført.
2. Neste steg i brukertesten var at observatøren startet skjermen og gjorde klar programmet.
3. Innledning Spørsmålene ble stilt til brukeren.
4. Observatøren begynner å gi spørsmålene, etter hvert steg som blir gjort, avkrysser observatøren boksene på skjemaet.
5. Etter oppgavene har blitt gjort, blir oppfølgingsspørsmålene stilt. Mens brukeren forklarte både positive og negative ting med kartet, skriver observatøren ned tilbakemeldingen.
6. Testen blir avsluttet med at observatøren skrur av skjerm og program.

7. Diskusjon

Det har blitt diskutert en del mellom både gruppen internt, med arbeidsgiver og veileder om størrelsen på 3D modellerte bygninger. Prosjektgruppen mener at bygninger som skiller seg ut i forhold til vanlige hus burde kunne være skalert opp så brukere kan få en positivt inntrykk av attraksjonene. Brukertestene har også gitt tilbakemelding på at spesielle attraksjoner kunne vært større, men vi forstår veileder og arbeidsgiver sitt svar på hvorfor de burde være i realistisk størrelse i forhold til kartet. I teksten kan en se hvor bygninger ligger og hvor store de er, det ville blitt litt rart hvis de fleste bygningene skal være realistiske, mens noen er mye større. På en annen side er det ingen "vanlige" 3D modellerte hus på kartet. Med begrensningene WebGL kommer med har gruppen gjort noen tanker om en nedlastbar versjon til Windows og MacOS skal være tilgjengelig. Argumentene for det er et alternativ til de med trege maskiner eller dårlig internett slik at de kan få den beste opplevelsen ved å kjøre 3D kartet i et eget program.

7.1. Oppgave mål

Gruppens mål fra begynnelsen av bachelor oppgaven var å lage et 3D kart over Hessdalen som fremviste området geografi og skape ønsker om å utforske der både personlig, og gjennom et 3D kart. Gruppen føler at vi har nådd dette målet. vi har laget et ganske nøyaktig representasjon av Hessdalen, både i form av området rundt, og bygdekulturen. det er lite informasjon på selve

kartet, men i henhold til nettsidens informasjon og kartet som et hjelpemiddel, så mener vi at det gir et godt grunnlag for å starte utforskelsen for en bruker.

Vi føler også at vi har klart å lage en opplevelse som besøkende kan utforske, for å lære om hvordan Hessdalen er. Vi fikk ikke helt til å lage alle målene, men som en begynnelse, så er dette kartet et brukbart punkt som kan bli videre utviklet av arbeidsgiver til en senere tid. Vi føler at dette kan bli etterarbeidet av prosjekt Hessdalen, og siden hjelper besøkende med litt informasjon.

7.2. Levering

Prosjektgruppen hadde faste leveringer gjennom hele prosjektperioden. disse leveransene var deler av prosjektet som for det meste skulle bli satt sammen til hoveddokumentet. Hoved leveransene i prosjektet var Forprosjekt rapporten, Hoveddokument versjon 1 og 2, og til slutt hoveddokument leveransen med refleksjonsnotater. Med unntak av disse hoved leveransene, hadde gruppen under ukentlige møter satt arbeidskrav for hverandre, og samarbeidet på mindre leveranser som var viktig for oppbyggingen. Disse leveransene var kontrakter og hjemmeside, mens ukentlige arbeidsoppgaver var oppbyggingen til hoveddokumentet. Gruppen klarte å håndtere enhver leveranse med god struktur og på tiden, samtidig som at vi klarte å levere delmålene som ble satt i mellom hoved leveransene.

Under enhver leveranse, så jobbet gruppen effektivt samlet for å skrive, finpusse og koordinere leveransen. Gruppen har også vært utfordrende med å komme med ideer om mulige forbedringer og løsninger til leveranser, og med å forbedre ytelsen som gruppen kan klare tilsammen.

7.3. Tekniske utfordringer

3D kartet består av totalt 9 komplekse terrengmodeller i high polygon mesh, der hver har en størrelse på 2048x2048px. I Unity er det satt av opptil 1600 mb minne(RAM) som skal tilføre kartet gjennom en nettleter, Det er veldig mye minne, men ellers ville nettleter gitt «Out of memory» feilmelding. Feilmeldingen kommer av at programmet får for lite minne til å kunne lastes inn. Standard minne innstilling i Unity WebGL er 256 mb. Hvis nettleteren (særlig Chrome) allerede bruker en del minne kan det gi både innlastning og minneproblemer for spesielt litt dårlig utstyrte og eldre maskiner. Innlastningstiden avhenger også av internett hastigheten til brukeren og kommunikasjon med serveren.

Typiske problemer kan være:

- Treg innlastningstid, på grunn av dårlig internett eller en dårlig maskin.
- Færre bilder per sekund(frame drop) når kamera er i bevegelse.

Et mindre typisk problem kan være at kartet ikke laster inn i det heletatt, om minne overstiger hva maskinen har tilgjengelig å fordele rundt.

Unity i seg selv har bydd på problemer i denne utviklingen. Ved å importere større terrenger krasjer programmet nærmest hele tiden om man foretar seg et valg. Det betyr at i prosessen av å importere terreng må man lagre ved hvert minste endringer som foretas. Det har blitt gjort forsøk og research på hvorfor Unity krasjer, men til ingen nytte. Per dags dato står det lite på Unity forumene som kan gjøres for å hindre at programmet krasjer så ofte.

7.4. Videre utvikling

Teleport-klikk funksjon

Gjennom brukertest feedback og gjennom gruppens egne meninger, så mangler kartet enda en teleporterings funksjon som teleporterer brukeren til et objekt gjennom et museklikk. Denne funksjonaliteten ble utforsket og forsøkt på, men på grunn av for liten tid så klarte ikke denne funksjonen å bli til en realitet. En tilleggsdetalj kan være at brukeren flyr til et punkt, når det velges en destinasjon fra nedtrekksmenyen, istedenfor at en blir teleportert. På den måten kan brukeren gjøre seg litt orientert i område når en ser hvor man er på vei.

Instillings meny

Gruppen mener at produktet kunne forbedres med å ha mer visuell informasjon til brukeren, og med å tildele informasjon på en litt bedre måte. En innstillingsmeny, eller “options menu”. Med dette så vil vi kunne gi brukerne valgmuligheter til både å forandre innstillingene på kartet, men også en oppsummering av kontrollene og andre viktige innstillinger.

En mulig videreutvikling av 3D kartet ville være å gi flere valgmuligheter for brukeren som besøker systemet. Dette kan være innenfor valg av hvordan man kontrollerer kameraet, skreddersydd forandring av kontrollene som brukeren kan forandre etter egne behov(kjent som “custom key mapping”), vi hadde også tilbakemeldinger fra første brukertest på at speilvendte kontroller kunne også være nyttig for brukeren, hvis de følte at dette kunne håndteres bedre. Det førte til at en funksjon for å speilvende kamera rotasjonen(inverted) ble implementert.

Begynner tutorial/guide

En funksjon som kan hjelpe veldig mange av fremtidige brukere med å håndtere dette produktet og gjøre produktet tilgjengelig til et større antall mennesker er en form for introduksjon eller “tutorial” som brukeren kan velge å få når de starter opp kartet. Dette kan hjelpe eldre mennesker/glemske brukeren med å effektivt lære seg kartet, og å gi en hjelpende hånd til å forbedre brukeropplevelsen. Dette tutorialen tenkte vi at kunne ha en veldig simpel introduksjon med en spørsmål om brukeren vil ha litt hjelp eller ikke med å komme i gang.

Større og mer levende verden

For å gjøre kartet mer levende kan fremtidig utvikling være å implementere flere 3D objekter som bygninger, vegetasjon, værforandringer og visuelle fremstillinger av lysfenomenet. Ulempen er at programmet vil kreve mer minne og lagring for å kunne kjøres, men 3D modellene i det nåværende programmet tar minimalt med plass i forhold til terrenget.

Til å begynne med skulle kartet omfattes et større område av Holtålen kommune som Hessdalen befinner seg i. Det gikk helt fint med et mål der hver pixel er 10x10 meter, men ble nedskalert til kun Hessdalen da gruppen økte til 1x1 meter per pixel. Å bruke en større del av Holtålen kommune kan være aktuelt dersom en får redusert maskinkraft og internett som kreves for å rendre programmet. Kartet har heller ingen synlig ramme som indikerer grenser for bevegelig område. Til videre utvikling kan det legges til et større område som ikke kan ferdes i, men som brukes til å en slags tåke som skaper en illusjon om at kartet fortsetter. Det ble gjort et enkelt forsøk, men prosjektgruppen mente det tok mer maskinkraft enn nødvendig.

Lysfenomen

Noe som var veldig lavt prioritert, arbeidsgiver ønsket heller ikke å ha med en visuell fremstilling av lysfenomenet på grunn av at det kan gi feil indikasjon til turister som vil oppsøke Hessdalen på grunn av fenomenet. Å legge til og utforske partikkelsystemer i Unity er et alternativ en kunne illustrert lysfenomenet på, ved eksempel å se på dokumenterte lys fra Hessdalen og prøve å gjenskape det sam

Mer informasjon

Bruker testene har gitt tilbakemelding på at mer informasjon om ulike steder kunne vært nyttig. Det kunne poppet opp over et objekt/sted og vist generell informasjon som år det ble skapt, historie og hva det er i dag.

7.5. Feil og mangler

Feil og mangler er ting som prosjektgruppen hovedsakelig hadde satt seg mål om å implementere og eller ferdigstille, men kom ikke helt i mål. En av tingene er kollidering med terreng, kamera er satt opp til å kunne bevege seg innenfor et spesifikt område. På grunn av hvordan kamera kontrollerer scriptet var satt opp klarte ikke gruppen å implementere en løsning for å hindre kamera å kunne fly gjennom terrenget. Det ble forsøkt å legge til fysikk på både kamera og terrenget til ingen nytte. Planen var også å implementere flere bygninger og vegetasjon for å gjøre kartet mer interaktivt, men på grunn av tid og maskinkraft limitasjoner blir kun de viktigste bygningene modellert.

8. Konklusjon

Gruppen føler at vi har nådd det viktigste målet, som var å lage et 3D kart over Hessdalen som fremviser områdets geografi, slik at brukere av nettstedet til Hessdalen kan utforske området gjennom et 3D kart. Ut ifra målene satt i kapittel 1.4, så har vi gjennomført hovedmålet og delmålene, utenom delmål nummer 3, som var mer informasjon om viktige bygninger og steder. Ettersom dette ikke var så viktig for arbeidsgiver ble det nedprioritert. Men ut ifra tilbakemeldinger på brukertester, kan det godt være noe for fremtidig arbeid. I stedet har gruppen fokusert på å lage kartet så godt som mulig, og navngi viktige bygninger og steder, som var viktigere for arbeidsgiver.

Annet som ikke ble gjort, og kan være muligheter å utforske for fremtidig arbeid er blant annet kollidering med terreng, som vi ikke fant en løsning på. Teleport-klikk funksjonen ble også forsøkt utført, men uten suksess, og kan utforskes ved videre arbeid. Ellers var innstillingsmeny og en begynner tutorial noe som vi hadde tenkt på, men ikke hatt tid til å forsøke oss på. Lysfenomenet og mer informasjon om viktige steder og bygninger var noe som vi nedprioriterte, men det er også mulig å utforske dette i videre arbeid. Det er også mulighet for å lage kartet for et større område enn bare Hessdalen, samt lage flere 3D objekter, som vanlige hus og vegetasjon. Men ettersom vi slet med at kartet allerede tar mye maskinkraft, så må enten kartet optimaliseres bedre, eller kjøres som et nedlastbart program.

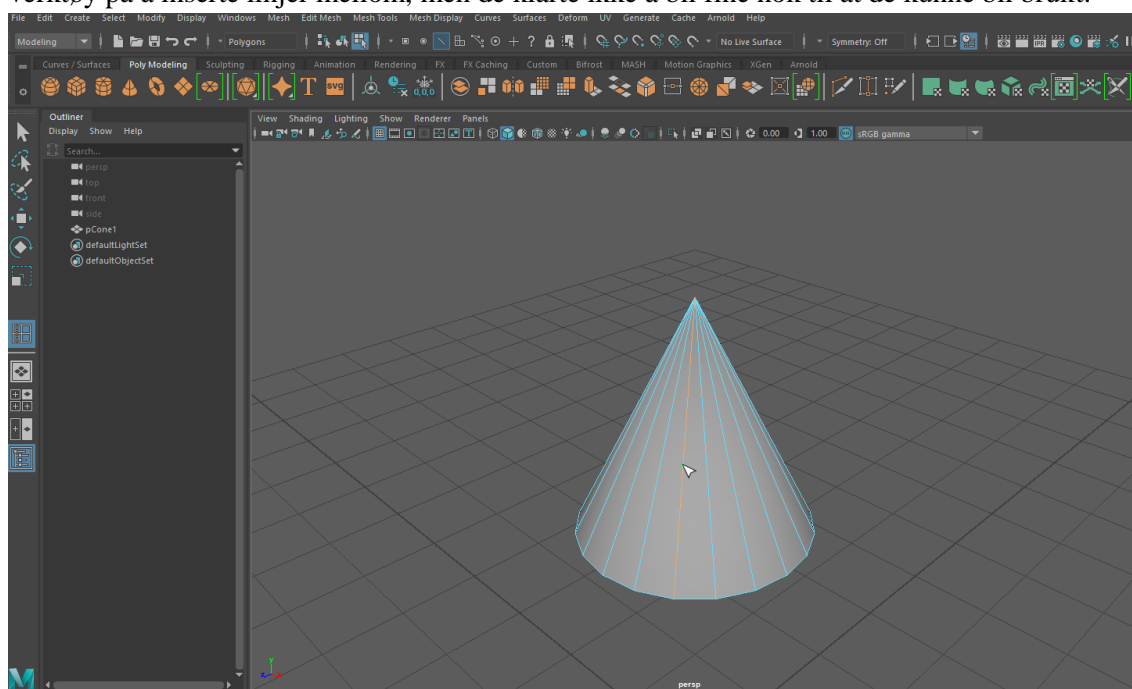
9. Vedlegg

9.1. 3D modellering

9.1.1. Hessdalen UFO camp

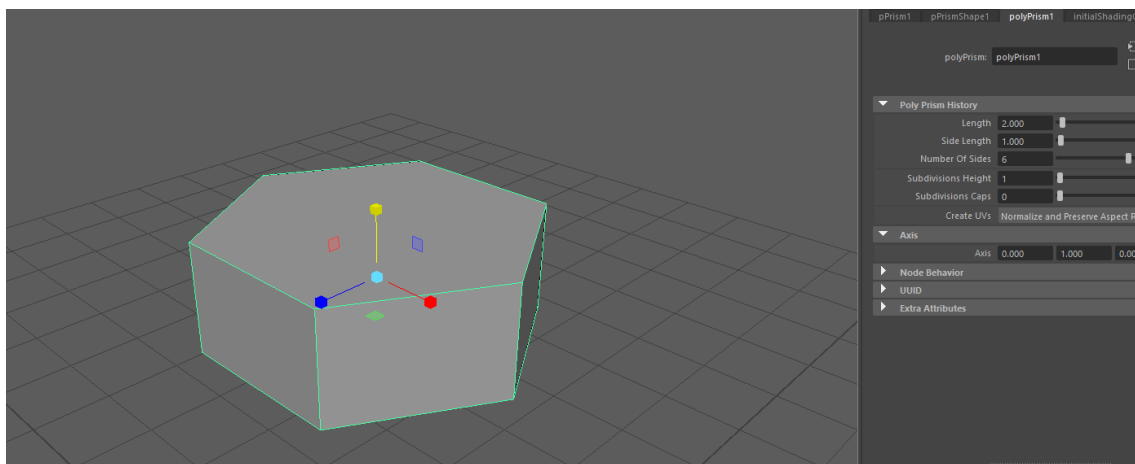
Begynnelsen av UFO campen var ganske lett å håndtere. Lagde en basis form og forandret på sidene for å få en viktig vinkel.

En cone i maya kan ikke bli redigert på faces veldig lett, og insert line tool fungerer ikke. Linjene blir satt på siden og under, og blir bare surrete. Det ble prøvd å eksperimentere med flere verktøy på å inserte linjer mellom, men de klarte ikke å bli fine nok til at de kunne bli brukt.



Figur 61: En uendret poly cone fra maya i kunn mesh.

Flere av polygon mesh kan til tider henge seg opp og med mindre man gjør ting på spesielle måter så har man ikke rådighet til å redigere disse
Sidene ble redigert på for å få den til å virke noenlunde riktig i form av bilder som vi hadde fra google.



Figur 62: Bunn strukturen av UFO campen, et heksagon i grunn mesh

Vi kombinerte den til en sylinder til å lage pipen, og la på en flat sylinder til på toppen for å dekke over pipen som på bildet. Ble en litt rar fasong på den på grunn av at programmet ikke fikk lov til å redigere på sidene for en eller annen grunn, men vi hadde brukt for mye tid på en liten detalj allerede.



Figur 63: toppen av UFO campen, "pipen" i full tekstur

Vi prøvde også en cone for å se om vi kunne manipulere den på en veldig lett måte. En av versjonene vi lagde var en cone på toppen men den klarte ikke å forandre den som vi ville ha det, så dette ble droppet tidlig.

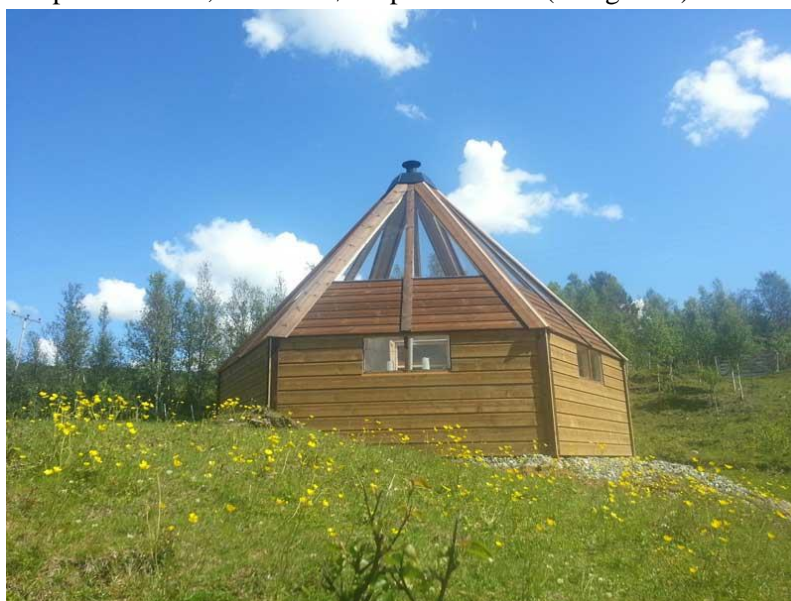
Taket ble laget av et manipulert sylinder som ble "extracted" sidene for å lage tre bjelkene på sidene. Dette gjorde vi ved å velge kanter og så den ble litt mer "kantete"



Figur 64: Toppen av inngangs partiet til UFO campen

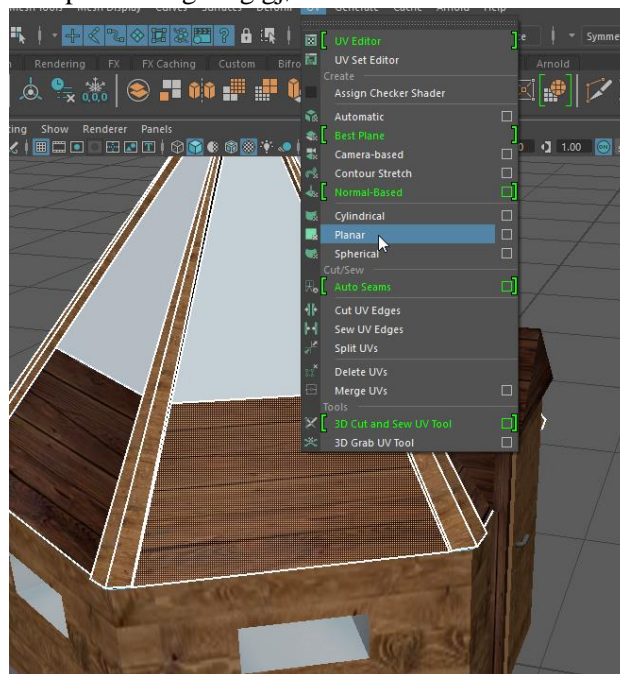
Vanskeligste delen av modellen var glasset, jeg satt ikke inn når jeg lagde formen først, så vi måtte finne en tool eller en måte å få glasset fint inn. Prøvde flere forsøk med insert line(samme som conen over skjedde såklart) og vi fikk ikke delt opp i trekantvinduer som det er på bildene(eksempel under) jeg prøvde å forme vinduene ut fra trekanter som var laget og klippet gjennom toppen av bygningen, for så å formes etter formene men det gikk ikke noe bra, og ble ekstra arbeid for ingen nytte.

Vi prøvde å gjøre research på problemet men endte til slutt med å bruke multicut til å lage vinduene manuelt på strukturen, noe som løste problemet ok(se figur 64).



Figur 65: Bilde av 1 UFO camp fra Hessdalen

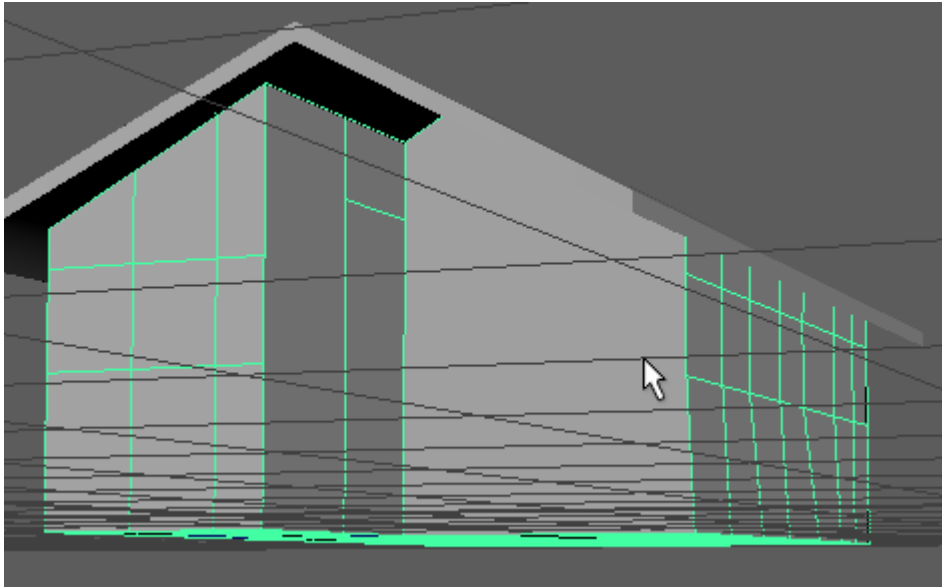
Teksturene54 kom etterpå. Fant disse på google så kan ikke si at det var mye arbeid, men slet litt med å finne riktig tre farge på hoved kroppen. Fant nærme nok farger og la de på. UV mappingen tok tid med finsettingen på grunn av problematikk med veggene, men vi klarte å mappe de relativt bra med planar valget og gjøre det manuelt 1 for 1.



Figur 66: Visning av det mest effektive valget som vi fant når det handlet om UV mappingen av veggene.

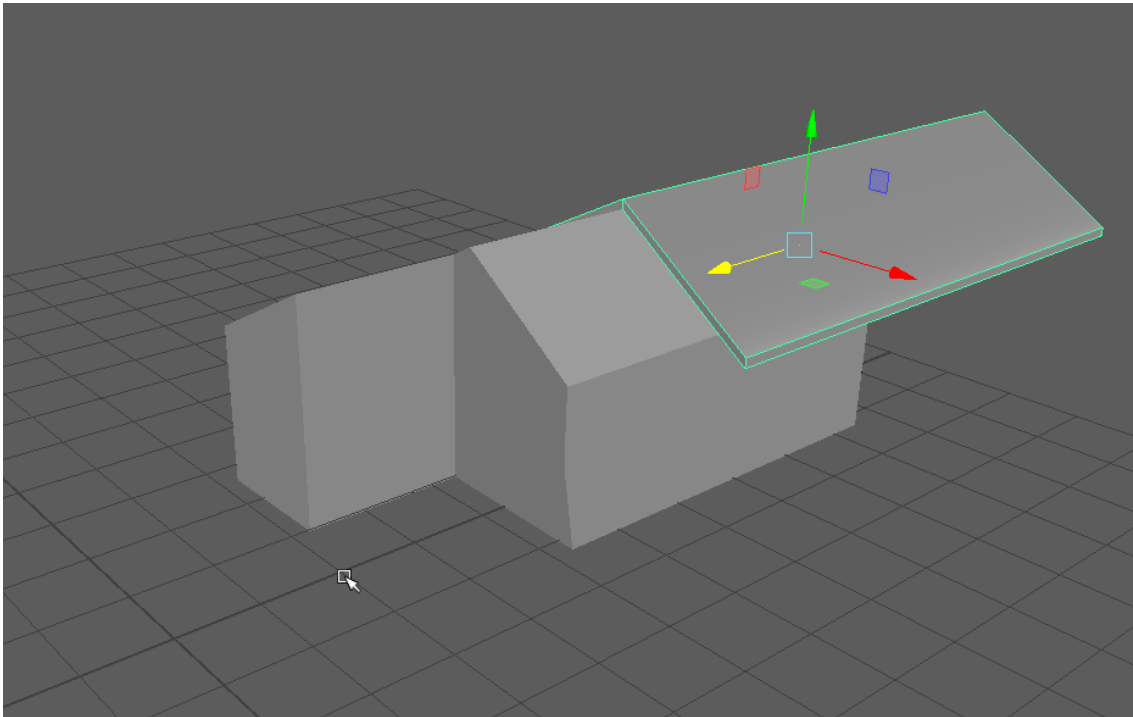
10.1.2 Røde kors hytten

Problemet med denne modellen var først og fremst taket. Utformingen av veggene og formen på huset var relativt greit, men å få skrå taket i boks med gode vegg proporsjoner var litt vanskelige. Første versjon var to bokser som ble satt nær hverandre, for å forme hovedbygget, og inngangspartiet. Dette ble ikke veldig pent, så den ble kastet. Andre versjon var en firkant blokk som ble ansiktsmanipulert til å ha et inngangsparti, dette ble laget med face removal og bridget det sammen til det ble utformet, fill hole og bridge er veldig spesifikk på hvordan den må bli laget så det ble mye krongle arbeid og masse feil.



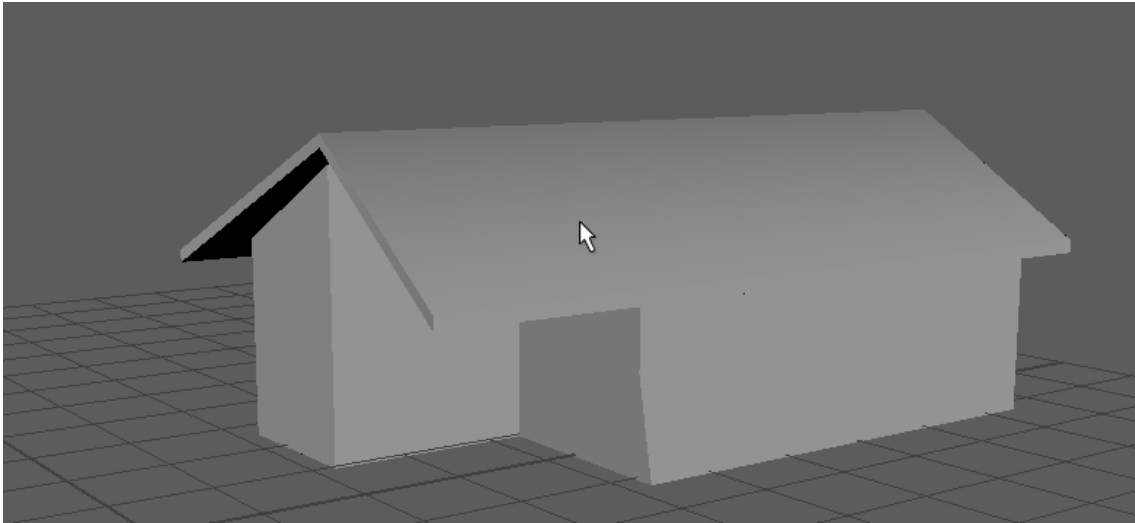
Figur 67: front inngangen i kun mesh form

Taket var laget med å bruke booleans, men dette ble ikke noe av før etter litt knot. Booleansene ville enten fjerne deler av taket på utsiden, eller ville fjerne hele bygget annet enn kantene. Jeg brukte differentiate etter å ha kopiert taket og flyttet den på plass etter over 30 forsøk.



Figur 68: plassering av taket på meshen.

Resultatet ble veldig glatt overgang mellom taket og hovedbygningen.

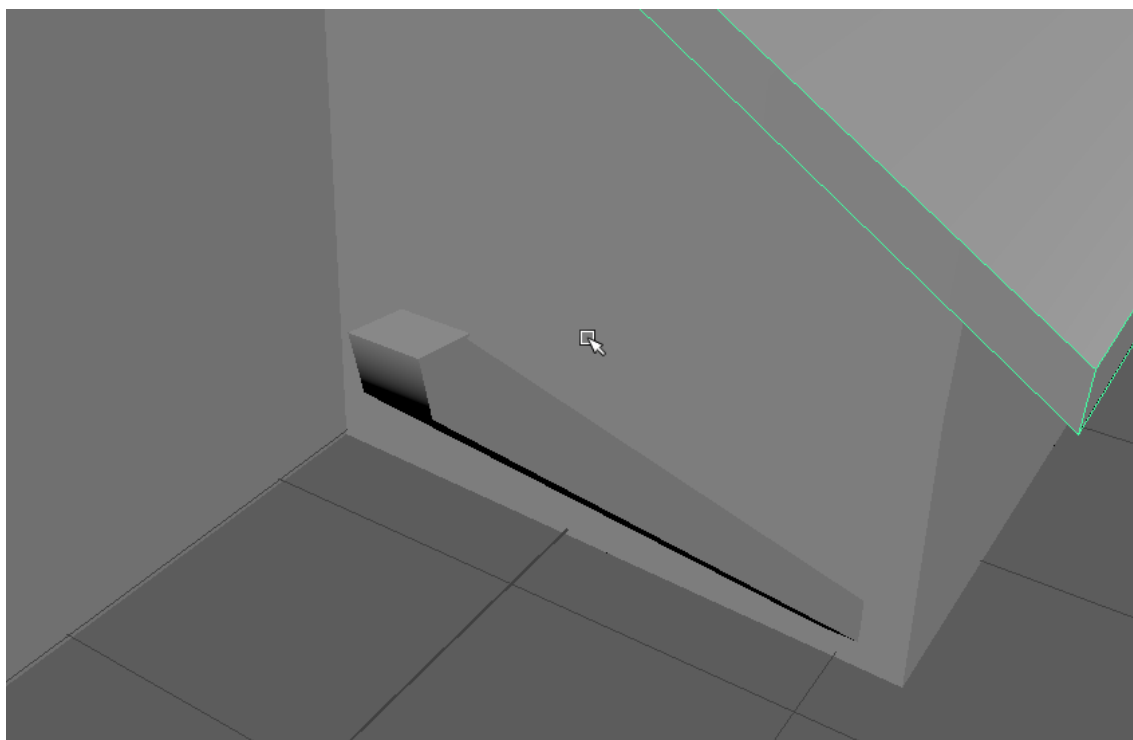


Figur 69: Tak utforming av hytten i kunn mesh



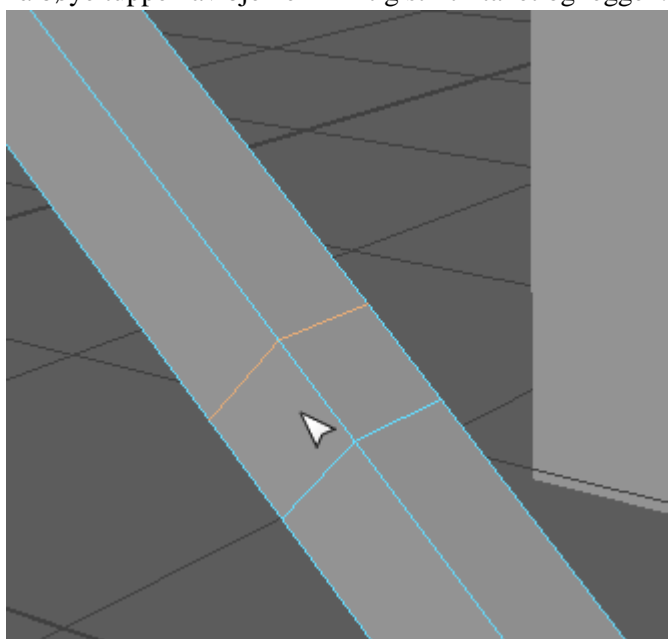
Figur 70: bilde av røde kors hytten i Hessdalen.55

55 Bilde av røde kors hytten tilsendt av Erling P. Strand

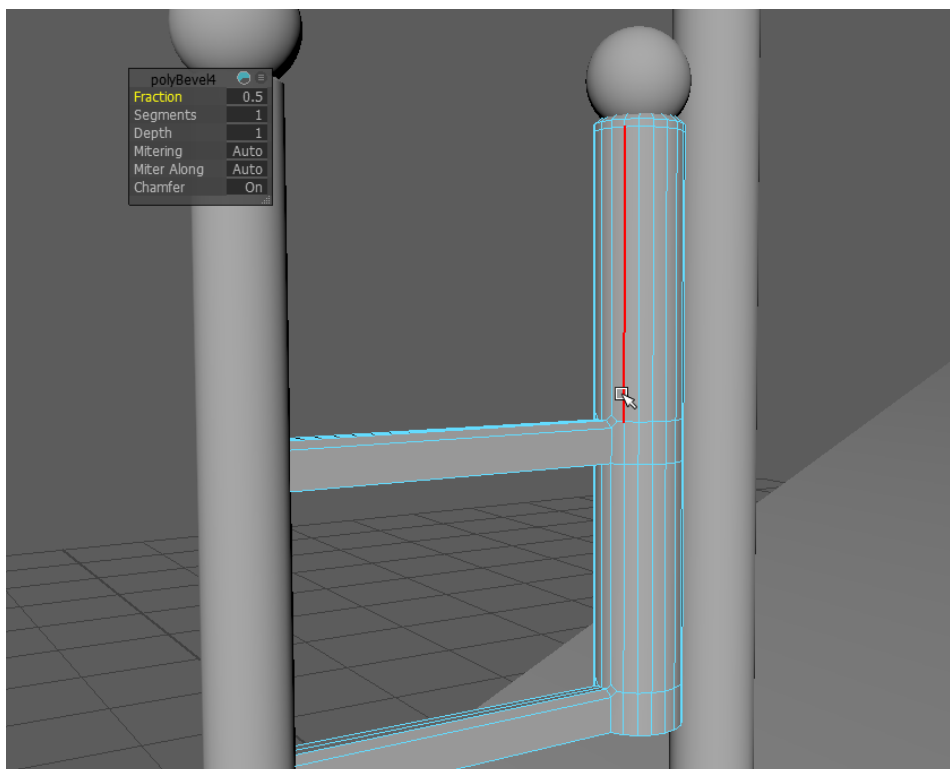


Figur 71: modellering av tak stolpen

Trebjerkene som kan sees på fronten av hytten ble laget med polygon firkant forandret og extrude verktøyet til å bøye tuppen av bjelken i riktig stil til taket og leggen.

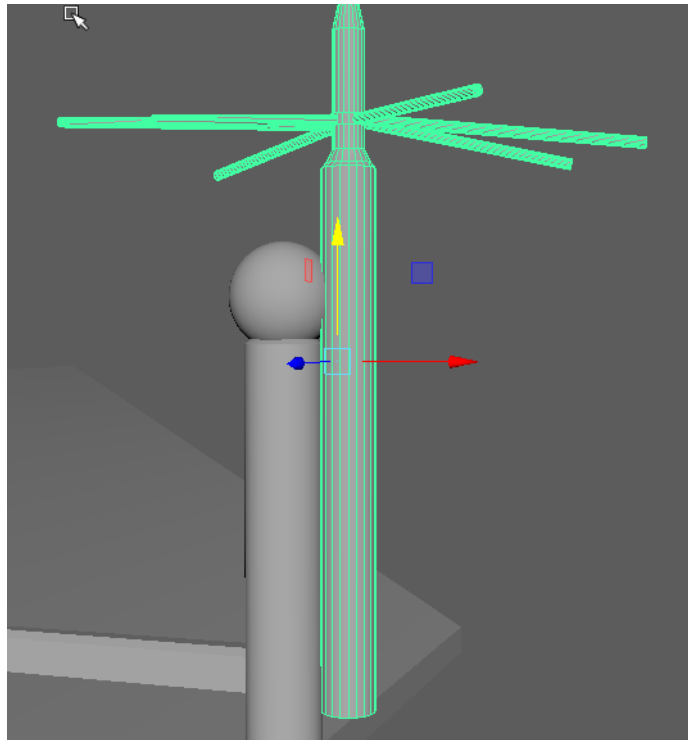


Figur 72: Tak stolpen bed inngangspartiet I kunn mesh
Støtte bjelken ble laget med insert edge tool og utformet på samme måte som bjelke nr 1.



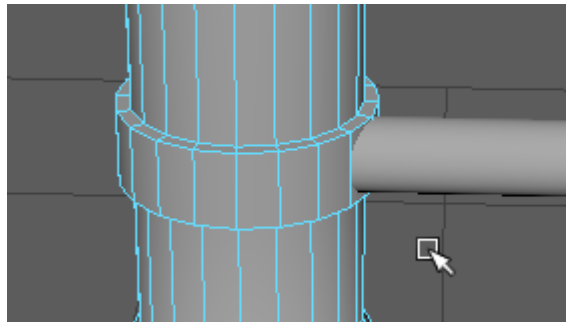
Figur 73: Modellering av stolen i strømstolpen

Antennen ble laget med å forme en sylinder i riktig høyde i forhold til huset. Jeg fant ikke noe dokumentasjon på antenne beinet så jeg lot den være default fra polygon modellen(bildene blir litt blokkert). Jeg utformet så trebenk beinet(eller stol, er ikke helt sikker) med å lage en sylinder til, strekke den i riktig lengde, extracte toppen til å ta imot beinenden som ble laget med en polygon ball. Jeg brukte insert edge loop redskapet til å lage linjene hvor det andre beinet skal være tilkoblet og brukte extrude verktøyet til å strekke ut overgangen til bein nummer to(bilde over).



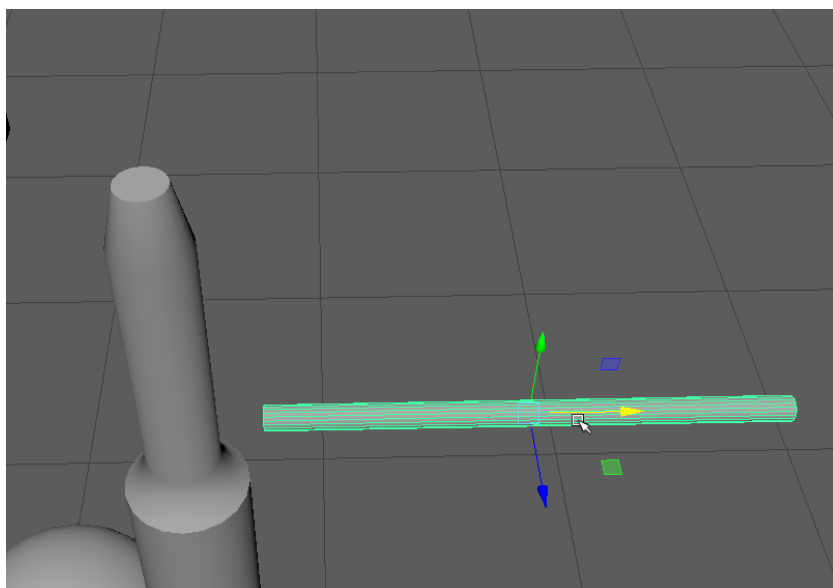
Figur 74: Antennen i kunn mesh

Antennen ble laget med en sylinder som jeg extrudert i toppen og formet etter bildet.

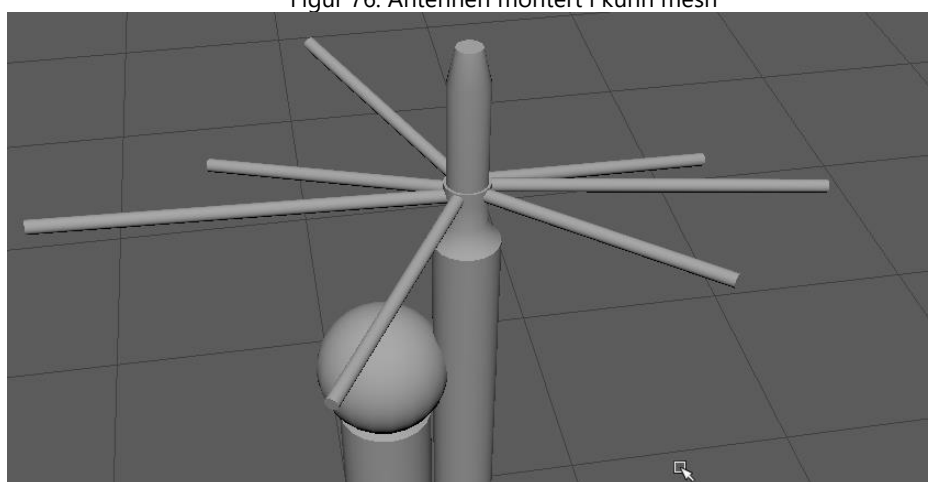


Figur 75: Modellering av koblingen til førerne kunn i mesh

Antennen ble festet på antenne stolpen med en extrudert sirkel rundt meshen for å antyde at den er festet til noe konkret.

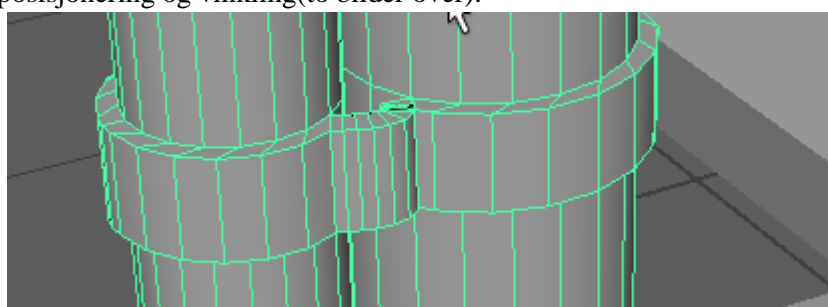


Figur 76: Antennen montert i kunn mesh

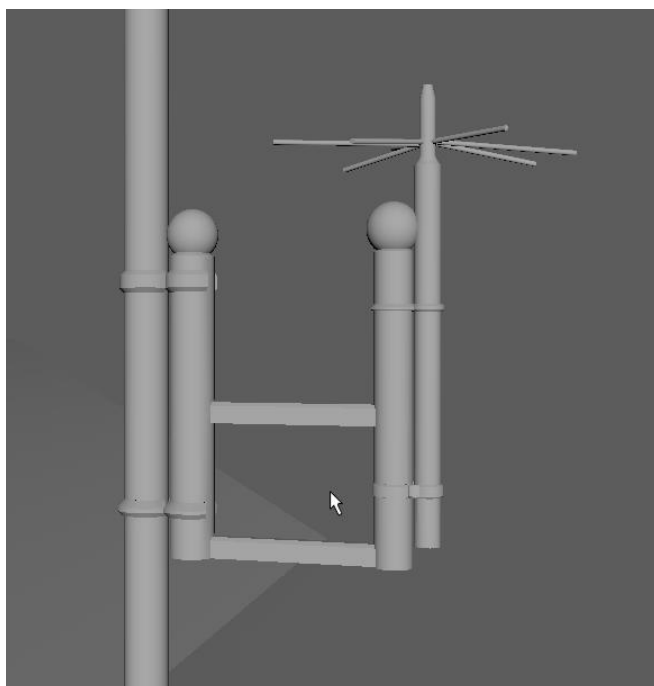


Figur 77: Strømstolpe kobling til stol kunn mesh

Antenne følerene ble laget av sylindere som så ble kopiert og vinklet med ctrl+D og roteringsverktøyet for å virke litt mer ekte. Disse ble plassert basert på bildet og hvordan de ser ut i forhold til posisjonering og vinkling(to bilder over).

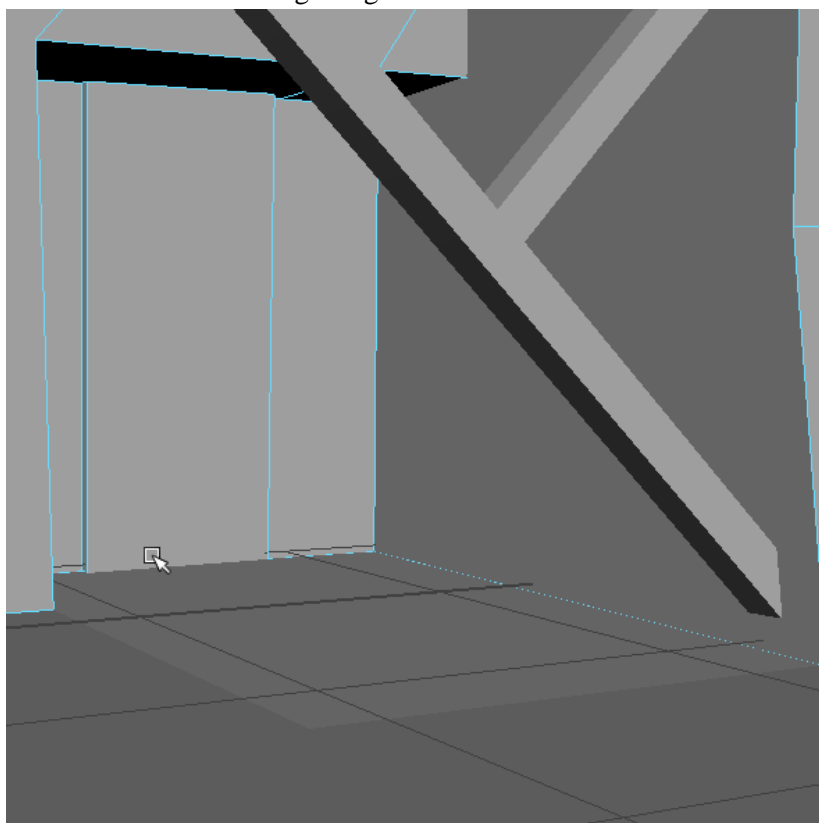


Figur 78: strømstolpe kobling til stol i kunn mesh



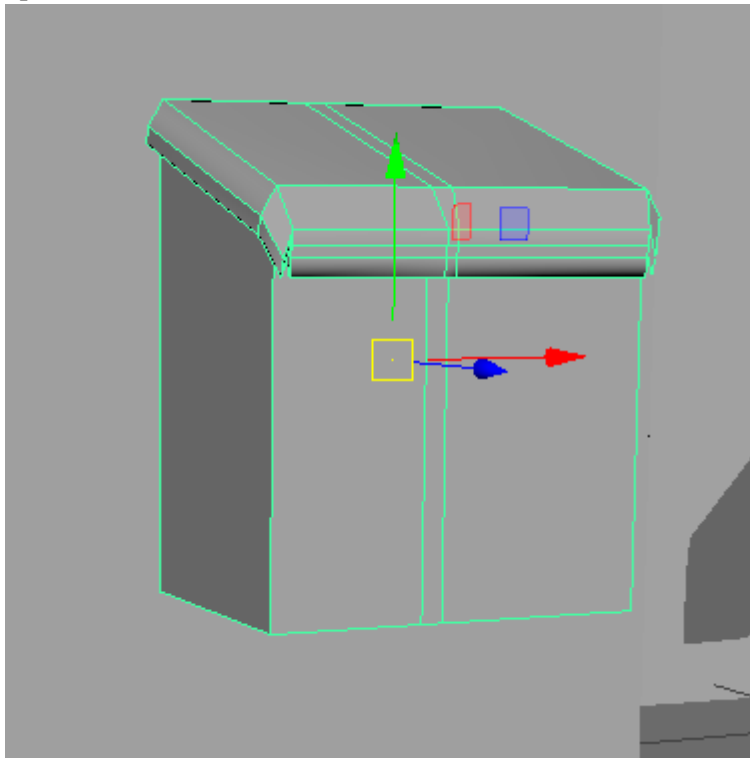
Figur 79: strømstolpen i kunn mesh form

Antennen ble koblet til i bildet med et eller annet snøring av noe slag. Jeg har prøvd å replikere dette med å extrudere kanter og bridge disse sammen som sett i bildene over.



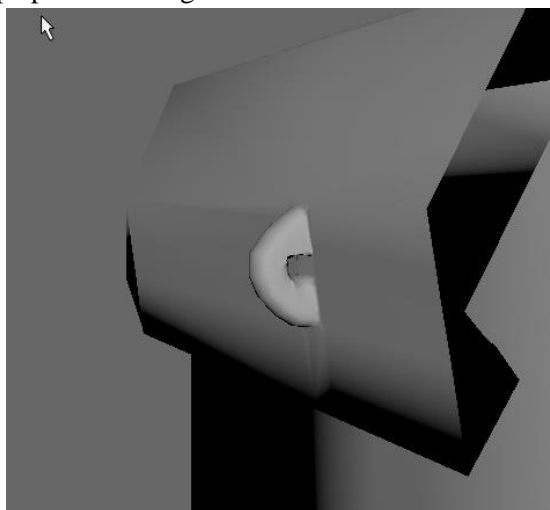
Figur 80: utformingen av front inngangen på hytten i kunn mesh.

Front Seksjonen ble laget med extrude utover for å lage døråpningen på hytten. veranda/gulvplaten foran døren ble laget av en simple overflate polygon. Døren ble så extrudert fra denne platen etter retting av linjene med multi cut verktøyet og bevegelse funksjonen til å sette linjene rett og på plass.

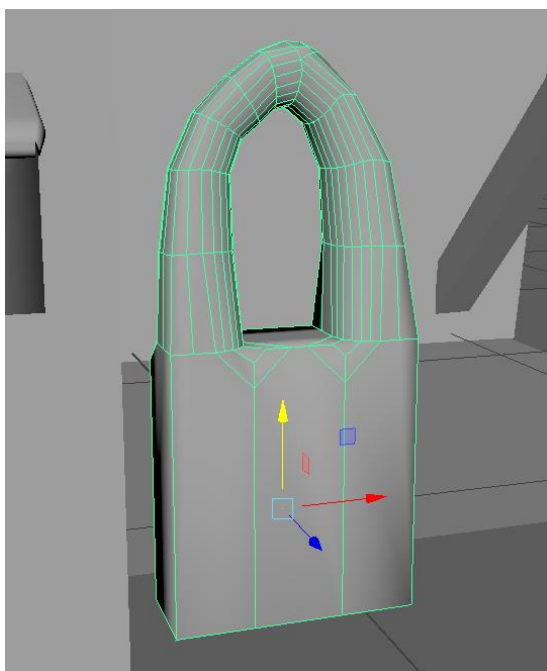


Figur 81: postkasse i kunn mesh

Postkassen ble laget med en simpel firkant polygon, etterfulgt av extrude på toppen og sidene for å lage lokket på postkassen og kanten rundt lokket.



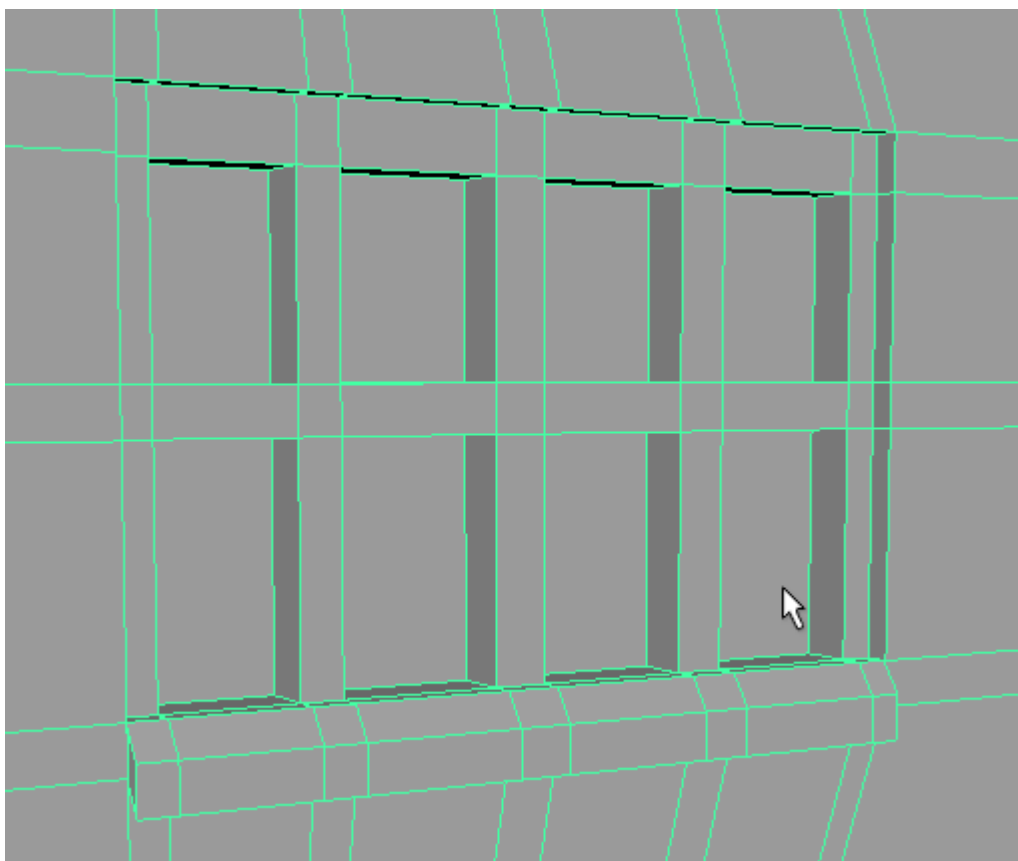
Figur 82:låsring til post kasse i kunn mesh.



Figur 83: lås utforming til postkasse i kunn mesh.

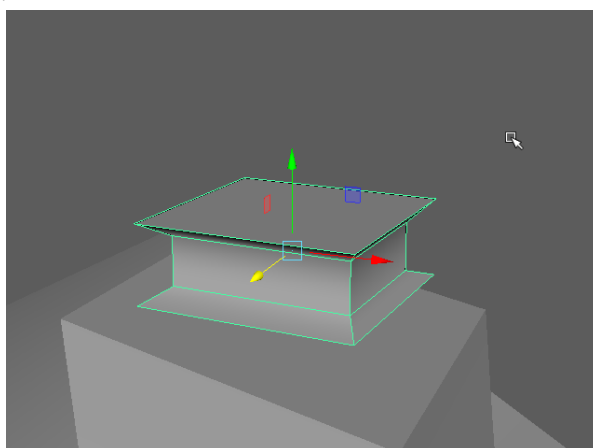
Hengelås ringen ble laget med en bridge fra et ansikt til en annen og smooth for å få den rund. Låsen ble formet av en firkant, og formet låsbuen med extrude og smooth deretter ble skalert ned til å passe låseringen på postkassen.

Vinduene ble formet med extrude innover og insert edge loop verktøyet til å lage rammene. Etter at de ble formet riktig i forhold til hverandre, extrudet jeg ut karmene og bøyde nedsiden nedover.

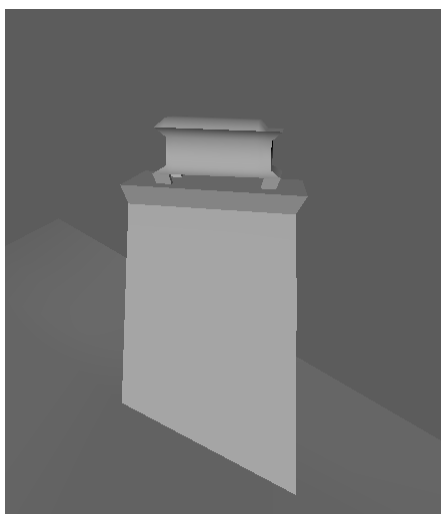


Figur 84: vindu utforming i kunn mesh

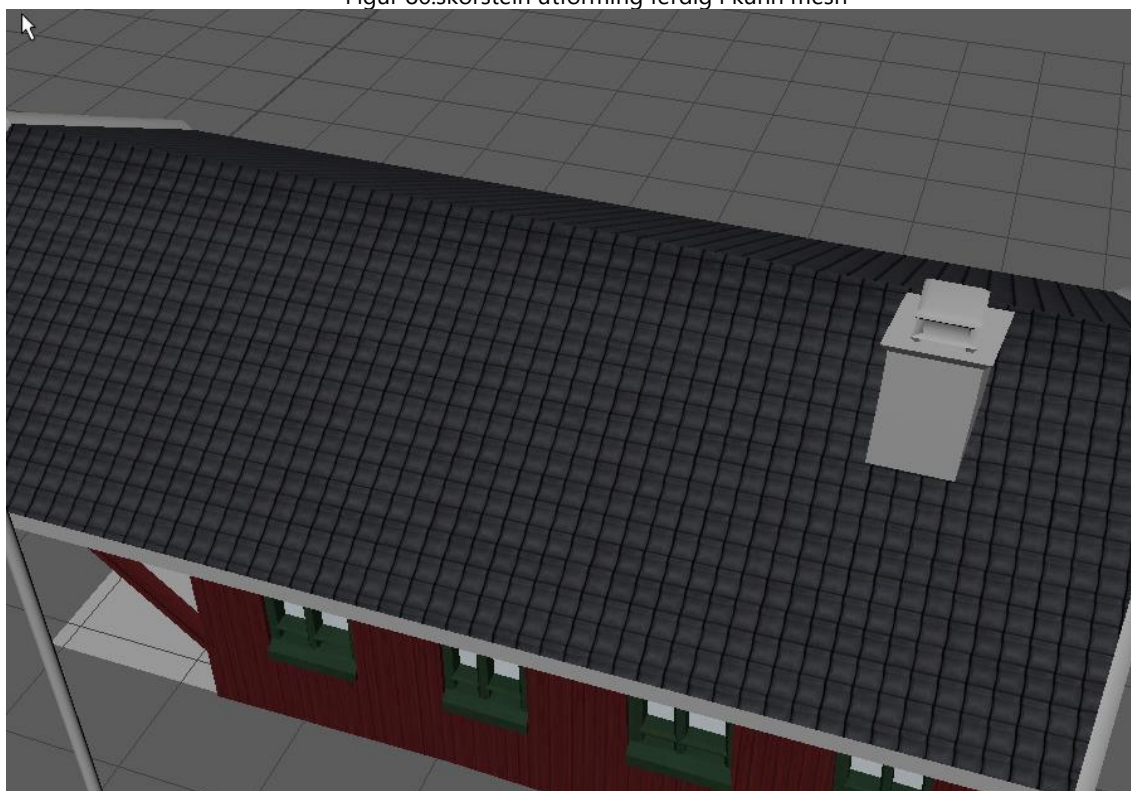
Skorsteinen på hytten ble modellert med å lage en firkant med insert edge loop verktøyet cirka rundt punktet som den antyder å være på bildet. Deretter ble skorsteinen ekstrudert fra taket og utformet. Top seksjonen ble laget av en firkant som ble manipulert innover på midt ansiktene av modellen, og dyttet opp toppen for å få en buet form, dette ble gjort med bevegelses verktøyet. Beina på topp delen av skorsteinen ble laget med å extrudere de fire hjørnene og forme de i en bue med extrude verktøyet.



Figur 85: skorstein topp utforming i kunn mesh



Figur 86: skorstein utforming ferdig i kunn mesh



Figur 87: takstein utforming på hytten, med teksturer.

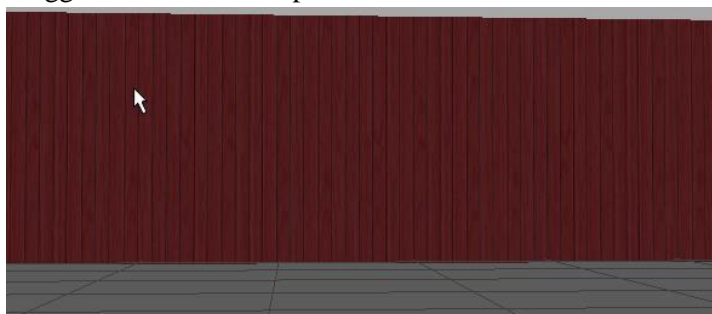
Taket er teksturert av med et bildet funnet fra google som passet til typen taket⁵⁶. Først ble hele taket dekket og byttet materialet til lambert og fil, så ble planar brukt på hver side av taket og forandret til å ha flere steiner.

56 <https://i.pinimg.com/236x/00/e7/a9/00e7a9cc34ec3ba582709a3749993ebb.jpg>



Figur 88: Hytteveggen nær inngangen, teksturert

Veggen ble teksturert på samme måte som taket, men top seksjonen ble rotert også for å etterligne hvordan veggen er satt sammen på bildet⁵⁷.



Figur 89: langvegg uten vinduer, fult teksturert.

⁵⁷https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRqPGiOodDjn8t1nZPTcJvZZ3hRdjUE11-Qn0DDcZXRp_kwmhSC



Figur 90: skorskein fult teksturert.



Figur 91: hytte vegg med vinduer, fult teksturert.

Bildene over viser teksturerte vegger, og skorstein teksturert. Det var ikke så mye å finne angående inngangs partiet til hytten, så teksturen ble grønn på postkassen og vinduskarmene ovenfor⁵⁸ og lys tre på inngangspartiet⁵⁹. Glass teksturen ble også hetet fra google søk.⁶⁰

58

https://www.amazingpaper.com.au/auto/thumbnail/persistent/catalogue_images/products/abdar2-150x150.jpg?maxheight=280&style=cropped&maxwidth=295

59 <https://cmkt-image-prd.global.ssl.fastly.net/0.1.0/ps/3880016/1820/1213/m1/fpnw/wm1/uaskjxcpunoijmldpjlazztatiswm18esrfnztu480xovz6wo7fgi652hpkuity-.jpg?1516606436&s=465d3711df9392b18fb56356fe8fd2df>

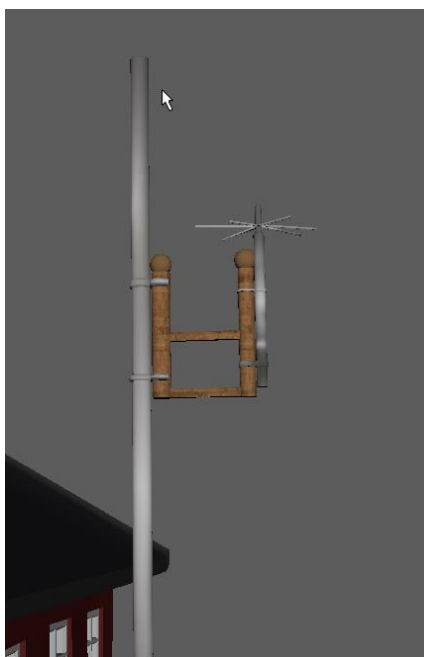
60

https://t4.ftcdn.net/jpg/00/83/03/61/240_F_83036164_DFBDNKEExRw2zdty21Az4e2PvZKN4pbTt.jpg



Figur 92: inngangs partiet til hytten.

Metall teksturen brukt på stolpen utenfor og er en sølv metall tekstur også funnet fra google.61



Figur 93: strømstolpe

Tre teksturen er den samme som inngangspartiet bruker, se fotnote 58.

61 <https://creativemarket.com/Myimagine/2021154-Silver-metal-texture-background>

10.1.3 Bekken Gård

Bildereferanser

Bildene som er brukt til modelleringen til huset, låven og hønsehuset kan du se i Figur 94 og 95. Som man kan se på bildene, så er fargene på bygningene varierende ut ifra hvor slitt det er, som gjør det vanskelig å legge på materiale/tekstur.



Figur 94: Bekken Gård. Huset, låven og hønsehuset avbildet.



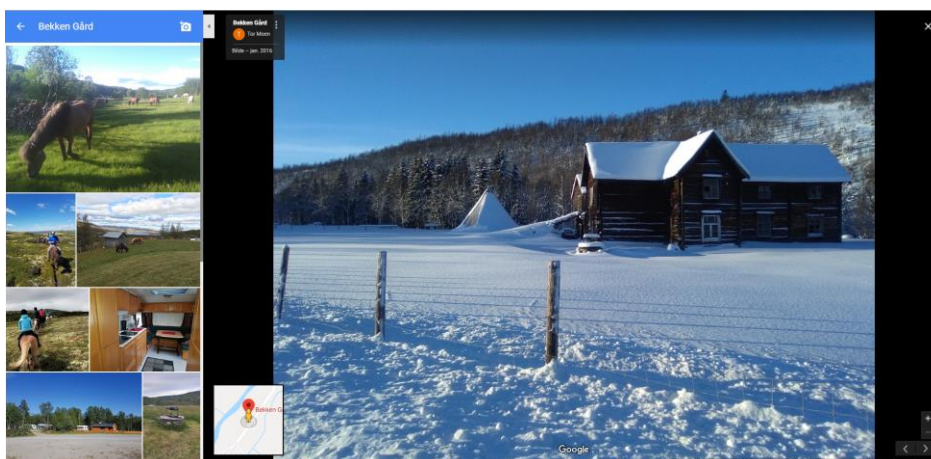
Figur 95: Bekken Gård. Huset og låven litt nærmere.

Bildet som er brukt til modelleringen til verandaen på huset er på Figur 96.



Figur 96: Bekken Gård. Verandaen på huset.

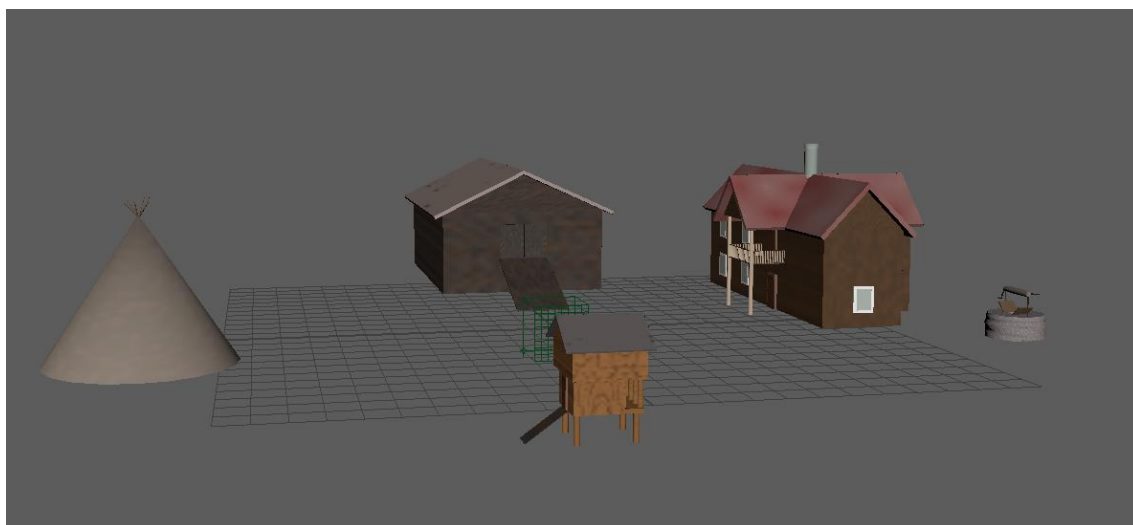
Bildet som er brukt til modelleringen til lavvoen og en annen side av huset, samt brønnen kan du se i Figur 97.



Figur 97: Bekken Gård. Lavvoen, brønnen og annen side av huset.

3D Modeller

I Figur 98 kan du se 3D modellen av hele Bekken Gård området. Ved å ta i bruk alle bildene over, er bygningene som er modellert under plassert med en distanse fra hverandre som tilsvarer slik de antas å stå ut ifra bilde referansene.



Figur 98: Helhetsbilde av 3D modellen til Bekken Gård.

Brønnen, som du kan se på Figur 99. Selve basen er laget med en sirkel hvor det er flyttet på vertex og faces for å ligne på steiner plassert oppå hverandre. Hullet er laget ved å extrudere sirkelen som skaper åpningen for hullet. Plankene ved siden av er tynne kuber om er rotert. Håndtakene er tynne sylindere som er justert passende ved å flytte edges og vertex. Den vannrette sylinderen er litt mindre i endene ved å minske sirkelen av edges lengst ute, og forstørre de litt i midten.

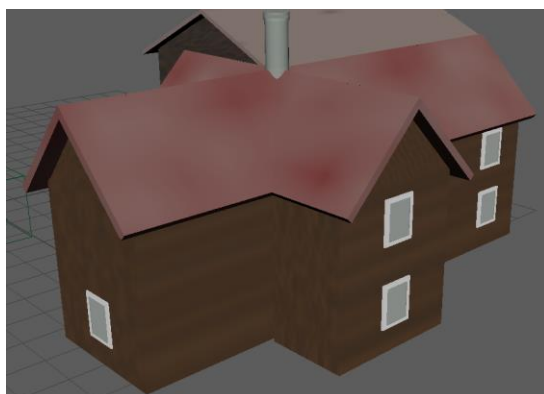


Figur 99: Bekken Gård 3D modell. Brønnen

Hovedhuset, vist i Figur 100 og 101. Veggene, dører og vinduer er alle laget av kuber i forskjellige størrelser. Takpipa er laget av to sylindere. Taket er laget med tynne kuber som er rotert. Verandaen er laget med kopierte tynne sylindere. Ingen av dørene er synlige på bildereferansene, men antas å være der gjennom elimineringsprosess.



Figur 100: Bekken Gård 3D modell. Forside av huset.



Figur 101: Bekken Gård 3D modell. Bakside av huset.

Hønsehuset, som du kan se på Figur 102 og 103. Laget med kuber og sylindere i forskjellige størrelser. Taket og rampa er roterte kuber. Sylindrene har hakk i seg, som er laget ved å legge til edge-loops og minske noen av de. Døren er ikke synlig på bildereferanse, men antas å være der gjennom elimineringsprosess.

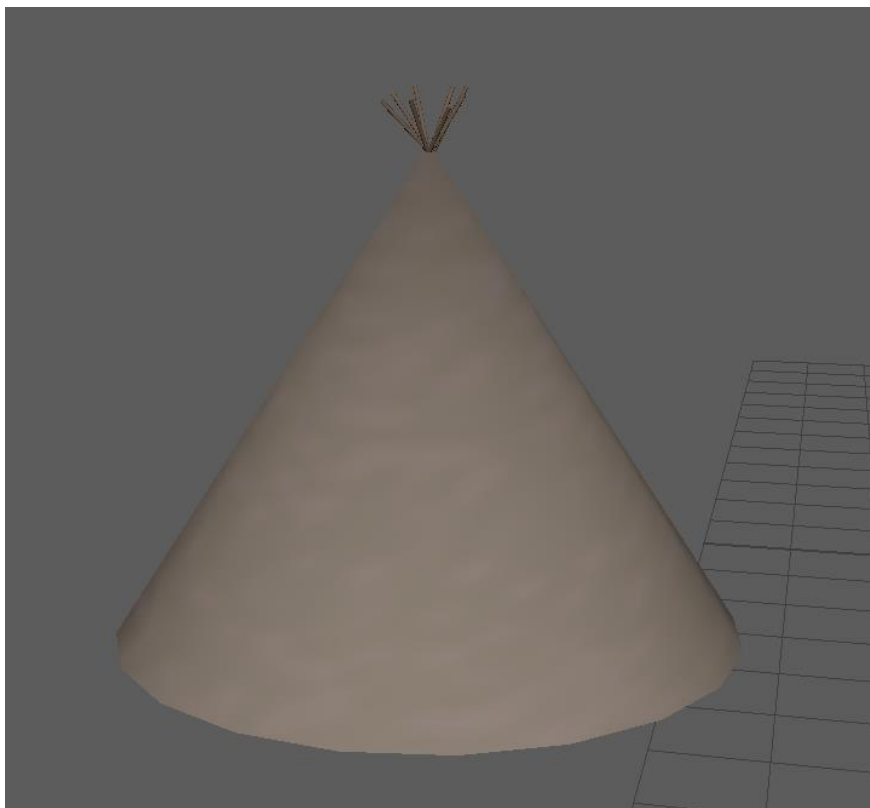


Figur 102: Bekken Gård 3D modell. Hønehuset.



Figur 103: Bekken Gård 3D modell. Andre side av hønehuset.

Lavvoen, som vist på Figur 104. Er laget med en polygon cone, og trestykkene på toppen er sylindere som er rotert rundt toppen av lavvoen.



Figur 104: Bekken Gård 3D modell. Lavvoen.

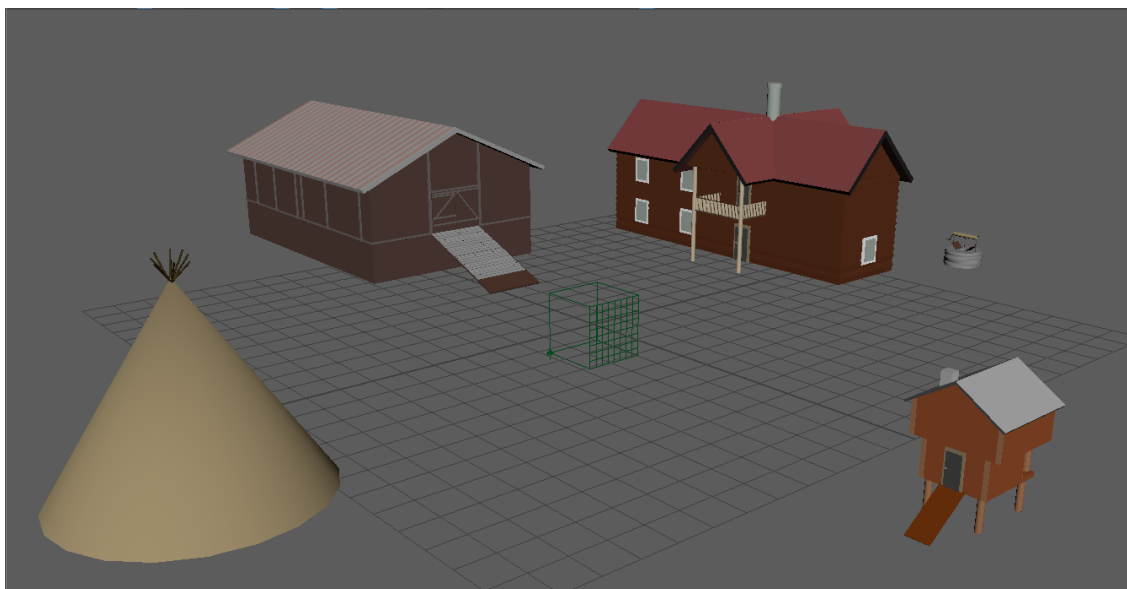
Figur 105 viser låven. Laget med kuber som er justert til å passe målene fra bildereferanser.



Figur 105: Bekken Gård 3D modell. Låven.

Endring av materiale og farger

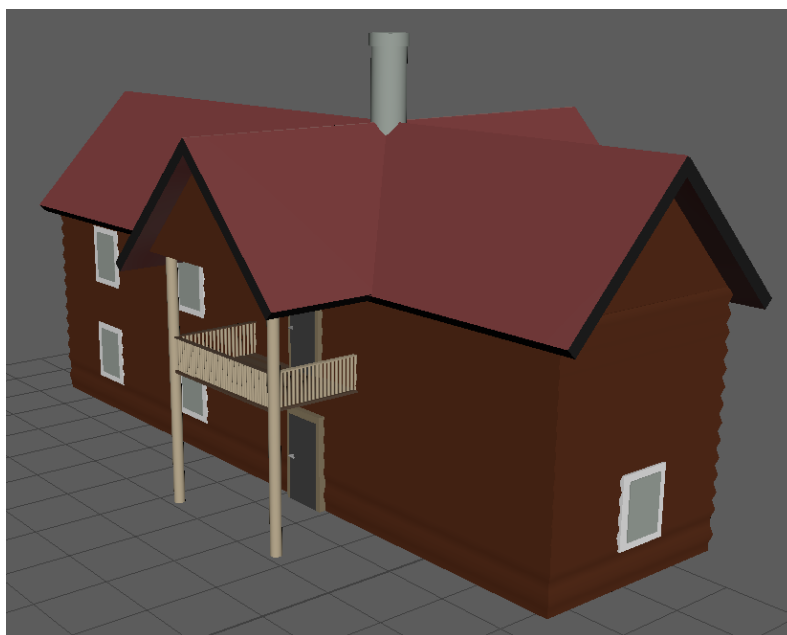
Figur 106 viser et helhetsbilde av gården etter endring av materiale og farger. Hadde problemer med å få overført teksturen av wood, leather og rock som .mb fil. Prøvde også å eksportere som FBX fil, men fikk også problemer med dette. Siden det ikke så veldig bra ut med de originale teksturene, valgte vi å bare legge til materiale og farge på gården.



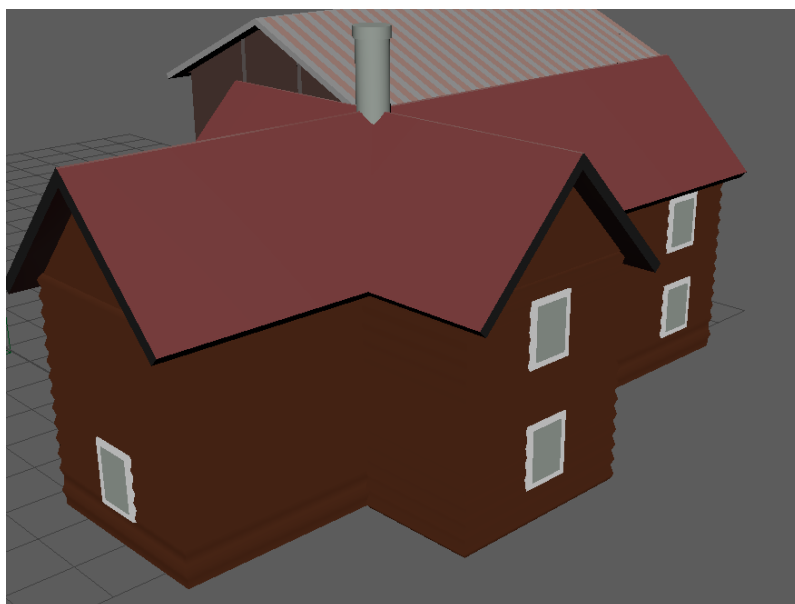
Figur 106 Bekken Gård 3D modell etter endring fra tekstur til materiale. Helhetsbilde av gården.
 Figur 107 viser brønnen etter endring fra tekstur til materiale.



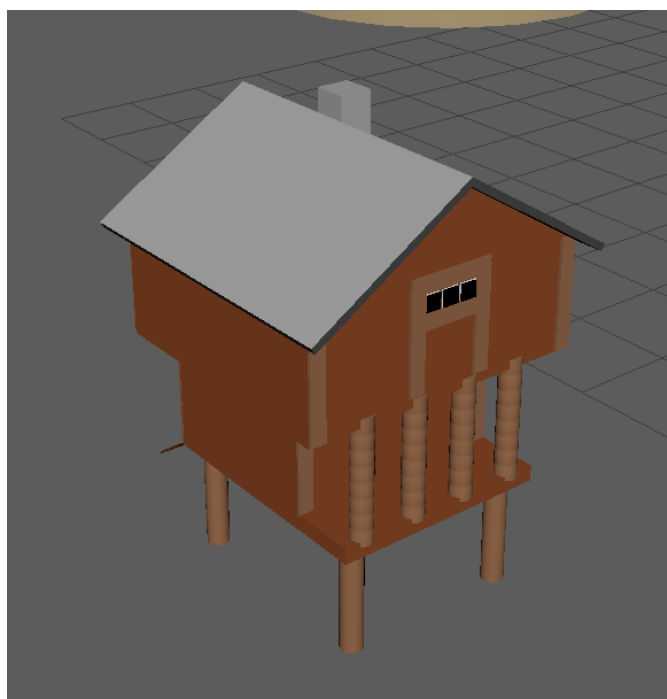
Figur 107: Bekken Gård 3D modell etter endring fra tekstur til materiale. Nærbilde av brønnen. Figur 108 og 109 viser huset etter endring fra tekstur til materiale.



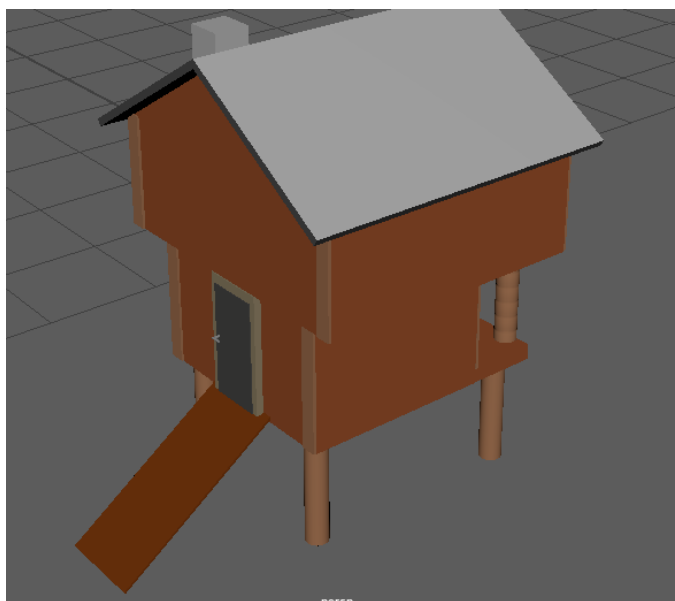
Figur 108: Bekken Gård 3D modell etter endring fra tekstur til materiale. Forside av huset.



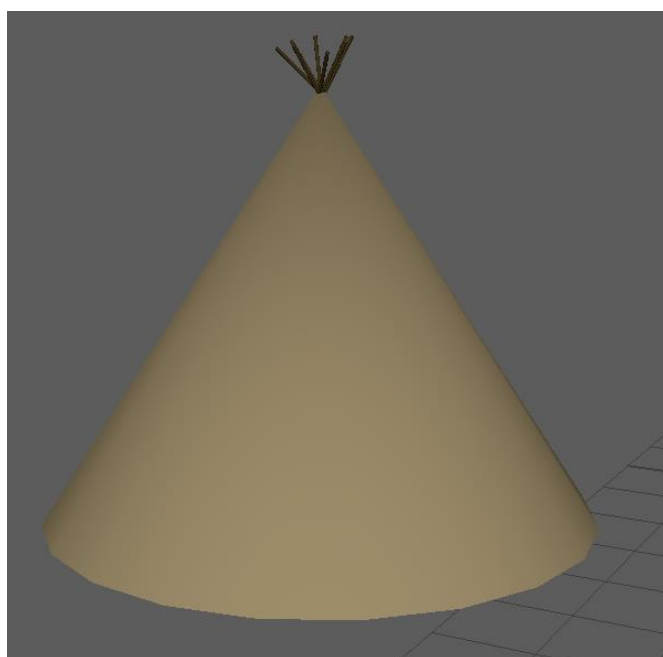
Figur 109: Bekken Gård 3D modell etter endring fra tekstur til materiale. Bakside av huset. Figur 110 og 111 viser hønehuset etter endring fra tekstur til materiale.



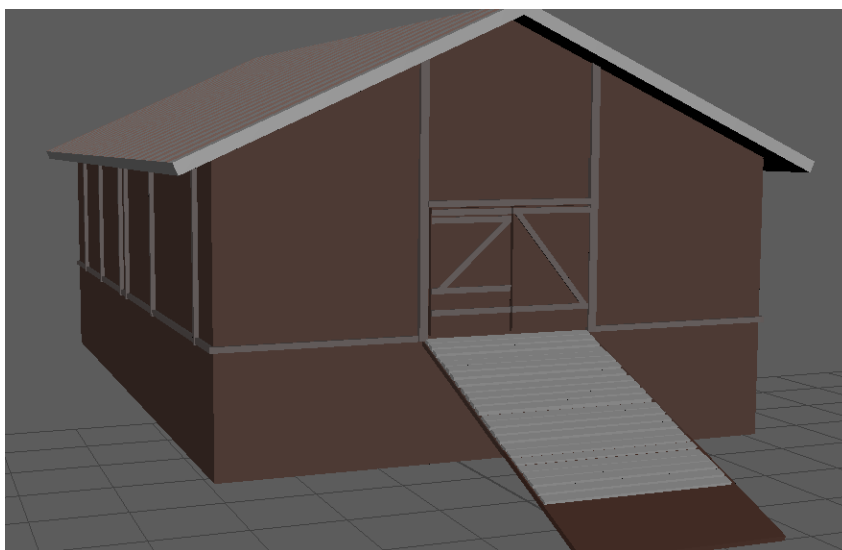
Figur 110: Bekken Gård 3D modell etter endring fra tekstur til materiale. Hønehuset.



Figur 111: Bekken Gård 3D modell etter endring fra tekstur til materiale. Andre side av hønehuset.
Figur 112 viser lavvoen etter endring fra tekstur til materiale.



Figur 112: Bekken Gård 3D modell etter endring fra tekstur til materiale. Lavvoen.
Figur 113 viser låven etter endring fra tekstur til materiale.



Figur 113: Bekken Gård 3D modell etter endring fra tekstur til materiale. Låven.

10.1.4 Samfunnshuset

Bildereferanse

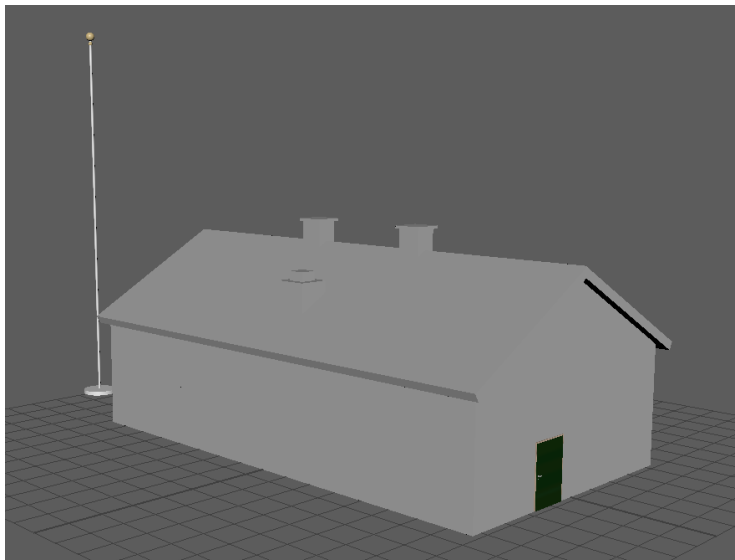
Figur 114 viser bildet som er brukt som referanse til å modellere Samfunnshuset.



Figur 114: Samfunnshuset.

3D Modeller

Figur 115 viser 3D modellen av samfunnshuset underveis. Som man kan se mangler den en inngang til, vinduer og materiale.



Figur 115: Samfunnshuset 3D modell underveis.

Figur 116 viser den ferdige modell av samfunnshuset. Alt på huset er laget med kuber i forskjellige størrelser, farger og rotinger, bortsett fra dørhåndtak som er laget med sylindere. Flaggstanga er laget med en lang, tynn sylinder, en kule på toppen, og en bredere sylinder i bunn.



Figur 116: Samfunnshuset 3D modell ferdig.

10.1.5 Pub1

Bildereferanser

Figur 117 viser google maps street view av Pub1. Figur 118 viser et nærmere bilde funnet på google bilder. Bildene ble brukt som referanse under 3D modelleringen av Pub1.

Google maps street view:

<https://www.google.com/maps/@62.7892917,11.1805388,3a,75y,310.48h,82.09t/data=!3m6!1e1!3m4!1s9nPYaKezTWF2dasuqDiDOw!2e0!7i13312!8i6656>



Figur 117: Google street view av Pub1.



Figur 118: Google bilde av Pub1.

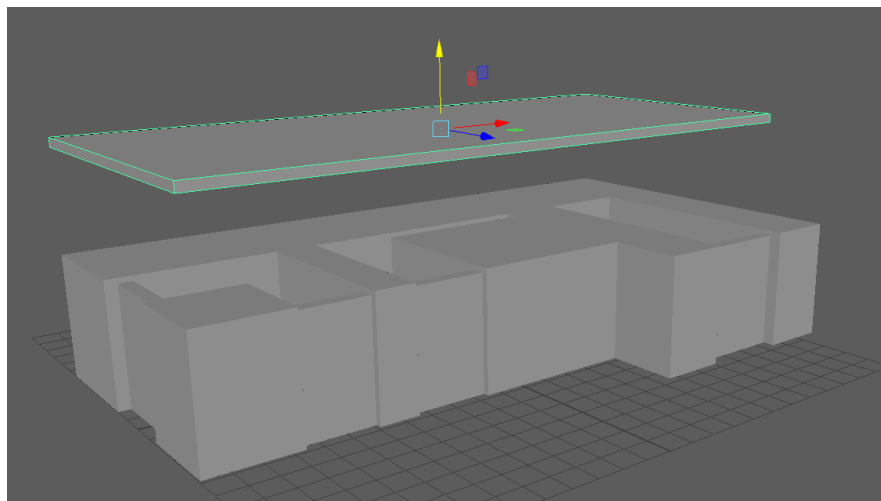
3D Modeller

Figur 119 viser en ferdig 3D modell av Pub1. Endret litt på gul/oransje fargen underveis, og endte opp med en sterk oransje farge.



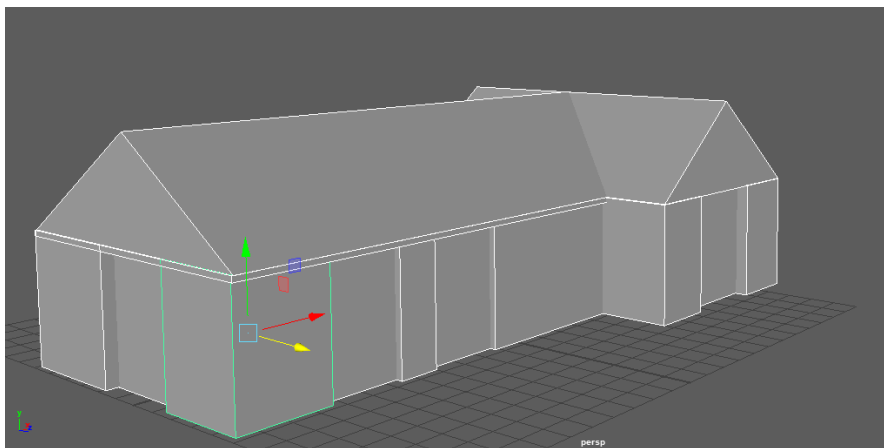
Figur 119: Pub1 3D modell ferdig.

Grunnivået, som vist på figur 120, av bygningen er satt sammen av forskjellige kuber som er justert for å lettere kunne lage innrykk ved garasjedørene, samt gi materialene farger som passer. En annen metode for å lage disse innrykkene ved garasjedørene ville være å bruke extrude, men valgte å gå for en enklere metode.



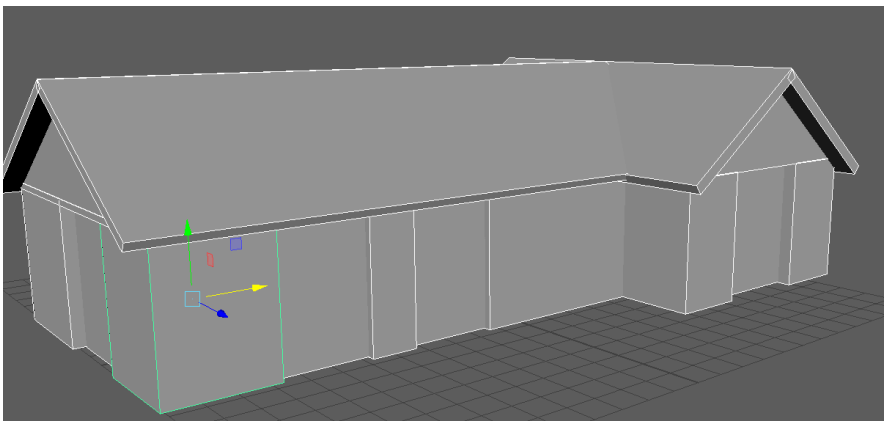
Figur 120: Pub1 3D modell. Grunnivået av bygningen.

Figur 121 viser trekant-figurene som er plassert over er laget med polygon prisms, som er flatet ut til å passe bildereferansene.



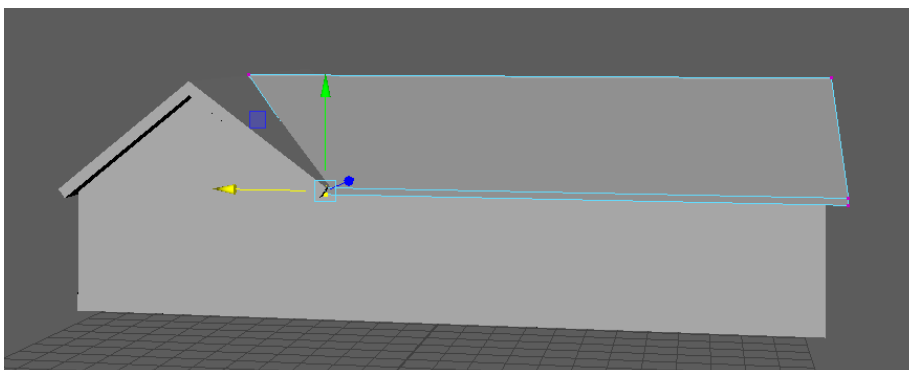
Figur 121: Pub1 3D modell. Grunnstruktur av bygningen.

Figur 122 viser taket som er lagt til, laget med tynne kuber som er rotert skrått for å passe til husoverflaten.



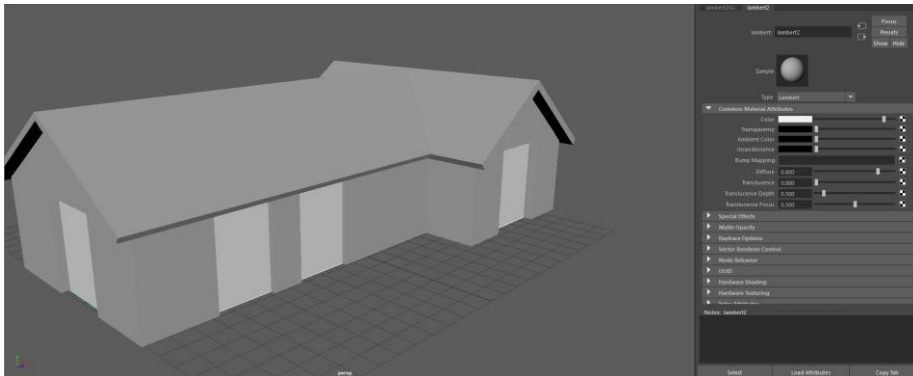
Figur 122: Pub1 3D modell. Taket lagt til.

Flyttet vertex slik at den lange siden av taket ikke stikker ut på den korte siden, på venstre side av huset, som vist på Figur 123 (kunne også vært gjort med edges).

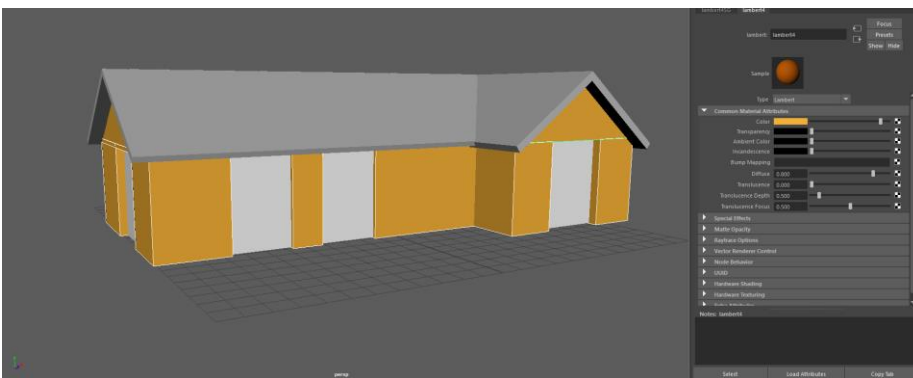


Figur 123: Pub1 3D modell. Justert tak.

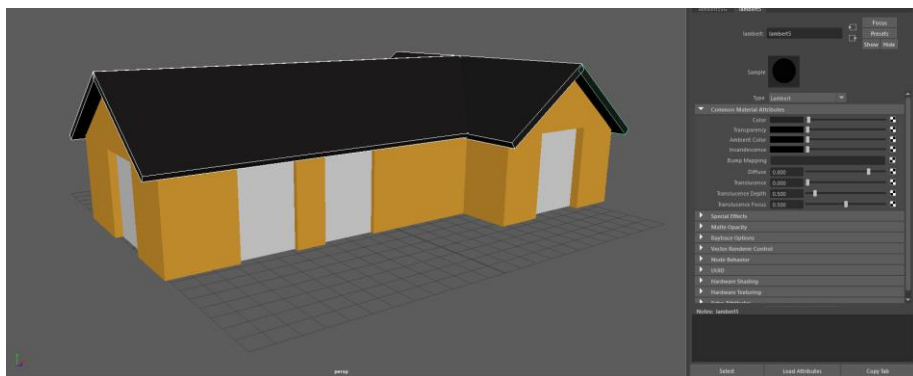
Figur 124 viser materiale og farge gitt til garasjedørene.



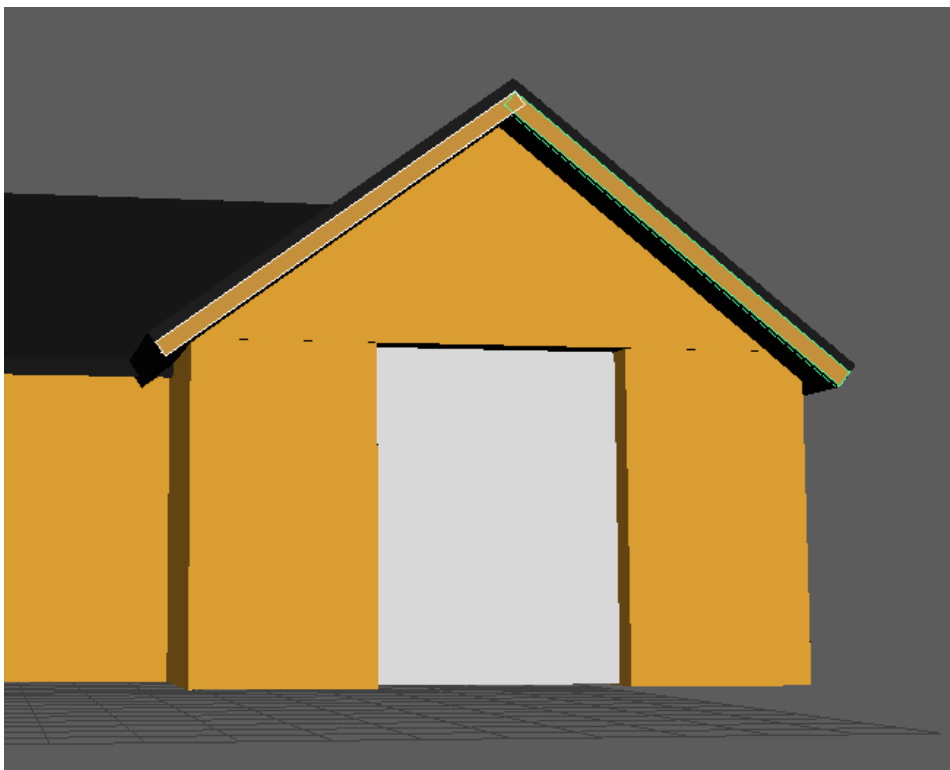
Figur 124: Pub1 3D modell. Materiale og farge til garasjedørene.
Figur 125 viser materiale og farge gitt til veggene.



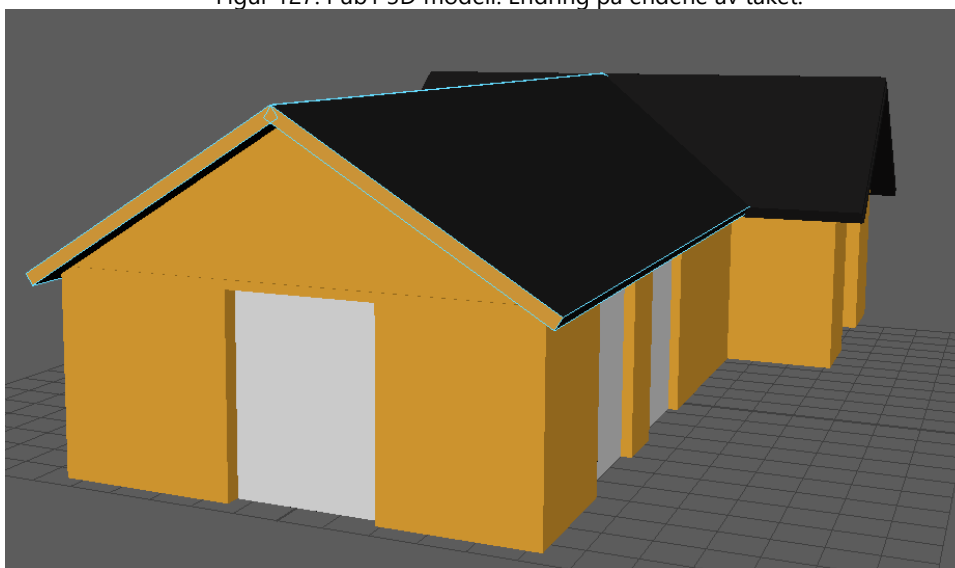
Figur 125: Pub1 3D modell. Materiale og farge til veggene.
Figur 126 viser materiale og farge gitt til taket.



Figur 126: Pub1 3D modell. Materiale og farge til taket.
La merke til at det var gult på endene av taket, og la til det ved å legge til ekstra kuber utenpå, som vist på Figur 127 og 128.

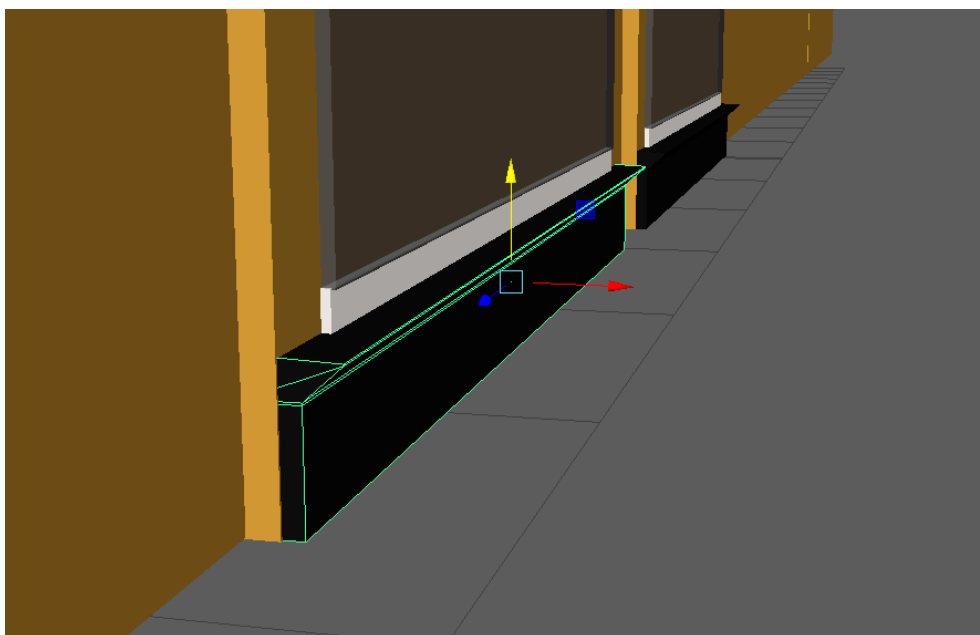


Figur 127: Pub1 3D modell. Endring på endene av taket.

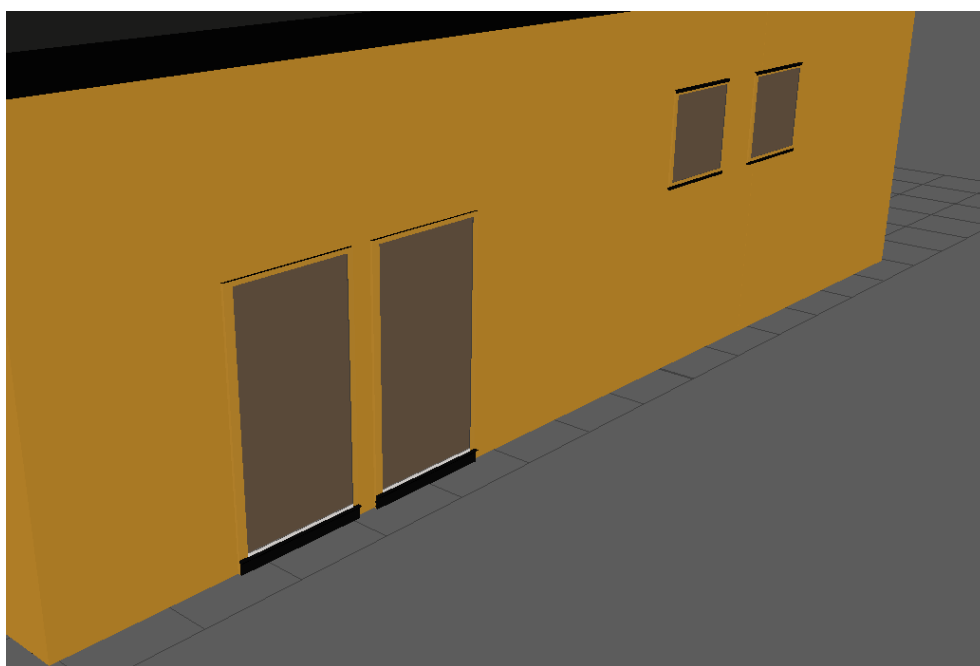


Figur 128: Pub1 3D modell. Endring på taket, annen side.

Figur 129 viser detaljer på vinduene. Nederste delen laget ved å legge til edge loops, for å så dra ut og krympe tykkelsen.



Figur 129: Pub1 3D modell. Detaljer på vindu.
Figur 130 viser en sidevegg gjort ferdig.

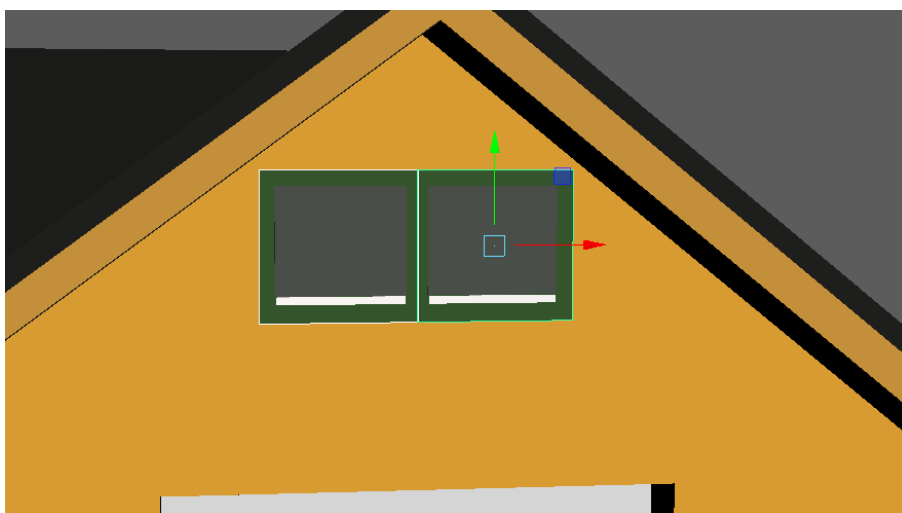


Figur 130: Pub1 3D modell. Sidevegg ferdig.
Figur 131 viser et helhetsbilde underveis i 3D modellerings prosessen.



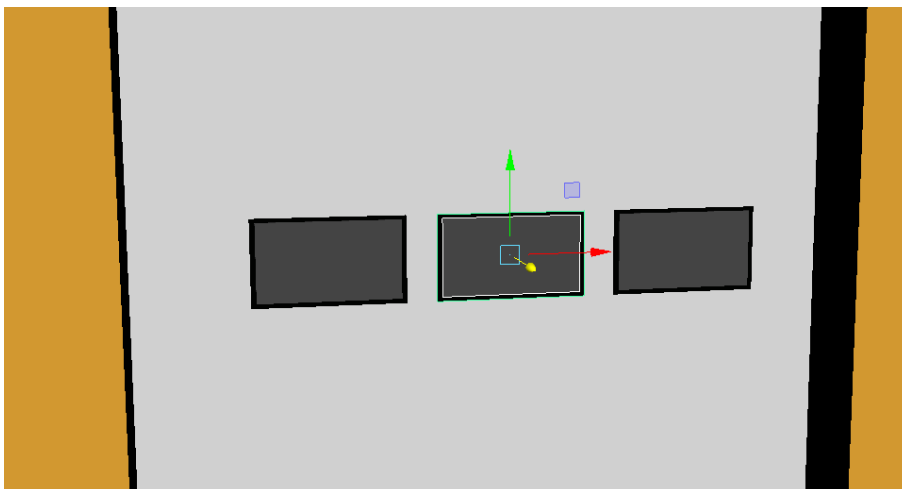
Figur 131: Pub1 3D modell. Helhetsbilde underveis.

Figur 132 viser vinduene over garasjedøren. Kopiert fra de andre vinduene, og endret farge.



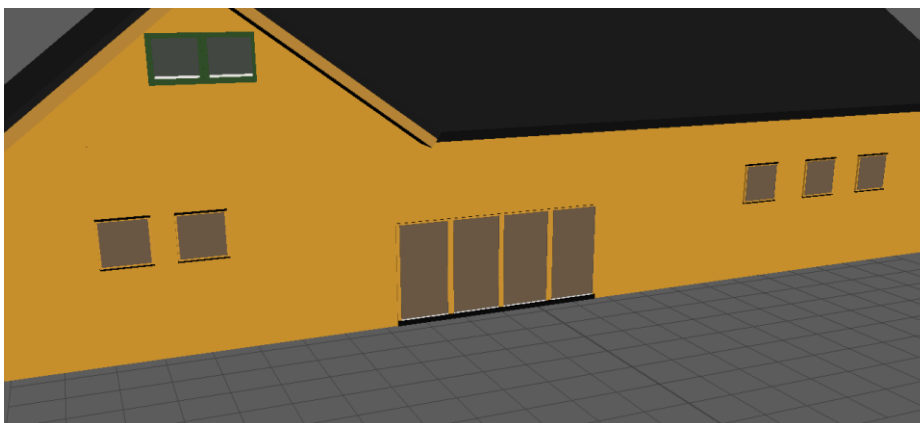
Figur 132: Pub1 3D modell. Vinduene over garasjedøren.

Figur 133 viser vinduene på garasjedørene. Også kopiert fra andre vinduer, og tilpasset bredde.



Figur 133: Pub1 3D modell. Vinduene på garasjedørene.

Figur 134 viser baksiden av Pub1. Mangler bildereferanse eller google street view tilgang til å se baksiden, så det lagt til vinduer for at det ikke skal være helt blankt. Vil også si at med tanke på hvor mange dører og garasjedører som er synlig på forsiden, er det lite sannsynlig at det er flere på baksiden.



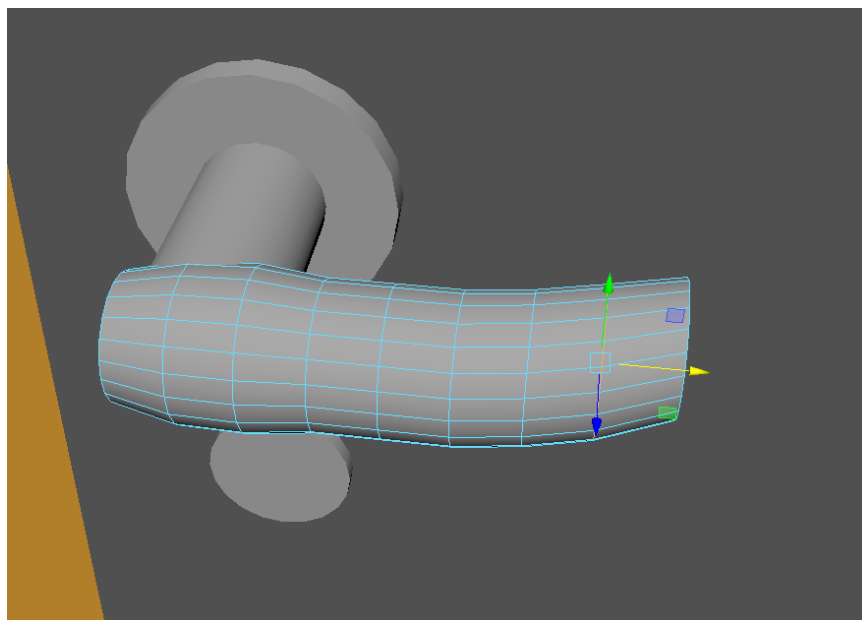
Figur 134: Pub1 3D modell. Baksiden av bygningen.

Figur 135 viser dørene som er synlige på forsiden. Kopiert fra de lange vinduene, og endret litt på størrelse og materiale.



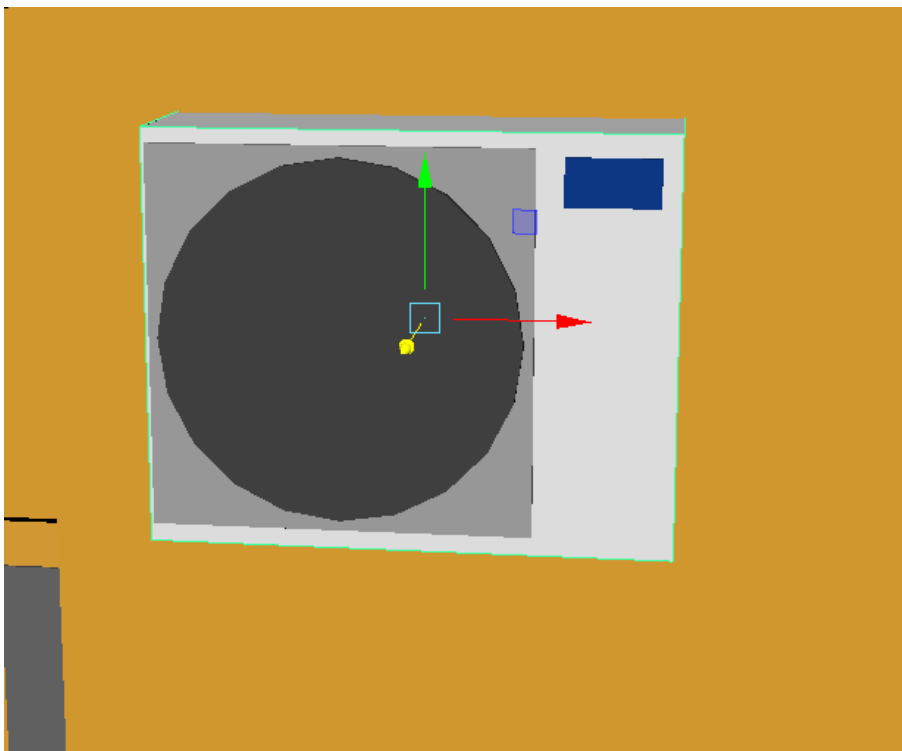
Figur 135: Pub1 3D modell. Dørene.

Figur 136 viser et dørhåndtak. Flyttet litt på edges, for at den ikke skulle være helt rett.



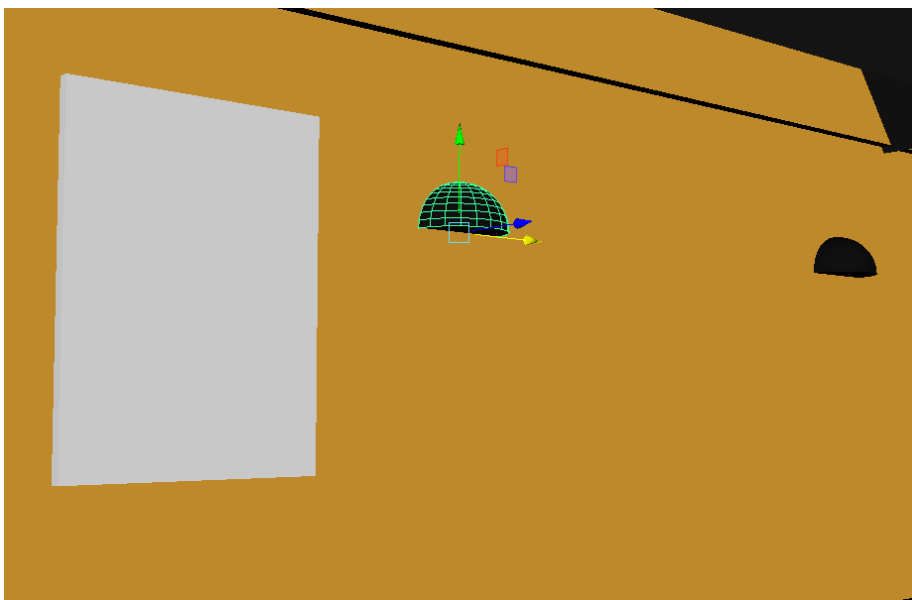
Figur 136: Pub1 3D modell. Dørhåndtak.

Figur 137 viser et aircondition anlegg, som er plassert over garasjedøren på forsiden. Laget med kuber og sylindere.



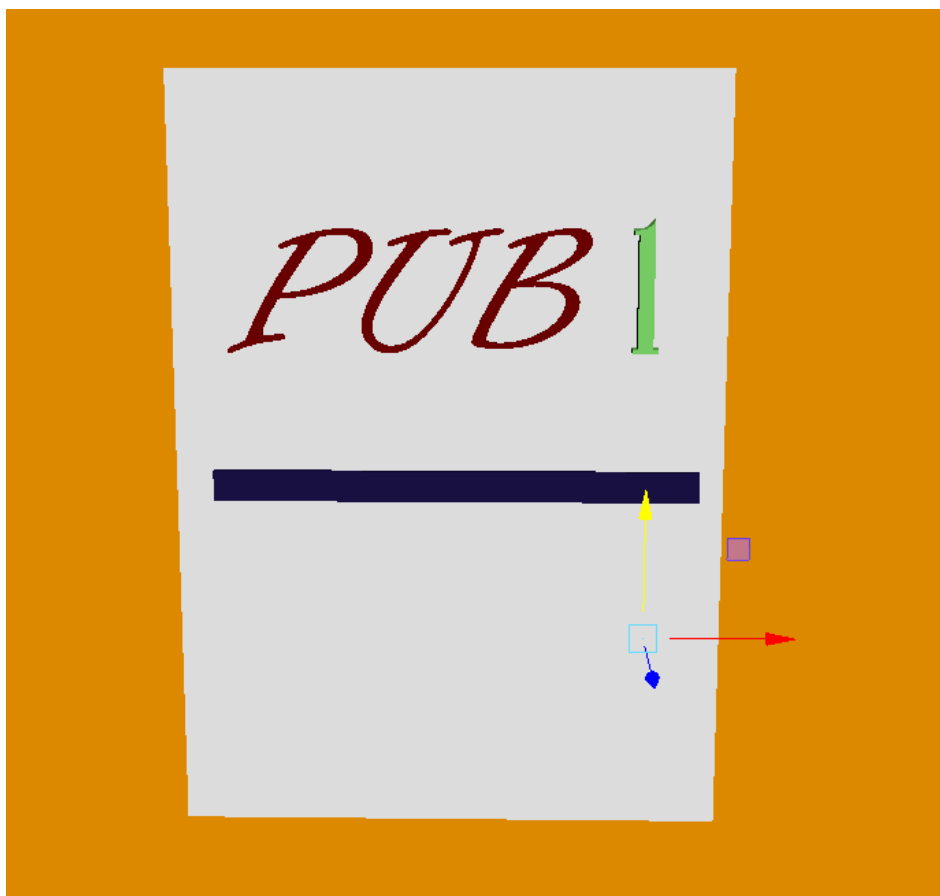
Figur 137: Pub1 3D modell. Aircondition anlegg.

Figur 138 viser utelys, og Pub1 skilt (underveis). Utelysene er laget ved å slette halvparten av en sphere (kule/ball), også fylle den nederste edge ved å bruke bridge funksjonen.



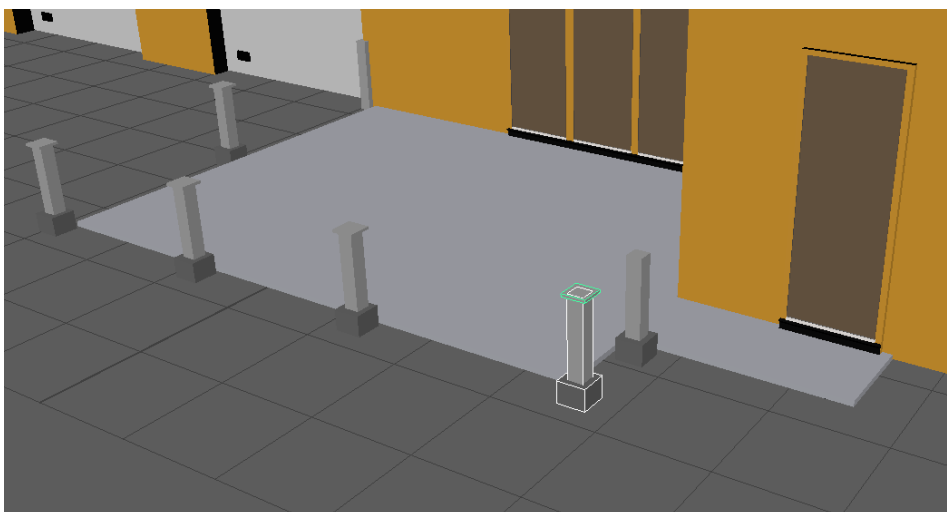
Figur 138: Pub1 3D modell. Utelys og Pub1 skilt underveis.

Figur 139 viser Pub1 skiltet gjort ferdig. Skriften er laget med type tool, der det er en egen for "PUB" og en egen for "1". Under er en avlang kube.



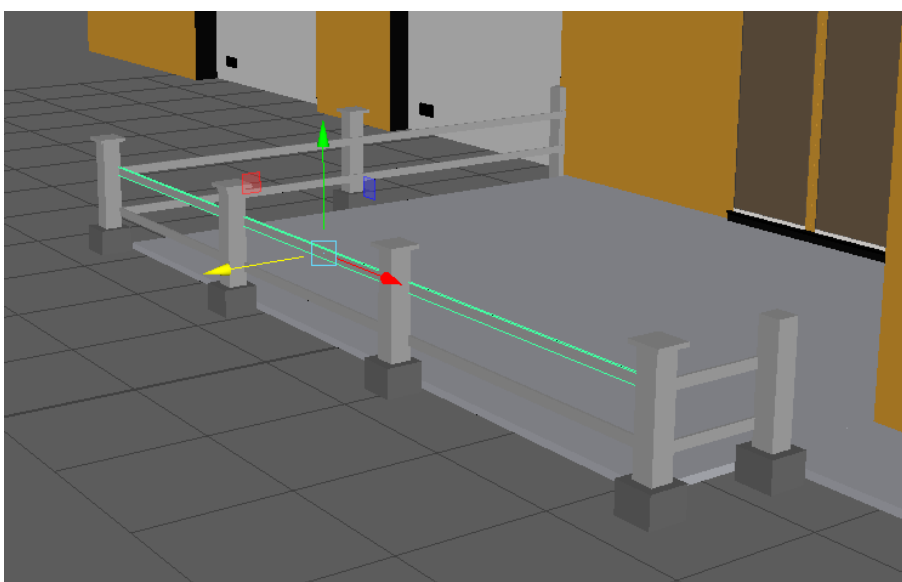
Figur 139: Pub1 3D modell. Pub1 skilt ferdig.

Figur 140 viser starten av uteplassen av området. Alt er laget med kuber med forskjellige størrelser.

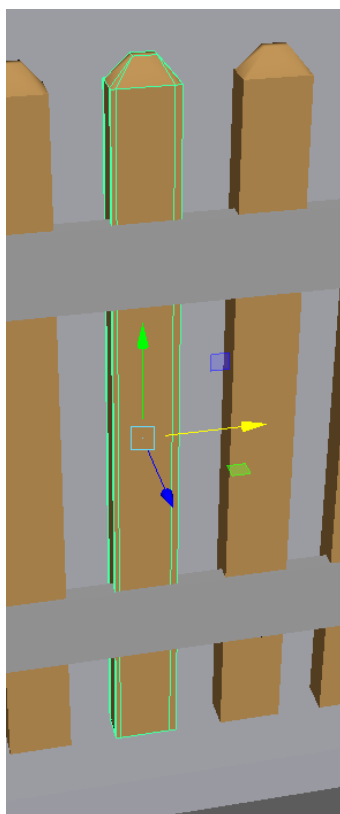


Figur 140: Pub1 3D modell. Starten på uteplass området.
Plankene bakpå gjerdene som går vannrett, også laget med lange kuber, som vist på figur

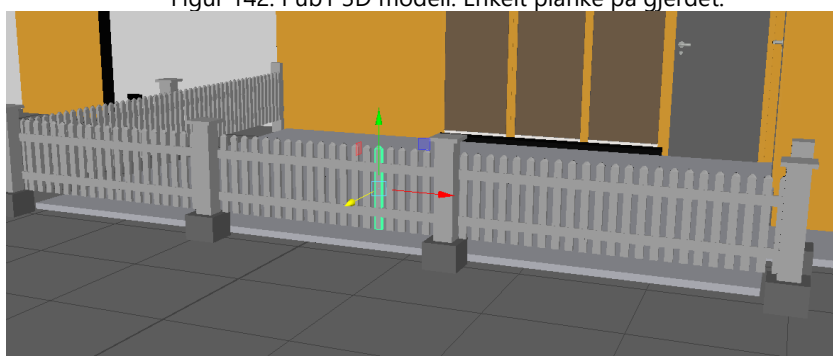
141.



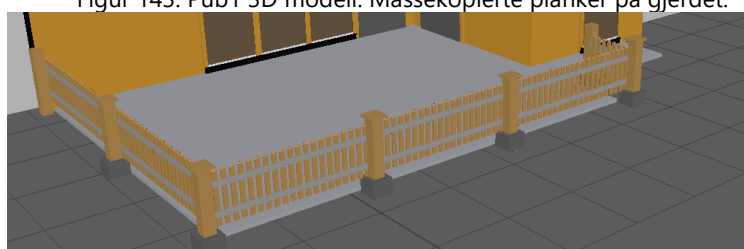
Figur 141: Pub1 3D modell. Gjerde på uteplass området.
Lagde en modell av plankene på gjerdet som går loddrett, og så masse kopierte de, vist på Figur 143 og 144. Toppen er gjort mindre ved å minske den øverste edgen på planken, vist på Figur 142.



Figur 142: Pub1 3D modell. Enkelt planke på gjerdet.

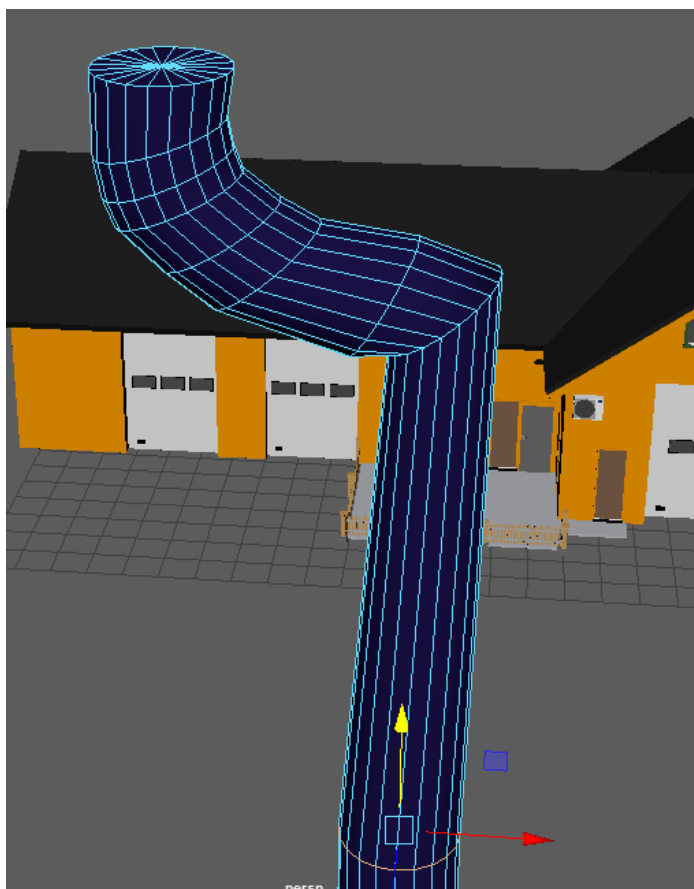


Figur 143: Pub1 3D modell. Massekopierte planker på gjerdet.

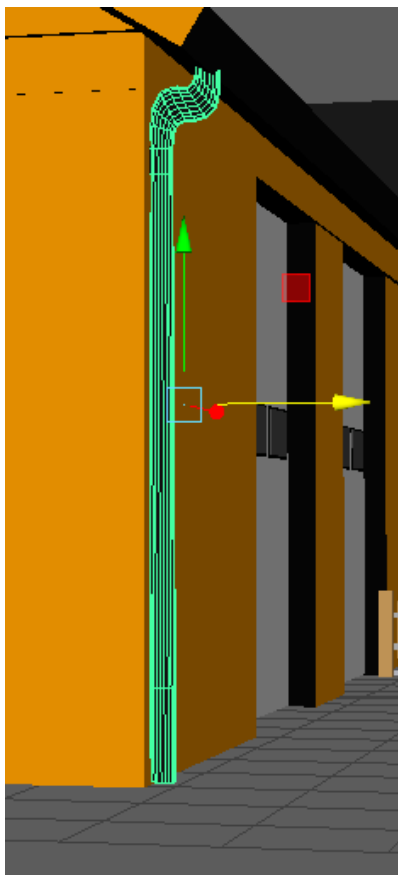


Figur 144: Pub1 3D modell. Uteplass området ferdig.

Figur 145 viser takrenne nedløpet underveis, og Figur 146 viser det som ferdig. Laget med en sylinder som er endret på edges og vertex for å lage formen av et takrenne nedløp.



Figur 145: Pub1 3D modell. Takrenne nedløp.



Figur 146: Pub1 3D modell. Takrenne nedløp ferdig.

10.1.6 Blue Box

Figurene 147-153 viser forskjellige bilder av de ulike delene av Blue Box fra ulike vinkler som er brukt som bildereferanser. Noen av bildene ser ut til å være tatt under byggingen og oppsettet av Blue Box, men modellen er laget med utgangspunkt at det er det er bildene hvor mest er på plass som er de nyeste, hvor målet med 3D-modellen var å lage den slik den ser ut i dag.

Bildereferanser



Figur 147: Blue Box. Lavt stativ med kamera/måler støttet til et tre.



Figur 148: Blue Box. Høyt stativ med kameraer/målere.



Figur 149: Blue Box. Nærbilde av lavt stativ med kamera/måler støttet mot et tre.



Figur 150: Blue Box. Nærbilde av den blå boksen.



Figur 151: Blue Box. Skiltet til den blå boksen.



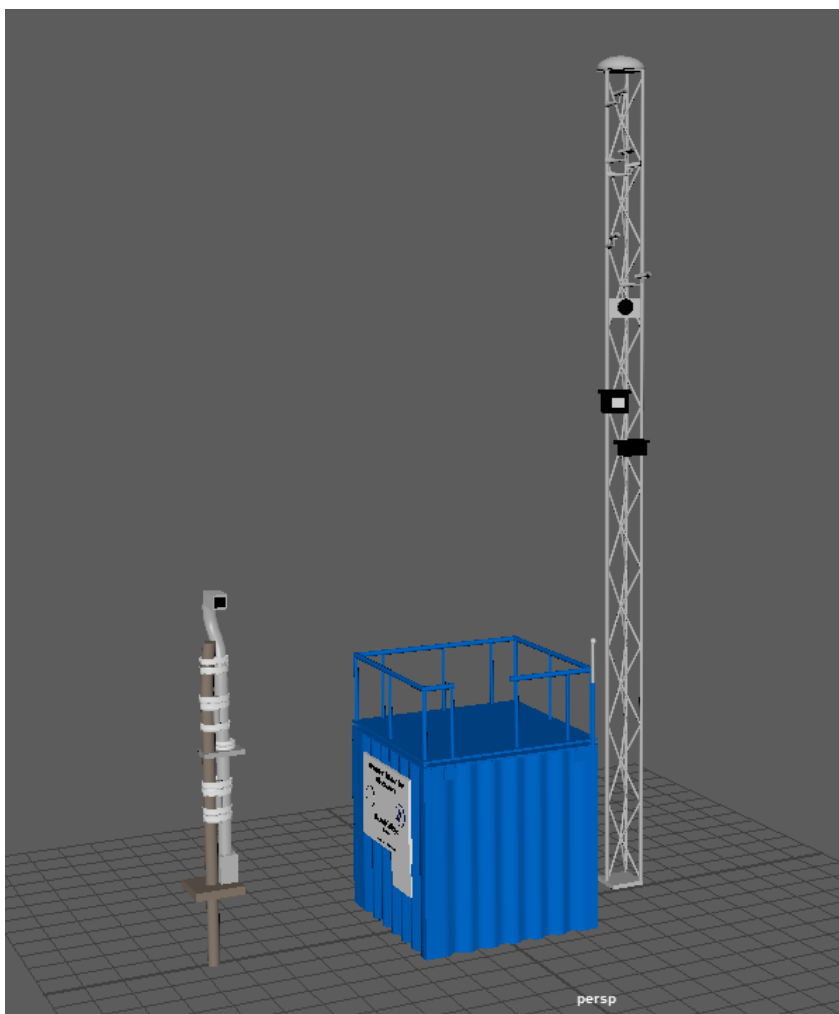
Figur 152: Blue Box. Helhetsbilde.



Figur 153: Blue Box. Høyt stativ med kameraer/målere ferdigstilt.

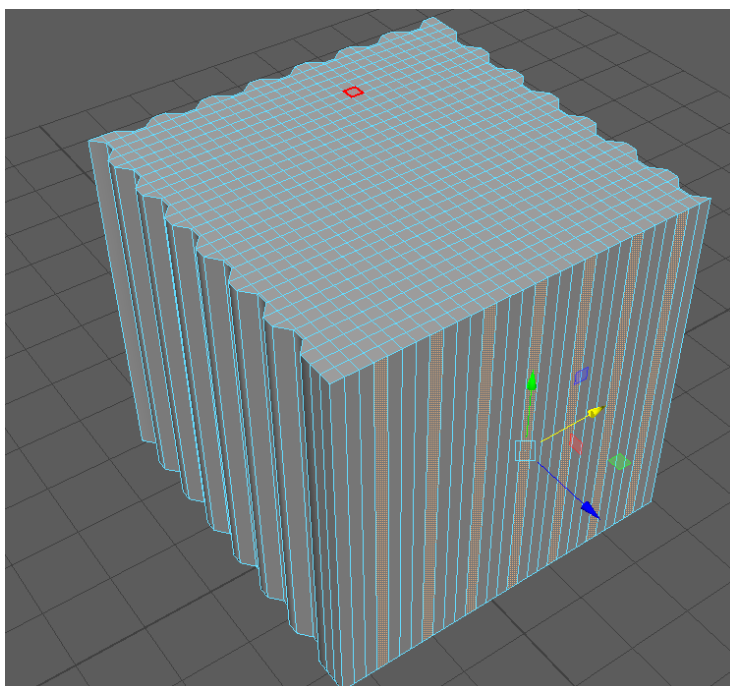
3D Modeller

Figur 154 viser et ferdig helhetsbilde av 3D modell av Blue Box.

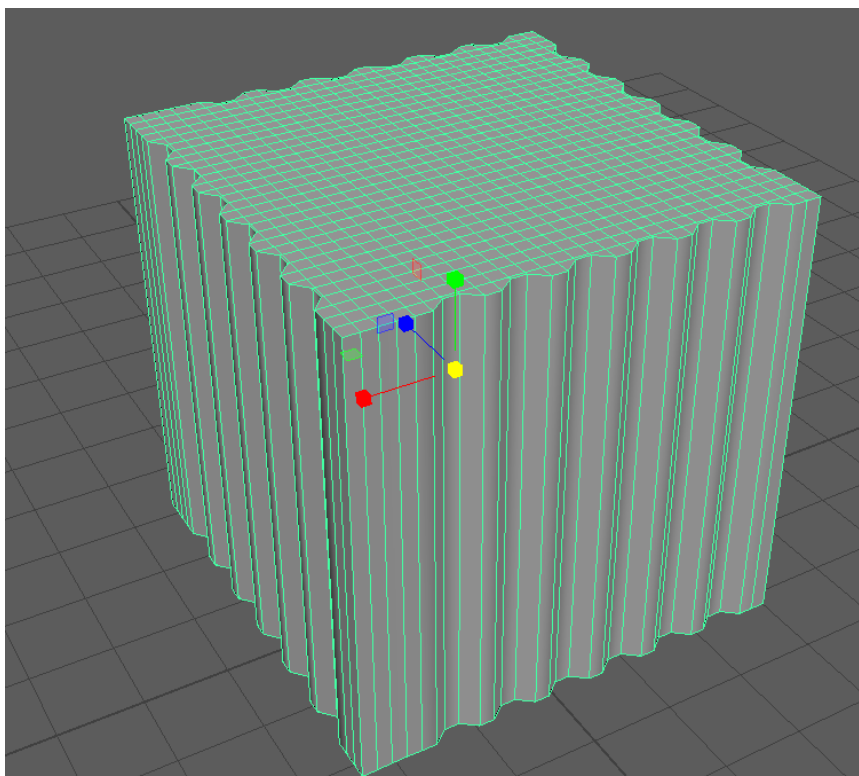


Figur 154: Blue Box 3D modell. Ferdig.

Den blå boksen av Blue box ble laget med en kube som er justert til passende størrelse. Deretter er hver fjerde face, som vist på Figur 155, på hver side trukket inn for å lage det bølgete mønsteret som boksen har i virkeligheten.

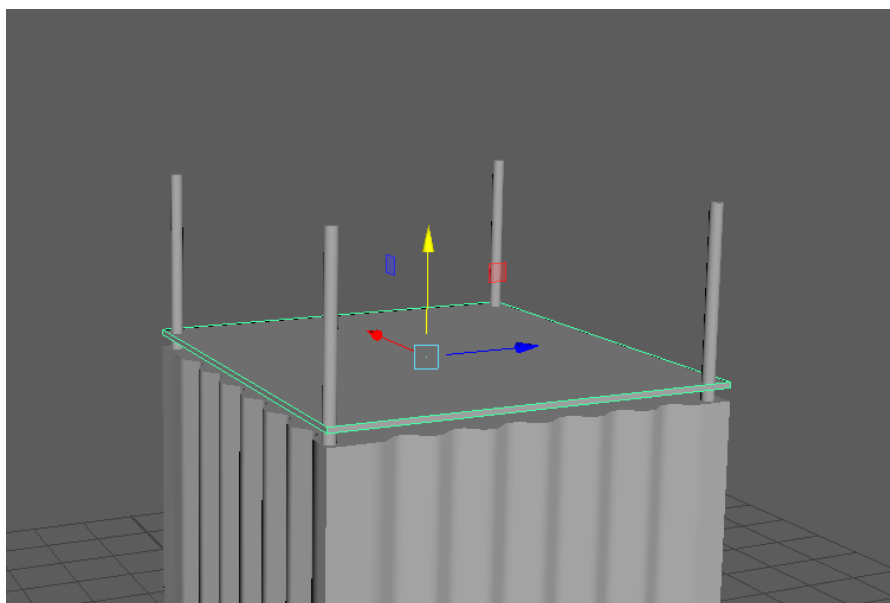


Figur 155: Blue Box 3D modell. Grunnstruktur av den blå boksen.
Fjernet noen av endene som ble for skarpe. Slik som vist på Figur 156 ser det endelige mønsteret ut.



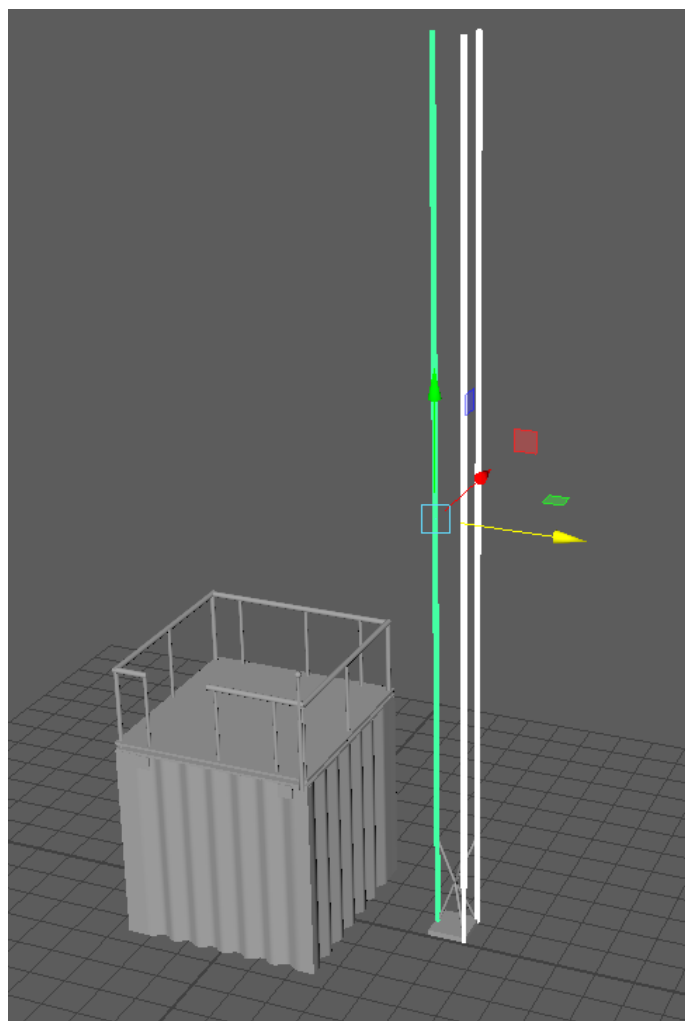
Figur 156: Blue Box 3D modell. Endelig mønster på den blå boksen.

Figur 157 viser platået over boksen, som er laget av kuber som er tynnet ut, tynne sylindere som er lagt vannrett og loddrett for å lage rekkverket.

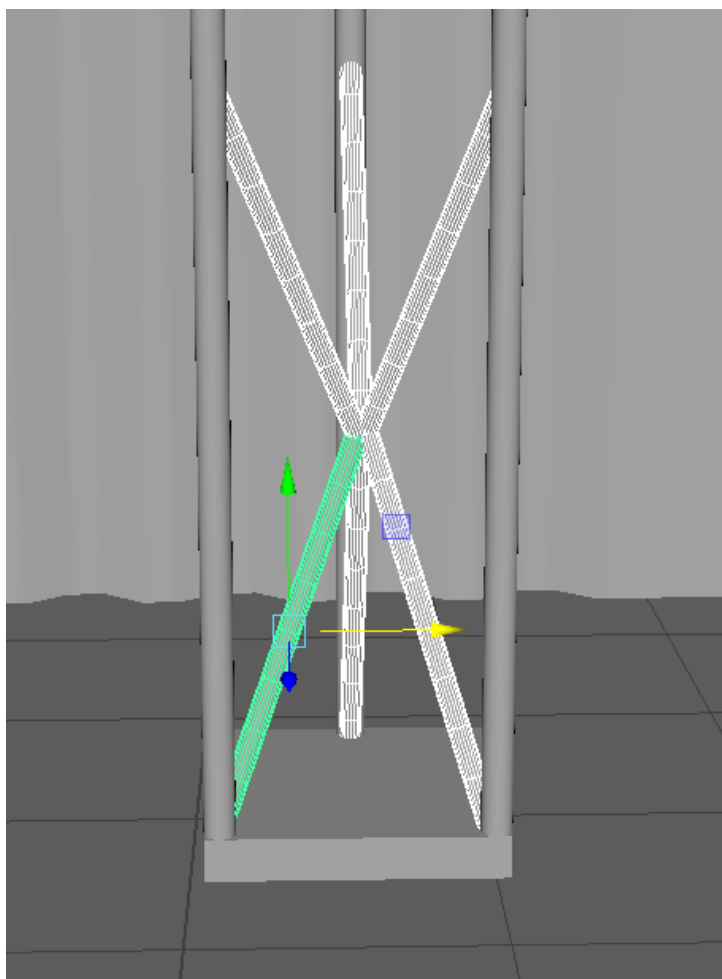


Figur 157: Blue Box 3D modell. Platå over den blå boksen.

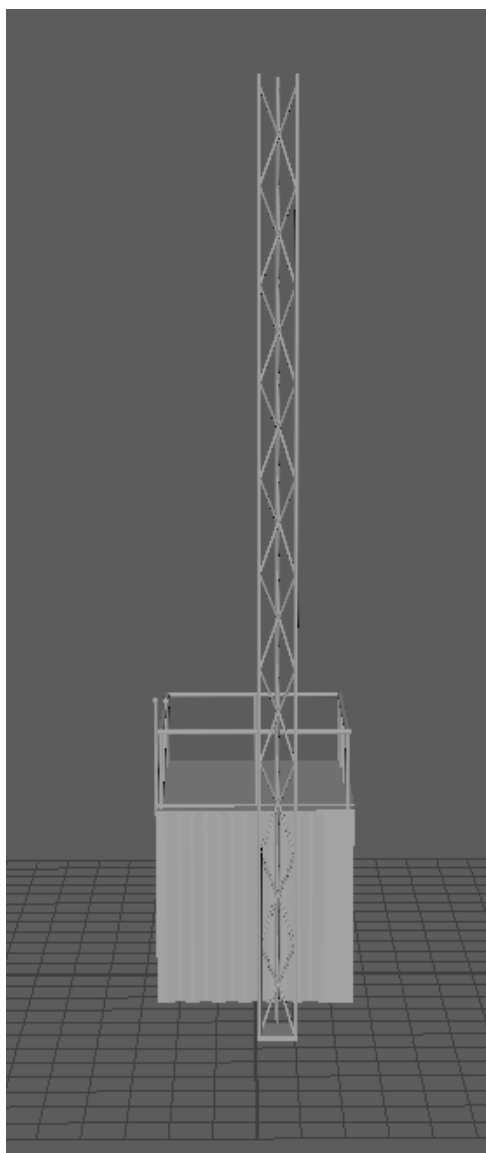
Figur 158 viser det høye stativet ved siden av den blå boksen. Ut i fra bildene ser det ut til at stativet når omtrent 2,5-3 ganger så høyt som boksen. Laget med lange sylindere som står loddrett, og mindre som er rotert for å møtes i midten av de tre lange stolpene.



Figur 158: Blue Box 3D modell. Starten på det høye stativet.
Det er litt uklart hvordan mønsteret er mellom de tre høye stolpene ut i fra bildene, men det ser noe slikt ut som på Figur 159.

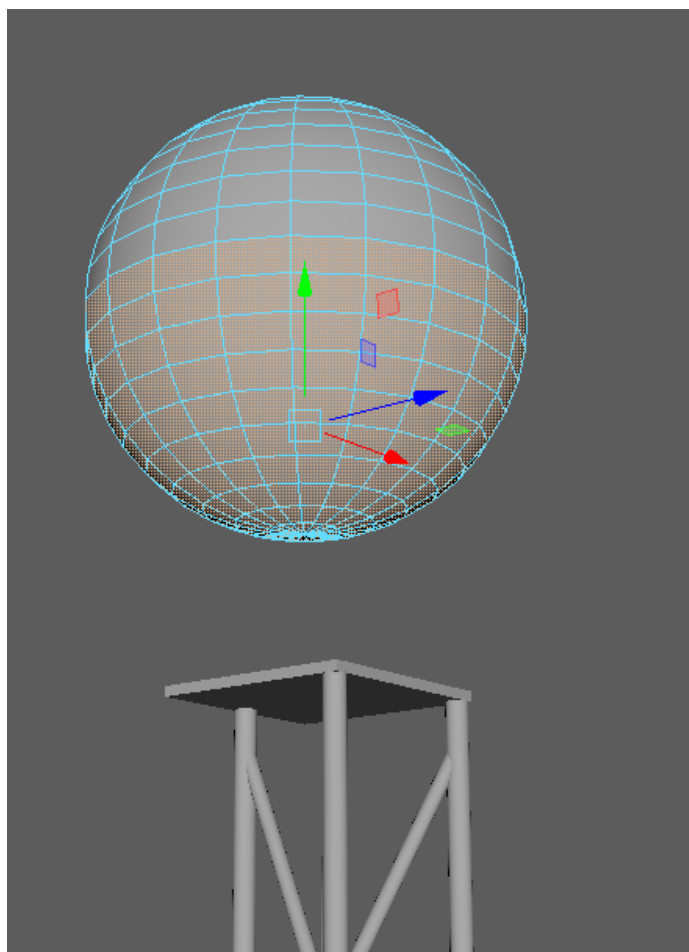


Figur 159: Blue Box 3D modell. Mønster mellom de tre høye stolpene på det høye stativet.
Figur 160 viser mønsteret kopiert hele veien opp stativet.

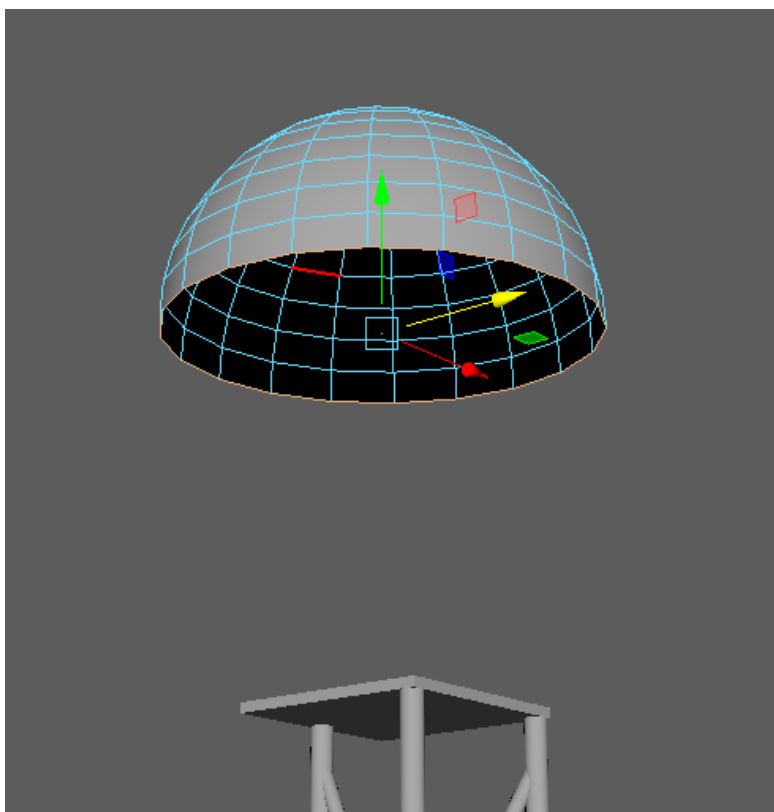


Figur 160: Blue Box 3D modell. Mønsteret massekopiert.

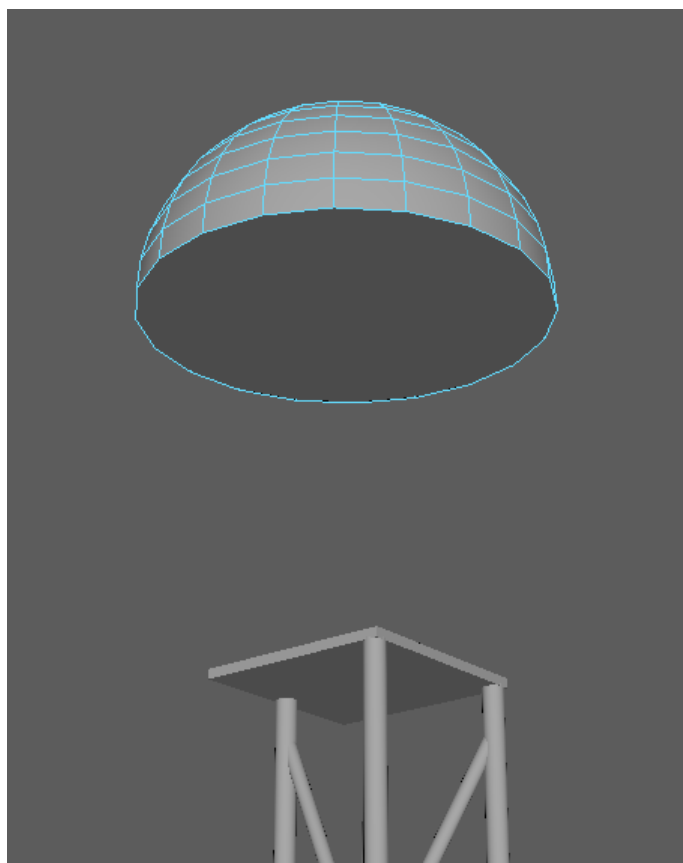
Figur 161 viser toppen av stativet, med en kule som skal bli til en slags halvkule på toppen. Viser nederste halvparten av kulen er valgt i face mode, for å så slettes.



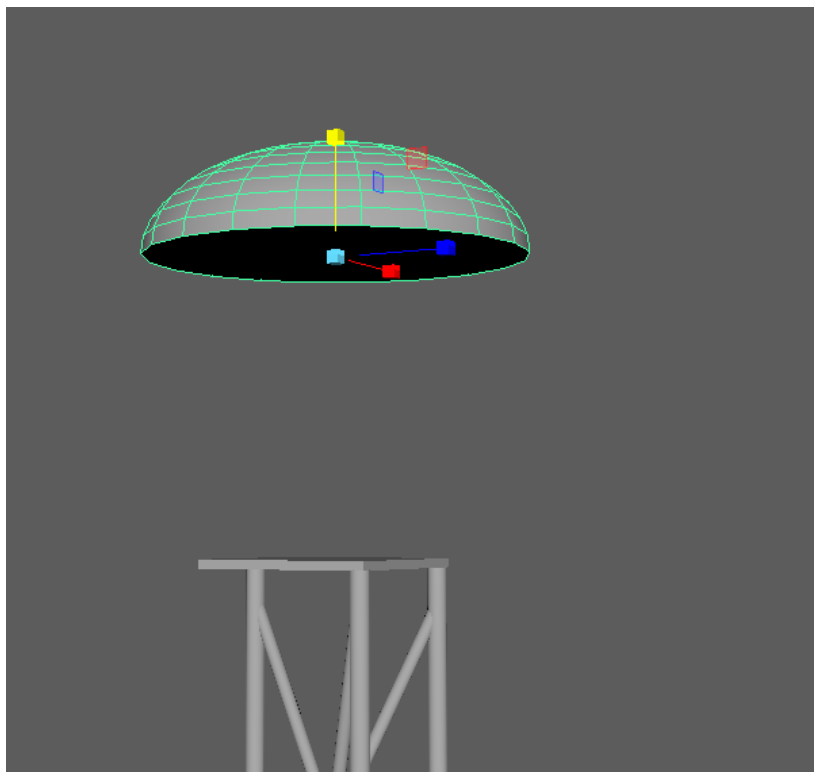
Figur 161: Blue Box 3D modell. Kule på toppen av det høye stativet.
Figur 162 viser den nederste sirkelen med edges valgt, den som lager den åpne hullet.



Figur 162: Blue Box 3D modell. Nederste sirkel i edge mode.
Deretter ble fill hole funksjonen til å fylle hullet, som vist på Figur 163.

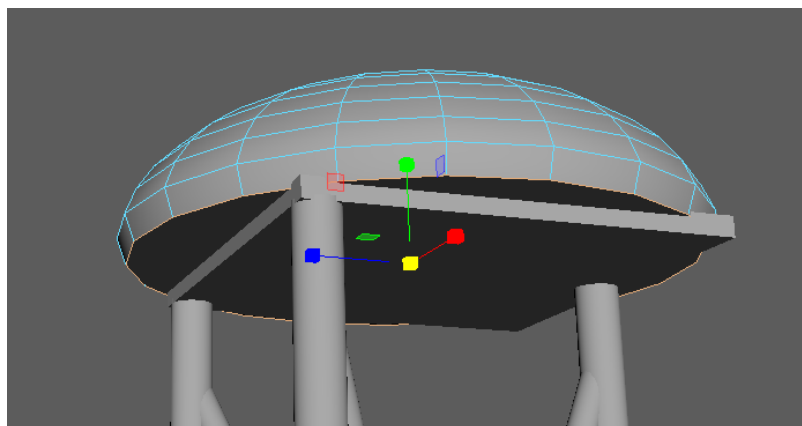


Figur 163: Blue Box 3D modell. Fill hole funksjon.
Figur 164 viser halvkulen tynnet ut ved å skalere høyden.



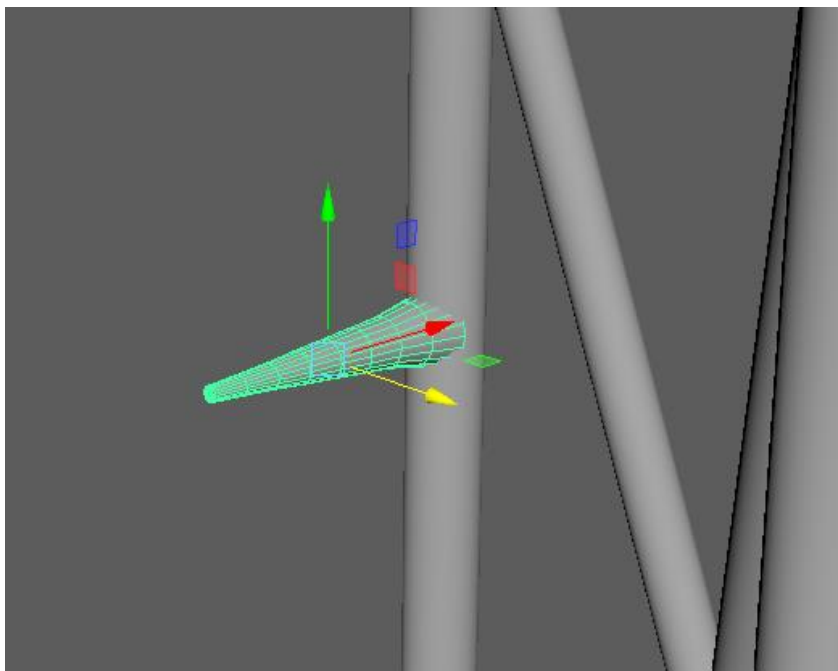
Figur 164: Blue Box 3D modell. Halvkulen tynnet ut.

Figur 165 viser den nederste sirkelen av edges av skalert litt inn for å lage en mykere kant.



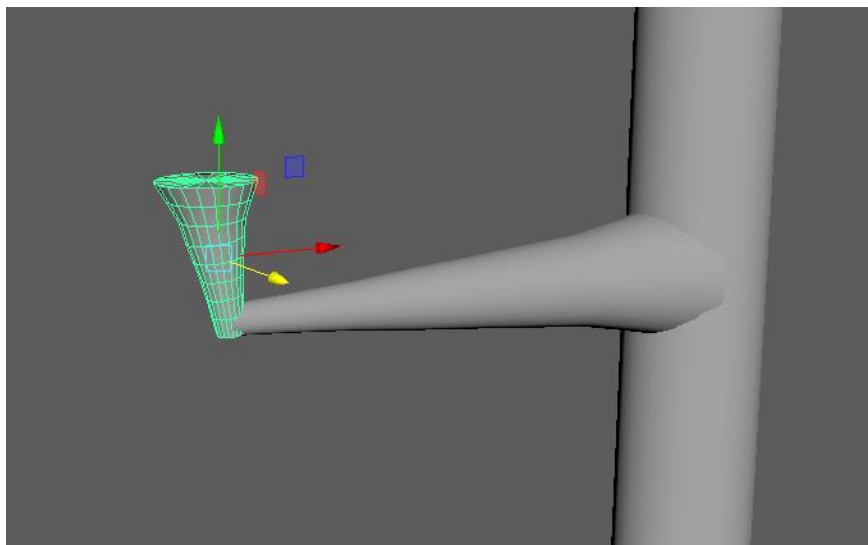
Figur 165: Blue Box 3D modell. Nederste sirkel i edge mode skalert inn.

Figur 166 viser begynnelsen av et stativ som kameraene/målerne skal stå på. Laget med en sylinder som er skalert med sirkler av edges, for å gjøre den mindre mot enden.



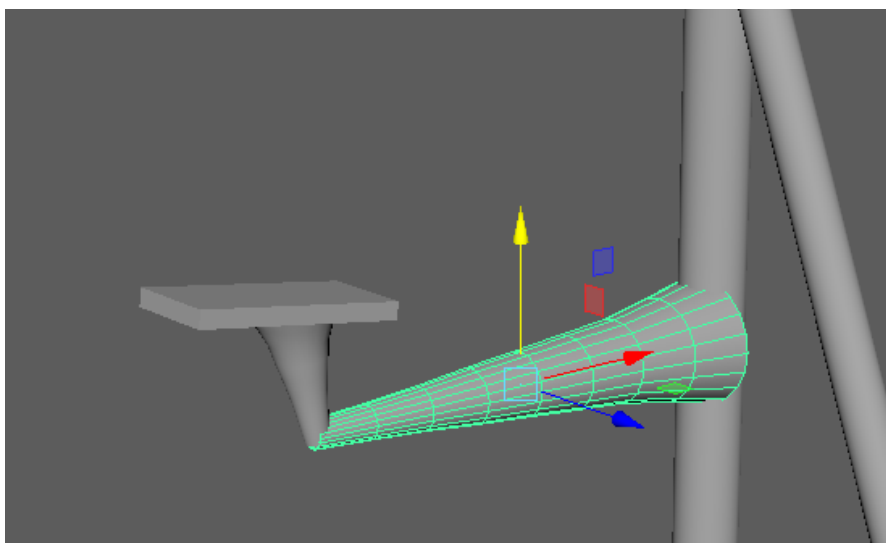
Figur 166: Blue Box 3D modell. Begynnelse på stativ til kamera/måler.

Figur 167 viser den andre delen av stativet til kameraene/målerne, laget på samme måte som nevnt for det forrige bildet.

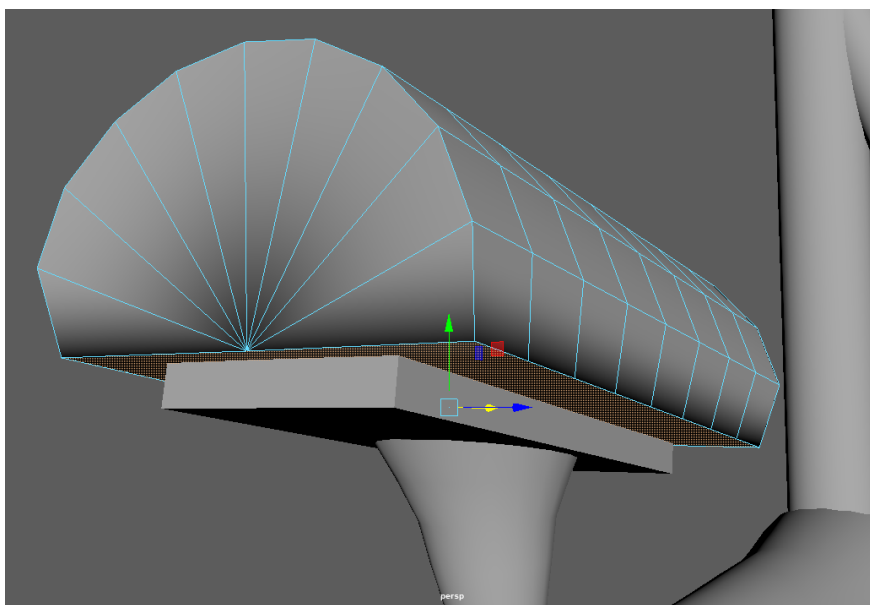


Figur 167: Blue Box 3D modell. Andre del av stativ til kamera/måler.

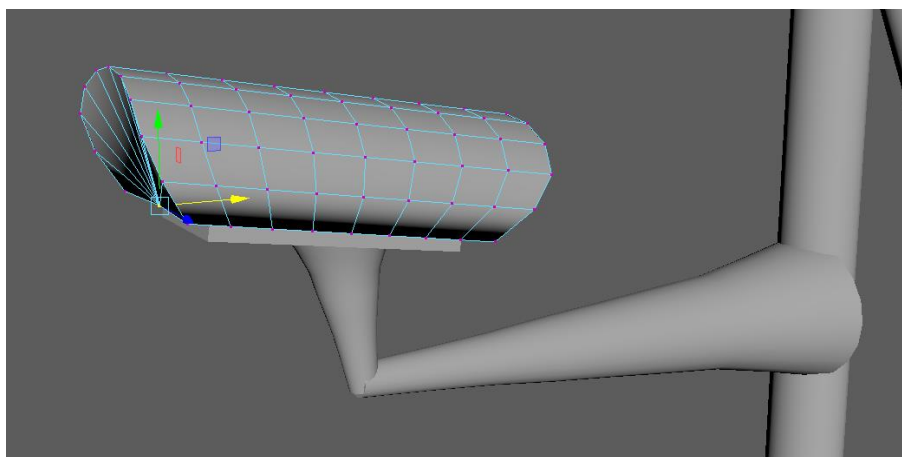
Etter det kommer et slags platå som kameraet ligger på. Laget med en tynn kube, som er skalert ned i bredden, som vist på Figur 168.



Figur 168: Blue Box 3D modell. Platå til stativ for kamera/måler.
Kameraet/måleren er laget med en sylinder, så ble nedre halvdel av faces valgt, og fjernet, som vist på Figur 169.

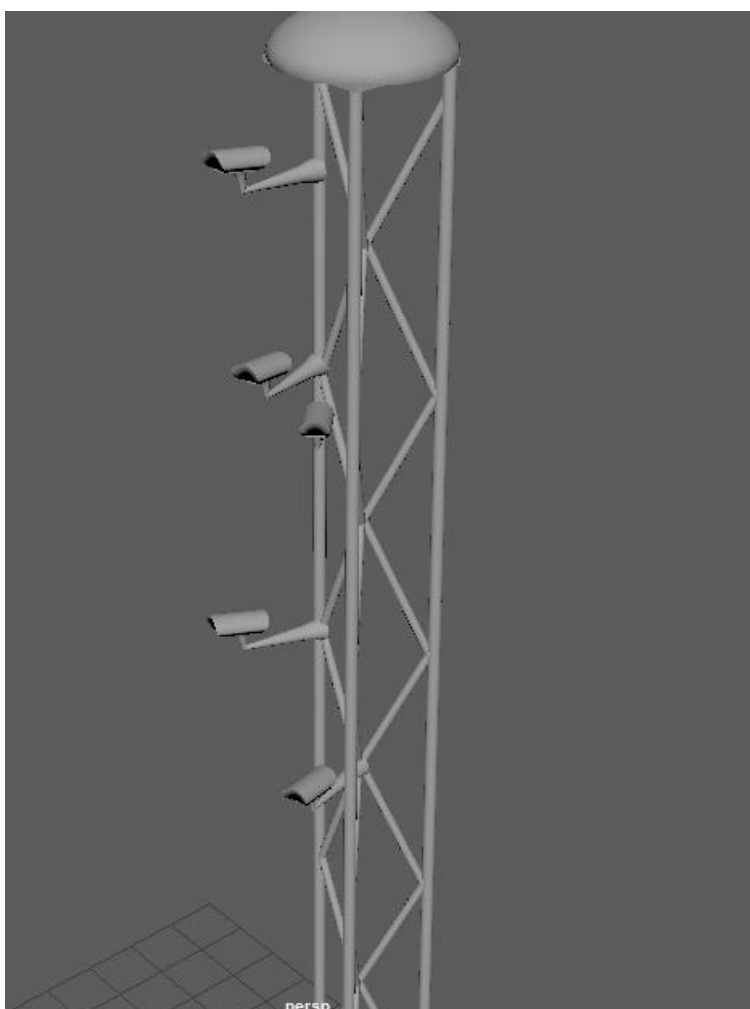


Figur 169: Blue Box 3D modell. Starten på "kamera/måler til det høye stativet.
Deretter ble edges skalert ned mot bunnen av halvsylinderen, for å forme det som et kamera. Også den ene vertexen som er valgt måtte flyttes siden den ikke kom med i edge mode, vist på Figur 170.



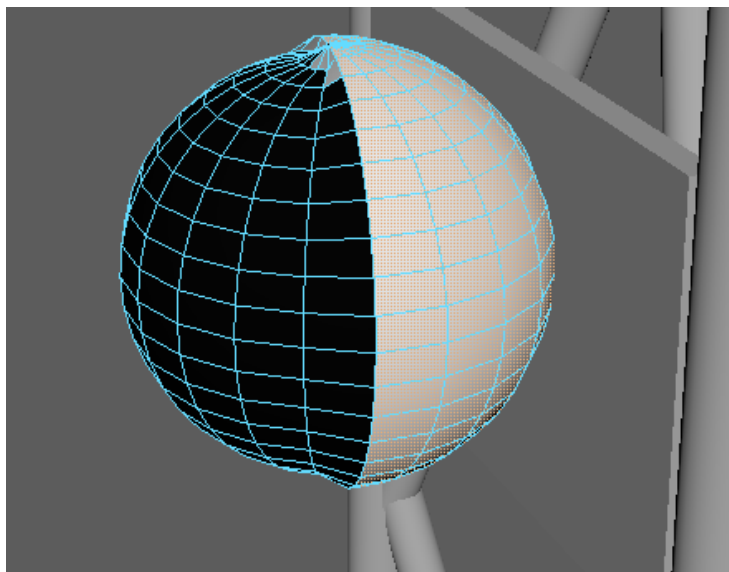
Figur 170: Blue Box 3D modell. Vertex justert.

Figur 171 viser flere kameraer/målere kopierte og rotert for å ligne bildene av det ferdige stativet.



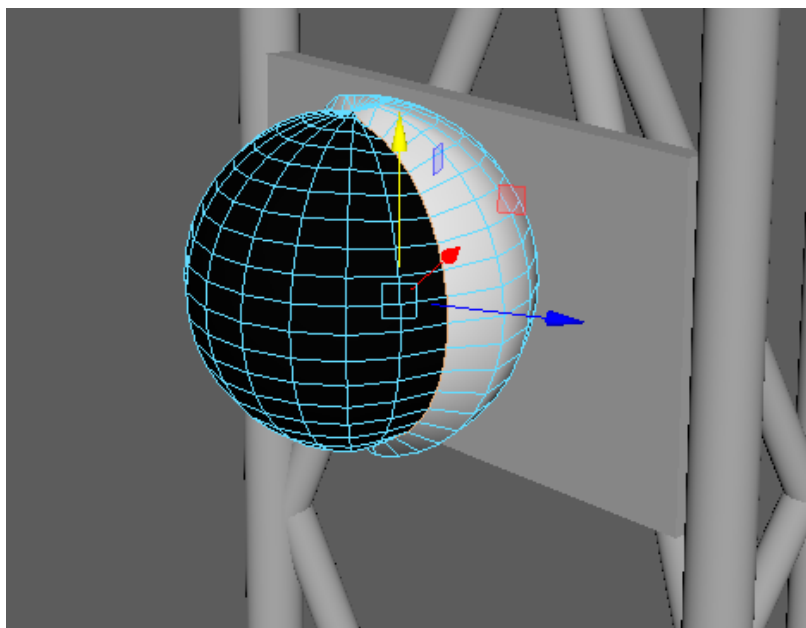
Figur 171: Blue Box 3D modell. Kameraer/målere kopierte.

Figur 172 viser et kamera/måler som så annerledes ut. Laget med en kule, hvor halvparten av faces er minsket litt, som skal være linsen.



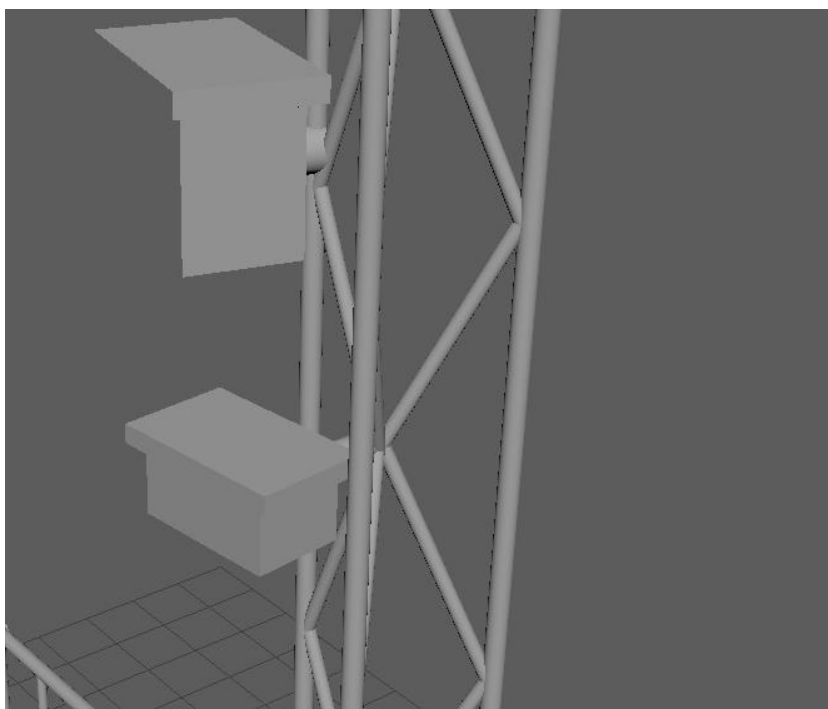
Figur 172: Blue Box 3D modell. Start på annet kamera/måler.

Figur 173 viser enkelte faces langs enden av linsen justert for at det skulle se bedre ut. I tillegg til noen faces som ikke ble valgt som du kan se i bildet over.

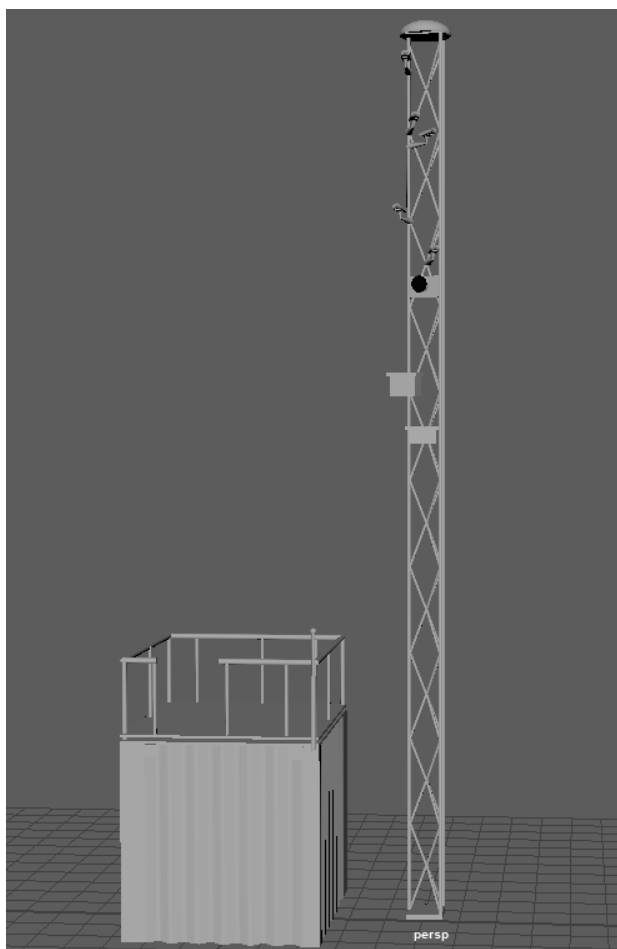


Figur 173: Faces justert på annet kamera/måler.

Figur 174 viser boksene som er plassert under kameraene/målerne på stativet. Laget med kuber som er justert i størrelse. Ut ifra bildereferansene er det ikke helt synlig hva boksene er for noe, og er derfor laget ganske generisk.

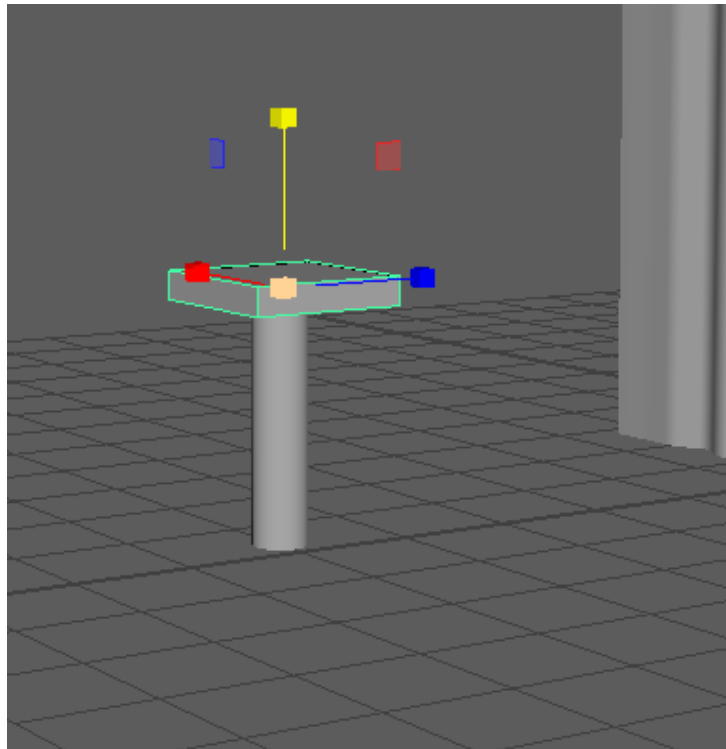


Figur 174: Blue Box 3D modell. Bokser under kameraene/målerne på det høye stativet.
Figur 175 viser et helhetsbilde av stativet ved siden av den blå boksen.



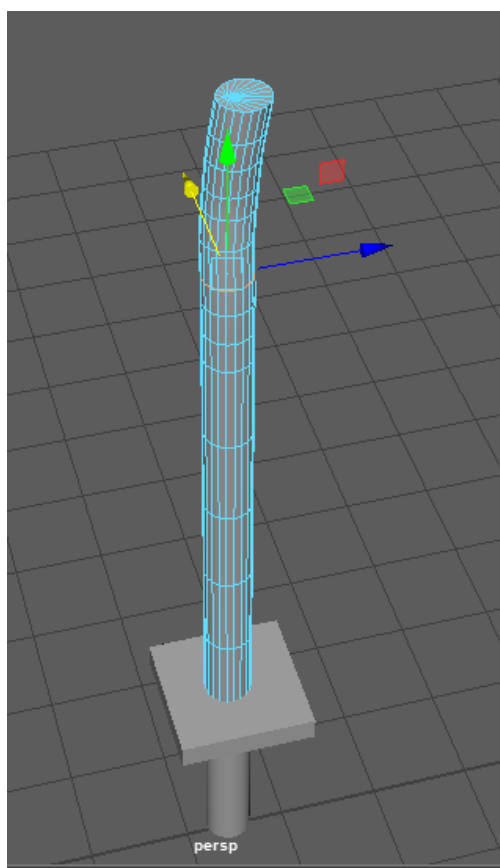
Figur 175: Blue Box 3D modell. Helhetsbilde av det høye stativet.

Figur 176 viser starten av et mindre kamera/måler- stativ, og treet som støtter det, som står på andre siden av den blå boksen. Laget med en simpel tynnet kube og sylinder.

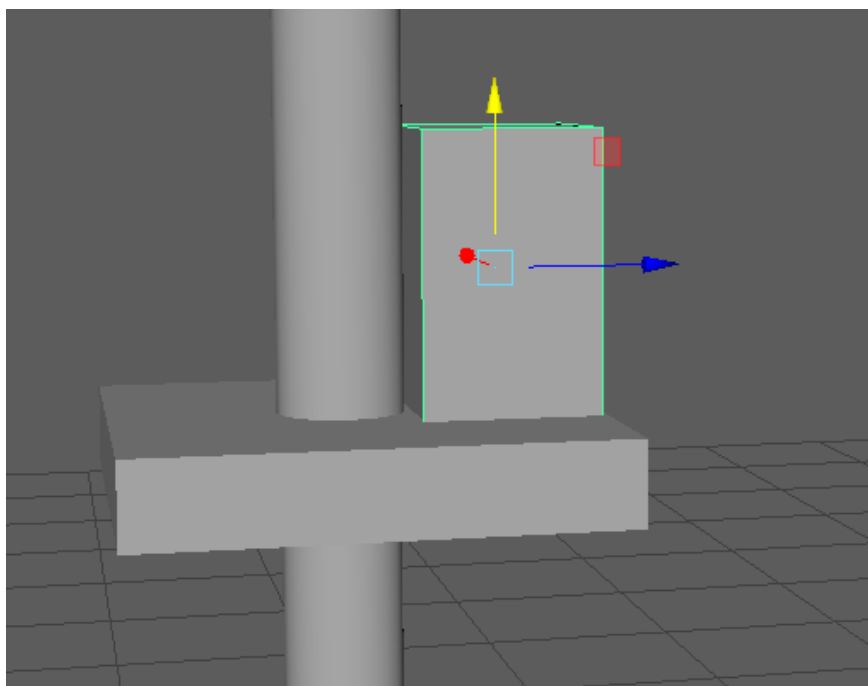


Figur 176: Blue Box 3D modell. Starten på lavt kamera/måler- stativ og treet som det støttes på.

Figur 177 viser treet som stativet er støttet ved siden av. Det er en sylinder hvor enkelte sirkler med edges er flyttet til siden for lage formen til treet.

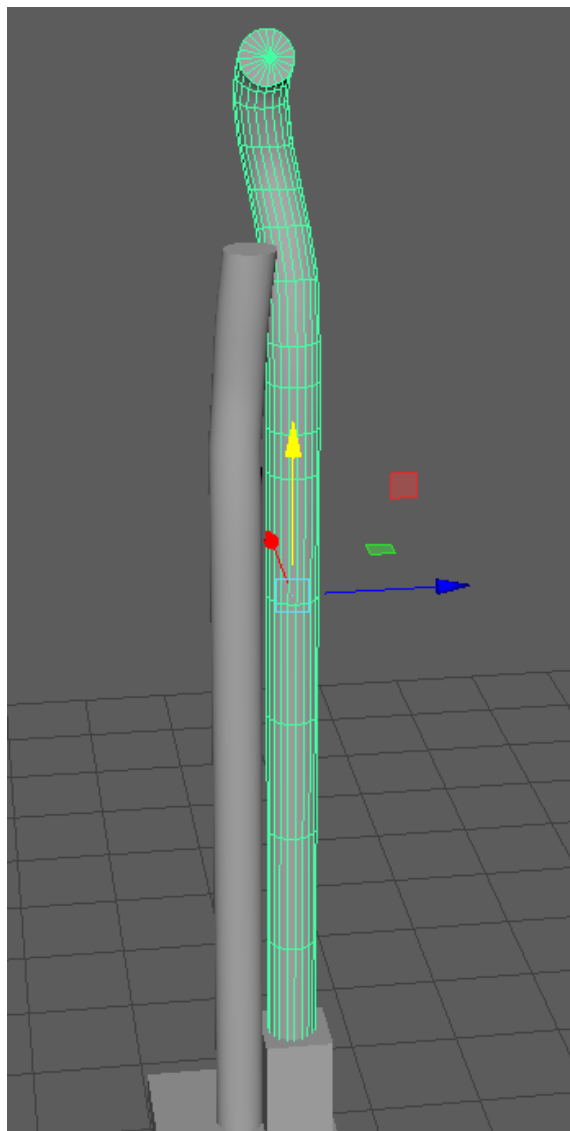


Figur 177: Blue Box 3D modell. Treet ved som stativet støttes på.
Figur 178 viser bunnen av stativet, som er laget med en simpel avlang kube.



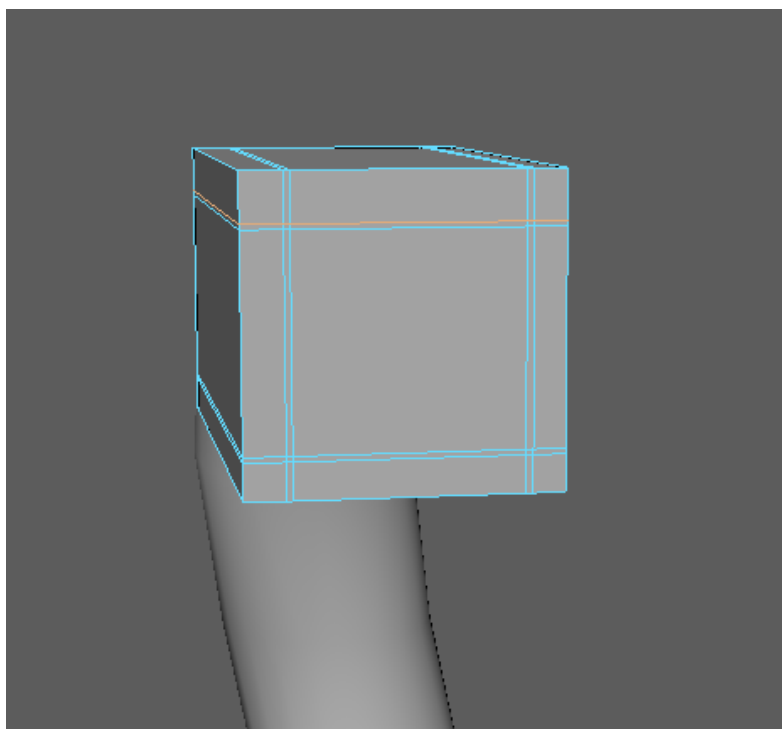
Figur 178: Blue Box 3D modell. Bunnen av stativet.

Figur 179 viser stativet som står lent inntil treet. Laget på samme måte som treet, i tillegg til at noen sirkler av edges på toppen er både rotert og flyttet på.

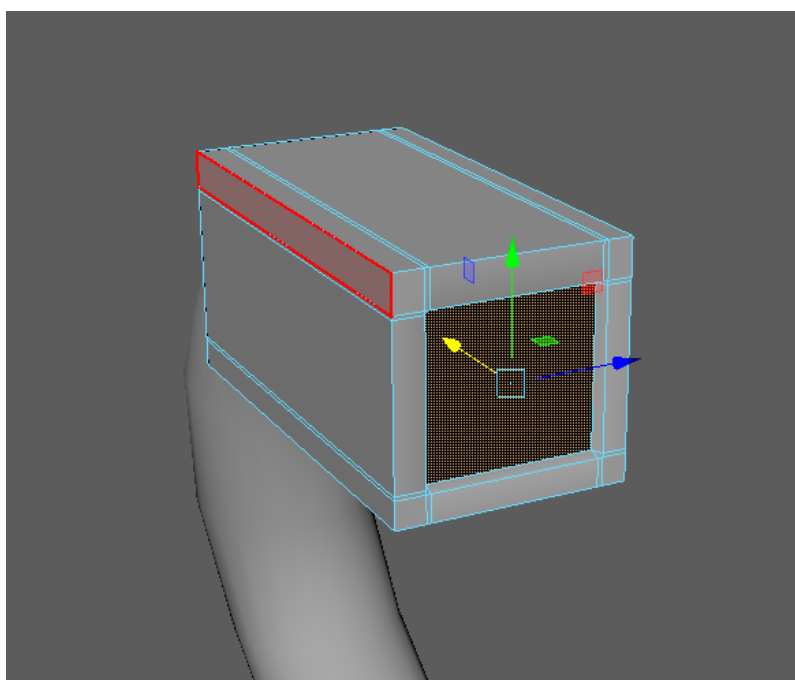


Figur 179: Blue Box 3D modell. Stativet som står støttet av treet ferdig.

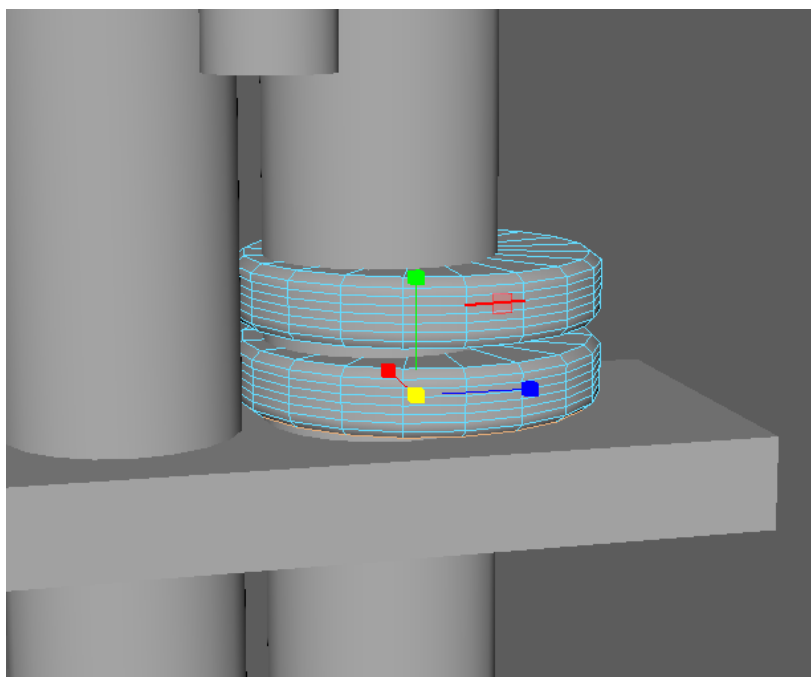
Figur 180 viser et kamera/måler som er festet på tuppen av stativet. Laget med en avlang kube, som også har lagt til ekstra edge loops.



Figur 180: Blue Box 3D modell. Kamera/måler på det lave stativet.
Figur 181 viser at den midterste facen er så flyttet litt innover, som skal være linsen.

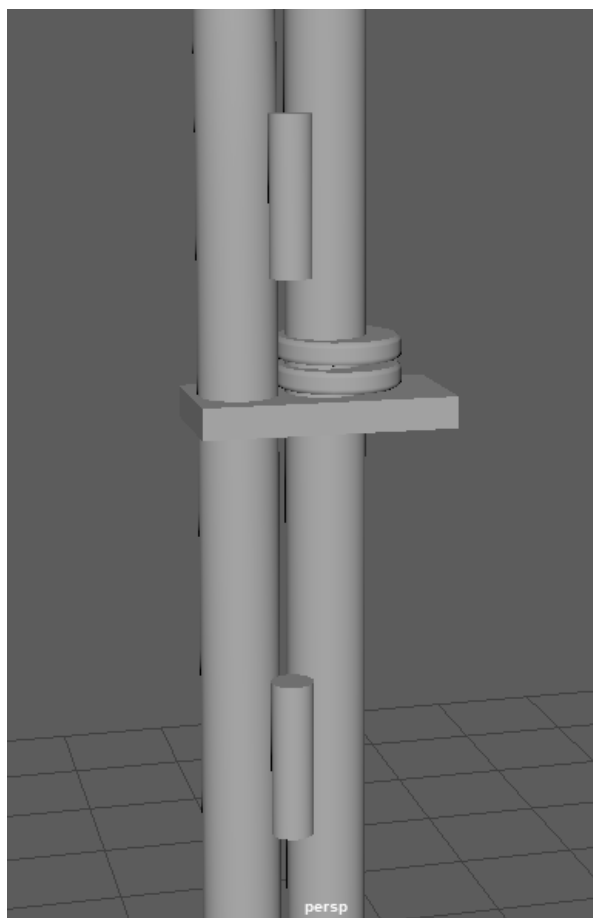


Figur 181: Blue Box 3D modell. Face justert på kamera/måler til det lave stativet.
Figur 182 viser slindere som hjelper å støtte stativet. Sirkelen av edges på kantene er myknet litt ved å dra de litt innover.



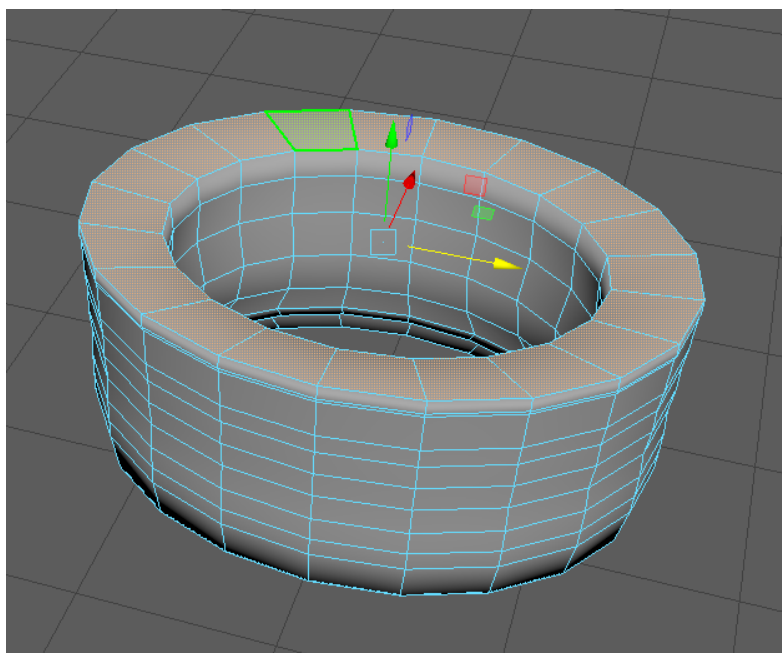
Figur 182: Blue Box 3D modell. Ekstra støtter til det lave stativet.

Figur 183 viser sylindere som er plassert mellom treet og stativet, som skal støtte de mot hverandre.



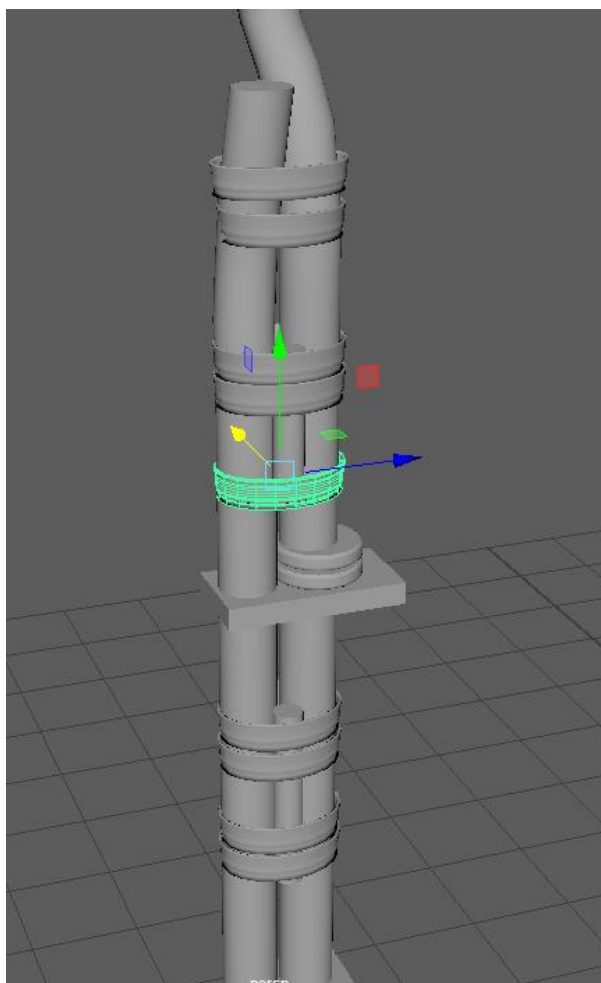
Figur 183: Blue Box 3D modell. Ekstra støtter mellom det lave stativet og treet.

Figur 184 viser gaffateip som går rundt treet og stativet for å støtte det mer. Laget med en polygon torus hvor det er justert på edges og faces for å gjøre innsiden større og utsiden minsket, for å gjøre den tynn som noen lag med gaffateip.



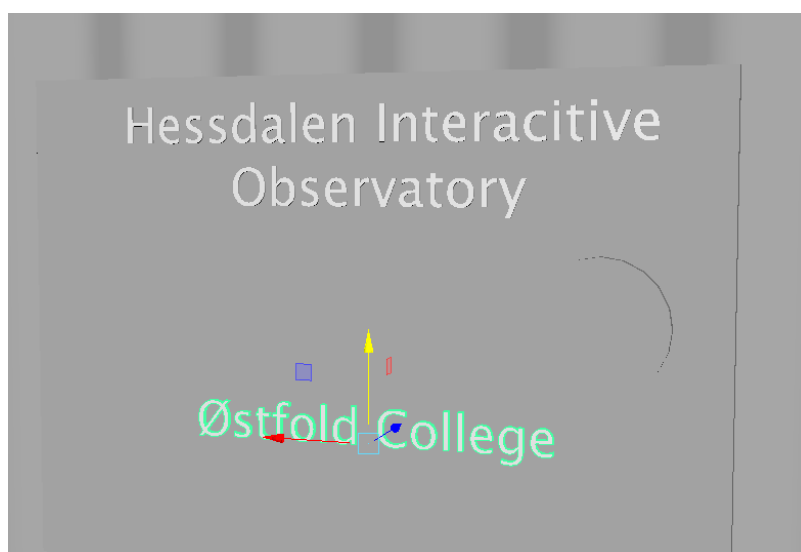
Figur 184: Gaffateip til det lava stativet og treet.

Figur 185 viser gaffateipen kopiert og justert noe enkeltvis for å passe rundt treet, stativet og støttesylinderene.



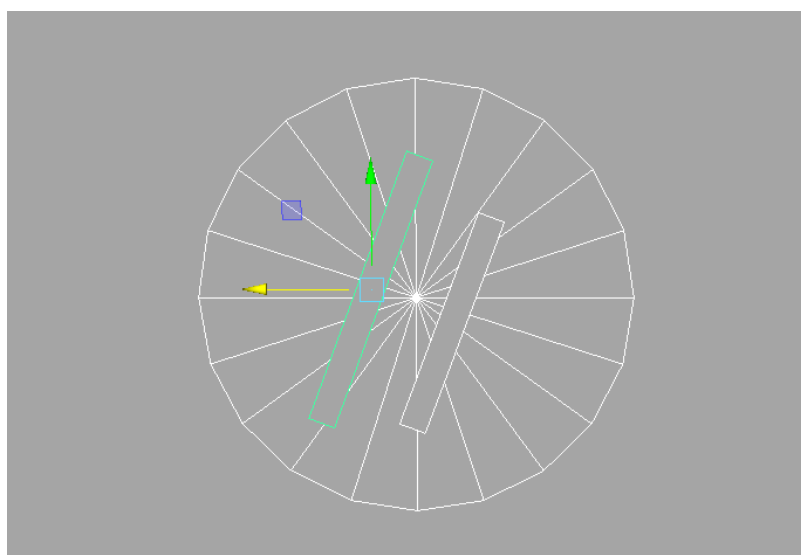
Figur 185: Blue Box 3D modell. Kopiert gaffateip.

Tilbake til den blå boksen, hvor et skilt utenpå boksen lages. Skiltet er en tynn kube. Teksten er laget med type tool, og så justert i størrelse, som vist på Figur 186. Sirklene for logoene laget med tynne sylindere.



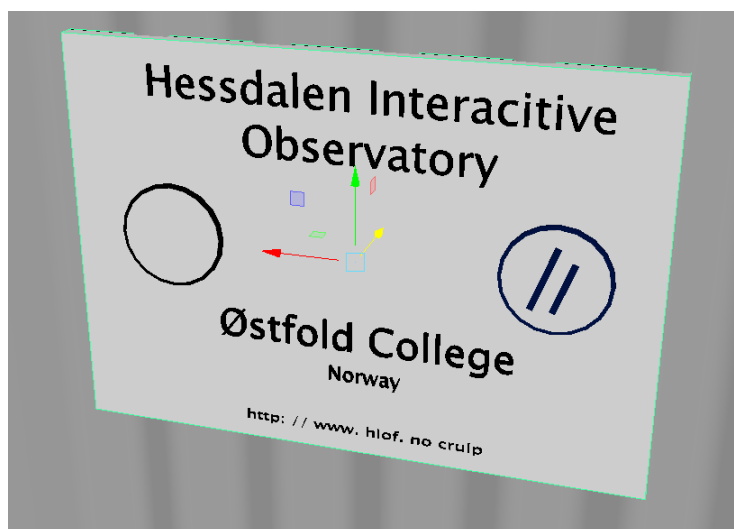
Figur 186: Blue Box 3D modell. Skilt utenpå den blå boksen underveis.

Figur 187 viser Høgskolen i Østfold sitt symbol. Laget med tynne kuber som er rotert litt til siden.



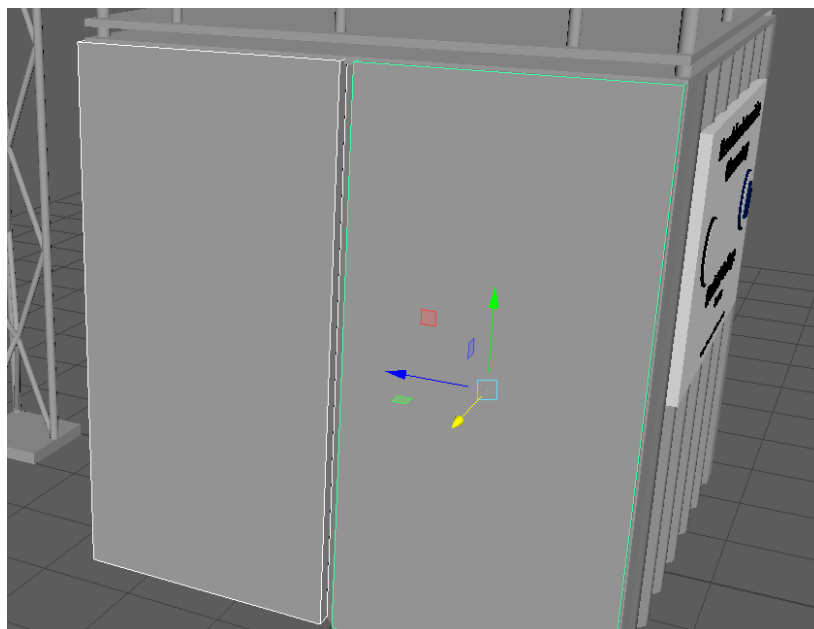
Figur 187: Blue Box 3D modell. Høgskolen i Østfolds logo på skiltet.

Figur 188 viser hvordan det ferdige skiltet ut. Den andre logoen ble for komplisert for noe som er såpass smått, i tillegg til at det var vanskelig å lese hva som stod ut i fra bildereferanser.



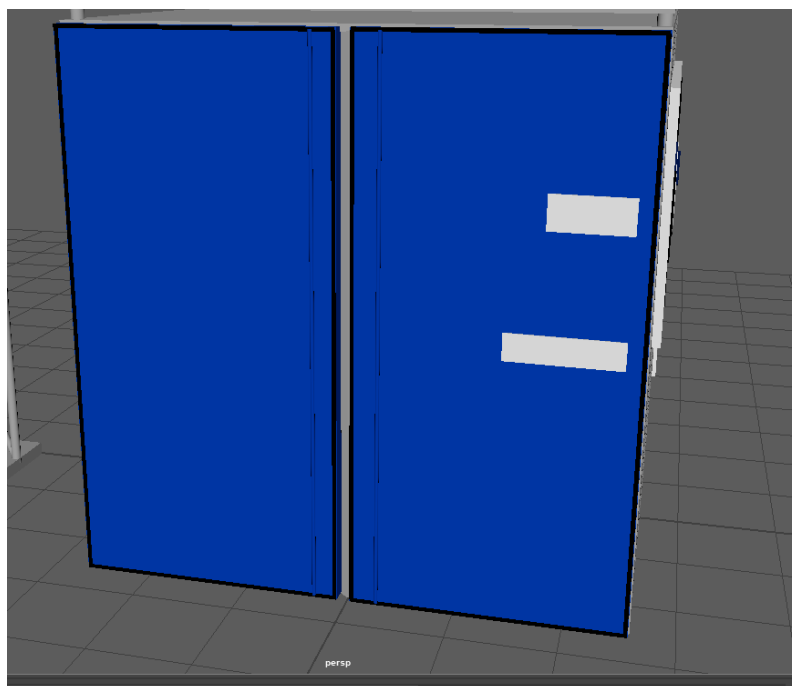
Figur 188: Skiltet utenpå den blå boksen ferdig.

Figur 189 viser dørene på den blå boksen. Laget med tynne, avlange kuber.



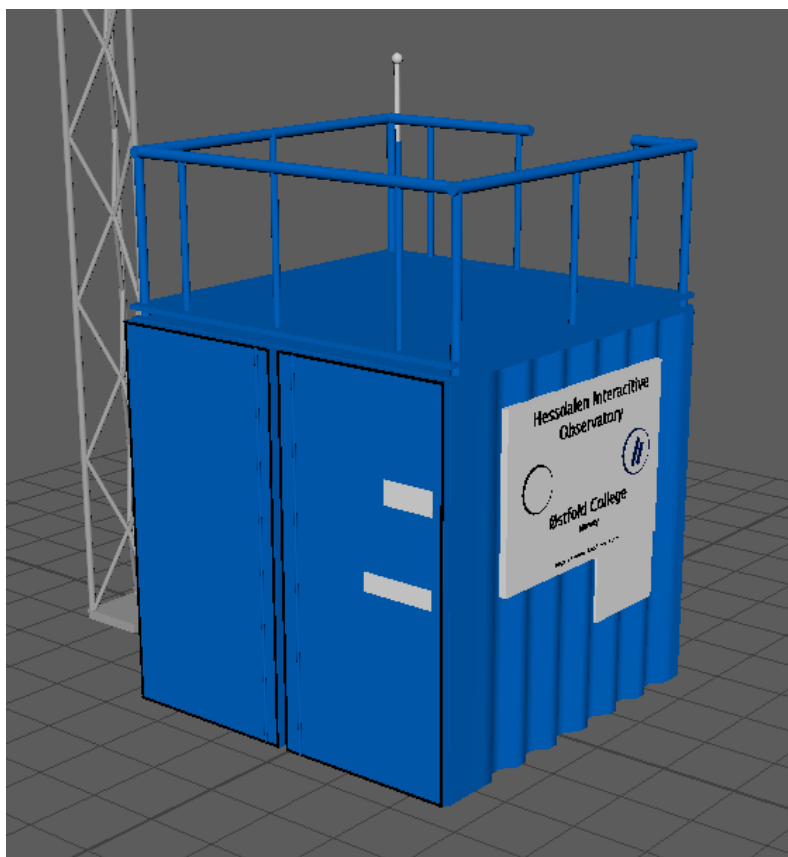
Figur 189: Blue Box 3D modell. Dørene utenpå den blå boksen.

Figur 190 viser dørhåndtakene mot midten er laget med veldig tynne sylindere. Det samme gjelder de svarte kantene rundt. Også første forsøk på fargen av den blå boksen.

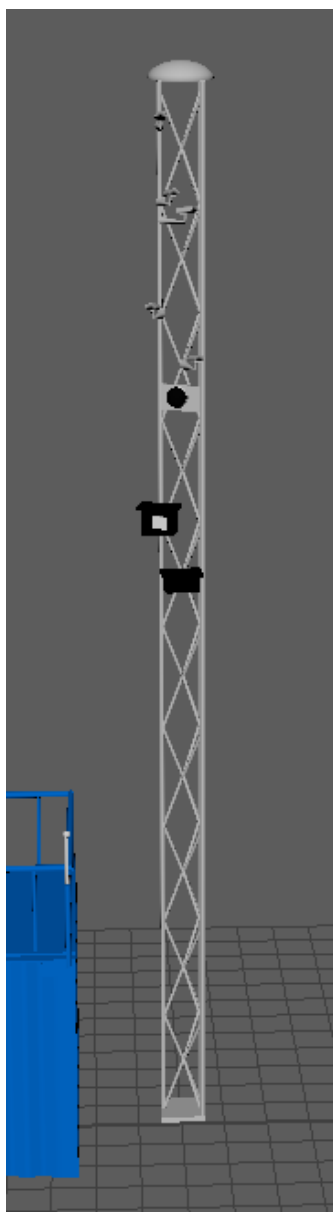


Figur 190: Blue Box 3D modell. Dørhåndtak, ramme rundt og første forsøk på blåfargen til den blå boksen.

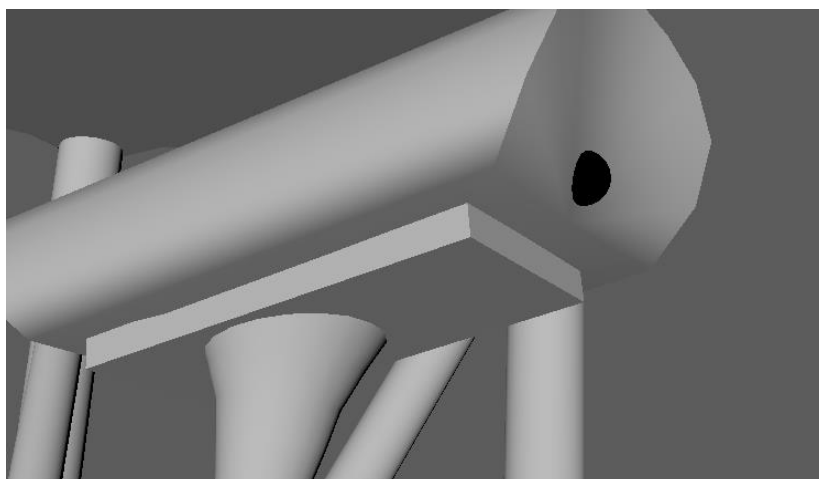
Figur 191 viser et helhetsbilde av den blå boksen. Som du kan se er blåfargen endret litt, til noe som ligner mer på den virkelige fargen.



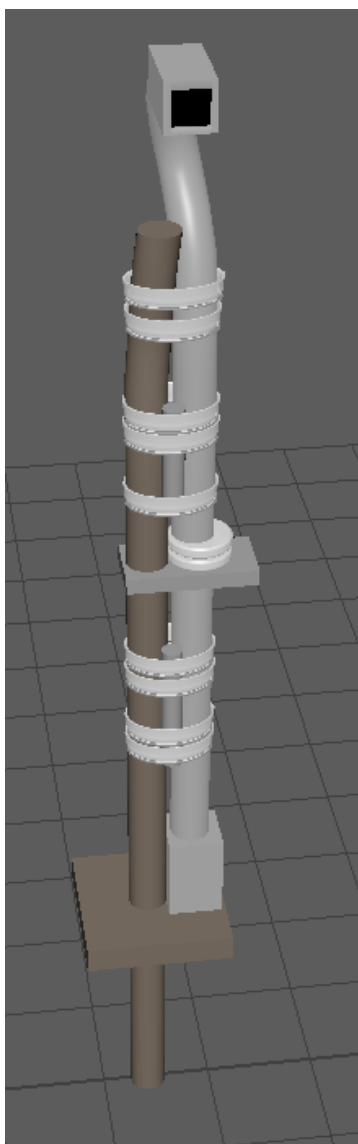
Figur 191: Blue Box 3D modell. Helhetsbilde av den blå boksen med fargeendring. Figur 192 viser et helhetsbilde av det høye stativet med materiale og farge.



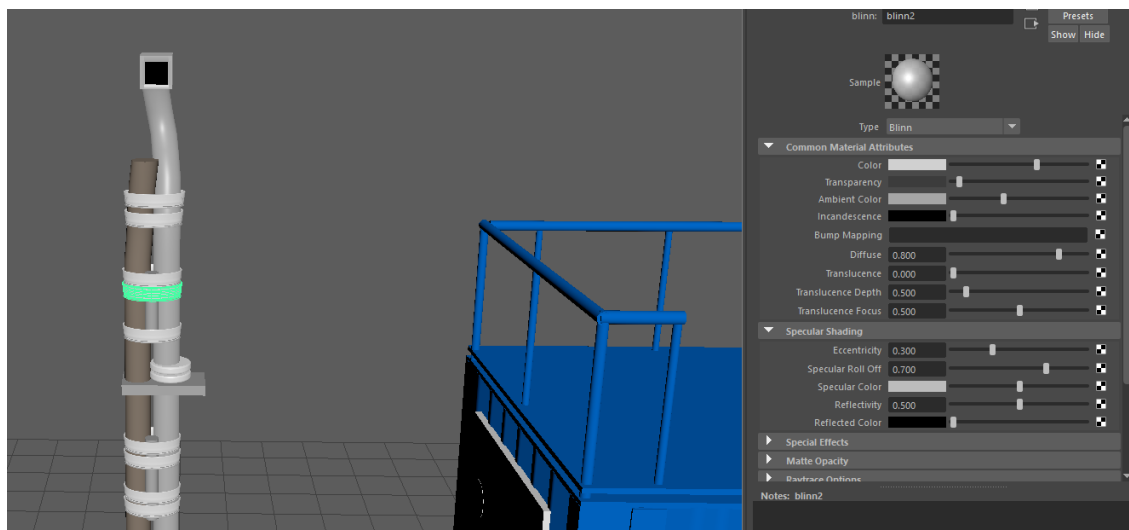
Figur 192: Blue Box 3D modell. Helhetsbilde av det høye stativet med materiale og farge.
Figur 193 viser et nærbilde av kamera/måler på det høye stativet, med linse som ble laget med en kule.



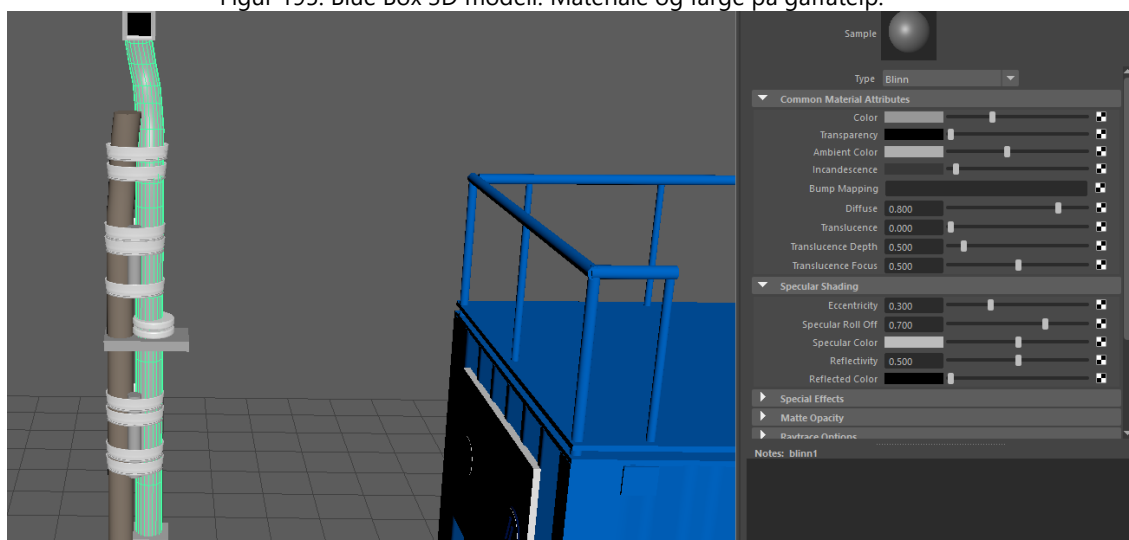
Figur 193: Nærbilde av kamera/måler på det høye stativet.
Figur 194 viser et helhetsbilde at det korte stativet med materiale og farge.



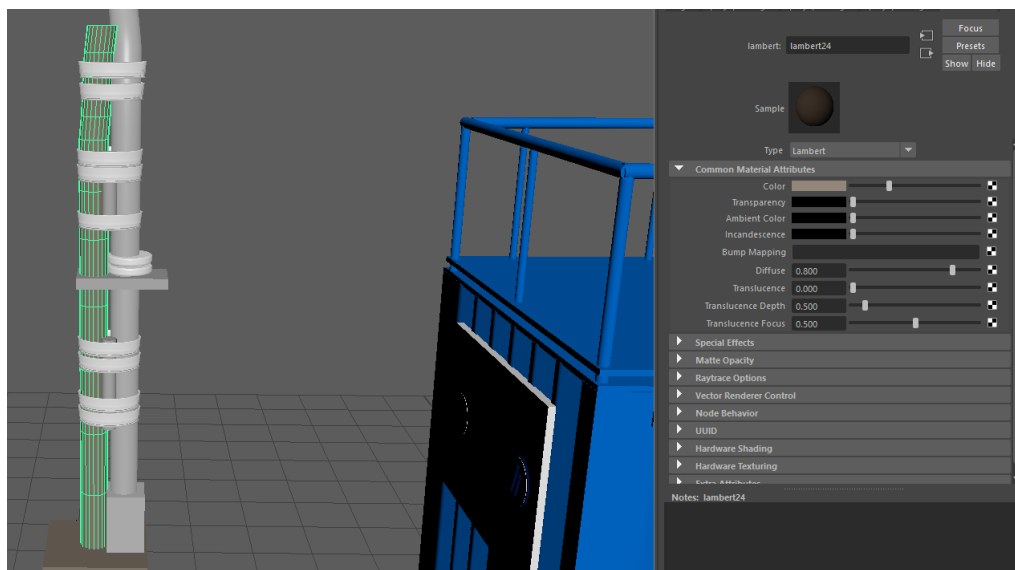
Figur 194: Blue Box 3D modell. Helhetsbilde av kort stativ med materiale og farge. Under vises materiale og farge på de ulike delene av Blue Box, sett på Figur 196-199.



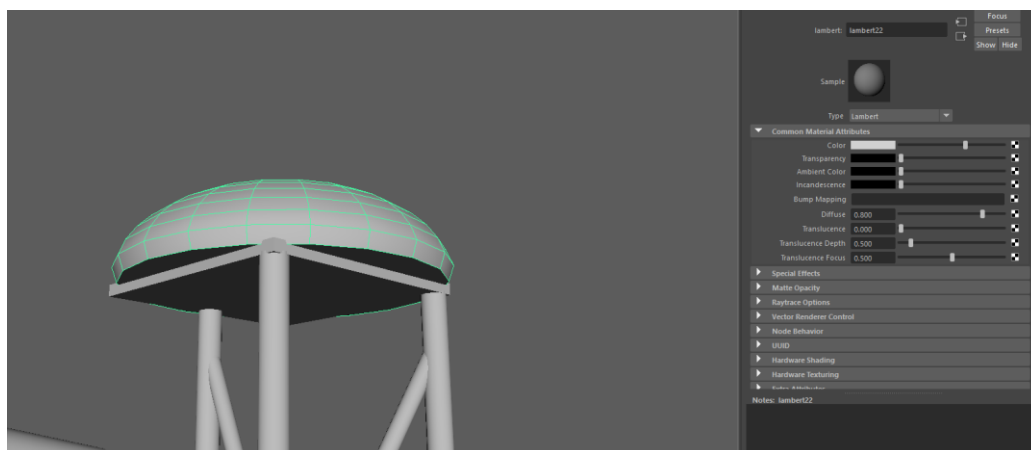
Figur 195: Blue Box 3D modell. Materiale og farge på gaffateip.



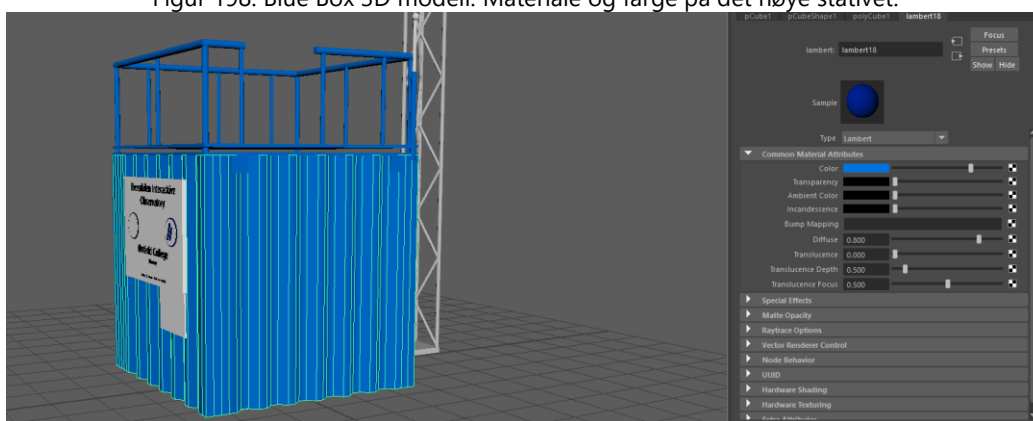
Figur 196: Blue Box 3D modell. Materiale og farge på det lave stativet.



Figur 197 Blue Box 3D modell. Materiale og farge på treet



Figur 198: Blue Box 3D modell. Materiale og farge på det høye stativet.



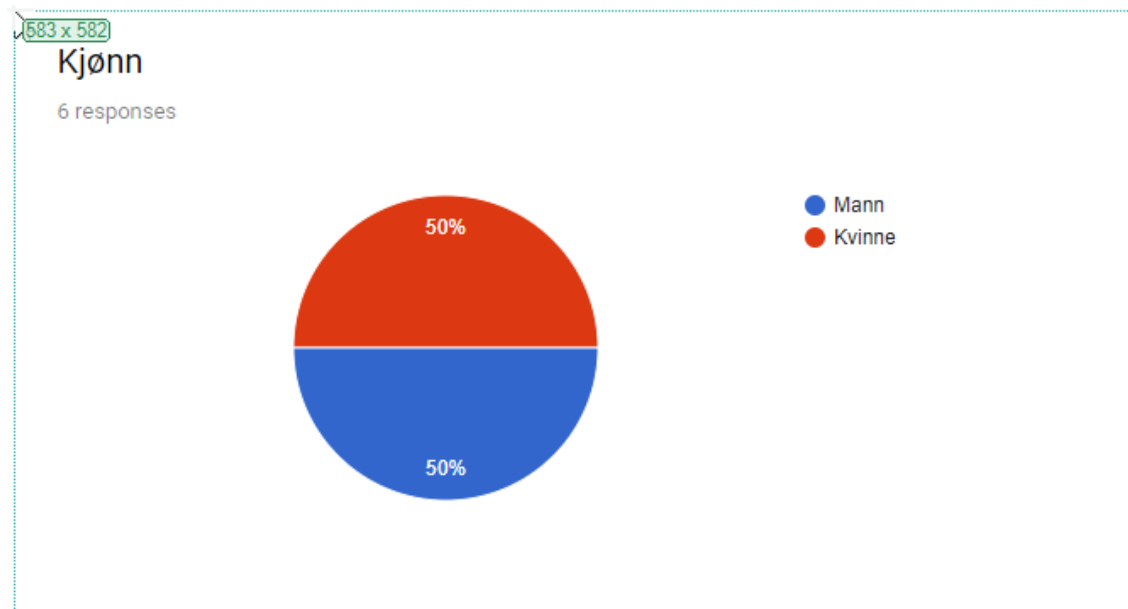
Figur 199: Blue Box 3D modell. Materiale og farge på den blå boksen.

9.2. Brukertest resultater

Brukertest 1

Resultat av brukertest 1 var som følger:

Kjønn:



Alder:

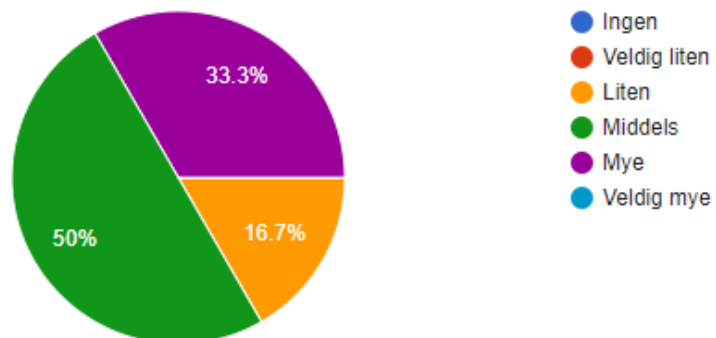


Datakunnskaper:



Datakunnskaper

6 responses



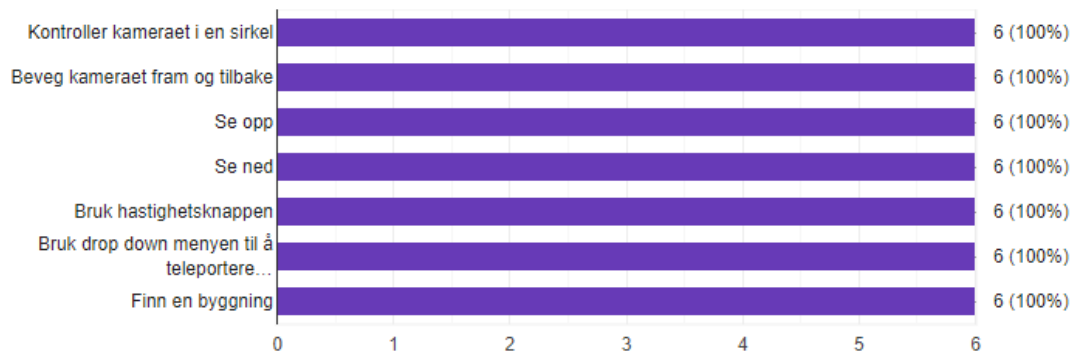
Kontroller brukertest:



Kontroll brukertest(marker etter gjort)



6 responses



Beskrivelse av prosessen:

✎ Beskriv prosessen mens den foregår med spørsmålet over

6 responses

Brukte left-mouse button til å kontrollere kameraet i en sirkel. Brukte WASD til å bevege fram og tilbake. Brukte left-mouse button til å se opp og ned. Brukte shift, som det står på siden til hastighet. Fant fort drop-down menyen til å teleportere seg til et sted. Fant fort fram til en 3D-modellert bygning.

brukte venster mus for å se meg rund.
brukte piltastene og WASD for å flytte på meg.
hold venstre klick mens jeg såg både opp og ned.
holdt shift, gikk kjappere.
teleporterte meg selv ved bruk av dropdown til UFO camp.
det var en bygning rett ved siden av UFO campen.

Slet litt med å kontrollere kameraet. musen reagerer merkelig. og farten på w ble litt for fort. tok litt tid før hun klarte å komme inn i hvordan kontrollene oppførte seg.

klarte lett å bevege seg gjennom kartet, alle oppgavene ble løst med litt henvisning til kontrollene. kom seg veldig lett in i kontrollene.

Bruker litt tid på å bli vant til kontrollene, og å kontrollere kameraet. klarer å bevege seg någenlunde bra, men må bli guidet gjennom.

Bruker litt tid til å venne seg til bevegelsene, klarer å få til oppgavene men med litt hjelp. synet er litt dårlig, så har problemer med å lese listen over steder i drop down menyen.

Hva var bra?

🖱️ Hva var bra(kort)

6 responses

kult ide, kontrollene er rett fram, navigasjonen er lett å forstå og å bruke

responsiviteten

Lett bevegelighet, fint kart, ser veier osv. navn er veldig bra på fjell osv. Kompasset var veldig bra og kartet var veldig bra for navigasjon.

bra manvrering, brukervenlig

veldig fint å se ting som det på kartet.

det er veldig morsomt å kunne gå inn i landskapet, og bevege seg ut i landskapet. kontrollene er bra etter litt trening.

Hva var dårlig?

Hva var dårlig(kort)

6 responses

når man sikter opp eller ned, det er et punkt hvor kamera bare går helt rundt og mister oprinnelig retning man er på.

det at jeg må holde venstre klick for å se meg rund i fullscreen, er plagsomt og i min mening unødvendig.

ikke veldig brukervenlig i hennes syn.

litt dårlig tekstur detaljer men det går greit om det er meningen.

vet ikke helt hva som er dårlig med sånt,

liten skrift for svaksynhet. boksen i høyre er for liten for dårlig syn.

Forbedringer

Forbedringer?

6 responses

muligheten kanskje å bruke 2 knappene samtidig, f.eks. opp/ned + høyre/venstre istedenfor at bare en blir aktivert på en gang. hvis det er mulig å slette kartet som ligger i bunn av 3d modelert elementer

fjern det med at jeg må holde venstre klick for å se meg rund i fullscreen

når man trykker på en knapp så blir kameraet mot byggingen/stedet som blir valgt. bevegelsene burde være sakte så man kan oppleve det bedre, navnene må være klikkbare og teleporterer deg som drop down menyen. navnene når man kommer nærmere må bli mindre og ikke så massive inn i ansiktet ditt. kontrollene blir lett glemt, og burde derfor være i kartet som en ekstra knapp.

når man kommer nærmere så kommer det opp flere informasjons områder, lettere å manuvrere seg 360 grader rundt. burde gå automatisk mot stedet man trykker på med drop down menyen. hadde vært greit å ha kontrollene nede i hjørnet et sted. navigasjons informasjonen burde være større og gi mer tydelig informasjon.

Usikker på hva som kan bli forbedret. tanke fra observatøren er en guide/kontroll mapping på kartet som kan hjelpe med å vise hva som er hva

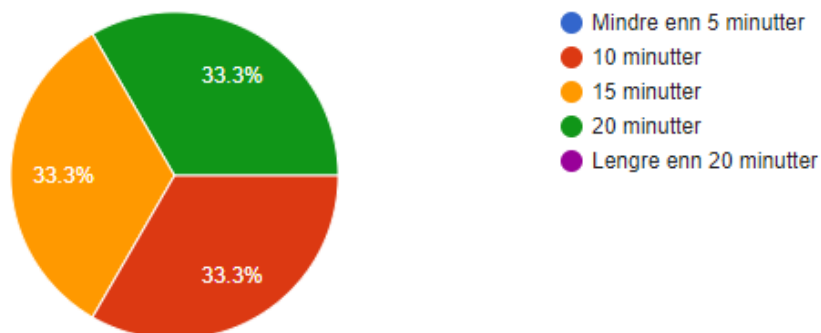
ting skjer veldig fort, bevegelsene er litt for fort på grunn av svaksynhet. dette er ikke et problem for alle, men det kan være problematisk. kontroll siden er grei men kanskje litt større?

Tid brukt



Tid brukt(Velg nærmeste)?

6 responses



Konklusjon av brukertest 1

Brukertest 1 gav oss en god ide på områdene av produktet som vi gjorde feil, og områdene som vi gjorde for det meste riktig. Vi fant ut at Ulen og tekststørrelse virkelig har noe å si, spesielt siden en av testpersonene har problemer med synet. Dette fikk oss til å forandre på skriftstørrelser og på UI elementene for å gjøre alt lett leselig for alle.

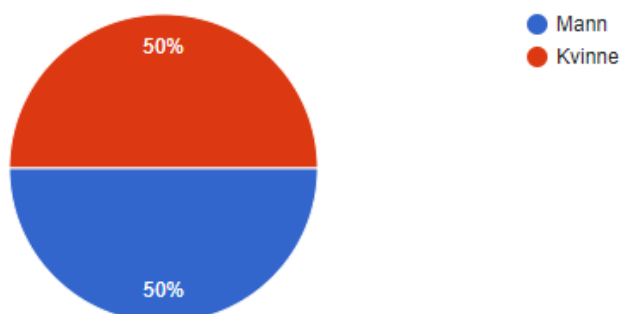
Brukertest 2

Kjønn



Kjønn

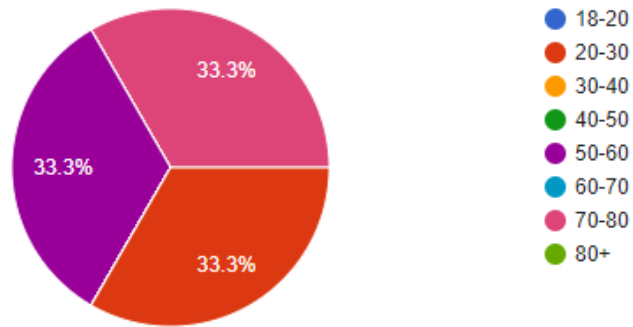
6 responses



Alder

Alder

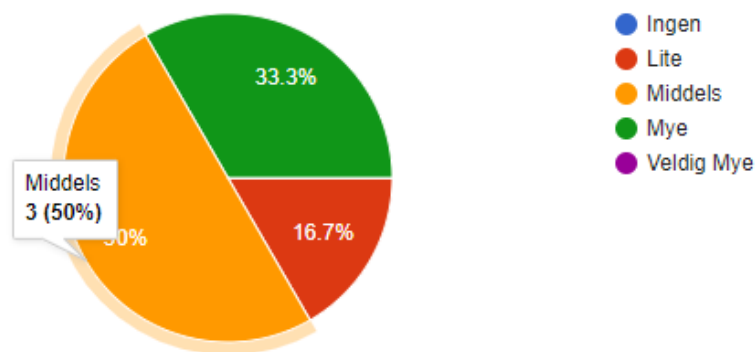
6 responses



Data Kunnskaper

Datakunnskaper

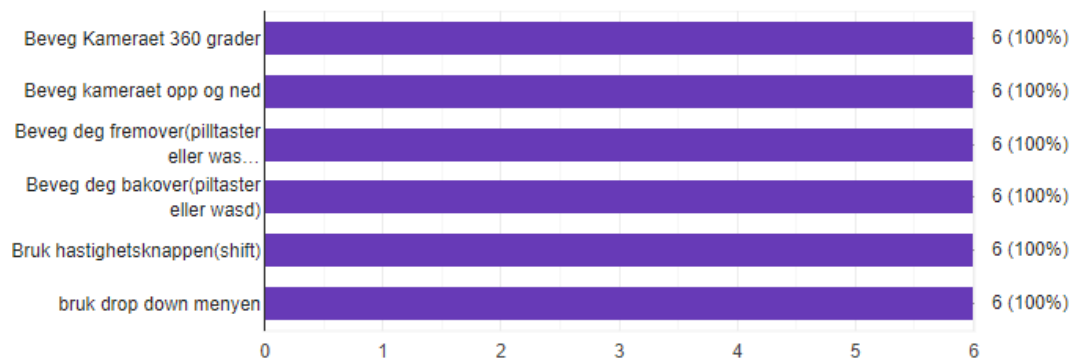
6 responses



Test av kontroller

Kontroll Test

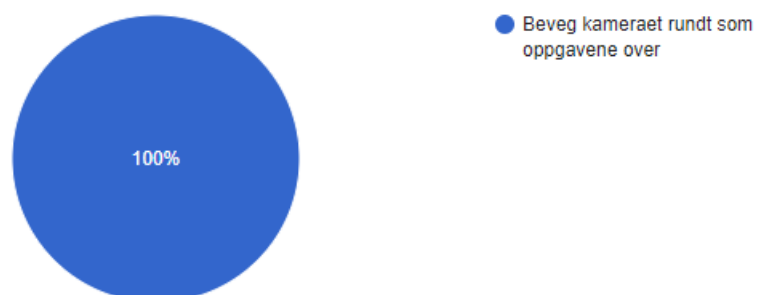
6 responses



Laptop touchpad

Laptop touchpad test

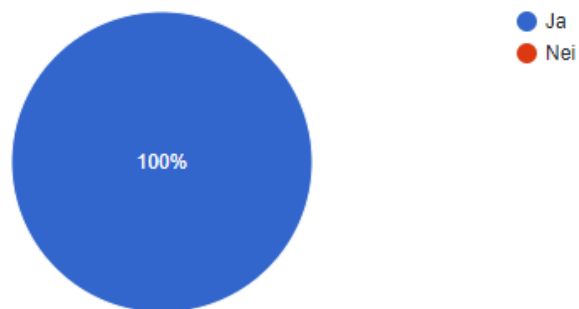
6 responses



Touchpad resultat

Fungerer touchpad kontrollene?

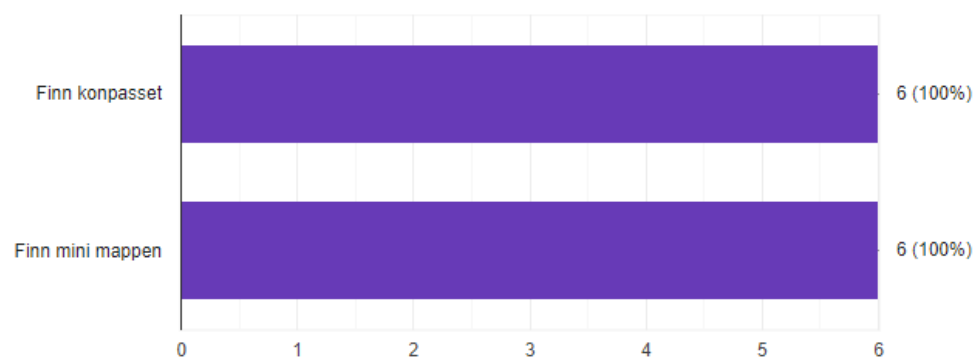
6 responses



UI test

Bruke Ulen

6 responses



Oppgave prosessen

Beskriv hvordan prosessen foregikk og hvordan deltaker overkom oppgavene over

6 responses

gikk veldig bra, og kom over oppgavene lett
kom lett over oppgavene med få problemer
Kom lett over oppgavene.
Veldig lett for deltakeren, ingen problemer.
Veldig lett, ingen problemer.
Kom gjennom oppgavene lett, ingen problem

Brukerens tanker om funksjonene

Hva er brukerens tanker om disse funksjonene?

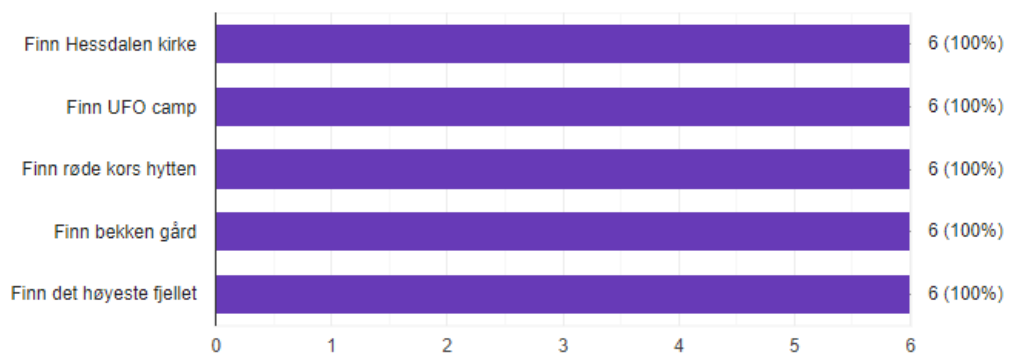
6 responses

fungerer bra. gir kontrollene fungerer utmerket, både det at man kan gå sakte og fort.
syntes at det var bra men mangler røde kors hytten som drop down meny.
Veldig ryddig, men mangler flere ting på kartet som er listet opp, og på drop down menyen.
Det fungerte bra. Kunne vært en mulighet å ha inverted controls som option.
Fungerer fint.
Landskapet virket fint å oversiktlig, detaljnivå er bra. Litt hakkete, men sånt skjer. Oversiktlig med navneskiltene.

Finne viktige steder

Finne steder på kartet Test

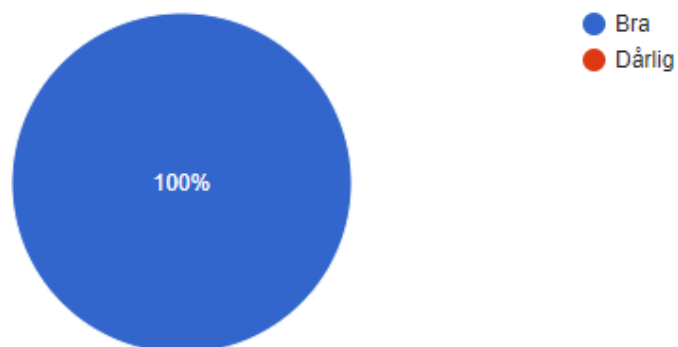
6 responses



Brukerens mening om kartet er bra eller dårlig

Syntest brukeren at kartet er bra eller dårlig?

6 responses



Brukerens helhetsinntrykk

Hvordan er helheten av 3D kartet for brukeren?

6 responses

detaljrikt, litt svake farger.
syntes det er bra, man finner og beveger seg 360 grade ri et område er fantastisk. at du kan lete deg fram til ting du har lyst til å besøke. kartet er tydelig.
Begynner å bli veldig fint, mye detaljer på kartet, og veldig lett å bevege seg med hastigheten på kameraet. Lett å finne alt som er på kartet.
Bygningene kunne vært større.
Helheten er bra.
Bra detaljer, veldig fint satt opp. Forventer å finne stedet når man trykker fram til det. Figurene er ikke helt integrerte i landskapet(kirke området). Fargene kan være grunn til dette.

Brukerens klager

Har brukeren noen spesifikke klager?

6 responses

ingen spesielle klager på kartet, bare litt hakkete når man bruker det.
ingen klager
To like hus er litt rart, men det går.
Det gikk veldig tregt på bærbar PC (laggy), mulighet for optimalisering.
Kunne hatt skriften ved stedene gjort større (skalert) når det er langt unna slik at man fortsatt kan se skriften. Gjøre veiene lettere å se. Detaljene i teksturene er bedre enn sist, men kunne fortsatt sett bedre ut.
Ingen direkte klager, bare noen pirketing.

Brukerens forbedringsforslag

Forbedringer som brukeren syntes trengs?

6 responses

ingen forbedringer som jeg kan tenke på.
legge til stedene som står at de skal være der, men ikke er det. vet ikke helt hva somkan forbedre seg.
Alt fylt ut av bygninger, og alt på drop down menyen fylt ut.
Ingenting utover det som allerede er nevnt.
Hvis du dobbeltklikker med touchpad kan man teleportere seg til området man klikker på.
Mer informasjon om område og bygninger. Pop up med informasjon. Kunne lagt inn klokkeslett og årstid for kartet.

Brukertest 2 konklusjon

Brukertest 2 forsikret oss om at kontrollene vi har brukt gjennom prosjektet fungerer veldig bra. Med nye funksjoner som Q og E for heving og senking av kameraet, hjalp dette brukerne til å håndtere kameraet på en mer effektiv måte om man ble forvirret. Vi fikk også se hva som må videre arbeides i etterkant av prosjektet i forbedrings delen av testen.

Med denne brukertesten så fant vi også ut at nettside begrensning vil ha mye å si i ferdige produktet som skal bli brukt av prosjekt Hessdalen.

10. Bibliotekliste

i Hentet fra Wikipedia i tidsrommet mellom Februar og Mai 2019

ii Nasjonalbiblioteket: Hessdalen kyrkje 1949-1989. Hessdalen menighetsråd

iii Nasjonalbiblioteket: Våre kirker. Norsk kirkeleksikon ISBN 82-75-27022-7