

RAPPORT

Bacheloroppgaven 2016

Gruppe 14

Rammeverk for WebGL

av: Eirik Thommessen

Sammendrag

I denne rapporten har det blitt sett på rammeverk og tjenester for implementering av 3D grafikk med WebGL på nettsider. Det har blitt lagt vekt på tjenestene og rammeverkernes selvstendighet, utviklermiljø og dokumentasjon.

Rammeverket som i undersøkelsen har pekt seg ut som vår anbefaling er Three.js. Three.js er et selvstendig javascript rammeverk for implementering av 3d og 2d grafikk med WebGL. Anbefalingen er basert på at Three.js ikke er proprietært, har et stort og levende utvikler og support miljø. Rammeverket er veldokumentert med eget api og eksempel bibliotek, samt flere eksterne ressurser både på web og i bokform.

Et sentralt argument er muligheten for å utvikle egne tilpassede verktøy for viderebruk, basert på rammeverket. Analyse av levedyktigheten til rammeverktøyet mål opp mot konkurrentene, gjør Three.js til den beste kandidaten.

Innhold

Sammendrag.....	1
Innhold.....	2
Bakgrunn.....	3
WebGL (Web Graphics Library).....	4
WebGL Rammeverk.....	4
Kriterier.....	5
Rammeverket eller tjenesten må ikke være proprietær.....	5
Gjenbruk av kode.....	5
Støtte for Collada (.dae).....	5
Babylon.js.....	6
Blend4Web.....	7
Clara.io.....	8
Goo create.....	9
Three.js.....	10
X3Dom.....	11
Konklusjon.....	12
Ordforklaring.....	13

Bakgrunn

Med utgangspunkt i at gruppen igjennom bachelorprosjektet skal utvikle et nettsted med 3D grafikk som hovedfokus, har vi i denne rapporten sett på de mest populære rammeverkenne og tjenestene innenfor WebGL. Oppgaven skal etter ønske fra arbeidsgiver løses igjennom dagens standarder og best practice innen HTML 5 og javascript. Det har derfor vært naturlig å ta utgangspunkt i rammeverk og tjenester som er basert på dette.

Oppdragsgiver (IFE) ønsker samtidig å kunne bruke løsningen fra nettsiden videre i sin drift, for på en enkel måte kunne dele og vise 3D modeller via interne og eksterne nettsider.

Vi skal i dette prosjektet ikke utvikle vårt eget rammeverk direkte basert på WebGL, men bygge videre på eksisterende rammeverk. gruppen vil derfor være avhengig av å bygge på et rammeverk som gruppen mener vil være bærekraftig og ikke kunne ilegges restriksjoner igjennom betalingstjenester eller tredjeparts software for å kunne utnyttes maksimalt.

Konsekvens ved valg av feil løsning kan føre til at prosjektets resultat ikke vil kunne være bærekraftig til viderebruk. Grunnen til dette kan være at utviklingsmiljøet rundt rammeverket forsvinner som igjen vil føre til at og essensielle oppdateringer for nye nettlesere og teknologi ikke blir gjennomført.

WebGL (Web Graphics Library)

WebGL¹ er et none-profit JavaScript basert api for visning av 3D og 2D grafikk i nettlesere, grafikken gjengis uten bruk av ekstra tilleggs verktøy også kjent som “plug-ins”. WebGL bruker nettleseren og datamaskinens GPU (Graphics Processing Unit) for å gjengi (“rendre”) ut 3d og 2d grafikk.

WebGL startet som et eksperiment av Vladimir Vukićević ved Mozilla. Han baserte WebGL på OpenGL, og bruker HTML5 DOM elementer og javascript til å gjengi 3d i browseren. WebGL ble første gang demonstrert i 2006, og har siden den gang økt i popularitet. Fra kun å være støttet av Mozilla og Opera, er WebGL i dag støttet av de fleste nettlesere og mobile enheter. Mye av grunnen til dette er at selskapet Khronos group i 2011 satte ned en arbeidsgruppe kalt “WebGL Working Group”. Deltakere i denne arbeidsgruppen finner vi blant annet Apple, Google, Mozilla og Opera.

WebGL er nå fullstendig eller delvis støttet av 84.01% av nettlesere²

Fullt integrert									
Delvis støttet									
Støtter ikke									
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
8								4.3	
9			45					4.4	
10	12	42	46			8.4		4.4.4	
11	13	43	47	9	34	9.2	8	46	47
	14	44	48	9.1	35	9.3			
		45	49		36				
		46	50						

WebGL Rammeverk

WebGL sitt api er komplisert og langtekkelig å programmere direkte i. Det er derfor blitt utviklet forskjellige rammeverk som gjør det lettere for utviklere å ta i bruk WebGL i sine applikasjoner og nettsider. Vi skal derfor videre i denne rapporten se på forskjellige ramme verks for WebGL

¹ <https://en.wikipedia.org/wiki/WebGL>

² Data hentet fra caniuse.com

Kriterier

Rammeverket eller tjenesten må ikke være proprietær

Det vil si at rammeverket må være åpent og gratis for brukere, og utviklere. Det skal ikke ligge restriksjoner eller krav om lisenser for å bruke rammeverket eller tjenesten. Tjensten eller rammeverket skal heller ikke være avhengig av programvare som er lisensbelagt eller inneholder restriksjoner.

Gjenbruk av kode

I utviklingen av prosjektet ønsker vi å bruke en DRY (don't repeat yourself) metode. Det vil si at vi ønsker å kunne utvikle prosjektet ved bruk av moduler som vi kan bruke igjen og kalle på i forskjellige deler. Dette støtter igjen ønsket fra oppdragsgiver om en sluttløsning for viderebruk av IFE.

Støtte for Collada (.dae)

Collada er et ISO-format (International Organization for Standardization). De aller fleste 3D verktøy støtter eksportering til dette formatet. Som ISO format for 3D-modeller, ser vi dette som minste felles multiplum for hva rammeverket skal støtte. Det er med på å sikre en bærekraftig løsning på lengre sikt.

Babylon.js

Babylon er et open-source prosjekt og har som fokus å tilrettelegge for programmering av 3D spill i nettleseren basert på HTML og javascript. Babylon.js er basert på Javascript som tar inn modeller som JSON strenger eller instanser av primitiver fra babylon.js eget bibliotek/rammeverk. JSON strengen eller primitivene gengis videre igjennom et div- eller canvas-element.

For bruk av babylon.js har brukeren ikke behov for ekstra programvare utover en tekst-editor og en browser som støtter WebGL. Babylon.js bibliotek filer lenkes inn i html-dokumentet og er klart til bruk.

Babylon kommer med en rekke tredjeparts javascript som er med på å forenkle utviklingsprosessen i rammeverket. Babylon har i dag støtte for å laste inn 3D modeller i enkelte filformater. Disse er .stl, .obj og glTF, men har per dags dato ingen "out-of-the-box" støtte for Collada.

Babylon.js er fullstendig rammeverk med et godt strukturert dokumentasjon, som gjør det enkelt for utviklere å lage egne moduler basert på rammeverket.



*Aktiviteten rundt babylon.js fra juni 2013 - januar 2016.³

December 20, 2015 – January 20, 2016

Period: 1 month

Overview			
43 Active Pull Requests		0 Active Issues	
42 Merged Pull Requests	1 Proposed Pull Request	0 Closed Issues	0 New Issues

Excluding merges, **10 authors** have pushed **60 commits** to master and **60 commits** to all branches. On master, **130 files** have changed and there have been **9,921 additions** and **4,805 deletions**.



*Aktivitet siste måned⁴

³ <https://github.com/BabylonJS/Babylon.js/graphs/contributors>

⁴ <https://github.com/BabylonJS/Babylon.js/pulse/monthly>

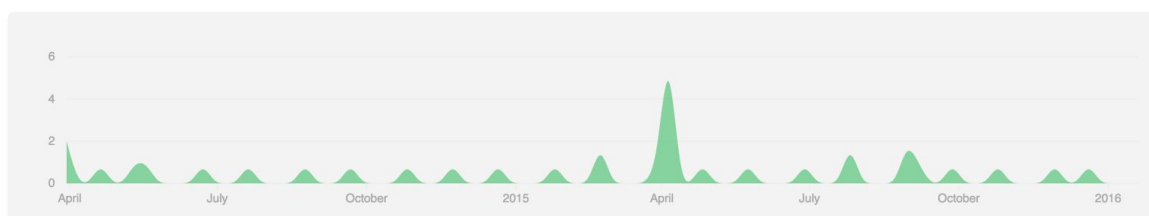
Blend4Web

Blend4Web er et open-source rammeverk for utvikling av 3D grafikk. Blend4Web er ikke et rammeverk som står på egne ben, men er et verktøy for å eksportere innhold fra 3D-modelleringsverktøyet Blender, derav navnet Blend4Web som er et akronym for Blender for web.

Blend4Web er en plugin som installeres i Blender. Ved hjelp av Blender og Blender4Web kan brukere uten erfaring fra programmering eksportere 3D scener til browseren og selvstendige HTML dokumenter.

Output formatet til Blend4Web er JSON. Dette gjør at Modeller laget i Blender eksportert med Blend4Web plugin kan brukes i andre rammeverk som three.js og babylon.js.





Bruk av Blend4Web avhenger at brukeren også bruker Blender som utviklingsverktøy, eller converterer modeller igjennom blender som en del av arbeidsflyten.



*Aktivitet fra mars 2014 - januar 2016⁵

December 20, 2015 – January 20, 2016

Period: 1 month ▾

Overview			
0 Active Pull Requests		0 Active Issues	
 0	 0	 0	 0
Merged Pull Requests	Proposed Pull Requests	Closed Issues	New Issues

Excluding merges, **1 author** has pushed **1 commit** to master and **1 commit** to all branches. On master, **1,203 files** have changed and there have been **89,014 additions** and **66,082 deletions**.



*Aktivitet siste måned⁶

⁵ <https://github.com/TriumphLLC/Blend4Web/graphs/contributors>

⁶ <https://github.com/TriumphLLC/Blend4Web/pulse/monthly>

Clara.io

Clara.io er et fullstendig nettbasert utviklingsverktøy for 3d-modelering og animasjon. Igjenom Clara.io kan brukeren importere og modellere 3D modeller, lage teksturer og materialer.

Clara tilbyr å dele direkte fra deres webtjeneste eller eksportere til 3d formater. Clara.io støtter eksport direkte til blant annet Babylon.js, Blender, Three.js og collada.

Clara.io er ikke open-source, eller fullstendig gratis. Clara.io er basert på abonnementer. Hvor brukere kan få et gratis abonnement som inneholder begrenset lagringsplass og renderingstid. Clara.io opererer med et anbefalt abonnement på 25 dollar i måneden.

Bruk av Clara.io forutsetter en fullstendig arbeidsflyt i eller igjennom Clara.io. Clara.io tilbyr import av colladafilmer av en rekke forskjellige filformater.

Dersom brukeren velger å benytte seg av et Clara.io abonnement for senere nedgradere abonnementet vil brukeren kunne oppleve av innhold over de lagring grensene vil kunne bli slettet.



Free	\$10	\$25	Contact
Always	PER MONTH	PER MONTH	Sales
	\$100	\$250	
PER YEAR	PER YEAR	PER YEAR	
Basic	Standard	Professional	Enterprise
	Recommended	Recommended	& Education
5GB cloud storage Unlimited public scenes 10 private scenes 1 hour personal rendering Unlimited rendering jobs** WebGL embeds VR-day embeds No credit card necessary	20GB cloud storage Unlimited public scenes Unlimited private scenes 1 hour personal rendering Unlimited rendering jobs** Custom WebGL embeds VR-day embeds Risk-free 30 day trial	100GB cloud storage Unlimited public scenes Unlimited private scenes 1 hour personal rendering Unlimited rendering jobs** Custom WebGL embeds Custom VR-day embeds Priority rendering Risk-free 30 day trial	Unlimited cloud storage Unlimited public scenes Unlimited private scenes Unlimited rendering Priority rendering Custom WebGL embeds Custom VR-day embeds Priority rendering Free Trial Periods
Get Started	Get Started	Get Started	Contact Sales

Goo create

Goo er et svensk merke støttet av blant annet Ravio, selskapet bak angry birds. Målet til Goo er å skape unike opplevelser på nettet. Goo create er et nettbasert verktøy for utvikling av 3D igjennom WebGL.

16. november 2015 ble Goo Engine gjort til et open source prosjekt. Goo Engine er rammeverket som Goo Create er basert på.

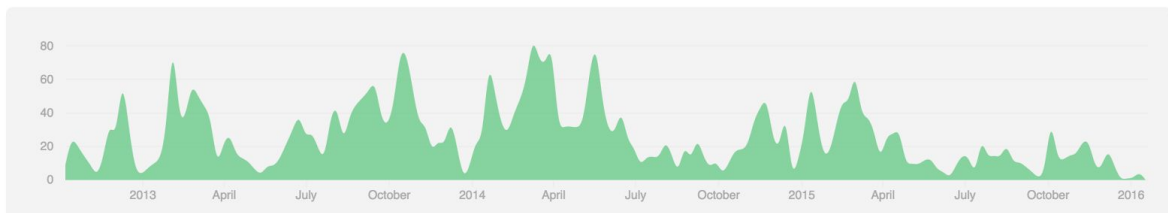
Goo Create gir deg tilganger til deres servere og deres teknologi basert på forskjellige abononenter. Goo Create aksepterer blant annet collada-filer og har mulighet for scripting.

Det er også mulighet for å bruke Goo Engine selvstendig, men det finnes lite dokumentasjo om goo engine utover Goo sine sider. Siden Goo først ble gjort til et open-source prosjekt i slutten av 2015, har ikke rammeverket rukket å bygge opp en support base. Goo Engine ser ut til å ta opp kampen med Babylon.js om å bli ledene som selvstendige rammeverk for WebGL innenfor spill sjangren.

Oct 7, 2012 – Jan 20, 2016

Contributions to master, excluding merge commits

Contributions: **Commits** ▾



*Aktivitet fra oktober 2012 - januar 2016⁷

December 20, 2015 – January 20, 2016

Period: **1 month** ▾

Overview			
0 Active Pull Requests		0 Active Issues	
0 Merged Pull Requests	0 Proposed Pull Requests	0 Closed Issues	0 New Issues

Excluding merges, **3 authors** have pushed **7 commits** to master and **58 commits** to all branches. On master, **20 files** have changed and there have been **559 additions** and **227 deletions**.



*Aktivitet siste måned⁸

⁷<https://github.com/GooTechnologies/goojs/graphs/contributors>

⁸<https://github.com/GooTechnologies/goojs/pulse/monthly>

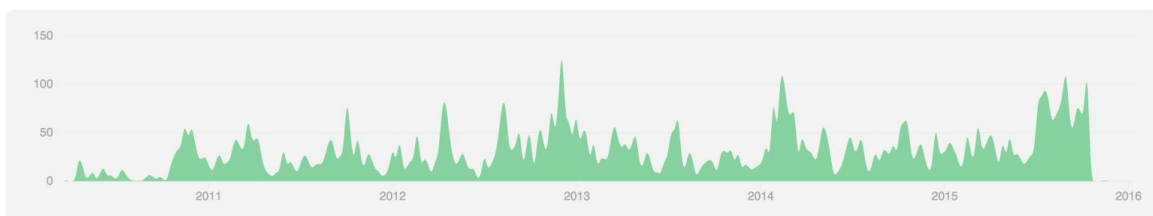
Three.js

Three.js er et open-source rammeverk som ble lagt ut på Github i April 2010, og er basert på JavaScript og HTML5. Three.js støtter en rekke forskjellige 3d formater, deriblant Collada igjennom en egen collada-loader som parser colladas xml-format over til JSON. Three.js har også støtte for keyframe animasjoner fra collada.

Three.js er på lik linje som Babylon.js selvstendig og ikke avhengig av tredjepart software. Dette gjør det mulig å bygge egne moduler på eksisterende rammeverket. Three.js har som open-source tilgang til flere utvidelser igjennom tredjeparts biblioteker som blant annet physijs, som tilbyr fysikkmotor.

For uten stort og veldokumenter brukerveiledning, har Three.js et stor bibliotek med eksempler på hvordan Three.js kan brukes.

Three.js har et aktivt miljø med utviklere og har på grunn av sin popularitet opparbeidet seg en god dokumentasjon også i bokform og ved nettundervisning med veiledninger.



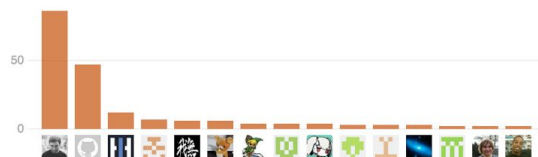
*Aktivitet fra mars 2010 - januar 2016⁹

December 20, 2015 – January 20, 2016

Period: 1 month ▾

Overview			
68 Active Pull Requests		111 Active Issues	
57 Merged Pull Requests	11 Proposed Pull Requests	85 Closed Issues	26 New Issues

Excluding merges, **22 authors** have pushed **0 commits** to master and **199 commits** to all branches. On master, **0 files** have changed and there have been **0 additions** and **0 deletions**.



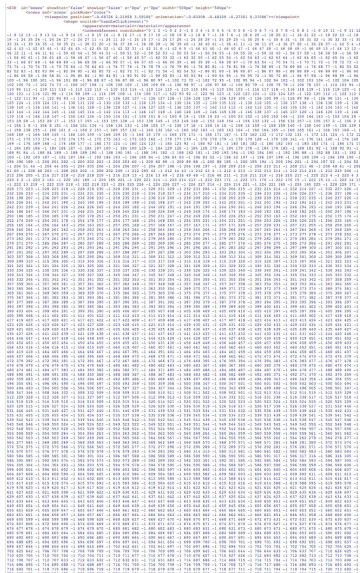
*Aktivitet siste måned¹⁰

⁹ <https://github.com/mrdoob/three.js/graphs/contributors>

¹⁰ <https://github.com/mrdoob/three.js/pulse>

X3Dom

X3Dom er et selvstendig open-source rammeverk. I motsetning til andre selvstendige rammeverk hvor 3D implementeringen ligger i Javascript, ligger mye av implementeringen av X3Dom rammeverket i DOM-elementer. Det vil si at der rene javascript baserte rammeverk som Three.js og Babylon.js har basert 3D grafikken i javascriptfilen, flytter X3Dom 3D-elementene i html dokumentet som tagger med attributter og verdier.



11

Fordelen med dette er at utviklere ikke trenger å lære seg javascript for å komme i gang. Ulempen er store html filer og mye repetisjon, som fører til “code-bloating” og strider med DRY-prinsippene¹². Code blokkene i X3Dom er også avhengig av åpning og slutt tagger. Dette gjør at det blir mer data som må overføres, og laste tiden til siden vil øke.

```
<shape>
  <appearance>
    <material diffuseColor='1 0 0'></material>
  </appearance>
  <box></box>
</shape>
```

Fordelen med bruk av X3Dom er at det er veldig godt dokumentert igjennom nettsiden www.x3dom.org , samtidig er x3Dom perfekt til mindre prosjekter, fordi du enkelt kan få tak i elementer, stilsette dem via CSS.



X3Dom er lærebok eksempelet på hvordan en nettside skal fungere i teorien, hvor innhold ligger i HTML-dokumentet, stilen ligger i CSS-dokumentet og funksjonalitet i Javascript-dokumentet.

¹¹ [Eksempel på “code-bloating” i x3dom](#)

¹² DRY - **D**on't **R**epeat **Y**ourself

Konklusjon

Valg av en proprietær løsning hvor prosjektets resultat og videre bruk avhenger av betalingsløsninger og valgt selskaps evne til å eksistere, samt ha ressurser til å holde seg oppdatert, vil være en for stor risiko for gruppen.

Samtidig ønsker ikke gruppen å gjøre endelig løsning avhengig av et bestemt software. Selv om programvaren er open-source er det ingen garanti for vedlikehold eller fornyelse. Selv om Blender og blender4Web ser ut som en perfekt "out of the box" løsning. Både når det gjelder modellering og animasjon i prosjektet, Vil det gjøre alle fremtidige endringer avhengig av Blender.

Av rammeverkene og løsningene vi har sett på er det to løsninger som tilfredsstiller de fleste kravene gruppen har satt til valg av rammeverk. x3Dom og Three.js. bakgrunnen for dette er basert på implementering av ISO formatet collada, at begge er selvstendige open-source rammeverk, dokumentasjon, og utviklermiljø rundt rammeverkene.

Av de to er det Three.js som blir anbefalingen til gruppen. Three.js er det mest selvstendige og som over tid har vist seg å være det mest robuste none-proprietære rammeverket.

Bakgrunnen for at x3Dom ikke blir anbefalt er i hovedsak av to grunner. Bloating i HTML koden, noe som fører til uoversiktlig HTML kode. Og at koden ikke er mulig å gjøre objektorientert. Dette gjør x3dom for statisk til å gjøre modulbasert og møte gruppens ønske om gjenbrukbarhet.

Three.js har fordelen av å ha en integrert colladaloader for modell, keyframe animasjon og kinematics. Dette kombinert med det mest aktive utviklermiljøet både nå og over tid, gjør at per dags dato er Three.js det rammeverket som passer best til de forutsetningene vi har for dette prosjektet.

Ulempen med Three.js er parsingen av collada over til JSON format. Denne prosessen har få feilhåndteringer og ved uforutsette elementer i collada filen vil formateringen skape unhandled exceptions. Dette vil føre til at siden krasjer. Bakgrunnen er at collada som format er ISO med håndteringen av collada er det ikke. Dette er noe gruppen må se på i sitt fremtidige arbeid.

En annen utfordring gruppen burde se på er hvorvidt parsing av collada formatet til JSON må foregå hver gang siden lastes, eller om dette er noe som foregår ved opplasting av filer til nett.

Ordforklaring

JSON - JavaScript Object Notation, er en enkel tekstbasert standar for datautveksling. JSON er opprinnelig avledet fra Javascript med standarden er imidlertid uavhengig av JavaScript eller andre programmeringsspråk.

XML - Extensible Markup Language, er et markeringsspråk for deling av strukturert data mellom informasjonssystemer. XML er organisert i en hierarkisk struktur, bygget på tagger.

DOM - Document Object Model. Brukes som betegnelse på taggene i et html eller XML dokument.

DRY - Don't Repeat Yourself. Et prinsipp om at kode ikke skal bli gjentatt eller dublisert. Alternativet er refakturering av koden i metoder.

Code bloating - En betegnelse på kode i et dokument som på grunn av mengde ikke blir lett oversiktlig eller lesbar for utvikleren.