



HoGent

Faculteit Bedrijf en Organisatie

React 360, een framework voor virtual reality applicaties te bouwen

Michiel Glibert

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Johan Van Schoor
Co-promotor:
Jasper Dansercoer

Instelling: —

Academiejaar: 2017-2018

Tweede examenperiode

Faculteit Bedrijf en Organisatie

React 360, een framework voor virtual reality applicaties te bouwen

Michiel Glibert

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Johan Van Schoor
Co-promotor:
Jasper Dansercoer

Instelling: —

Academiejaar: 2017-2018

Tweede examenperiode

Woord vooraf

Samenvatting

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus.

Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Inhoudsopgave

1	Inleiding	17
1.1	Probleemstelling	18
1.2	Onderzoeksvraag	18
1.3	Onderzoeksdoelstelling	19
1.4	Opzet van deze bachelorproef	19
2	Stand van zaken	21
2.1	De virtuele realiteiten	21
2.1.1	Virtual reality	21
2.1.2	3D en 4D	22
2.1.3	Augmented Reality	22
2.2	De werking van virtual reality	23
2.2.1	Stereoscopie	23

2.2.2	De virtuele ervaring	24
2.2.3	Interactie met de virtuele wereld	25
2.3	De gevolgen voor de gebruiker	25
2.3.1	User Experience	26
2.4	JavaScript, HTML en CSS	26
2.4.1	HTML	27
2.4.2	CSS	28
2.4.3	Belangrijkste onderdelen	29
2.5	De mogelijke VR frameworks naast React 360	32
2.5.1	A-Frame	32
2.5.2	Primrose	32
2.5.3	Agon.js	33
3	Methodologie	35
3.1	Het framework, React 360, bestuderen	35
3.2	Ervaring van virtual reality voor een gebruiker	35
3.3	De doeleinden en stand van zaken van React 360	36
4	React 360	37
4.1	React.js	38
4.1.1	JSX	38
4.1.2	Components en props	38
4.1.3	State en levenscyclus	38
4.1.4	Events	38
4.2	Het ruimtelijke systeem	38

4.3	Enkele belangrijke componenten	38
4.3.1	View	38
4.3.2	Environment	38
4.3.3	Video	38
4.3.4	VrButton	38
4.4	Input in VR	38
4.5	Developing in VR	38
4.6	Prestaties	38
5	Conclusie	39
A	Onderzoeksvoorstel	41
A.1	Introductie	41
A.2	State-of-the-art	41
A.3	Methodologie	42
A.4	Verwachte resultaten	42
A.5	Verwachte conclusies	42
	Bibliografie	43

Lijst van figuren

2.1	Augmented reality tegenover virtual reality.	23
2.2	Stereoscopie voorbeeld.	24
2.3	De hiërarchie van de benodigdheden in virtual reality.	27

Lijst van tabellen

Listings

2.1	HTML Element voorbeeld	27
2.2	HTML Element voorbeeld	27
2.3	Voorbeeld van een HTML bestand	28
2.4	Voorbeeld van een CSS bestand	28
2.5	Hond object aanmaken	29
2.6	Eigenschap toevoegen aan en object	29
2.7	Het nieuwe hond object	29
2.8	Een functie met var	30
2.9	Een functie met let	30
2.10	Een functie die 2 getallen optelt	31
2.11	Een functie de som functie oproept	31
2.12	Voorbeeld van primrose omgeving	32

1. Inleiding

Virtual Reality, of in het Nederlands virtuele werkelijkheid, is een technologie die het mogelijk maakt om een bepaalde omgeving zowel auditief als visueel te simuleren en het gevoel te geven aan de gebruiker dat hij/zij zich echt bevindt in die omgeving. Men gaat letterlijk op de menselijke zintuigen gaan inspelen door middel van elektronica om dus de gebruiker een gevoel van realiteit te geven dat eigenlijk niet echt is. Het voornaamste apparaat dat hiervoor gebruikt wordt is een virtual reality bril. Dit is een bril waarmee voor elk oog een beeld van de virtuele wereld wordt weergegeven en waarbij dan ook rekening wordt gehouden met de afstand van de ogen. Deze brillen hebben meestal ook extra sensoren zoals bijvoorbeeld een gyroscoop die ook alle bewegingen met het hoofd opvolgt en weergeeft in de virtuele wereld.

De toepassingen waar dat virtual reality kan gebruikt worden zijn zeer uitgebreid. De bekendste hiervoor is entertainment, namelijk gaming. Hiermee kan men dus gaan simuleren alsof de persoon zich in het spel bevindt. Daarnaast kan men dit ook gebruiken voor 3D cinema waarbij men dus videofragmenten kan doen laten afspelen rondom de gebruiker. Ook een belangrijke toepassing is in het medische gebied. Men kan VR gaan gebruiken als behandeling tegen bepaalde aandoeningen zoals bepaalde fobieën en PTSS (contributors, 2018). Daarnaast kan aan de hand van virtual reality bepaalde ingrepen gaan simuleren als training. Deze training kan dan een goede voorbereiding zijn op de echte ingreep. Het gebruik van VR als training komt ook voor in andere gebieden dan het medische, zoals in het leger, astronaut, vliegsimulators, ...

Virtual reality heeft dus zeker een ruim aanbod van toepassingsgebieden waarvoor het gebruikt kan worden. Maar toch is het ontwikkelen van virtual reality applicaties een grote uitdaging. Daar gaan wij dan ook dieper op in deze bachelor proef. Waar wij vooral de focus zullen leggen op het gebruik van React 360 als virtual reality framework om

applicaties mee te bouwen.

1.1 Probleemstelling

Virtual reality is een nog opkomende trend die nog in zijn kinderschoenen staat. Hierdoor is het nog moeilijk om toe te treden tot deze markt aangezien de prijzen van deze virtual reality brillen soms hoog kunnen oplopen. Dit is momenteel nog een struikelblok voor de consument om de aankoop van een VR headset uit te stellen, maar zeker niet de enigste. In het artikel van (Abarrera, 2017) wordt goed aangehaald wat precies de redenen zijn waarom VR headsets nog niet volledig aangeslaan zijn bij het publiek. Een andere zeer belangrijk probleem met VR momenteel is het tekort aan ontwikkelaars voor deze technologie. Een virtual reality applicatie is zeker niet hetzelfde als een gewone webapplicatie. Er moet ten eerste al letterlijk in 360°gedacht worden aangezien een persoon in VR perfect rond zich moet kunnen kijken. Daarnaast mogen we niet vergeten dat bepaalde beelden die ongepast zijn voor VR een negatieve weerslag kunnen geven aan de gebruiker en zelfs kan lijden tot fysieke pijn. Ten slotte is gaat het ontwikkelen van een virtual reality applicatie niet zomaar. Er is al snel nood aan een groot aantal frameworks zoals een manier om een 3D omgeving te ondersteunen. Dit kan voor een ontwikkelaar geen gemakkelijke taak zijn om zomaar aan te leren en dit zorgt er dan weer voor dat ontwikkelaars moeilijker de stap naar het ontwikkelen van een virtual reality applicatie gaan nemen.

React 360 zou hier een oplossing voor moeten bieden. Dit framework biedt niet alleen een groot aantal componenten aan waarmee er al snel een solide VR app kan worden ontwikkelt, maar daarnaast wordt er bij React 360 ook gebruik gemaakt van React JS en het daarbij horende javascript. Dit zorgt er dus voor dat de kloof tussen het ontwikkelen een webapplicatie en een virtual reality applicatie pakken kleiner wordt. Ook is het zo dat men voor deze apps uit te voeren niet verplicht een virtual reality headset hoeft te gebruiken aangezien deze applicaties gewoon in de browser kunnen worden uitgevoerd. Daarnaast zijn de mobiele virtual headsets, die voor React 360 kunnen gebruikt worden, een pak goedkoper.

1.2 Onderzoeksvraag

Met de uitleg die ik zonet aangaf kunnen we een aantal onderzoeksvragen definiëren waar ik in mijn bachelorproef een antwoord tracht naar te vinden:

- Hoe ervaart een gebruiker een VR applicatie ontwikkeld met React 360 ten opzichte van een gewone webapplicatie?
 - Hoe voelt de user experience aan
 - Wat zijn de fysieke opmerkingen?
- Voor welke doeleinden kan React 360 gebruikt worden
- Is React 360 al een goed framework voor volwaardige VR applicaties
- Wat is de performantie van React 360 in de browser?

- Is er nood aan dure hardware voor React 360?

1.3 Onderzoeksdoelstelling

In dit onderzoek zoeken we een antwoord op al onze onderzoeksvragen en vooral of React 360 een goed framework is voor virtual reality applicaties voor bepaalde doeleinden. Het antwoord hoeft dus niet noodzakelijk positief te zijn. We willen vooral het framework gaan ontleden en alle aspecten dat het aanbiedt gaan bekijken terwijl we ook rekening houden met alle concepten van de wereld van virtual reality. Zowel React 360 als virtual reality zijn nog relatief jong ookal bestaat VR op zich wel al lang. Het is pas sinds de laatste jaren dat computers hier klaar voor waren en hiermee moeten we dus ook zeker rekening houden.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie. Onder andere de technologie virtual reality zelf en de taal waarmee React 360 gebouwd is, javascript.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4 wordt React 360 bestudeert en bekijken we enkele belangrijke onderwerpen van het framework.

In Hoofdstuk 5, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

Voordat React 360 kan worden bekeken is het belangrijk een goed beeld te scheppen van virtual reality zelf. We gaan dieper in op wanneer iets virtual reality is, hoe het werkt en wat de gevolgen zijn voor de gebruiker van deze applicaties. We zullen tot slot, nog een overzicht geven van mogelijke alternatieven voor React 360 met ook een korte uitleg van deze frameworks.

2.1 De virtuele realiteiten

We weten al wat virtual reality is, maar wanneer is iets nu virtual reality. Er zijn namelijk nog een aantal andere vormen van valse realiteiten die al snel verward kunnen worden met virtual reality, maar wat zijn deze nu precies en wanneer kunnen we iets als VR aanzien.

2.1.1 Virtual reality

Om gemakkelijk te kunnen aantonen wanneer nu iets virtual reality is maken we gebruik van de 4 sleutel elementen van virtual reality die (Sherman, 2000) in zijn boek aanhaalt.

Een virtuele wereld

Immersion/Onderdompeling

Feedback van de zintuigen

Interactiviteit

2.1.2 3D en 4D

3D wil letterlijk zeggen driedimensionaal. Hiermee bedoelt men dat iets 3 meetkundige dimensies heeft namelijk diepte, breedte en hoogte. Deze technologie kan men dan gaan toepassen op beeldschermen a.d.h.v. stereoscopie, waar we in hoofdstuk 2.2 wat meer uitleg over geven. Op deze manier krijgt de gebruiker dus een illusie van diepte. Meestal worden hier dan speciale brillen voor gebruikt ook wel 3D-brillen genoemd. De klassiekere brillen hebben dan een kleurfilter, meestal rood en blauw. Rood laat enkel rood door en blauw enkel blauw. Hiermee kan men dan vanaf 2 perspectieven een blauw afbeelding en een rode afbeelding gaan weergeven waardoor de diepte van een object dus zichtbaar is.

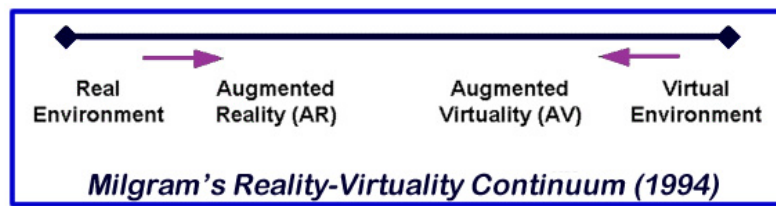
3D wordt al veel toegepast in de filmwereld, vooral in de cinema dan. Hierdoor ontstond dan ook de uitgebreidere 4D maar is eerder een marketingterm dan een technologie en wordt bijna uitsluitend in de filmwereld gebruikt als entertainment. Bij 4D gaat men de beelden nog altijd driedimensionaal gaan weergeven maar zorgt men ook voor extra fysieke effecten die synchroon lopen met de film. Als bijvoorbeeld een film zich in het water afspeelt kan men af en toe kleine spatjes water op de kijker schieten.

Het grote verschil hier is de immersion die amper aanwezig is. Je hebt wel een extra dimensie maar je kan niet rond je kijken, dat kan je met een virtual reality headset natuurlijk wel. Daarnaast is interactie met de wereld in de meeste gevallen niet aanwezig, spelconsoles zoals de Nintendo 3DS vormen dan weer een uitzondering. 3D of 4D is dus zeker geen volwaardige virtual reality, maar is wel al een stap in de goede richting.

2.1.3 Augmented Reality

Augmented Reality wordt heel veel verward met virtual reality maar zijn toch 2 andere technologieën. Je kan augmented reality gaan vertalen naar toegevoegde realiteit. In tegenstelling tot virtual reality gaat men bij augmented reality letterlijk iets gaan toevoegen aan wat al bestaat. Bij virtual reality gaat men dan iets volledig virtueel gaan scheppen. Aangezien augmented reality iets toevoegt, is er ook niet verplicht nood aan een bril. Vandaag de dag wordt augmented reality regelmatig gebruikt bij de smartphone. Denk maar aan een applicatie waarbij je een object kan afbeelden op een plaats. Bijvoorbeeld een bepaald meubilair die virtueel gegeneerd en getoond wordt in een kamer dat echt bestaat.

Er is niet echt een definitie van wat augmented reality precies inhoudt. Je kan eigenlijk al een scorebord bij live uitzending van een sport op de televisie zien als een vorm van augmented reality. Maar volgens (Azuma, 1997) zou iets aan volgende 3 karakteristieken



Figuur 2.1: Augmented reality tegenover virtual reality.

moeten voldoen om augmented reality te kunnen zijn:

- Combineert de echte wereld met de virtuele wereld
- Men kan interacties doen met het virtuele
- De virtuele wereld bezit de 3 dimensies (diepte, breedte en hoogte)

Als we deze 3 karakteristieken toepassen op het voorbeeld van het scorebord, zien we dus dat een scorebord bij een live uitzending van sport geen augmented reality is. De 1ste voorwaarde wordt wel voldaan, maar de 2de en de 3de niet. Men kan geen interacties gaan uitvoeren op dit scorebord, de programmamaker zou dit wel kunnen dus voor hem is dan enkel de 3de regel niet voldaan. Zou het beeld voor de programmamaker dan ook nog in 3D (een 3D scorebord dus, niet noodzakelijk een 3D beeldscherm) worden weergegeven, dan kunnen we stellen dat dit een vorm van augmented reality is.

2.2 De werking van virtual reality

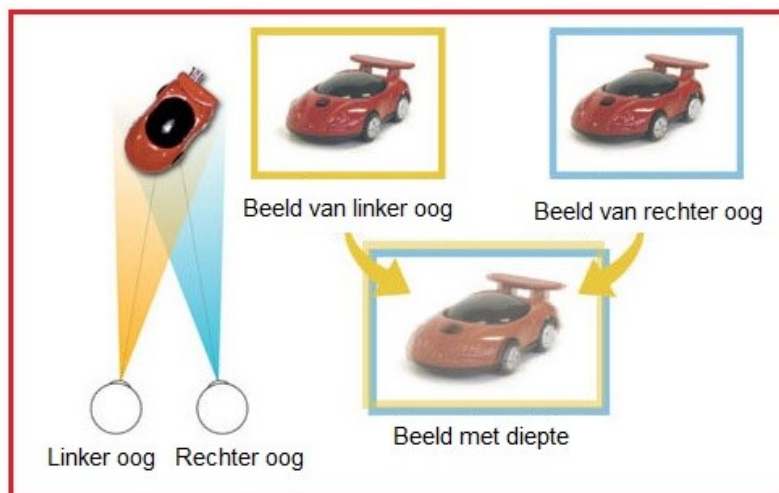
Er werd al duidelijk gemaakt wat virtual reality precies is, maar het is ook belangrijk te weten hoe de technologie werkt. Er zal hier niet te technisch worden op ingegaan aangezien dit niet enorm relevant is voor deze bachelorproef.

2.2.1 Stereoscopie

Bij virtuele werkelijkheid wordt er gebruik gemaakt van een illusie. Dit doet men ten eerste a.d.h.v. stereoscopie. Hierbij gaat men diepte meegeven aan een afbeelding. Dit doet men door 2 afbeeldingen vanaf een verschillende afstand (meestal de afstand tussen de ogen) te maken. Hierna gaat men dit combineren tot één stereoafbeelding. Deze technologie wordt ook gebruikt bij 3D (Rouse, 2011).

Door stereoscopie kan men dus diepte gaan simuleren. Hiermee is al een een grote voorwaarde voldaan om iets realistisch te laten lijken, namelijk 'Immersion' (key elements VR), onderdompeling. Zo kan men beter inschatten hoe ver en hoe groot een object is in de virtuele wereld. Men kan deze 2 afbeeldingen dan gaan tonen door zo een VR headset. Dit kan op meerdere manieren.

- Ofwel maakt men gebruik van een smartphone waarbij men dan een headset heeft waarbij gebruik wordt gemaakt van lenzen. Op de smartphone worden er 2 beelden



Figuur 2.2: Stereoscopie voorbeeld.

geprojecteerd en zorgen de lenzen ervoor dat de omgeving ruimer lijkt dan het werkelijk is. Hiermee kan men dan op een goedkope manier virtuele realiteit gaan tonen. Maar dit is meestal ten koste van de kwaliteit doordat de resolutie op smartphones meestal te laag is voor virtuele werkelijkheid scherp te kunnen weergeven.

- Daarnaast heeft men de meer geavanceerde virtuele headsets. Hierbij is er per oog een scherm met een hoge resolutie. Deze tonen dan elk hun beeld en simuleren dan de virtuele omgeving. Deze headsets worden dan ook het meest gebruikt bij gaming, maar zijn automatisch ook een pak duurder en vereisen een krachtige computer.

2.2.2 De virtuele ervaring

In hoofdstuk 2.1.2 hebben we al vermeld dat het niet enkel door de extra dimensie is dat iets al virtual reality is. Er zijn nog meerdere aspecten waarmee men rekening moet houden bij een virtuele headset. Zo zijn de beelden per seconde, ofwel frames per second (FPS), van de beeldschermen in die headsets zeer belangrijk. De 2 bekendste virtuele headsets momenteel op de markt, die dan vooral naar gaming gericht zijn, zijn de HTC vive en de Oculus Rift. Beide headsets maken gebruik van een 90Hz scherm. Dat wil zeggen dat er maximaal 90 beelden per seconde kunnen worden weergegeven. Hoe sneller, hoe realistischer. Een andere bekende headset, de Playstation VR, draait maar op 60Hz, 60 beelden per seconde dus. Bij smartphones kan dit nog lager liggen, tot 30Hz zelfs. Het is dus duidelijk dat de HTC vive en Oculus Rift een realistische ervaring zullen geven dan de Playstation VR of de smartphone.

Wat ook een grote impact kan hebben op de ervaring is latency. Latency is de hoeveelheid tijd er tussen zit als je een bepaalde input geeft en die dan wordt weergegeven in de virtuele wereld. Een goed voorbeeld hiervan is het moment dat je een stap vooruit zet, zal er een bepaalde hoeveelheid tijd zijn tot je vooruit beweegt in de virtuele wereld. Bij een te hoge latency zal de virtuele ervaring als heel slecht ervaren worden en zelfs vanaf er al een

latency is van 20ms zal het menselijk brein duidelijk onderscheidt kunnen maken tussen iets dat echt is en iets dat vals is. Dit zal niet alleen meiden tot een mindere ervaring voor de gebruiker maar kan ook tot fysieke pijn leiden zoals motion sickness, iets dat ook voorkomt bij onze moderne manieren van transport bij bijvoorbeeld autorijden (wagenziekte). Het is dus zeer belangrijk dat deze latency zo laag mogelijk is, niet alleen voor de ervaring maar ook om ongemakkelijkheid bij de gebruiker te vermijden.

Ten laatste is de FOV, field of view ofwel het gezichtsveld in de virtuele wereld ook zeer belangrijk. Een mens ziet ongeveer 180° rond zich, maar dit kan oplopen tot 270° als er met de ogen bewogen wordt. De meeste headsets hebben maar een gezichtsveld tussen de 90° en 110°, wat dus eigenlijk niet genoeg is. Dit heeft dan ook een impact op de virtuele ervaring en kan ook hier opnieuw leiden tot motion sickness.

Als er dus niet wordt voldaan aan een hoge FPS, voldoende FOV en een lage latency zal dit dus een impact hebben op de virtuele ervaring. Hierdoor ontstaan dan fysieke ongemakken omdat het menselijk brein weet dat er iets niet klopt. Het is dus zeer belangrijk aan ontwikkelaars van virtuele applicaties om hiermee rekening te houden. Als een applicatie je ziek maakt, dan ga je dat automatisch ook minder of zelfs niet meer gebruiken.

2.2.3 Interactie met de virtuele wereld

Vandaag zijn we in staat om een virtuele wereld zeer realistisch weer te geven, maar toch blijft interactie met de virtuele wereld een moeilijk aspect van virtual reality en het is nochtans één van de belangrijkste. Wat men al sowieso goed doet, is rond je kunnen kijken in de virtuele wereld, dit werd al in de introductie vermeldt en gebeurt aan de hand van een gyroscoop. Maar hoe kan men dan bewegingen van de echte wereld gaan omzetten in die virtuele wereld?

Dit kan ten eerste gebeuren aan de hand van afstandsbediening. Men kan gebruik maken van een simpele afstandbediening met joystick. Waarbij men dus gewoon in de virtuele wereld gaat bewegen als hoe men dit zou doen in een videospel, maar dit is natuurlijk niet de perfecte ervaring aangezien er weinig rekening wordt gehouden met hoe de handen bewegen. Een betere oplossing hiervoor waren dan de controllers die ook bewegingen registreren (denk maar aan de Nintendo Wii). Dit is al een grote stap vooruit. Verder gaan we hier niet dieper op ingaan.

... afwerken

2.3 De gevolgen voor de gebruiker

Een virtual reality app is erg verschillend van een gewone webapplicatie. Hierbij is het doel om de gebruiker zich in de wereld te laten voelen. Zoals we al uitgebreid in hoofdstuk 3.2 hebben aangehaald kan een slechte opbouw van een virtuele applicatie lijden tot 'fysieke pijn' bij de gebruiker. Hier hebben we het dan eerder over onschuldige kwaaltjes zoals hoofdpijn, desoriëntatie, misselijkheid, ... Daarom is het belangrijk voor de ontwikkelaar

dat hij/zij hiermee rekening houdt bij het ontwikkelen van een VR applicatie. Het is immers niet de bedoeling dat een gebruiker de applicatie niet kan gebruiken omwille van deze fysieke gevolgen.

2.3.1 User Experience

Het is van belang dat de user experience bij een virtual reality applicatie optimaal is. Hiervoor heeft Beau Cronin inspiratie gehaald van de piramide van Maslow om zo op te lijsten wat de belangrijkste puntjes zijn om ongemak te vermijden (Cronin, 2015).

Ten eerste is er **Comfort**. Deze zijn eigenlijk ook al zeer duidelijk aangehaald in hoofdstuk 3.2. Met comfort heeft men het over de ervaring voor de gebruiker die bepaald wordt door onze zintuigen. Het belangrijkste is dus om de manier waarop onze zintuigen inwerken op de virtuele technologie. De hardware is hiervoor dus het meeste verantwoordelijk.

Dan komt **Interpreteerbaarheid**. Hoe realistisch is de virtuele wereld nog? Men bedoelt hier eerder het verschil tussen wat nog non-fictie is en wat fictie is. Een ervaring waarbij je door het heelal wordt gezogen op een ultieme snelheid met beelden die je nog nooit eerder gezien hebt zal dus voor de gebruiker zeer overweldigend zijn. Men kan wel enkele regels van de fysica gaan aanpassen, maar het blijft belangrijk dat deze ervaringen uitbreidingen moeten vormen op het echte leven dan een volledig nieuwe wereld te zijn.

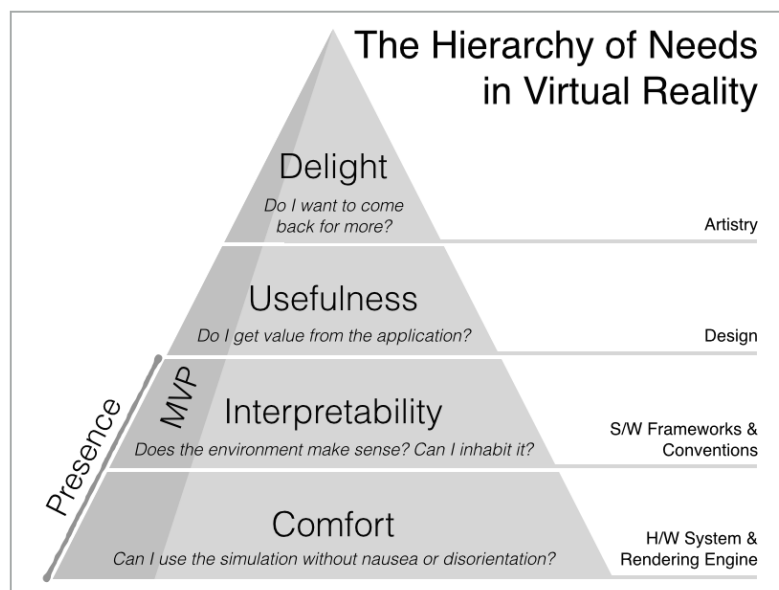
Vervolgens heb je **Bruikbaarheid**. Dit hangt af van applicatie tot applicatie, maar hier gaat het dus over hoe nuttig iets was, wat was de waarde achter de VR ervaring. Enkele voorbeelden hiervan zijn: Was het verhaal achter de film realistisch? Was de virtuele wandeling door het huis een aanzet tot denken om het huis te kopen? Het gaat hier dus vooral over de design van de applicatie.

Ten slotte heb je nog **Genot**. Dit is dan eigenlijk een uitbreiding op de bruikbaarheid. Hier heeft het men dus over het feit of je meer wilt van je virtuele ervaring. Denk hier maar aan de wandeling door het virtuele huis, heeft dat ook werkelijk je aangezet om het huis wel degelijk aan te kopen? Hier is de aandacht voor detail zeer belangrijk.

Het is duidelijk dat de 2 belangrijkste componenten van een goede user experience bij een VR applicatie het comfort en de interpreteerbaarheid zijn aangezien deze vooral de nadruk leggen op de aanwezigheid in de virtuele wereld.

2.4 JavaScript, HTML en CSS

Javascript is de taal waarop het framework ReactJS gebaseerd is. De taal is een objectgeoriënteerde taal maar kan ook zeker gebruikt worden als een functionele programmeertaal. Samen met HTML, CSS en Javascript vormen de basis van de hedendaagse websites op het wereldwijde web de dag van vandaag. Bij HTML gaat het over de inhoud van de pagina ,bij CSS over hoe het eruitziet en javascript zorgt dan voor de interactiviteit met de webpagina. Denk maar aan de animatie bij een uitschuifbaar menu of tekst die verandert



Figuur 2.3: De hiërarchie van de behoeften in virtual reality.

als je op een knop klikt.

2.4.1 HTML

HTML staat voor **HyperText Markup Language** en is een opmaaktaal voor documenten die werd uitgebracht in 1993. De taal HTML bestaat uit verschillende HTML elementen. Met behulp van deze elementen, ook wel tags genoemd, kan men structuur gaan creëren in stukken tekst. Een tag ziet er als volgt uit: '``'. Met dit element wil men een image aanhalen in het document. Daarnaast hebben deze elementen attributen. Deze attributen kunnen dan een bepaalde waarde hebben die het HTML element zal beïnvloeden. Bij deze `` tag kunnen we een attribuut 'src' toevoegen. Dit attribuut verwijst dan naar een source, een bron. Deze bron is dan de locatie van een afbeelding. Door het element en de attribuut weet HTML precies wat hij moet weergeven. Een html element zal altijd als volgt opgebouwd zijn:

```
1 <tag attribute1="value1" attribute2="value2">content
2 </tag>
```

Listing 2.1: HTML Element voorbeeld

```
1 <tag attribute1="value1" attribute2="value2" />.
```

Listing 2.2: HTML Element voorbeeld

Men kan deze HTML elementen ook in een ander HTML element gaan stoppen. Dit wordt ook wel nesting genoemd. De HTML elementen die zich dan binnen in een ander element bevinden noemen we ook wel de kinderen. Dit is een voorbeeld van hoe een zeer eenvoudig en klein HTML document er uit ziet.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Titel van de pagina</title>
5   </head>
6   <body>
7     <h1>Een hoofdtitel</h1>
8     <h2>Een subtitel</h2>
9     <p>Een stukje tekst</p>
10  </body>
11 </html>
```

Listing 2.3: Voorbeeld van een HTML bestand

2.4.2 CSS

Het enigste wat HTML nog mist is een manier om het document ook mooi voor het oog te maken. De oplossing hiervoor is Cascading Style Sheets, zoals de naam zegt, bladeren met de stijl op. Dit zijn bestanden waar voor alle HTML elementen een stijl kan worden gegeven. Op deze manier kan men dus een pagina vormgeving geven dat veel aantrekkelijker is voor het oog. De manier waarop CSS een bepaald HTML element aanspreekt is door middel van een klasse. Door deze klasse kan men het uitzicht van een specifiek HTML element gaan aanpassen die deze klasse. Het is ook bijvoorbeeld mogelijk om voor alle '<h1>' elementen een globale stijl te voorzien. Je kan een klasse toevoegen aan een html element door 'class=""' toe te voegen als attribuut. Bij een '<p>' zou dat er dan zo uitzien: '<p class="grootBlauw">'.

Een CSS document is net zoals HTML zeer simpel opgebouwd. In het volgende voorbeeld is er een stijl gegeven aan een klasse en een HTML tag:

```
1 h1 {
2   font-size: 36pt;
3   color: red;
4 }
5
6 .grootBlauw {
7   font-size: 20pt;
8   color: blue;
9 }
```

Listing 2.4: Voorbeeld van een CSS bestand

In het vorige voorbeeld hebben we ervoor gezorgd dat alle '<h1>' elementen een grootte zullen hebben van 36pt (point-size) en de kleur rood zal zijn. Daarnaast zullen alle elementen die de klasse '.grootBlauw' bevatten een tekst grootte hebben van 20pt en een blauwe kleur hebben.

2.4.3 Belangrijkste onderdelen

Javascript is de meest geavanceerde van de 3 en ook het belangrijkste voor deze bachelor-proef. Met javascript kan je dus je pagina interactief gaan maken en is daarom ook een volwaardige programmeertaal. Javascript heeft zoals elke programmeertaal bepaalde aspecten waar het zich onderscheidt van de andere talen. Javascript is vooral populair om het feit dat het door zeer veel browsers ondersteund wordt. Daarnaast is javascript ook een zeer flexibele taal. Daar waar een andere taal al snel een error voor zou aanrekenen, gaat javascript dit net niet doen.

Objecten

Zoals eerder al vermeld is javascript een object georiënteerde taal. Dit wil zeggen dat de taal met zogenaamde objecten werkt. Deze objecten kan je vergelijkingen met objecten in het echte leven en zijn verzamelingen van bepaalde eigenschappen die dan op zich een eigen naam/key (sleutel) en value (waarde) hebben (contibutors, 2018). De waarde van een eigenschap kan onder andere een getal zijn, maar ook een functie of een ander object kan perfect de waarde van een eigenschap zijn. Javascript heeft al enkele vooraf gedefinieerde objecten, maar je kan ook natuurlijk je eigen objecten gaan maken.

Laten we dit beter tonen a.d.h.v. een voorbeeldje. We gaan een hond object gaan definiëren in javascript. In code ziet dat er als volgt uit:

```
1 var hond = {  
2   naam: "Cyra",  
3   leeftijd: 3,  
4   ras: "Labrador",  
5   kleur: "Zwart"  
6 }
```

Listing 2.5: Hond object aanmaken

In dit stukje code hebben we dus een hond aangemaakt met de eigenschappen: naam, leeftijd, ras en kleur. Het grote voordeel aan javascript is, is dat het een zeer flexibele taal is. Bij andere object georiënteerde talen zou het moeilijker zijn om nu een extra eigenschap aan hond toe te voegen, we zouden al snel moeten gebruik maken van andere programmeer technieken zoals overerving. In javascript kunnen we bijvoorbeeld de eigenschap oogkleur zeer gemakkelijk toevoegen als volgt:

```
1 hond.oogkleur = "Blauw"
```

Listing 2.6: Eigenschap toevoegen aan en object

Het eerder aangemaakte hond object zal er dan zo uitzien:

```
1 {  
2   naam: "Cyra",  
3   leeftijd: 3,  
4   ras: "Labrador",  
5   kleur: "Zwart",  
6 }
```

```
6   oogkleur: "Blauw"  
7 }
```

Listing 2.7: Het nieuwe hond object

Variabelen

Bij het hoofdstuk 2.4.3 werd het hond object aangemaakt door 'var hond' te gebruiken. Dit is een goed voorbeeld van een variabele. Variabelen zijn vergelijkbaar met de X en de Y die we uit de wiskunde kennen, ze houden een bepaalde waarde bij. Een variabele kan je maken door eerst aan te geven welke soort variabele je wilt maken. In vele andere talen moet je voor een nummer iets anders gebruiken dan als je een stukje tekst in een variabele wilt stoppen. Maar ook hier zien we opnieuw de flexibiliteit die javascript te bieden heeft. Javascript heeft 3 verschillende soorten declaraties voor een variabele:

- **var**: Declareert een variabele, maar een waarde is optioneel.
- **let**: Declareert een lokale,'block-scoped' variabele, maar een waarde is optioneel.
- **const**: Declareert een lokale,'block-scoped' variabele, maar een waarde is verplicht en read-only.

Var is dus de meest flexibele declaratie. Bij een var zal de scope, hoe ver de variabele zich uitstrekt, binnen de gehele functie zijn. Bij een let daarentegen, zal dit enkel binnen het blok zichtbaar zijn. Met de volgende voorbeeld is dit goed aantoonbaar:

```
1 function eenFunctieMetVar() {  
2   //eenVar is hier zichtbaar  
3   for (var eenVar = 0; eenVar < 10; eenVar++) {  
4     //eenVar is hier zichtbaar  
5   }  
6   //eenVar is hier zichtbaar  
7 }
```

Listing 2.8: Een functie met var

```
1 function eenFunctieMetLet() {  
2   //eenLet is hier NIET zichtbaar  
3   for (var eenLet = 0; eenLet < 10; eenLet++) {  
4     //eenLet is hier zichtbaar  
5   }  
6   //eenLet is hier NIET zichtbaar  
7 }
```

Listing 2.9: Een functie met let

Zoals je kan zien, is een let een lokale declaratie. Ook wel block-scoped, binnen de accolades, genoemd. Een const is hetzelfde als een let, maar daarbij is het verplicht om een waarde te geven. Dit is omdat een const read-only is, het kan enkel gelezen worden en dus achteraf niet meer worden gewijzigd.

Functionies

De echte core functionaliteit van javascript zijn de functies. Een functie is een verzameling van verschillende declaraties, statements die bepaalde taken uitvoeren. Javascript zelf heeft al een aantal standaard gedefinieerde functies maar zoals bij objecten kan je deze functies ook zelf gaan schrijven. Dit is een voorbeeld van een simpele functie dus 2 nummers optelt:

```
1 function som(nummer1, nummer2) {  
2     var oplossing = nummer1 + nummer 2;  
3     return oplossing;  
4 }
```

Listing 2.10: Een functie die 2 getallen optelt

Zoals je dus kan zien is een functie zeer simpel opgebouwd. We beginnen altijd met eerst te zeggen dat het een functie is door 'function' te gebruiken. Daarna geven we een naam voor de functie, in dit geval is dat 'som'. Uiteindelijk gaan we aan deze functie 2 parameters meegeven, nummer1 en nummer2. Uiteindelijk kunnen we in de functie dan de uitvoering ervan gaan doen en daarna het resultaat retourneren. Vervolgens kunnen we deze functie gaan oproepen in andere functies.

```
1 function eenAndereFunctie() {  
2     var optelling = som(5,10);  
3     console.log(optelling)  
4 }
```

Listing 2.11: Een functie de som functie oproept

Deze functie zou dan de optelling gaan maken van 5 + 10 en vervolgens dit gaan weergeven in de console.

Asynchroon programmeren

Een iets geavanceerder aspect van javascript, maar wel een zeer groot pluspunt van de taal is dat er zeer gemakkelijk asynchroon kan geprogrammeerd worden. Asynchroon programmeren wil zeggen dat men meerdere taken tegelijkertijd kan uitvoeren. Dit gebeurt dan doormiddel van een 'Promise'. Dit is een object dat aangeeft dat het een waarde ofwel gaat ontvangen, ofwel niet zal ontvangen.

Frameworks

Waarschijnlijk het allergrootste reden dat javascript zoveel gebruikt wordt is het grote aanbod aan frameworks. Een framework is eigenlijk een grote verzameling van software componenten die gebruikt kan worden bij het programmeren van applicaties. Een aantal voorbeelden van bekende javascript frameworks zijn: Angular, React, Vue, Ember en Meteor

Aan het gebruiken van frameworks zijn een groot aantal voordelen verbonden (Korotya,

2017):

- **Efficiëntie:** Met een javascript framework kan men veel sneller een goed werkende website gaan opbouwen met een zeer goede structuur.
- **Veiligheid:** Javascript frameworks worden onderhouden door de community en worden dus ook veel getest. Hierdoor scoort de veiligheid bij javascript frameworks zeer hoog.
- **Kost:** De meeste javascript frameworks zijn open-source en gratis. Hierdoor en door het feit dat het ontwikkelen zo snel gaat, kost het een pak minder voor een goede website te ontwikkelen.

2.5 De mogelijke VR frameworks naast React 360

Het ontwikkelen van virtual reality applicaties gaat natuurlijk niet zomaar en vereisen een goede technische kennis. Zeker het feit dat je in een wereld zit waar er volledig om zich kan worden gekeken en ook eventueel bewogen worden is natuurlijk een grote uitdaging. Om dit allemaal makkelijker te maken zijn er momenteel al een aantal frameworks beschikbaar zoals React 360 maar naast dit framework bestaan er nog een aantal andere mogelijke frameworks (Lab, 2017). Wat opvalt is dat enkel A-Frame en React360 momenteel nog actieve projecten zijn.

2.5.1 A-Frame

Het mozilla team begon met virtual reality in de browser te tonen aan de buitenwereld door A-Frame. Met A-Frame kunnen ontwikkelaars 3D werelden gaan bouwen die bestemd zijn voor virtual reality. Doordat A-Frame beschikbaar is in de browser door simpelweg HTML elementen te gaan gebruiken is het ook voor meerdere toestellen beschikbaar zoals mobile en desktop. Net zoals React 360, maakt A-Frame gebruik van javascript en heeft een goede performantie.

2.5.2 Primrose

Primrose is net zoals A-Frame en React360 een VR framework dat ook in de browser draait door behulp van javascript. Het doel bij primrose is om letterlijk omgeving te gaan maken. Dit ziet er in de code zo uit:

```
1 var env = new Primrose.BrowserEnvironment({
2   backgroundColor: 0x000000,
3   groundTexture: "deck.png",
4   useFog: true
5 });
```

Listing 2.12: Voorbeeld van primrose omgeving

Aan deze omgeving kan men dan UI elementen toevoegen zodat men hiermee interacties kan uitvoeren. Net zoals A-Frame en React360 ondersteund het meerdere toestellen.

2.5.3 Agron.js

AgronJS is niet een VR framework maar wel een AR (Augmented Reality) framework. Met AgronJS is het mogelijk om AR applicaties in elke browser uit te voeren. Het werd ontwikkeld voor een onderzoeks project. Daarom wordt het momenteel ook niet meer actief geupdate en werkt zoals alle andere opgenoemde frameworks hier, met javascript.

3. Methodologie

3.1 Het framework, React 360, bestuderen

We gaan eerst kort React.JS zelf bekijken. Hiermee wordt React 360 namelijk opgebouwd en is dus cruciaal om hier ook kennis over te hebben. Daarna zullen we de belangrijkste delen van het React 360 framework gaan bekijken. Zowel de manier van opbouw in de applicatie, als de beschikbare componenten in het framework. Ten slotte nemen we een kijkje naar hoe zo een React 360 applicatie presteert.

3.2 Ervaring van virtual reality voor een gebruiker

Daarna kunnen we met al de informatie die we verworven hebben doormiddel van React 360 enkele eenvoudige applicaties gaan maken. Deze applicaties zullen getest worden door enkele personen en zij zullen dan hun mening geven a.d.h.v. een korte vragenlijst. Deze zal dan een beter inzicht geven in hoe een persoon een virtual reality applicatie ervaart. App 1 De eerste applicatie zal de simpelste zijn. Dit is een applicatie waarbij de nadruk ligt op het simuleren van een echt omgeving. Hiervoor zal een gebied dat echt bestaat in 360° worden weergegeven met het bijhorende omgevingsgeluid. App 2 De 2de applicatie zal meer de nadruk leggen op de interactie met de virtuele wereld. Er zal hier dus vooral gekeken worden naar de input mogelijkheden en hoe de gebruiker dit ervaart. App 3 ...?

3.3 De doeleinden en stand van zaken van React 360

Ten slotte zullen we een besluit opmaken van React 360. We gaan bekijken voor welke doeleinden we React 360 zouden kunnen gebruiken en hoe het scoort op vlak van performantie. Hier gaan we dan een stand van zaken opstellen hoe React 360 scoort als framework voor virtual reality. We zullen dan ook een conclusie kunnen opmaken van dit onderzoek.

4. React 360

React VR -> React 360

4.1 React.js

4.1.1 JSX

4.1.2 Components en props

4.1.3 State en levenscyclus

4.1.4 Events

4.2 Het ruimtelijke systeem

4.3 Enkele belangrijke componenten

4.3.1 View

4.3.2 Environment

4.3.3 Video

4.3.4 VrButton

4.4 Input in VR

4.5 Developing in VR

4.6 Prestaties

5. Conclusie

Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit

lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consectetur libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem.

Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Hier introduceer je werk. Je hoeft hier nog niet te technisch te gaan.

Je beschrijft zeker:

- de probleemstelling en context
- de motivatie en relevantie voor het onderzoek
- de doelstelling en onderzoeksvraag/-vragen

A.2 State-of-the-art

Hier beschrijf je de *state-of-the-art* rondom je gekozen onderzoeksdomein. Dit kan bijvoorbeeld een literatuurstudie zijn. Je mag de titel van deze sectie ook aanpassen (literatuurstudie, stand van zaken, enz.). Zijn er al gelijkaardige onderzoeken gevoerd? Wat concluderen ze? Wat is het verschil met jouw onderzoek? Wat is de relevantie met jouw onderzoek?

Verwijs bij elke introductie van een term of bewering over het domein naar de vakliteratuur, bijvoorbeeld (Doll & Hill, 1954)! Denk zeker goed na welke werken je refereert en

waarom.

Je mag gerust gebruik maken van subsecties in dit onderdeel.

A.3 Methodologie

Hier beschrijf je hoe je van plan bent het onderzoek te voeren. Welke onderzoekstechniek ga je toepassen om elk van je onderzoeksvragen te beantwoorden? Gebruik je hiervoor experimenten, vragenlijsten, simulaties? Je beschrijft ook al welke tools je denkt hiervoor te gebruiken of te ontwikkelen.

A.4 Verwachte resultaten

Hier beschrijf je welke resultaten je verwacht. Als je metingen en simulaties uitvoert, kan je hier al mock-ups maken van de grafieken samen met de verwachte conclusies. Benoem zeker al je assen en de stukken van de grafiek die je gaat gebruiken. Dit zorgt ervoor dat je concreet weet hoe je je data gaat moeten structureren.

A.5 Verwachte conclusies

Hier beschrijf je wat je verwacht uit je onderzoek, met de motivatie waarom. Het is **niet** erg indien uit je onderzoek andere resultaten en conclusies vloeien dan dat je hier beschrijft: het is dan juist interessant om te onderzoeken waarom jouw hypothesen niet overeenkomen met de resultaten.

Bibliografie

- Abarraera. (2017, november 20). Virtual Reality isn't functional yet, here is why. Verkregen van <https://thealeph.com/articles/2017/11/virtual-reality-vr-oculus-htc-why/>
- Azuma, R. T. (1997). *A Survey of Augmented Reality*. Hughes Research Laboratories.
- contributors, M. (2018). JavaScript. Verkregen van <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- contributors, W. (2018). Virtuele werkelijkheid. Verkregen van https://nl.wikipedia.org/wiki/Virtuele_werkelijkheid
- Cronin, B. (2015, januari 19). The hierarchy of needs in virtual reality development. Verkregen van <https://medium.com/@beaucronin/the-hierarchy-of-needs-in-virtual-reality-development-4333a4833acc>
- Doll, R. & Hill, A. B. (1954). The mortality of doctors in relation to their smoking habits: a preliminary report. *British Medical Journal*, 328(7455), 1529–1533.
- Korotya, E. (2017, januari 19). 5 Best JavaScript Frameworks in 2017. Verkregen van <https://hackernoon.com/5-best-javascript-frameworks-in-2017-7a63b3870282>
- Lab, U. (2017, mei 29). 5 Web VR Frameworks to Help Developers Build Interesting Design. Verkregen van <https://uiux.cc/blog/5-web-vr-frameworks-to-help-developers-build-interesting-design/>
- Rouse, M. (2011). Stereoscopy. Verkregen van <https://whatis.techtarget.com/definition/stereoscopy-stereoscopic-imaging>
- Sherman, W. (2000). *Understanding Virtual Reality: Interface, Application, and Design*.