

# **NEW TECHNOLOGIES**

**ALEX, FRANS, JOEY & STEN**



# INHOUD

<b>INLEIDING</b>	<b>03</b>
EXPERIMENT 01	
<b>LEAP MOTION &amp; VR</b>	<b>04 - 08</b>
Waarom Leap Motion & VR?	04
Over Leap Motion	05
Waarom Gestures?	06
Toegepaste gestures	06
Het resultaat	07
EXPERIMENT 02	
<b>TOBII EYE TRACKER</b>	<b>09 - 14</b>
Waarom de Tobii Tracker?	09
Opzetten	10
Doel	10
Onderzoeks vragen	10
Verwachte problemen	10
Interessante bevindingen	10
Problemen	11 - 12
Antwoord op onderzoeks vragen	13
Wat zouden wij nog willen doen met de Tobii Eyetracker?	14
<b>TAAKVERDELING</b>	<b>15</b>

# INLEIDING

De afgelopen 7-8 weken hebben wij tijdens New Technologies veel geëxperimenteerd met tools waaronder VR, Leap Motion en de Tobii Eye Tracker.

Ons eerste voornamelijk doel was het combineren van twee interessante tools met elkaar. Dat was erg interessant omdat we veel onderzoek moesten doen hoe we de twee vrij verschillende tools met elkaar konden verbinden.

Ons tweede experiment liet ons experimenteren met iets wat we nog nooit hadden gebruikt en voor sommigen zelfs 'nog nooit van had gehoord'. En dat opende voor ons een nieuwe inzicht in de toepassing van deze tool in gaming, maar ook in andere velden.

New Technologies heeft ons uitgedaagd om op een andere manier te kijken naar tools en hoe je ze op een innovatieve manier kunt inzetten. Dat is iets wat we uit



**EXPERIMENT #01**

# **LEAP MOTION & VR**

## **Waarom Leap Motion & VR?**

Voor ons eerste experiment hebben wij gekozen voor de combinatie VR en Leap Motion. De reden hiervoor was dat wij nog niet erg bekend waren met de Leap Motion. Het idee was om dus een combinatie te pakken die we nog nooit hadden gezien of zelf ingezet.

Terwijl het hedendaagse VR nog vooral bestaat uit controllers, is Leap Motion (hand gestures) een nog nieuwere manier van het besturen van VR. Wij zochten echter wel een setting waarin we het konden gebruiken. Met drie game-developers en één interaction designer hebben we gekozen voor de gulden middenweg, namelijk het inzetten van hand gestures in een smart home door middel van VR (prototype).

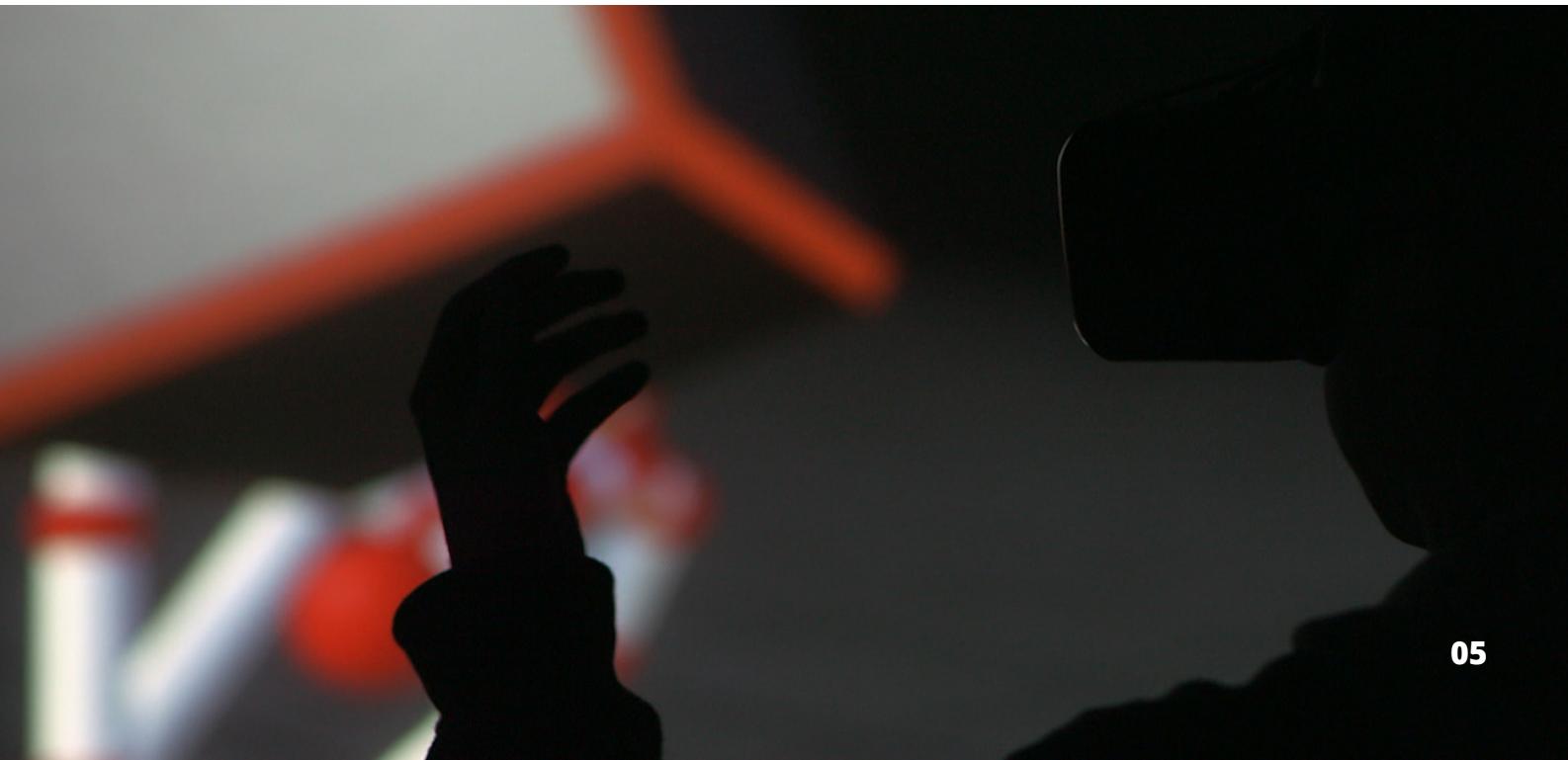
# **Over Leap Motion.**

Omdat Leap Motion nieuw voor ons was, hebben we hier het meeste onderzoek naar gedaan. We hebben vooral gekeken naar wat er wel en wat er niet mogelijk is met de Leap Motion.

Leap Motion is een vrij nauwkeurige manier van het registreren van hand gestures. Nauwkeuriger en laagdrempeliger dan bijvoorbeeld de kinect, die al snel andere elementen registreert. De Leap Motion leek ons daarom de meest interessante tool om dit experiment mee te doen.

Echter blijkt Leap Motion wel enkele beperkingen te kennen, zoals het aantal mogelijke hand gestures. Je kunt namelijk 'slecht' vier voorgeprogrammeerde gestures gebruiken. En het toevoegen van eigen gestures is geen optie bij de leap. Er bestaan echter wel weer verschillende soorten manieren om het systeem aan te passen.

Maar voor dit onderzoek leek het ons goed om door te gaan op deze gestures, en vervolgens zelf wat gestures te verzinnen op basis van real-life onderzoek.



# Waarom gestures?

Eén van de vragen tijdens de presentaties was:

## Waarom gebruiken jullie geen voice commando's, in plaats van gestures?

Simpel, wij denken namelijk dat gestures de gebruiker een gevoel van controle geven. Spraakcommandos worden uitgevoerd door je 'digitale assistent'. Met gestures heb je zelf als mens zijnde de controle over de dingen die je doet. Ook kun je op deze manier preciezer zijn in de aansturing, zo kun je bijvoorbeeld een gordijn niet alleen 'open' en 'dicht' doen maar ook ergens halverwege laten hangen. Daarnaast geloven wij niet in een digitale assistent, uit recente cijfers blijkt dat mensen het nauwelijks actief gebruiken in hun huis.

## Toegepaste gestures

Na aan de slag te zijn gegaan met Leap Motion in Unity kwamen we er al snel achter dat de ingebouwde leap-gestures niet meer beschikbaar waren voor Unity 5.6 en later. Uit de downloadbare SDK waren alle classes gehaald die iets met gestures te maken hadden. Leap vond namelijk dat deze gestures te inconsistent werkten om ze te blijven ondersteunen.

Voor ons was dit erg vervelend, want hoe nu verder? Na een korte brainstorm sessie hebben we alsnog besloten om een eigen gesture framework op te zetten. Deze werkt als volgt:

Van iedere vinger wordt een positie bijgehouden ten opzichte van de handpalm. Toen hebben we een functie geschreven die checkt op de relatieve afstanden tussen iedere vinger. Voldoen



de posities bijvoorbeeld aan de conditie 'IsPointing', waarbij 'IsPointing' een vooraf vastgestelde waarde eist van de vingers, dan schiet die gesture op true.

Omdat alle gestures los worden getrackt, is het heel eenvoudig om complexere gebaren te detecteren. Bijvoorbeeld niet alleen een vuist of een wijzende beweging, maar bijvoorbeeld ook het "Sign of the Horns" gebaar dat vaak gebruikt wordt tijdens metal concerten.

Met deze vrijheid zijn we gaan experimenteren, met de volgende video als resultaat:

Korte demonstratie van ons gesture-framework  
<https://streamable.com/i6vf5>



# Het resultaat

Met het maken van een eigen gesture framework voor alle gebaren zijn we langer bezig geweest dan verwacht. We hebben daarop besloten om het framework verder uit te werken en om de toepassing in een smart-home achterwege te laten. Dit vonden we zeker jammer, maar aan de andere kant is het enorm leerzaam geweest om alles from-scratch op te bouwen.

## Het framework kent drie main classes:

- Gestures
- Motions
- Gesture control

In de Motions class worden alle motions opgeslagen die de gebruiker kan maken. Alle vingers worden hier los bijgehouden in statics zodat de andere classes deze informatie op kunnen vragen bij deze class. In de gestures class staan de gebaren die geregistreerd kunnen worden door het framework. Denk hierbij aan gestures zoals het maken van een vuist of een vlakke hand. Ten slotte gebruiken we een Gesture management class om deze functies aan te roepen.

Het toepassen van ons framework is eenvoudig.

Bijvoorbeeld:

**"if (Motions.GetHandGesture(rightHand, Gestures.POINT))"**

*Gaat op true zodra de rechterhand een wijzende beweging maakt.*

Ook non-bools zoals:

**"if (Motions.GetHandRoll(leftHand) > 1f)"** worden ondersteund.

De GetHandRoll functie geeft een waarde tussen de -5 en de 5. Hiermee kan je bijvoorbeeld heel accuraat objecten rotaten of verplaatsen.



**EXPERIMENT #02**

# **TOBII EYE TRACKER**

## **Waarom de Tobii Eye Tracker?**

De reden dat we voor de Tobii Eye Tracker zijn gegaan is omdat we er alle vier nog niet bekend mee waren. Tijdens de lessen demonstreerde Valentijn de Tobii Eye Tracker waardoor we erg nieuwsgierig werden om er mee te gaan experimenteren.

De Tobii Eye Tracker biedt nieuwe manieren om te gamen door het besturen of bewegen van objecten binnen een game.

Naast het feit dat de Tobii Eye Tracker natuurlijk veel toepassingen heeft binnen games, die Valentijn ook demonstreerde. Zagen we ook veel potentie rondom marketing doeleinden. Om die reden hebben we ook een klein concept ontwikkeld.

We hebben voornamelijk onderzoek gedaan naar de werking van de Eye Tracker. Ook de eventuele problemen die we zagen tijdens user testing.

# **Opzetten**

Koppelen van de Tobii eye tracker ging eenvoudig door de uitgebreide documentatie en demo content.

## **Doe**

Een spel maken waar de speler een bal moet volgen met zijn ogen. In de omgeving worden ander gekleurde ballen rond gestuiterd om de speler af te leiden. Middels een timer willen wij een high score gebaseerd spel maken. In het kort: 'Wie het langst kan kijken'. Tevens zullen er gelijk gekleurde ballen in het scherm ontstaan, dit om de speler uit te lokken daarnaar te kijken.

## **Onderzoeksvragen**

Hoe reageren de ogen op het intensieve gebruik van de ogen als een middel van input. Wat zijn de instinctieve acties van een speler die zijn ogen kan gebruiken bij het spelen. Wat is het effect van feedback krijgen over waar je kijkt.

## **Verwachte problemen**

Het volgen en dit meten met de tracker gaat moeilijk zijn omdat het een constante verandering is die gevolgd moet worden met de ogen. Als de speler een tel van de bal af zit dan is de score al af.

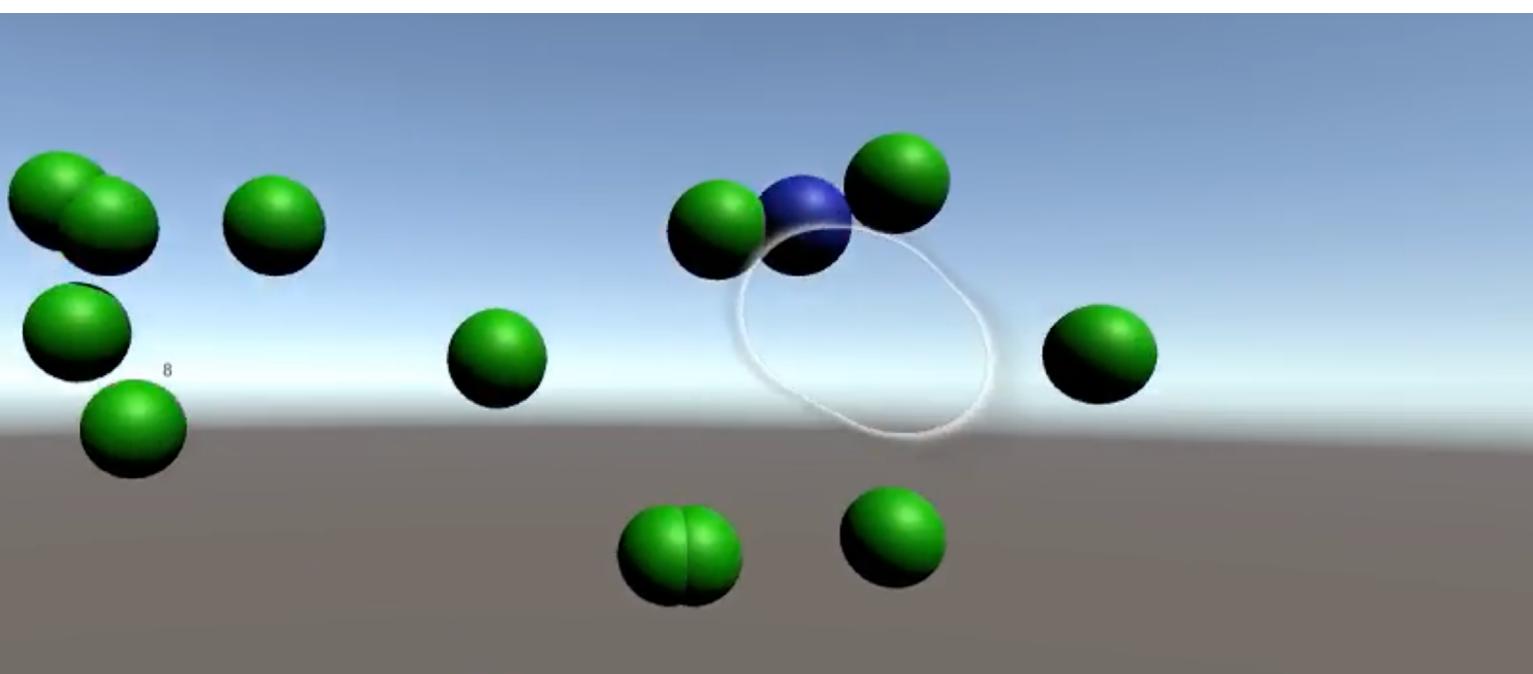
## **Interessante bevindingen**

- Als je scheel kijkt gaat de Tobii raar doen, deze kan dan niet meer meten wat er gebeurd. En de selectie beweegt dan snel heen en weer tussen beide ogen.
- Het toevoegen en eigen maken van de demo scripts werkt heel goed en eenvoudig.

## Na het experiment

# Problemen

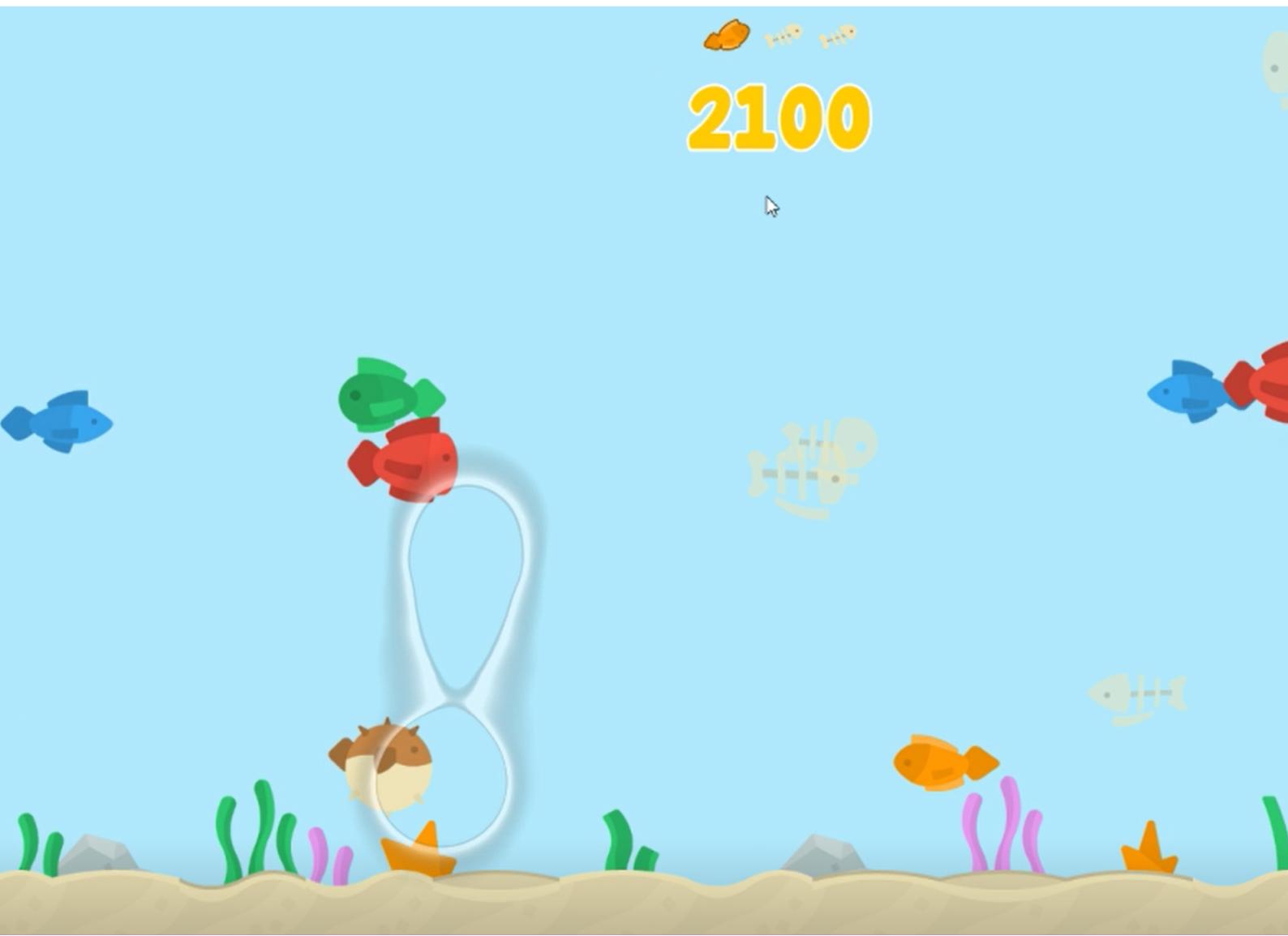
De ogen volgen eigenlijk het object, waardoor je vrij lastig het volledige object in de marker kan krijgen. Hieronder een voorbeeld, de ogen lopen regelmatig achter (zoals hier te zien: <https://youtu.be/hhxkHr7a39Y>). Naast de kleuren die veranderen van de bal zoek je met je ogen naar meer aanwijzingen dat je iets 'goed' doet. Hierdoor ben je snel geneigd om te kijken naar de score tekst.



Daarnaast voelt het niet natuurlijk/instinctief om met je ogen te bewegen en je hoofd stil te houden. Zoals te zien is in deze video: <https://youtu.be/i615JLqtS1s> probeert het testpersoon door middel van het hoofd draaien verder te kijken dan mogelijk is.

Bij het vis spel ben je geneigd naar de gevaren te kijken, echt ga je hierdoor op het gevaar af. Het is dus weer heel erg tegenstrijdig met je instincten om weg te kijken van het gevaar. Het is erg instinctief om de gevaren te analyseren. In deze video <https://youtu.be/MBof0b1-2tk> is dan ook te zien dat de speler eerst kort kijkt naar alle gevaarlijke vissen

Bij het vis spel ben je geneigd naar de gevaren te kijken, echt ga je hierdoor op het gevaar af. Het is dus weer heel erg tegenstrijdig met je instincten om weg te kijken van het gevaar. Het is erg instinctief om de gevaren te analyseren. In deze video <https://youtu.be/MBof0b1-2tk> is dan ook te zien dat de speler eerst kort kijkt naar alle gevaarlijke vissen.



Hier is te zien dat zodra de speler de gevaarlijke vis ziet, hij direct wegglijkt om hem te ontwijken. Dit gebeurde pas na drie keer spelen.

# **Antwoord op de onderzoeksvragen**

## **Hoe reageren de ogen op het intensieve gebruik van de ogen als een middel van input?**

Over het algemeen zijn de ogen een middel van observatie. Hier halen wij veel informatie over de omgeving, huidige status en sfeer vandaan. Doordat de ogen input zijn, wordt de functie heel anders, hierdoor raak je het overzicht kwijt. Ook worden de ogen sneller moe omdat je constant gefocused bent om te kijken en zo dus vergeet te knipperen. Als je een bewegend object volgt worden de spieren van je oog constant aangespannen, dit merk je ook.

## **Wat zijn de instinctieve acties van een speler die zijn ogen kan gebruiken bij het spelen?**

Al gauw werd duidelijk dat testpersonen geneigd zijn hun hoofd mee te draaien als zij het scherm willen laten bewegen. Daarnaast zijn mensen niet gewend hun ogen te gebruiken als input, omdat zij hier onbekend mee zijn en feedback krijgen waar ze kijken is het lastiger om te verwerken wat er gebeurd of hoe je moet reageren. Tevens zorgt de feedback (De grijze cirkel waar je kijkt) ervoor dat je heel snel focust op de feedback en dus constant analyseert wat er gebeurd.

## **Wat is het effect van feedback krijgen over waar je kijkt?**

De eerste reactie is nieuwsgierigheid en onder de indruk zijn van het feit dat je precies ziet waar je kijkt.

# Wat zouden wij nog willen doen met de Tobii Eyetracker?

Het lijkt ons leuk om de marketing mogelijkheden verder uit te werken en te testen. Ook zouden wij de highscore game uit willen werken.



# Taakverdeling



**Alex van der  
Linden**

VR & Leap Motion  
(development)  
Research



**Frans Huntink**

VR & Leap Motion  
(development)  
Research



**Joey Versepout**

IAD Research  
Brainstorm  
Documentatie



**Sten Duindam**

Tobii Eye Tracker  
(development)  
Research  
Brainstorm

