

HAUTE TECHNIQUE

Ontwikkelen van een applicatie voor het aansturen van DMX apparaten aan de hand van locatie updates

Extern bijlagenboek

Student	: Jan de Boer
Studentnummer	: 11071133
Onderwijsinstelling	: Haagse Hogeschool te Delft
Opleiding	: Technische Informatica
Afstudeerperiode	: 1 september 2015 t/m 8 januari 2016
Begeleider	: Dhr. T. Andrioli
Tweede examinerator	: Dhr. J. Peski
Bedrijf	: HAUTE TECHNIQUE
Bedrijfsmentor	: Sander ter Braak
Datum	: 8 januari 2016
Versie	: V1.0

Inhoudsopgave

Bijlagen A	Afstudeerplan	1
Bijlagen B	Plan van aanpak	2
Bijlagen C	Requirements Document	3
Bijlagen D	Hardware testrapport	4
Bijlagen E	User interface rapport	5
Bijlagen F	Scrumrapport	6
Bijlagen G	Handleiding effect schrijven	7
Bijlagen H	Ontwerprapport	8

Bijlage A

Afstudeerplan

Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2015-2.1 (start uiterlijk 31 augustus 2015)

Startdatum uitvoering afstudeeropdracht: 1 september 2015

Inleverdatum afstudeerdossier volgens jaarrooster: 8 januari 2016

Studentnummer: 11071133
Achternaam: ~~mvw~~ / dhr de Boer (*) *weghalen niet van toepassing*
Voorletters: J
Roepnaam: Jan
Adres: Doctor van Noortstraat 113
Postcode: 2266GV
Woonplaats: Stompwijk
Telefoonnummer: -
Mobiel nummer: 0681020166
Privé emailadres: jdboer@hotmail.com

Opleiding: Technische Informatica
Locatie: ~~Den Haag~~ / Delft / Zoetermeer (*) *weghalen niet van toepassing*
Variant: voltijd

Naam studieloopbaanbegeleider: J. van Peski
Naam begeleidend examiner: A. Andrioli
Naam tweede examiner: J. van Peski

Naam bedrijf: Haute Technique
Afdeling bedrijf:
Bezoekadres bedrijf: Jaarbeursplein 6
Postcode bezoekadres: 3521AL
Postbusnummer: -
Postcode postbusnummer: -
Plaats: Utrecht
Telefoon bedrijf: 030 820 12 71
Telefax bedrijf: -
Internetsite bedrijf: www.hautetechnique.com

Achternaam opdrachtgever: ~~mvw~~ / dhr ter Braak (*) *weghalen niet van toepassing*
Voorletters opdrachtgever: S
Titelatuur opdrachtgever: master of Arts in Design for Digital Cultures
Functie opdrachtgever: Eigenaar
Doorkiesnummer opdrachtgever: -
Email opdrachtgever: sander@hautetechnique.com

Achternaam bedrijfsmentor: ~~mvw~~ / dhr ter Braak (*) *weghalen niet van toepassing*
Voorletters bedrijfsmentor: S
Titelatuur bedrijfsmentor: master of Arts in Design for Digital Cultures
Functie bedrijfsmentor: Eigenaar
Doorkiesnummer bedrijfsmentor: -
Email bedrijfsmentor: sander@hautetechnique.com

NB: mag dezelfde zijn als de opdrachtgever

Doorkiesnummer afstudeerder: -
Functie afstudeerder (deeltijd/duaal): -

Titel afstudeeropdracht: Ontwikkelen van een locatie link bij Haute Technique

Opdrachtomschrijving

1. Bedrijf

Haute Technique is een jong bedrijf die zich bezig houden met de nieuwe media. Ze ontwikkelen unieke oplossingen voor problemen die hun klanten hebben. Elke klant komt met zijn eigen probleem(en) daarom heeft Haute Technique ervaring op tal van verschillende gebieden. Zo hebben ze games, interactie installaties, aanraakgevoelige tafels, prototypes, websites en mobile applicaties gemaakt.

Bij Haute Technique werken drie mensen in vaste dienst. Per project dat wordt gedaan worden de juiste freelancers ingehuurd. Op deze manier is Haute Techniek flexibel in de opdrachten die ze kunnen aannemen en hebben ze altijd de juiste man voor de "Job".

2. Probleemstelling

Haute Technique heeft software ontwikkeld waarmee het mogelijk is om de bewegingen van iemands armen te tracken. De software spreekt via Bluetooth de twee armbanden aan (elk om 1 hand) om te bepalen in welke positie de armen zich bevinden. Deze data wordt vervolgens gebruik om de verlichting en de videobeelden aan te sturen. Dit gebeurt respectievelijk via het DMX512 protocol en HDMI.

Deze software wordt op dit moment gebruikt bij de shows van Armin van Buuren om zo een interactievere show te geven. Het huidige systeem is echter alleen in te zetten als de locatie van Armin tijdens de show niet veranderd. Daarom is het gewenst om naast de positie van de armen van Armin ook de locatie op het podium van Armin van Buuren te tracken. Op deze manier kunnen de shows nog interactiever worden en kan Armin steeds meer de verloop van de show zelf bepalen. In plaats dat alle show elementen voorgeprogrammeerd zijn.

Haute Technique heeft zelf niet de benodigde kennis in-house om dit systeem te ontwikkelen. Daarom wordt de expertise van Sendrato gebruikt om samen tot een werkende oplossing te komen. Sendrato heeft al veel ervaring op het gebied van locatie bepaling. Zo leveren zij oplossingen voor grote nationale en internationale festivals op het gebied van: crowdcontrol en beveiliging. Dit zijn echter systemen die beperkte nauwkeurigheid beiden, tot ongeveer 2 meter. De gewenste nauwkeurigheid voor dit systeem is een stuk groter. Haute Technique heeft de wens om een nauwkeurigheid van 2 centimeter te bereiken.

Haute Techniek heeft de wens om een generieke applicatie te ontwikkelen die het gemakkelijk maakt om te kunnen reageren op de locatie van een persoon of personen om zo bijvoorbeeld verlichting of video beelden aan te passen. Deze applicatie moet instaat zijn om de verkregen X, Y en Z waarden van de hardware om te zetten naar ruimtelijke waardes en biedt deze informatie aan in een toegankelijke interface. Hierdoor kunnen andere applicaties / apparaten reageren op de locaties van een persoon of personen op het podium.

In eerste instantie heeft Haute Techniek de wens om deze software in te zetten om de locatie van Armin van Buuren te tracken om zo een interactievere show te kunnen geven. Maar Haute Techniek heeft ook de wens om deze software te gebruiken om volg lichten aan te sturen om deze wens te kunnen verwezenlijken zal binnen dit project dus ook een prototype worden gemaakt die het mogelijk maakt op volg lampen automatisch personen te laten volgen.

3. Doelstelling van de afstudeeropdracht

Het ontwikkelen van een generieke applicatie die het mogelijk maakt om de locatie van een persoon of personen te volgen op een podium en deze informatie aanbiedt door middel van een interface. Hierbij is een koppeling gemaakt met het bestaande aansturingssysteem dat Haute Techniek gebruikt voor de aansturing van de lichten en de videobeelden en is er een prototype applicatie geschreven voor een “automatische” volglamp.

4. Resultaat

Haute Techniek beschikt aan het einde van dit project over een applicatie die locatie data om kan zetten in bruikbare data en aanbiedt in een gebruiksvriendelijke interface. Hierbij is een koppeling gemaakt met hun bestaande aansturingen systeem voor shows van Armin van Buuren waardoor er een extra “dimensie” aan de shows kan worden toegevoegd namelijk de locatie gevoeligheid. Ook wordt het voor Haute Techniek makkelijker om andere applicaties of hardware locatie gevoelig te maken door gebruik te maken van de interface die de geschreven applicatie aanbiedt. Naast deze applicatie beschikt Haute Techniek aan het einde van dit project ook over een prototype applicatie die het automatisch volgen van een persoon of personen met een volglamp mogelijk maakt.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

De eerste weken van het afstuderen worden gereserveerd voor het opstarten van het project en het onderzoeken van de verschillende onderdelen van het systeem. Het systeem is namelijk op te delen in een aantal verschillende subsystemen. De werking van het locatiesysteem is als volgt: Er worden een aantal ontvangers op het podium geplaatst. Deze moeten vervolgens worden gekalibreerd om zo de locatie te kunnen bepalen. Na de kalibratie is het mogelijk om de locatie van een persoon realtime te volgen dit gebeurt aan de hand van een antenne die de persoon bijzigt draagt. Deze antenne zendt data uit. Aan de hand van AoA (angle of arrival) of ToA (time of arrival) kan vervolgens de locatie worden bepaald. Deze locatie gegevens moeten vervolgens worden omgezet naar ruimtelijke waarden en worden aan geboden in een toegankelijk interface

De volgende subsystemen zijn te onderscheiden:

1. Uitlezen van de sensoren en de ruimtelijk waarden berekenen.
2. Een subsysteem om de sensoren te kalibreren zodat de gewenste nauwkeurigheid kan worden gehaald.
3. Generieke interface die de locatie gegevens aanbiedt.

Er zijn grof weg ook vier onderzoeken die gedaan moeten worden. Zie Figuur 1. Hiernaast moet ook de creatieve kant van de locatie bepaling onderzocht worden. Wat kan Haute Techniek met deze nieuwe technologie en hoe kan dit de shows van Armin verbeteren? Dit loopt naast de verschillende technische onderzoeken. Op deze manier kunnen zaken die in dit onderzoek worden gevonden meteen worden meegenomen in de andere (technische) onderzoeken.

Het is belangrijk om tijdens de eerste weken de kaders / beperkingen van het systeem te onderzoeken. Hier kunnen de volgende vragen naar voren komen: welke apparatuur wordt er gebruikt om de locatie te bepalen, wat heeft invloed op de nauwkeurigheid van het systeem of in welke landen mag dit gebruikt worden.

De eerste weken van het project zijn gereserveerd voor het opzetten van het project en het onderzoeken van de verschillende facetten van het systeem zie Figuur 1. Ook wordt er in deze weken getest met de hardware (sensoren en antenne) die Sendrato heeft aangeleverd. Tijdens deze tests wordt bepaald of de geleverde hardware voldoet aan de gestelde eisen. Hierna wordt er een prototype ontwikkeld die de X, Y en Z waarden van de sensoren kan omzetten naar ruimtelijke waarden. De informatie die hierbij en bij de voorafgaande iteraties is vergaard wordt vervolgens gebruikt om het systeem te ontwerpen hierbij wordt ook het kalibratie systeem meegenomen. Na het ontwerpen kan er begonnen worden met het implementeren van het design. Dit zal een iteratief proces worden van implementeren, testen en fouten uit het originelen design corrigeren. Nadat de applicatie is ontwikkeld moet er een koppeling worden gebouwd met het bestaande systeem dat Haute Techniek gebruik voor het aansturen van de lichten en de videobeelden.

De laatste weken van het project zijn gereserveerd voor het implementeren van een prototype van een “automatisch” volglicht. Dit volglicht gebruikt de interface van de ontwikkelde applicatie om zo de locatie van een persoon te verkrijgen. Het volglicht kan deze persoon vervolgens automatisch tracken.

Week	Actie
1	Plan van Aanpak
2	Testen van de hardware
	Onderzoek naar:
	- Uitlezen van de sensoren
	- Kalibratie van de sensoren
	- Interface naar 3 ^e systemen
	- Meerdere personen tracken op een podium
3	Opzetten ontwikkelomgeving
4	Prototype ontwikkelen
5	X, Y en Z waarden van de sensoren omzetten naar ruimtelijk waarden
6	Systeem ontwerpen met input van tot nu toe vergaarden informatie
7	Implementeren van de applicatie
8	
9	
10	
11	Link naar bestaande aansturingssysteem
12	
13	Prototype van een automatische volglamp
14	
15	Kwaliteitscontrole en mogelijke uitloop

Figuur 1 Planning

6. Op te leveren (tussen)producten

Er zijn verschillende tussenproducten te onderscheiden dit zijn:

- Een onderzoeksrapport van de uitgevoerde onderzoeken
- De applicatie met de daarbij horende source code van de software die de locatie van een persoon of personen bepaald en aanbiedt in een toegankelijke interface.
- Documentatie die het softwaresysteem beschrijft hieronder vallen de volgende documenten: ontwerpdocument, testrapporten en een handleiding van het systeem.
- Prototype applicatie die de volglamp aanstuurt met de daarbij horende source code.

7. Te demonstreren competenties en wijze waarop

Ontwerpen van een technisch informatie systeem (C8)

De applicatie die aan het einde van dit project wordt opgeleverd dient als prototype. Daarom is het van belang dat de geschreven code duidelijk en gestructureerd is. Op deze manier kan het in de toekomst makkelijker worden uitgebreid en hoeft er minder code geschreven te worden.

Kiezen van een ontwikkelstrategie en ontwikkelmethodiek (A4)

Om het project beheersbaar te houden moet er een bijpassende ontwikkelstrategie en methode worden gekozen. Het is van belang dat deze methode voldoet aan de eigenschappen van het dit project.

Creëren en innoveren (H8)

Locatie bepaling is op zich zelf niet nieuw maar de toepassing in deze industrie wel. Er is op dit moment nog geen kant en klare oplossing voor het bepalen van de locatie van een persoon op een podium. Er wordt dus een nieuwe technologie gebruikt binnen de dance industrie.

Professioneel werken: Zelfstandig werken (H5)

Tijdens dit project is een grote mate van zelfstandigheid en creativiteit nodig. Het is de bedoeling dat ik dit systeem ontwerp en ontwikkel is samenwerking met Sendrato. De keuzes die ik tijdens het project moet maken moeten goed onderbouwt zijn zodat dit project een succes kan worden.

Analyseren van het probleemdomein (A1)

Zoals al eerder genoemd moet er in de eerste weken van het project een duidelijk beeld worden gevormd van het oplossen probleem. Dit wordt gedaan aan de hand van de verschillende onderzoeken die gedaan moeten worden.

Bijlage B

Plan van aanpak

HAUTE TECHNIQUE

Plan van Aanpak

Auteur	: Jan de Boer
Bedrijf	: HAUTE TECHNIQUE
Opdrachtgever	: Sander ter Braak
Datum	: 4 september 2015
Versie	: 1.0

Versiebeheer

Versie	Datum	Opmerking
0.1	02-09-2015	Eerste versie
1.0	05-11-2015	Opmaak consistent gemaakt

Inhoudsopgave

1	Achtergrond.....	5
1.1	Bedrijfsomschrijving	5
1.2	Probleemstelling.....	5
1.3	Doelstelling.....	5
2	Project.....	6
2.1	De opdracht.....	6
2.2	Resultaat.....	6
2.3	Grenzen	6
2.4	Randvoorwaarden	7
2.5	Project organisatie.....	7
2.6	Op te leveren producten	7
2.7	Business case	7
3	Aanpak	8
3.1	Methodiek	8
3.2	Scrum.....	9
3.3	Technologieën	10
3.4	Programmeertaal	10
4	Planning	11
4.1	Toelichting	11
5	Risicoanalyse	13
	Bibliografie	14

1 Achtergrond

1.1 Bedrijfsomschrijving

De afstudeeropdracht wordt uitgevoerd bij het bedrijf Haute Technique. Haute Techniek is een jong bedrijf dat zich bezighoudt met de zogenoemde nieuwe media en werkt op projectbasis. De focus van Haute Technique ligt in het creëren van unieke en passende oplossingen voor zijn klanten. Zo hebben ze ervaring met het maken van games, interactieve installaties, aanraakgevoelige tafels en verschillende andere zaken. Haute Technique bestaat uit een vast team van drie personen. Hiernaast worden afhankelijk per project verschillende freelancers ingehuurd.

Een van de projecten die Haute Technique heeft lopen is een project samen met Armin van Buuren. In dit project helpt Haute Technique Armin van Buuren om zijn shows interactiever te maken. Hierbij maakt Haute Technique gebruik van de Myo [1]. De Myo is een bluetooth armband die de positie van de armen kan volgen en deze doorstuurt via bluetooth. Haute Technique heeft een software systeem ontwikkeld die de Myo kan uitlezen en aan de hand van deze data lampen en of videobeelden kan aansturen. Deze techniek wordt op dit moment toegepast bij de sets van Armin van Buuren en wordt goed ontvangen door het publiek.

Samen met Sendrato [2] wil Haute Technique dit systeem naar het volgende level tillen door het uit te breiden met locatie gevoeligheid. Sendrato heeft veel expertise op het gebied van locatiebepaling van grote groepen mensen. Zo leveren ze het toegangssysteem van een aantal grote festivals. Met dit toegangssysteem wordt het mogelijk om de locatie van iedere bezoeker te volgen.

HAUTE TECHNIQUE

SENDRATO

1.2 Probleemstelling

Het door Haute Technique ontworpen systeem om de shows van Armin van Buuren interactiever te maken is op dit moment alleen nog maar in te zetten als de locatie van Armin tijdens de show niet veranderd. Dit omdat een aantal van de visuele effecten die gebruikt worden afhankelijk zijn van de positie van Armin van Buuren. Daarom is het gewenst om het systeem zodanig uit te breiden zodat er kan worden gereageerd op de positie van Armin van Buuren. Dit nieuwe systeem kan in de toekomst ook zorgen voor nieuwe elementen in de show waardoor de shows steeds 'interactiever' worden.

1.3 Doelstelling

De doelstelling van dit project is om een software applicatie te ontwikkelen die gebruik makend van de hardware van Sendrato lampen en of visuals laat reageren op de positieveranderingen van een persoon of personen. Hierbij zijn tests uitgevoerd om de werking van de hardware te valideren.

2 Project

In dit hoofdstuk worden de verschillende aspecten van de opdracht behandeld. Hierbij wordt de nieuwe opdracht verder toegelicht.

2.1 De opdracht

Aan het begin van het afstuderen was de opdracht als volgt geformuleerd: het ontwikkelen van een softwaresysteem dat gebruikmakend van de sensoren die Sendrato levert de locatie van een persoon op een podium kon berekenen. Deze locatiegegevens zouden vervolgens via een gebruiksvriendelijke interface worden gedeeld met andere applicaties. Ook zou er een applicatie worden ontwikkeld die gebruikmakend van de interface die de ontwikkelde applicatie aanbood een volglicht kon aansturen waardoor dit volglicht een persoon automatisch kon volgen. Zie het afstudeerplan voor verdere informatie over de originele afstudeeropdracht.

Echter bij aanvang van het project bleek dat een groot deel van deze opdracht al uitgevoerd was. Haute Technique en Sendrato hadden samen een testdag gehad en deze was veelbelovend verlopen. Ze kregen het voor elkaar om een volglicht een persoon te laten volgen. Hierbij droeg de persoon een tracking device bij zich. Een groot deel van de afstudeeropdracht was toen al voltooid. Haute Technique had namelijk een prototype applicatie geschreven die de sensoren kon uitlezen en vervolgens een volglamp kon aansturen. De moeilijkheidsgraad van de initiële afstudeeropdracht is aan het begin verkeerd ingeschat. Dit kwam mede doordat Sendrato zijn werk beter deed dan verwacht. In eerste instantie was het plan dat de te schrijven applicatie de sensordata nog moest omzetten naar bruikbare waarden dit was echter niet nodig, aangezien Sendrato dit al deed. Het systeem van Sendrato leverde dus al een interface waarmee de X, Y en Z waarden van een persoon konden worden ontvangen. Het was vervolgens een kleine klus om deze data uit te lezen en daarmee een volglamp aan te sturen. Een andere reden dat het initiële project moeilijker werd ingeschat dan het was kwam doordat de afstudeer zeer weinig ervaring en kennis heeft over de technieken die voor het project werden gebruikt.

De prototype applicatie die ontwikkeld was voor de testdag is echter verre van af. Hier moet nog tijd en onderzoek in worden gestoken om de applicatie bruikbaar te maken. Dit is dus ook waar het project zich op gaat focussen. Hierbij komen hele nieuwe problemen naar boven. Zoals: hoe worden de volglampen aan het systeem toegevoegd en gekalibreerd? Hoe kunnen er meerde personen gevolgd worden? Hoe reageren de volglampen op locatie veranderingen? Wat wordt er gedaan met de visuals, hoe worden deze locatiegevoelig gemaakt? Wat wordt de rol van de applicatie binnen de shows van bijvoorbeeld Armin van Buuren? Hiernaast zullen tijdens de loop van het project ook een hoop nieuwe vragen ontstaan. Naast het antwoord vinden op deze vragen moet ook het systeem dat Sendrato levert getest worden.

2.2 Resultaat

Haute Technique beschikt aan het einde van dit project over een software systeem die het mogelijk maakt om lampen en of visuals te laten reageren op de positieveranderingen van een persoon of personen. Met dit systeem is Haute Technique in staat om nieuwe elementen aan de shows van bijvoorbeeld Armin van Buuren toe te voegen waardoor deze nog interactiever worden.

2.3 Grenzen

Tijdens dit project worden niet de visuals of lampschema's bedacht en gemaakt die gebruikt kunnen worden voor de shows. Er is echter wel kennis nodig over hoe zulke schema's werken zodat de applicatie op de juiste manier kan worden ontwikkeld.

2.4 Randvoorwaarden

Voor dit project zijn een aantal randvoorwaarden van toepassing. Als er niet aan deze randvoorwaarden kan worden voldaan kan het project niet succesvol worden afgerond.

- Haute Technique is zelf niet instaat om de hardware te ontwikkelen die het bepalen van een locatie van een persoon mogelijk maakt. Hiervoor zijn ze afhankelijk van Sendrato. Sendrato moet voor dit project de hardware leveren.
- Het moet mogelijk zijn om de te schrijven applicatie te testen met fysieke volglampen.

2.5 Project organisatie

Dit project dient als afstudeeropdracht voor de afstudeerder Jan de Boer. Hierbij zijn vanuit de opleiding twee docenten aangesteld namelijk: A. Andrioli en J. van Peski; hier heeft de heer Peski de rol van tweede beoordelaar en de heer Andrioli die van begeleider. Vanuit Haute Techniek is Sander ter Baak de opdrachtgever en tevens de begeleider van de afstudeerder. Tijdens de loop van de afstudeerperiode zijn een aantal contactmomenten met de opleiding ingepland om de voortgang van het afstudeerproject door te nemen. Kijkend naar de risico's uit hoofdstuk 5 is het van belang dat er met de begeleider vanuit Haute Technique vaste momenten worden ingepland om de voortgang van het project te bespreken.

2.6 Op te leveren producten

Tijdens en aan het einde van dit afstudeerproject worden verschillende producten aan de opdrachtgever geleverd. Deze op te leveren producten zijn tot stand gekomen in samenspraak met de opdrachtgever en met de afstudeerbegeleider.

- Een onderzoeksrapport met daarin de verschillende onderzoeken die tijdens het project zijn uitgevoerd. Zoals bijvoorbeeld de analyse naar het kalibreren van de volglampen.
- Het software systeem met de daarbij horende source code van de software.
- Documentatie die het software systeem beschrijft, hieronder vallen de volgende documenten: requirements document, ontwerpdocument met daarin de UML diagrammen die het systeem beschrijven, een testrapport en een handleiding van het systeem.
- Een testrapport met betrekking op het systeem dat Sendrato levert.

2.7 Business case

Haute Technique beschikt aan het einde van dit project over een systeem waarmee ze lampen en of visuals kunnen laten reageren op de positieveranderingen van een persoon. Zulke systemen bestaan al in beperkte mate maar worden nog niet ingezet op de manier hoe Haute Techniek dat wil gaan doen. Haute Technique wil dit systeem gaan inzetten in de dance industrie.

Met het gebruik van dit nieuwe systeem kunnen de shows van bijvoorbeeld Armin van Buuren nog interactiever worden gemaakt. Als het systeem op de juiste manier wordt ingezet is dit een verrijking voor de show en zijn mensen bereid meer te betalen. Ook zou Haute Technique dit systeem kunnen verkopen aan andere partijen.

3 Aanpak

In dit hoofdstuk wordt de aanpak van het afstudeer project behandeld. Hierin wordt de keuzen voor de gekozen methodiek verantwoord en worden de gebruikte standaarden verder toegelicht.

3.1 Methodiek

Het kiezen van de juiste methodiek voor dit project is van belang voor het (op de juiste manier) behalen van het einddoel van de afstudeeropdracht. Er zijn verschillende methodieken die gebruikt kunnen worden bijvoorbeeld: RUP, XP of Scrum. Elke methodiek is in meerdere of in mindere mate geschikt voor dit project. Het is daarom belangrijk om eerst de kenmerken van dit project te beschrijven. Hier volgen de kenmerken van het project:

- Tijdens dit project moeten er verschillende onderzoeken (analyses) worden gedaan.
- Tijdens het project wordt er software ontwikkeld.
- Het precieze einddoel van de te schrijven software staat nog niet vast. Hierdoor kunnen in de loop van het project de eisen veranderen.
- Tijdens dit project moet er documentatie worden opgeleverd.
- Er is sprake van een derde partij die de hardware levert.

Vanuit deze kenmerken zijn een aantal eisen opgesteld. Deze eisen zijn terug te vinden in Tabel 1. Per methodiek wordt er gekeken hoe deze zich aan de gestelde eis houdt. Vervolgens kan er een gegronde keuze worden gemaakt over een passende methodiek voor dit project. De legenda bevindt van Tabel 1 bevindt zich in Tabel 2.

Tabel 1 Vergelijkingstabel ontwikkelmethodes

Nr.	Eis	Waterval	RUP	openUp	Scrum	XP
E1	Geschikt voor software ontwikkeling	+	+	+	+	+
E2	Eisen kunnen worden aangepast tijdens het project	-	~	~	+	+
E3	Werkt met iteraties	-	+	+	+	+
E4	Kan (goed) overweg met een onduidelijk einddoel	-	~	~	+	+
E5	Ervaring van de afstudeerder met de methodiek	~	+	~	+	-
E6	Biedt ruimte voor het uitvoeren van onderzoeken	+	+	+	+	~
E7	Heeft plek voor documentatie	+	+	+	~	~
E8	Kan solo worden uitgevoerd	+	~	~	~	-

** In tegenstelling tot RUP en openUp biedt SRUM standaard geen plek aan voor onderzoek. Dit is echter prima binnen Scrum te passen zie de bronnen [3] [4]*

Tabel 2 Legenda software methodieken

Legenda van Tabel 1	
+	De methodiek voldoet aan deze eis
~	De methodiek voldoet deels aan deze eis
-	De methodiek voldoet niet (of nauwelijks) aan deze eis

Kijkend naar Tabel 1 valt de keuze op *Scrum* [5]. De waterval methode valt af omdat deze methode niet overweg kan met veranderende eisen. Extreme programming (XP) valt af omdat deze niet solo is uit te voeren. XP is namelijk gebaseerd op pair programming. RUP en openUp zouden goede kandidaten zijn alleen kunnen ze minder goed omgaan met een onduidelijk einddoel. Scrum kan dit beter aangezien Scrum per sprint de verschillende fasen van RUP en openUp doorloopt. Scrum biedt echter niet zoals RUP of openUp een standaard set van documenten aan, deze moeten binnen Scrum zelf naar gelang het project bepaald worden.

3.2 Scrum

Scrum is een methodiek die valt onder de AGILE vlag. Scrum maakt gebruik van sprints van één tot vier weken. In een sprint worden de items van de sprint backlog afgewerkt. De sprint backlog wordt aan het begin van de sprint gevuld met items uit de product backlog tijdens de zogenoemde sprint planning. Het resultaat van elke systeem is een werkend (of deels werkend) product. Zie Figuur 1.



Figuur 1 Scrum proces

Scrum is in eerste instantie bedoeld voor een team van meerdere mensen, er is daarom een kleine vertaalslag nodig om Scrum bruikbaar te maken voor een project met een persoon.

- De dagelijkse Scrum meetings hebben weinig zin met een teamlid. Deze kunnen dus niet in deze vorm worden gebruikt. Normaal wordt de daily scrum alleen door het ontwikkelteam gehouden. Maar aangezien dit project alleen wordt uitgevoerd wordt de projectbegeleider bij deze meeting betrokken. In dit geval is dit dezelfde persoon als de opdrachtgever, het is daarom belangrijk dat de opdrachtgever puur de rol van begeleider speelt tijdens deze meetings.
- Normaal zijn de product owner, scrummaster en de developer verschillende taken die worden uitgevoerd door verschillende personen. Dit is echter niet mogelijk binnen dit project. Daarom is er voor gekozen om deze rollen samen te nemen. Hierbij ontstaat het risico dat de scrum methode niet correct wordt nageleefd. Dit is echter op te lossen door aan het einde van elke

HAUTE TECHNIQUE

sprint te controleren of scrum nog wel correct wordt toegepast. De rollen die de uitvoerende heeft zijn dan als volgt: het beheren van de product backlog en sprint backlog, het controleren of scrum nog wel correct wordt toegepast, een sprint plannen en deze uitvoeren, aan het einde van een sprint een sprint retrospective inplannen.

- In samenspraak met de opdrachtgever worden de items in de product backlog en de sprintbacklog geplaatst. Dit omdat de opdrachtgever de prioriteit van de verschillende werkzaamheden kan bepalen.
- Bij een sprint review zijn de uitvoerende van dit project en de opdrachtgever aanwezig.

3.3 Technologieën

Aan het begin van dit project zijn er al een aantal standaarden bekend. Deze standaarden zullen in de loop van het project worden gebruikt.



GIT

Binnen dit project wordt git [6] gebruikt als versie control systeem. Git is geschreven door Linus Torvalds, de ontwikkelaar van Linux en wordt gebruikt door de Linux Kernel.



openFrameworks

Voor de prototype applicatie heeft Haute Technique openFrameworks [7] gebruikt. openFrameworks is een open source C++ framework dat helpt bij het creëren van grafische applicaties. Haute Techniek heeft al veel ervaring met dit framework.



Microsoft office

De documenten en grafieken die binnen dit project worden gecreëerd zijn gemaakt met applicaties uit de Microsoft office familie.

3.4 Programmeertaal

Aangezien de prototype applicatie gebruik maakt van *openFrameworks* wordt ook de nieuwe applicatie geschreven in C / C++. Aangezien tal van verbeteringen van C++ ten op zichte van C en snelheid / geheugengebruik verwaarloosbaar zijn is er gekozen om de applicatie in C++ te ontwikkelen.

4 Planning

Het opstellen van een planning aan het begin van het project is belangrijk voor het goed verlopen van het project. De gekozen methode, in dit geval Scrum heeft invloed hoe de planning er uit ziet. Scrum werkt in het algemeen alleen maar met sprintplanningen. Deze worden aan het begin van een sprint bepaald en daarna uitgevoerd. Kennis over hoe de volgende sprint of sprints er uit gaan zien is er dan vaak nog niet. Toch is het een praktisch om aan het begin van het project een globale planning te maken. Deze planning is tot stand gekomen rekening houdend met de risico's in hoofdstuk 0. Zie Figuur 2 voor de globale planning.

Er is gekozen om met een sprintlengte van twee weken te werken. Dit omdat er dan snel kan worden bijgestuurd mocht het project de verkeerde kant op gaan. Ook zijn er nog veel onduidelijkheden; door gebruik te maken van korte sprints kan er snel geïtereerd worden.

4.1 Toelichting

Bij het inrichten van dit project zijn bepaalde technieken van RUP overgenomen. De eerste twee sprints lijken namelijk op de *inception* en *elaboration* fasen binnen RUP. Binnen deze sprints wordt de opdracht verduidelijkt en worden de eisen van de hardware en de software bepaald. Het stuk analyse dat wordt uitgevoerd in sprint een en twee bestaat uit een aantal aparte analyses. Namelijk:

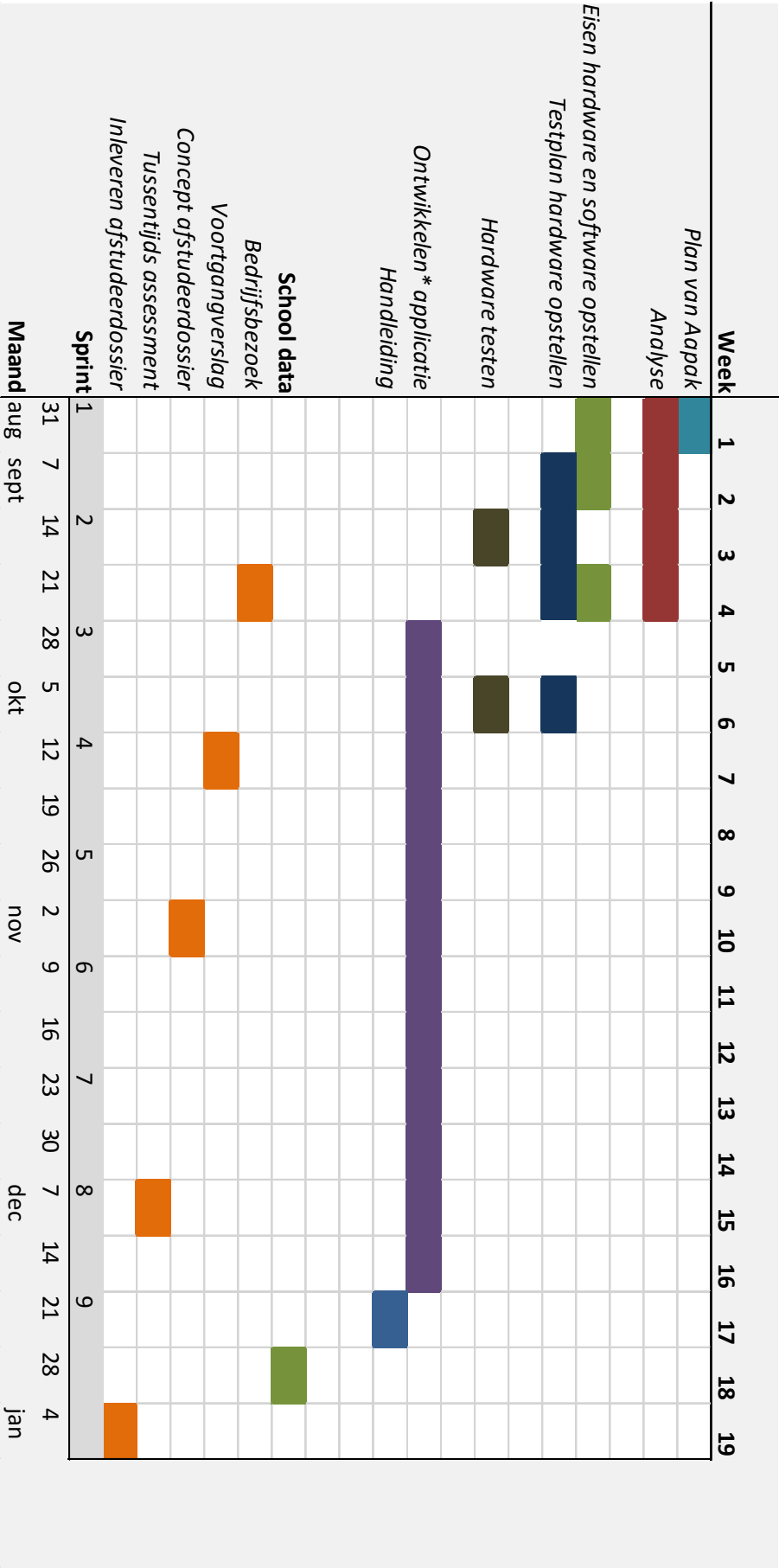
- Analyse naar de werking van het DMX protocol. Dit protocol wordt gebruikt om volglampen en andere apparaten die tijdens een show worden gebruikt aan te sturen.
- Een analyse naar de werking van volglampen. Deze kennis is nodig om deze lampen correct te kunnen aansturen.
- Een analyse naar de werking van de hardware die Sendrato levert. Met deze kennis kan de hardware beter getest worden.
- Een onderzoek dat kijkt hoe de volglampen het beste kunnen worden toegevoegd aan de applicatie en hoe deze vervolgens moeten worden gekalibreerd. Hierbij kost het onderzoeken naar een goede kalibratie methode waarschijnlijk de meeste tijd.
- Een onderzoek naar hoe de visuals worden aangestuurd en wat er moet gebeuren om visuals locatiegevoelig te maken.

Het is op dit moment nog moeilijk in te schatten hoelang het onderzoek naar het kalibreren van de volglampen duurt. Er moet dus aan het einde van sprint twee worden gekeken of er nog meer onderzoek nodig is.

In sprint een en twee wordt er ook een testrapport opgesteld voor de hardware. Dit testrapport wordt gebruikt om de hardware die wordt geleverd door Sendrato uitgebreid te testen. Nadat de hardware is getest worden de resultaten van het testen toegevoegd aan het testrapport. Het feit dat de hardware moet worden getest wil niet zeggen dat deze het helemaal niet doet. Er is al een succesvolle testdag geweest waarbij een volglamp een persoon kon volgen, maar voordat dit systeem kan worden ingezet in grote shows moet het eerst uitgebreid getest worden. In de globale planning is er aangenomen dat de hardware na twee testrondes voldoet aan de gestelde eisen. Mocht dit niet zo zijn dan worden er meerdere testmomenten ingepland.

Nadat sprint één en twee zijn afgerond is het voor een groot deel duidelijk wat de te schrijven applicatie precies moet gaan doen. De sprints hierna zijn dus ook gereserveerd voor het ontwikkelen van de applicatie. Hierbij betekent ontwikkelen: analyseren, ontwerpen, implementeren en testen. Het is belangrijk om aan het einde van sprint twee de nog te komen sprints beter in te plannen.

In week 18 zitten een aantal vrije dagen namelijk: eerste en tweede kerstdag en oud en nieuwjaarsdag. De datums die vanuit school zijn opgesteld zijn ook weergegeven in de planning.



Figuur 2 Globale planning

5 Risicoanalyse

Het is belangrijk om aan het begin van een project de verschillende risico's in kaart te brengen. De maatregelen van de gevonden risico's kunnen vervolgens mee worden genomen in de opbouw van de planning en de verdere verloop van het project.

Elk risico heeft een kans in procenten en een impact een waarde tussen één en tien. Hieruit volgt de prioriteit, dit is de kans x de impact. Het risico met de hoogste prioriteit moet als eerste getackeld worden. In Tabel 3 volgt de risicoanalyse van dit project.

Tabel 3 Risicoanalyse

Nr.	Beschrijving	K	I	P	Maatregel
R1	Niet alle aspecten van de opdracht zijn nog even duidelijk. Zo is er nog niet duidelijk wat de applicatie allemaal moet doen.	100	8	800	Door de juiste ontwikkelmethodiek te kiezen kan het doel stap voor stap achterhaald worden. De ontwikkel methodiek moet flexibel genoeg zijn om in korte iteraties een werkend (deel) systeem op te leveren. Zodat de opdrachtgever snel ziet in welke richting het project gaat en kan bijsturen zo nodig. Ook moet de planning rekening houden met dit feit. Door vroeg in het project de eisen duidelijker te krijgen.
R2	De opdrachtgever heeft een drukke agenda en is mogelijk veel op tour, waardoor er geen vergaderingen kunnen plaats vinden.	100	6	600	Door van te voren vaste dagen af te spreken om te vergaderen kan de opdrachtgever hier rekening mee houden.
R3	De hardware die wordt geleverd door Sendrato voldoet niet aan de eisen van de opdrachtgever.	60	8	480	Door vroeg in het project de verkregen hardware te testen is er nog tijd beschikbaar voor Sendrato om de hardware aan te passen.
R4	Bij aanvang van het project is de kennis bij de afstudeerder over locatiebepaling en de sensoren die geleverd worden door Sendrato beperkt.	100	4	400	Er moet vroeg in het project een onderzoek worden gedaan hoe de sensoren van Sendrato werken.
R5	Het project komt niet af binnen de gestelde deadline.	40	10	400	Door aan het begin van het project een planning te maken kan dit deels worden opgelost. Ook helpt het om de eisen van het systeem te prioriteren zodat de belangrijkste eisen als eerste worden gedaan.
R6	Sendrato krijgt het niet voor elkaar om een werkend prototype van de hardware voor het project op tijd te leveren.	10	9	90	Door dit op tijd binnen het project te signaleren kunnen er alternatieven worden gezocht.

Bibliografie

- [1] „Myo,” [Online]. Available: www.myo.com. [Geopend 3 September 2015].
- [2] „Sendrato,” [Online]. Available: <http://sendrato.com/>. [Geopend 2 September 2015].
- [3] Vin, „Scrum Isn’t Just for Product Development Projects,” [Online]. Available: <http://brainslink.com/2012/04/scrum-isnt-just-for-product-development-projects>. [Geopend 2 September 2015].
- [4] jmort253, „How can a team working on research based items work in an Agile project?,” [Online]. Available: <http://pm.stackexchange.com/questions/1358/how-can-a-team-working-on-research-based-items-work-in-an-agile-project>. [Geopend 2 September 2015].
- [5] „Scrum guide,” [Online]. Available: <https://www.scrumalliance.org/why-scrum/scrum-guide>. [Geopend 2 September 2015].
- [6] „Git,” [Online]. Available: <https://git-scm.com/>. [Geopend 3 September 2015].
- [7] „www.openframeworks.cc,” [Online]. Available: <http://www.openframeworks.cc>. [Geopend 2 September 2015].
- [8] „Scrum,” [Online]. Available: <https://www.scrumalliance.org>. [Geopend 2 September 2015].

Bijlage C

Requirements Document

HAUTE TECHNIQUE

Requirements

Auteur	: Jan de Boer
Bedrijf	: HAUTE TECHNIQUE
Opdrachtgever	: Sander ter Braak
Datum	: 4 september 2015
Versie	: 1.0

Versiebeheer

Versie	Datum	Opmerking
0.1	07-09-2015	Eerste versie
0.2	09-09-2015	Na vergadering met de opdrachtgever extra use case: configuratie testen toegevoegd.
0.3	10-10-2015	Opsturen eerste concept naar afstudeerbegeleider
1.0	05-01-2015	Verschillende wijzigingen in de opmaak doorgevoerd.

Inhoudsopgave

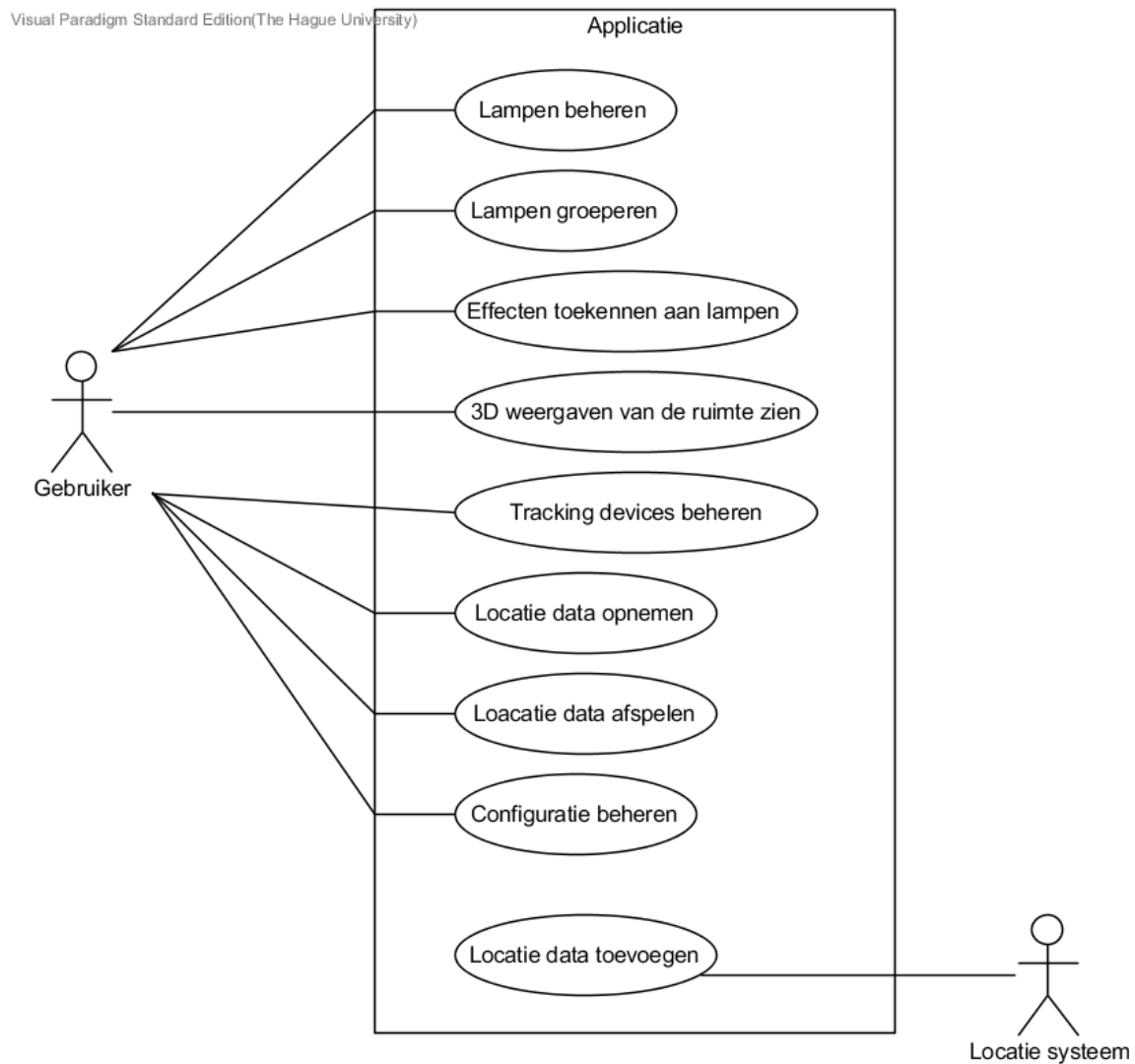
1	Introductie.....	5
2	Use case	6
2.1	Scenario's	7
3	Eisen.....	11
3.1	Prioritering.....	11
3.2	Software eisen.....	11
3.2.1	Functionele eisen	11
3.2.2	Niet functionele eisen	12
3.3	Hardware eisen	12
3.3.1	Functionele eisen	12
3.3.2	Niet functionele eisen	13
	Bibliografie	14

1 Introductie

In dit document worden de hard- en software eisen behandeld die op dit project van toepassing zijn. De software eisen zijn afgeleid uit een use case met de daarbij horende scenario's. De hardware eisen zijn tot stand gekomen in samenspraak met de opdrachtgever. Deze dienen als input voor het hardware test rapport.

2 Use case

In dit hoofdstuk bevindt zich het use case diagram met de daarbij horende scenario's. Zie Figuur 1 voor het use case diagram. Een aantal van de use cases is samengenomen tot de algemene term 'beheren,' aangezien het use case diagram anders te groot zou worden wat de leesbaarheid niet ten goede zou komen. De scenario's uit Hoofdstuk 2.1 werken echter wel de afzonderlijke situaties verder uit.



Figuur 1 Use Case diagram applicatie

2.1 Scenario's

Scenario 1.1 Lampen beheren Toevoegen	
Actoren	Gebruiker
Pre conditie	-
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker drukt op een knop om een lamp toe te voegen. 2. De gebruiker krijgt een scherm te zien waarin die de volgende gegevens van de lamp moet invullen: positie van de lamp in de ruimte, type lamp. 3. Het systeem voegt de lamp toe.
Alternatief	<ol style="list-style-type: none"> 1. De lamp die wordt toegevoegd door de gebruiker is niet bekend bij het systeem. Het systeem maakt dit kenbaar door een melding op het scherm te tonen. 2. De gebruiker kiest voor de optie automatisch toevoegen. Het systeem bepaalt vervolgens automatisch de positie van de lamp.
Post conditie	Er is een lamp aan het systeem toegevoegd.

Scenario 1.2 Lampen beheren Verwijderen	
Actoren	Gebruiker
Pre conditie	-
Beschrijving	De gebruiker kan door middel van een knop lampen uit het systeem verwijderen.
Alternatief	-
Post conditie	Er is een lamp verwijderd uit het systeem.

Scenario 1.3 Lampen beheren Groeperen	
Actoren	Gebruiker
Pre conditie	Er zijn minstens twee lampen toegevoegd aan het systeem.
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker klikt op een knop om het groeperen te beginnen. 2. De gebruiker klikt op de verschillende lampen die moeten worden toegevoegd aan de desbetreffende groep. 3. De gebruiker klikt op een knop om het groeperen te beëindigen.
Alternatief	De gebruiker heeft maar een lamp toegevoegd aan de groep. Het systeem maakt dit kenbaar door een melding en verwijdert de groep.
Post conditie	Het systeem heeft een groep gecreëerd met daarin de verschillende door de gebruiker geselecteerde lampen.

Scenario 3.1 Effecten toekennen aan lampen	
Actoren	Gebruiker
Pre conditie	Er is minstens een lamp aan en een tracking device aan het systeem toegevoegd.
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker klikt op een lamp waaraan hij een effect wil toekennen. 2. De gebruiker krijgt een lijstje te zien met effecten die aan de lamp kunnen worden toegevoegd. 3. De gebruiker kiest een effect uit de lijst.
Alternatief	De lamp heeft al een effect, dit effect zal worden verwijderd en het nieuwe toegevoegde effect wordt aan de lamp toegekend.
Post conditie	De lamp heeft een effect toegekend gekregen.

HAUTE TECHNIQUE

Scenario 4.1 3D weergaven van de ruimte zien	
Actoren	Gebruiker
Pre conditie	-
Beschrijving	De gebruiker kan binnen de applicatie navigeren naar een scherm waar de ruimte in 3D wordt weergegeven.
Alternatief	-
Post conditie	De gebruiker ziet een 3D weergave van de omgeving met daarin de verschillende lampen, tracking devices en antennes. De lampen geven weer waar ze naar toe wijzen en de tracking devices geven hun positie weer.

Scenario 5.1 Tracking devices beheren Toevoegen	
Actoren	Gebruiker
Pre conditie	-
Beschrijving	<ol style="list-style-type: none">1. De gebruiker drukt op een knop binnen de applicatie om een tracking device toe te voegen.2. De gebruiker krijgt een lijst te zien met de verschillende tracking devices3. De gebruiker selecteert een tracking device uit de lijst.
Alternatief	Er zijn geen tracking devices gevonden. Het systeem geeft de gebruiker een melding.
Post conditie	Er is een tracking device toegevoegd aan de applicatie.

Scenario 5.2 Tracking devices beheren Verwijderen	
Actoren	Gebruiker
Pre conditie	Er is minstens een tracking device aan het systeem toegevoegd.
Beschrijving	<ol style="list-style-type: none">1. De gebruiker drukt op een knop om een tracking device te verwijderen.2. De gebruiker ziet een lijst met de toegevoegde tracking devices.3. De gebruiker kiest een tracking device om te verwijderen.
Alternatief	De gebruiker bedenkt zich en verwijderd geen tracking device.
Post conditie	Er is een tracking device verwijderd.

Scenario 6.1 Locatie data opnemen	
Actoren	Gebruiker
Pre conditie	Er is een tracking device aan het systeem toegevoegd.
Beschrijving	<ol style="list-style-type: none">1. De gebruiker drukt op een knop om het opnemen van de locatie data te starten.2. De gebruiker kiest een tracking device uit de lijst van bekende tracking devices.
Alternatief	Er is al een opname gestart, het systeem geeft een melding dat deze eerst moeten worden gestopt om een volgende te starten.
Post conditie	De locatie data van een tracking device wordt opgenomen.

Scenario 7.1 Locatie data afspelen	
Actoren	Gebruiker
Pre conditie	Er is een locatie opnamen geweest.
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker klikt op een knop binnen de applicatie om een tracking device te simuleren aan de hand van opgenomen locatie data. 2. De gebruiker krijgt een lijst te zien van beschikbare opnamen. 3. De gebruiker kiest een opname uit de lijst. 4. Het systeem simuleert een tracking device.
Alternatief	-
Post conditie	Er wordt een tracking device gesimuleerd met de opgenomen locatie data.

Scenario 8.1 Configuratie beheren Exporteren	
Actoren	Gebruiker
Pre conditie	-
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker drukt op een knop om de huidige configuratie op te slaan. 2. De gebruiker krijgt van het systeem de vraag waar de configuratie moet worden opgeslagen. 3. Het systeem exporteert alle data uit het systeem (bijvoorbeeld locatie van lampen, toegevoegde tracking devices ect.) naar een bestand.
Alternatief	Er zijn geen lampen of tracking devices aan het systeem toegevoegd. Het systeem geeft een melding dat er niks kan worden geëxporteerd.
Post conditie	Er is een configuratie bestand aangemaakt met daarin de huidige configuratie van de applicatie.

Scenario 8.2 Configuratie beheren Importeren	
Actoren	Gebruiker
Pre conditie	-
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker drukt op een knop binnen de applicatie om het importeren van een configuratie te starten. 2. Het systeem vraagt de gebruiker waar het bestand kan worden gevonden. 3. De gebruiker vertelt het systeem waar het bestand kan worden gevonden. 4. Het systeem laadt de nieuwe configuratie.
Alternatief	Het bestand dat de gebruiker selecteert is geen geldig configuratie bestand. Het systeem toont een melding aan de gebruiker.
Post conditie	De oude configuratie is verwijderd en de nieuwe configuratie uit het bestand is ingeladen in het systeem.

Scenario 8.3 Configuratie beheren Testen	
Actoren	Gebruiker
Pre conditie	Er is minstens een tracking device of lamp aan het systeem toegevoegd.
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker drukt op een knop om de huidige configuratie te testen. 2. Het systeem loopt een testcase door waarbij er feedback van de gebruiker wordt gevraagd om te controleren of de verschillende lampen en of tracking devices naar behoren werken en correct zijn ingesteld. 3. Alle tests zijn doorlopen en de gebruiker krijgt een melding dat het systeem correct werkt.
Alternatief	De gebruiker ontdekt een fout, de gebruiker stopt of pauzeert het testproces en verhelpt de fout. Vervolgens drukt de gebruiker op start of doorgaan.
Post conditie	De huidige configuratie werkt naar behoren en is daarbij correct ingesteld.

Scenario 9.1 Locatie data toevoegen	
Actoren	Locatie systeem
Pre conditie	De applicatie is opgestart door de gebruiker.
Beschrijving	Het locatie systeem stuurt via UDP pakketjes data over de antennes en de positie van de verschillende tracking devices door.
Alternatief	Het systeem ontvangt geen data van het locatie systeem. De gebruiker krijgt een melding te zien met daarin de reden.
Post conditie	De applicatie beschikt over de locatie van de verschillende tracking devices en de antennes.

3 Eisen

In dit hoofdstuk worden de functionele en de niet functionele eisen van zowel de software applicatie als de hardware behandeld. De hardware eisen worden gebruikt als input voor het hardware test rapport.

3.1 Prioritering

Zoals besproken in het plan van aanpak in hoofdstuk 5, risicoanalyse worden de eisen ingedeeld in verschillende prioriteiten. Dit wordt gedaan volgens de MoSCoW[1] methode. De MoSCoW methode definieert vier verschillende prioriteiten namelijk:

Must have's	M	Aan deze eisen moet worden voldaan anders is het product niet bruikbaar.
Should have's	S	Deze eisen zijn zeer gewenst maar als er aan deze eisen niet wordt voldaan is er nog wel een bruikbaar product.
Could have's	C	Deze eisen komen alleen aan bod als er genoeg tijd is om deze eisen te implementeren
Won't have's	W	Deze eisen zullen in dit project niet aan bod komen maar kunnen voor eventuele vervolg projecten interessant zijn.

3.2 Software eisen

Uit het use case diagram met de daarbij horende scenario's uit hoofdstuk 2 zijn een groot deel van de functionele eisen afgeleid. De niet functionele eisen en de rest van de functionele eisen zijn tot stand gekomen in gesprekken met de opdrachtgever.

3.2.1 Functionele eisen

Nr.	Beschrijving	Prioriteit
F1.1	Er moeten meerdere lampen aan het systeem kunnen toegevoegd.	Must
F1.2	Er moeten verschillende typen lampen aan het systeem worden toegevoegd.	Must
F1.3	Een lamp moet in staat zijn om zich volgens een effect te gedragen.	Must
F1.4	De positie van de lamp wordt semi automatisch bepaald (gekalibreerd).	Should
F1.5	De applicatie geeft weer waar de verschillende lamp zich bevinden in de ruimte.	Must
F1.6	De applicatie laat zien waar de verschillende tracking devices zich bevinden in de ruimte.	Must
F1.7	De applicatie laat zien waar de verschillende antennes zich bevinden in de ruimte.	Must
F1.8	Er kunnen tracking devices aan het systeem worden toegevoegd.	Must
F1.9	(effect) Lampen moeten in staat zijn om personen binnen een bepaald gebied te volgen.	Should
F1.10	Het systeem is in staat om videobeelden aan te sturen.	Could
F1.11	De applicatie moet de lampen zo aansturen dat de vertraging minimaal is.	Must
F1.12	Volglampen moeten kunnen worden gegroepeerd, om zo allemaal hetzelfde effect te kunnen doen.	Should
F1.14	De applicatie geeft de stand van de volglampen in de 3D omgeving weer.	Must
F1.15	De applicatie moet in staat zijn op de locatie data die binnenkomt op te nemen.	Should
F1.16	De applicatie moet in staat zijn om de opgenomen locatie data af te spelen in dezelfde volgorde als dat deze was ontvangen.	Should
F1.17	Locatie data kan worden gesimuleerd.	Should

HAUTE TECHNIQUE

F1.18	Er kunnen meerdere tracking devices tegelijk worden opgenomen en afgespeeld.	Could
F1.19	Er moet een mogelijkheid zijn binnen de applicatie om de configuratie (de verschillende lampen en of tracking devices) te testen of deze naar behoren werken / zijn ingesteld.	Must
F1.20	Het moet mogelijk zijn na het corrigeren van een gefaalde test (bijvoorbeeld een lamp had een afwijking) verder te gaan met testen waar de gebruiker gebleven was.	Should

3.2.2 Niet functionele eisen

Nr.	Beschrijving	Prioriteit
NF1.1	Het systeem stuurt de volglampen aan via het DMX protocol gebruik makend van ArtNet.	Must
NF1.2	Het systeem moet zo geschreven worden dat het relatief gemakkelijk is om in de toekomst nieuwe effecten toe te voegen.	Should
NF1.3	De applicatie draait onder zowel Apple OSX als onder Windows.	Must

3.3 Hardware eisen

De hardware eisen die in dit hoofdstuk zijn beschreven dienen als input voor het hardware testrapport. Deze hardware eisen zijn tot stand gekomen door gesprekken met de opdrachtgever te voeren. De eisen zijn opgedeeld in functionele en in niet functionele eisen.

3.3.1 Functionele eisen

Nr.	Beschrijving	Prioriteit
F2.1	Het locatie systeem moet binnen vier uur opgezet en klaar voor gebruik zijn.	Must
F2.2	De locatie data moet realtime worden doorgestuurd. Dit houdt in dat er genoeg samples per seconden worden doorgestuurd zodat de resolutie hoog genoeg wordt. Hoeveel samples er per seconden moeten zijn moet nog worden bepaald.	Must
F2.3	Het systeem moet in staat zijn om gebieden van verschillende groottes te voorzien van tracking. Hier wordt 10 x 25 x 5 als ondergrens en 100 x 75 x 20 als bovengrens gebruikt.	Must
F2.4	Het systeem moet in staat zijn om minimaal 5 tracking devices te kunnen volgen.	Should
F2.5	Het systeem moet de X, Y en Z waarden van een tracking device doorsturen.	Must
F2.6	De sensoren van het systeem mogen niet te verstoren zijn met materialen of omstandigheden die tijdens een show aanwezig (kunnen) zijn.	Must
F2.7	Het systeem moet een interface aanbieden waarmee informatie over het systeem kan worden ontvangen.	Must

3.3.2 Niet functionele eisen

Nr.	Beschrijving	Prioriteit
NF2.1	Het systeem moet klein genoeg zijn om mee te nemen in een flight case.	Must
NF2.2	Het systeem moet te bedienen zijn door een leek.	Won't
NF2.3	Het systeem moet minimaal 8 uur achtereenvolgens werken. Hier bij moet het systeem constant blijven voldoen aan de gestelde functionele en niet functionele eisen.	Must
NF2.4	Het systeem levert een nauwkeurigheid van 1 centimeter.	Must
NF2.5	Het systeem levert de correcte positie van een tracking device.	Must

Bibliografie

- [1] *Key words for use in RFCs to Indicate Requirement Levels*. (2015, September 3). Opgehaald van IETF: <http://www.ietf.org/rfc/rfc2119.txt>
- [2] *MoSCoW method*. (2015, September 3). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/MoSCoW_method

Bijlage D

Hardware testrapport

HAUTE TECHNIQUE

Hardware Testrapport

Auteur	: Jan de Boer
Bedrijf	: HAUTE TECHNIQUE
Opdrachtgever	: Sander ter Braak
Datum	: 4 september 2015
Versie	: 1.0

Versiebeheer

Versie	Datum	Opmerking
0.1	11-09-2015	Eerste versie.
0.2	30-10-2015	Naar aanleiding van de eerste testdag verschillende zaken aangepast en uitgebreid.
1.0	05-01-2016	Format aangepast.

Referentie documenten

Referentie	Document
D1	Requirements document
D2	Bevindingen Testdag 1

Inhoudsopgave

1	Introductie.....	5
2	Hardware.....	6
2.1	Protocol	7
2.1.1	Locatie updates	7
2.1.2	Antenne updates	8
3	Testmethode	9
3.1	Testcases	9
3.1.1	Functioneel	9
3.1.2	Niet functioneel	12
4	Resultaten	14
4.1	Testdag #1	14
	Bijlagen.....	16

1 Introductie

Binnen het project wordt er hardware van Sendrato gebruikt om het volgen van personen mogelijk te maken. Aan de hand van de positie van een persoon kunnen de lampen worden aangestuurd.

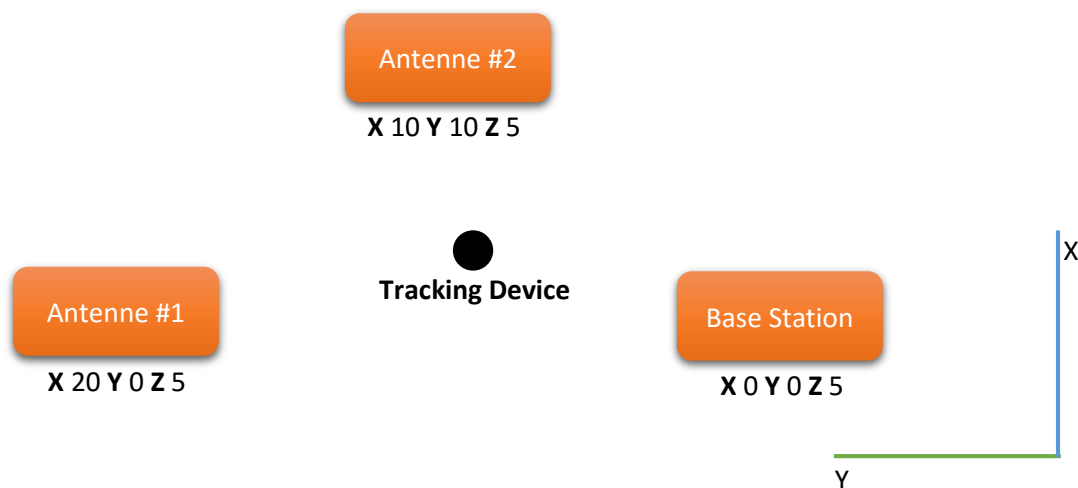
Dit document begint met een beschrijving van het systeem van Sendrato; hier wordt in het kort de werking van de hardware uitgelegd. Ook wordt de werking van het protocol om met het systeem van Sendrato te communiceren nader toegelicht. Hierna volgen een aantal testcases die tot stand zijn gekomen uit de functionele en niet functionele eisen uit het requirements document[D1]. De test resultaten van deze testcases worden in het hoofdstuk Resultaten beschreven. Dit gebeurt aan de hand van de verschillende testdagen die er zijn geweest.

Het doel van dit document is het beschrijven van de werking van de hardware van Sendrato en het testen hiervan. De hardware moet namelijk voldoen aan de eisen die de opdrachtgever stelt.

2 Hardware

De hardware die Sendrato levert bestaat uit een aantal verschillende componenten; deze componenten zullen stuk voor stuk behandeld worden.

Antennes	Het systeem bestaat over het algemeen over drie antennes. Hiervan is een antenne ook het base station. De tracking device(s) communiceren met de antennes om de locatie te kunnen bepalen.
Tracking device(s)	Een tracking device, ook wel tag, is het apparaat waarvan de positie wordt bepaald. De tag is voorzien van een lithium batterij die ongeveer 5 uur mee kan. De tag communiceert met de verschillende antennes in de opstelling om de locatie te kunnen bepalen.
Base station	Het basestation is verantwoordelijk voor het berekenen van de positie van de tag. De positie van de tag wordt via WiFi of Ethernet gebroadcast over het netwerk.



Figuur 1 Opstelling (Z as weggelaten)

In bovenstaand figuur is een opstelling te zien van het systeem. Hierbij is de Z as niet getekend. Het base station bevindt zich altijd op X 0 en Y 0. De eerste antenne bevindt zich altijd op Y 0.

Het systeem maakt gebruik van radio golven om onderling te kunnen communiceren. Afhankelijk van de opstelling communiceert het systeem op een bepaalde frequentie. Hoe groter het gebied is dat moet worden voorzien van tracking hoe lager de frequentie moet zijn. Er kan gekozen worden om meer antennes aan de opstelling toe te voegen, dit heeft tot gevolg dat er een groter gebied kan worden voorzien van tracking. Het tegeneffect van meer antennes is dat dit gevolgen heeft voor de frequentie van het doorsturen van locatie updates. Dit komt doordat het tracking device met meerdere antennes moet communiceren. Er zijn verschillende zaken die invloed hebben op het aantal locatie updates dat het systeem per tracking device kan doorsturen.

1. Hoe groter het gebied dat van tracking moet worden voorzien hoe lager de frequentie. Bij een lagere frequentie kan er minder snel data worden overgestuurd dan bij een hoge frequentie. Dit heeft tot gevolg dat het bepalen van een locatie langer duurt.
2. Hoe meer antennes zijn toegevoegd hoe langer het duurt voordat een tracking device een locatie kan berekenen. Dit komt doordat de tag met alle antennes moet communiceren om een nauwkeurige positie te bepalen.
3. Hoe meer tracking devices worden toegevoegd aan het systeem hoe lager de frequentie van de locatie updates per tracking device.

2.1 Protocol

Sendrato maakt gebruik van het UDP protocol om de data te verzenden naar andere systemen. Hierbij wordt elke pakketje over het netwerk gebroadcast naar poort 11235. Er zijn twee typen pakketjes te onderscheiden in het protocol namelijk locatie updates en antenne updates.

2.1.1 Locatie updates

Locatie updates worden gebruikt om de positie van een tracking device door te sturen. Het pakketje bestaat uit 24 bytes waarvan de eerste twee de header bevatten. Hoe vaak dit pakketje wordt doorgestuurd hangt af per opstelling. Hier volgt de layout van het pakketje:

<u>Bytes</u>	0 – 1	2	3 – 10	11	12 – 15	16 – 19	20 – 23	24
<u>Format</u>	uint8_t[]	uint8_t	uint64_t	uint8_t	int32_t	int32_t	int32_t	uint8_t
<u>Name</u>	Header	Type	Tijdstip	Tag ID	Positie X	Positie Y	Positie Z	Betrouwbaarheid

- Header** – Een header van 2 bytes deze bevat de waarden: 0x4F50
- Type** – Het type pakketje dit is voor een locatie update: 0x53
- Tijdstip** – Het tijdstip van de locatie meting.
- Tag ID** – Het unieke nummer van de tag. Dit is een waarde tussen de 0 en 255.
- Positie X** – De X positie van de tag in millimeters vanaf het base station
- Positie Y** – De Y positie van de tag in millimeters vanaf het base station
- Positie Z** – De Z positie van de tag in millimeters vanaf het base station

Notitie: De X, Y en Z assen van het systeem kunnen veranderen per opstelling.

HAUTE TECHNIQUE

2.1.2 Antenne updates

Antenne updates worden elke x seconden doorgestuurd. De pakketjes worden gebruikt om de positie van de verschillende antennes door te sturen. De structuur van het pakketje ziet er als volgt uit:

<u>Bytes</u>	0 – 1	2	3	4 -
<u>Format</u>	uint8_t[]	uint8_t	Uint8_t	Antenne[0-255]
<u>Name</u>	Header	Type	NumAnt	Antennes

Header – Een header van 2 bytes, deze bevat altijd de waarde: 0x4F50.

Type – Het type pakketje, dit is voor een antenne update: 0x54.

NumAnt – Totaal aantal antennes van het systeem.

Antennes – Informatie van de verschillende antennes volgens het tabel hieronder. Alle antennes volgen elkaar op in het zelfde pakketje. Hierbij wordt het NumAnt veld gebruikt om het totaal aantal antennes te bepalen.

Tabel 1 Antenne

<u>Bytes</u>	0	1 – 4	5 – 8	9 - 12
<u>Format</u>	uint8_t	int32_t	int32_t	int32_t
<u>Name</u>	ID	Positie X	Positie Y	Positie Z

ID – Het unieke nummer van de antenne. Dit is een waarde tussen de 0 en 255.

Positie X – De X positie van de antenne in millimeters vanaf het base station.

Positie Y – De Y positie van de antenne in millimeters vanaf het base station.

Positie Z – De Z positie van de antenne in millimeters vanaf het base station.

Notitie: De X, Y en Z assen van het systeem kunnen veranderen per opstelling.

3 Testmethode

De hardware van Sendrato wordt getest aan de hand van de eisen die zijn beschreven in het requirements document [D1]. Voor de verschillende functionele en niet functionele eisen in dit document zijn er testcases uitgewerkt. De testcases worden vervolgens stuk voor stuk getest tijdens de ingeplande testdag. De uitkomst van de testen worden in het hoofdstuk resultaten beschreven. Voor de tests die falen wordt de reden beschreven en een mogelijk oplossing. Zonodig wordt Sendrato in dit proces betrokken.

3.1 Testcases

Hieronder bevinden zich de verschillende testcases. Elke testcase heeft een uniek referentie nummer. Dit nummer wordt in het hoofdstuk resultaten gebruikt om naar een specifieke testcase te verwijzen.

Sommige testcases spreken over ‘de applicatie’, hiermee wordt de applicatie bedoelt die tijdens de loop van dit project wordt ontwikkeld. Het is van belang dat de informatie die de applicatie levert over bijvoorbeeld de positie van een tracking device betrouwbaar is. Om de correctheid van de applicatie te garanderen worden verschillende testmethodes gebruikt.

3.1.1 Functioneel

Testcase 1		Opzettijd
Beschrijving	Het locatie systeem moet minimaal binnen vier uur op te zetten zijn en klaar voor gebruik.	
Referentie specificatie	F2.1	
Voorwaarden	-	
Opmerking	Hoe groter de ruimte die moeten worden getracked hoe langer het opzetten van het systeem over het algemeen duurt.	
Teststappen		
1. Start een stopwatch. 2. Installeer het locatie systeem op de gewenste locatie. 3. Stop de stopwatch.		
Verwacht resultaat		
De tijd die op de stopwatch zichtbaar is is minder dan vier uur.		

HAUTE TECHNIQUE

Testcase 2 Locatie data moet realtime worden ontvangen	
Beschrijving	De locatie data die het systeem doorstuurt moet vaak genoeg door worden gestuurd zodat er geen stotter zichtbaar is. Anders verwoord het systeem moet meerdere keren per seconde de locatie data doorsturen zodat de resolutie hoog genoeg wordt.
Referentie specificatie	F2.2
Voorwaarden	<ol style="list-style-type: none">1. Het locatie systeem is opgezet en gekalibreerd2. Er is een koppeling gemaakt tussen het locatie systeem en de applicatie die de locatie data ontvangt.
Opmerking	Het hangt per opstelling af hoeveel updates het systeem per seconden doorstuurt.
Teststappen	
<ol style="list-style-type: none">1. Start de applicatie op.2. Koppel het volgeffect aan een lamp en aan het tracking device.3. Neem waar hoe de lamp reageert.	
Verwacht resultaat.	
Er worden genoeg samples per seconden doorgestuurd zodat er geen sortering zichtbaar is in het bewegen van de lamp.	

Testcase 3 Verschillende groottes tracken	
Beschrijving	Het moet mogelijk zijn om gebieden van verschillende groottes te voorzien van tracking.
Referentie specificatie	F2.3
Voorwaarden	Het systeem is opgezet.
Opmerking	Bij deze test is rekening gehouden met een minimale grootte van 10 x 25 x 5 meter en een maximalen grootte van 100 x 75 x 20 meter. Deze waarden vallen namelijk binnen de use case van het systeem.
Teststappen	
<ol style="list-style-type: none">1. Test of het systeem voldoet aan de verschillende beschreven testcases op de ondergrens.2. Test of het systeem voldoet aan de verschillende beschreven testcases op de bovengrens.	
Verwacht resultaat.	
Het systeem voldoet zowel aan de maximale als aan de minimale waarde van de beschreven test cases.	

Testcase 4 Ondersteuning voor meerdere tracking devices	
Beschrijving	Het moet mogelijk zijn om meerdere (minimaal 5) tracking devices aan het systeem toe te voegen.
Referentie specificatie	F2.4
Voorwaarden	<ol style="list-style-type: none">1. Het systeem is opgezet en gelinkt aan de test applicatie.2. Er is minimaal een tracking device aan het systeem toegevoegd.3. Er is een koppeling gemaakt tussen het locatie systeem en de applicatie die de locatie data ontvangt.
Opmerking	-
Teststappen	
<ol style="list-style-type: none">1. De verschillende tracking devices worden aan het systeem toegevoegd.2. Er wordt data ontvangen van de verschillende tracking devices.	
Verwacht resultaat	
De locatie van de toegevoegde tracking devices wordt ontvangen door de test applicatie en voldoet aan testcase 2, 3, N3 en N4	

Testcase 5 Doorsturen van de positie	
Beschrijving	Het systeem stuurt de X, Y en Z positie van de verschillende tracking devices door.
Referentie specificatie	F2.5
Voorwaarden	<ol style="list-style-type: none"> 1. Het systeem is opgezet en gelinkt aan de test applicatie. 2. Er is minimaal een tracking device aanwezig. 3. Er is een koppeling gemaakt tussen het locatie systeem en de applicatie die de locatie data ontvangt.
Opmerking	-
Teststappen	
<ol style="list-style-type: none"> 1. Start de applicaties op. 2. Voeg een tracking device toe aan de applicatie. 3. Druk op de knop om de X, Y en Z waarden van de toegevoegde tracking devices weer te geven. 	
Verwacht resultaat	
De X, Y en Z positie van het toegevoegde tracking device is zichtbaar binnen de applicatie.	

Testcase 6 Verstoring	
Beschrijving	Het systeem moet niet te verstoren zijn met materialen of omstandigheden die tijdens een show aanwezig (kunnen) zijn.
Referentie specificatie	F2.6
Voorwaarden	<ol style="list-style-type: none"> 1. Het systeem is opgezet en gelinkt aan de test applicatie. 2. Er is minimaal een tracking device aanwezig. 3. Er is een koppeling gemaakt tussen het locatie systeem en de applicatie die de locatie data ontvangt.
Opmerking	Een verstoring is zichtbaar doordat de doorgestuurde positie van de toegevoegde tracking devices niet meer voldoet aan testcases 2, N2 en N3.
Teststappen	
<ol style="list-style-type: none"> 1. Start de applicatie op. 2. Voeg een tracking device toe aan de applicatie. 3. Laat een persoon het systeem verstoren. 4. Neem waar of de waardes van de tracking devices voldoen aan de testcase 2, N3 en N4. 	
Verwacht resultaat	
De waarden die worden ontvangen van de verschillende tracking devices blijven, ongeacht of er geprobeerd wordt om ze te verstoren, voldoen aan de testcases 2, N3 en N4.	

Testcase 7 Interface naar 3 ^e programma's	
Beschrijving	Het systeem biedt een interface aan waardoor andere applicaties informatie over het locatiesysteem kunnen ontvangen. Zoals de positie van de verschillende tags of de posities van de verschillende antennes.
Referentie specificatie	F2.7
Voorwaarden	<ol style="list-style-type: none"> 1. Het systeem is opgezet en gekalibreerd. 2. Er is minimaal een tracking device beschikbaar. 3. Er is een tracking device aan het systeem toegevoegd. 4. Er is een koppeling gemaakt tussen het locatie systeem en de applicatie die de locatie data ontvangt.
Opmerking	
Teststappen	
1. Neem of de UDP broadcast pakketjes worden ontvangen.	
Verwacht resultaat.	
Het systeem stuurt de UDP pakketjes door.	

HAUTE TECHNIQUE

3.1.2 Niet functioneel

Testcase N1	Klein genoeg om mee te nemen
Beschrijving	Het systeem moet klein genoeg zijn om in een flight case te passen.
Referentie specificatie	NF2.1
Voorwaarden	Het locatie systeem moet beschikbaar zijn.
Opmerking	Hoe groter de ruimte is die moet worden voorzien van tracking hoe meer antennes er nodig zijn. Er is daarom rekening gehouden met de eis F2.3
Teststappen	
1. Test of het systeem past in de daarvoor beschreven grootte.	
Verwacht resultaat.	
Het systeem is meeneembaar.	

Testcase N2	Blijven werken
Beschrijving	Het systeem moet minimaal 8 uur achter elkaar kunnen draaien; hierbij moet het systeem aan het eind nog steeds voldoen aan de gestelde eisen.
Referentie specificatie	NF2.3
Voorwaarden	1. Het systeem is opgezet en gekalibreerd. 2. Er is minimaal een tracking device beschikbaar. Er is een tracking device aan het systeem toegevoegd.
Opmerking	De batterijen van tracking device mogen tijdens test vervangen worden mochten ze op zijn.
Teststappen	
1. Test of het voldoet aan de beschreven testcases. 2. Wacht 8 uur. 3. Test het systeem nogmaals aan de hand van de beschreven testcases	
Verwacht resultaat	
Het systeem is na 8 uur nog steeds volledig functioneel. Hierbij voldoet het aan de gestelde test cases.	

Testcase N3	Nauwkeurigheid
Beschrijving	Het systeem kan locatie veranderingen van minimaal 1 centimeter detecteren.
Referentie specificatie	F2.5
Voorwaarden	1. Het systeem is opgezet en gekalibreerd. 2. Er is minimaal een tracking device beschikbaar. 3. Er is een koppeling gemaakt tussen het locatie systeem en de applicatie die de locatie data ontvangt.
Opmerking	-
Teststappen	
1. Start de applicatie op. 2. Voeg een tracking device toe aan het systeem. 3. Druk op de knop om de X, Y en Z positie van het tracking device weer te geven. 4. Verplaats de tracking device 1 centimeter. 5. Is de X, Y of Z waarde van de tracking device veranderd.	
Verwacht resultaat	
De locatie die wordt doorgestuurd door het tracking device is veranderd nadat het tracking device 1 centimeter is verschoven.	

Testcase N4		Correctheid
Beschrijving	Het systeem stuurt de correcte positie van een tracking device	
Referentie specificatie	F2.5	
Voorwaarden	<ol style="list-style-type: none">1. Het systeem is opgezet en gekalibreerd.2. Er is minimaal een tracking device beschikbaar.	
Opmerking	-	
Teststappen		
<ol style="list-style-type: none">1. Start de test applicatie op.2. Voeg een tracking device toe.3. Ga op de positie van het base station staan.4. Loop 1 meter over de Y as,5. Loop 1 meter over de X as.6. Zorg ervoor dat het tracking deivce 1 meter boven het Z nul punt ligt.		
Verwacht resultaat		
Binnen de applicatie bevindt het tracking device zich op de coördinaten (1, 1, 1)		

Testcase N5		Draagbaarheid
Beschrijving	Het tracking device moet klein genoeg zijn zodat het makkelijk draagbaar is. Hier zou het bijvoorbeeld kunnen worden bevestigd aan een kleding stuk.	
Referentie specificatie		
Voorwaarden	Er is een tracking device aanwezig.	
Opmerking	-	
Teststappen		
1. Is het tracking device draagbar.		
Verwacht resultaat		
Het tracking device is draagbaar.		

4 Resultaten

In dit hoofdstuk worden de resultaten van de verschillende testdagen behandeld. Elke testdag worden verschillende testcases getest. Hieronder is de legenda te vinden van de tabellen die worden gebruikt per testdag om aan te geven of een test wel of niet is gelukt.

- Y** Ja, deze test is succesvol uitgevoerd
- N** Nee, deze test is niet succesvol uitgevoerd
- Deze test kon niet worden getest

4.1 Testdag #1

De eerste testdag vond plaats in het Amsterdam theater. De resultaten en bevindingen van deze testdag zijn te vinden in het document bevindingen van testdag 1[D2]. De volgende opstelling werd gebruikt tijdens de testdag:

- 1 tracking device (tag)
- 3 antennes (waarvan 1 ook het base station was)
- Grootte van het gebied dat voorzien was van tracking was ongeveer 40 x 25 x 5 meter

Tabel 2 Resultaten testdag #1

Testcase	Voldaan	Opmerking
T1	Y	
T2	Y	Met de opstelling die tijdens deze testdag werd gebruikt werden er ongeveer 16 locatie updates per seconden ontvangen. Er is gebleken dat dit genoeg is om geen haperingen te zien bij het volgen. Er is nog verder onderzoek nodig naar het minimum aantal locatie updates.
T3	-	Er is tijdens deze dag alleen maar getest met een gebied van ongeveer 40 x 25 x 5 meter.
T4	-	Er was tijdens deze testdag maar 1 tracking device aanwezig.
T5	Y	
T6	N	Het systeem was te verstoren door een body block. Dus dat je lichaam zich tussen het tracking device en een antenne bevond. Tijdens de testdag moet er dus worden rondgelopen met het tracking device boven je hoofd.
T7	Y	
TN1	Y	Het systeem paste op dit moment al in een normale rugtas en het wordt in de toekomst nog kleiner.
TN2	-	Er was tijdens deze testdag niet genoeg tijd om een duurttest uit te voeren.
TN3	-	Deze test is niet uitgevoerd tijdens de testdag. Kijken naar de resultaten van de testdag is wel naar voren gekomen dat de huidige nauwkeurigheid van het systeem waarschijnlijk binnen een paar centimeters ligt. Dit is vanwege tijdsgebrek nog niet getest.
TN4	-	Deze test is niet uitgevoerd tijdens de testdag. Kijken naar de resultaten van de testdag is wel naar voren gekomen dat de huidige waarden die het systeem doorstuurt correct genoeg zijn om een lamp zodanig aan te sturen dat deze je volgt. Wegens tijdsgebrek is de test echter nog niet uitgevoerd.
TN5	NIEUW ¹	

Notitie (1): Tijdens de testdag is er een nieuwe eis van het systeem naar voren gekomen namelijk: het tracking device moet gemakkelijk draagbaar zijn door de gebruiker bijvoorbeeld door het tracking device te verwerken in een kledingstuk.

Vervolgstappen

- Er zit een verband tussen de grootte van het oppervlak dat moet worden voorzien van tracking, het aantal antennes in het systeem en het aantal tracking devices en het aantal locatie updates per seconden (testcase 2) (zie hoofdstuk 2). Dit verband moet beter onderzocht worden.
- Er moet worden onderzocht hoeveel locatie updates er minimaal per seconden nodig zijn om geen haperingen in de effecten waar te nemen.
- De tests die deze testdag niet konden worden getest moeten de volgende testdag worden getest.

Bijlagen

- I. Bevindingen testdag #1

Testdag #1

Deze testdag heeft plaatsgevonden op maandag 26 oktober 2015 in het Amsterdam theater. De volgende zaken zijn bereikt:

- Een tracking device weergeven in de 3D wereld en deze de tracker laten volgen.
- Een moving head zodanig aansturen dat deze met zijn lichtstraal de tracker volgt.
- Een moving head zodanig aansturen dat deze van kleur verandert zodra er een bepaald gebied wordt binnengegaan met de tracker.

Er waren tijdens de testdag twee problemen die meer tijd kostte dan verwacht. Het eerste probleem was dat het parsen van de locatie data van Sendrato niet geheel klopte: bepaalde assen waren namelijk omgedraaid, ook moesten de positiewaarden (X, Y en Z) worden geconverteerd naar (little | big) endian. Verder was het niet duidelijk waar het tracking device zich precies bevond in de ruimte. Dit is opgelost door een grid te tekenen waardoor het gemakkelijk werd om te controleren of de positie van het tracking device klopte.

Het andere probleem had betrekking op hoe de moving head de pan en de tilt berekende op basis van de positie van de tracker. Door dit probleem scheen de moving head niet op de juiste positie. Samen met Sander is de code die hiervoor verantwoordelijk was zodanig aangepast dat deze wel correct werkte.

Notitie (1): de werking van de methode die de pan en tilt berekent voor de moving head zodat deze bijvoorbeeld een tracking device kan volgen is nog niet helemaal bekend. De werking van deze methode moet verder onderzocht worden.

Bevindingen:

Tijdens de testdag zijn een aantal bevindingen gedaan. Deze worden hieronder behandeld. Voor sommige bevindingen zijn al oplossingsrichtingen bepaald.

1. Tijdens de testdag bleek dat de precieze betekenis van de X, Y en Z coördinaten per opstelling kan veranderen. Zo betekent een positieve Z waarde de ene keer dat een object voor je ligt en de andere keer dat een object achter je ligt. Ook zouden de assen geheel van betekenis kunnen veranderen. Dus dat een X as de diepte aangeeft in plaats van de lengte. Het is van belang om deze waarden te converteren naar voor de applicatie bruikbare waarden. De applicatie werkt namelijk altijd met hetzelfde coördinaten systeem.
2. Het is gebleken (zoals bekend) dat het toevoegen van lampen zeer lastig is. Dit komt doordat je niet de precieze positie en rotatie van de lamp weet. De X en Y positie van een lamp kunnen wel relatief nauwkeurig worden bepaald door precies onder de lamp te gaan staan (als dit mogelijk is) en vervolgens de X en Y positie van het tracking device noteren. De Z positie (hoogte) en de rotatie van de lamp zijn dan echter nog steeds onbekend en moeten worden geraden. Door trail and error kan er dan op waarde gekomen worden waarmee kan worden gewerkt. Om de applicatie bruikbaar te maken moet er een manier worden bedacht zodat de lampen sneller kunnen worden ingevoerd met de correcte positie en rotatie in de applicatie.

Notie (2): bevinding twee kan deels worden opgelost door bijvoorbeeld het toetsenbord te gebruiken om de positie of rotatie van een lamp bij te werken totdat deze klopt. Zolang de

positie van de lamp wordt 'fine getuned' heeft de lamp een andere kleur zodat deze makkelijker te herkennen is.

3. De applicatie bood nog geen ondersteuning voor het automatisch weergeven van de ankerpunten (antennes) in de applicatie. Het toevoegen van de ankerpunten is echter wel gewild omdat deze referentiepunten vormen voor de gebruiker. Ook is het misschien mogelijk om aan de hand van de posities van de ankerpunten de assen te bepalen en dus bevinding 1 op te lossen.
4. Het volgeffect (en het gebied effect) reageerde naar ons idee te traag. Er was een duidelijke vertraging merkbaar tussen het veranderen van de positie van de tracker en het updaten van de lamp. Om dit probleem te kunnen tackelen moet de vertraging die de applicatie veroorzaakt worden gemeten. Vervolgens kunnen er optimalisaties worden doorgevoerd.
5. Tijdens de loop van het programma is het interessant om een aantal verschillende waarden weer te geven in de GUI. Dit zijn: hoeveel locatie updates er per seconde worden ontvangen per tracking device, zien wat de positie van een tracking device is, inzicht krijgen in het aantal frames per seconden van de applicatie (FPS) en de delay in milliseconden van het ontvangen van een locatie update tot het uitzenden van de DMX waarden.
6. Sommige moving heads zitten gemonteerd aan een truss die kan bewegen. Op het moment dat de moving head gaat bewegen (door de pan en tilt aan te passen) kan het zijn dat de truss een beetje gaat schommelen. Dit kan er voor zorgen dat je de lichtstraal ziet bewegen terwijl de moving head stil staat.

Vervolgstappen:

Vanuit de bevindingen van de testdag kunnen de vervolgstappen worden bepaald.

1. Wanneer komt de volgende test dag?

Sander gaat samen met de betrokken partijen samenzitten om de resultaten van deze testdag door te nemen. Afhankelijk van wat hier uitkomt wordt er een nieuwe testdag georganiseerd.

2. Wat willen we de volgende testdag 'af' hebben en testen?

De resultaten van vervolgstap 3 worden de volgende testdag getest. Er gaat waarschijnlijk meer worden getest, maar dit kan pas worden bepaald als er met de verschillende betrokken partijen is gesproken.

3. Afwerken van het lijstje van bevindingen die met deze testdag zijn gevonden. Deze bevindingen waar mogelijk voorzien van een prioriteit zodat er kan worden gefocust op de belangrijkste bevinding.

Samen met Sander zijn de verschillende bevindingen voorzien van een prioriteit. Dit heeft geleid tot de volgende verdeling:

Opzettijd van de applicatie versnellen

Het opzetten van de applicatie kostte tijdens deze eerste testdag ongeveer vier uur. Dit moet de volgende testdag minder tijd kosten (maximaal een halfuur). De applicatie is opgezet als: de locatie van een tracker correct wordt weergegeven in de applicatie.

Volgende punten moeten hiervoor worden geïmplementeerd:

- De verschillende antennes moeten worden weergegeven in de applicatie.

- Er moet een grid worden weergegeven die het aflezen van de positie van een tracking device makkelijker maakt
- Er moet een configuratie bestand komen met daarin de configuratie van de opstelling. Vanuit dit bestand kan de applicatie de X, Y, Z waarden converteren naar voor binnen de applicatie bruikbare waarden

Inzicht in verschillende variabelen van de applicatie

Er is gebleken dat het handig is om verschillende debug waarden realtime te kunnen zien binnen de applicatie. Dit zijn: de frames per seconden (FPS) en meer informatie over de tracking devices.

De volgende info moet per tracking device worden weergegeven in bijvoorbeeld een lijstje binnen de applicatie: positie van het tracking device binnen de applicatie, positie van het tracking device die wordt ontvangen, het aantal locatie updates per seconde van het tracking device en het tijdstip van de laatste locatie update.

Rest van de bevindingen

De bevindingen B2 en B4 worden na de volgende testdag verder uitgewerkt.

- 4. De code die tijdens de testdag is geschreven moet worden opgeschoond zodat het beter binnen de applicatie past. Ook moet de werking van de methode die de pan en tilt berekent van een moving head worden onderzocht.*

Dit moet worden meegenomen in de volgende sprint

Bijlage E

User interface rapport

HAUTE TECHNIQUE

User Interface

Auteur	: Jan de Boer
Bedrijf	: HAUTE TECHNIQUE
Opdrachtgever	: Sander ter Braak
Datum	: 12 oktober 2015
Versie	: 1.0

Versiebeheer

Versie	Datum	Opmerking
0.1	14-10-2015	Eerste versie
0.2	30-10-2015	Keuze voor UI framework uit analyse document toegevoegd en nader toegelicht.
1.0	05-01-2016	Verschillende wijzigingen in de opmaak doorgevoerd.

Gerelateerde documenten

Ref	Auteur	Naam
D1	Jan de Boer	Requirements Document

Inhoudsopgave

1	Introductie	6
2	Scenario's	7
3	Lampen beheren	8
3.1	Lampen toevoegen	8
3.2	Lampen aanpassen	9
3.3	Lampen verwijderen	9
4	Effecten beheren	10
4.1	Overzicht	10
4.2	Koppelen	10
4.3	Parameters weergeven en aanpassen	11
4.3.1	Parameters aanpassen met een MIDI controller	11
4.4	Ontkoppelen	11
4.5	Effect scripts	11
5	Tracking devices beheren	12
5.1	Tracking device toevoegen	12
5.2	Keuze menu	13
5.3	Overzicht van tracking devices	13
5.4	Tracking device verwijderen	13
6	Opnamen	14
6.1	Opnemen	14
6.2	Afspelen	14
6.3	Verwijderen van opnamen	14
7	Configuratie beheren	15
8	Instellingen beheren	15
9	DMX nodes beheren	16
9.1	Node	16
9.1.1	Toevoegen	16
9.1.2	Overzicht	16
9.1.3	Verwijderen	16
9.2	Universes	16
9.2.1	Toevoegen	16
9.2.2	Overzicht	17
9.2.3	Verwijderen	17

10	User Interface	18
10.1	Bestaande interfaces	18
10.2	Gevonden interfaces	19
11	Frameworks	22
11.1	Assets.....	22
1.1.1	UI Frameworks	23
11.1.1	Plugins	25
11.2	Keuzen voor framework.....	27

1 Introductie

Voordat er kan begonnen worden met het kiezen van een geschikt framework voor het creëren van de user interface van de applicatie is het belangrijk om eerst alle losse aspecten van de user interface beschreven te hebben. De bevindingen die hierbij worden gedaan kunnen vervolgens worden meegenomen in de keuze voor een geschikt framework.

2 Scenario's

In sprint 1 van het project zijn er aan de hand van een use-case diagram verschillende use case scenario's beschreven; deze zijn terug te vinden in requirements document [D1]. De scenario's die toen zijn beschreven kunnen worden gebruikt als input voor het beschrijven van de verschillende onderdelen van de UI. In de tabel hieronder zijn de verschillende onderdelen te vinden. De onderdelen O6 en O7 zijn niet direct gelinkt aan een scenario. Dit komt omdat ze de andere scenario's ondersteunen.

Tabel 1 Onderdelen van de user interface

Nr.	Onderdeel naam	Scenario's
O1	Lampen beheren	1.1, 1.2, 1.3
O2	Effecten beheren	3.1,
O3	Tracking devices beheren	5.1, 5.2
O4	Locatie data opslaan / afspelen	6.1, 7.1
O5	Configuratie beheren	8.1, 8.2, 8.2
O6	Instellingen beheren	-
O7	DMX nodes beheren	-

Elk onderdeel dat in de tabel hierboven wordt beschreven wordt in de loop van dit document verder uitgewerkt en toegelicht.

3 Lampen beheren

Binnen de applicatie is het mogelijk om lampen toe te voegen. Deze lampen corresponderen met fysieke lampen. De fysieke lampen worden aangestuurd via het DMX protocol. Binnen de applicatie zijn er op dit moment twee verschillende typen lampen te onderscheiden namelijk: normale lampen (deze kunnen alleen aan en uit en van kleur veranderen) en moving heads. Moving heads kunnen het zelfde als normale lampen maar kunnen hiernaast ook bewegen en verschillende effecten op de lichtstraal toepassen.

Op dit moment wordt er voornamelijk gefocust op het aansturen van verlichting. De applicatie is echter zo opgezet dat er (in de toekomst) gemakkelijk ook andere DMX apparaten kunnen worden aangestuurd. Denk bijvoorbeeld aan rookmachines. In het vervolg van dit document wordt er soms gesproken van apparaten in plaats van lampen. Hiermee worden dus DMX apparaten in het algemeen bedoeld.

3.1 Lampen toevoegen

Om lampen te kunnen toevoegen aan het systeem zijn er een aantal verschillende gegevens nodig. Dit zijn de volgende gegevens:

Tabel 2 Parameters om een lamp toe te voegen

Naam	Type	Omschrijving
Positie	Vector3	Bestaat uit 3 componenten namelijk de X, Y en Z waarden.
Rotatie	Quaternion	Bestaan uit 3 componenten namelijk de X, Y en Z rotatie.
Type lamp	String	Het type van de lamp, dit komt overeen met het type uit een XML bestand.
DMX universe	DmxUniverse	In welk DMX universe de lamp zich bevindt.
DMX adres	int	Het adres binnen het universe.

Validatie

De input die de gebruiker invoert dient te worden gevalideerd volgens bepaalde regels. De regels worden in de tabel hieronder beschreven:

Tabel 3 Validatie regels voor de parameters van een lamp

Naam	Type
Positie	Een geldig Vector3 type. Hierbij bevinden de verschillende componenten van de vector (X, Y en Z) zich tussen de minimale en de maximale waarden die een float kan opslaan.
Rotatie	Een geldig Quaternion type, hierbij bevinden X, Y en Z rotatie zich tussen de float waarde 0 en de 360 graden.
Type lamp	-
DMX universe	-
DMX adres	Een int waarden tussen de 0 en de 512.

Batch toevoegen

Een wens van de opdrachtgever is de batch toevoegen van lampen. Hierbij wordt gebruik gemaakt van bijvoorbeeld een XML bestand om in een keer meerdere lampen aan het systeem toe te voegen. Om deze functionaliteit te ondersteunen is er nog verder onderzoek nodig.

Nadat de gegevens van de lamp zijn ingevuld kan de lamp worden toegevoegd aan de 3D wereld. Op het moment dat dit is gedaan kan de lamp worden gebruikt in effecten.

3.2 Lampen aanpassen

Het moet mogelijk zijn om nadat de lamp is toegevoegd de verschillende gegevens van een lamp aan te passen. Hierbij moet rekening worden gehouden dat het aanpassen van de positie en de rotatie van de lamp gemakkelijk moet worden gemaakt, aangezien dit iets is wat regelmatig gebeurt tijdens het kalibreren van de lampen.

De exacte positie en rotatie van een lamp zijn van groot belang voor de applicatie en de effecten binnen de applicatie. Op het moment dat de positie en of de rotatie van een lamp niet kloppen kunnen de effecten niet goed functioneren. Haute Technique is bezig met het ontwerpen van een systeem dat de positie en rotatie van een lamp (semi) automatisch moet kunnen bepalen. Zolang dit systeem er nog niet is moeten de positie en de rotatie van een lamp nog met de hand worden bepaald. Aangezien dit foutgevoelig is moet het binnen de applicatie mogelijk zijn om gemakkelijk de positie en rotatie van een lamp aan te passen.

3.3 Lampen verwijderen

Een lamp kan om wat voor reden dan ook weer van de applicatie worden verwijderd. Het verwijderen van een lamp kan alleen gebeuren als de lamp niet is gekoppeld aan een effect. Er zijn op dit moment twee scenario's beschreven voor het verwijderen van een lamp:

- Door te klikken op de 3D representatie van de lamp komt er een menu tevoorschijn. In dit menu is een knop beschikbaar die het mogelijk maakt om de lamp te verwijderen.
- Binnen de applicatie is een lijst van alle toegevoegde apparaten te zien. Vanuit deze lijst kan een apparaat worden verwijderd.

4 Effecten beheren

Effecten binnen de applicatie zorgen er voor dat lampen zich volgens een bepaald patroon gaan gedragen. Een effect stuurt aan de hand van gebeurtenissen lampen aan. Zo kan een effect bijvoorbeeld het laten knipperen van een lamp zijn.

Binnen de applicatie zijn er twee soorten effecten te onderscheiden namelijk effecten die gebruik maken van een tracking device en dus kunnen reageren op veranderingen van de positie van een tracking device en effecten die dat niet doen. Het laatste soort effecten kan alleen lampen over de tijd aanpassen. Elk effect beschikt over de volgende eigenschappen:

- Naam, auteur en omschrijving van het effect.
- Of het effect tracking devices ondersteunt en hoeveel.
- Of het effect apparaten (lampen) ondersteunt en hoeveel.
- Welk type apparaten (lampen) het effect ondersteunt.
- Parameters van het effect; deze kunnen verschillende typen hebben, voor nu zijn dit: int, float, bool en een string en kunnen tijdens de loop van een effect worden veranderd door de gebruiker.

4.1 Overzicht

De applicatie biedt een overzicht van de verschillende effecten die binnen de applicatie beschikbaar zijn. Hierbij zijn de volgende gegevens te zien per effect: naam, auteur, omschrijving en met welk type apparaten het effect overweg kan.

De applicatie biedt ook een overzicht van actieve effecten. Dit zijn effecten die zijn gekoppeld met tracking devices en of apparaten. De gebruiker kan op deze lijst acties uitvoeren bijvoorbeeld: effect aan / uit zetten, effect verwijderen, effect aanpassen (dus nieuwe apparaten of tracking devices toekennen) of de parameters lijst van het effect laten zien.

4.2 Koppelen

De applicatie beschikt over een lijst van effecten die kunnen worden toegepast op de beschikbare apparaten en tracking devices. Deze apparaten en tracking devices moeten op de een of andere manier worden gekoppeld met een effect. De volgende scenario's zijn hiervoor bedacht:

- De gebruiker selecteert een aantal verschillende apparaten en of tracking devices binnen de applicatie. Door op een knop te drukken krijgt de gebruiker een lijst van beschikbare effecten te zien (de applicatie laat alleen de effecten zien die kunnen worden gekoppeld met alle apparaten die geselecteerd zijn dus het effect moet het type apparaat ondersteunen en het effect moet ondersteuning hebben voor het aantal toe te voegen apparaten en tracking devices).
- De gebruiker creëert uit een lijst van alle beschikbare effecten een effect. De applicatie komt vervolgens in een soort link modus waarbij alle apparaten die kunnen worden gekoppeld met het effect oplichten. Door te klikken en te selecteren kunnen de apparaten vervolgens aan het effect worden toegevoegd. Dit geldt ook voor de tracking devices. Hierbij is een label te zien die weergeeft hoeveel apparaten en tracking devices er nog kunnen worden toegevoegd aan het effect.

4.3 Parameters weergeven en aanpassen

Zoals beschreven in het hoofdstuk overzicht kan vanuit het overzichtsscherf van de actieve effecten een scherm worden weergegeven die de verschillende parameters van een effect laat zien en aanpasbaar maakt. Hierdoor kan het gedrag van een effect 'on the fly' worden aangepast.

4.3.1 Parameters aanpassen met een MIDI controller

De opdrachtgever heeft de wens om de parameters van een effect te kunnen koppelen met een 3^e party remote om zo makkelijker de parameters van een effect aan te passen. Er moet nog nader onderzoek worden gedaan naar de verschillende mogelijkheden hiervoor.

4.4 Ontkoppelen

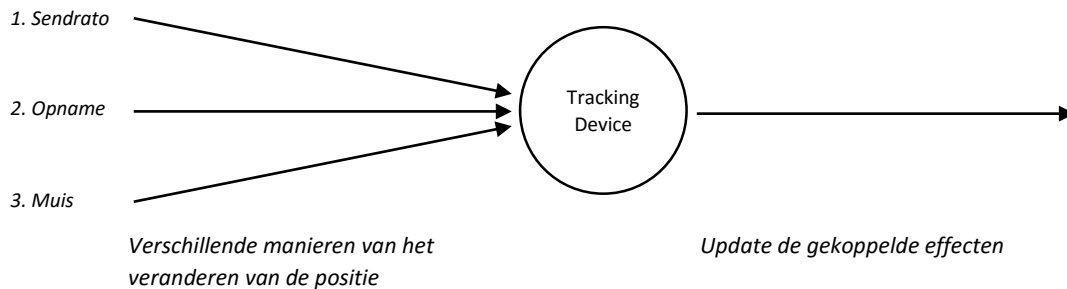
Een actief effect kan gekoppeld zijn met verschillende apparaten en tracking devices. Vanuit de lijst met actieve effecten heeft de gebruiker de mogelijkheid om het effect te ontkoppelen (ook wel verwijderen). Bij het ontkoppelen van een effect wordt eerst het effect uitgezet om alle apparaten terug te zetten in hun standaard status om vervolgens te ontkoppelen met de verschillende apparaten en of tracking devices. Hierna kan het effect worden verwijderd.

4.5 Effect scripts

Effecten kunnen on the fly worden toegevoegd door nieuwe effect scripts te schrijven en deze in de effect folder van de applicatie te plaatsen. De applicatie compileert de gevonden effecten en voegt deze toe aan de beschikbare effecten als er geen fouten in het script zitten. Op het moment dat er fouten worden gevonden in het effect script moet de applicatie deze fouten weergeven binnen de applicatie.

5 Tracking devices beheren

Tracking devices zijn naast effecten het kloppend hart van de applicatie. Tracking devices kunnen worden gelinkt aan effecten zodat effecten kunnen reageren op het veranderen van de positie van een tracking device. Een tracking device kan zijn locatie updates uit verschillende bronnen halen voor nu zijn dit: de locatie sensoren van Sendrato, de positie van de muis in de 3D wereld of een opname.



5.1 Tracking device toevoegen

Een tracking device kan op verschillende manieren worden toegevoegd. De eerste manier is doordat een nieuw tracking device wordt 'gevonden' door de server. Deze server communiceert met het systeem van Sendrato. Er zijn twee manieren om dit scenario af te handelen:

- Op het moment dat er een nieuw tracking device wordt gevonden wordt deze direct aan de applicatie toegevoegd. Er komt een 3D weergave van het tracking device in de wereld en de gebruiker is in staat om effecten aan het tracking device toe te voegen.
- Op het moment dat er een nieuw tracking device wordt gevonden krijgt de gebruiker een melding te zien. Door op een knop te drukken ziet de gebruiker een lijst met gevonden devices. Deze kan de gebruiker vervolgens toevoegen door op een knop te drukken. Op het moment dat de gebruiker op de knop heeft gedrukt wordt het tracking device toegevoegd aan de 3D wereld en kunnen er effecten aan worden gekoppeld.

De tweede manier van toevoegen is als de gebruiker een opname wil starten. De gebruiker krijgt bij het starten van een opname namelijk de keus om een tracking device te creëren en daar de opname aan te koppelen of om de opname aan een bestaand tracking device te koppelen. Bij de laatste keus heeft de gebruiker dus al eerder een tracking device toegevoegd aan het systeem.

Zoals hierboven beschreven zijn er dus verschillende manieren waardoor een tracking device van positie kan veranderen. Voor nu zijn dit: door een update van het systeem van Sendrato, door het afspelen van een opname of door het veranderen van de positie van de muis.

Een gebruiker kan een tracking device toevoegen zonder dat deze nog een update methode heeft (de update methode bepaald waardoor het device van positie veranderd bijvoorbeeld door de muis of door het systeem van Sendrato). De gebruiker is op dat moment in staat om effecten aan het desbetreffende tracking device toe te voegen. De effecten zullen (waarschijnlijk) nog niks doen aangezien het tracking device nog niet van positie veranderd. Op het moment dat de gebruiker een update methode toevoegt kan het tracking device van positie veranderen. Nu hoeft de gebruiker dus niet telkens opnieuw de koppeling met de effecten te maken als de gebruiker van input methode wil veranderen.

5.2 Keuze menu

Door op een tracking device in de 3D wereld te klikken komt er een keuzemenu tevoorschijn. Met dit menu kan je de volgende acties doen: verwijderen en input methode aanpassen.

5.3 Overzicht van tracking devices

Binnen de applicatie is een lijst van alle tracking devices te zien. Vanuit deze lijst kunnen nieuwe tracking devices worden toegevoegd, tracking devices worden verwijderd en de input methode van tracking devices veranderen.

5.4 Tracking device verwijderen

Een tracking device kan op twee manieren worden verwijderd namelijk: doordat de gebruiker op de knop verwijderen drukt in de lijst met tracking device of door op verwijderen te klikken in het keuze menu dat de gebruiker te zien krijgt als deze op de 3D weergave van het tracking device drukt.

Op het moment dat een tracking device wordt verwijderd wordt het tracking device ook van de effecten waaraan het desbetreffende tracking device is gekoppeld verwijderd. Hierdoor 'werken' deze effecten niet meer. Een andere mogelijkheid is dat op het moment van verwijderen ook de gekoppelde effecten worden verwijderd.

6 Opnamen

Het is mogelijk om de locatie van een tracking device op te nemen en deze op een later tijdstip weer af te spelen.

6.1 Opnemen

Voor het opnemen van locatie data is een tracking device nodig. De gebruiker kiest een tracking device dat moet worden opgenomen en drukt op een knop om de opname te starten. Er zijn verschillende scenario's te bedenken om de opname te starten:

- De gebruiker drukt op de visuele representatie van de tracking device in de 3D wereld; er komt vervolgens een menu tevoorschijn die een keuze biedt om een opname te starten.
- De gebruiker heeft een lijst van beschikbare tracking devices voor zich. Om een opname te starten drukt de gebruiker op het desbetreffende tracking device.

6.2 Afspelen

Voor het afspelen van een opname kiest de gebruiker uit een lijst met opnamen. Op het moment dat de gebruiker een keuze heeft gemaakt voor een opname drukt de gebruiker op een knop om de opname te initialiseren. Er wordt dan een nieuw tracking device aan de applicatie toegevoegd. De opname wordt dan nog niet gestart. De gebruiker heeft op dit moment de mogelijkheid om het gecreëerde tracking device toe te voegen aan effecten.

Een andere mogelijkheid is dat de gebruiker van te voren een playback tracking device heeft toegevoegd en deze aan bepaalde opnamen koppelt.

Starten

De gebruiker kan een lijst van geïnitieerde opnamen opvragen en deze desgewenst starten / stoppen of pauzeren. Of de gebruiker klikt op het tracking device en krijgt een keuzemenu te zien met de optie om een opname te starten.

Pauzeren

Op het moment dat de gebruiker een opname wil pauzeren heeft deze twee keuzes: door te klikken op het desbetreffende tracking device en uit het keuzemenu dat tevoorschijn komt kiezen voor stoppen. Of door in de lijst met geïnitieerde opnamen de opnamen te pauzeren.

Stoppen

Op het moment dat de gebruiker een opname wil stoppen heeft deze twee keuzen: door te klikken op het desbetreffende tracking device en uit het keuzemenu dat tevoorschijn komt kiezen voor stoppen. Of door in de lijst met geïnitieerde opnamen de opnamen te stoppen. Op het moment dat de gebruiker op stop drukt krijgt hij de vraag of hij ook het gelinkte tracking device wil verwijderen.

6.3 Verwijderen van opnamen

Binnen de applicatie is een lijst beschikbaar met de verschillende opnamen. Vanuit deze lijst kan de gebruiker een opname verwijderen.

7 Configuratie beheren

Dit onderdeel wordt niet tijdens dit project geïmplementeerd. Daarom wordt het ook niet mee genomen in het ontwerp voor de user interface.

Het configuratie onderdeel zou het volgende inhouden: opslaan van toegevoegde apparaten, tracking devices, effecten en DMX nodes en deze weer kunnen inladen en het doorlopen van een test sequentie om te testen of alle toegevoegde lampen naar behoren werken.

8 Instellingen beheren

Binnen de applicatie zijn er een aantal verschillende globale instellingen te vinden. De gebruiker moet in staat zijn om deze instellingen in te zien en aan te passen. De volgende instellingen zijn beschikbaar:

Tabel 4 Instellingen binnen de applicatie

Naam	Type	Omschrijving
Sendrato port	Int	De UDP port waar het systeem van Sendrato de UPD pakketjes naar toe stuurt.
Effects map	String	De map waar de verschillende effect scripts zich bevinden.
Opnamen map	String	De map waarin de opnamen worden geplaatst.

9 DMX nodes beheren

9.1 Node

DMX nodes worden binnen de applicatie gebruikt voor het aansturen van de fysieke lampen via het DMX protocol. Er zijn verschillende type DMX nodes te onderscheiden bijvoorbeeld ArtNet nodes of USB nodes. Op dit moment biedt de applicatie alleen ondersteuning voor ArtNet nodes.

9.1.1 Toevoegen

Afhankelijk van het type node dat wordt toegevoegd zijn er verschillende parameters nodig. Er geldt dus dat de gebruiker eerst een type node invult of selecteert en vervolgens de desbetreffende parameters te zien krijgt.

ArtNet node

Voor een ArtNet node zijn de volgende parameters nodig:

Tabel 5 Parameters ArtNet node

Naam	Type	Omschrijving
IP	String	IP adres van de ArtNet node
Port	Int	Port van de ArtNet node

ArtNet nodes kunnen ook automatisch worden ontdekt door gebruik te maken van een ArtNet poll pakketje. Dit is een pakketje binnen het ArtNet protocol dat gebruikt wordt voor het ontdekken van nodes. Deze functie zou binnen de applicatie ook gebruikt kunnen worden.

9.1.2 Overzicht

Binnen de applicatie is een overzicht te zien van de verschillende nodes binnen de applicatie.

9.1.3 Verwijderen

Vanuit de node overzicht lijst kan een node worden verwijderd. Lampen zijn gekoppeld aan een DMX universe. Zo'n universe zit weer gekoppeld aan een node. Er zijn een aantal verschillende scenario's die kunnen gebeuren op het moment dat een node wordt verwijderd namelijk:

- Alle lampen die (indirect) gekoppeld zijn met de desbetreffende node worden meeverwijderd.
- Alleen de node wordt verwijderd, de desbetreffende universes niet. Deze blijven bestaan in de applicatie en kunnen op een later tijdstip weer aan de node worden toegevoegd.

9.2 Universes

De verschillende lampen binnen de applicatie horen bij een DMX universe. Een DMX universe wordt gebruikt om de verschillende kanalen bij te houden en over DMX door te sturen.

9.2.1 Toevoegen

Elke DMX node kan meerdere DMX universe bevatten. Deze universes kunnen op het moment dat de node wordt gecreëerd worden toegevoegd en gekoppeld met de node, maar de universes zouden ook later aan de node kunnen worden toegevoegd. Voor het toevoegen van een DMX universe zijn de volgende gegevens nodig:

Tabel 6 Parameters universe

Naam	Type	Omschrijving
Universe	Int	Het nummer van het universe.

9.2.2 Overzicht

Binnen de applicatie is een lijst beschikbaar met daarin alle DMX universes van de applicatie. Vanuit deze lijst kunnen universes worden verwijderd en toegevoegd. Ook kunnen universes vanuit deze lijst worden gekoppeld met een DMX node.

9.2.3 Verwijderen

Het verwijderen van een DMX node kan alleen gebeuren op het moment dat het universe niet in gebruik is door een lamp.

10 User Interface

Nu de applicatie voor een groot deel is beschreven kan er worden gekeken naar hoe de user interface er uit kan komen te zien. Er kan hierbij worden gekeken naar user interfaces van bestaande programma's. Het is echter van belang om eerst de gebruikscenario's van de applicatie te beschrijven. Het gebruik van de applicatie is grofweg op te delen in drie onderdelen namelijk:

Voor de show: Voordat de show begint moet de applicaties worden ingeregeld. De verschillende lampen moeten worden toegevoegd aan de applicaties en de werking van de lampen moet worden getest. Ook moeten de effecten die tijdens de show worden gebruikt getest worden.

Tijdens de show: Tijdens de show worden bepaalde effecten aan- of uitgezet. Ook kunnen de parameters van een effect worden aangepast. Hiermee wordt de 'show' gecreëerd. In eerste instantie neemt de applicatie niet de hele lichtshow over. Binnen een show worden bepaalde momenten in gepland waarbij de applicatie wordt gebruikt voor het aansturen van de lichten aan de hand van de locatie van een tracking device of devices.

Effecten schrijven: Er kunnen nieuwe effecten aan de applicatie worden toegevoegd. Deze effecten moeten worden 'ontwikkeld' dit houdt in, ze moeten worden toegevoegd aan de applicatie en getest worden.

De onderdelen 'effecten schrijven' en 'voor de show' hebben overlap met elkaar. De nieuwe effecten kunnen namelijk vaak pas echt getest worden als de applicatie de verschillende fysieke lampen die tijdens de show worden gebruikt heeft toegevoegd. Het is grofweg te zeggen dat de applicatie een live en een niet live gedeelte heeft.

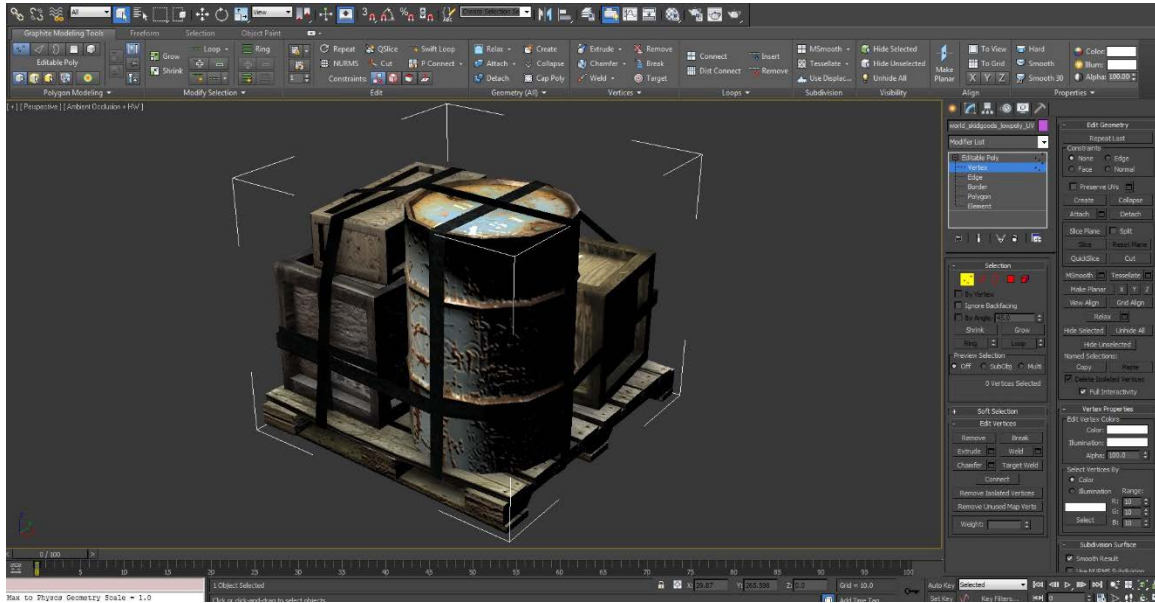
10.1 Bestaande interfaces

Om een beeld te krijgen van de verschillende mogelijkheden die er zijn voor de UI is er een analyse uitgevoerd naar bestaande interfaces van andere programma's. Er is tot een lijst van programma's gekomen door binnen Haute Technique rond te vragen welke programma's er worden gebruikt en welke bij de afstudeerder bekend zijn. Deze lijst is vervolgens gefilterd op bruikbaarheid aan de hand van de volgende criteria:

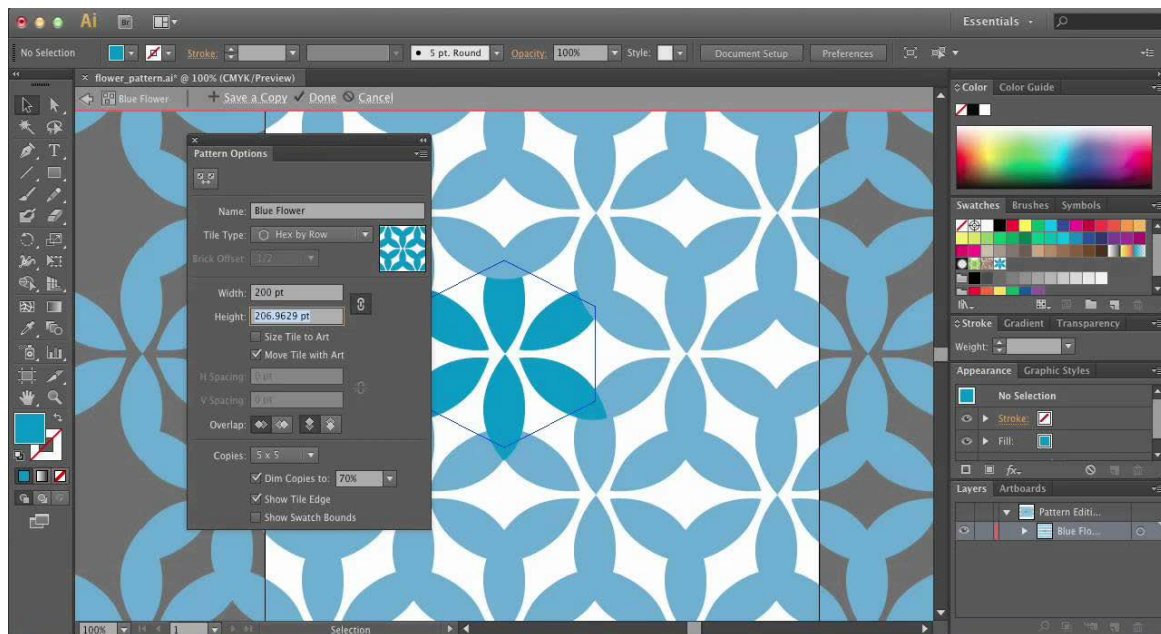
- Laat de applicatie een 3D omgeving zien of wordt de applicatie gebruikt bij live performances door bijvoorbeeld DJ's of VJ's?
- Heeft de interface unieke elementen die nog niet te zien zijn in de al onderzochte interfaces?

10.2 Gevonden interfaces

In de onderstaande figuren zijn de interfaces te vinden van de verschillende programma's die na het onderzoek zijn gevonden.

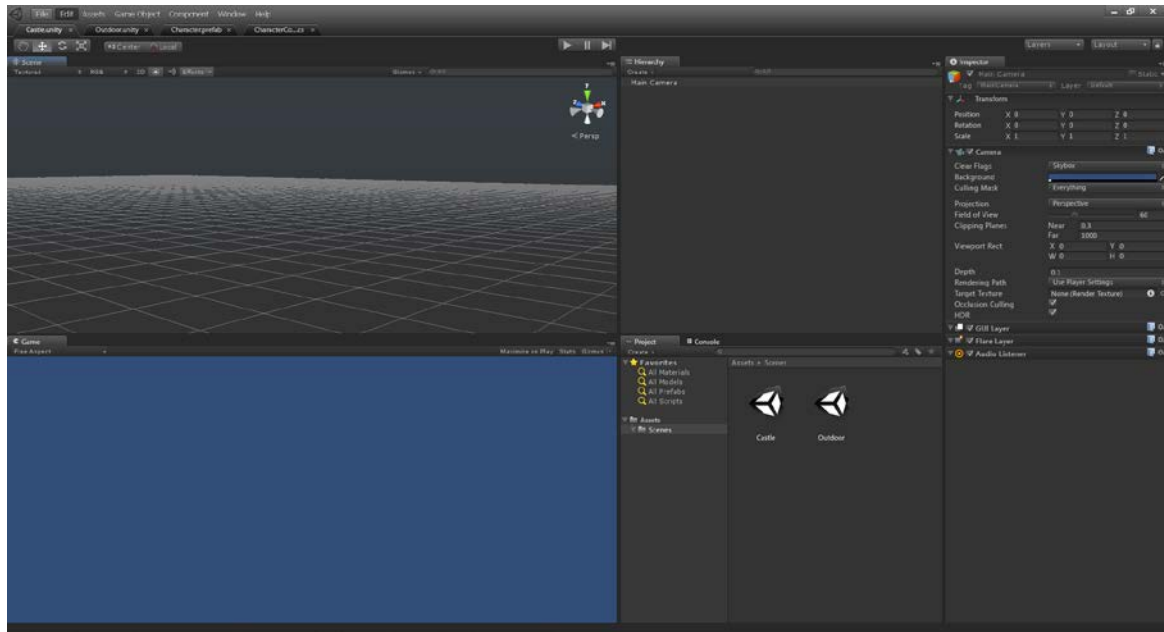


Figuur 1 UI 3DS max

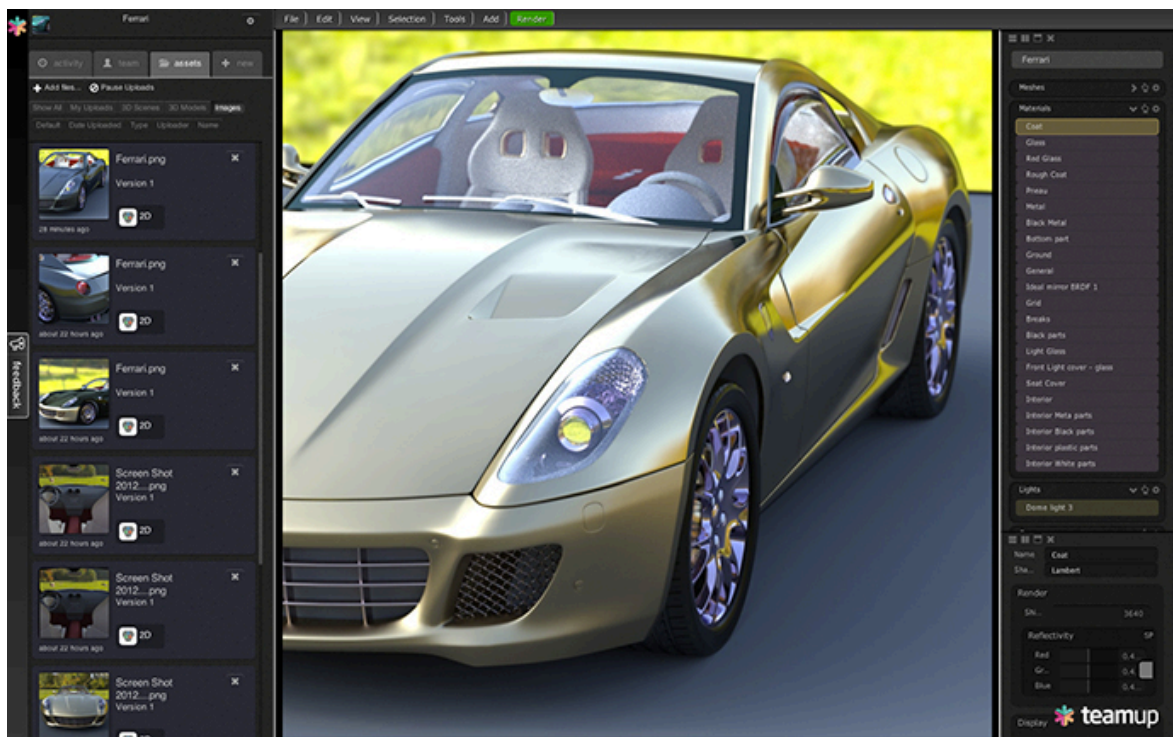


Figuur 2 UI Illustrator

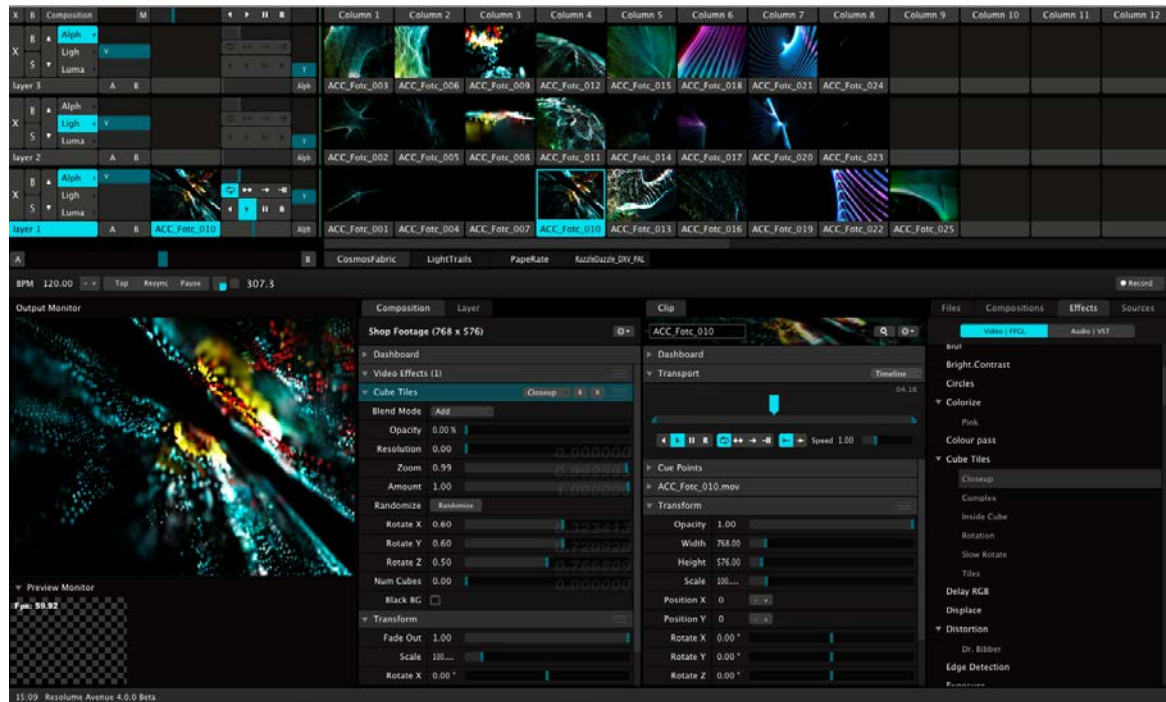
HAUTE TECHNIQUE



Figuur 3 UI Unity



Figuur 4 UI Teamup



Figuur 5 UI Resolume Arena



Figuur 6 UI Traktor Pro

11 Frameworks

De user interface is voor een groot deel beschreven en de verschillende functionaliteiten van de UI zijn bekend. Deze ideeën voor de user interface kunnen nu worden omgezet in een werkende UI binnen Unity. Voordat dit kan worden gedaan moet er onderzocht worden hoe dit binnen Unity kan worden gedaan. Hiervoor zijn verschillende opties binnen Unity. Zo kan er gebruik worden gemaakt van Unity's eigen UI systeem of van dat van een derde ontwikkelaar.

Het UI systeem van Unity biedt een basisset aan UI componenten zoals: knoppen, checkboxen of invoervelden. Met deze componenten kan in principe elke UI worden gecreëerd. Er zijn echter plugins voor UI's beschikbaar die het ontwikkelen van een UI versimpelen, zo hebben sommige van deze plugins standaard ondersteuning voor alert boxen of tabellen. Door gebruik te maken van een plugin kan er tijd worden bespaard op het ontwikkelen van de UI.

Hieronder zijn een aantal verschillende aspecten van de UI opgesomd die in de hoofdstukken hiervoor zijn beschreven:

- De UI maakt gebruik van windows (net zoals in Windows, Linux en OSX) om de gebruiker content te laten zien.
- Er komen binnen de applicatie verschillende soorten lijsten voor onder andere: tracking devices, effecten, apparaten of opnamen. Deze lijsten moeten zo nodig scrollbaar zijn en moeten de data realtime weergeven, dus de content moet veranderen als de data verandert.
- De UI maakt gebruik van verschillende standaard UI componenten namelijk: input velden, labels, checkboxen en comboboxen.
- De mogelijkheid om de teksten binnen de applicatie te vertalen (localization)
- Het scheiden van de data en de view, dit leidt tot een systeem dat makkelijker uitbreidbaar is.
- Een window manager zodat er verschillende schermen tegelijk kunnen worden weergegeven.

11.1 Assets

Unity biedt op zijn website een asset store aan. Dit is een online winkel waar uitbreidingen voor Unity kunnen worden aangeschaft. Deze uitbreidingen lopen uiteen van 3D modellen tot script om het ontwikkelen binnen Unity makkelijker te maken. Voor het zoeken naar UI frameworks is er gebruik gemaakt van deze winkel. Er zijn twee soorten assets gevonden die op dit moment toegevoegde waarde zouden kunnen hebben voor de applicatie. Dit zijn: UI frameworks die het ontwikkelen van een UI binnen Unity makkelijker (beloven) maken en plugins die een bepaalde standaard functionaliteit leveren.

Alle gevonden assets zijn beschreven in een tabel. Hieronder volgt een uitleg van het template dat gebruikt is. Bij sommige assets is de prijs 'per seat.' Unity is niet duidelijk over de precieze betekenis van deze toevoeging maar over het algemeen wordt er mee bedoeld dat de prijs moet worden betaald voor iedere ontwikkelaar die de asset gebruikt.

Naam van de asset	
Waardering	Waardering in sterren van 0 tot 5 met daarachter het aantal reviews.
Prijs	De prijs van de plugin.
Unity versie	Met welke Unity versie de plugin werkt.
Datum	Datum dat de asset voor het eerst in de winkel verscheen en de laatste datum dat de asset is geüpdatet.
Omschrijving	Een omschrijving van de asset.
URL	URL van de asset in de asset store.

1.1.1 UI Frameworks

NGUI	
Waardering	5 (2800)
Prijs	\$95 (per seat)
Unity versie	4.6.7 of hoger
Datum	10 december 2011 – 7 oktober 2015
Omschrijving	NGUI ook wel next-gen UI is een van de oudste UI frameworks voor Unity. Het biedt een tal van standaard elementen om je snel op weg te helpen met het ontwikkelen van een UI. De maker van NGUI heeft een tijd bij Unity gewerkt voor het creëren van de nieuwe Unity UI. De nieuwe Unity UI lijkt daarom erg op die van NGUI. Verder heeft NGUI de grootste community van de verschillende frameworks in deze lijst (zoals af te leiden aan het aantal reviews).
URL	https://www.assetstore.unity3d.com/en/#!/content/2413

New UI Widgets	
Waardering	5 (22)
Prijs	\$40
Unity versie	4.6.1 of hoger
Datum	7 januari 2015 – 28 september 2015
Omschrijving	New UI widgets heeft als een van de weinige UI frameworks ondersteuning voor een TreeView. Het framework is echter niet zo flexibel als de andere frameworks, zo heeft de ListView alleen maar ondersteuning voor het weergeven van Strings.
URL	https://www.assetstore.unity3d.com/en/#!/content/27226

HAUTE TECHNIQUE

NoesisGUI 1.1	
Waardering	5 (101)
Prijs	Gratis versie beschikbaar voor Unity 4; voor de laatste development versie moet het bedrijf gecontacteerd worden. Het bedrijf geeft aan dat een licentie \$250 kost
Unity versie	4.1.4 of hoger. (geen support voor Unity5)
Datum	22 juli 2013 – 25 mei 2015
Omschrijving	NoesisGUI is de het duurste UI framework in deze lijst. Het maakt gebruik van de XAML op de UI in te schrijven. Het framework maakt gebruik van het MVVM patroon om de applicatie logica te scheiden van de view. NoesisGUI richt zich vooral op de professionele markt.
URL	https://www.assetstore.unity3d.com/en/#!/content/9282

EZ GUI	
Waardering	5 (59)
Prijs	\$200 (per seat)
Unity versie	4.5.2 of hoger
Datum	6 november 2010 – 11 maart 2015
Omschrijving	EZ GUI is de oudste plugin in deze lijst. Het is geschreven voordat de nieuwe Unity UI geschreven was en maakt daar dus ook geen gebruik van. Dit framework is echter verouderd als het wordt vergeleken met de nieuwere frameworks.
URL	https://www.assetstore.unity3d.com/en/#!/content/32

PowerUI - HTML/CSS	
Waardering	5 (36)
Prijs	\$90 (per seat)
Unity versie	4.2.0 of hoger
Datum	8 augustus 2013 – 9 maart 2015
Omschrijving	Met dit framework wordt het mogelijk om je UI te schrijven in HTML in samenwerking met CSS.
URL	https://www.assetstore.unity3d.com/en/#!/content/10374

One Minute GUI	
Waardering	4 (34)
Prijs	\$20
Unity versie	5.0.1 of hoger
Datum	1 april 2015 – 27 mei 2015
Omschrijving	One Minute GUI is een UI framework dat achter de schermen gebruik maakt van het nieuwe UI systeem. Het framework focust zich vooral op het creëren van standaard elementen en minder op het creëren van uitbreidbare elementen. Het framework is vooral geschikt (zoals de naam al doet vermoeden) voor het snel creëren van een standaard game UI.
URL	https://www.assetstore.unity3d.com/en/#!/content/10374

MarkUX	
Waardering	5 (16)
Prijs	\$30 (per seat)
Unity versie	4.6.0 of hoger
Datum	26 juni 2015 – 16 oktober 2015
Omschrijving	MarkUX is een relatief nieuw UI framework voor Unity. Achter de schermen maakt MarkUX gebruik van de nieuwe Unity UI API. De UI wordt gecreëerd in de vorm van XML bestanden en kan door middel van data bindings met de applicatie logica communiceren. MarkUX biedt een tal van verschillende standaard UI elementen en deze kunnen gemakkelijk worden uitgebreid.
URL	https://www.assetstore.unity3d.com/en/#!/content/37466

11.1.1 Plugins

PanelManager	
Waardering	4 (14)
Prijs	\$20 (per seat)
Unity versie	4.6.1 of hoger
Datum	25 september 2014 – 18 mei 2015
Omschrijving	Is een plugin die helpt bij het beheren van verschillende windows (ook wel panels). Hierdoor wordt het gemakkelijker om verschillende panelen te laten zien en transitie tussen panelen te schrijven.
URL	https://www.assetstore.unity3d.com/en/#!/content/21867

HAUTE TECHNIQUE

EnhancedScroller	
Waardering	5 (26)
Prijs	\$15 (per seat)
Unity versie	5.1.0 of hoger
Datum	28 mei 2015 – 12 oktober 2015
Omschrijving	Is een plugin die het tonen van data in lijsten gemakkelijk maakt. Het heeft tal van optimalisaties doorgevoerd zodat de lijst met grote datasets overweg kan. Achter de schermen maakt deze plugin gebruik van de Unity UI.
URL	https://www.assetstore.unity3d.com/en/#!/content/36378

ListView for Unity UI	
Waardering	4 (21)
Prijs	\$15
Unity versie	5.2.0 of hoger
Datum	10 september 2014 – 16 oktober 2015
Omschrijving	ListView for Unity UI is een plugin die het creëren van een lijst met data gemakkelijker maakt. De plugin maakt gebruik van de nieuwe Unity UI. Verder is de plugin voor een groot deel code compatible met de windows forms list view waardoor (als je hier al ervaring mee hebt) je de listview snel kunt gebruiken.
URL	https://www.assetstore.unity3d.com/en/#!/content/21430

Data Bind for Unity	
Waardering	5 (12)
Prijs	\$20
Unity versie	4.6.1 of hoger
Datum	6 februari 2015 – 27 juli 2015
Omschrijving	Data Bind is een plugin die een architectuur biedt die het gemakkelijk maakt om je game logica te scheiden van je presentatie code. Dit moet uiteindelijk leiden tot beter te onderhouden code.
URL	https://www.assetstore.unity3d.com/en/#!/content/36378

11.2 Keuzen voor framework

Binnen Unity zijn er twee verschillende manieren voor het creëren van een UI. De eerste methode is sinds de beginversies binnen Unity aanwezig. Deze methode is echter lastig in gebruik en zorgt voor een afname in performance (doordat er meerdere draw calls worden gemaakt). Er zijn toentertijd verschillende UI frameworks geschreven die het creëren van een UI makkelijker en efficiënter maakte waaronder de beschreven: NGUI en EZ GUI. Unity heeft in versie 4.6 een nieuw UI systeem geïntroduceerd met behulp van de ontwikkelaar van NGUI. Dit systeem is gebaseerd op NGUI en biedt tal van verbeteringen ten opzichte van de oude methode. Dit is dan ook de methode die door veel van de nieuwere assets wordt gebruikt als basis.

Er zijn een aantal verschillende frameworks gevonden elk met hun eigen voordelen en nadelen. Er zit echter een groot verschil in de kwaliteit en uitbreidbaarheid tussen de verschillende gevonden frameworks. Hier volgt een lijst met de reden waarom het desbetreffende framework niet is gekozen.

- NGUI: De nieuwe Unity UI is gebaseerd op dit framework. NGUI biedt echter niet genoeg toegevoegde waarde voor de prijs boven op het nieuwe UI systeem van Unity.
- EZ GUI: Dit is een verouderd framework. Het is gecreëerd als oplossing toen de UI oplossing van Unity nog slecht was. Nu Unity een nieuwe UI systeem heeft gecreëerd is EZ GUI geen goede kandidaat meer.
- New UI Widgets: Biedt een tal van highlevel componenten zoals tree views en dialogs. Deze componenten zijn echter slecht uitbreidbaar. Ook ligt de prijs vergeleken met de andere frameworks aan de hoge kant voor wat het framework biedt.
- One Minute GUI: Focust zich op het snel creëren van een game UI met elementen die een standaard lay-out hebben. Het focust zich niet in het aanbieden van een set van uitbreidbare componenten wat wel gewenst is voor het te kiezen framework.
- PowerUI - HTML/CSS: Maakt gebruik van HTML, CSS en Javascript voor het tonen van een UI. Dit framework maakt dus geen gebruik van de nieuwe Unity UI. Dit framework valt af mede omdat de documentatie veel te wensen overlaat en omdat het geen gebruik maakt van het nieuwe UI framework van Unity.
- NoesisGUI 1.1: Is het duurste framework. Het biedt echter ook tal van opties. Het gebruik van dit framework is echter overkill voor wat er (op dit moment) gepland staat voor de UI.

In MarkUX schrijf je net als in NoesisGUI de UI in XML. MarkUX vertaalt de XML naar voor Unity te gebruiken UI elementen. Het framework biedt tal van standaard componenten die gemakkelijk zijn uit te breiden. Verder heeft MarkUX ondersteuning voor thema's waardoor het gemakkelijk is om de weergave van elementen aan te passen. De licentiekosten voor MarkUX bedragen 30 dollar, dit is ten opzichte van de andere frameworks relatief goedkoop.

MarkUX is dus een van de betere frameworks om te gebruiken voor het creëren van de UI voor de applicatie. Er is echter gekozen om op dit moment nog geen gebruik te maken van het framework. Dit zou in een latere sprint nog wel kunnen. Voor nu wordt er gewerkt met de Unity UI voor het creëren van de verschillende elementen.

Bijlage F

Scrumrapport

HAUTE TECHNIQUE

Scrum Rapport

Auteur	: Jan de Boer
Bedrijf	: HAUTE TECHNIQUE
Opdrachtgever	: Sander ter Braak
Datum	: 4 september 2015
Versie	: 1.0

Versiebeheer

Versie	Datum	Opmerking
0.1	11-09-2015	Eerste versie.
0.2	10-10-2015	Opsturen eerste concept naar afstudeerbegeleider.
0.3	13-10-2015	Ontwerpen gesplitst van scrumrapport. Ontwerpen bevinden zich nu in eigen ontwerprapport.
0.4	28-10-2015	De taken van de verschillende sprints verwijderd uit het document. Verder de structuur van het document aangepast naar een duidelijkere vorm.
0.5	06-11-2015	Opgestuurd naar begeleiders.
1.0	05-01-2016	Verschillende wijzigingen in de opmaak doorgevoerd.

Inhoudsopgave

1	Introductie.....	6
2	Sprint 1 – Analyse	7
2.1	Sprint planning	7
2.2	Backlog toevoegingen	7
2.3	Sprint evaluatie.....	8
3	Sprint 2 – Architectuur	9
3.1	Sprint planning	9
3.2	Backlog toevoeging	9
3.3	Sprint evaluatie.....	10
4	Sprint 3 – Effecten.....	11
4.1	Sprint planning	11
4.2	Backlog toevoeging	11
4.3	Sprint evaluatie.....	11
5	Sprint 4 – User interface.....	12
5.1	Sprint planning	12
5.2	Backlog toevoeging	12
5.2.1	Lampen	13
5.2.2	Effecten	13
5.2.3	Tracking	13
5.2.4	Opnamen.....	14
5.2.5	Instellingen	14
5.2.6	DMX	14
5.2.7	Camera	14
5.2.8	Testdag	14
5.3	Sprint evaluatie.....	15
6	Sprint 5 – Documentatie.....	16
6.1	Sprint planning	16
6.2	Backlog toevoeging	16
6.3	Sprint evaluatie.....	16
7	Sprint 6 – Tracking devices	17
7.1	Sprint planning	17
7.2	Backlog toevoegingen	17
7.3	Sprint evaluatie.....	17

8	Sprint 7 – Apparaten aanpasbaar.....	18
8.1	Sprint planning	18
8.2	Backlog toevoegingen	18
8.3	Sprint evaluatie.....	19
9	Sprint 8 – Koppeling	20
9.1	Sprint planning	20
9.2	Backlog toevoegingen	20
9.3	Sprint evaluatie.....	20
10	Sprint 9.....	21
10.1	Sprint planning	21
10.2	Backlog toevoegingen	21
10.3	Sprint evaluatie.....	21
11	Productbacklog.....	22

Tabellen

Tabel 1	Sprint planning sprint 1	7
Tabel 2	Backlog toevoegingen sprint 1	7
Tabel 3	Sprint planning sprint 2	9
Tabel 4	Backlog toevoegingen sprint 2	9
Tabel 5	Sprint planning sprint 3	11
Tabel 6	Backlog toevoegingen sprint 3	11
Tabel 7	Sprint planning sprint 4	12
Tabel 8	Backlog toevoegingen sprint 4	12
Tabel 9	Sprint planning sprint 5	16
Tabel 10	Backlog toevoegingen sprint 5	16
Tabel 11	Sprint planning sprint 6	17
Tabel 12	Backlog toevoegingen sprint 6	17
Tabel 13	Sprint planning sprint 7	18
Tabel 14	Backlog toevoegingen sprint 7	19
Tabel 15	Sprint planning sprint 8	20
Tabel 16	Backlog toevoegingen sprint 8	20
Tabel 17	Sprint planning sprint 9	21
Tabel 18	Backlog toevoegingen sprint 9	21
Tabel 19	Productbacklog.....	22

1 Introductie

In dit document bevindt zich de productbacklog en de sprint planningen van de verschillende sprints die tijdens het project zijn uitgevoerd. Het doel van dit document is om een duidelijk overzicht te creëren van het verloop van de verschillende sprints. Tevens is dit document een naslagwerk voor de opdrachtgever om te zien wat de status is van de verschillende user stories in de product backlog.

Elke sprint wordt behandeld in een eigen hoofdstuk met daarin de sprint planning, de user stories die tijdens de sprint zijn toegevoegd aan de productbacklog en een evaluatie van de sprint. De sprint planning laat zien welke user stories in de sprint zijn geïmplementeerd. Aan het begin van een sprint is er ingeschat hoeveel tijd (in uren) het implementeren van de user stories zou kosten. Mocht het implementeren van een user story meer tijd kosten dan was ingeschat dan wordt dat in de des betreffende sprint toegelicht. Per sprint worden ook de toegevoegde backlog items behandeld.

Tijdens elke sprint is er ook tijd genomen om te werken aan het afstudeerdossier; deze tijd is niet weergegeven per sprint maar het komt er op neer dat er elke sprint ongeveer twaalf uur aan het afstudeer dossier is gewerkt.

Aan het eind van dit document is een totale product backlog van het project te vinden. Hierin zijn de verschillende user stories van het project te vinden met daarbij in welke sprint ze zijn gecreëerd en geïmplementeerd.

2 Sprint 1 – Analyse

2.1 Sprint planning

De eerste week van het project werd gebruikt om het project op te starten. De nieuwe opdracht werd vastgesteld en deze werd verder beschreven in het plan van aanpak. Aangezien de keuze voor scrum in de eerste week is gemaakt, is deze week niet meegenomen in de sprint planning. De eerste sprint startte de tweede week van het project. Deze sprint had een lengte van 1 week en stond in het teken van het helder krijgen van de eisen van de te maken software. Ook werden verschillende technieken die binnen het project werden gebruikt nader uitgezocht.

Tabel 1 Sprint planning sprint 1

Nr.	Story	Uren
1	Als product owner wil ik de globale eisen van het te maken software systeem duidelijk hebben gedefinieerd.	7
2	Als ontwikkelaar wil ik weten hoe volglampen werken en hoe deze moeten worden aangestuurd.	12
3	Als ontwikkelaar wil ik de werking van het openFrameworks framework analyseren.	7
4	Als opdrachtgever wil ik een rapport die de hardware beschrijft en test.	7

2.2 Backlog toevoegingen

Nadat de globale eisen van het systeem in samenspraak met de opdrachtgever waren beschreven konden deze worden vertaald naar user stories.

Tabel 2 Backlog toevoegingen sprint 1

Nr.	Story	Herkomst
5	Als gebruiker wil ik meerdere lampen van mogelijk verschillende typen aan het systeem kunnen toevoegen.	Requirements Document
6	Als gebruiker wil ik dat lampen worden aangestuurd volgens een bepaald effect.	Requirements Document
7	Als gebruiker wil ik de mogelijkheid hebben om de positie en rotatie van een moving head semi automatisch te laten bepalen door de applicatie.	Requirements Document
8	Als gebruiker wil ik de verschillende lampen in een 3D omgeving zien.	Requirements Document
9	Als gebruiker wil ik zien waar de verschillende tracking devices zich bevinden in de 3D ruimte.	Requirements Document
10	Als gebruiker wil ik zien waar de verschillende antennes zich bevinden in de 3D ruimte.	Requirements Document
11	Als gebruiker wil ik de mogelijkheid hebben om tracking devices te kunnen toevoegen.	Requirements Document
12	Als gebruiker wil ik dat het systeem ook videobeelden kan aan aansturen.	Requirements Document
13	Als gebruiker wil ik de mogelijkheid om de locatie van een of meerdere tracking devices op te nemen zodat deze later weer kunnen worden afgespeeld.	Requirements Document

HAUTE TECHNIQUE

14	Als gebruiker wil ik de mogelijkheid om de opgenomen locatie data van een of meerdere tracking devices weer af te spelen.	Requirements Document
15	Als ontwikkelaar wil ik dat de locatie data die binnenkomt kan worden gesimuleerd.	Requirements Document
16	Als gebruiker wil ik de huidige configuratie (positie en rotatie van lampen, toegevoegde tracking devices, actieve effecten) kunnen exporteren.	Requirements Document
17	Als gebruiker wil ik de huidige configuratie (positie en rotatie van lampen, toegevoegde tracking devices, actieve effecten) kunnen importeren.	Requirements Document

2.3 Sprint evaluatie

Het doel van deze sprint was om de eisen van het systeem helder te krijgen en de verschillende technieken te onderzoeken en dit is gelukt. Tijdens deze sprint is naar voren gekomen dat er een onderzoek nodig is naar betere alternatieven voor openFrameworks. openFrameworks is mogelijk niet de beste keuze als het gaat om 3D werelden. In sprint twee moet hier nader onderzoek naar worden gedaan. Verder zijn in deze sprint verschillende risico's van het project geëlimineerd. Zo is er een onderzoek gedaan naar moving heads en het DMX protocol dat het aansturen van moving heads mogelijk maakt.

3 Sprint 2 – Architectuur

3.1 Sprint planning

Deze sprint moest er een keuze worden gemaakt voor het framework dat gebruikt zou worden voor de applicatie. Hiernaast moest de architectuur van de applicatie worden bepaald. De basis die in deze sprint wordt gecreëerd kan in de volgende sprints waar nodig worden uitgebreid.

Het is belangrijk om op te merken dat er in deze sprint nog niet werd gewerkt aan een user interface. In een vervolg sprint word er aan de user interface gewerkt en zullen er user stories ontstaan als: als gebruiker wil ik de mogelijkheid om een lamp toe te voegen via een user interface.

Tabel 3 Sprint planning sprint 2

Nr.	Story	Uren
20	Als ontwikkelaar wil ik onderzoeken welk framework het beste is geschikt om de applicatie in te ontwikkelen.	21
21	Als ontwikkelaar wil ik een basis ontwerp van de applicatie creëren om vandaaruit verder te ontwikkelen.	14
5	Als gebruiker wil ik meerdere lampen van mogelijk verschillende typen aan het systeem kunnen toevoegen.	7
8	Als gebruiker wil ik de verschillende lampen in een 3D omgeving zien.	7
6	Als gebruiker wil ik dat lampen worden aangestuurd volgens een bepaald effect.	7
9	Als gebruiker wil ik zien waar de verschillende tracking devices zich bevinden in de 3D ruimte.	1
11	Als gebruiker wil ik de mogelijkheid hebben om tracking devices te kunnen toevoegen.	3
13	Als gebruiker wil ik de mogelijkheid om de locatie van een of meerdere tracking devices op te nemen zodat deze later weer kan worden afgespeeld.	3
14	Als gebruiker wil ik de mogelijkheid om de opgenomen locatie data van een of meerdere tracking devices weer af te spelen.	4
15	Als ontwikkelaar wil ik dat de locatie data die binnenkomt kan worden gesimuleerd.	2

3.2 Backlog toevoeging

Er zijn veel verschillende manieren om een lamp of moving head weer te geven in de 3D wereld. Deze sprint is er gekozen om lampen en moving heads weer te geven als 3D kubussen. Er is echter voor te stellen dat de gebruiker in de toekomst 'echte' 3D modellen van lampen en moving heads wil. Er zijn daarom een tweetal user stories bijgekomen.

Deze sprint is de bestaande applicatie die de opdrachtgever gebruikt tijdens de shows van Armin van Buuren ook nader geanalyseerd. Hierbij zijn een drietal nieuwe user stories ontstaan.

Tabel 4 Backlog toevoegingen sprint 2

Nr.	Story	Herkomst
22	Als gebruiker wil ik de mogelijkheid om een 3 ^e party controller te kunnen koppelen met de applicatie waardoor het mogelijk wordt om de parameters van effecten te kunnen aanpassen.	Myo applicatie

HAUTE TECHNIQUE

23	Als gebruiker wil ik dat de applicatie gebruik maakt van een echt 3D model van een moving head in plaats van een kubus.	Sprint 2
24	Als gebruiker wil ik dat de applicatie gebruik maakt van een echt 3D model van een lamp in plaats van een kubus.	Sprint 2
18	Als gebruiker wil ik een lamp met de daarbij horende DMX kanalen automatisch laten inladen door gebruik te maken van het des betreffende XML bestand ¹ .	Myo applicatie
19	Als gebruiker wil ik de mogelijkheid hebben om meerdere lampen tegelijkertijd te kunnen toevoegen door middel van een XML bestand waarin de lampen staan beschreven.	Myo applicatie

Notitie (1): De opdrachtgever beschikt over verschillende XML bestanden waarin de meest gebruikte lampen worden beschreven. In deze XML bestanden staat onder andere de volgende informatie: naam van de fabriek, type naam van de lamp, de verschillende kanalen die de lamp gebruikt, de standaard waarden voor de kanalen en de naam van het kanaal. De XML bestanden waarover de opdrachtgever beschikt zijn geëxporteerd uit verschillende lichttafels.

3.3 Sprint evaluatie

Deze sprint had als doel om een keuze te maken voor een framework dat zou worden gebruikt voor het ontwikkelen van de applicatie. Hiernaast moest de basisstructuur van de applicatie ontworpen worden.

Deze sprint heeft een opzet gemaakt voor het aansturen van lampen volgens een bepaald effect maar dit onderdeel van de applicatie moet nog verder worden uitgebreid. Zo is het op dit moment niet mogelijk om gemakkelijk nieuwe effecten toe te voegen. De opdrachtgever heeft aangegeven dat dit een belangrijke wens is. De volgende sprint zal zich hierop gaan focussen.

4 Sprint 3 – Effecten

4.1 Sprint planning

Deze sprint zal zich gaan focussen op het effect gedeelte van de applicatie. Het doel van deze sprint is het creëren van een systeem dat het toelaat om gemakkelijk nieuwe effecten aan het systeem toe te voegen. Dit zal in deze sprint nog niet via een user interface gaan om dezelfde reden die wordt aan gegeven in sprint 2. Wel wordt er rekening gehouden met een user interface.

Tabel 5 Sprint planning sprint 3

Nr.	Story	Uren
25	Als ontwikkelaar wil ik onderzoeken wat een effect precies inhoudt.	8
26	Als ontwikkelaar wil ik onderzoeken welke scriptaal het meest geschikt is om te gebruiken voor het schrijven van effecten.	21
27	Als gebruiker wil ik effecten kunnen toevoegen aan een lamp.	12
28	Als gebruiker wil ik een effect kunnen verwijderen van een lamp.	2
29	Als effect schrijver wil ik een effect script kunnen toevoegen aan de applicatie.	7
30	Als effect schrijver wil ik een effect script kunnen verwijderen uit de applicatie.	2
31	Als student wil ik het bedrijfsbezoek voorbereiden.	2

4.2 Backlog toevoeging

Tijdens het verloop van deze sprint zijn er een aantal effect gerelateerde user stories ontstaan. Aangezien het aan het einde van deze sprint mogelijk was om effect scripts toe te voegen is er een nieuwe rol ontstaan. Dit is de effect schrijver. Dit is iemand (de naam zegt het al) die effect scripts voor het systeem schrijft.

Tabel 6 Backlog toevoegingen sprint 3

Nr.	Story	Herkomst
32	Als effect schrijver wil ik een effect script 'on the fly' kunnen aanpassen of toevoegen.	Sprint3
33	Als effect schrijver wil ik een handleiding over hoe ik een effect kan schrijven voor de applicatie.	Sprint3
34	Als effect schrijver wil ik de status van het compileren van mijn script zien binnen de applicatie.	Sprint3
35	Als effect schrijver wil ik de status van het compileren kunnen zien binnen het programma waarmee ik het effect schrijf.	Sprint3
36	Als effect schrijver wil ik autocorrectie op het moment dat ik effecten aan het schrijven ben.	Sprint3

4.3 Sprint evaluatie

In deze sprint is er een keuze gemaakt om C# als scripttaal te gebruiken voor het schrijven van effecten. Hiernaast is er een uitbreidbaar systeem opgezet waardoor het mogelijk wordt om tal van verschillende typen effecten aan het systeem te kunnen toevoegen. Hierbij is ook rekening gehouden met de toekomstige user interface.

5 Sprint 4 – User interface

5.1 Sprint planning

In deze sprint was het tijd om te onderzoeken hoe er binnen Unity het beste een user interface kon worden gecreëerd. Hiervoor moest eerst worden bepaald wat de verschillende functies van de user interface zouden worden. Tevens werd er tijdens deze sprint een dag meegelopen met de opdrachtgever bij het opzetten en inregelen van de Myo applicatie tijdens een show van Armin.

Tabel 7 Sprint planning sprint 4

Nr.	Story	Uren
37	Als ontwikkelaar wil ik onderzoeken welke functionaliteit de user interface krijgt.	14
38	Als ontwikkelaar wil ik de verschillende mogelijkheden onderzoeken voor het tonen van een user interface binnen Unity.	28
39	Als ontwikkelaar wil ik een dag meelopen met de opdrachtgever om te kijken in wat voor soort situatie de applicatie later zou kunnen worden ingezet.	8

De tweede week van de sprint werd duidelijk dat de eerstkomende maandag een testdag met Sendrato kon plaatsvinden. Na een gesprek met de opdrachtgever over wat het doel van de testdag was werd duidelijk dat de applicatie nog verder moest worden uitgebreid met ondersteuning voor het inlezen van XML bestanden met daarin de configuratie van lampen. Er is toen besloten om de taken die nog stonden gepland voor deze sprint (dit waren vooral beschrijvende taken) door te schuiven naar de volgende sprint om zo plaats te maken voor de nieuwe user story. Deze zijn hieronder te lezen. Er is ook besloten om de sprint met twee dagen te verlengen zodat er tijd beschikbaar werd gemaakt om de resultaten van de testdag te beschrijven.

18	Als gebruiker wil ik een lamp met de daarbij horende DMX kanalen automatisch laten inladen door gebruik te maken van het desbetreffende XML bestand (zie notitie in sprint 2 voor extra uitleg).	14
43	Als ontwikkelaar wil ik de eerste testdag met Sendrato bijwonen.	7
44	Als ontwikkelaar wil ik de resultaten van de eerste testdag beschrijven.	4

5.2 Backlog toevoeging

Deze sprint is er een document geschreven die de verschillende functies van de user interface beschrijft. In dit document zijn een tal van nieuwe user stories te vinden; deze zijn hieronder nader beschreven en gecategoriseerd. Tevens zijn er in de loop van sprint 4 en tijdens de testdag nieuwe user stories ontstaan, deze zijn ook hieronder te vinden.

Tabel 8 Backlog toevoegingen sprint 4

Nr.	Story	Herkomst
40	Als gebruiker wil ik een manier om het podium in te laden in de 3D wereld. Dus dat bijvoorbeeld het podium (met hoogte verschillen) zichtbaar wordt in de 3D wereld.	Sprint 4
42	Als gebruiker wil ik de manier hoe een tracking device wordt geüpdatet (bijvoorbeeld door een muis of door een tracker) kunnen veranderen zonder dat ik hiervoor de effecten opnieuw moet instellen die aan het tracking device waren gekoppeld.	Sprint 4

41	Als gebruiker wil ik dat de teksten binnen de applicatie in mijn eigen taal zijn.	Sprint 4
-----------	---	----------

5.2.1 Lampen

45	Als gebruiker wil ik de mogelijkheid om een lamp via de UI aan de applicatie toe te voegen.	UI Document
46	Als gebruiker wil ik de mogelijkheid om een lamp uit de applicatie te kunnen verwijderen via de UI.	UI Document
47	Als gebruiker wil ik de mogelijkheid om de verschillende instellingen van een lamp (zoals bijvoorbeeld de positie en de rotatie) te kunnen aanpassen via de UI.	UI Document

5.2.2 Effecten

48	Als gebruiker wil ik de mogelijkheid om de verschillende beschikbare effecten te kunnen inzien.	UI Document
49	Als gebruiker wil ik de mogelijkheid hebben om een effect te creëren en te koppelen aan verschillende lampen en tracking devices via de UI.	UI Document
50	Als gebruiker wil ik kunnen zien welke effecten actief zijn (op welke lamp(en) en of tracking device(s) binnen de applicatie.	UI Document
51	Als gebruiker wil ik de mogelijkheid hebben om effecten te kunnen starten of te kunnen stoppen via de UI	UI Document
52	Als gebruiker wil ik de mogelijkheid hebben om effecten te kunnen verwijderen via de UI.	UI Document
53	Als gebruiker wil ik de verschillende parameters van een actief effect kunnen inzien en kunnen veranderen.	UI Document
54	Als gebruiker wil ik de verschillende parameters van een actief effect kunnen koppelen met een 3d party MIDI controller via de UI.	UI Document
55	Als gebruiker wil ik de mogelijkheid hebben om de verschillende parameters van een actief effect te kunnen koppelen aan een 3 ^e party controller via de UI.	UI Document

5.2.3 Tracking

56	Als gebruiker wil ik een tracking device kunnen toevoegen aan de applicatie via de UI.	UI Document
57	Als gebruiker wil ik een tracking device kunnen verwijderen van de applicatie via de UI.	UI Document
58	Als gebruiker wil ik een overzicht hebben van de verschillende tracking devices binnen de applicatie.	UI Document
59	Als gebruiker wil ik de input methode van een tracking device binnen de UI kunnen veranderen.	UI Document

HAUTE TECHNIQUE

5.2.4 Opnamen

60	Als gebruiker wil ik de mogelijkheid om een opname te starten binnen de UI.	UI Document
61	Als gebruiker wil ik een opname kunnen starten vanuit een lijst van opnamen.	UI Document
62	Als gebruiker wil ik de status van een opname (start, stop of pauzeren) kunnen veranderen binnen de UI.	UI Document
63	Als gebruiker wil ik de mogelijkheid om een opname te kunnen verwijderen binnen de UI.	UI Document

5.2.5 Instellingen

64	Als gebruiker wil ik de verschillende instellingen van de applicatie kunnen inzien en veranderen binnen de UI.	UI Document
----	--	-------------

5.2.6 DMX

65	Als gebruiker wil ik een DMX node kunnen toevoegen via de UI.	UI Document
66	Als gebruiker wil ik een DMX universe kunnen toevoegen aan de applicatie via de UI.	UI Document
67	Als gebruiker wil ik een DMX universe kunnen koppelen aan een DMX node via de UI.	UI Document
68	Als gebruiker wil ik de mogelijkheid hebben om een DMX node te verwijderen via de UI.	UI Document
69	Als gebruiker wil ik de mogelijkheid om een DMX universe te verwijderen via de UI.	UI Document
70	Als gebruiker wil ik dat de applicatie automatisch zoekt naar ArtNet nodes en deze laat zien.	UI Document
71	Als gebruiker wil ik dat DMX universes niet worden verwijderd als er een DMX node wordt verwijderd.	UI Document

5.2.7 Camera

72	Als ontwikkelaar wil ik de verschillende opties voor het aansturen van de applicatie onderzoeken, hierbij vooral kijkend hoe de camera zich gedraagt en hoe de camera zou kunnen worden aangestuurd binnen de applicatie.	UI Document
----	---	-------------

5.2.8 Testdag

73	Als gebruiker wil ik een grid zien waarover de verschillende tracking devices zich bewegen.	Testdag #1
74	Als gebruiker wil ik dat ik snel het systeem van Sendrato kan koppelen met de applicatie ongeacht de opstelling van de hardware.	Testdag #1
75	Als gebruiker wil ik inzicht in verschillende parameters van de applicatie zoals: het aantal frames per seconde en het aantal locatie updates per seconde.	Testdag #1

5.3 Sprint evaluatie

Deze sprint is er onderzoek gedaan naar de user interface van de applicatie. Er is gekeken naar de verschillende functionaliteiten van de applicatie; hieruit zijn een tal van nieuwe user stories ontstaan. Hiernaast is er onderzocht hoe de UI in Unity het beste kan worden gecreëerd en geïmplementeerd. Hiervoor zijn verschillende UI frameworks onderzocht. Er is uiteindelijk gekozen om de UI te bouwen met het UI framework dat Unity standaard meelevert.

Tevens is er deze sprint een testdag met Sendrato geweest. Hieruit zijn een aantal nieuwe user stories ontstaan die zijn toegevoegd aan de backlog. Een aantal geplande werkzaamheden zijn door deze testdag doorgeschoven naar de volgende sprint. Dit zijn vooral de beschrijvende taken dus het bijwerken van het afstudeerverslag en het verder uitwerken van het UI document.

6 Sprint 5 – Documentatie

6.1 Sprint planning

In deze sprint wordt user story 38 uit de vorige sprint afgemaakt. Deze story kon vanwege de testdag niet worden afgemaakt. Verder richt deze sprint zich op het aanvullen van het hardware test rapport met de resultaten van de testdag. Hiernaast worden verschillende documenten die belangrijk zijn voor het afstudeerdossier aangevuld en aangepast waar nodig; het gaat hierbij om het afstudeerverslag en het scrum rapport. Deze sprint wordt er namelijk een eerste concept versie van het afstudeerdossier naar de begeleidende docenten gestuurd. De vorige sprint liep door de testdag twee dagen uit. Hierdoor zijn er in deze sprint twee dagen minder beschikbaar om in te plannen.

Tabel 9 Sprint planning sprint 5

Nr.	Story	Uren
38	Als ontwikkelaar wil ik de verschillende mogelijkheden onderzoeken voor het tonen van een user interface binnen Unity.	12
76	Als opdrachtgever wil ik dat de testdag wordt verwerkt in het hardware rapport.	7

Het schrijven van de documentatie ging sneller dan verwacht, hierdoor was de sprint eerder klaar dan gepland. Er is toen besloten om alvast een aantal user stories die gepland stonden voor de volgende sprint te implementeren. Deze user stories zijn hieronder te vinden.

58	Als gebruiker wil ik een overzicht hebben van de verschillende tracking devices binnen de applicatie.	10
75	Als gebruiker wil ik inzicht in verschillende parameters van de applicatie zoals: het aantal frames per seconde en het aantal locatie updates per seconde.	4

6.2 Backlog toevoeging

Er zijn deze sprint geen backlog toevoegingen gedaan.

Tabel 10 Backlog toevoegingen sprint 5

Nr.	Story	Herkomst
-	-	-

6.3 Sprint evaluatie

De sprint was begonnen met het afmaken van user story 38 die door de testdag in de vorige sprint niet kon worden afgemaakt. Ook zijn de resultaten van de testdag verwerkt in het hardware testrapport. Verder is de documentatie van het project verder uitgebreid en als conceptversie naar de begeleidende docenten opgestuurd.

Het bijwerken van de documentatie ging sneller dan verwacht, hierdoor bleef er tijd over in de sprint. Deze tijd is opgevuld met twee user stories die stonden gepland voor de volgende sprint. Dit waren user stories die betrekking hadden op de user interface. In samenwerking met MarkUx is de eerste versie van de GUI tot stand gekomen.

7 Sprint 6 – User interface implementeren

7.1 Sprint planning

Deze sprint staat in het teken van het implementeren van de bevindingen van de testdag. Zo wordt er gewerkt aan een manier om de antennes binnen de applicatie weer te geven en de opzettijd van de applicatie te versnellen. Tevens wordt er deze sprint voor het eerst 'echt' gewerkt met het UI framework MarkUX dat in Sprint 4 is aangeschaft. In de vorige sprint was al een begin gemaakt met het creëren van een UI voor de tracking devices; deze werd deze sprint verder afgemaakt.

Tabel 11 Sprint planning sprint 6

Nr.	Story	Uren
10	Als gebruiker wil ik zien waar de verschillende antennes zich bevinden in de 3D ruimte.	14
73	Als gebruiker wil ik een grid zien waarover de verschillende tracking devices zich bewegen.	4
74	Als gebruiker wil ik dat ik snel het systeem van Sendrato kan koppelen met de applicatie ongeacht de opstelling van de hardware.	10
42	Als gebruiker wil ik dat de manier hoe een tracking device wordt geüpdatet (bijvoorbeeld door een muis of door een tracker) kan veranderen zonder dat ik hiervoor de effecten opnieuw moet instellen die aan het tracking device zijn gekoppeld.	20
56	Als gebruiker wil ik een tracking device kunnen toevoegen aan de applicatie via de UI.	4
58	Als gebruiker wil ik een overzicht hebben van de verschillende tracking devices binnen de applicatie.	2
59	Als gebruiker wil ik de input methode van een tracking device binnen de UI kunnen veranderen.	8

Het implementeren van de bovenstaande user stories ging sneller dan verwacht. Dit kwam deels doordat er extra tijd was ingeschat voor het maken van de UI aangezien dit onderdeel nieuw was. Maar deze tijd bleek niet nodig te zijn. Daardoor bleef er extra tijd over. Er is gekozen om een handleiding te schrijven voor het ontwikkelen van een effect.

33	Als effect schrijver wil ik een handleiding over hoe ik een effect kan schrijven voor de applicatie.	10
----	--	----

7.2 Backlog toevoegingen

Er zijn deze sprint geen backlog toevoegingen gedaan.

Tabel 12 Backlog toevoegingen sprint 6

Nr.	Story	Herkomst
-	-	-

7.3 Sprint evaluatie

Deze sprint zijn de verschillende user stories die gepland zijn geïmplementeerd. Er bleef tevens tijd over om een handleiding te schrijven voor het ontwikkelen van een effect.

8 Sprint 7 – Apparaat aanpassen

8.1 Sprint planning

Deze sprint stond in het teken van het aanpasbaar maken van de positie en of rotatie van een lamp door de gebruiker. Al vanaf Sprint 1 bleek dat deze wens er was. Na de eerste testdag werd deze wens verder versterkt. Er is daarom besloten om deze functionaliteit in deze sprint te implementeren. De user story die deze functionaliteit beschreef was reeds beschreven in Sprint 4 in het User Interface document. Zie de tabel hieronder voor de desbetreffende user story.

47	Als gebruiker wil ik de mogelijkheid om de verschillende instellingen van een lamp (zoals bijvoorbeeld de positie en de rotatie) te kunnen aanpassen via de UI.	UI Document
-----------	---	-------------

De bovenstaande user story is ten tijde van schrijven globaal opgeschreven. Met de kennis die er nu is kan de story specifieker worden beschreven. Hierbij is de story in vier verschillende (sub) stories verdeeld.

79	Als gebruiker wil ik vanuit de applicatie de rotatie van een apparaat aanpassen.	User Story 47
80	Als gebruiker wil ik vanuit de applicatie de positie van een apparaat aanpassen.	User Story 47
81	Als gebruiker wil ik vanuit de applicatie de DMX instellingen van een lamp veranderen.	User Story 47
82	Als gebruiker wil ik dat de instellingen van een apparaat worden opgeslagen zodat ik handmatige bijstellingen niet telkens opnieuw hoeft te doen.	User Story 47

Het aanpassen van de DMX instellingen en het opslaan van de instellingen had een lagere prioriteit dan het aanpasbaar maken van de positie en of rotatie van een lamp. Dit omdat het aanpassen van DMX instellingen niet vaak voorkwam en het opslaan van de instellingen meer betrekking had op het configuratie sub onderdeel. In Sprint 3 is reeds aangegeven dat er niet op dit onderdeel wordt gefocust. Daarom zal er deze sprint worden gefocust op het aanpassen van de positie en of rotatie van een lamp. Deze sprint vindt ook het tussentijds assessment plaats, er is in deze sprint daarom meer tijd voor het maken en bijwerken van de documentatie.

Tabel 13 Sprint planning sprint 7

Nr.	Story	Uren
77	Als effectschrijver wil ik dat als ik de pan en tilt van een moving head in de code aanpas dit ook zichtbaar wordt in de 3D wereld (fix).	4
79	Als gebruiker wil ik vanuit de applicatie de rotatie van een apparaat aanpassen.	20
80	Als gebruiker wil ik vanuit de applicatie de positie van een apparaat aanpassen.	20

8.2 Backlog toevoegingen

Tijdens deze sprint zijn de volgende backlog toevoegingen gedaan. Deze user stories zijn naar voren gekomen tijdens het implementeren van de verschillende functionaliteiten die voor de sprint stonden gepland.

Tabel 14 Backlog toevoegingen sprint 7

Nr.	Story	Herkomst
84	Als gebruiker wil ik dat als ik op een apparaat klik een dashboard te zien krijg met daarin informatie over de lamp (type apparaat, positie, rotatie, DMX universe).	Sprint 7
85	Als gebruiker wil ik de positie of rotatie van meerdere apparaten te gelijktijd kunnen aanpassen.	Sprint 7

8.3 Sprint evaluatie

Voor deze sprint stond er gepland om het mogelijk te maken om de positie en rotatie van een lamp te kunnen aanpassen vanuit de applicatie. De tijd die de verschillende user stories kostte was echter onderschat waardoor niet alle gewenste functies van het aanpassen van de positie en rotatie van een lamp konden worden geïmplementeerd.

Het implementeren van user story 77 kostte meer tijd dan verwacht. Dit kwam doordat het implementeren van een algoritme dat vanuit een gegeven pan en tilt de lamp in de 3D wereld op de juiste rotatie zette langer duurde. Verder werd er tijdens het inplannen van deze sprint van uitgegaan dat het aanpassen van de positie en rotatie van een object, (lamp) zoals dat binnen de Unity Editor werkt al eerder zou zijn gedaan. Dit bleek echter niet het geval, hierdoor moest de functionaliteit vanaf de grond af worden geïmplementeerd. User story 79 en 80 zijn daarom doorgeschoven naar de volgende sprint.

9 Sprint 8 – Loskoppelen UI laag

9.1 Sprint planning

In de vorige sprint is niet gelukt om alle geplande user stories te implementeren. Deze user stories zijn dan ook doorgeschoven naar deze sprint. Het gaat hier om user story 79 en 80. Verder wordt er deze sprint gekeken hoe de koppeling tussen de UI (view) package en de rest van het systeem kan worden gereduceerd. Hiernaast zullen de verschillende packages van het systeem worden beschreven zodat toekomstige ontwikkelaars beter grip kunnen krijgen op de applicatie.

Tabel 15 Sprint planning sprint 8

Nr.	Story	Uren
79	Als gebruiker wil ik vanuit de applicatie de rotatie van een apparaat aanpassen.	5
80	Als gebruiker wil ik vanuit de applicatie de positie van een apparaat aanpassen.	10
78	Als ontwikkelaar van het systeem wil ik een omschrijving van de verschillende packages van het systeem.	20
83	Als ontwikkelaar wil ik dat het systeem gemakkelijk te initialiseren is.	5
86	Als ontwikkelaar wil ik dat de koppeling tussen de UI package en de rest van het systeem zo laag mogelijk is.	20

9.2 Backlog toevoegingen

In sprint 8 zijn geen backlog toevoegingen gedaan.

Tabel 16 Backlog toevoegingen sprint 8

Nr.	Story	Herkomst
-	-	-

9.3 Sprint evaluatie

In deze sprint is de koppeling van de UI package met de rest van het systeem verminderd. Er is een Controller package toegevoegd. Deze package zorgt voor een bridge tussen de UI en de business logica. Hierbij is gebruik gemaakt van het Model View Presenter ontwikkelpatroon. Tevens is er een klasse ontworpen die het initialiseren van de verschillende packages van het systeem op zich neemt.

Hiernaast zijn in deze sprint de verschillende packages van het systeem nader toegelicht. Deze beschrijvingen dienen als begin van het ontwerpproject.

10 Sprint 9 - Afronding

10.1 Sprint planning

Sprint 9 is met zijn drie weken de langste sprint van dit project, tevens is dit de laatste sprint van het project. Er is gekozen om deze sprint een week langer te laten duren dan de rest van de sprint aangezien er deze sprint een aantal feestdagen in het midden van de week vallen. Het aantal indeelbare dagen blijft ongeveer hetzelfde als dat van de vorige sprint. Er kan dus niet perse meer werk in deze sprint worden gestopt. Deze sprint zal zich focussen op het ontwerpdocument en het afmaken van het afstudeerverslag met de bijhorende bijlagen.

Tabel 17 Sprint planning sprint 9

Nr.	Story	Uren
87	Als ontwikkelaar van het systeem wil ik een ontwerpdocument waarin de structuur van de applicatie wordt beschreven.	10

10.2 Backlog toevoegingen

In sprint 9 zijn geen backlog toevoegingen gedaan.

Tabel 18 Backlog toevoegingen sprint 9

Nr.	Story	Herkomst
-	-	-

10.3 Sprint evaluatie

In deze sprint is het ontwerprapport gemaakt. Dit rapport beschrijft de applicatie aan de hand van de verschillende UML beschrijvingen die gaandeweg het project zijn gecreëerd. Tevens is de applicatie met de daarbij horende broncode en documentatie overgedragen aan de opdrachtgever.

11 Productbacklog

In het onderstaande is het productbacklog (tot nu toe) te zien. Hierin is af te lezen wanneer een user story is gecreëerd en wanneer deze is geïmplementeerd.

Tabel 19 Productbacklog

Nr.	Story	Herkomst	Geimp..
1	Als product owner wil ik de globale eisen van het te maken software systeem duidelijk hebben gedefinieerd.	x	Sprint 1
2	Als ontwikkelaar wil ik weten hoe moving heads werken en hoe deze moeten worden aangestuurd.	x	Sprint 1
3	Als ontwikkelaar wil ik de werking van het openFrameworks framework analyseren.	x	Sprint 1
4	Als opdrachtgever wil ik een rapport die de hardware beschrijft en test.	x	Sprint 1
5	Als gebruiker wil ik meerdere lampen van mogelijk verschillende typen aan het systeem kunnen toevoegen.	Requirements Document	Sprint 2
6	Als gebruiker wil ik dat lampen worden aangestuurd volgens een bepaald effect.	Requirements Document	Sprint 2
7	Als gebruiker wil ik de mogelijkheid hebben om de positie en rotatie van een moving head semi automatisch te laten bepalen door de applicatie.	Requirements Document	x
8	Als gebruiker wil ik de verschillende lampen in een 3D omgeving zien.	Requirements Document	Sprint 2
9	Als gebruiker wil ik zien waar de verschillende tracking devices zich bevinden in de 3D ruimte.	Requirements Document	Sprint 2
10	Als gebruiker wil ik zien waar de verschillende antennes zich bevinden in de 3D ruimte.	Requirements Document	Sprint 6
11	Als gebruiker wil ik de mogelijkheid hebben om tracking devices te kunnen toevoegen.	Requirements Document	Sprint 2
12	Als gebruiker wil ik dat het systeem ook videobeelden kan aansturen.	Requirements Document	x
13	Als gebruiker wil ik de mogelijkheid om de locatie van een of meerdere tracking devices op te nemen zodat deze later weer kunnen worden afgespeeld.	Requirements Document	Sprint 2
14	Als gebruiker wil ik de mogelijkheid om de opgenomen locatie data van een of meerdere tracking devices weer af te spelen.	Requirements Document	Sprint 2
15	Als ontwikkelaar wil ik dat de locatie data die binnenkomt kan worden gesimuleerd.	Requirements Document	Sprint 2
16	Als gebruiker wil ik de huidige configuratie (positie en rotatie van lampen, toegevoegde tracking devices, actieve effecten) kunnen exporteren.	Requirements Document	x
17	Als gebruiker wil ik de huidige configuratie (positie en rotatie van lampen, toegevoegde tracking devices, actieve effecten) kunnen importeren.	Requirements Document	x
18	Als gebruiker wil ik een lamp met de daarbij horende DMX kanalen automatisch laten inladen door gebruik te maken van het desbetreffende XML bestand ¹ .	Sprint 2	Sprint 4

19	Als gebruiker wil ik de mogelijkheid hebben om meerdere lampen tegelijkertijd te kunnen toevoegen door middel van een XML bestand waarin de lampen staan beschreven.	Sprint 2	Sprint7
20	Als ontwikkelaar wil ik onderzoeken welk framework het beste geschikt is om de applicatie in te ontwikkelen.	Sprint 2	Sprint 2
21	Als ontwikkelaar wil ik een basisontwerp van de applicatie creëren om van daaruit verder te ontwikkelen.	Sprint 2	Sprint 2
22	Als gebruiker wil ik de mogelijkheid om een 3 ^e party controller te kunnen koppelen met de applicatie waardoor het mogelijk wordt om de parameters van effecten te kunnen aanpassen.	Sprint 2	x
23	Als gebruiker wil ik dat de applicatie gebruik maakt van een echt 3D model van een moving head in plaats van een kubus.	Sprint 2	x
24	Als gebruiker wil ik dat de applicatie gebruik maakt van een echt 3D model van een lamp in plaats van een kubus.	Sprint 2	x
25	Als ontwikkelaar wil ik onderzoeken wat een effect precies inhoudt.	Sprint3	Sprint3
26	Als ontwikkelaar wil ik onderzoeken welke scriptaal het meest geschikt is om te gebruiken voor het schrijven van effecten.	Sprint3	Sprint3
27	Als gebruiker wil ik effecten kunnen toevoegen aan een lamp.	Sprint3	Sprint3
28	Als gebruiker wil ik een effect kunnen verwijderen van een lamp.	Sprint3	Sprint3
29	Als effect schrijver wil ik een effect script kunnen toevoegen aan de applicatie.	Sprint3	Sprint3
30	Als effect schrijver wil ik een effect script kunnen verwijderen uit de applicatie.	Sprint3	Sprint3
31	Als student wil ik het bedrijfsbezoek voorbereiden.		Sprint3
32	Als effect schrijver wil ik een effect script 'on the fly' kunnen aanpassen of toevoegen	Sprint3	x
33	Als effect schrijver wil ik een handleiding over hoe ik een effect kan schrijven voor de applicatie.	Sprint3	Sprint 6
34	Als effect schrijver wil ik de status van het compileren van mijn script zien binnen de applicatie.	Sprint3	x
35	Als effect schrijver wil ik de status van het compileren kunnen zien binnen het programma waarmee ik het effect schrijf.	Sprint3	x
36	Als effect schrijver wil ik auto correctie op het moment dat ik effecten aan het schrijven ben	Sprint3	x
37	Als ontwikkelaar wil ik onderzoeken welke functionaliteit de user interface krijgt.	Sprint 4	Sprint 4
38	Als ontwikkelaar wil ik de verschillende mogelijkheden onderzoeken voor het tonen van een user interface binnen Unity.	Sprint 4	Sprint 5
39	Als ontwikkelaar wil ik een dag meelopen met de opdrachtgever om te kijken in wat voor soort situatie de applicatie later zou kunnen worden ingezet.	Sprint 4	Sprint 4
40	Als gebruiker wil ik een manier om het podium in te laden in de 3D wereld. Dus dat bijvoorbeeld het podium (met hoogte verschillen) zichtbaar wordt in de 3D wereld.	Sprint 4	x
41	Als gebruiker wil dat de teksten binnen de applicatie in mijn eigen taal zijn.	Sprint 4	x

HAUTE TECHNIQUE

42	Als gebruiker wil ik de manier hoe een tracking device wordt geüpdatet (bijvoorbeeld door een muis of door een tracker) kunnen veranderen zonder dat ik hiervoor de effecten opnieuw moet instellen die aan het tracking device waren gekoppeld.	Sprint 4	Sprint 6
43	Als ontwikkelaar wil ik de eerste testdag met Sendrato bijwonen.	Sprint 4	Sprint 4
44	Als ontwikkelaar wil ik de resultaten van de eerste testdag beschrijven.	Sprint 4	Sprint 4
45	Als gebruiker wil ik de mogelijkheid om een lamp via de UI aan de applicatie toe te voegen	UI Document	x
46	Als gebruiker wil ik de mogelijkheid om een lamp uit de applicatie te kunnen verwijderen via de UI.	UI Document	x
47	Als gebruiker wil ik de mogelijkheid om de verschillende instellingen van een lamp (zoals bijvoorbeeld de positie en de rotatie) te kunnen aanpassen via de UI.	UI Document	x
48	Als gebruiker wil ik de mogelijkheid om de verschillende beschikbare effecten te kunnen inzien.	UI Document	x
49	Als gebruiker wil ik de mogelijkheid hebben om een effect te creëren en te koppelen aan verschillende lampen en tracking devices via de UI.	UI Document	x
50	Als gebruiker wil ik kunnen zien welke effecten actief zijn (op welke lamp(en) en of tracking device(s)) binnen de applicatie.	UI Document	x
51	Als gebruiker wil ik de mogelijkheid hebben om effecten te kunnen starten of te kunnen stoppen via de UI.	UI Document	x
52	Als gebruiker wil ik de mogelijkheid hebben om effecten te kunnen verwijderen via de UI.	UI Document	x
53	Als gebruiker wil ik de verschillende parameters van een actief effect kunnen inzien en kunnen veranderen.	UI Document	x
54	Als gebruiker wil ik de verschillende parameters van een actief effect kunnen koppelen met een 3d party MIDI controller via de UI.	UI Document	x
55	Als gebruiker wil ik de mogelijkheid hebben om de verschillende parameters van een actief effect te kunnen koppelen aan een 3 ^e party controller via de UI.	UI Document	x
56	Als gebruiker wil ik een tracking device kunnen toevoegen aan de applicatie via de UI.	UI Document	Sprint 6
57	Als gebruiker wil ik een tracking device kunnen verwijderen van de applicatie via de UI.	UI Document	x
58	Als gebruiker wil ik een overzicht hebben van de verschillende tracking devices binnen de applicatie.	UI Document	Sprint 6
59	Als gebruiker wil ik de input methode van een tracking device binnen de UI kunnen veranderen.	UI Document	Sprint 6
60	Als gebruiker wil ik de mogelijkheid om een opname te starten binnen de UI.	UI Document	x
61	Als gebruiker wil ik een opname kunnen starten van uit een lijst van opnamen.	UI Document	x
62	Als gebruiker wil ik een de status van een opname (start, stop of pauzeren) kunnen veranderen binnen de UI.	UI Document	x
63	Als gebruiker wil ik de mogelijkheid om een opname te kunnen verwijderen binnen de UI.	UI Document	x

64	Als gebruiker wil ik de verschillende instellingen van de applicatie kunnen inzien en veranderen binnen de UI.	UI Document	x
65	Als gebruiker wil ik een DMX node kunnen toevoegen via de UI.	UI Document	x
66	Als gebruiker wil ik een DMX universe kunnen toevoegen aan de applicatie via de UI.	UI Document	x
67	Als gebruiker wil ik een DMX universe kunnen koppelen aan een DMX node via de UI.	UI Document	x
68	Als gebruiker wil ik de mogelijkheid hebben om een DMX node te verwijderen via de UI.	UI Document	x
69	Als gebruiker wil ik de mogelijkheid om een DMX universe te verwijderen via de UI.	UI Document	x
70	Als gebruiker wil ik dat de applicatie automatisch zoekt naar ArtNet nodes en deze laat zien.	UI Document	x
71	Als gebruiker wil ik dat DMX universes niet worden verwijderd als er een DMX node wordt verwijderd.	UI Document	x
72	Als ontwikkelaar wil ik de verschillende opties voor het aansturen van de applicatie onderzoeken, vooral kijkend naar hoe de camera zich gedraagt en zou kunnen worden aangestuurd binnen de applicatie.	UI Document	x
72.1	Als gebruiker wil ik een keuzemenu te zien krijgen als ik op een lamp in de 3D wereld druk.	UI Document	x
72.2	Als gebruiker wil ik een keuzemenu te zien krijgen als ik op een tracking device in de 3D wereld druk.	UI Document	x
72.3	Als gebruiker wil ik een overzicht zien van alle toegevoegde apparaten aan de applicatie.	UI Document	x
73	Als gebruiker wil ik een grid zien waarover de verschillende tracking devices zich bewegen.	Testdag #1	Sprint 6
74	Als gebruiker wil ik dat ik snel het systeem van Sendrato kan koppelen met de applicatie ongeacht de opstelling van de hardware.	Testdag #2	Sprint 6
75	Als gebruiker wil ik inzicht in verschillende parameters van de applicatie zoals: het aantal frames per seconde en het aantal locatie updates per seconde.	Testdag #3	Sprint 5
76	Als opdrachtgever wil ik dat de testdag wordt verwerkt in het hardware rapport.	Sprint 5	Sprint 5
77	Als effectschrijver wil ik dat als ik de pan en tilt van een moving head in de code aanpas dit ook zichtbaar wordt in de 3D wereld (fix).	Sprint 7	Sprint 7
78	Als ontwikkelaar van het systeem wil ik een opzet van het ontwerpdocument waarin het systeem nader wordt beschreven.	Sprint 7	Sprint 8
79	Als gebruiker wil ik vanuit de applicatie de rotatie van een apparaat aanpassen.	User Story 47	Sprint 8
80	Als gebruiker wil ik vanuit de applicatie de positie van een apparaat aanpassen.	User Story 47	Sprint 8
81	Als gebruiker wil ik vanuit de applicatie de DMX instellingen van een lamp veranderen.	User Story 47	x
82	Als gebruiker wil ik dat de instellingen van een apparaat worden opgeslagen zodat ik handmatige bijstellingen niet telkens opnieuw hoeft te doen.	User Story 47	x

HAUTE TECHNIQUE

83	Als ontwikkelaar wil ik dat het systeem gemakkelijk te initialiseren is.	Sprint 8	Sprint 8
84	Als gebruiker wil ik dat als ik op een apparaat klik een dashboard te zien krijg met daarin informatie over de lamp (type apparaat, positie, rotatie, DMX universe).	Sprint 7	x
85	Als gebruiker wil ik de positie of rotatie van meerdere apparaten tegelijker tijd kunnen aanpassen.	Sprint 7	x
86	Als ontwikkelaar wil ik dat de koppeling tussen de UI package en de rest van het systeem zo laag mogelijk is.	Sprint 8	Sprint 8

Bijlage G

Handleiding effect schrijven

HAUTE TECHNIQUE

Handleiding

Auteur	: Jan de Boer
Bedrijf	: HAUTE TECHNIQUE
Opdrachtgever	: Sander ter Braak
Datum	: 16 november 2015
Versie	: 1.0

Versiebeheer

Versie	Datum	Opmerking
0.1	17-11-2015	Eerste versie
1.0	05-01-2016	Verschillende wijzigingen in de opmaak doorgevoerd.

Inhoudsopgave

1	Introductie.....	5
2	Installatie	6
2.1	Unity installeren	6
2.2	Ontwikkelomgeving opzetten	6
3	Unity	7
4	Effecten	8
4.1	Opbouw van een effect	8
4.2	EffectDescriptor.....	10
4.2.1	Methodes	10
4.2.2	Variabelen.....	12
4.3	Effect.....	13
4.3.1	Methodes	13
4.3.2	Variabelen.....	16

1 Introductie

Dit document beschrijft de verschillende stappen die nodig zijn voor het schrijven van een effect. Er wordt uitgelegd hoe Unity moet worden geïnstalleerd en hoe de ontwikkelomgeving moet worden opgezet.

2 Installatie

2.1 Unity installeren

De applicatie is ontwikkeld met Unity het is daarom noodzakelijk om de Unity Editor te installeren. Unity kan worden geïnstalleerd vanaf de volgende URL: <https://unity3d.com/get-unity>. De applicatie is op dit moment alleen nog maar getest met Unity versie 5.2.0. Het is daarom aan te raden om bij deze versie te blijven. Deze versie van Unity kan vanaf de volgende URL worden gedownload: <https://unity3d.com/get-unity/download/archive>. Kies in beide gevallen voor de *Unity Installer* en loop de setup procedure door.

2.2 Ontwikkelomgeving opzetten

Als Unity succesvol op het systeem is geïnstalleerd kan er begonnen worden met het verder opzetten van de ontwikkelomgeving. De source code van het project is te vinden in de volgende GitHub repository. Let op: toegang tot deze repository is vereist.

Repository: <https://github.com/hautetechnique/PositionTracking/tree/master>

Er zijn grofweg twee manieren voor het verkrijgen van de source code van de applicatie. De eerste en makkelijkste manier is door een zip bestand van de repository te downloaden. Deze manier is echter niet aan te raden omdat veranderingen aan de code base dan niet gemakkelijk aan de repository zijn toe te voegen. Daarom wordt er aangeraden om gebruik te maken van git. Er wordt hierbij van uitgegaan dat er basiskennis van git beschikbaar is om de source code te beheren.

Hier volgen de twee URL die kunnen worden gebruikt om de repository te klonen. De SSH manier is het makkelijkst in gebruik maar vergt wat extra opzetwerk aangezien je je public key moet toevoegen aan je GitHub account. De HTTPS methode kan in een keer worden gebruikt.

SSH: <git@github.com:hautetechnique/PositionTracking.git>

HTTPS: <https://github.com/hautetechnique/PositionTracking.git>

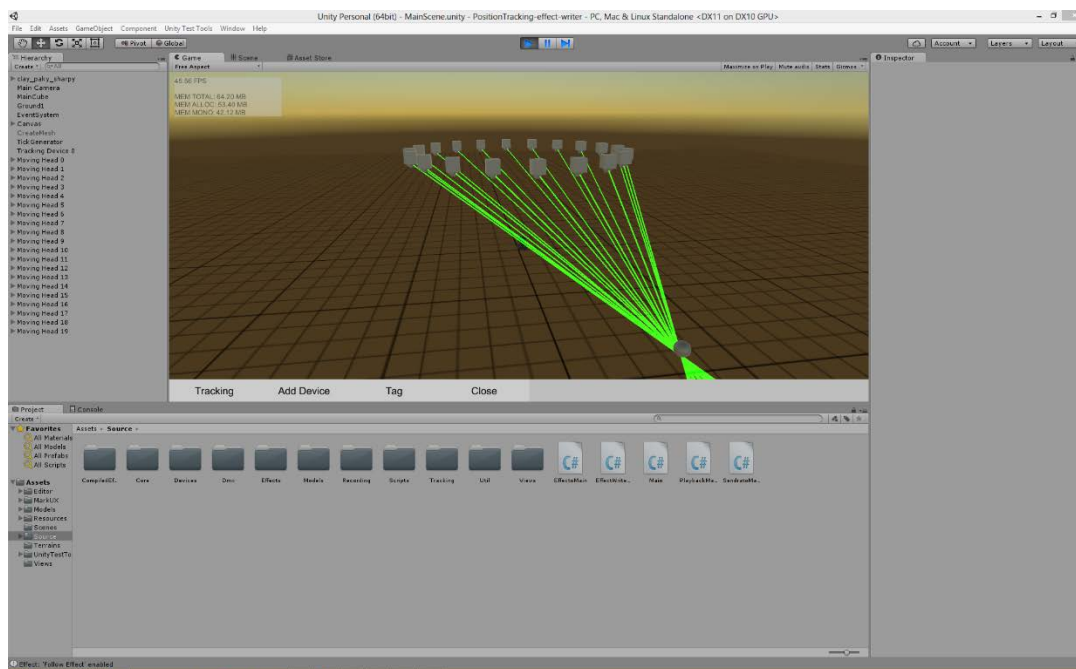
De repository is voorzien van een aantal README.md bestanden die verder gebruik van de repository uitleggen. Om het project in Unity te openen kan er gebruik worden gemaakt van de *MainScene.unity* in *Assets/Scenes/MainScene.unity*.

Het project maakt gebruik van verschillende branches. In deze branches zitten verschillende versies van de applicatie. Er is een speciale branch aangemaakt waar het een en ander is klaargezet voor het schrijven van effecten; deze branch heet *effect-writer*. Switch dus naar deze branch nadat je de code gekloond hebt. Mocht je voor de eerste methode gaan, het downloaden van een zip bestandje, let dan op dat je de zip van de *effect-writer* branch download en niet die van bijvoorbeeld de *master*.

3 Unity

Na het installeren van Unity en het verkrijgen van de source code kan Unity voor het eerst worden opgestart. Dit kan als volgt worden gedaan:

1. Navigeer naar de map waar je de applicatie code hebt neergezet.
2. Navigeer naar de map `/Assets/Scenes/MainScene.unity` en open dit bestand. Als het goed is wordt Unity automatisch geopend en zal Unity het project 'importeren'.
3. Nadat Unity is opgestart krijg je ongeveer het volgende scherm te zien:



Navigeer (binnen Unity) naar `/Source` en open het bestand *EffectWriterMain.cs*. Dit bestand is de zogenoemde 'main' van de applicatie. Het bestand is voorzien van commentaar om de werking uit te leggen. Voor meer informatie over de werking van de Unity Editor kan de volgende link geraadpleegd worden:

Unity Editor: <http://docs.unity3d.com/Manual/LearningtheInterface.html>

4. In het bestand *EffectWriterMain.cs* bevindt zich de klasse *EffectWriterMain*, deze klasse is een zogenoemde *MonoBehaviour*. Voor meer informatie over *MonoBehaviour* en game object kunnen de volgende links worden geraadpleegd.

MonoBehaviour: <http://docs.unity3d.com/ScriptReference/MonoBehaviour.html>

Game objecten: <http://docs.unity3d.com/Manual/GameObjects.html>

4 Effecten

Effecten worden binnen de applicatie gebruikt om gedrag aan DMX apparaten toe te voegen. Dit gebeurt aan de hand van gebeurtenissen zoals het voorbijgaan van de tijd of het veranderen van de positie van een tracking device.

Effect kunnen op twee manieren worden toegevoegd aan de applicatie. Namelijk door zogenoemde effect scripts of door ze mee te compileren met de applicatie. Op dit moment is de ondersteuning voor het scripten van effecten nog niet gebruiksvriendelijk genoeg om gebruikt te worden. Er wordt daarom aangeraden om effecten mee te compileren. Het schrijven van een effect is via beide wegen hetzelfde.

Het creëren van een nieuw effect wordt gedaan door een nieuw C# bestand te maken in de map */Assets/Source/CompiledEffects*. De naam van het bestand is hetzelfde als de naam van het effect maar dan in camel casing. Voorbeeld: het effect heet: 'Simple Effect' dan wordt de bestandsnaam: *SimpleEffect.cs*. In dit bestand bevinden zich de twee klassen die samen het effect representeren.

4.1 Opbouw van een effect

Een effect bestaat uit twee verschillende klassen, namelijk een klasse die het effect omschrijft, de zogenoemde *EffectDescriptor*. Deze klasse biedt informatie aan over het effect zoals welke apparaten het effect ondersteund. Het andere onderdeel is het effect zelf in deze klasse zit het gedrag voor het aansturen van de DMX apparaten aan de hand van bepaalde gebeurtenissen.

Om een nieuw effect aan het systeem toe te voegen moeten de twee hierboven klassen worden geïmplementeerd. Dit wordt gedaan door twee klassen te creëren. Een overerft van de klasse *AbstractEffectDescriptor* en de andere van *AbstractLocationEffect*. Om een beeld te creëren van wat dit inhoudt wordt hier het simpelst mogelijke effect getoond.

```
class SimpleEffectDescriptor : AbstractEffectDescriptor
{
    public override IEffect Create() {
        AbstractLocationEffect effect = new SimpleEffect();
        effect.SetDescriptor(this);

        return effect;
    }

    public override string GetAuthor() {
        return "Simple Author";
    }

    public override string GetDescription() {
        return "The most simple effect possible";
    }

    public override string GetName() {
        return "Simple Effect";
    }

    public override void Initialize() {
        // Called to initialize the effect
    }

    public override bool SupportForDevice(DmxDevice device) {
```

```
        // Called to determine if this effect supports this specific device
        return true;
    }
}

class SimpleEffect : AbstractLocationEffect {
    public override void Initilize() {
        // Called to initialize the effect
    }
}
```

De hierboven geïmplementeerde methodes moeten door elk effect worden geïmplementeerd. Er zijn echter nog meer methodes die kunnen worden gebruikt afhankelijk per effect. Deze worden in de volgende hoofdstukken verder toegelicht. Voor meer voorbeelden van effecten kan de source code van het project bekeken worden. Vooral de map */Assets/Source/ComplidEffects* is dan interessant.

4.2 EffectDescriptor

De effect descriptor vertelt het systeem verschillende zaken over het effect. Deze informatie wordt door het systeem gebruikt om het effect correct te kunnen gebruiken. Naast verplicht te implementeren methodes zijn er ook een aantal methodes die optioneel zijn. Deze methodes kunnen worden geïmplementeerd mocht dat gewenst zijn.

4.2.1 Methodes

4.2.1.1 Initialize()

Template: void Initialize()

Omschrijving: Wordt aangeroepen op het moment dat de *EffectDescriptor* voor het eerst wordt gecreëerd door het systeem. Deze methode wordt gebruikt om features aan het effect toe te voegen.

Opmerking: Deze methode wordt als eerste aangeroepen nog voor de methode *Load()*.

Voorbeeld:

```
public override void Initialize() {  
  
    // Add support for unlimited devices and one tracking devices  
    AddSupportForFeature(EffectFeature.Devices, Unlimited);  
    AddSupportForFeature(EffectFeature.TrackingDevices, 1);  
}
```

4.2.1.2 AddSupportForFeature()

Template: void AddSupportForFeature(EffectFeature feature, int num)

Omschrijving: Voeg ondersteuning toe voor een bepaalde feature. Op dit moment zijn er twee features namelijk Devices en TrackingDevices. De tweede parameter (*num*) wordt gebruikt om aan te geven voor hoeveel apparaten en of tracking devices het effect ondersteuning heeft.

Opmerking: -

Voorbeeld:

```
public override void Initialize() {  
  
    // Add support for unlimited devices and one tracking devices  
    AddSupportForFeature(EffectFeature.Devices, Unlimited);  
    AddSupportForFeature(EffectFeature.TrackingDevices, 1);  
}
```


4.2.1.3 Load()

Template: void Load()

Omschrijving: Wordt aangeroepen op het moment dat de effect descriptor wordt geladen door het systeem.

Opmerking: Vergeet niet base.Load() als eerste lijn aan te roepen in de methode.

Voorbeeld:

```
public override void Load() {
    base.Load();

    // Do your thing...
}
```

4.2.1.4 Unload()

Template: void Unload()

Omschrijving: Wordt aangeroepen op het moment dat de effect descriptor wordt uitgeladen door het systeem.

Opmerking: Vergeet niet base.Unload() als eerste lijn aan te roepen in de methode.

Voorbeeld:

```
public override void Unload() {
    base.UnLoad();

    // Do your thing...
}
```

4.2.1.5 GetName()

Template: string GetName()

Omschrijving: Wordt door het systeem aangeroepen om de naam van het effect op te vragen.

Opmerking: De naam van het effect moet uniek zijn binnen de applicatie anders wordt het effect niet ingeladen.

Voorbeeld:

```
public override string GetName() {
    return "My Effect";
}
```

4.2.1.6 GetDescription()

Template: string GetDescription()

Omschrijving: Wordt door het systeem aangeroepen om de omschrijving van het effect op te vragen.

Opmerking: Moet minimaal bestaan uit 1 karakter.

Voorbeeld:

```
public override string GetDescription() {
    return "My description";
}
```

HAUTE TECHNIQUE

4.2.1.7 GetAuthor()

Template: string GetAuthor()

Omschrijving: Wordt door het systeem aangeroepen om de maker van het effect op te vragen.

Opmerking: Moet minimaal bestaan uit 1 karakter.

Voorbeeld:

```
public override string GetAuthor() {  
    return "The Author";  
}
```

4.2.1.8 Create()

Template: IEffect Create()

Omschrijving: Wordt door het systeem aangeroepen om een nieuwe instantie van het effect te creëren.

Opmerking: Vergeet niet om van het nieuwe gecreëerde effect de descriptor te zetten.

Voorbeeld:

```
public override IEffect Create() {  
    AbstractLocationEffect effect = new SimpleEffect();  
    effect.SetDescriptor(this);  
  
    return effect;  
}
```

4.2.1.9 SupportForDevice()

Template: bool SupportForDevice(DmxDevice device)

Omschrijving: Wordt door het systeem aangeroepen om te controleren of het effect ondersteuning heeft voor het apparaat.

Opmerking: -

Voorbeeld:

```
public override bool SupportForDevice(DmxDevice device) {  
    if(device is MovingHead)  
        return true;  
  
    return false;  
}
```

4.2.2 Variabelen

4.2.2.1 Unlimited

Template: const in Unlimited = int.MaxValue

Omschrijving: Een int die de maximum waarde van een integer representeert.

4.3 Effect

Een effect is, zoals al eerder gezegd verantwoordelijk voor het daadwerkelijk aansturen van de gekoppelde apparaten. Een effect heeft grof weg twee verschillende states namelijk actief en inactief. De gebruiker kan de state van een effect op elk moment veranderen.

Een gebruiker kan een effect creëren en er apparaten en of tracking devices aan toevoegen (afhankelijk per effect). Vervolgens kan de gebruiker het effect starten. Een gebruiker kan ook kiezen om een effect te verwijderen. Het systeem zal dan de actieve instantie van het effect verwijderen.

4.3.1 Methodes

4.3.1.1 Initialize()

Template: void Initialize()

Omschrijving: Wordt als eerste aangeroepen op het moment dat het effect wordt gecreëerd. Deze methode wordt mede gebruikt om de verschillende parameters van een effect toe te voegen.

Opmerking: -

Voorbeeld:

```
public override void Initialize() {
    AddParameter(0, EffectParameter.Create<Color>(Color.red, "Color"));
}
```

4.3.1.2 AddParameter()

Template: void AddParameter(int id, EffectParameter p)

Omschrijving: Voeg een parameter toe aan het effect. Deze parameter kan vanuit het systeem (of van uit het effect zelf) worden veranderd.

Opmerking: Id van de parameter moet uniek zijn

Voorbeeld:

Parameters zijn zo ontworpen dat ze elk typen kunnen opslaan. Zie hieronder een voorbeeld voor het toevoegen van een Color, float en int parameter aan een effect.

```
public override void Initialize() {
    AddParameter(0, EffectParameter.Create<Color>(Color.red, "Color"));
    AddParameter(1, EffectParameter.Create<float>(1.5F, "Float"));
    AddParameter(2, EffectParameter.Create<int>(1, "Int"));
}
```

HAUTE TECHNIQUE

4.3.1.3 GetParameter()

Template:	EffectParameter GetParameter(int i)
Omschrijving:	Verkrijg de parameter die het id i heeft. Als de parameter niet wordt gevonden wordt er null teruggegeven.
Opmerking:	-
Voorbeeld:	

Onderstaand voorbeeld laat zien hoe een parameter kan worden opgevraagd en aangepast.

```
// Get parameter value
Color color = GetParameter(0).Get<Color>();

// Set parameter value
GetParameter(0).Set(Color.green);
```

4.3.1.4 OnTick()

Template:	void OnTick()
Omschrijving:	Wordt elk frame door het systeem aangeroepen.
Opmerking:	Wordt alleen aangeroepen als het effect actie is (<i>Enabled</i>)
Voorbeeld:	

```
public override void OnTick() {
    // Do your thing
}
```

4.3.1.5 OnPositionChanged()

Template:	void OnPositionChanged(TrackingDevice t)
Omschrijving:	Wordt aangeroepen op het moment dat een tracking device van positie verandert.
Opmerking:	Wordt alleen aangeroepen als het effect actief is. (<i>Enabled</i>)
Voorbeeld:	

In onderstaand voorbeeld worden alle moving heads die zijn gekoppeld met het effect geüpdatet zodat ze wijzen naar de (nieuwe) positie van het tracking device. Hierbij is er van uitgegaan dat alle *devices* van de klassen *MovingHead* zijn.

```
public override void OnPositionChanged(TrackingDevice trackingDevice) {
    Vector3 pos = trackingDevice.position;

    foreach (MovingHead movingHead in devices) {
        movingHead.LookAt(pos);
    }
}
```

4.3.1.6 Enable()

Template: void Enable()

Omschrijving: Wordt aangeroepen op het moment dat het effect actief wordt gemaakt.

Opmerking: Vergeet niet om *base.Enable()* als eerste aan te roepen.

Voorbeeld:

Deze methode kan bijvoorbeeld gebruikt worden om de apparaten in een bepaalde initiële staat te zetten.

```
public override void Enable() {
    base.Enable();

    // Do your thing...
}
```

4.3.1.7 Disable()

Template: void Disable ()

Omschrijving: Wordt aangeroepen op het moment dat het effect inactief wordt gemaakt.

Opmerking: Vergeet niet om *base.Disable()* als eerste aan te roepen.

Voorbeeld:

Deze methode kan bijvoorbeeld worden gebruikt om de gekoppelde apparaten terug te zetten in hun initiële staat.

```
public override void Disable() {
    base.Disable();

    // Do your thing...
}
```

4.3.1.8 Dispose()

Template: void Dispose()

Omschrijving: Wordt aangeroepen voordat het effect wordt verwijderd en kan worden gebruikt om geïnitieerd objecten op te ruimen. Dit is alleen nodig als het effect gebruik maakt van object die zelf de *IDisposable*¹ interface implementeren.

1: [https://msdn.microsoft.com/en-us/library/system.idisposable\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.idisposable(v=vs.110).aspx)

Opmerking: -

Voorbeeld:

```
public override void Dispose() {
    base.Dispose();

    // Do your clean up...
}
```

4.3.2 Variabelen

4.3.2.1 devices

Template:	List<DmxDevice> devices
Omschrijving:	Een lijst met alle gekoppelde apparaten.
Opmerking:	-
Voorbeeld:	

Als het effect ondersteuning heeft voor Devices (zie *AddSupportForFeature()*) kunnen deze worden benaderd door gebruik te maken van de *devices* variabelen die binnen het effect toegankelijk is. Hieronder staan twee voorbeelden voor het aanpassen van apparaten.

```
foreach(DmxDevice d in devices) {  
    d.SetValue(DmxChannel.RED, 0.5F);  
}
```

```
foreach(MovingHead h in devices) {  
    h.LookAt(Vector3.zero);  
}
```

Het eerste voorbeeld neemt aan dat alle apparaten (devices) het DMX kanaal RED hebben om hier zeker van te zijn moet dit in de *SupportForDevice()* van *EffectDescriptor* worden gevalideerd. Het zelfde geldt voor de aanname dat alle apparaten instanties zijn van de klasse *MovingHead* in het tweede voorbeeld.

4.3.2.2 trackingDevices

Template:	List<TrackingDevice> trackingDevices
Omschrijving:	Een lijst met alle gekoppelde tracking devices.
Opmerking:	-
Voorbeeld:	

Bijlage H

Ontwerprapport

HAUTE TECHNIQUE

Ontwerprapport

Auteur	: Jan de Boer
Bedrijf	: HAUTE TECHNIQUE
Opdrachtgever	: Sander ter Braak
Datum	: 21 december 2015
Versie	: 1.0

Versiebeheer

Versie	Datum	Opmerking
0.1	21-09-2015	Eerste versie
1.0	02-01-2016	Ontwikkelomgeving en alle beschrijvingen en diagrammen geüpdatet naar de laatste versie.

Gerelateerde documenten

Ref	Auteur	Naam
D1	Jan de Boer	Effect Handleiding
D2	Jan de Boer	Hardware Testrapport

Inhoudsopgave

Versiebeheer.....	3
Gerelateerde documenten.....	3
Figuren.....	5
1 Introductie.....	6
2 Ontwikkelomgeving	7
2.1 Unity	7
2.2 Projectstructuur	7
2.3 Documentatie.....	8
2.4 Unit tests	8
2.5 Sendrato	8
3 Ontwerp.....	9
3.1 Package.....	9
3.2 Core	11
3.2.1 Werking TickProvider	12
3.3 Tracking	13
3.3.1 Afhandeling van een locatie update.....	14
3.3.2 Werking positieverandering van een TrackingDevice	14
3.4 Devices.....	16
3.4.1 Werking DMX apparaat updaten.....	17
3.5 Dmx.....	18
3.5.1 Werking verzenden van DMX data.....	18
3.6 Models.....	19
3.6.1 Werking positie van een Model aanpassen	20
3.7 Effects.....	21
3.7.1 Werking van het inladen van een effect	22
3.7.2 Werking instantiëren van een nieuwe effect	23
3.8 Scripts	24
3.9 Recording.....	25
3.9.1 Werking van opnemen	26
3.10 Presenters.....	27
3.11 Views	28
3.11.1 Tag linken met een TrackingDevice.....	29
3.12 Util	30

3.13 Exceptions	31
Bibliografie	32

Figuren

Figuur 1 Project structuur	7
Figuur 2 Documentatie structuur.....	8
Figuur 3 Package diagram	9
Figuur 4 Klassendiagram	10
Figuur 5 Klassendiagram: Core package.....	11
Figuur 6 Sequentiediagram: werking TickProvider	12
Figuur 7 Klassendiagram: Tracking package.....	13
Figuur 8 Sequentiediagram: afhandeling van een locatie update	14
Figuur 9 Sequentiediagram: updaten van een tracking device.....	15
Figuur 10 Klassendiagram: Devices package.....	16
Figuur 11 Sequentiediagram: DMX apparaat updaten	17
Figuur 12 Klassendiagram: DMX package	18
Figuur 13 Sequentiediagram: verzenden DMX data	18
Figuur 14 Klassendiagram: Models package	19
Figuur 15 Sequentiediagram: positie van een model aanpassen.....	20
Figuur 16 Klassendiagram: Effects package	21
Figuur 17 Sequentiediagram: inladen van een effect	22
Figuur 18 Sequentiediagram: initieer een nieuw effect.....	23
Figuur 19 Klassendiagram: Scripts package	24
Figuur 20 Klassendiagram: Recording package	25
Figuur 21 Statediagram: PlaybackHandler	26
Figuur 22 Sequentiediagram: opnemen.....	26
Figuur 23 Klassendiagram: Presenters package	27
Figuur 24 Klassendiagram: View package	28
Figuur 25 Sequentiediagram: Tag linken met een TrackingDevice	29
Figuur 26 Klassendiagram: Util package.....	30

1 Introductie

Dit document beschrijft de werking van de PosistionTracking applicatie. Deze applicatie is ontwikkeld tijdens de afstudeerperiode van Jan de Boer bij Haute Technique.

Er wordt begonnen met het omschrijven van de ontwikkelomgeving. Hierna wordt het ontwerp van de applicatie nader toegelicht. Dit gebeurt per package. Hierbij worden sequentie en statediagrammen gebruikt om de flow van de code beter weer te geven.

2 Ontwikkelomgeving

2.1 Unity

De applicatie is ontwikkeld in samenwerking met de Unity Engine [1]. Tijdens het ontwikkelen van de applicatie is er gebruik gemaakt van versie 5.2.0. De applicatie maakt gebruik van twee plugins voor de Unity Engine, dit zijn:

MarkUX [2] MarkUX is een GUI framework en wordt gebruikt om de user interface mee weer te geven. Dit is een betaalde plugin en kan worden gevonden in het Unity account van Haute Technique. De applicatie maakt gebruik van versie 1.4.2

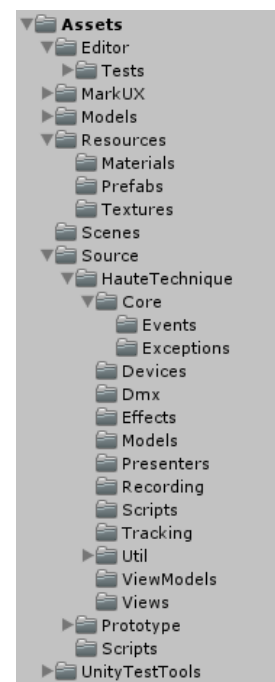
UnityTestTools [3] Is een test framework gemaakt door Unity. Dit framework wordt gebruikt als basis voor de unit tests die tijdens dit project zijn geschreven. De applicatie maakt gebruik van versie 1.5.7

De code van de applicatie is te vinden in de *PostistionTracking* repository in het Github account van Haute Technique. In de *master* branche bevindt zich de laatste stabiele code.

2.2 Projectstructuur

Het project is opgedeeld in een aantal verschillende mappen; op deze manier zijn de verschillende assets van de applicatie beter te vinden. De structuur van het project is ook beschreven aan de hand van verschillende *README.md* bestanden in het project. Hier volgt een korte beschrijving van de belangrijkste mappen in het project. Voor een volledige beschrijving kunnen de verschillende *README.md* bestanden worden geraadpleegd.

De twee beschreven frameworks bevinden zich respectievelijk in de mappen *MarkUX* en *UnityTestTools*. De code van het project bevindt zich in de *HauteTechnique* map. Deze folder is op de zelfde manier ingericht als de packages van het systeem (Figuur 3). De Unit test bevinden zich in de map *Tests*. In de *Resources* map bevinden zich de verschillende prefabs met bijhorende materialen en texturen. Alle items die zich in een Resource map bevinden worden door de Unity Editor run time beschikbaar gemaakt voor de applicatie [4].



Figuur 1 Project structuur

2.3 Documentatie

De meeste klassen en methoden binnen de applicatie zijn voorzien van XML commentaar [5]. Aan de hand van dit commentaar is de werking van de klassen of methoden uitgelegd. Zie onderstaand figuur voor een voorbeeld.

```
/// <summary>
/// Add a TrackingDevice to the effect
/// </summary>
/// <exception cref="MethodNotSupportedException">
/// Thrown when the effect has no support for TrackingDevices
/// </exception>
/// <exception cref="EffectFullException">
/// Thrown when the effect has reached the maximum number of supported
/// TrackingDevices
/// </exception>
/// <param name="trackingDevice">The TrackingDevice to add</param>
public void AddTrackingDevice(TrackingDevice trackingDevice) { ... }
```

Figuur 2 Documentatie structuur

2.4 Unit tests

Om de werking van de klassen in het systeem te testen zijn er verschillende unit tests geschreven; deze bevinden zich in de *Tests* map. De unit tests maken gebruik van het UnityTestTools framework. Dit framework biedt een aantal klassen die het maken van unit tests vergemakkelijkt. Tevens zorgt dit framework ervoor dat de verschillende unit tests door middel van een GUI binnen de Unity Editor te beheren zijn. Achter de schermen maakt het framework gebruik van het C# unit test framework NUnit [6]. De documentatie van dit project kan dan ook gebruikt worden voor het schrijven van unit tests.

2.5 Sendrato

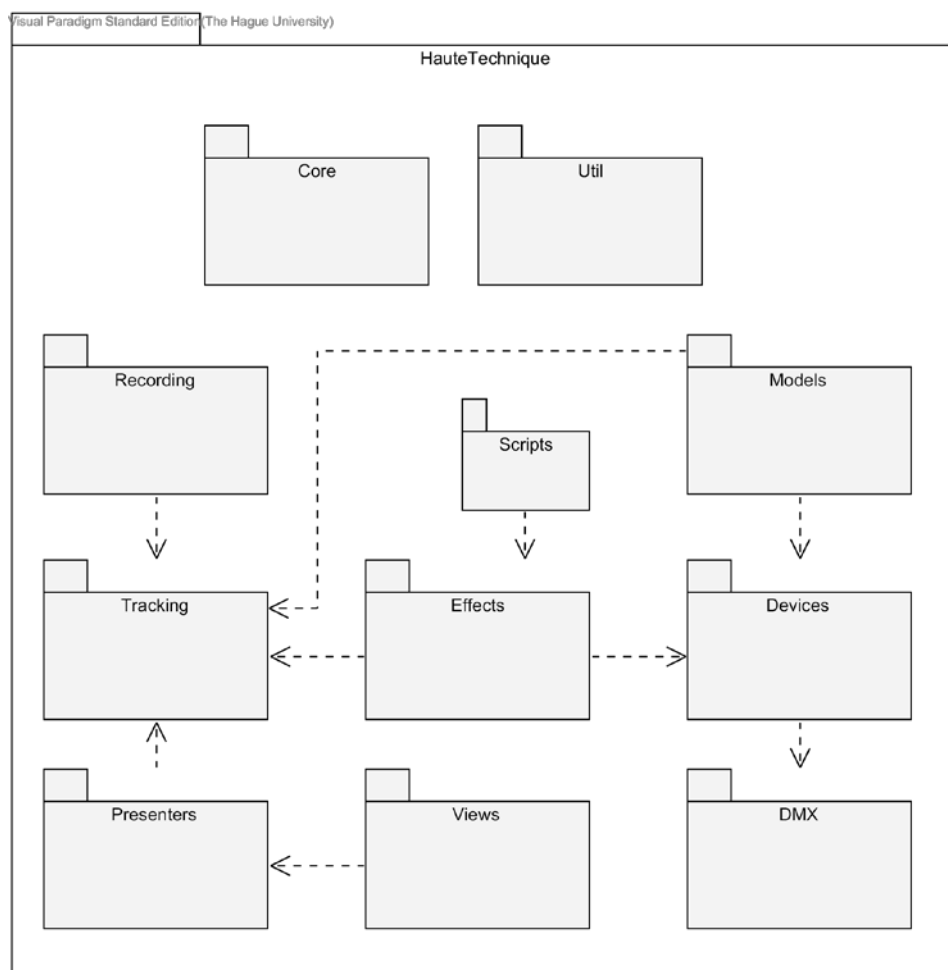
Sendrato levert de hardware voor dit project. Voor een beschrijving van deze hardware en het protocol dat wordt gebruikt om met deze hardware te communiceren kan het Hardware Testrapport [D2] worden geraadpleegd. Er is een applicatie ontwikkeld die het protocol van Sendrato simuleert. Op deze manier kan de applicatie getest worden zonder dat daarvoor de hardware van Sendrato aanwezig hoeft te zijn. Deze applicatie is ook te vinden in de git repository van het project

3 Ontwerp

In dit hoofdstuk wordt het ontwerp van de applicatie beschreven. Dit gebeurt aan de hand van een tal van UML diagrammen. Er zal begonnen worden met een globale omschrijving van het systeem en de verschillende packages. Hierna zal elke package afzonderlijk verder worden toegelicht.

3.1 Package

In onderstaand figuur is het package diagram van de applicatie te zien. Om het diagram te verduidelijken zijn de use pijlen van de *Core* en *Util* package niet weergegeven. In deze packages zitten klassen die vanuit elke package benaderbaar zijn. Meer hierover in de desbetreffende hoofdstukken van de *Core* en *Util* package. De verschillende packages van het systeem worden verder in dit hoofdstuk toegelicht.



Figuur 3 Package diagram

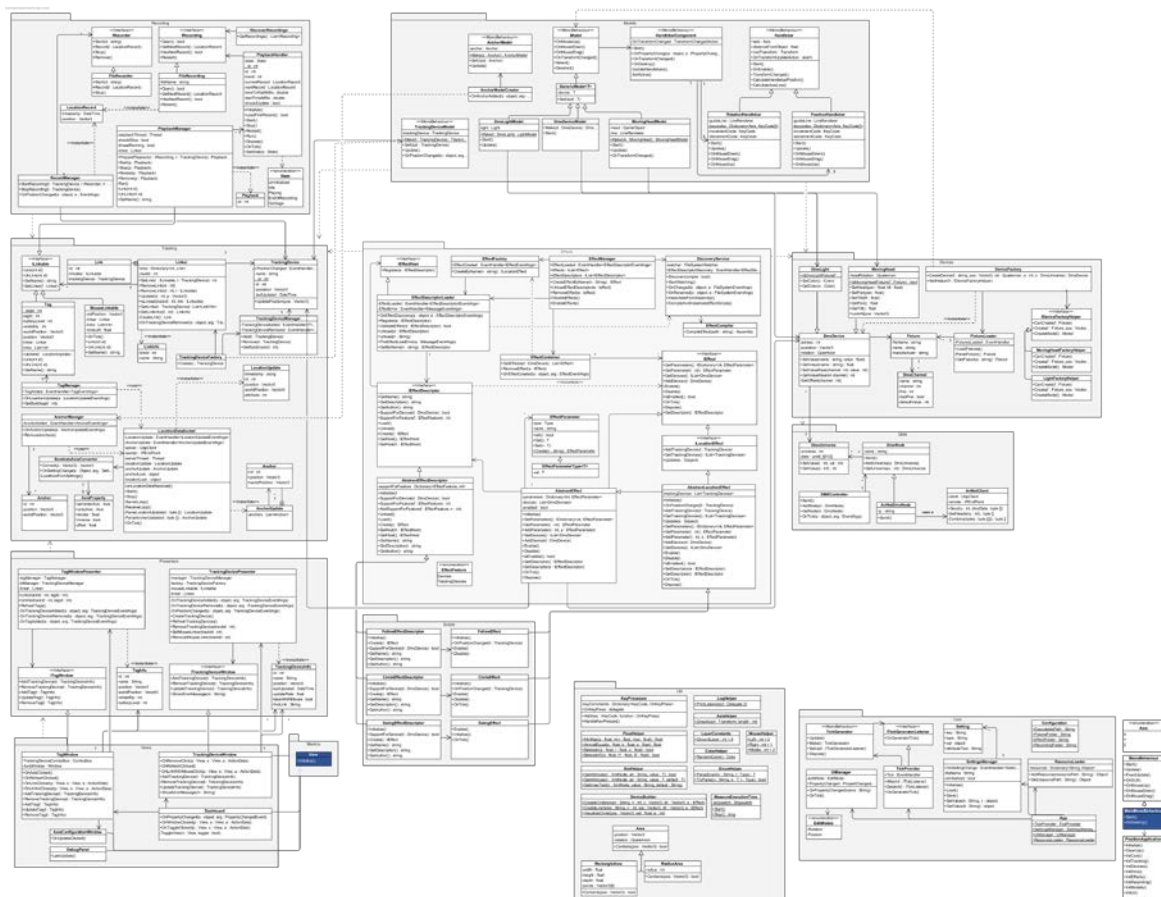
Voor het klassendiagram kan Figuur 4 geraadpleegd worden. In het klassendiagram zijn voor de overzichtelijkheid niet alle use pijlen weergegeven. Zo worden verschillende klassen uit de *Core* en *Util* package gebruikt door klassen in andere packages maar zijn de use pijlen hier weggelaten. Ook zijn voor de hand liggende getters en setters van klassen weggelaten. Verder zijn de klassen die overerven van de *Exception* of *EventArgs* klassen niet weergegeven. De verschillende exceptions worden echter wel nader toegelicht in Hoofdstuk 3.13.

De applicatie is opgedeeld in verschillende packages om zo de structuur van de applicatie te verduidelijken en de uitbreidbaarheid van het systeem te vergroten. Er is getracht om binnen de

HAUTE TECHNIQUE

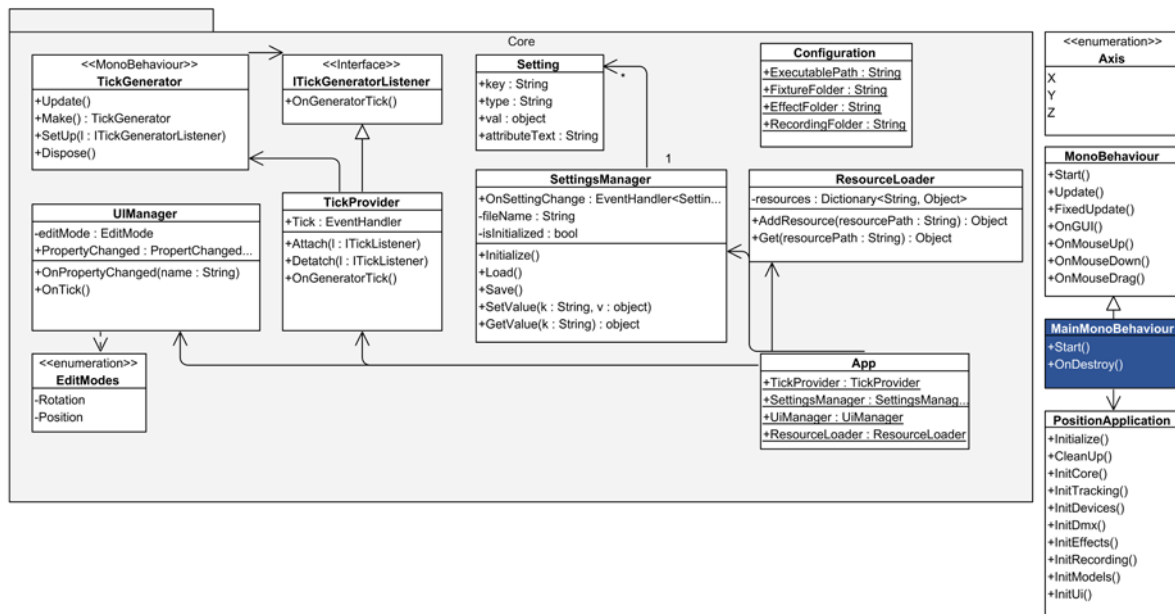
packages de cohesie hoog te houden en de koppeling met andere packages laag, dit om te voorkomen dat een aanpassing in een package leidt tot aanpassingen in andere packages. Voor sommige packages echter woog de keuze voor een duidelijke structuur zwaarder dan een hoge cohesie en een lage koppeling. Dit geldt voornamelijk voor de *Core* en *Util* package. Tevens is er op plekken waar uitbreiding wordt verwacht gebruik gemaakt van interfaces of abstracte klassen. Hierdoor kan er in de toekomst gemakkelijk functionaliteit worden toegevoegd. Dit is bijvoorbeeld terug te zien bij de abstracte *Model* klassen.

Een aantal klassen binnen de applicatie maken gebruik van events om zo bepaalde gebeurtenissen met andere klassen te communiceren. Hiervoor heeft C# een speciaal event type ontwikkeld [7]. Dit type maakt achter de schermen gebruik van het *Observer* [8] patroon. In het klassendiagram zijn deze events te herkennen aan het type *EventHandler* of *EventHandler<T>* waarbij *T* het argument dat wordt meegegeven aan een event is. Het is alleen mogelijk om klassen die overerven van de *EventArgs* klasse mee te geven als argument voor een event. Er zijn dan ook tal van klassen die overerven van de *EventArgs* klasse maar deze zijn niet weergegeven in de verschillende klassendiagrammen.



Figuur 4 Klassendiagram

3.2 Core



Figuur 5 Klassendiagram: Core package

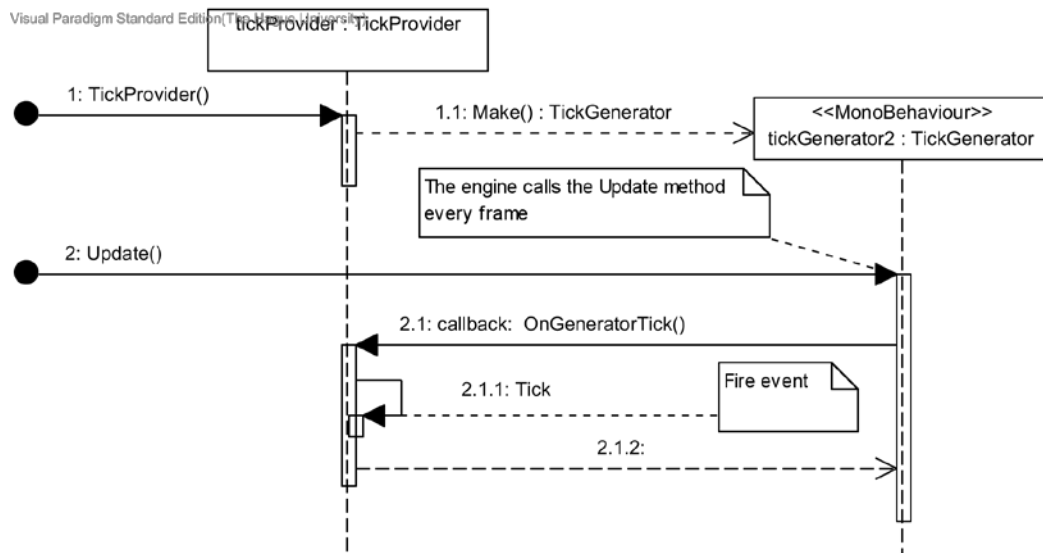
In de *Core* package bevinden zich een aantal klassen die door het gehele systeem te gebruiken zijn door middel van de static klasse *App*. De *App* klasse is een klasse met alleen maar static attributen; er zou dus kunnen worden gesproken van een *Singleton* [9]. Door de *App* klasse zijn de: *SettingsManager* voor het ophalen en opslaan van instellingen, de *ResourceLoader* voor het verkrijgen van resources uit de *Resources* map, de *UiManager* die de modes van de applicatie bijhoudt en de *TickProvider* die kan worden gebruikt om bij elke frame update een notificatie (event) te ontvangen bereikbaar.

Naast de *Core* package zijn drie klassen weergegeven. De klasse *MonoBehaviour* is een klasse binnen de Unity Engine die als basisklasse voor scripts fungeert. Een script binnen Unity stelt je in staat om eigen 'gedrag' aan objecten in de engine toe te voegen. Binnen de applicatie worden een aantal verschillende *MonoBehaviour* scripts gebruikt, deze zijn weergegeven met het stereotype <MonoBehaviour>.

De klasse *MianMonoBehaviour* fungeert als *main* van de applicatie. Deze klasse creëert een instantie van de *PositionApplication* klasse en roept bij het starten van de engine *Initialize()* en bij het afsluiten van de engine *CleanUp()* aan. De *PositionApplication* klasse is verantwoordelijk voor het initialiseren van de verschillende klassen van de applicatie.

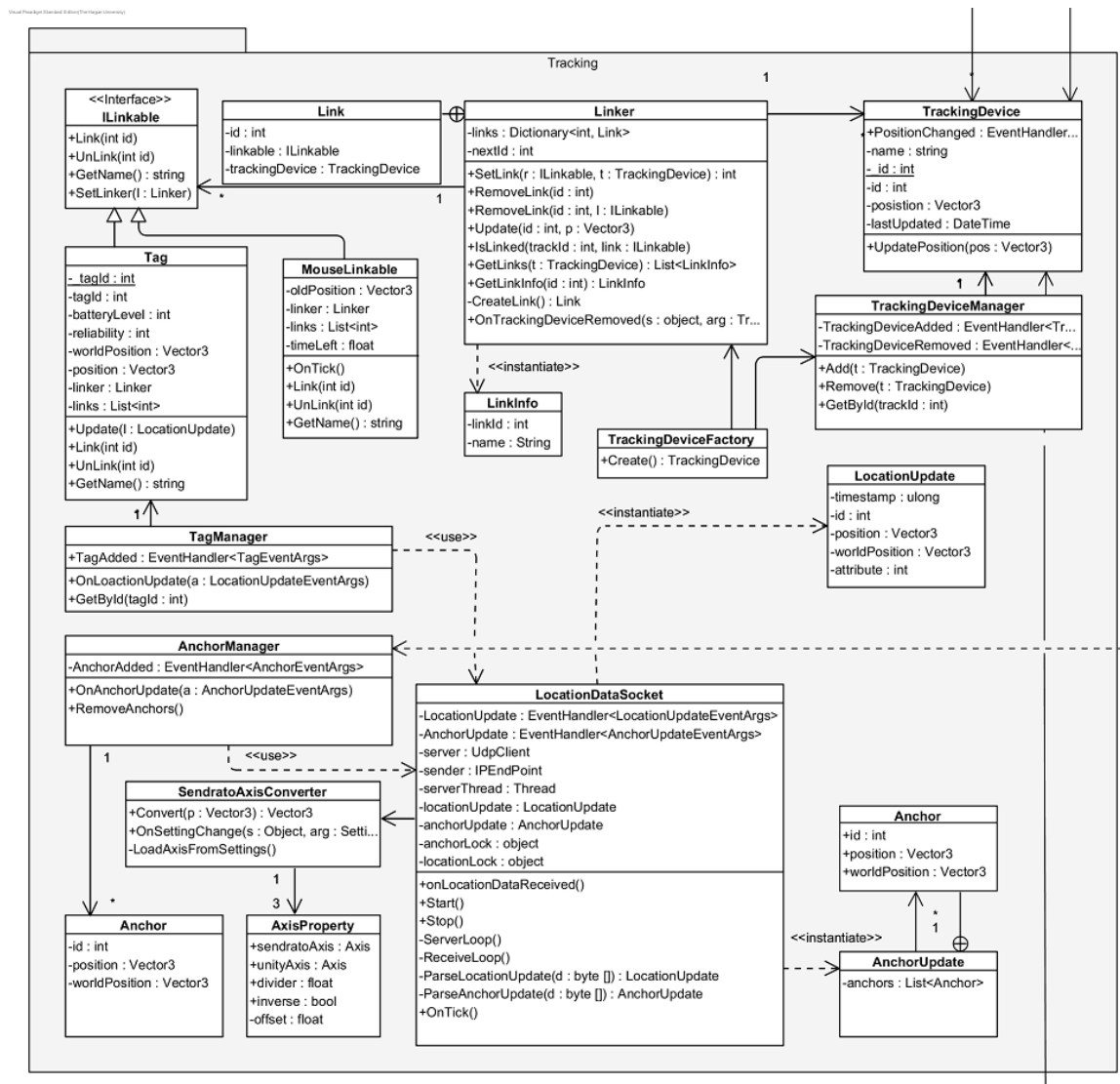
3.2.1 Werking TickProvider

In onderstaand figuur wordt de werking van de *TickProvider* beschreven. De *TickProvider* is verantwoordelijk voor het uitdelen van zogenoemde *Ticks* of updates bij iedere frame update van de engine. Om updates van de engine te ontvangen creëert de *TickProvider* een instantie van de *TickGenerator* klasse. Door gebruik te maken van de *TickProvider* hoeven klassen zelf geen kennis te hebben over *MonoBehaviours*.



Figuur 6 Sequentiedigram: werking TickProvider

3.3 Tracking



Figuur 7 Klassendiagram: Tracking package

In de *Tracking* package bevinden zich verschillende klassen die zich bezighouden met tracking sub deel van de applicatie. Hierbij wordt de data die wordt ontvangen van het systeem van *Sendrato* gebruikt om verschillende klassen te voorzien van data. Op deze manier kan er gereageerd worden op bijvoorbeeld locatie veranderingen van een tracker.

De klasse *LocationDataSocket* is verantwoordelijk voor het parsen van de pakketjes die door het systeem van *Sendrato* worden verstuurd. Voor het ontvangen en parsen van de pakketjes wordt een thread gebruikt aangezien het ontvangen en parsen van de pakketjes relatief lang kan duren, waardoor de framerate negatief kan worden beïnvloed. Er bestaan twee verschillende soorten pakketjes namelijk antenne (*Anchor*) updates en locatie (*Tag*) updates. Beide pakketjes bevatten coördinaten. Deze coördinaten worden met gebruik van de *SendratoAxisConverter* omgezet naar de correcte coördinaten binnen Unity. Op het moment dat een pakketje is geparsed wordt het desbetreffende event afgevuurd. De events worden afgevuurd in de context van de *main* thread; hiervoor wordt gebruik gemaakt van twee lock objecten respectievelijk *anchorLock* en *locationLock* om een race conditie te voorkomen.

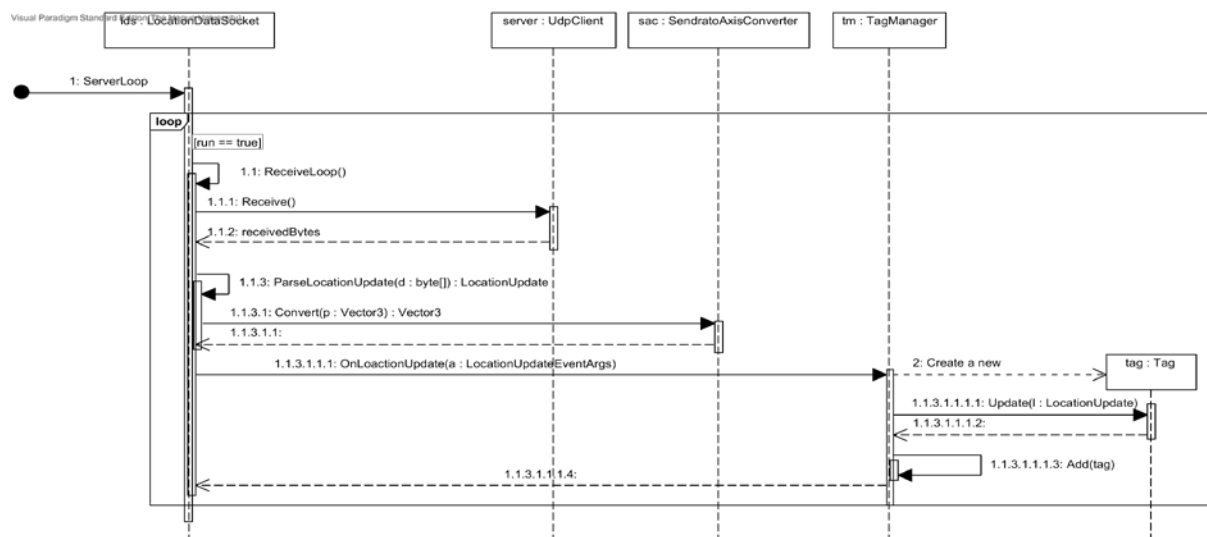
HAUTE TECHNIQUE

LocationDataSocket kent twee events namelijk *LocationUpdate* en *AnchorUpdate*. Bij elk event hoort ook een eigen data structuur die de data uit de pakketjes representeert; deze klassen hebben dezelfde naam als de events (*LocationUpdate* en *AnchorUpdate*). Voor elk event is een *Manager* klasse ontworpen die het event verder afhandelt.

Een andere belangrijke klasse in deze package is de *Linker* klasse. Deze klasse maakt het mogelijk om de manier waarop een *TrackingDevice* wordt geüpdatet om the fly te kunnen veranderen. Hierbij wordt gebruik gemaakt van de *ILinkable* interface waardoor het toevoegen van nieuwe methodes in de toekomst makkelijker wordt. De *TrackingDevice* klasse representeert een tracking device (zoals de naam al zegt). De *TrackingDevice* klasse biedt het event *PositionChanged* aan zodat ander klassen kunnen reageren op positie verandering. Dit wordt bijvoorbeeld gebruikt in de *Effects* package om een effect locatie gevoelig te maken.

3.3.1 Afhandeling van een locatie update

In onderstaand sequentie diagram wordt de afhandeling van een locatie update beschreven; hierbij wordt er vanuit gegaan dat er nog geen Tag is gecreëerd. In dit sequentie diagram is de synchronisatie tussen de ontvang thread en de main Unity thread weggelaten. In werkelijkheid vindt de methode aanroep *OnLocationUpdate()* niet plaats vanuit de thread maar vanuit de Unity thread.

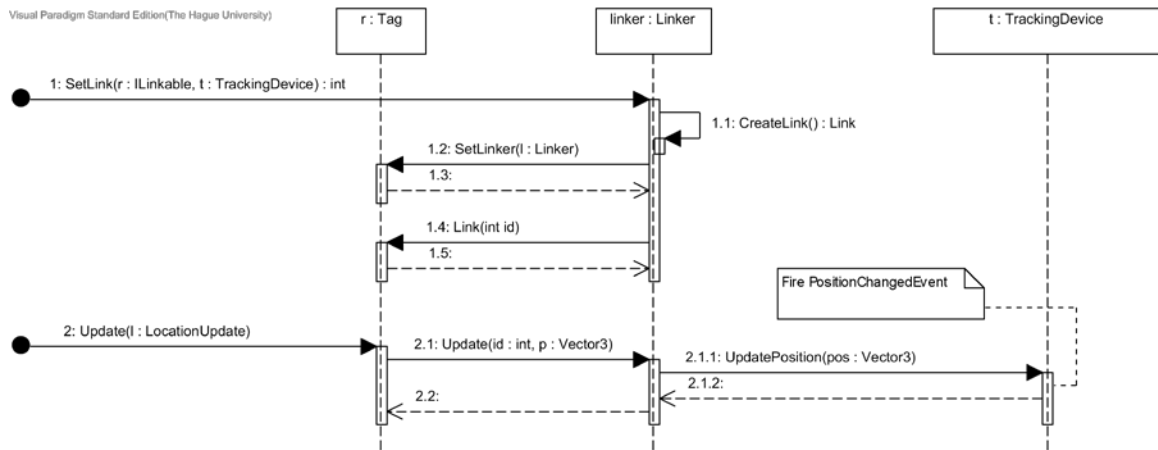


Figuur 8 Sequentiedigram: afhandeling van een locatie update

3.3.2 Werking positieverandering van een TrackingDevice

Zie onderstaand sequentiedigram voor het updaten van een tracking device. Hierbij wordt eerst de betreffende *Tag* gelinkt met het *TrackingDevice* via de methode *SetLink()*. Vervolgens wordt het *TrackingDevice* geüpdatet als er een nieuwe *LocationUpdate* wordt ontvangen (zie FIGUUR 8).

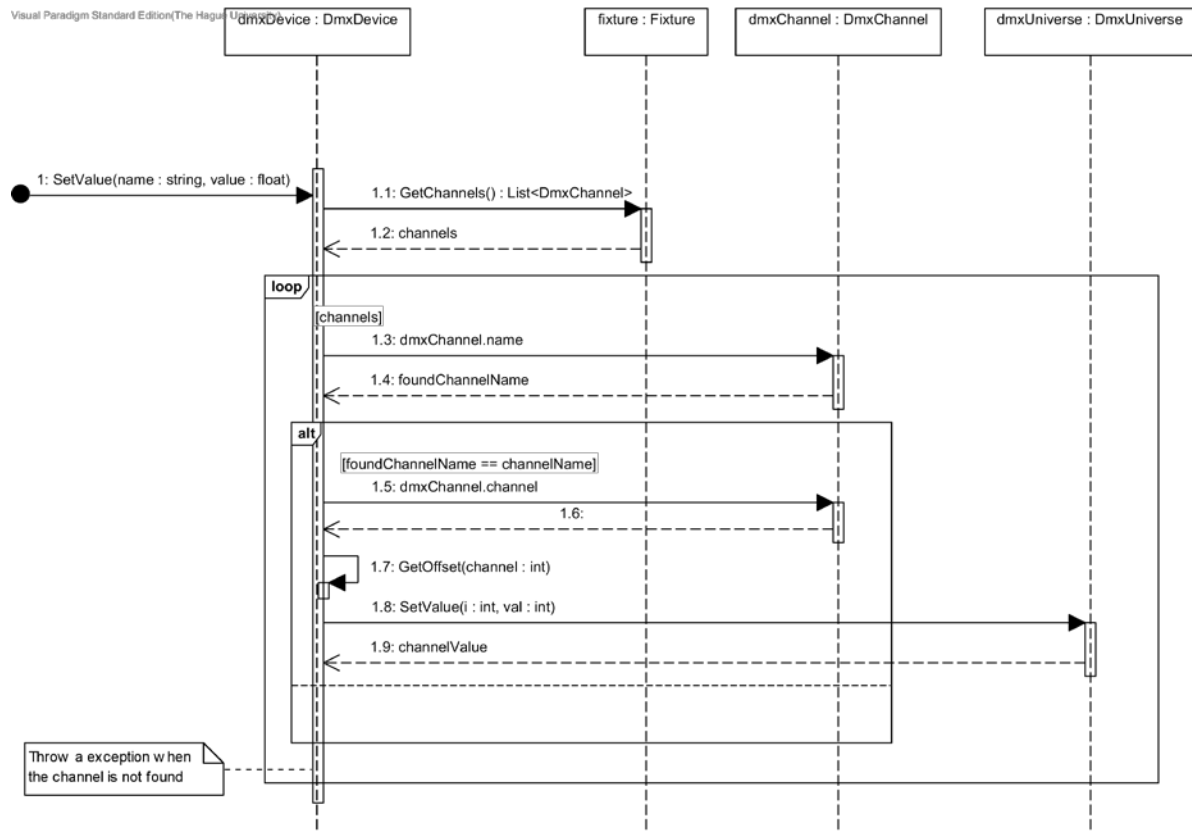
Visual Paradigm Standard Edition(The Hague University)



Figuur 9 Sequentiediagram: updaten van een tracking device

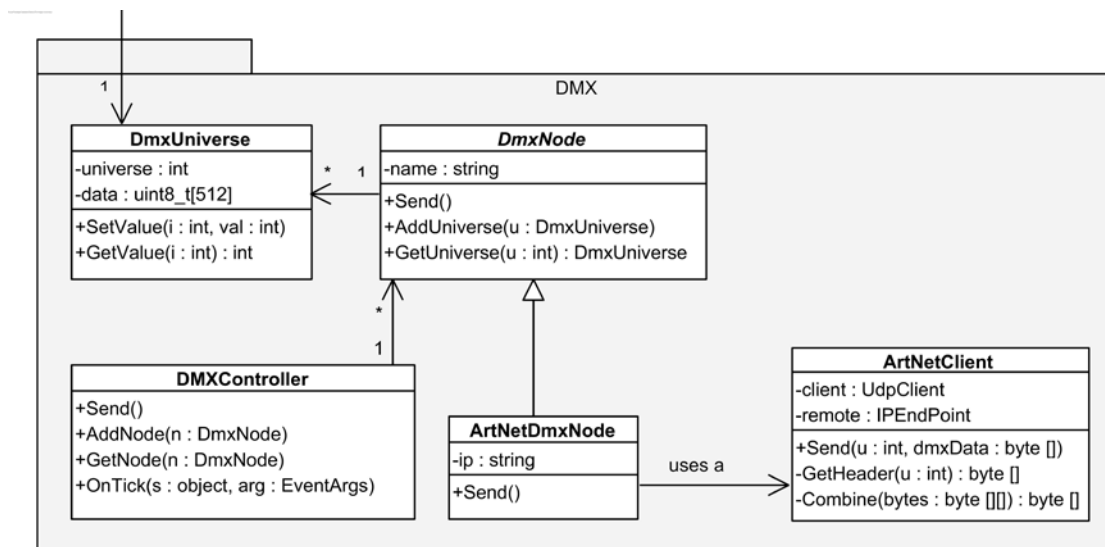
3.4.1 Werking DMX apparaat updaten

Een DMX apparaat (*DmxDevice*) maakt gebruik van een *DmxUniverse* voor het bijhouden van de verschillende waarden van de DMX kanalen. Zie onderstaand sequentiediagram voor de werking van het updaten van een DMX kanaal. Een DMX apparaat kan meerdere kanalen hebben. Op het moment dat het meegegeven kanaal niet wordt gevonden geeft de methode *SetValue* een *exception*.



Figuur 11 Sequentiediagram: DMX apparaat updaten

3.5 Dmx



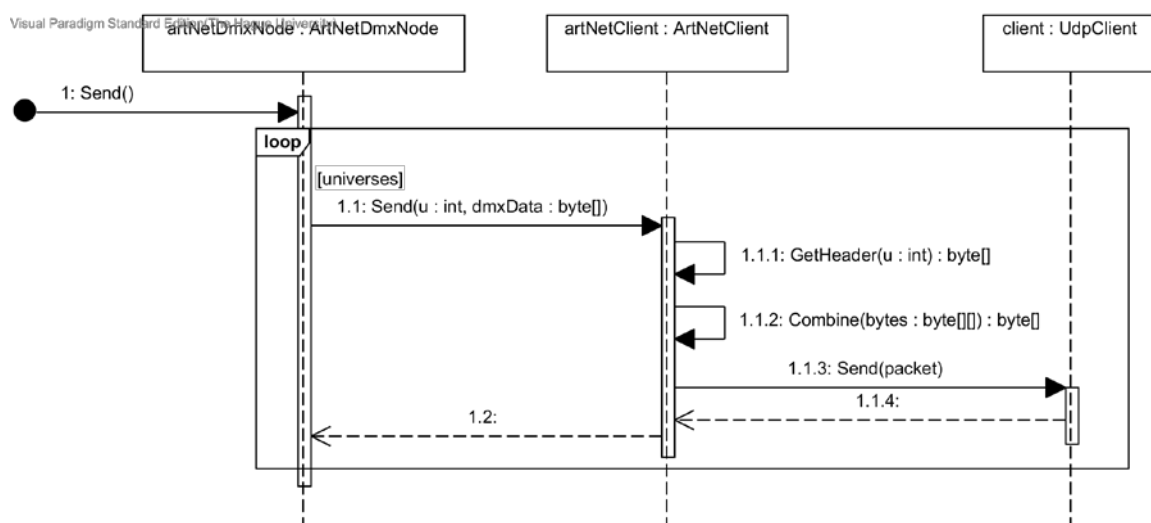
Figuur 12 Klassendiagram: DMX package

De DMX package is verantwoordelijk voor het verzenden van de DMX data. Op dit moment gebeurt dit via het ArtNet protocol. Dit protocol maakt het mogelijk om DMX data over UDP te verzenden. Deze functionaliteit is geïmplementeerd in de *ArtNetDmxNode* en *ArtNetClient*. Er is gebruik gemaakt van een abstracte klasse *DmxNode*, hierdoor is het in de toekomst gemakkelijk de DMX data op een andere manier te verzenden dan via het ArtNet protocol.

Alle *DmxDevices* maken gebruik van een *DmxUniverse* waarin het apparaat de waarden van de verschillende kanalen opslaat. Deze universes worden later verstuurd door een *DmxNode* via de *DmxController*.

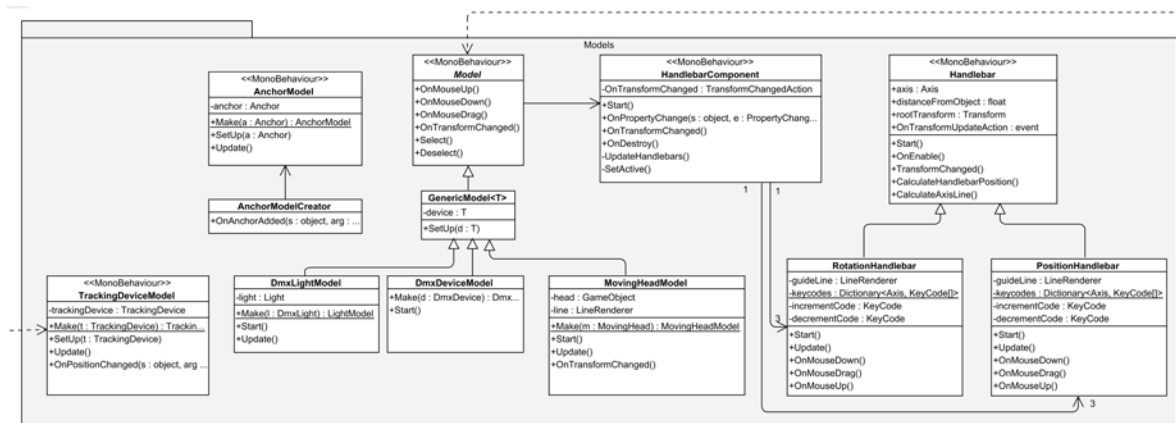
3.5.1 Werking verzenden van DMX data

Zie onderstaand sequentiediagram voor een overzicht hoe het verzenden van DMX data voor *ArtNetDmxNode* in zijn werk gaat.



Figuur 13 Sequentiediagram: verzenden DMX data

3.6 Models



Figuur 14 Klassendiagram: Models package

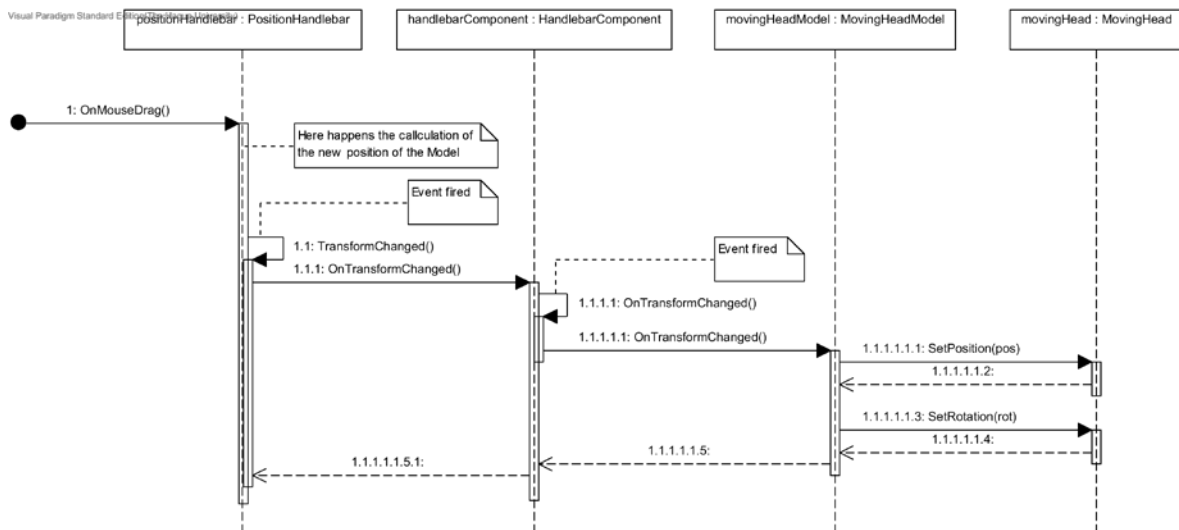
In deze package bevinden zich de verschillende scripts die verantwoordelijk zijn voor het tonen van *DmxDevices*, *TrackingDevices* en *Anchors* in de 3D wereld. De *Model* klassen moeten niet in verwarring worden gebracht met database models. Dit zijn ze namelijk niet. Deze klassen zorgen ervoor dat de verschillende hierboven benoemde klassen zichtbaar kunnen worden gemaakt in de 3D wereld.

Een klasse die overerft van de *MonoBehaviour* klasse kan niet met het `new` keyword worden gecreëerd. Een *MonoBehaviour* klasse moet namelijk aan een *Unity GameObject* worden gekoppeld. Dit is precies wat de methode *Make* van de verschillende *Models* doet. Deze methode creëert een nieuwe instantie van een voorgedefinieerde prefab, [10] waarin het script is toegevoegd. Voor alle *Models* is er een prefab gecreëerd die beschrijft hoe het *Model* er uit moet komen te zien en welke componenten het *Model* gebruikt. Door het gebruik van prefabs is het gemakkelijk om het uiterlijk van een *Model* te veranderen zonder dat hiervoor de code hoeft te worden veranderd. De verschillende prefabs bevinden zich in de *Resources/Prefabs* map.

Het *HandlebarComponent* maakt het mogelijk om de positie of rotatie van een *Model* in de 3D wereld aan te passen. Het *HandlebarComponent* is zo ontworpen dat het in de toekomst gemakkelijk is om ook andere objecten van positie of rotatie te veranderen. Voor het veranderen van de positie maakt het *HandlebarComponent* gebruik van *Handlebars* (handvaten) afhankelijk van de modes van de applicatie zijn dit *PositionHandlebars* of *RotationHandlebars*. De *Handlebars* stellen de gebruiker in staat om de positie of rotatie van een *Model* aan te passen. Door middel van een event communiceren de *Handlebars* met het *HandlebarComponent*.

3.6.1 Werking positie van een Model aanpassen

Op het moment dat de gebruiker op een van de *PositionHandlebars* drukt en begint te slepen wordt de methode *OnMouseDown()* door de Unity engine aangeroepen. Door middel van events wordt de positie verandering doorgecommuniceerd met het *DmxDevice*. In dit geval een *MovingHead*.



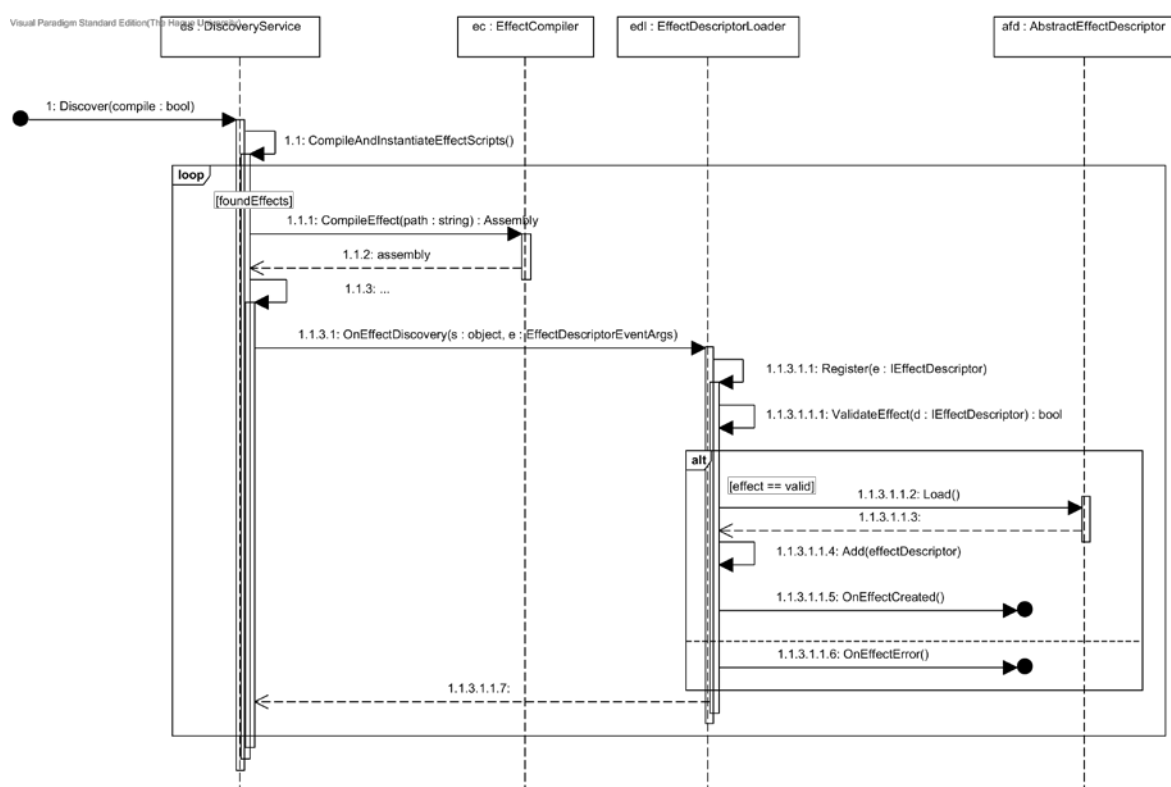
Figuur 15 Sequentiediagram: positie van een model aanpassen

Een effect kan op twee manieren aan de applicatie worden toegevoegd. Door middel van zogenaemde Effect Scripts, dit zijn csharp bestanden in de *effects* map in de root van het project waarin een effect gedefinieerd is of door een effect aan de code base van de applicatie toe te voegen; dit laatste gebeurt in de *EffectScripts* map. Voor de specifieke details kan de Effect Handleiding[D1] worden geraadpleegd.

Van het inladen tot het creëren van een instantie van een effect worden een aantal verschillende klassen gebruikt. Deze klassen communiceren met elkaar door middel van events en worden samen genomen in de *EffectManager*. Deze klasse dient als een facade [11] tot de verschillende klassen van de package. Er zijn in totaal zes verschillende klassen verantwoordelijk voor het beheren van de effecten. Zo worden de effecten 'ontdekt' door de *DiscoveryService*; deze klasse vuurt een event af op het moment dat een effect is gevonden. De *EffectDescriptorLoader* luistert naar dit event en probeert om het effect in te laden; hiervoor wordt de methode *ValidateEffect()* gebruikt. Op het moment dat een effect succesvol is ingeladen wordt de Descriptor toegevoegd aan de lijst van beschikbare descriptors en wordt het event *EffectLoaded* afgevuurd. Als het laden van het effect faalt wordt het event *EffectError* afgevuurd. Voor een verdere omschrijving kan het onderstaande sequentie diagram worden geraadpleegd.

3.7.1 Werking van het inladen van een effect

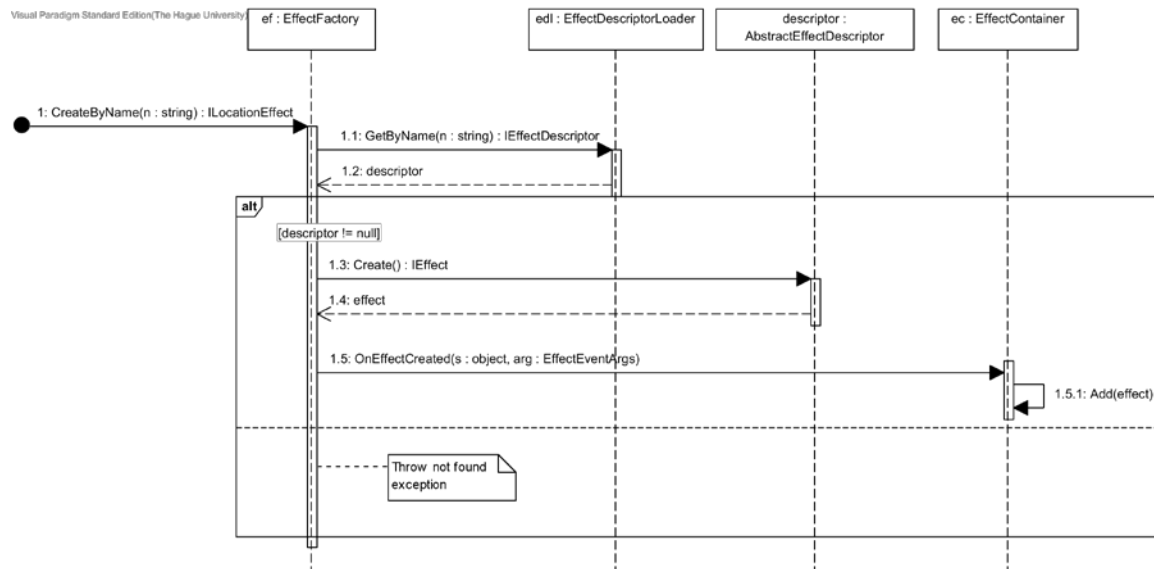
Op het moment dat de applicatie opstart worden de effecten ingeladen. Het inladen van effecten begint bij de *DiscoveryService*. Zie onderstaand sequentie diagram voor de verloop van het inladen van een *EffectDescriptor*.



Figuur 17 Sequentiediagram: inladen van een effect

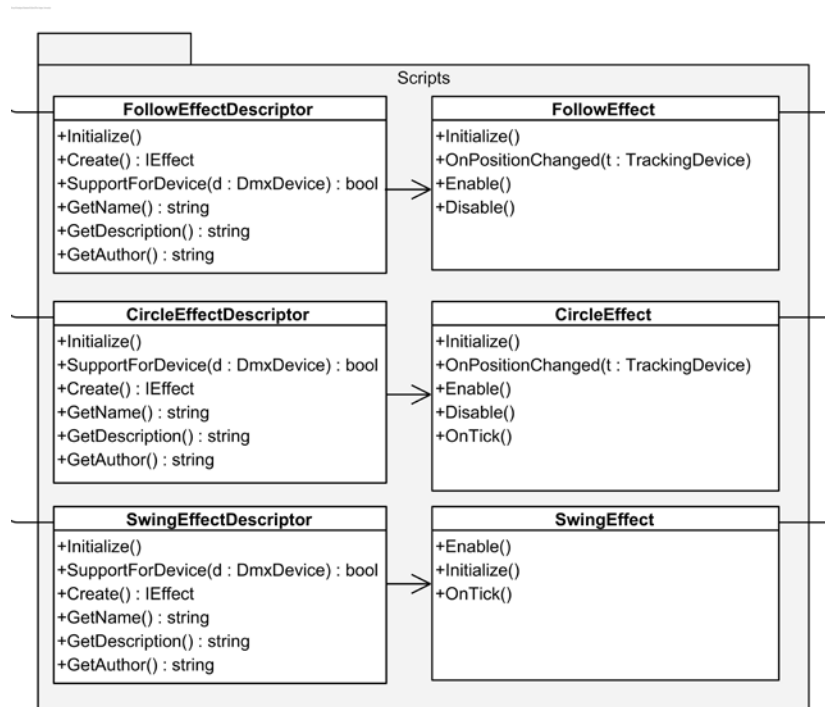
3.7.2 Werking instantiëren van een nieuwe effect

Op het moment dat een EffectDescriptor is ingeladen kunnen er instanties van het effect worden gemaakt. Deze instanties worden gecreëerd door de *EffectFactory*.



Figuur 18 Sequentiediagram: initieer een nieuw effect

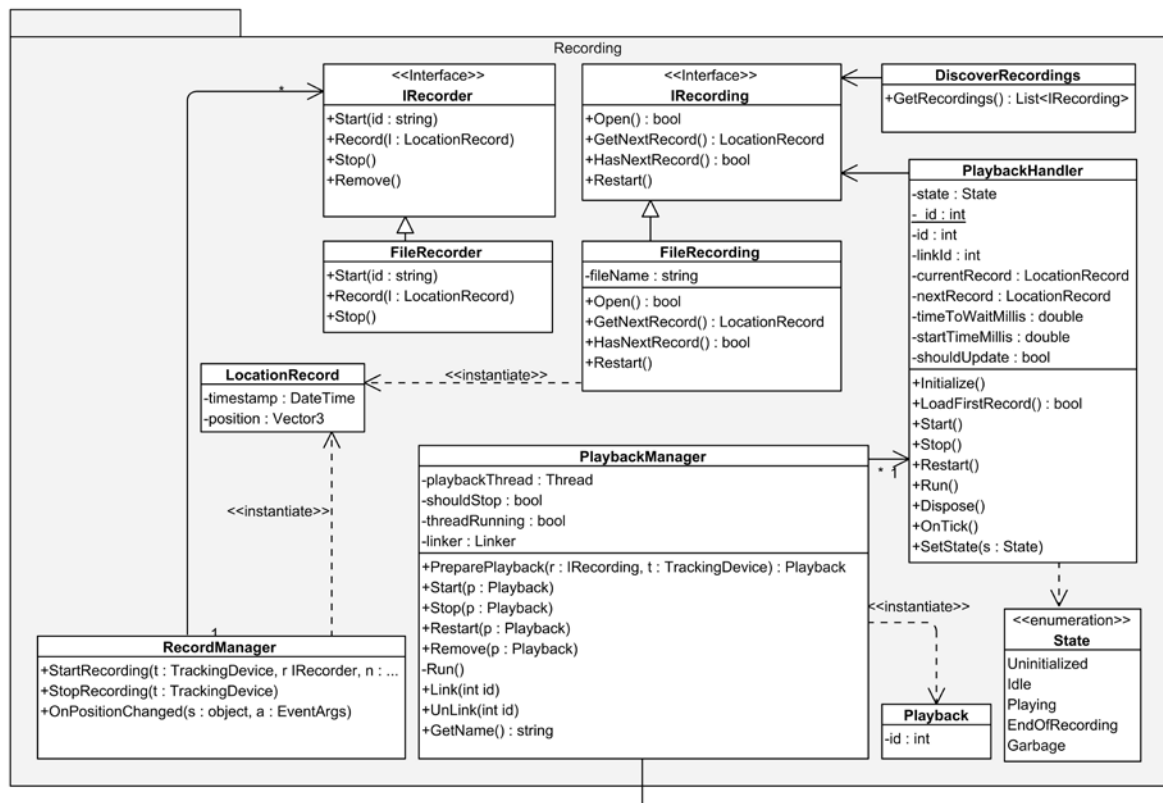
3.8 Scripts



Figuur 19 Klassendiagram: Scripts package

In de *Scripts* package bevinden zich de verschillende effecten van de applicatie. Er zijn op dit moment drie verschillende effecten geschreven. Dit zijn: *FollowEffect*, *CircleEffect* en *SwingEffect*. Deze effecten implementeren de abstracte klassen: *AbstractEffectDescriptor* en *AbstractLocationEffect*.

3.9 Recording

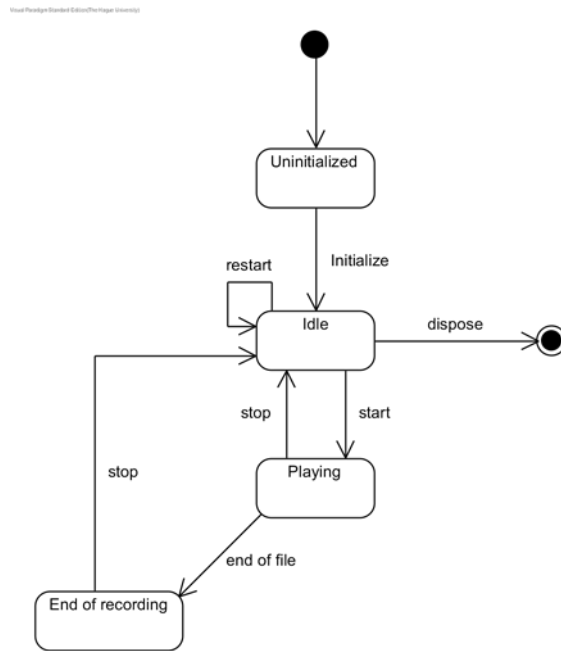


Figuur 20 Klassendiagram: Recording package

De *Recording* package is verantwoordelijk voor het opnemen en later weer afspelen van de locatie veranderingen van een *TrackingDevice*. De klasse *RecordManager* is verantwoordelijk voor het opnemen van de locatie data. Voor het opnemen van data heeft de *RecordManager* een klasse nodig die de interface *IRecorder* implementeert. Op dit moment is dit de *FileRecorder*. De *FileRecorder* schrijft de data weg naar een binair bestand. Het is gemakkelijk om de manier van opslaan in de toekomst te veranderen door een nieuwe klasse te implementeren die de *IRecorder* interface implementeert.

Het afspelen van een opname gebeurt door de *PlaybackManager*. De *PlaybackManager* creëert een instantie van de klasse *PlaybackHandler* op het moment dat er een opname moet worden afgespeeld. De *PlaybackHandler* is verantwoordelijk voor het uitlezen van de opname en het doorsturen van locatie updates naar de *Linker*. Hiervoor maakt de *Handler* gebruik van een klasse die de interface *IRecording* implementeert. Voor deze interface geldt hetzelfde als de *IRecorder* interface. Op dit moment is er een klasse die deze interface implementeert namelijk de *FileRecording* klasse. De *PlaybackHandler* kent een aantal verschillende states, deze zijn te zien in FIGUUR 21. De enige state verandering die 'automatisch' optreedt is die naar de *End of recording* state. De rest van de state veranderingen treden op door acties van de gebruiker. Door een van de methodes van de *PlaybackManager* aan te roepen.

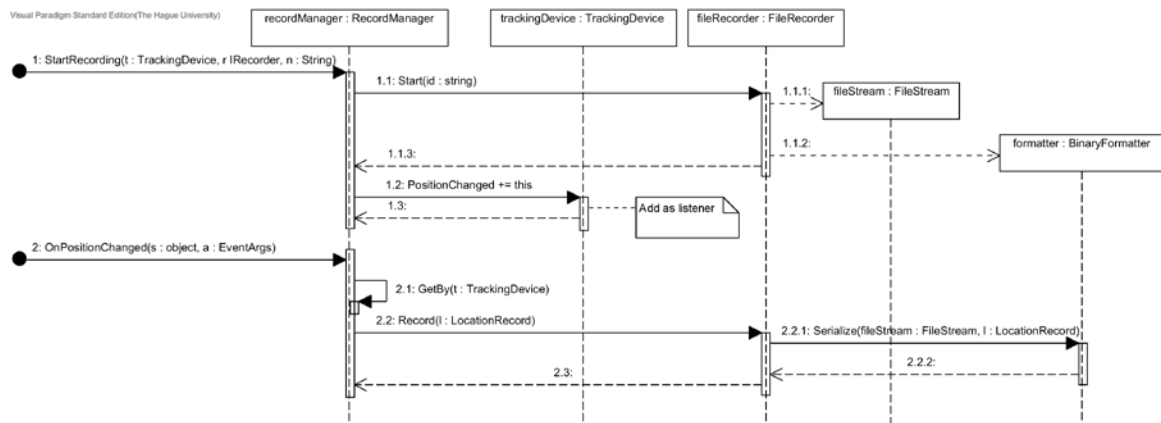
De *PlaybackManager* beheert een lijst van de verschillende actieve *PlaybackHandlers*. Voor het afspelen van de opname gebruikt de *PlaybackManager* een thread. Deze thread roept de *Run()* methode van de verschillende *PlaybackHandler* aan. De *Run()* methode leest de opname uit en bepaald of de positie van een *TrackingDevice* moet worden veranderd.



Figuur 21 Statediagram: PlaybackHandler

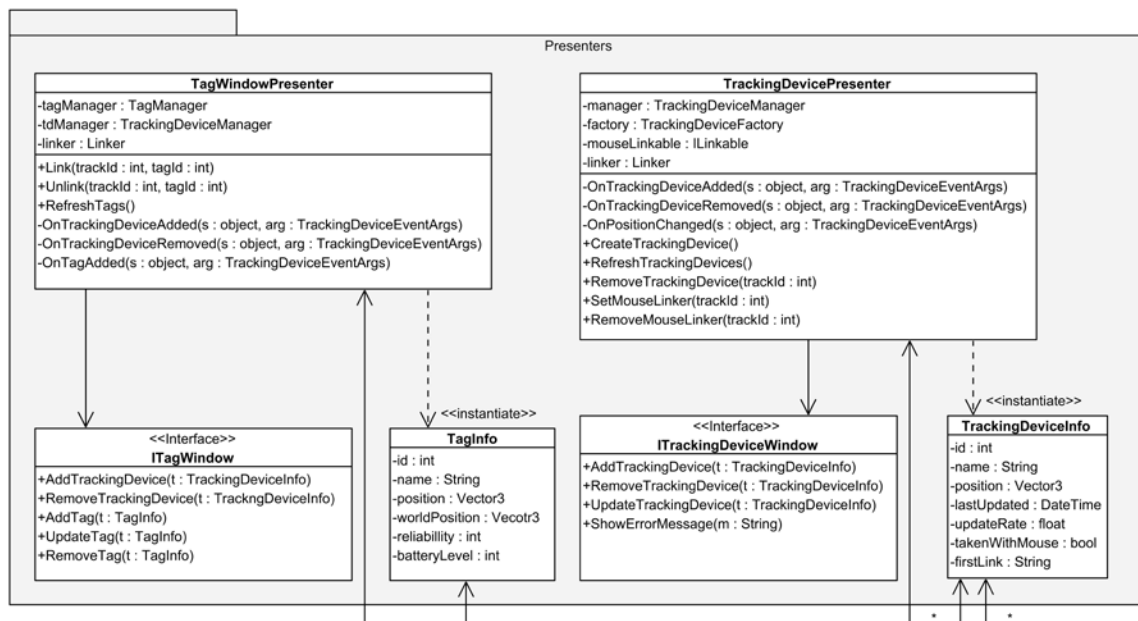
3.9.1 Werking van opnemen

In onderstaand figuur is de werking van het starten van een opname en het opslaan van de data beschreven. De methode *OnPositionChanged()* wordt aangeroepen op het moment dat de positie van een *TrackingDevice* veranderd.



Figuur 22 Sequentiediagram: opnemen

3.10 Presenters



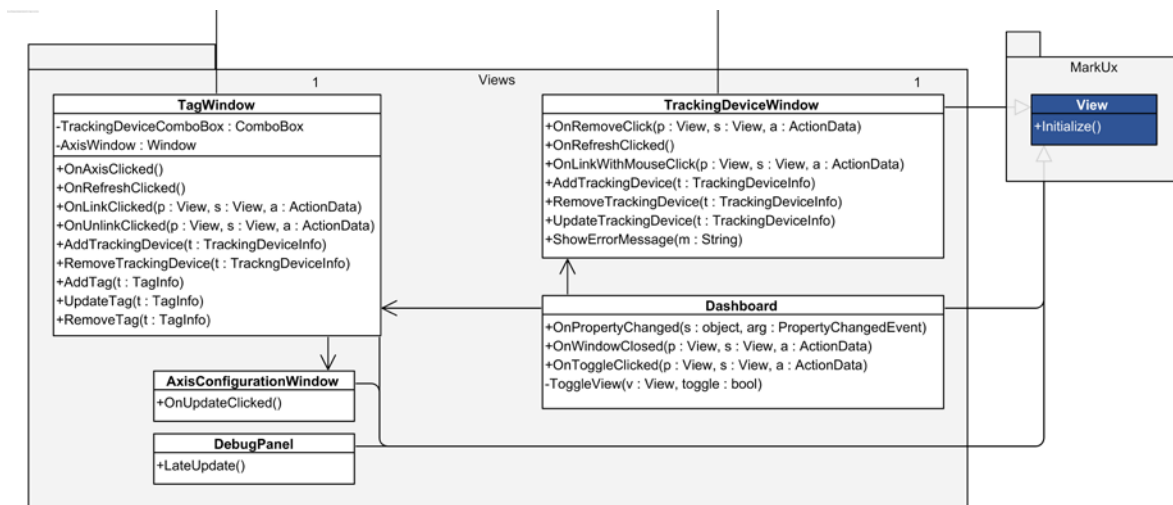
Figuur 23 Klassendiagram: Presenters package

In deze package bevinden zich de verschillende *Presenters* en bijhorende interfaces en models van de applicatie. Deze package zorgt ervoor dat de UI laag is losgekoppeld van de rest van het systeem. Hierdoor is het in de toekomst gemakkelijk om veranderingen in de onderliggende packages door te voeren zonder dat de UI laag hoeft te worden aangepast. De *Presenters* communiceren met de verschillende *Views* van het systeem door middel van interfaces. Hierdoor is het mogelijk om grote delen van de UI te unit testen zonder tussenkomst van een gebruiker die op knoppen hoeft te drukken.

Over het algemeen heeft elke *View* een eigen *Presenters*; de *Presenters* communiceert met de *View* door middel van een interface. Op het moment dat een *Presenters* wordt gecreëerd abonneert de controller zich op de gewenste events van het systeem.

Om te voorkomen dat *Views* kennis nodig hebben van de verschillende business models van de applicatie worden deze in de *Presenters* package vertaald naar *InfoModels*. Deze *InfoModels* (op dit moment zijn dit *TagInfo* en *TrackingDeviceInfo*) worden vervolgens doorgegeven aan de *View* package om ze zo bijvoorbeeld in lijsten te tonen. De *InfoModels* bevatten over het algemeen dezelfde informatie als de business models maar dan zonder operaties. Sommige van de attributen van de *Models* worden door een *Presenter* klasse 'on the fly' bepaalt; dit geldt bijvoorbeeld voor de *takenWithMouse* en *firstLink* (de naam van de link) attributen van de *TrackingDeviceInfo* klassen.

3.11 Views



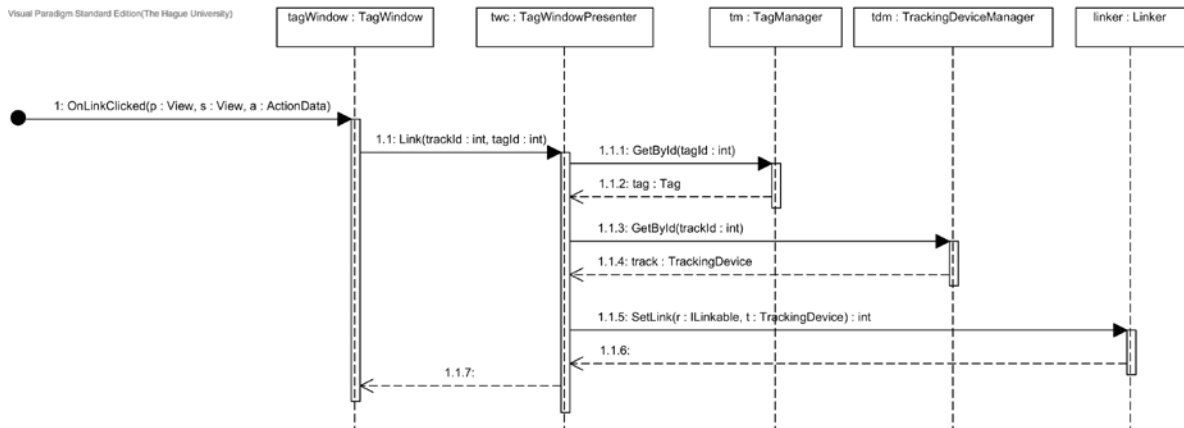
Figuur 24 Klassendiagram: View package

In deze package bevinden zich de verschillende views van de applicatie. Voor het tonen van de user interface maakt de applicatie gebruik van het MarkUX framework. MarkUX maakt gebruik van het MVVM ontwerppatroon; [12] hierbij worden de views geschreven in XML en de view logica in zogenoemde *ViewModels*. Om een *ViewModel* te implementeren moet er van de *View* klasse in de MarkUX package worden overerft zoals te zien is in bovenstaand figuur.

Om de koppeling van de view code met de rest van het systeem zo laag mogelijk te houden wordt er gebruik gemaakt van de al eerder beschreven *Presenters* package. Deze package vormt een bridge tussen de business logica en de view logica van de applicatie. De *ViewModels* vangen events van de gebruiker af (zoals bijvoorbeeld muis klikken) en vertalen deze naar acties op de desbetreffende *Presenter* klasse. De *Presenters* communiceren op hun beurt weer met de desbetreffende view door middel van een interface. Op dit moment zijn er twee Views die gebruik maken van deze constructie, dit zijn: *TagWindow* en *TrackingDeviceWindow*. Dit zijn respectievelijk twee venster waardoor de *Tags* en de *TrackingDevices* van het systeem kunnen worden beheert. Deze vensters worden beheert door de *Dashboard* klasse. De *Dashboard* klasse is verantwoordelijk voor het tonen en verbergen van de verschillende vensters.

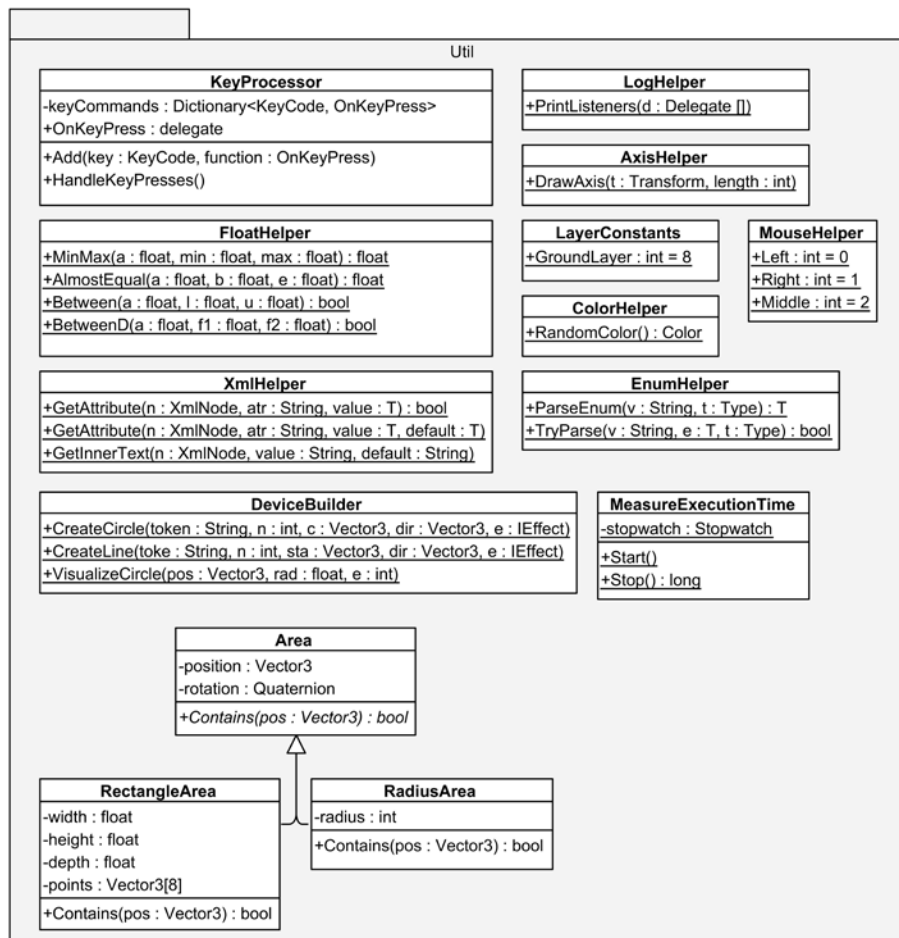
3.11.1 Tag linken met een TrackingDevice

Voor de werking van het linken van een *Tag* aan een *TrackingDevice* kan het onderstaande sequentie diagram worden geraadpleegd. Het resultaat van deze actie is dat het gelinkte *TrackingDevice* voortaan wordt geüpdatet door de *Tag* waarmee die gelinkt is. Zie de beschrijving van de *Tracking* package voor een beschrijving van de werking van de *Linker*.



Figuur 25 Sequentiediaagram: Tag linken met een TrackingDevice

3.12 Util



Figuur 26 Klassendiagram: Util package

De *Util* package bevat een set van helper klassen die het gebruik van bepaalde datatypen vergemakkelijkt. Zo bevat deze package klassen om het werken met: XML bestanden, floats, logs, kleuren en enums makkelijker te maken. De meeste klassen in deze package bestaan geheel uit static methodes, dit zorgt ervoor dat de methodes zonder een instantie van de klassen kunnen worden gebruikt. Voor een uitgebreide beschrijving van de klassen kan het commentaar in de code worden geraadpleegd.

De *KeyProcessor* klasse maakt het afhandelen van toetscombinaties in samenwerking met Unity gebruiksvriendelijker door de poll based API van Unity te omsluiten met een event based API. Door gebruik te maken van deze klassen kunnen grote *if else* constructies om te bepalen welke toets is ingedrukt worden voorkomen.

Om te bepalen of een positie binnen een bepaald gebied ligt kan een van de subklassen van de abstracte *Area* klasse worden gebruikt. Op dit moment zijn er twee klassen die gebruikt kunnen worden namelijk de *RectangleArea* klasse, deze klasse maakt het gemakkelijk om te controleren of een punt binnen een rechthoekige vorm in de ruimte ligt en de *RadiusArea*, deze klasse kan worden gebruikt om te controleren of een punt binnen een bepaalde straal (radius) van een punt ligt. Door te overerven van de *Area* klasse kunnen er gemakkelijk meer manieren worden toegevoegd.

3.13 Exceptions

Sommige methodes binnen de applicatie gooien een exception als er een fout optreedt. Mocht een methode een exception gooien dan is dit gedocumenteerd met de exception tag zoals te zien is in Figuur 2. De applicatie kent een aantal eigen implementaties van de Exception klasse. Deze bevinden zich in de HauteTechnique/Core/Exceptions map. Het gaat om de volgende exceptions:

- ClassHasListenersException
- ConfigurationException
- DeviceNotAllowedException
- EffectFullException
- MethodNotSupportedException

Naast de bovengenoemde exceptions maakt de applicatie ook nog gebruik van een tal van standaard exception uit de System package.

Bibliografie

- [1] „Unity,” Unity, [Online]. Available: <http://unity3d.com/>. [Geopend 2 Januari 2016].
- [2] „MarkUX: uGUI Markup Framework,” Unity, [Online]. Available: <https://www.assetstore.unity3d.com/en/#!/content/37466>. [Geopend 2 Januarie 2016].
- [3] „Unity Test Tools,” Unity, [Online]. Available: <https://www.assetstore.unity3d.com/en/#!/content/13802>. [Geopend 2 Januari 2016].
- [4] „Resources,” Unity, [Online]. Available: <http://docs.unity3d.com/ScriptReference/Resources.html>. [Geopend 2 Januarie 2016].
- [5] „XML Documentation Comments (C# Programming Guide),” Microsoft, [Online]. Available: <https://msdn.microsoft.com/nl-nl/library/b2s063f7.aspx>. [Geopend 2 Januarie 2016].
- [6] „NUnit,” NUnit, [Online]. Available: <http://www.nunit.org/>. [Geopend 2 Januarie 2016].
- [7] „Handling and Raising Events,” Microsoft, [Online]. Available: [https://msdn.microsoft.com/en-us/library/edzhd2t\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/edzhd2t(v=vs.110).aspx). [Geopend 18 December 2015].
- [8] „Observer Design Pattern,” SourceMaking, [Online]. Available: https://sourcemaking.com/design_patterns/observer. [Geopend 18 December 2015].
- [9] „Singleton Design Pattern,” SourceMaking, [Online]. Available: https://sourcemaking.com/design_patterns/singleton. [Geopend 18 December 2015].
- [10] „Prefabs,” Unity, [Online]. Available: <http://docs.unity3d.com/Manual/Prefabs.html>. [Geopend 18 December 2015].
- [11] „Facade Design Pattern,” SourceMaking, [Online]. Available: https://sourcemaking.com/design_patterns/facade. [Geopend 21 December 2015].
- [12] „The MVVM Pattern,” Microsoft, [Online]. Available: <https://msdn.microsoft.com/en-us/library/hh848246.aspx>. [Geopend 21 December 2015].

