

30
mei
2016

Vision2Watch ITable systeem

Afstudeerverslag Final (Herkansing)

Student

Naam: Chris Reinking

Studentnummer: 08057575

Opleiding: Informatica

Bedrijf

Naam: Vision2Watch

Bedrijfsmentor: Dhr. D. Frencken

Haagse Hogeschool

Examinatoren

Dhr. G.M. Tuk

Mevr. A.M.J.J. Lousberg-Orbons (Begeleidend)

Referaat

Auteur: Chris Reinking

Titel: Vision2Watch ITable systeem

Onderwerp: Het ontwikkelen van het ITable systeem

Dit afstudeerverslag beschrijft mijn afstudeeropdracht bij Vision2Watch. In dit verslag zijn de activiteiten beschreven ten behoeve van het ontwikkelen van het ITable systeem. De software is geschreven in de C++ programmeertaal gebruik makende van de Visual Studio 2015 IDE (Integrated Development Environment). Verder is gebruik gemaakt van meerdere Software Development Kits (SDK)

In dit document komen de volgende technieken, tools en Frameworks voor:

- Hardware
 - Intel Realsense R200 Sensor Camera
 - Intel Realsense F200 Sensor Camera
 - Kinect voor Xbox One Sensor
 - Kinect voor Xbox 360 Sensor
 - Kinect voor Windows Sensor
- Software
 - Visual Studio IDE
 - Eclipse IDE
 - Intel Realsense SDK
 - Kinect SDK
 - UMLet
 - Paint.NET
 - Point Cloud Library (PCL)
 - Usbwebserver (MySQL)

Voorwoord

Dit afstudeerverslag beschrijft de werkzaamheden die ik tijdens mijn afstudeerperiode bij Vision2Watch heb verricht. De afstudeerperiode loopt van 20 oktober 2015 tot en met 20 maart 2016. Tijdens het afstuderen heb ik het ITable systeem ontwikkeld.

Ik bedank hierbij graag de volgende mensen, die het mogelijk hebben gemaakt om mijn afstudeerstage succesvol te kunnen doorlopen.

Vision2Watch:

- Dhr. D. Frencken (CEO & Bedrijfsmentor)

Haagse Hogeschool:

- Mevr. A.M.J.J. Lousberg-Orbons (1ste & Begeleidend Examiner)
- Dhr. G.M. Tuk (2de Examiner)

Chris Reinking

Rijswijk, maart 2016

Inhoudsopgave

1. Inleiding.....	1
2. Organisatie.....	2
3. De Beginsituatie.....	3
4. Inwerken.....	4
5. Plan van Aanpak.....	6
6. Requirements.....	7
7. User Stories.....	10
8. Storyboard.....	12
9. Architectuur van het Systeem.....	16
10. De Relatieve Database.....	18
11. Entity Relationship Diagram (ERD).....	19
12. Wijziging camera.....	20
13. Klassendiagram Proof-of-Concept.....	22
14. Toestand Diagram PoC.....	25
15. Testplan Proof-of-Concept.....	26
16. Proof-of-Concept.....	28
17. Klassendiagram uitbreiding.....	32
18. Testplan Uitbreiding.....	34
19. Uitbreiding.....	36
20. Afwijkingen Afstudeerplan.....	45
21. Evaluatie.....	46
22. Conclusie.....	49
23. Bijlagen.....	I

1. Inleiding

Als laatstejaars-student van de Informatica opleiding van de Haagse Hogeschool moet ik voor het afstuderen bij een bedrijf een HBO-waardige opdracht voltooien. De opdracht die ik hiervoor koos was het ITable systeem bij het bedrijf Vision2Watch waar ik 19 weken lang heb mogen werken.

In dit document zal ik het volgende behandelen:

1. De Organisatie: In dit hoofdstuk leg ik uit hoe het bedrijf Vision2Watch is georganiseerd en wat de geschiedenis van het bedrijf is.
2. De Beginsituatie: In dit hoofdstuk leg ik de stand van zaken bij Vision2Watch uit bij het begin van mijn afstudeerstage en vervolgens wat de doelstelling van het ITable systeem is.
3. De Activiteiten: In de hoofdstukken 4 t/m 15 leg ik uit welke activiteiten ik heb doorlopen. Ik vertel over wat het doel van het product is, welke problemen ik tegenkwam en welke oplossingen ik daarvoor heb bedacht. Verder vindt u in elk hoofdstuk ook een voortgangsplanning terug.
4. Afwijkingen Afstudeerplan: 'Geen enkel plan overleeft contact met de vijand' - Helmuth Karl Bernhard von Moltke¹, zo is het ook in de IT. In dit hoofdstuk geef ik aan welke obstakels ik tijdens het project ben tegengekomen die ertoe leiden dat de opdracht anders liep dan ik in het afstudeerplan heb aangegeven.
5. Evaluatie: In dit hoofdstuk blik ik terug op mijn handelen tijdens het project. Ik kijk per activiteit wat ik heb gedaan en wat mijn gedachtegang was. Als laatste behandel ik de beroepstaken die ik heb moeten vervullen tijdens het afstuderen.

1. <http://nl.wikipedia.org/wiki/Helmuth>

2. Organisatie

Vision2Watch is een bedrijf dat audiovisuele midelen levert aan bedrijven. Zo hebben ze bij New Babylon in Den Haag o.a. een LED Wall geleverd. Ze zijn sinds 2007 actief en binnen de Internationale markt bekend als een bedrijf dat uitstekende kwaliteit audiovisuele midelen levert.

Het bedrijf is gevestigd in Rijswijk en heeft een klein team van mensen die elk een eigen expertiseveld hebben. Tijdens de periode dat ik bij Vision2Watch afstudeerde bestond het team uit de CEO, een Content Designer, een parttime Administratief Medewerker en mezelf. Begin november 2015 kwam er nog een stagiair bij.

Een paar voorbeelden van de producten van Vision2Watch zijn de Virtual Host(ess) en de Interactieve katapult. De Virtual Host(ess) (afbeelding 1) is een holografische host die informatie kan geven aan voorbijlopend publiek. Elke host wordt afgestemd op wat de klant wil en er worden acteurs of actrices voor ingehuurd om de geprojecteerde video te maken. Elke host heeft daarom ook de vorm van de desbetreffende acteur of actrice.

De Interactieve katapult (afbeelding 2) is een houten constructie met ingebouwde laser en printplaat. Het heeft een elastiek eraan verbonden waardoor het kan reageren op de gebruiker. Ook zit er een iPod ingebouwd waarop de gebruiker zijn/haar tekst kan invoeren. De gebruiker trekt vervolgens aan het elastiek en mikt met de laser op de bijbehorende projectie. Als de gebruiker vervolgens het elastiek weer loslaat verschijnt de tekst groots op de projectie. Dit product wordt vooral gebruikt bij stands van bedrijven op beurzen.



Afbeelding 1 Holografische Hostess



Afbeelding 2 Interactieve Katapult

3. De Beginsituatie

3.A. De situatie

Mijn afstudeeropdracht bij Vision2Watch bestaat uit het ontwikkelen van het ITable Systeem. Dit is een nieuw systeem waarvoor geen kennis bij het bedrijf aanwezig is. Hierdoor ben ik van mezelf en het internet afhankelijk voor de vragen die ik heb met betrekking tot de code. De producten die bij Vision2Watch worden gemaakt zijn gebaseerd op een Engine waarbij geen kennis van de achterliggende code nodig is. Dit heeft ertoe geleid dat ze voor een nieuw product waarbij de Engine niet wordt gebruikt externe kennis nodig hebben.

Vision2Watch levert verschillende interactieve producten en wil er een nieuw product aan toevoegen. Ze leveren hun producten vooral in het reguliere bedrijfsleven en willen nu ook uitbreiden naar het restaurantleven. Iedereen is wel eens in een restaurant geweest waar een familie zit met kleine kinderen die rumoerig zijn. Vision2Watch heeft een visie van een tafel die deze kinderen bezig kan houden. Hierdoor kunnen de volwassenen zelf ook beter genieten van hun avondje uit.

Vision2Watch wil de interactieve tafels door middel van projectie en bewegingsdetectie toepassen op normale tafels waarin geen hardware is verwerkt. De bedoeling is dat het systeem op elke willekeurige huis, tuin en keuken tafel zonder hardware kan reageren en zich in bijvoorbeeld een lampenkap boven deze tafel bevindt. Momenteel bestaan Touch-Tafels al met het E-Table™ systeem van Inamo™² in een restaurant in Groot-Brittannië, echter vereist dit speciale (E-Table™) tafels.

3.B. De Doelstelling

Doelstelling van het project is het ontwikkelen van de software voor het ITable systeem. Hierbij moet de Motion Detectie en Projectie op elkaar reageren. Tijdens mijn onderzoek voorafgaande aan het afstuderen blijkt dat er gebruik zal moeten worden gemaakt van een Sensor Camera met bijbehorende Software Development Kit (SDK) en een losse projector. Verder moet op basis van de bewegingen en gebaren van de gebruikers een specifieke reactie teweeg worden gebracht. Ook moet deze software kunnen aansluiten op de overige software en hardware van Vision2Watch.

3.C. Aanpak

Het bedachte ITable systeem is opgesplitst in een Proof-of-Concept en een uitbreiding op dit Proof-of-Concept. Elk product van dit systeem heeft weer andere eisen waarmee wordt bepaald of het af is.

1. Een Proof-of-Concept waarin de detectie en projectie op elkaar reageren met daarbij een kleurplaat om in te kleuren met de bewegingen van de handen.

De bedoeling van dit product is het aantonen dat er met behulp van de SDK en Sensor Camera een systeem kan worden gebouwd dat aan de eisen voldoet. Ook wordt hier gekeken naar het registreren van gebaren voor meer gedetailleerde reactie mogelijkheden.

2. De uitbreiding bevat een module waarmee nieuwe spelletjes en bezigheden aan het systeem kunnen worden toegevoegd en bevat direct al een nieuw spel.

Om de gebruikers bezig te kunnen houden moet er variatie zijn in wat er met de tafel kan worden gedaan. De gedachten voor deze variatie bestaan uit verschillende spelletjes zoals bordspelen, sudoku of pong en bezigheden zoals een krant of een nieuwssite.

Het spelletje wat er als eerste inkomt is pong, omdat Vision2Watch dit spel al in een ander product heeft zitten.

2. <http://www.inamo-restaurant.com/gallery.php>

4. Inwerken

Als je ergens gaat werken heb je een periode aan het begin waar je jezelf inwerkt in de omgeving en de opdracht waarvoor je bent aangenomen. Dit is ook het geval voor mij bij Vision2Watch. Allereerst onderzoek ik de verschillende mogelijkheden voor de camera sensor bij dit project. Omdat mijn opdrachtgever de heer Frencken het systeem graag met de Intel Realsense SDK wil maken is dit het uitgangspunt. Ik kijk welke camera's hiervoor beschikbaar zijn en welke vereisten ze hebben.

Allereerst stel ik echter de eisen voor de Camera op waarbij ik kijk naar wat de meeste huidige camera's nodig en beschikbaar hebben:

- USB 2.0 ondersteuning voor het dataverkeer
- Minimaal 640x480 Resolutie voor de geleverde beelden
- Kleuren en Diepte Camera's

Vervolgens kijk ik wat ik voor eisen moet stellen aan de SDK van de camera's om ervoor te zorgen dat het mee kan met de tijd. Hierbij kijk ik ook naar mogelijke combinaties tussen SDK's en Camera's en de programmeertalen. Hierdoor kom ik tot de volgende eisen:

- Windows 7 of later OS.
- SDK ondersteunt C++ en/of C#.
- SDK & Camera komen als een geheel pakket.
- De geschiedenis van de SDK's geeft Backwards Compatibility aandacht.

Aan de hand van deze eisen ben ik gaan onderzoeken welke mogelijkheden ik heb binnen de Intel Realsense camera's. Ik heb de keuze gemaakt voor de Intel Realsense SDK, omdat in het sollicitatiegesprek met de opdrachtgever deze mogelijkheid al was genoemd en het pas is uitgebracht. De resultaten van dit onderzoek heb ik hieronder in de sub-hoofdstukken uiteengezet. Voor beide camera's gelden de volgende hardware en software vereisten: een Intel 4^{de} generatie of later processor, Windows 8.1 of later en een USB 3.0 poort, verder wordt de Microsoft Visual Studio IDE of de Eclipse IDE aangeraden voor de SDK zelf.

4.A. R200 Rear-Facing Camera

Bij deze camera kan de omgeving binnenshuis op 30-400 centimeter afstand detecteren en de bewegingen registreren. Dit betekent dat als ik het systeem hiermee ga ontwikkelen ik de camera tot op 4 meter afstand van de tafel kan plaatsen.

Volgens Intel zelf is deze opstelling bedoeld voor “3D Capture: faces, people, and environments; Depth-enhanced Augmented Reality; Depth-enhanced photography and video; Measurement; Face detection and tracking”³. Dit betekent dat hoewel het menselijke bewegingen kan registreren het niet tot op vinger niveau kan detecteren waardoor geen gedetailleerde gebaren kunnen worden gebruikt.

4.B. F200 Front-Facing Camera

Bij deze camera kan de omgeving op 20-120 centimeter afstand detecteren en de bewegingen registreren.

Deze opstelling is bedoeld voor “Full hand-skeletal tracking and gesture control; 3D segmentation; Facial analysis; Depth-enhanced Augmented Reality; Speech; 3D Capture for faces”⁴. Dit betekent dat deze opstelling in tegenstelling tot de R200 wel specifieke gebaren kan detecteren waardoor het mogelijk is het systeem te besturen.

3. <https://software.intel.com/en-us/RealSense/Devkit/>

4. <https://software.intel.com/en-us/articles/intel-realsense-data-ranges>

4.C. SR300 Front-Facing Camera

Deze camera zal begin februari 2016 de F200 camera vervangen, echter is het nog in de voorverkoop en dus niet bruikbaar voor nu. Het kan alles wat de F200 kan maar heeft een bereik van 20-200 centimeter, dit heeft echter geen invloed op het project omdat het pas begin februari 2016 beschikbaar zal zijn.

4.D. Vergelijking Camera's

Voor een beter beeld zet ik in de onderstaande tabel de kosten voor de Sensor Camera's op een rij.

Sensor Camera	Prijs in \$	USB	Min OS	Min Resolutie	Spectra	Gebaren
R200	\$99,-	3.0	Win 8.1	640x480	Kleur, Diepte, Infrarood	Nee
F200	\$99,-	3.0	Win 8.1	640x480	Kleur, Diepte, Infrarood	Ja
SR300	\$129,-	3.0	Win 8.1	640x480	Kleur, Diepte, Infrarood	Ja

Vergelijking Sensor Camera's

4.E. Planning

In de onderstaande tabel vind u de globale planning voor het project. Deze planning heb ik gemaakt op basis van mijn ervaring in andere ontwikkelprojecten binnen de Haagse Hogeschool. Verder geef ik aan het eind van elk (sub)hoofdstuk over een (deel)product of Sprint aan of er één of meer activiteiten zijn voltooid in die periode.

Product	Startdatum	Deadline
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016
Maken Plan van Aanpak	20 oktober 2015	23 oktober 2015
Requirements Opstellen	23 oktober 2015	30 oktober 2015
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015
Database voor ITable	6 november 2015	10 november 2015
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016
Testplan Opstellen voor Uitbreiding	4 januari 2016	8 januari 2016
Bouwen Uitbreiding	11 januari 2016	11 maart 2016
Tests Uitbreiding Uitvoeren	14 maart 2016	17 maart 2016
Testrapportage Uitbreiding Opstellen	14 maart 2016	17 maart 2016

Planning van het ITable project

5. Plan van Aanpak

In dit PvA (bijlage A) heb ik uitgezet op welke wijze ik van plan ben het ITable project door te lopen en wat de vooraf bedachte oplossingen voor het project zijn.

Tijdens het maken van het PvA moet ik ook beslissen welk software ontwikkel methodiek (SOM) ik voor het project ga gebruiken en wat de scope van het project zal zijn.

5.A. Software Ontwikkel Methodiek (SOM)

Gezien de beperkte tijdsduur en de omvang heb ik besloten om het iteratief te doorlopen maar met enkele beperkingen. Als ik het als een waterval zou doorlopen kan ik niet een revisie toepassen op een deel-product als er wat verandert in de omstandigheden.

Om er voor te zorgen dat ik constant een werkend product zal hebben, heb ik ervoor gekozen om de code in Sprints te schrijven. Het idee voor deze Sprints heb ik uit de Scrum⁵ SOM waar ze als Sprints bekend staan en er wordt mee bedoeld dat het korte perioden zijn waaruit een werkend deelproduct komt. In dit geval codeer ik in Sprints van 2 weken.

Als ik de SCRUM SOM zou volgen moet in een sprint worden ontwikkeld en vervolgens moet daar bij voorkeur een unittest op worden uitgevoerd. Echter, wegens hardware afhankelijkheid kan geen unittest worden toegepast op de code. Hiermee bedoel ik dat de SDK niet kan werken zonder dat de camera aanstaat, en bij een unittest wordt die niet aangezet. Daarom heb ik ervoor gekozen om het zonder unittests, maar wel volgens een vooraf opgesteld testplan, te testen waarin verschillende nog te bepalen testmethodieken worden gebruikt.

5.B. Scope

De scope van het project is zeer belangrijk omdat het product groter is dan wat in de gegeven tijdsperiode mogelijk is te ontwikkelen. Hierbij moet worden gekeken wat de eisen van een product zijn en wat haalbaar is voor het product.

In het Proof-of-Concept (PoC) moet de minimale functionaliteit van het systeem naar voren komen. Deze functionaliteit is de basis Actie-Reactie en een basis bezigheid met in dit geval een kleurplaat.

Verder moet gekeken worden welke functionaliteit het belangrijkste is om daarna te ontwikkelen. Zoals ik bij de beginsituatie al heb aangegeven zal er naast het Proof-of-Concept ook een uitbreiding op het systeem worden ontwikkeld. Deze uitbreiding zal zoals het er nu naar uit ziet een spelletjes module worden waarmee nieuwe bezigheden kunnen worden toegevoegd aan het systeem.

5.C. Voortgang

Product	Startdatum	Deadline	Product Af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Maken Plan van Aanpak	20 oktober 2015	23 oktober 2015	Ja
Requirements Opstellen	23 oktober 2015	30 oktober 2015	Nee
User Stories Opstellen	23 oktober 2015	30 oktober 2015	Nee
Storyboard Maken	23 oktober 2015	30 oktober 2015	Nee

Voortgang PvA

5. <http://www.scrumguides.org/history.html>

6. Requirements

Dit zijn de eisen die de stakeholders van het project aan het systeem stellen. De stakeholders voor dit project zijn de opdrachtgever (D. Frencken) en de projectmanager (ik). De reden dat er bij dit project enkel deze 2 stakeholders zijn is dat het ITable systeem een idee van de opdrachtgever is dat tijdens mijn sollicitatiegesprek bedacht is.

Omdat het product nog geen klant heeft, zijn alle requirements door mij en de heer Frencken opgesteld. Dit betekent dat het aantal requirements niet groot zal zijn en dat we ons moeten verplaatsen in de gedachten en wensen van de toekomstige klanten en gebruikers. Gelukkig is het product duidelijk waardoor ik mezelf makkelijker in kan beelden wat hun eisen zouden zijn.

Wat ik hierbij wel moet aangeven is dat het ITable systeem eerst wordt gemaakt om te kijken of het idee haalbaar is. Hiermee bedoel ik dat ik een prototype zal maken van het systeem. Daarna zal worden gekeken of het verkoopbaar kan worden gemaakt of dat het toch niet als een product van Vision2Watch wordt toegevoegd.

1. Haalbaarheid
 - Detectie van bewegingen met een reactie op het scherm
 - Specifieke bewegingen in te programmeren
2. Verkoopbaarheid
 - Het aantal functionaliteiten
 - Goedkoop te dupliceren

In onderstaande tabel 1 ziet u alle requirements. Ik zet uit of de eis Functioneel(F) of Niet Functioneel(NF) is, of het tijdens het afstuderen wordt behandeld in een product, of het de haalbaarheid bevordert en of het de verkoopbaarheid bevordert.

6.A. Overzicht Requirements

ID	Naam	Beschrijving	Stakeholder	Product	Haalbaarheid	Verkoopbaarheid
F01	Actie-Reactie	De Projectie en de Motion Detectie moeten op elkaar reageren	Opdrachtgever	Proof-of-Concept	Ja	Ja
F02	Verschillende Modules	Het moet mogelijk zijn om nieuwe spelletjes en andere bezigheden toe te voegen	Project-manager	Uitbreiding	Ja	Ja
F03	Versleuteling	De projectie moet stil gezet kunnen worden en dan enkel met een sleutelgebaar weer verder gaan	Project-manager	Nee	Nee	Nee
F04	Gebruikersverdeling	Er moet kunnen worden aangegeven hoeveel gebruikers er aan een tafel zitten en of het volwassenen of kinderen zijn	Opdrachtgever	Uitbreiding	Nee	Ja
F05	Gebaren	Er moet een uitbreidbare set gebaren zijn waarop het systeem exclusief reageert	Project-manager	Nee	Ja	Ja
F06	Objecten	Het systeem moet objecten kunnen detecteren en er een berichtje bij kunnen zetten	Opdrachtgever	Nee	Nee	Ja
F07	Etenstijd	Als het eten op de tafel staat moet het systeem de projectie uitzetten en met een sleutelgebaar weer verder gaan	Project-manager	Nee	Nee	Nee
F08	Bestelling	De gebruiker moet op de tafel zijn bestelling kunnen maken	Project-manager	Nee	Nee	Nee
NF 01	Tafel Onafhankelijk	Het systeem moet op elke willekeurige tafel werken	Opdrachtgever	Nee	Nee	Ja
NF 02	Druk Op De Knop	Het systeem moet met één druk op de knop zonder verdere inbreng klaar voor gebruik zijn	Opdrachtgever	Proof-of-Concept	Ja	Ja
NF 03	Menu	Het menu van het restaurant moet getoond kunnen worden	Project-manager	Nee	Nee	Nee
NF 04	Reactietijd	Het systeem moet binnen één seconde reageren op gebaren	Project-manager	Proof-of-Concept	Ja	Nee
NF 05	Tekenen	De kinderen moeten zichzelf met een kleurplaat kunnen bezighouden	Opdrachtgever	Proof-of-Concept	Ja	Ja
NF 06	Nieuws	De Volwassen en tiener gebruikers moeten een krant of nieuwssite kunnen bekijken	Project-manager	Nee	Nee	Ja
NF 07	Spelen	De Gebruikers van de tafel moeten zichzelf met een spelletje bezig kunnen houden	Project-manager	Uitbreiding	Ja	Ja

Tabel 1. Requirements

6.B. Voortgang

Product	Startdatum	Deadline	Product Af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Maken Plan van Aanpak	20 oktober 2015	23 oktober 2015	Ja
Requirements Opstellen	23 oktober 2015	30 oktober 2015	Ja
User Stories Opstellen	23 oktober 2015	30 oktober 2015	Nee
Storyboard Maken	23 oktober 2015	30 oktober 2015	Nee

Voortgang Requirements

7. User Stories

Bij User Stories worden de vragen wie, wat en waarom beantwoord met betrekking tot het product. Bij het Product ITable hebben de vragen vooral betrekking op wat de tafel moet kunnen voor welk soort gebruiker. Ik heb de User Stories opgesteld volgens de template ALS <rol> WIL IK <gebeurtenis> OMDAT <reden>. Net zoals bij de Requirements moet hierbij gekeken worden vanuit verschillende oogpunten. Ik heb de User Stories vanuit de volgende oogpunten opgesteld: De klant (Restaurant) van Vision2Watch, Een jonge gebruiker (tot 12 jaar) en Een volwassen gebruiker (12+).

Ik heb voor deze drie rollen gekozen omdat deze het meest met het systeem om zullen gaan. Ik heb de eindgebruikers opgesplitst in twee categorieën, omdat tieners en volwassenen op andere manieren bezig kunnen worden gehouden dan basisschool kinderen en pre-school kinderen.

In onderstaande tabel kunt u alle bedachte User Stories terugvinden.

7.A. Overzicht User Stories

ID	ALS <rol> WIL IK <gebeurtenis> OMDAT <reden>	Prioriteit
US01	ALS Gebruiker WIL IK dat het systeem op gebaren reageert OM de projectie te veranderen	1
US02	ALS Gebruiker WIL IK een spelletje spelen OM mezelf bezig te houden	6
US03	ALS Gebruiker WIL IK de projectie uitzetten OM geen projectie op mijn eten te hebben	10
US04	ALS Gebruiker WIL IK het systeem pauzeren OM te kunnen eten zonder dat het systeem reageert	9
US05	ALS Gebruiker WIL IK mijn eigen stuk tafel hebben OM zelf te bepalen van ik zie	8
US06	ALS Gebruiker WIL IK een gerecht selecteren OM een bestelling te maken	12
US07	ALS Gebruiker WIL IK de projectie versleutelen OM geen reactie meer te krijgen	13
US08	ALS Gebruiker WIL IK bij een object een weetje getoond krijgen OM weer wat nieuws te leren	14
US09	ALS Gebruiker (kind) WIL IK een kleurplaat op de tafel OM het in te kleuren	4
US10	ALS Gebruiker (volwassene) WIL IK een krant op de tafel OM het nieuws te kunnen lezen	5
US11	ALS Klant WIL IK met één druk op de knop het systeem starten OM zo min mogelijk voorbereidingstijd kwijt te raken	2
US12	ALS Klant WIL IK dat het menu op de tafel getoond wordt OM de kosten voor de papieren versie uit te sparen	11
US13	ALS Klant WIL IK mijn eigen tafel kiezen OM onafhankelijk te projecteren	3
US14	ALS Klant WIL IK een gebruikers limiet instellen OM een tafel goed te verdelen	7

Tabel 2. User Stories

7.B. Voortgang

Product	Startdatum	Deadline	Product Af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Maken Plan van Aanpak	20 oktober 2015	23 oktober 2015	Ja
Requirements Opstellen	23 oktober 2015	30 oktober 2015	Ja
User Stories Opstellen	23 oktober 2015	30 oktober 2015	Ja
Storyboard Maken	23 oktober 2015	30 oktober 2015	Nee

Voortgang User Stories

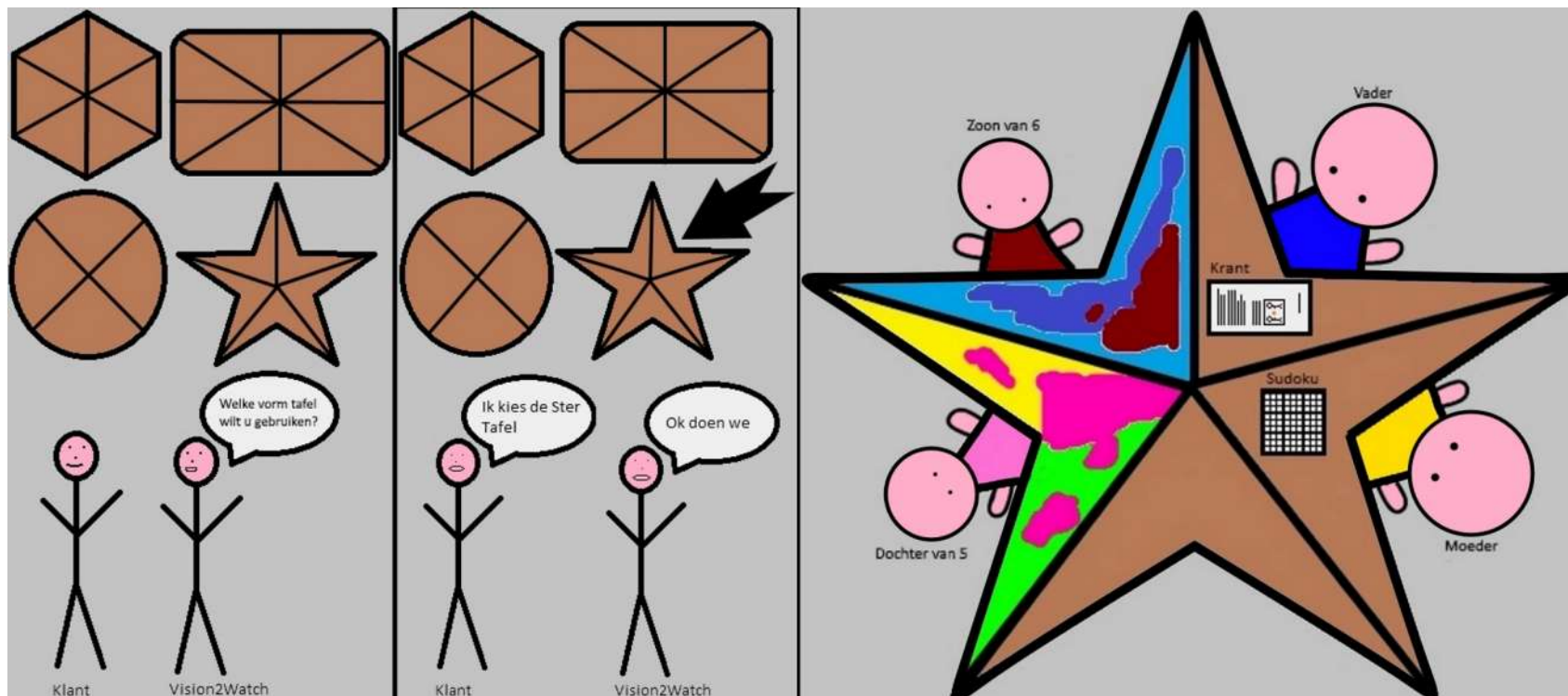
8. Storyboard

Een storyboard is simpel gezegd een visuele representatie van een heel proces. Ze worden vooral gebruikt voorafgaande aan het maken van film, animatie- of ander televisieprogramma. Aangezien Vision2Watch gespecialiseerd is in Audiovisuele midelen worden ook hier storyboards gebruikt om het voor de klant makkelijker te maken om het te visualiseren. Dit is ook de reden dat mij gevraagd is om voor het ITable systeem ook een storyboard te maken.

De tekeningen in het onderstaand [STORYBOARD] zijn bewust simpel gehouden en voorzien van een verklarende tekst. Het gaat hierbij ook om scenario's die niet met het Proof-of-Concept of de uitbreiding kunnen plaatsvinden. Deze scenario's geven echter aan een klant weer wat het potentieel van het systeem is en bevordert daarmee de verkoopbaarheid.

8.A. Bezigheden

In dit storyboard ziet u hoe een tafel wordt gekozen en welke bezigheden er kunnen zijn in het systeem.

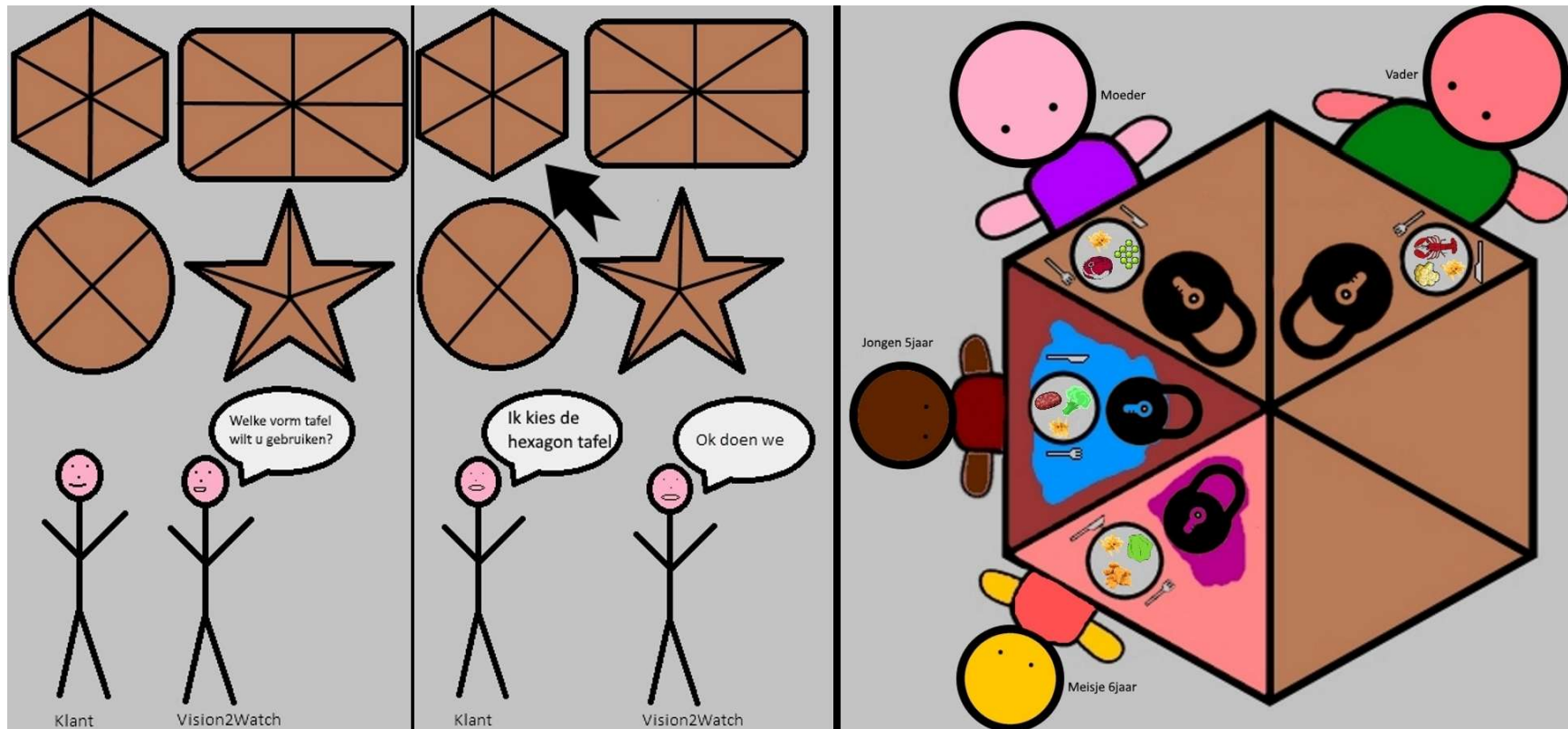


Storyboard 1. Bezigheden

8.B. Versleuteling

In dit storyboard ziet u dat de projectie is versleuteld en dus geen reactie vertoont zolang er geen sleutelgebaar wordt gemaakt.

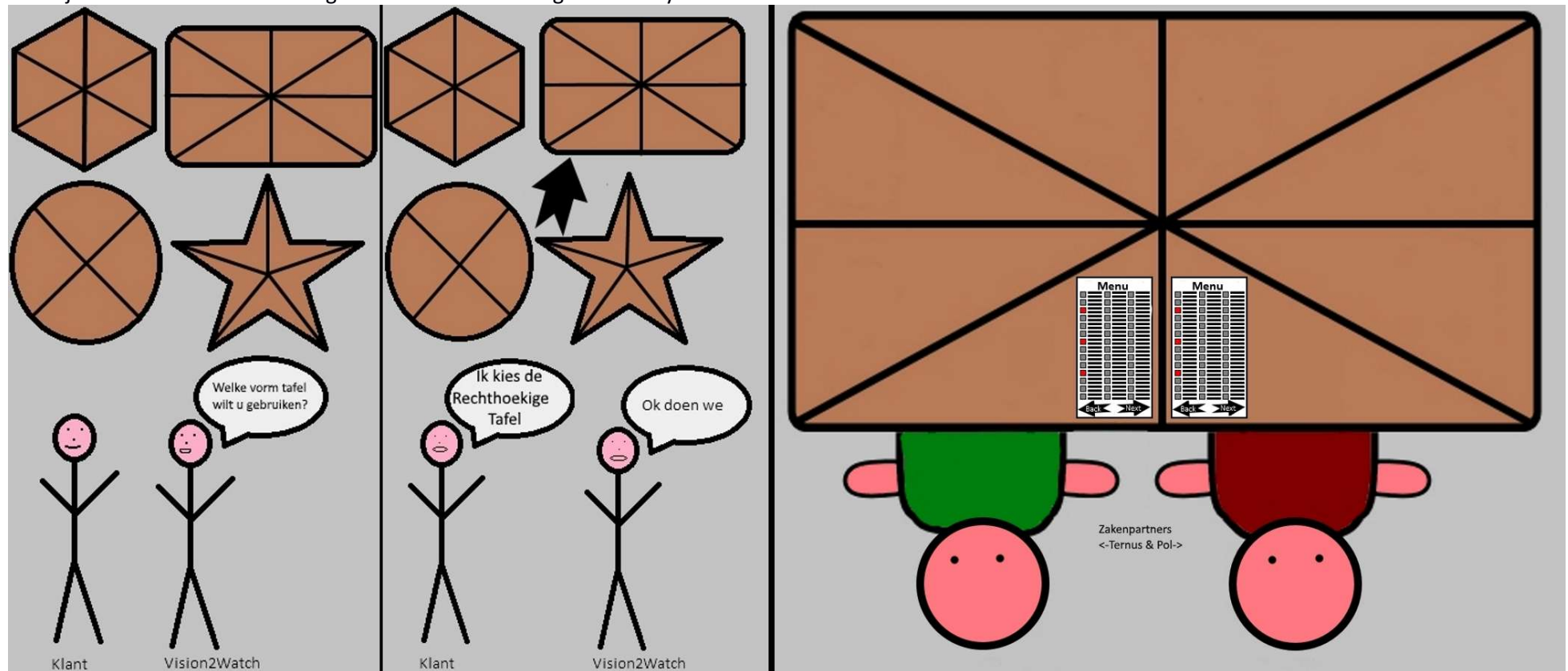
Hierbij is tevens een andere tafel gekozen dan in het voorgaande storyboard.



Storyboard 2. Versleuteling

8.C. Bestelling Plaatsen

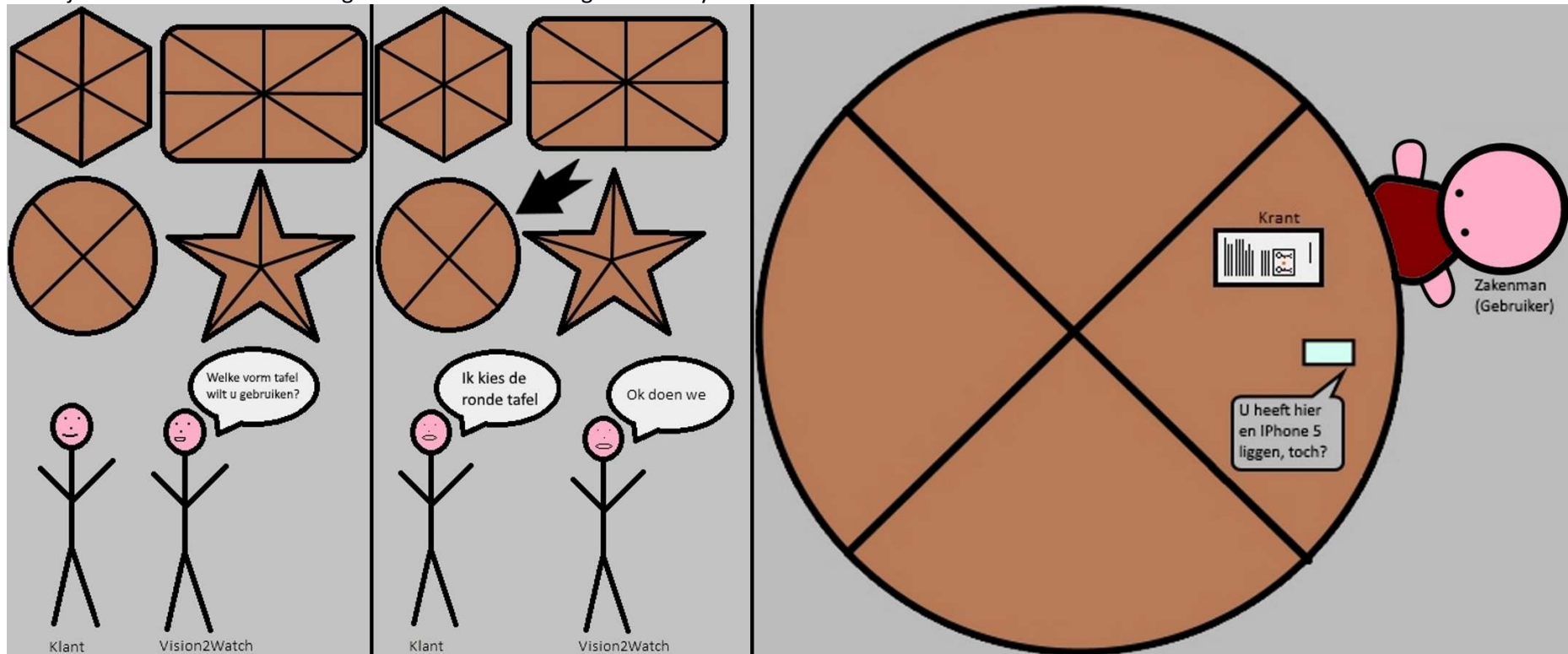
In dit Storyboard ziet u een menu waaruit de gebruikers kunnen kiezen om hun bestelling te maken. Hierbij is tevens een andere tafel gekozen dan in het voorgaande storyboard.



Storyboard 3. Bestelling Plaatsen

8.D. Objectherkenning

In dit storyboard ziet u hoe het systeem een object herkent en er een reactie op geeft.
Hierbij is tevens een andere tafel gekozen dan in het voorgaande storyboard.



Storyboard 4. Objectherkenning

8.E. Voortgang

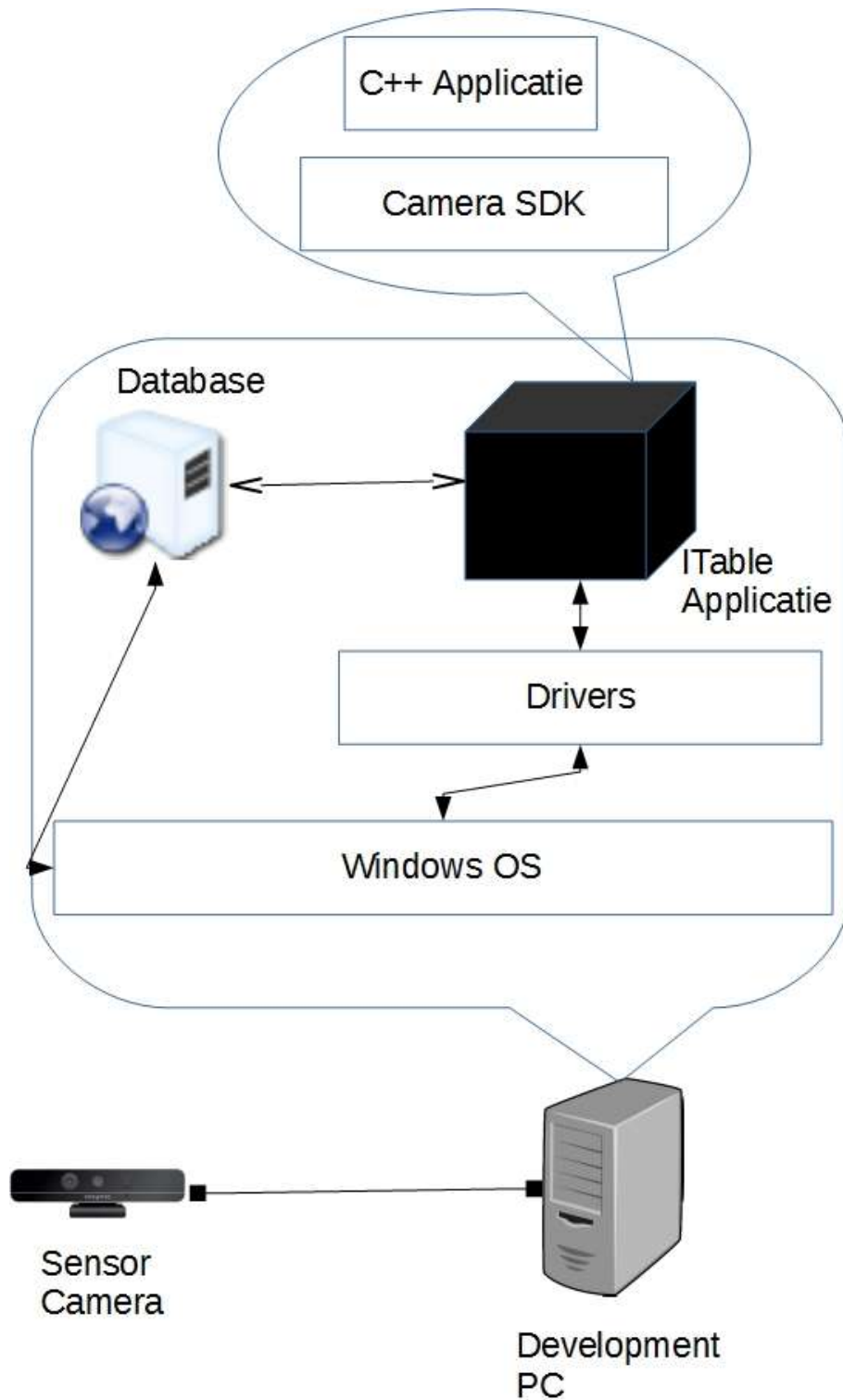
Product	Startdatum	Deadline	Product Af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Maken Plan van Aanpak	20 oktober 2015	23 oktober 2015	Ja
Requirements Opstellen	23 oktober 2015	30 oktober 2015	Ja
User Stories Opstellen	23 oktober 2015	30 oktober 2015	Ja
Storyboard Maken	23 oktober 2015	30 oktober 2015	Ja
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015	Nee

Voortgang Storyboard

9. Architectuur van het Systeem

Voor een beter overzicht van het systeem en de architectuur heb ik een eenvoudig diagram in elkaar gezet waarop te zien is waar de applicatie en de database zich bevinden. Zoals u in het onderstaande diagram kunt zien bevinden zowel de database als de rest van het systeem zich op een fysieke server. Deze server is in het geval van het ITable project mijn werk-pc. Als het product af is zal de fysieke server een klein PC'tje zijn waaraan de overige hardware verbonden zal zitten, denk hierbij aan de projector en de sensor camera.

9.A. Architectuur Diagram



De architectuur van het systeem en de database

9.B. Voortgang

Product	Startdatum	Deadline	Product af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015	Ja
Toestand Diagram voor Testplan	30 oktober 2015	6 november 2015	Ja
Architectuur Diagram	6 november 2015	8 november 2015	Ja
Entity Relationship Diagram	6 november 2015	8 november 2015	Nee
Database voor ITable	6 november 2015	10 november 2015	Nee
Klassendiagram Proof-of-Concept	11 november 2015	13 november 2015	Ja
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016	Nee
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016	Nee

Voortgang architectuur

10. De Relationale Database

Bij een relationele database worden de gegevens opgeslagen in tabellen waarbij de kolommen de attributen zijn (bijvoorbeeld naam, adres, leeftijd) en de rijen de gegevens.

Ik heb ervoor gekozen om dit product nu te maken omdat de camera nog onderweg is vanuit de Verenigde Staten. Ik kan hierdoor niet weten of de camera werkt met de hardware die ik heb en of het communiceert met de SDK.

10.A. Gegevens database

Ik heb ervoor gekozen om bij het ITable systeem gebruik te maken van het opslaan van parameters in een database, zodat het systeem flexibeler is in te stellen, waar dit bij hard-coded parameters niet of nauwelijks mogelijk is. Het maakt de applicatie ook goedkoper in de exploitatiefase. De hardware (Camera) die bij de SDK hoort heeft namelijk verschillende mogelijkheden. Bij de Intel Realsense SDK zijn verschillende resoluties, framerate's en kleurformaten mogelijk voor de binnenkomende camera data. Deze worden bepaald bij het opvragen van de data als een gecombineerde waarde, bijvoorbeeld RGB16:1920x1080:60 oftewel 16-bit kleur op 60 frames per second in 1920x1080 pixels.

Verder is het de uiteindelijke bedoeling dat het systeem meerdere modellen tafels ondersteunt. De meest gangbare modellen heb ik opgenomen in de Database. Om te zorgen dat het mogelijk is om deze tafels ook op locatie te veranderen moet er dynamisch een vorm en formaat worden aangegeven. Deze gegevens zijn daarom het beste op te slaan in een database zodat het niet elke keer hard-coded in het systeem moet worden gezet.

Als laatste bedacht ik me dat er maar een maximum aantal gebruikers aan een tafel zou kunnen zitten en dat dit ook vanaf de bediening moet worden doorgegeven. Zo kan een kleine ronde tafel bijvoorbeeld maar maximaal vier mensen aan terwijl aan een kleine zeshoek er zes zou kunnen zitten.

10.B. Voortgang

Product	Startdatum	Deadline	Product af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015	Ja
Entity Relationship Diagram	6 november 2015	8 november 2015	Ja
Database voor ITable	6 november 2015	10 november 2015	Ja
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016	Nee
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016	Nee
Testplan Opstellen voor Uitbreiding	4 januari 2016	8 januari 2016	Nee
Klassendiagram Uitbreiding	11 januari 2016	13 januari 2016	Nee
Bouwen Uitbreiding	11 januari 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren	14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen	14 maart 2016	17 maart 2016	Nee

Voortgang database

11. Entity Relationship Diagram (ERD)

Voorafgaande aan het ontwikkelen van een database is het nodig dat wordt bepaald hoe het er uit moet gaan zien. Omdat ik voor het ITable systeem een database ga bouwen is het nodig dat ik hiervoor een model opstel. Het ERD is hiervoor uitermate goed geschikt omdat het de relaties tussen de tabellen en de inhoud daarvan samen weergeeft. Met behulp van het MySQL Workbench Programma is het onderstaande ERD midels Reverse Engineering gecreëerd. U ziet hierin de opbouw van de database voor het ITable systeem.

11.A. ERD

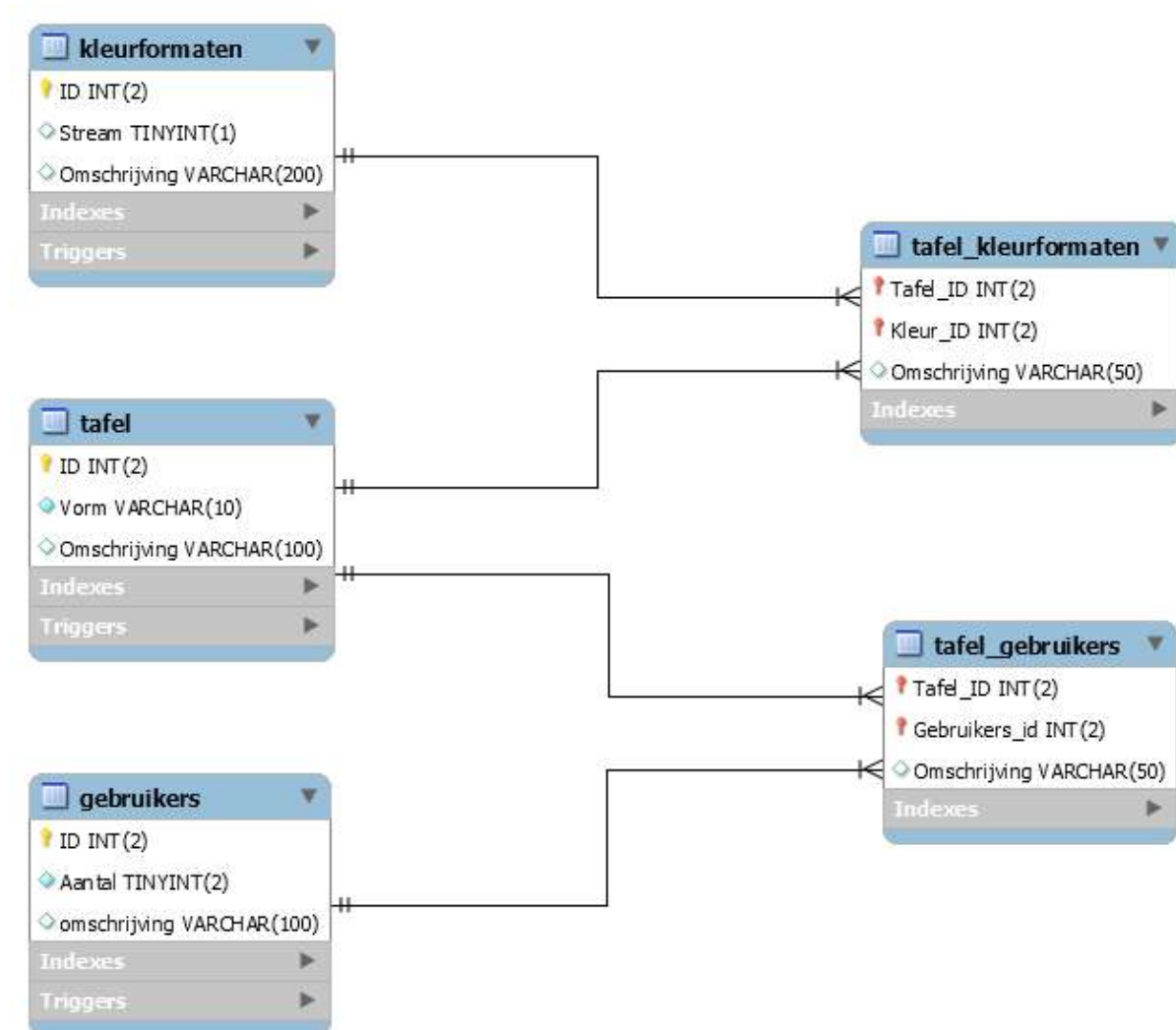


Diagram 3. ERD voor Database ITable

11.B. Voortgang

Product	Startdatum	Deadline	Product af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015	Ja
Toestand Diagram voor Testplan	30 oktober 2015	6 november 2015	Ja
Architectuur Diagram	6 november 2015	8 november 2015	Ja
Entity Relationship Diagram	6 november 2015	8 november 2015	Ja
Database voor ITable	6 november 2015	10 november 2015	Nee
Klassendiagram Proof-of-Concept	11 november 2015	13 november 2015	Ja
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016	Nee
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016	Nee

Voortgang ERD

12. Wijziging camera

Tijdens het ontwikkelen van de Database kwam de Intel Realsense Camera binnen. Deze camera heb ik vervolgens opgezet en geïnstalleerd. Ondanks dat ik alles wat bij de troubleshooting sectie van de SDK tutorial stond heb geprobeerd, lukte het niet om het werkend te krijgen. Dus moest ik snel gaan kijken naar andere mogelijkheden om het systeem mee te bouwen.

Gelukkig was hier een oplossing voor omdat in het magazijn nog een Kinect voor Xbox 360 Sensor Camera ligt. Dit leide ertoe dat ik nu met de Kinect SDK ging werken. Deze SDK had echter maar een enkele resolutie per spectrum, omdat ik de database had gebouwd om ervoor te zorgen dat de gebruiker een keuze kan maken is die nu overbodig. Ik kies er wel voor om de database te behouden voor het geval dat het later wel nuttig zal zijn. Verder had de wijziging van camera & SDK geen invloed op de gemaakte producten.

Ik moet echter eerst onderzoek doen op de Kinect SDK zodat ik zeker weet dat ik de juiste versie gebruik en dat mijn ontwikkelomgeving voldoet aan de eisen. Ook moet ik hierbij kijken naar de sensor zelf omdat er drie verschillende versies op de markt zijn met elk een eigen SDK versie. De resultaten van dit onderzoek heb ik in de onderstaande twee hoofdstukken verwerkt, de hoofdstukken die daarop volgen bevatten het ontwikkelingsproces van het ITable systeem zelf.

12.A. Beta versie met de Kinect for Xbox 360 Sensor Camera

Deze Xbox 360 Sensor opstelling is de enige uit de mogelijkheden die enkel een diepte en kleur detectie heeft zonder infrarood mogelijkheid. Ook is de kleurdetectie beperkt tot een resolutie van 640x480. Alhoewel deze opstelling niet de meest optimale is en ook geen exacte handgebaren kan registreren zit het toch in de race voor het ITable Systeem doordat het in het magazijn op de plank lag.

Helaas bleek na het inpluggen van de Xbox 360 sensor en installeren van de benodigde software voor de Windows versie dat deze combinatie niet werkt en dat de werkende software voor de Xbox 360 versie niet voor commerciële doeleinden mocht worden gebruikt.

12.B. Versie 1.8 met de Kinect for Windows Sensor Camera

Deze v1.8 opstelling bevat de SDK voor de oude Kinect voor Windows Sensor Camera die bedoeld is voor commercieel ontwikkelen in Windows. Deze sensor is gebaseerd op die van de Xbox 360 maar is uitgebreid voor ontwikkelen onder Windows. Er is ondersteuning voor registratie van 2 personen met elk 20 gewrichten die zich op ongeveer 2,5 meter afstand van de sensor bevinden. Hiernaast kan het ook de gebaren van deze personen begrijpen en gebruiken in een applicatie. Verder heeft het ook een infrarood camera maar is de resolutie van de camera's niet verandert.

12.C. Versie 2.0 met de Kinect for Xbox One Sensor Camera

Deze opstelling bevat de meer recente SDK versie 2.0 die gebouwd is voor de Sensor Camera van de Xbox One Gaming Console. Dankzij het bredere zichtveld kan nu van 0,5 tot 4,5 meter afstand beweging worden gedetecteerd. Daarnaast heeft het behalve kleur en diepte ook een Infrarood sensor. Dit geeft dus meer mogelijkheden ten opzichte van de Xbox 360 versie.

Omdat het de nieuwere versie is voor een latere generatie van Gaming Consoles kan deze opstelling tot 6 personen met 25 gewrichten per persoon registreren. Dit betekent dat het handen kan zien maar geen aparte vingers. Hoewel dit betekent dat gebaren beter geregistreerd worden en het ook beter geschikt is voor het ITable Project doet het toch onder voor de Intel Realsense opstelling die specifiek voor gebaarherkenning is gebouwd. Verder is de gezichtsherkenning uitgebreid zodat het een tot 20 keer betere resolutie heeft dan de Xbox360 versie. Hierdoor kan makkelijker worden bijgehouden welke gebruiker zich waar bevindt.

12.D. Vergelijking camera's

Voor een beter beeld zet ik in de onderstaande tabel de specificaties voor de Sensors op een rij.

Sensor Camera	Prijs in \$	USB	Min OS	Min Resolutie	Spectra	Gebaren	SDK
Kinect for Xbox 360	\$109,99	N.V.T.	N.V.T.	640x480	Kleur, Diepte	Nee	Beta
Kinect for Windows	\$189,99	N.V.T.	N.V.T.	640x480	Kleur, Diepte, Infrarood	Ja	V1.8
Kinect for Xbox One	\$99,99	N.V.T.	N.V.T.	1280x720	Kleur, Diepte, Infrarood	Ja	V2.0
USB-adapter (Alle Camera's)	\$49,99	USB 2.0	Win 7	N.V.T.	N.V.T.	N.V.T.	N.V.T.

Vergelijking Kinect camera's

13. Klassendiagram Proof-of-Concept

In dit Klassendiagram wordt visueel aangegeven welke Klassen er in de applicatie zullen zijn, welke attributen en methoden ze zullen bevatten en wat de onderlinge relaties tussen de Klassen zijn.

Omdat voor het systeem gebruik wordt gemaakt van een SDK en ik zonder dat ik er gebruik van heb gemaakt geen beeld kan vormen van een structuur voor het systeem heb ik ervoor gekozen het Klassendiagram op een later tijdstip van de ontwikkeling te modelleren. Dit heeft echter tot gevolg gehad dat er geen verschil zit tussen wat er in het klassendiagram staat en de daadwerkelijke structuur van de applicatie.

In de Kinect SDK die voor het systeem wordt gebruikt zijn vier verschillende detectie mogelijkheden aanwezig met drie ingebouwde camera's.

1. Er is Infrarood camera waarmee verschillen in object-temperatuur kan worden gemeten. Hiermee wordt het mogelijk om te bepalen of een levend wezen in het bereik van de camera zit.
2. Er is een Kleuren camera waarmee verschillen in kleur kunnen worden gemeten. Hiermee kan bij het bepalen van de identiteit van een object of gebruiker beter worden bepaald of het bekend is.
3. Er is een Diepte Laser camera waarmee 3D afstanden kunnen worden gemeten. Dit zorgt ervoor dat de projectie op de juiste locatie verandert.
4. Als laatste is er de Lichaam detectie die gebruik maakt van een algoritme om alle gewrichten te registreren. Hiermee kunnen dan vervolgens gebaren worden opgepikt en op basis van de in geprogrammeerde gebaren komt er een reactie.

Om ervoor te zorgen dat het systeem niet uit een enkele klasse bestaat ga ik elke detectie een eigen klasse geven in het systeem. Ik vul met de gegevens van de detecties aparte Windows om de calibratie van de camera beter te laten verlopen. Omdat dat alle detecties te maken hebben met de SDK hebben ze een gemeenschappelijk en een apart deel waarmee ze werken. Om te zorgen dat ik de gemeenschappelijke delen niet in elke klasse opnieuw hoeft aan te roepen stop ik dat deel in de hoofdklasse (Menu).

Verder heb ik een klasse bedacht waarin het inkleuren van de kleurplaat wordt afgehandeld zodat elke klasse een duidelijke functie heeft. De Menu klasse is de verbinding tussen alle afhandelingen en hierin worden de schermen aangemaakt. De laatste ImageRenderer klasse is de link tussen de Kinect SDK en de rest van het systeem, hierin wordt het beeld voor de schermen gegenereerd op basis van de ingegeven datastream.

Voor het maken van een Unified Modelling Language diagram zoals het klassendiagram is het aan te raden om een tool te gebruiken zodat alle onderdelen al zijn klaargezet. Nou zijn de meeste van deze tools zoals Microsoft Visio en Visual Paradigm duur en bevatten ze meer dan alleen de UML mogelijkheden.

Omdat ik alleen de UML functionaliteit nodig heb en de kosten zo laag mogelijk moeten blijven, heb ik ervoor gekozen om de UMLet⁶ Freeware te gebruiken. Ik ben zowel op school als in mijn voorgaande stages met deze tool in aanraking gekomen en ken de mogelijkheden van deze tool.

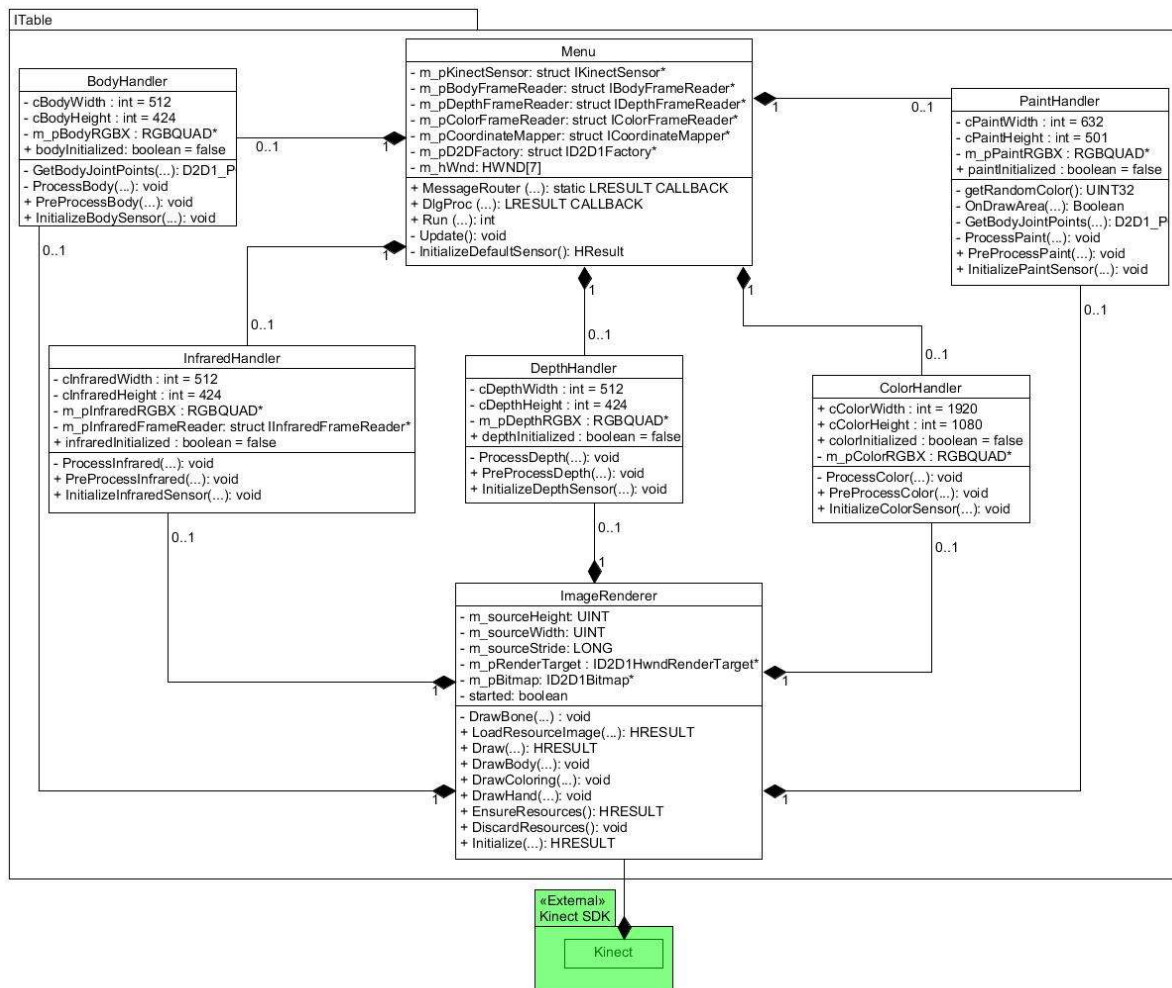
In het onderstaande diagram vind u het Klassendiagram van het Proof-of-Concept.

6. <http://www.umlet.com/>

13.A. Uitleg van de Klassen in het diagram

In het onderstaande diagram ziet u de klassen die in het Proof-of-Concept van het ITable Systeem zitten. Elke klasse heeft zijn eigen functie die ik voor het beter overzicht zal opsommen:

- **Menu:** Deze klasse bevat de Graphical User Interface van het Systeem. Het genereert de Windows waarin de verschillende camerabeelden worden getoond en het Main Window waar de eindgebruikers gebruik van maken.
- **BodyHandler:** In deze klasse wordt de Lichaams-detectie toegepast en de daarbij verkregen data omgezet tot een simpel getekend skelet. Dit skelet is een Stick Figure en wordt in het BodyWindow getoond.
- **ColorHandler:** Deze klasse zet de data van één van de drie camera's om in bruikbare beelden. Deze beelden worden in het ColorWindow gezet en tonen hetzelfde kleurenspectrum als het blote oog kan zien.
- **InfraredHandler:** Ook bij klasse wordt één van de camera's van de Kinect Sensor behandeld. Het zet de data van de Infrarood Camera om naar bruikbare beelden in de InfraredWindow.
- **DepthHandler:** Net als de ColorHandler en InfraredHandler behandelt deze klasse één van de Camera's van de Kinect Sensor. In dit geval gaat het om de Dieptecamera en geeft het zijn beelden aan het DepthWindow.
- **PainterHandler:** Deze Klasse behandelt het inkleuren van het Scherm op basis van de bewegingen van de gebruiker. Het maakt gebruik van de Body Stream die ook bij de BodyHandler wordt gebruikt om de handen te volgen. De uiteindelijke beelden die hier uit komen worden in het Main Window getoond.
- **ImageRenderer:** In deze klasse wordt de data van de Kinect Sensor omgezet naar de data waarmee de Handlers hun werk verrichten. Zonder deze klasse zou er geen manier zijn om de Kinect Sensor aan te spreken.
- **Kinect SDK:** Dit is de Externe Library waarin alle acties van de Kinect Sensor voor gebruik klaar zijn gemaakt



14. Toestand Diagram PoC

In het testplan voor het Proof-of-Concept heb ik een testmethodiek gebruikt waarbij wordt gekeken naar de toestand van het systeem. Voor een eenvoudig overzicht van deze toestanden heb ik er met Unified Modeling Language (UML) een diagram van gemaakt. Niet alle toestanden van het complete ITable systeem zijn in het diagram aanwezig, echter hebben degene die wel aanwezig zijn betrekking op het PoC en de latere uitbreiding.

De basis toestand voor elk systeem is Rust dus die heb ik ook in dit diagram verwerkt. Omdat de ITable zal worden gebruikt in restaurants heb ik een toestand voor het bestellen toegevoegd. Verder heb ik de toestanden voor de bezigheden verwerkt, namelijk spelletjes, nieuws en tekenen. Als laatste heb ik ook nog de 'Op slot' toestand toegevoegd waarbij de detectie alleen nog maar op een sleutelgebaar reageert.

In het onderstaande diagram zijn deze toestanden uitgewerkt.

14.A. Toestandsdiagram PoC

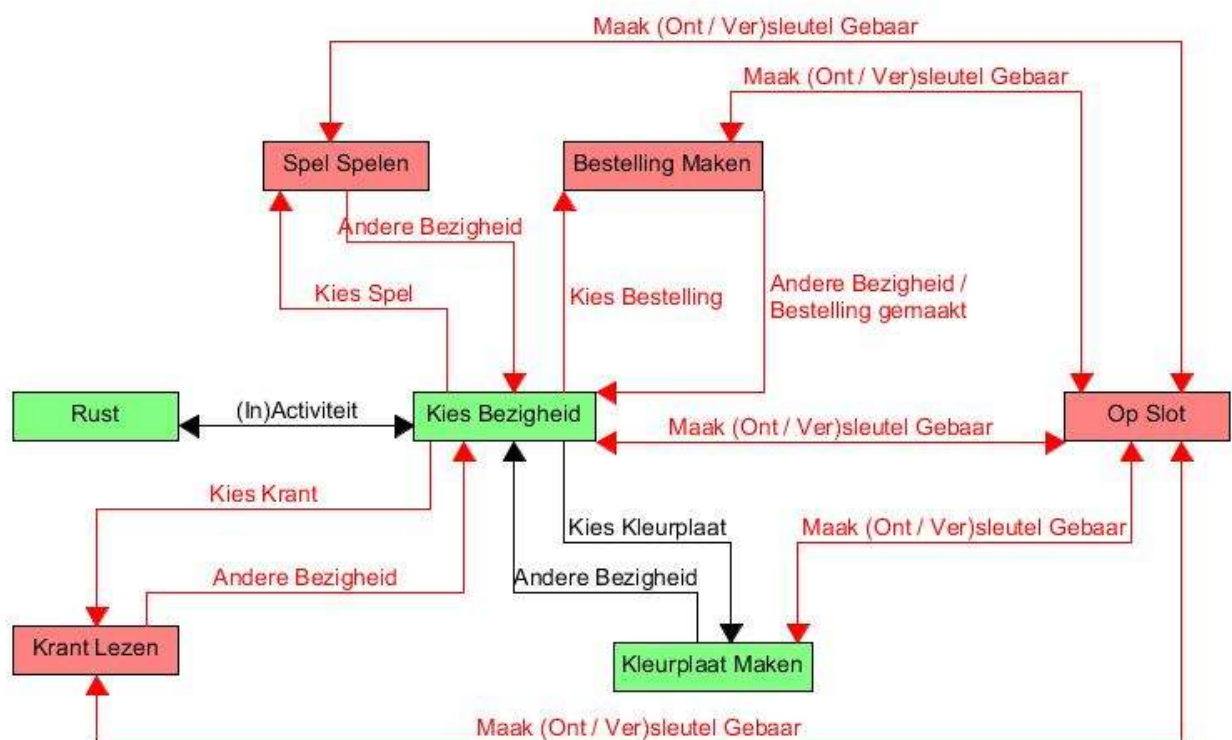


Diagram 2. Toestand Diagram PoC

14.B. Voortgang

Product	Startdatum	Deadline	Product Af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015	Ja
Toestand Diagram voor Testplan	30 oktober 2015	6 november 2015	Ja
Entity Relationship Diagram van Database	6 november 2015	8 november 2015	Nee
Database voor ITable	6 november 2015	10 november 2015	Nee
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016	Nee
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016	Nee

Voortgang Toestandsdiagram PoC

15. Testplan Proof-of-Concept

Het opstellen van een testplan betekent het bepalen wat er getest moet worden en welke testtechnieken daarbij gebruikt zullen worden. Volgens de richtlijnen van de meeste testtechnieken zijn er twee verschillende van elkaar afhankelijke soorten testplannen. Dit zijn het Master Test Plan (MTP) waarin wordt uitgewerkt hoe het gehele testproces moet verlopen en het Detail Test Plan (DTP) waarin de resultaten en specificaties van de testen staan.

Bij het bepalen van de testen voor het Proof-of-Concept heb ik ervoor gekozen om deze twee soorten samen te voegen. Ten eerste bevat het Proof-of-Concept enkel de Actie-Reactie functionaliteit wat in dit geval weinig testwaardige onderdelen oplevert. Verder is door het gebruik van de SDK minder code aanwezig en zal er voor de meeste code gelden dat het afhankelijk is van actieve hardware. Dit leidt ertoe dat er weinig Blackbox en Whitebox testtechnieken zijn die op de applicatie kunnen worden uitgevoerd.

15.A. Samenvatting testplan

In de onderstaande tabel vind u de testdekking van het Testplan voor het PoC. Hierbij geven de groene delen aan dat de test ze dekt en bij rood niet. Het testplan is in de bijlagen opgenomen.

Naam Test	PoC	DB	Omschrijving Test
CRUD-Matrix			Create, Read, Update & Delete de vier basis acties voor een Relationale Database. White Box test.
Syntaxis Test			Word de mogelijkheid op foutieve invoer goed afgedekt? In deze test word bijvoorbeeld een cijfer ingevoerd waar alleen A-z mag om te kijken of het een fout teruggeeft. Black Box Test.
State Transition			Kun je bij alle toestanden van het systeem bijvoorbeeld het versleutelen van het systeem. Black Box Test.
Systeem Test			Bij deze test wordt gekeken of het systeem aan de Requirements voldoen. Black Box Test.

Tabel 3. Dekking Testplan

15.B. Waarom deze Tests?

15.B.I. CRUD-matrix

De Create, Read, Update en Release ofwel CRUD-matrix is een van de basis tests voor een relationele database. Het gaat na of in de interactie tussen de applicatie en de database alle CRUD-acties bij alle tabellen worden behandeld. Als dit namelijk niet het geval is moet dat worden gecorrigeerd zodat de data die in de database staat de juiste mutaties kan ondergaan.

15.B.II. Syntaxis Test

Bij de meeste systemen waarbij een database wordt gebruikt moet men erop letten dat de invoer van data richting de database en uitvoer van data richting de applicatie correct zijn. Dit betekent dat er geen onjuiste informatie mag worden weggeschreven in de database waardoor het systeem niet meer zou functioneren.

In het huidige systeem zou je denken dat dit niet nodig is omdat de User Input alleen via de Sensor Camera gaat, echter is juist deze Sensor Camera het probleem. De Intel Realsense Camera heeft namelijk verschillende instellingen voor de doorgegeven streams, bijvoorbeeld RGB of CMYK kleuren. Dit moet worden afgevangen in het systeem en de database en daarop wordt hiermee getest.

15.B.III. State Transistion

De State Transition of in het Nederlands Toestand Doorloop test welke toestanden het systeem kan bereiken. De basis toestand voor elk systeem is de Rust of uit toestand en vervolgens gaat het dan naar de aan toestand en de andere toestanden van het systeem. Omdat bij dit systeem duidelijke toestanden te vinden zijn zoals een spelletje of een kleurplaat heb ik ook deze test toegevoegd.

15.B.IV. Systeemtest

Hierbij wordt gekeken of het product voldoet aan de vooraf opgestelde requirements, oftewel functioneert het systeem naar behoren. Om aan de opdrachtgever aan te tonen dat het Proof-of-Concept af is heb ik deze test toegevoegd aan het testplan. Ik toon met de resultaten aan of het systeem aan eisen voldoet zodat ik naar de uitbreiding kan.

15.C. Voortgang

Product	Startdatum	Deadline	Product Af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015	Ja
Toestand Diagram voor Testplan	30 oktober 2015	6 november 2015	Nee
Entity Relationship Diagram	6 november 2015	8 november 2015	Nee
Database voor ITable	6 november 2015	10 november 2015	Nee
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016	Nee
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016	Nee

Voortgang testplan Proof-of-Concept

16. Proof-of-Concept

Het doel van een Proof-of-Concept is het opleveren van een werkend product waarmee wordt aangetoond dat het idee kan worden ontwikkeld. Afhankelijk van de requirements van het product kan het Proof-of-Concept meer of minder functionaliteit bevatten.

In het geval van het ITable project heb ik ervoor gekozen om de functionaliteit te beperken tot de minimale vereisten. Deze keuze heb ik gemaakt om elke sprint een enkele functionaliteit te behandelen. Deze functionaliteit was de Actie-Reactie tussen de projectie en de detectie. De projectie wordt ingekleurd met je bewegingen. Om deze inkleur functie te (de)activeren moet je eerst je hand lang genoeg in het juiste gebied houden of op een knop drukken met je gebaar.

Ik heb ervoor gekozen om het Proof-of-Concept zelf op te splitsen in twee functionaliteiten en deze elk in een aparte sprint te voltooien. De eerste Functionaliteit is de detectie zelf, hierbij ga ik ervoor zorgen dat de beelden die de Kinect camera ziet ook op het scherm worden getoond zodat zeker is dat alles goed werkt. De andere functionaliteit is de reactie oftewel de applicatie laten reageren op de bewegingen van de gebruiker. Maar voor dat ik met het bouwen kan beginnen zorg ik er eerst voor dat de Kinect Hardware goed verbonden is en de SDK het ook kan vinden.

16.A. Eerste Sprint Hardware & SDK integreren

Ik heb besloten om de applicatie te schrijven in de C++ programmeertaal met behulp van de Kinect v2.0 SDK. Omdat bij de samples van SDK gebruik wordt gemaakt van de Microsoft Presentation Format (MPF) om de Graphical User Interface (GUI) in elkaar te zetten, heb ik tevens gekozen dit ook te gebruiken.

Om ervoor te zorgen dat het systeem goed werkt vereist het dat ik allereerst de SDK in een Visual Studio project integreer en de Hardware met de SDK kan communiceren. Nadat de Kinect voor Xbox One camera binnen was heb ik het verbonden met de PC waarop ik werk. Vervolgens heb ik de SDK, driver en andere bijbehorende Kinect tools geïnstalleerd en ben gaan testen of ik beeld kon krijgen.

In de tools van de Kinect SDK bevindt zich onder andere een Camera checker, deze tool checkt of de camera met de PC is verbonden en of het voldoende stroom krijgt. Na wat geëxperimenteer kwam ik erachter dat beide het geval waren. Eerst leek het erop dat ik ook deze niet werkend kon krijgen, echter bleek volgens de troubleshooting dat ik de camera als enige op de USB 3.0 hub moet zetten. Ook moet ik ervoor zorgen dat er stroom op de hub staat, hiervoor moet ik de computer herstarten na de initiële start-up. Als ik dit niet doe dan stottert de camera omdat de hub niet genoeg tijd krijgt voordat de OS is opgestart en de camera checkt op gereedheid.

Tijdens deze sprint heb ik nog geen Requirement of User Story behaald.

16.B. Voortgang eerste sprint

Product	Sprint doel	Startdatum	Deadline	Product af
Afstudeerverslag opstellen		20 oktober 2015	20 maart 2016	Nee
Bouwen Proof-of-Concept ITable	Hardware & SDK integreren	11 november 2015	4 december 2015	Ja
	Detectie werkend maken	7 december 2015	18 december 2015	Nee
	Kleurplaat werkend maken met reactie	21 december 2015	1 januari 2016	Nee
Tests Proof-of-Concept Uitvoeren		4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen		4 januari 2016	8 januari 2016	Nee
Testplan Opstellen voor Uitbreiding		4 januari 2016	8 januari 2016	Nee
Klassendiagram Uitbreiding		11 januari 2016	13 januari 2016	Nee
Bouwen Uitbreiding		11 januari 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren		14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen		14 maart 2016	17 maart 2016	Nee

Voortgang Proof-of-Concept 1ste sprint

16.C. Tweede Sprint Werkende Detectie

Allereerst heb ik de verschillende camera's elk een eigen klasse gegeven. Zo heb je de Infrarood camera, de Diepte Laser camera en de kleurencamera. Elk van deze camera's geeft een eigen beeld en zijn ook in een eigen Window weergegeven.

Allereerst ben ik gestart met de kleurencamera en heb hiervoor de tutorial gepakt die in de Kinect SDK beschikbaar is. In deze tutorial wordt uitgelegd hoe je de data van de kleuren camera kan ontvangen en omzetten naar een video die getoond kan worden. Deze tutorial volgend heb ik de kleuren detectie klasse van het ITable systeem gebouwd. Vervolgens ben ik doorgegaan met de Diepte detectie en de Infrarood detectie waarbij voor elk een tutorial beschikbaar was. Bij deze drie wordt in de tutorials duidelijk uitgelegd hoe het werkt en ze gebruiken de raw data van de camera's direct.

Nadat ik deze drie camera's werkend heb gekregen ben ik verder gegaan met het detecteren van de bewegingen. Voor deze detectie heb ik gebruik gemaakt van het Kinect voorbeeld van het getekend skelet. In dit voorbeeld wordt op basis van het ingebouwd algoritme een skelet op het scherm getoond dat de gebruiker voorstelt. Dit skelet volgt de bewegingen van de gebruiker en kan tonen of de gebruiker zijn hand open of dicht heeft. In tegenstelling tot de andere drie detectie mogelijkheden heb ik met deze detectie moeite gehad om het snel te begrijpen en in het ITable systeem te verwerken. Bij deze detectie wordt er namelijk gebruik gemaakt van een algoritme waarbij de gebuikte data eerst moet zijn omgezet uit de raw data. Aan de hand van het algoritme en een andere tutorial kan je verder je eigen gebaren registreren op basis van de gedetecteerde bewegingen. Dit eenmaal werkend gekregen is deze sprint afgelopen en is de eerste functionaliteit voltooid.

Tijdens deze Sprint heb ik de Requirement F01 en de User Story US01 behaald.

16.D. Voortgang tweede sprint

Product	Sprint doel	Startdatum	Deadline	Product af
Afstudeerverslag opstellen		20 oktober 2015	20 maart 2016	Nee
Bouwen Proof-of-Concept ITable	Hardware & SDK integreren	11 november 2015	4 december 2015	Ja
	Detectie werkend maken	7 december 2015	18 december 2015	Ja
	Kleurplaat werkend maken met reactie	21 december 2015	1 januari 2016	Nee
Tests Proof-of-Concept Uitvoeren		4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen		4 januari 2016	8 januari 2016	Nee
Testplan Opstellen voor Uitbreiding		4 januari 2016	8 januari 2016	Nee
Klassendiagram Uitbreiding		11 januari 2016	13 januari 2016	Nee
Bouwen Uitbreiding		11 januari 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren		14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen		14 maart 2016	17 maart 2016	Nee

Voortgang Proof-of-Concept 2de sprint

16.E. Eind Sprint Actie-Reactie

In deze sprint concentreer ik me op de reactie van de applicatie op de gebruiker. Aan het eind van de eerste sprint had ik dit al deels voltooid met het grafische skelet van de lichaamsdetectie maar was dit nog geen volledige interactiviteit. Het doel van deze sprint is om de gebruikers het scherm in te laten kleuren met hun beweging en zelf te kunnen bepalen of ze aan het inkleuren zijn of niet.

Allereerst ben ik ervoor gaan zorgen dat bij een beweging van de hand het scherm wordt ingekleurd zodat er duidelijk iets gebeurt, waarbij ik gebruik maak van de eerder gemaakte skelet klasse. Op basis van deze klasse heb ik een nieuwe klasse gemaakt waarin het kleuren wordt afgehandeld. Het inkleuren eenmaal werkende gekregen ben ik doorgegaan met het uit en aan zetten van deze inkleur functie.

Ik bedacht hierbij een knop te gebruiken echter bleek na meerdere malen proberen dat dit niet kon in combinatie met de kleurplaat. Ik maak namelijk gebruik van MPF om de GUI in elkaar te zetten en de kleurplaat wordt in een Picture Control object uitgelezen. Helaas kan een Button Control object (de knop) niet binnen de afbakening van het Picture Control object zitten of het is niet zichtbaar of bestuurbaar.

Ik ben daarom dus een alternatief gaan zoeken voor een knop en bedacht me dat ik dit met ook met de skelet detectie kon doen. Het idee is namelijk dat als de gebruiker zijn/haar hand binnen een gedefinieerd gebied houdt dan gaat er een teller af die na tien seconden de inkleur functie uit of aan zet afhankelijk van de huidige staat. Dit eenmaal voor elkaar hebbende is de Sprint afgelopen en is het Proof-of-Concept succesvol voltooid.

Als gevolg van deze Sprint zijn de User Story US09 en de Requirement NF05 behaalt.

16.F. Voortgang eindsprint

Product	Sprint doel	Startdatum	Deadline	Product af
Afstudeerverslag opstellen		20 oktober 2015	20 maart 2016	Nee
Bouwen Proof-of-Concept ITable	Hardware & SDK integreren	11 november 2015	4 december 2015	Ja
	Detectie werkend maken	7 december 2015	18 december 2015	Ja
	Kleurplaat werkend maken met reactie	21 december 2015	1 januari 2016	Ja
Tests Proof-of-Concept Uitvoeren		4 januari 2016	8 januari 2016	Nee
Testrapportage Proof-of-Concept Opstellen		4 januari 2016	8 januari 2016	Nee
Testplan Opstellen voor Uitbreiding		4 januari 2016	8 januari 2016	Nee
Klassendiagram Uitbreiding		11 januari 2016	13 januari 2016	Nee
Bouwen Uitbreiding		11 januari 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren		14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen		14 maart 2016	17 maart 2016	Nee

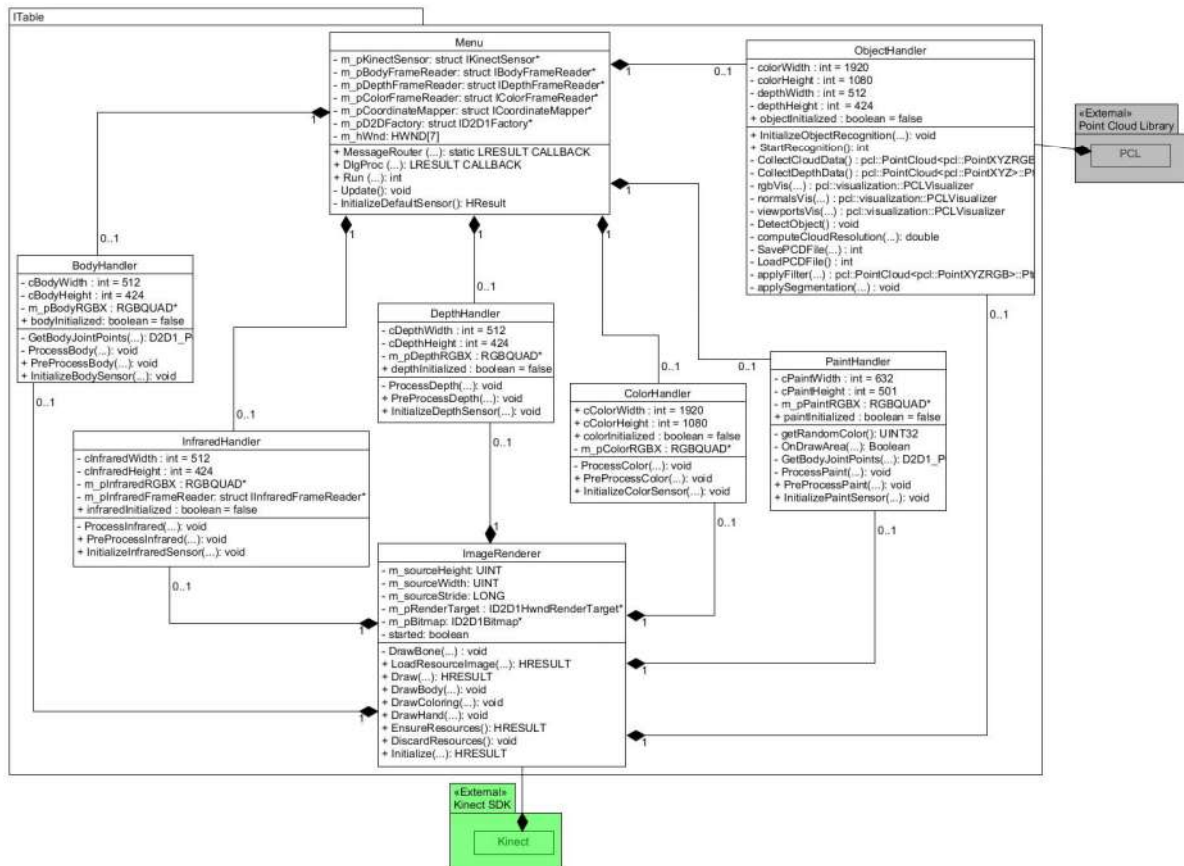
Voortgang Proof-of-Concept Eindsprint

17. Klassendiagram uitbreiding

Net als bij het Proof-of-Concept is ook bij de uitbreiding een nieuwe extern blok code in gebruik. Na overleg met de opdrachtgever is namelijk besloten om in plaats van de geplande spelletjesmodule een objectherkenningsmodule te maken. Helaas heeft objectherkenning verschillende algoritmes afhankelijk van de omgeving waarin het wordt gebruikt. Omdat ik deze algoritmes zelf niet ken en er ook geen tijd is om ze zelf te programmeren is er besloten om het met een Library op te lossen.

Omdat ik ten tijde van het maken van het Klassendiagram voor de uitbreiding nog niet duidelijk heb welke opties er zijn heb ik ervoor gekozen om voor het diagram voor de 3rd party software een aparte package te reserveren. Nadat ik duidelijk had wat ik ging gebruiken heb ik de benaming van deze package aangepast en de Library klasse toegevoegd en in het onderstaande diagram opgenomen.

17.A. Klassendiagram Uitbreiding



17.B. Diagram 4. Klassendiagram Uitbreiding Uitleg van de Nieuwe Klassen

Zoals u in het bovenstaande Diagram kunt zien zijn er maar 2 nieuwe Klassen bij gekomen.

- **ObjectHandler:** In deze klasse wordt het Object Herkenning afgehandeld. Het maakt gebruik van de verschillende Camera's van de Kinect om een Point Cloud te maken. Op deze Point Cloud worden de algoritmes van de PCL toegepast om zo een Point Cloud te maken wat alleen één object bevat. Vervolgens vergelijk het deze Point Cloud met de Point Cloud's van de bekende objecten.
- **Point Cloud Library:** Dit is de Externe PCL Library waarin de algoritmes voor de Object Herkenning en Vergelijking zich bevinden.

17.C. Voortgang

Product	Startdatum	Deadline	Product af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Testplan Opstellen voor Proof-of-Concept en Database	30 oktober 2015	6 november 2015	Ja
Entity Relationship Diagram	6 november 2015	8 november 2015	Ja
Database voor ITable	6 november 2015	10 november 2015	Ja
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016	Ja
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016	Ja
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016	Ja
Testplan Opstellen voor Uitbreiding	4 januari 2016	8 januari 2016	Nee
Klassendiagram Uitbreiding	11 januari 2016	13 januari 2016	Ja
Bouwen Uitbreiding	11 januari 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren	14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen	14 maart 2016	17 maart 2016	Nee

Voortgang Klassendiagram Uitbreiding

18. Testplan Uitbreiding

Voor het maken van dit testplan heb ik er wederom voor gekozen om een gecombineerd testplan te maken. Hoewel ik eerder heb aangegeven dat ik nu wel een MTP en DTP zou maken is het uiteindelijk toch maar een enkel document geworden, dit omdat ik de scope van de uitbreiding niet groot genoeg vind om ze te splitsen.

Ten opzichte van het testplan voor het Proof-of-Concept waarbij er weinig te testen viel is de uitbreiding veel ingewikkelder, omdat er objectherkenning in wordt uitgevoerd. Dit betekent tevens dat ik moet nadenken over hoe ik de objecten registreer en zal onthouden, via een database of anders.

Omdat ik eerst nog moet onderzoeken of ik, met behulp van een SDK of andere 3rd Party software, de object-detectie en -herkenning uit kan laten voeren heb ik het testplan voorafgaande aan het ontwikkelen hetzelfde gelaten als die van het Proof-of-Concept. Tijdens het maken van de uitbreiding heb ik dit testplan op de betreffende code aangepast.

18.A. Samenvatting Testplan Uitbreiding

Voor de uitbreiding geldt dat alle tests alleen op het systeem worden uitgevoerd. Ook dit document is in de bijlagen opgenomen.

Naam Test	Omschrijving Test
State Transition	Kun je bij alle toestanden van het systeem versleutelen
Systeem Test	Bij deze test wordt gekeken of het systeem aan de Requirements voldoet.
Informele test	Bij deze test verander ik de systeem code om te kijken wat het resultaat is.

Tabel 4. Dekking Testplan Uitbreiding

18.B. Waarom deze Tests

18.B.I. State Transistion

De State Transition of in het Nederlands Toestand Doorloop test welke toestanden het systeem kan bereiken. De basis toestand voor elk systeem is de Rust of uit toestand en vervolgens gaat het dan naar de aan toestand en de andere toestanden van het systeem. Omdat bij dit systeem duidelijke toestanden te vinden zijn zoals een spelletje of een kleurplaat heb ik ook deze test toegevoegd.

18.B.II. Systeemtest

Hierbij wordt gekeken of het product voldoet aan de vooraf opgestelde requirements, oftewel functioneert het systeem naar behoren. Om aan de opdrachtgever aan te tonen dat het Proof-of-Concept af is heb ik deze test toegevoegd aan het testplan. Ik toon met de resultaten aan of het systeem aan eisen voldoet zodat ik naar de uitbreiding kan.

18.B.III. Informele Test

Omdat het systeem niet te testen is met Unit Tests heb ik ervoor gekozen om de code met een informele test te testen. In deze test voer ik code in en kijk ik of deze code doet wat ik ervan wil. Als de code dit niet doet verander ik het totdat het dat wel doet.

18.C. Voortgang

Product	Startdatum	Deadline	Product af
Afstudeerverslag opstellen	20 oktober 2015	20 maart 2016	Nee
Bouwen Proof-of-Concept ITable	11 november 2015	1 januari 2016	Ja
Tests Proof-of-Concept Uitvoeren	4 januari 2016	8 januari 2016	Ja
Testrapportage Proof-of-Concept Opstellen	4 januari 2016	8 januari 2016	Ja
Testplan Opstellen voor Uitbreiding	4 januari 2016	8 januari 2016	Ja
Klassendiagram Uitbreiding	11 januari 2016	13 januari 2016	Ja
Bouwen Uitbreiding	11 januari 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren	14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen	14 maart 2016	17 maart 2016	Nee

Voortgang Testplan Uitbreiding

19. Uitbreiding

Het doel van deze uitbreiding is zorgen dat het systeem objecten kan herkennen. Het is een vervolg op het Proof-of-Concept en daardoor ligt de basis er al. Het belangrijkste probleem met het ontwikkelen van deze uitbreiding is dat de Kinect SDK alles behalve bewegende personen enkel als achtergrond ziet.

Na research op het internet vond ik een Scriptie over objectherkenning met de Kinect voor Xbox 360 uit 2011⁷. Hierbij hoort het Point Cloud Library (PCL)⁸ Open Source Project om de objecten mee te herkennen. Na deze documentatie door te hebben gelezen ben ik aan de slag gegaan met het verwerken in het ITable Systeem.

Ik deel deze uitbreiding op in drie activiteiten zodat ik steeds een werkend product heb. De eerste sprint zal de PCL integreren in het ITable systeem. De tweede Sprint gaat met behulp van de PCL objecten herkennen en er weetjes bij neerzetten. En de laatste Sprint gaat een leermodule maken waarmee het mogelijk wordt om het Systeem nieuwe objecten te laten herkennen en weetjes erbij onthouden.

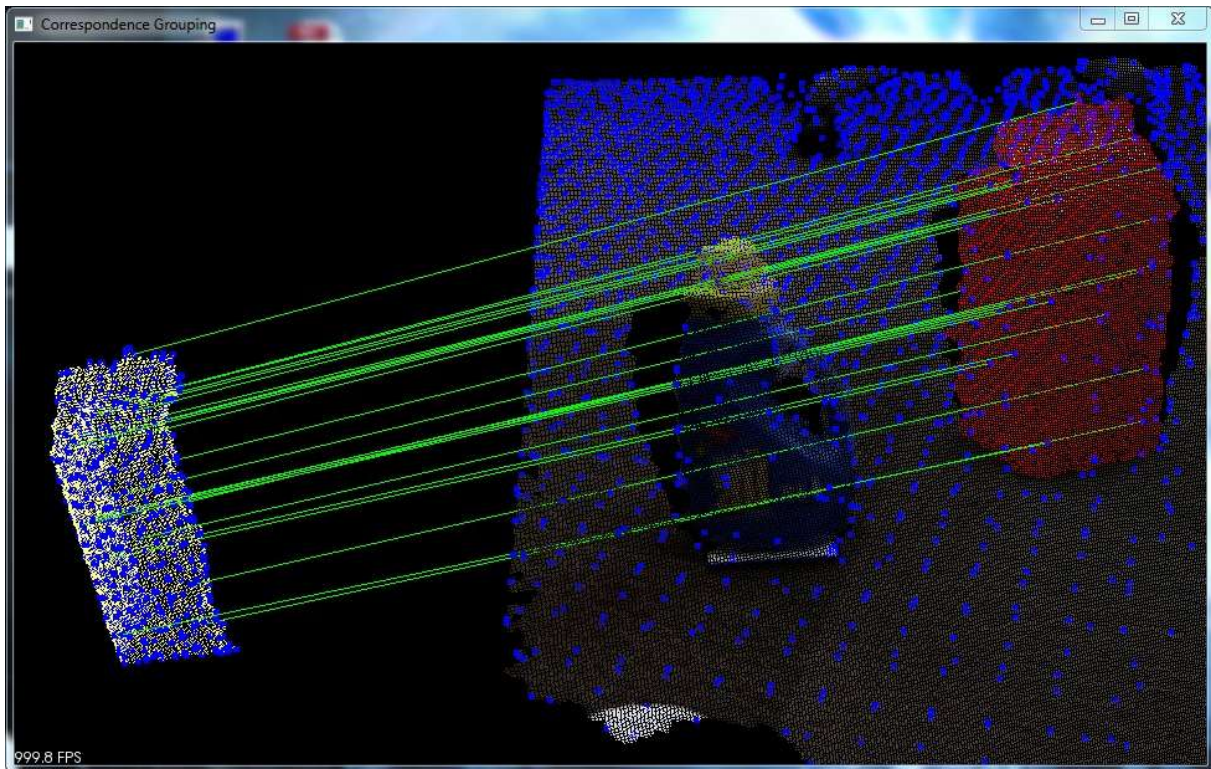
Als gevolg van de Wijziging van de uitbreiding worden Requirements F02, F04 en NF07 vervangen door F06 Objecten. Ook vind er een wijziging plaats in de prioriteit van de User Story's doordat US08 naar de 4de positie verplaatst, dit betekent dat behalve US11, US09 en US01 ze allemaal een plaatsje omlaag gaan met US09 die US13 vervangt en op de 3de positie komt.

7. nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2011/rapporter11/saponara_favio_11115.pdf

8. <http://pointclouds.org/>

19.A. De PCL Library

De Point Cloud Library is een Open Source Project voor “2D/3D image and Point Cloud processing”. Het kan uit camerabeelden een wolk van punten (Point Cloud) maken waarmee een object kan worden herkend. Er zitten verschillende algoritmes ingebouwd waarmee object kunnen worden herkend zoals het Hough algoritme en het Geometric Consistency (GC) algoritme. Met deze algoritmes kan worden bepaald of een object hetzelfde is als een bekend object. In onderstaande afbeelding 3 ziet u hoe PCL te werk gaat. In dit geval wordt een melkpak herkend, het blauwe is het vergelijkings-object en het rode is het fysieke, de groene lijnen zijn de overeenkomstige punten tussen de twee objecten.



Afbeelding 3. Object herkenning met PCL

19.B. Eerste Sprint (Integreren PCL)

In deze sprint ben ik aan de slag gegaan met het integreren van de PCL in het ITable Systeem. Allereerst heb ik de sourcecode gedownload en geïnstalleerd met bijbehorende 3rd party software. Vervolgens ben ik het gaan integreren in het systeem door eerst een nieuwe klasse aan te maken die de object herkenning gaat bevatten.

Het grootste probleem met het integreren van de PCL is het laten samenwerken met de Kinect SDK en de daarvan verkregen data omzetten in een Point Cloud. Gelukkig zijn er meerdere voorbeelden beschikbaar in de PCL documentatie van het maken van een Point Cloud waardoor ik makkelijker kon bepalen waar ik de Kinect data in moest voegen. Ook was na wat internet zoeken een voorbeeld gevonden hoe de Kinect data moest worden behandeld om een Point Cloud te kunnen krijgen, want anderen hadden de Kinect v1.8 en de PCL al eens samengevoegd voor gebruik⁹.

Toen ik deze integratie eenmaal goed rond had na wat informeel trial-and-error testen en er Point Cloud's werden gegenereerd op basis van de Kinect data was deze Sprint ook weer voltooid.

9. <http://unanancyowen.com/?p=867> Eerst bekeken op 1 Feb 2016

19.C. Voortgang Eerste sprint

Product	Sprint doel	Startdatum	Deadline	Product af
Afstudeerverslag opstellen		20 oktober 2015	20 maart 2016	Nee
Bouwen Proof-of-Concept ITable	Hardware & SDK integreren	11 november 2015	4 december 2015	Ja
	Detectie werkend maken	7 december 2015	18 december 2015	Ja
	Kleurplaat werkend maken met reactie	21 december 2015	1 januari 2016	Ja
Tests Proof-of-Concept Uitvoeren		4 januari 2016	8 januari 2016	Ja
Testrapportage Proof-of-Concept Opstellen		4 januari 2016	8 januari 2016	Ja
Testplan Opstellen voor Uitbreiding		4 januari 2016	8 januari 2016	Ja
Klassendiagram Uitbreiding		11 januari 2016	13 januari 2016	Ja
Bouwen Uitbreiding	Research Object Herkenning	11 januari 2016	15 januari 2016	Ja
	PCL toevoegen aan ITable	18 januari 2016	29 januari 2016	Ja
	Object Herkenning	1 Feb 2016	12 Feb 2016	Nee
	Object Herkenning Leermodule	14 Feb 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren		14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen		14 maart 2016	17 maart 2016	Nee

Voortgang Uitbreiding 1ste sprint

19.D. Tweede Sprint (Object Herkenning)

In deze sprint ben ik aan de slag gegaan met de objectherkenning. Allereerst moet ik hiervoor bepalen welke beschikbare objecten ik hiervoor het beste kan gebruiken. Ook moet ik kijken hoe ik deze objecten dan tot een bestand kan maken waarmee het later weer kan worden vergeleken.

Allereerst ben ik gestart met het doorlezen van de documentatie die betrekking heeft op het herkennen van objecten. Uit deze documentatie blijkt dat ik voor deze herkenning een Point Cloud nodig heb waarin het te vergelijken object aanwezig is. Deze Point Cloud kan uit een PCD (Point Cloud Data) bestand komen of uit een ander type bestand na omzetting.

Toen ik daar achter was gekomen bedacht ik me dat het naar een bestand zetten van een Point Cloud in de volgende sprint wordt uitgewerkt. Om geen dubbel werk te doen heb ik daarom besloten op dit punt de huidige sprint af te kappen en direct door te gaan naar de laatste. In deze laatste voltooi ik dan direct het hele systeem.

19.E. Voortgang tweede sprint

Product	Sprint doel	Startdatum	Deadline	Product af
Afstudeerverslag opstellen		20 oktober 2015	20 maart 2016	Nee
Bouwen Proof-of-Concept ITable	Hardware & SDK integreren	11 november 2015	4 december 2015	Ja
	Detectie werkend maken	7 december 2015	18 december 2015	Ja
	Kleurplaat werkend maken met reactie	21 december 2015	1 januari 2016	Ja
Tests Proof-of-Concept Uitvoeren		4 januari 2016	8 januari 2016	Ja
Testrapportage Proof-of-Concept Opstellen		4 januari 2016	8 januari 2016	Ja
Testplan Opstellen voor Uitbreiding		4 januari 2016	8 januari 2016	Ja
Klassendiagram Uitbreiding		11 januari 2016	13 januari 2016	Ja
Bouwen Uitbreiding	Research Object Herkenning	11 januari 2016	15 januari 2016	Ja
	PCL toevoegen aan ITable	18 januari 2016	29 januari 2016	Ja
	Object Herkenning	1 Feb 2016	12 Feb 2016	Nee
	Object Herkenning Leermodule	14 Feb 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren		14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen		14 maart 2016	17 maart 2016	Nee

Voortgang Uitbreiding 2de sprint

19.F. Laatste Sprint (Leermodule)

In deze sprint ga ik met behulp van PCL object herkenning en registratie toevoegen aan het systeem. Zoals ik in het vorige hoofdstuk al heb aangegeven is deze sprint eerder begonnen dan gepland en wordt het gecombineerd met het doel van de vorige.

Ik ben begonnen met het wegschrijven naar en uitlezen van een PCD bestand dat een Point Cloud bevat. Hiervoor maak ik gebruik van de Point Cloud die ik uit de data van de Kinect genereer. Het onderstaande screenshot bevat een voorbeeld van hoe een PCD bestand is opgebouwd.

```
# .PCD v.7 - Point Cloud Data file format
VERSION .7
FIELDS x y z rgb
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 213
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 213
DATA ascii
0.93773 0.33763 0 4.2108e+06
0.90805 0.35641 0 4.2108e+06
0.81915 0.32 0 4.2108e+06
0.97192 0.278 0 4.2108e+06
0.944 0.29474 0 4.2108e+06
0.98111 0.24247 0 4.2108e+06
0.93655 0.26143 0 4.2108e+06
0.91631 0.27442 0 4.2108e+06
0.81921 0.29315 0 4.2108e+06
0.90701 0.24109 0 4.2108e+06
0.83239 0.23398 0 4.2108e+06
0.99185 0.2116 0 4.2108e+06
0.89264 0.21174 0 4.2108e+06
0.85082 0.21212 0 4.2108e+06
0.81044 0.32222 0 4.2108e+06
0.74459 0.32192 0 4.2108e+06
0.69927 0.32278 0 4.2108e+06
```

Screenshot Point Cloud Data

De eerste 11 regels uit het bestand bestaan uit attributen van de Point Cloud Data, dit zijn respectievelijk:

1. Dat het een Point Cloud bestand is.
2. Welke versie van het bestand creatie algoritme er is toegepast.
3. De opbouw van de datawaarden, in dit geval XYZ-coördinaten en RGB-waarden. Alle regels vanaf nr.12 volgen deze volgorde.
4. Wat de maximale bytesize is van een gegeven.
5. Char type van een gegeven: I voor signed ints, U voor unsigned ints en F voor floats.
6. Hoeveel elementen een gegeven heeft.
7. De breedte van de Point Cloud oftewel hoeveel punten er naast elkaar staan in de Point Cloud. Over het algemeen is dit gelijk aan regel nr.10, omdat
8. Het aantal regels van de paragraaf aangeeft en die is over het algemeen 1.
9. Vanuit welk oogpunt er wordt gekeken.
10. Het aantal punten waaruit de Point Cloud bestaat. Dit geeft ook aan hoeveel regels er na nr.11 komen in het bestand.
11. Als laatste wordt aangegeven volgens welk character encoding het bestand is opgeslagen, in dit geval dus ASCII.

Toen ik dit opslaan en laden eenmaal voltooid had en tot eenvoudige methoden had teruggebracht, ging ik ervoor zorgen dat ik een enkel object uit de Point Cloud haal om weg te schrijven. Hiervoor heb ik gebruik gemaakt van de Algoritmes die bij de PCL ingebouwd zijn. Ik heb geëxperimenteerd met filters die deze algoritmes bevatten en deze steeds opnieuw aangepast om de Point Cloud terug te brengen naar enkel het/de object(en). Het beste resultaat wat ik er tegen het eind van het project van heb kunnen maken was van het object met een stuk tafel en een stuk van het plafond.

19.G. PCL WO-Niveau

Tijdens het initiële onderzoek naar de PCL heb ik wegens de tijdsbeperking niet de diepere achtergronden kunnen bekijken en waardoor het de Object Detectie kon toepassen. Tijdens het ontwikkelen van de Uitbreiding kwam ik er geleidelijk achter dat de achterliggende theorieën en algoritmes niet op een HBO niveau lagen echter had ik de hoop dat ik het toch zou kunnen gebruiken zonder die kennis.

Om ervoor te zorgen dat ik de algoritmes goed gebruik moest ik onder andere de onderstaande formules begrijpen. Ik heb tijdens de Sprints geprobeerd om het werkend te krijgen zonder deze formules te doorgronden, echter kreeg ik constant niet de resultaten die ik verwachtte en was het object altijd incorrect uit de Point Cloud gehaald. Uiteindelijk is gebleken dat ik met mijn Kennis Niveau niet de kans vond om met de PCL de object detectie en herkenning module compleet te maken. Ik ben van mening gezien de complexiteit van de onderstaande formules en het feit dat PCL ook in robotica wordt gebruikt dat het WO-Niveau is en ik dus met deze uitbreiding te hoog heb gegrepen.

Naam Theorie	Theorie	Link naam Scriptie
Rusu Dissertation	<p>The solution for estimating the surface normal is therefore reduced to an analysis of the eigenvectors and eigenvalues (or PCA – Principal Component Analysis) of a covariance matrix created from the nearest neighbors of the query point. More specifically, for each point \mathbf{P}_i, we assemble the covariance matrix \mathbf{C} as follows:</p> $\mathbf{C} = \frac{1}{k} \sum_{i=1}^k \cdot (\mathbf{p}_i - \bar{\mathbf{p}}) \cdot (\mathbf{p}_i - \bar{\mathbf{p}})^T, \quad \mathbf{C} \cdot \mathbf{v}_j = \lambda_j \cdot \mathbf{v}_j, \quad j \in \{0, 1, 2\}$ <p>Where k is the number of point neighbors considered <i>in the neighborhood of \mathbf{P}_i</i>, $\bar{\mathbf{p}}$ represents the 3D centroid of the nearest neighbors, λ_j is the j-th eigenvalue of the covariance matrix, and \mathbf{v}_j the j-th eigenvector.</p>	http://files.rbrusu.com/publications/RusuPhDThesis.pdf

Wordt door PCL gebruikt voor het afvangen van minieme kleurverschillen van een oppervlak

Naam Theorie	Theorie	Link naam Scriptie
<p>Hough Transforms and 3D SURF for robust three dimensional classification</p>	<p>For each visual word the statistical weight is calculated by the formula:</p> $W_{st}(c_i, v_j) = \frac{1}{n_{vw}(c_i)} \frac{1}{n_{vot}(v_j)} \frac{\frac{n_{vot}(c_i, v_j)}{n_{ftr}(c_i)}}{\sum_{c_k \in C} \frac{n_{vot}(c_k, v_j)}{n_{ftr}(c_k)}}$ <p>The statistical weight $W_{st}(c_i, v_j)$ weights all the votes cast by visual word v_j for class c_i. Here $n_{vot}(v_j)$ is the total number of votes from visual word v_j, $n_{vot}(c_i, v_j)$ is the number of votes for class c_i from v_j, $n_{vw}(c_i)$ is the number of visual words that vote for class c_i, $n_{ftr}(c_i)$ is the number of features from which c_i was learned. C is the set of all classes.</p> <p>For every keypoint (point for which feature was estimated) the learned weight is calculated by the formula:</p> $W_{lrn}(\lambda_{ij}) = f(\{e^{-\frac{d_a(\lambda_{ij})^2}{\sigma^2}} \mid a \in A\})$ <p>Authors of the article define λ_{ij} as the vote cast by a particular instance of visual word v_j on a particular training shape of class c_i; that is, λ_{ij} records the distance of the particular instance of visual word v_j to the center of the training shape on which it was found. Here A is the set of all features associated with word v_j on a shape of class c_i. The recommend value for σ is 10% of the shape size. Function f is simply a median. $d_a(\lambda_{ij})$ is the Euclidean distance between voted and actual center.</p> <p>For every feature</p> <ul style="list-style-type: none"> For every instance(which casts a vote for the class of interest) of every visual word from the trained model <ul style="list-style-type: none"> Ad vote with the corresponding direction and vote power computed by the formula $W(\lambda_{ij}) = W_{st}(v_j, c_i) * W_{lrn}(\lambda_{ij})$	<p>https://www.vision.ee.ethz.ch/publications/papers/proceedings/eth_biwi_00812.pdf</p>

Wordt door PCL gebruikt voor het herkennen van een object op basis van afstand tussen vergelijkbare punten

19.H. Voortgang eindsprint

Product	Sprint doel	Startdatum	Deadline	Product af
Afstudeerverslag opstellen		20 oktober 2015	20 maart 2016	Nee
Bouwen Proof-of-Concept ITable	Hardware & SDK integreren	11 november 2015	4 december 2015	Ja
	Detectie werkend maken	7 december 2015	18 december 2015	Ja
	Kleurplaat werkend maken met reactie	21 december 2015	1 januari 2016	Ja
Tests Proof-of-Concept Uitvoeren		4 januari 2016	8 januari 2016	Ja
Testrapportage Proof-of-Concept Opstellen		4 januari 2016	8 januari 2016	Ja
Testplan Opstellen voor Uitbreiding		4 januari 2016	8 januari 2016	Ja
Klassendiagram Uitbreiding		11 januari 2016	13 januari 2016	Ja
Bouwen Uitbreiding	Research Object Herkenning	11 januari 2016	15 januari 2016	Ja
	PCL toevoegen aan ITable	18 januari 2016	29 januari 2016	Ja
	Object Herkenning	1 Feb 2016	12 Feb 2016	Nee
	Object Herkenning Leermodule	14 Feb 2016	11 maart 2016	Nee
Tests Uitbreiding Uitvoeren		14 maart 2016	17 maart 2016	Nee
Testrapportage Uitbreiding Opstellen		14 maart 2016	17 maart 2016	Nee

Voortgang Uitbreiding Eindsprint

20. Afwijkingen Afstudeerplan

Tijdens het afstudeerproject kan het nog wel eens voorkomen dat er iets verandert aan de opdracht of dat de planning moet worden gewijzigd. In dit hoofdstuk geef ik aan welke veranderingen dit zijn geweest en vertel ik waarom.

20.A. F200 in plaats van R200

In het originele afstudeerplan had ik aangegeven dat de opstellingen waar mogelijk gebruik van zou worden gemaakt de Intel Realsense R200 sensor bevatten. Echter bleek na dieper onderzoek tijdens het project zelf dat de Intel Realsense F200 sensor beter bij het project paste omdat die gebaren kon registreren. Naar aanleiding van deze ontdekking is daarom in plaats van de R200 de F200 aangeschaft om voor het ontwikkelen te gebruiken. Nadat deze was gearriveerd heb ik het aangesloten en geprobeerd te laten werken. Ondanks het gebruik op verschillende configuraties die aan de specificaties van de sensor voldoen, bleek bij het testen dat de sensor niet naar behoren functioneerde.

20.B. Kinect in plaats van Intel Realsense

Zoals aangegeven aan het eind van het vorige sub-hoofdstuk werkte de Intel Realsense Sensor niet naar behoren waardoor er naar een andere opstelling moest worden gekeken. Dit betekent dat er werd gekeken naar de andere oplossingen die waren bedacht. Zoals ik in hoofdstuk 4 heb beschreven was er naast de Intel Realsense R200 en F200 Sensors ook gedacht aan de Kinect voor Windows en Kinect voor Xbox One Sensors gedacht. Zoals in hoofdstuk 4.B.I. beschreven was in het magazijn van Vision2Watch een van deze Kinect Sensors aanwezig, wat leide tot de beslissing om naar de Kinect SDK over te stappen.

20.C. Wijziging doelgroep & Inhoud Uitbreiding

In week 5 van het project werd besloten om in plaats van een uitbreidingsmodule voor spelletjes en restaurants als doelgroep het anders te doen. De opdrachtgever heeft zelf een jong kind en hij bedacht zich dat het leuk zou zijn om die wat leuks en lerend te geven. Zo kwam hij met het idee om de objectherkenning hoger op de prioriteit te zetten. De gedachte was dat het leuker zou zijn als het zich zou richten op kinderen om hun dingen te laten leren door middel van objectherkenning. Zo zou het een grotere doelgroep omvatten en aantrekkelijker zijn voor scholen.

21. Evaluatie

In dit hoofdstuk kijk ik terug op hoe het afstuderen is verlopen. Ik beoordeel hoe ik mijn werk heb gedaan, waar ik fouten en heb gemaakt en wat ik goed heb gedaan. Eerst kijk ik naar de producten die ik heb gemaakt, vervolgens komt het proces aan de orde en als laatste kijk ik naar de beroepstaken die ik had gesteld.

21.A. Product Evaluatie

In dit onderdeel behandel ik voor alle (deel)producten aan de hand van het afstudeerplan wat de verschillen zijn tussen de geplande producten en wat er daadwerkelijk van gekomen is.

21.A.I. Plan van Aanpak

Het Plan van Aanpak is volgens de regels gemaakt en is zoals aangegeven in het afstudeerplan aan het begin van project gemaakt. Ik heb het echter gemaakt met de originele opdracht als uitgangspunt waardoor het tegen het eind van het project niet meer zou voldoen. Tijdens het project heb ik het wel meerdere malen aangepast om het gelijk te houden aan de daadwerkelijke opdracht.

Alhoewel deze wijzigingen vaak klein waren is het niet altijd handig om een afgerond product achteraf te wijzigingen hoe aantrekkelijk dat ook mag zijn. Omdat de opdracht zelf was gewijzigd heb ik hier wel goed aan gedaan maar ik moet er in het vervolg op letten dat ik het niet te snel wijzig.

21.A.II. Requirements & User Stories

De Requirements en de bijbehorende User Stories zijn kort en bondig geformuleerd. Dit kan ertoe leiden dat een opvolger die verder zou gaan met het ITable systeem geen wijs eruit kan worden en een ongewenste richting op gaat. Tevens zijn de User Stories niet geformuleerd zoals ze in de regel horen te zijn, maar helaas heb ik geen duidelijk voorbeeld kunnen vinden van hoe het wel zou moeten.

21.A.III. Storyboard

Over het Storyboard heb ik weinig te zeggen behalve dat ik niet helemaal tevreden ben met mijn tekenkunsten. Gelukkig zijn mijn tekenkunsten (waarschijnlijk) later niet nodig tijdens mijn loopbaan dus dat scheelt.

21.A.IV. Klassendiagrammen

De klassendiagrammen zijn achteraf gemaakt op basis van de bestaande code. Dit is niet zoals het in praktijk hoort en ik heb er ook spijt van dat het zo is gelopen. Wat mij hierbij ook parten heeft gespeeld is dat ik het liefst niks anders zou doen dan code kloppen zonder om te kijken naar diagrammen of documentatie. Deze eigenschap van mij moet ik aanpassen want zo hoort het niet als je in een project werkt.

21.A.V. Entity Relationship Diagram

Net als bij de klassendiagrammen heb ik ook deze Reverse Engineered, gelukkig heeft dit geen invloed gehad op het slagen of falen van het project.

21.A.VI. Database

Ondanks dat de Database uiteindelijk niet gebruikt werd is het wel volgens de voorschriften gebouwd. Ik had voor MySQL gekozen omdat ik hier bekend mee ben en omdat het makkelijk te gebruiken is. Ik had om mezelf meer kennis te geven misschien beter een ander type database kunnen kiezen en zo mijn vaardigheden uitgebreid. Maar ik heb de keuze gemaakt omdat in het kader van de verdere ingewikkeldheid van het project mezelf hier wat moeite te besparen.

21.A.VII. Testplannen

Ik geef hier eerlijk toe dat ik geen fan ben van testscripts schrijven of uitvoeren. Vandaar dat de testplannen niet volgens de Master-Detail richtlijnen zijn gemaakt en ze flinterdun zijn. Origineel was het mijn plan om ze wel volgens de richtlijnen te maken, maar nadat ik mijn initiële onderzoek had gedaan besloot ik (mogelijk ten onrechte) om ze af te rafelen.

21.A.VIII. Proof-of-Concept

Het Proof-of-Concept is succesvol voltooid binnen de planning en functioneert naar behoren. Qua code is het simpel en goed van commentaar voorzien zodat een opvolger er direct mee aan de slag kan. Het enige wat ik jammer vind is dat het me niet gelukt was om het uit/aan zetten van het tekenen met een knop te laten werken.

21.A.IX. De Uitbreiding

Tja wat zal ik hier van zeggen? Het doel is niet bereikt en het product is niet af. Toen de opdrachtgever en ik overlegden over het wijzigen van de opdracht zag ik het wel zitten en leek het haalbaar. Helaas is achteraf gebleken dat de objectherkenning toch een zwaardere module is dan dat ik binnen de tijd kon voltooien. Verder hielp het ook niet dat ik voor het eerst met de algoritmes in aanraking kwam en ze zelfs nu nog niet helemaal begrijp.

Ik had waarschijnlijk voet bij stuk moeten houden en de opdracht ongewijzigd moeten later, maar ja dat weet ik nu wel en toen nog niet.

21.B. Proces Evaluatie

In dit hoofdstuk behandel ik het hele proces van het project. Ik kijk hoe alles is uitgevoerd, in welke mate de planning overeen kwam met de werkelijkheid en welke problemen en keuzes ik ben tegengekomen.

Ik had voor het project gekozen om een delen van verschillende Software Ontwikkel Methodes te pakken om mijn eigen SOM te maken voor de duur van het project. Voor de documentatie heb ik gekeken naar Test Driven Development en Agile methoden. Hieruit haalde ik de User Stories, het Storyboard en het originele idee van de Testplannen. Bij elke software ontwikkel traject moet je kijken naar hoe je de code gaat opbouwen en hierbij maak je gebruik van een Klassendiagram, dus heb ook ik deze gemaakt.

Voor de structuur het schrijven van de code heb ik gekeken naar de SCRUM SOM, omdat de korte sprints mij wel aanspraken. Ik bedacht me namelijk dat de opdrachtgever wel graag duidelijke voortgang zou willen zien van waar ik mee bezig ben om er zeker van te zijn dat alles op rolletjes loopt. De sprints van de SCRUM methode zijn van 2 tot 4 weken lang met als doel een werkend deel-product. Ik had ervoor gekozen om mijn sprints 2 weken te laten duren zodat ik duidelijke deadlines zou hebben en ook concrete ontwikkelstappen.

Bij de uitvoering zelf zie ik goede en slechte punten naar voren komen. Eerder in dit document heb ik al aangegeven waarom ik niet vooraf een klassendiagram heb gemaakt en waarom de testplannen volgen niet de Master Test Plan en Detail Test Plan richtlijnen. Wat ik wel heb gedaan is dat ik mij voor het grootste deel aan mijn planning heb gehouden en alles strak op schema liep tot met uitzondering van de laatste twee sprints van de uitbreiding.

21.C. Beroepstaken

In dit hoofdstuk ga ik per beroepstaak na of ik het doelniveau heb behaald en hoe. Voor alle doeltaken geldt dat ik ze zelfstandig heb moeten uitvoeren en met uitzondering van de beroepstaak Bouwen applicatie moeten ze tevens ook lastige complexiteit zijn. Bij elke beroepstaak geef ik eerst aan welke definitie voor het doelniveau geldt en vervolgens hoe ik het gedaald denk te hebben.

21.C.I. Uitvoeren analyse door definitie van Requirements

“Het betreft één applicatie met een groot aantal Requirements. Er is sprake van een beperkt aantal stakeholders met tegengestelde wensen en eisen t.o.v. de applicatie. Er is één set aan Requirements, die gedurende de analyse kan veranderen.”

Zoals ik eerder in dit document heb aangegeven was er bij dit project maar sprake van twee stakeholders, de opdrachtgever en ik zelf. Ook zijn het aantal Requirements niet groot omdat er geen overleg heeft kunnen plaatsvinden tussen ons en een potentiële klant. Wel heb ik de Requirements opgesteld aan de hand van de ingebeeld standpunten van andere stakeholders waardoor de eisen aan het systeem zo compleet mogelijk gemaakt zijn zonder inbreng van een derde. In tegenstelling tot de omschrijving is het in dit project om meer dan één applicatie gegaan, omdat er niet alleen gebruik is gemaakt van de Kinect in het Proof-of-Concept maar ook de PCL Library in de uitbreiding dat haast een standalone systeem deel is geworden.

21.C.II. Ontwerpen, bouwen en bevragen van een database

“Het betreft het ontwerpen, bouwen en bevragen van een database voor slechts één gebruiker. Daarnaast dient de integriteit van de database gewaarborgd te worden. Bij het beantwoorden van vragen op de database moeten er meerdere tabellen geraadpleegd worden en moeten verschillende constructies toegepast worden.”

Ik heb tijdens het project een database bebouwd dat aan bovenstaande eisen zou hebben voldaan. Helaas heeft deze database geen invloed gehad op het systeem en was het tevens voor een oudere versie van de opdracht. De Database bestaat zoals u in het ERD hoofdstuk heeft kunnen zien uit meerder tabellen die onderlinge relaties hebben. Ondanks dat de database geen deel heeft uitgemaakt van het systeem heb ik wel verschillende Foreign Keys en Constraints toegepast om te zorgen dat de integriteit gewaarborgd zou blijven.

21.C.III. Bouwen applicatie

“De applicatie sluit aan op bestaande software of maakt gebruik van een bestaand Framework. Het bouwen gebeurt in een geavanceerde ontwikkelomgeving inclusief testomgeving en versiebeheertool.”

Het ITable Systeem is gebouwd op basis van het Kinect SDK met bij de uitbreiding ook nog de PCL Library. Ik heb geen gebruik gemaakt van versiebeheer bij het bouwen van het systeem nog heb ik aparte omgevingen gebruikt. Dit betekent dat het niet heeft voldaan aan de eisen van de hoogste complexiteit (complex) maar gelukkig voldoet het wel aan de eisen van de lastige complexiteit waarbij die eisen als volgt zijn:

“Het betreft het bouwen van een objectgeoriënteerde applicatie , waarbij geavanceerde concepten van de gebruikte programmeertaal aan de orde komen. Verder wordt rekening gehouden met toekomstige wijzigingen, testbaarheid en hergebruik. Het bouwen gebeurt in een ontwikkelomgeving.”

21.C.IV. Ontwerpen Systeemdeel

“Het betreft het ontwerpen (structuur en gedrag) van een objectgeoriënteerde applicatie, met behulp van een ontwerpmethodiek en een tool. Hierbij wordt rekening gehouden met toekomstige wijzigingen, testbaarheid en hergebruik.”

Tja, wat kan ik hier nog toevoegen wat ik niet al bij de vorige beroepstaak heb aangegeven. Ik heb deze beroepstaak gekozen met de achterliggende gedachte dat de uitbreiding een apart systeemdeel is ten opzichte van het Proof-of-Concept. Deze beroepstaak past echter zowel bij het PoC als bij de uitbreiding en gezien de laatste niet succesvol is afgerond kan daar de beroepstaak bouwen applicatie niet op gelden. Wel heeft de uitbreiding voldaan aan de eisen van deze beroepstaak want ik heb het ontworpen in een klassendiagram met behulp van de UMLet tool en gebouwd in Visual Studio.

21.C.V. Uitvoeren van en rapporteren over het testproces

“Bij het opstellen van het logisch testontwerp wordt gebruik gemaakt van een testontwerptechniek. Er is aandacht voor herhaalbaarheid van de testen. Het betreft hoofdzakelijk het testen van de functionaliteit. Testrapportage betreft het volledige systeem.”

Ik heb tijdens het project gebruik gemaakt van een simpeler testplan dan originele mijn bedoeling was. Dit neemt echter niet weg dat ik gebruik heb gemaakt van meerder testontwerptechnieken voor mijn logische testontwerpen. Ook heb ik ervoor gezorgd dat de tests door een volgend persoon kunnen worden herhaald en voor nieuwe systeemdelen. Het enige wat ik jammer vind ten opzichte van deze beroepstaak is dat ik geen gebruik heb kunnen maken van Unit Tests en een MTP en DTP.

22. Conclusie

Het is mij niet is gelukt om het product naar wens te voltooien. Ik ben uiteindelijk tijd tekort gekomen als gevolg van de wijziging van de opdracht, omdat ik voor het eerst in aanraking ben gekomen met object detectie, object herkenning en de PCL Library. Hiermee bedoel ik echter niet dat als het niet was veranderd dat ik het dan zeker wel zou hebben voltooid want dat kan niet worden bepaald.

Ondanks ik de uitbreiding niet werkend heb kunnen krijgen is het idee wel haalbaar. In het Proof-of-Concept toon ik al aan dat het mogelijk is om met behulp van een Kinect een systeem te bouwen wat onafhankelijk van het detectie oppervlak opereert, oftewel geen hardware in de tafel. Alhoewel het me niet is gelukt om de objectherkenning en detectie te laten werken kan ik wel met zekerheid zeggen dat als er genoeg kennis van de PCL Library aanwezig was geweest bij Vision2Watch dan had ik het succesvol kunnen afronden.

Mocht Vision2Watch in de toekomst wederom een poging wagen lijkt het mij verstandig om dan iemand ervoor in te zetten die ervaring heeft met robotica en Artificial Intelligence, omdat de algoritmes die bij de objectherkenning nodig zijn van Universitair Niveau zijn. Verder moet er dan meer tijd beschikbaar zijn en van te voren duidelijkheid over de opdracht zonder een wijziging.

23. Bijlagen

23.A. Plan van Aanpak

21
oktober
2015

Plan van Aanpak ITable Vision2Watch

P.C. REINKING

STNR 08057575

23.A.I. Inleiding

In dit document vind u het Plan van Aanpak voor het Interactieve Tafel (ITable) project.

In dit Plan van Aanpak wordt behandeld hoe het ITable systeem tot stand zal worden gebracht. Verder zal de aanleiding en het doel van het project uitgelegd worden. Er zal behandeld worden wie bij het project betrokken zijn en wat de scope en randvoorwaarden zijn. Als laatste zullen de (tussen)producten, risico's en de planning aan bod komen.

23.A.II. Aanleiding

Vision2Watch wil zich nu ook in het restaurantleven mengen en daarvoor een product maken. Ze hebben een visie van interactieve tafels waarmee de kinderen zoet worden gehouden en de volwassenen niet hoeven worden afgeleid.

Bij dit ITable systeem moeten de Projectie en Motion Detectie op elkaar kunnen reageren. De bedoeling is dat het systeem op elke willekeurige huis, tuin en keuken tafel zonder hardware kan reageren en zich in bijvoorbeeld een lampenkap boven deze tafel bevindt.

Vision2Watch wil de interactieve tafels door middel van projectie en bewegingsdetectie toepassen op normale tafels zonder enige hardware erin verwerkt. Momenteel bestaan Touch-Tafels al met het E-Table™ systeem van Inamo™ een restaurant in Groot-Brittannië, echter vereist dit speciale (E-Table™) tafels. Dit E-Table™ systeem kunt u ook in het Izkaya restaurant in Rotterdam vinden.

23.A.III. Doelstelling

Doelstelling van het project is het ontwikkelen van de software voor het ITable systeem. Hierbij moet de Motion Detectie en Projectie op elkaar reageren. Er zal gebruik gemaakt worden van een Software Development Kit (SDK) met bijbehorende Motion Detectie Camera en een projector. Ook moet deze software aansluiten op de overige software en hardware van Vision2Watch.

23.A.IV. Projectorganisatie

23.A.IV.a. Wie en Wat

Naam	Functie
P.C. Reinking	Projectmanager / Developer
D. Frencken	Bedrijfsmentor / Opdrachtgever

23.A.IV.b. Projectmethode

In dit project zal gebruik worden gemaakt van de **SCRUM** Software Ontwikkel Methode. Alhoewel deze ontwikkelmethode beter geschikt is voor teamverband zijn de hoofdlijnen van de methodiek ook toe te passen op teams van één.

De SCRUM methode is resultaat gedreven dat betekent er weinig documentatie zal zijn binnen het project en dat de nadruk ligt op het ontwikkelen van het ITable systeem. Verder betekent dit dat het bouwen van het systeem in sprints van twee tot vier weken wordt gedaan met aan het eind van de sprint een werkend deel software met verder elke week een meeting met het team.

23.A.V. Scope & Randvoorwaarden

Het ITable Systeem zal worden gebouwd in de C++ programmeertaal. Verder zal er voor de bijbehorende Database met SQL moeten worden gewerkt. Het zal in de Visual Studio 2015 IDE worden gebouwd met de Kinect 2 SDK en/of Intel Realsense SDK. Afhankelijk van de gebruikte SDK('s) zal er gebruik worden gemaakt van de Intel Realsense F200 sensor camera en/of de Kinect voor Xbox One Sensor met bijbehorende adapter.

Sensor	Bereik	Geschikt voor	Diepte / Infrarood
Intel Realsense F200	20-120 cm	1. Registreren van hand en vinger gebaren 2. Bewegingsherkenning 3. Gezichtsherkenning	640 x 480
Intel Realsense R200	51-400 cm	1. Gezichtsherkenning 2. Bewegingsherkenning 3. Meten van de omgeving	640 x 480
Kinect voor Xbox One	50-450 cm	1. Bewegingsherkenning tot 6 Skeletten (mensen) met 25 gewrichtspunten elk 2. Meten van de omgeving	512 x 424

Tabel 1. vergelijking sensors

23.A.VI. Risico's

Tijdens het leven en dus ook het project kan er van alles misgaan. In de onderstaande tabel worden daarom ook de meest voorkomende risico's opgesomd met daarbij wat de kans, impact en de manier om het te voorkomen is.

Beschrijving Risico	Kans (1-10)	Impact (1-10)	Risico (1-100)	Hoe te Voorkomen of genezen
Ziekte	1	10	10	Goed op de gezondheid letten
ITable systeem-onderdeel niet af tegen eind project	3	10	30	Goede communicatie met Betrokkenen
Ongeoorloofde afwezigheid	1	10	10	Goede communicatie met Medewerkers
Verandering Requirements	2	7	14	Vaak en duidelijk communiceren

Tabel 2. Risico's

23.A.VII. Op te Leveren Producten

Tijdens het project moeten verschillende (tussen)producten worden opgeleverd. Hieronder staan deze (tussen)producten in een overzichtelijk opsomming. Hierbij wordt beschreven wat elk product inhoudt.

Naam Product	Beschrijving
Plan van Aanpak	Het document waarin wordt aangegeven hoe het project zal worden aangepakt.
Requirements rapport	In dit document worden alle Requirements op en rijtje gezet en aangegeven welke invloed ze hebben op het project.
User Stories	Hier worden de Requirements omgezet tot scenario's waarin Wie, Wat en Waarom centraal staan. Een voorbeeld is: Als restauranteigenaar wil ik met één druk op de knop het systeem starten om zo min mogelijk tijd eraan kwijt te zijn.
Storyboard	Net als de User Stories worden hierin de Requirements in een overzichtelijk scenario gezet. Echter is het in dit geval een visueel scenario met sketches en tekeningen waarin de visie wordt weergegeven.
Database	Dit is een verzameling gegevens waarmee het systeem werkt. Hierin komt onder anderen het aantal mensen aan een tafel, de verdeling van de mensen en de afmetingen van het gebied dat het systeem mot overzien.
Proof-of-Concept	Dit product is het bewijs dat het idee en de visie van het ITable systeem juist is en werkt. In dit systeem-onderdeel van het ITable systeem moet de Motion Detectie en de projectie met elkaar samenwerken en op elkaar reageren.
Uitbreidings-module	Als het Proof-of-Concept aantoont dat het idee juist is dan wordt met dit systeem-onderdeel de basis gelegd voor de rest van het systeem. Ten tijden van het schrijven van dit Plan van Aanpak is de gedachte voor dit systeem-onderdeel dat het een tafel op kan delen en elke gebruiker een eigen projectie te zien krijgt. Hierbij moet er ook al een spelletje zijn dat kan worden gespeeld.
Testrapporten	Dit is de documentatie die betrekking heeft op het testen van het systeem. Het omvat onder anderen de testresultaten, testplannen en testscripts.

Tabel 3. Producten

23.A.VIII. Planning

Een project heeft meestal maar een beperkte tijd totdat het eindproduct moet worden opgeleverd. Ook in dit project is dit het geval, om te zorgen dat de deadline wordt gehaald is een planning nodig. Deze planning vindt u hieronder in tabel 4. In de planning zijn de verwachte startdatums en de deadlines voor de (tussen)producten overzichtelijk uitgezet.

Product	Startdatum	Deadline
Maken Plan van Aanpak	20 oktober 2015	23 oktober 2015
Requirements Opstellen	23 oktober 2015	30 oktober 2015
User Stories Opstellen	23 oktober 2015	30 oktober 2015
Storyboard Maken	23 oktober 2015	30 oktober 2015
Testplan Opstellen voor Proof-of-Concept	30 oktober 2015	6 november 2015
Database voor ITable	6 november 2015	10 november 2015
Bouwen Proof-of-Concept ITable	10 november 2015	5 januari 2016
Testrapportage Proof-of-Concept Opstellen	5 januari 2016	10 januari 2016
Testplan Opstellen voor Uitbreidings-module	10 januari 2016	16 januari 2016
Bouwen Uitbreidings-module	16 januari 2016	14 maart 2016
Testrapportage Uitbreidings-module Opstellen	14 maart 2016	17 maart 2016

Tabel 4. Planning

Testplan Proof-of-Concept

P.C. REINKING
STNR 08057575

23.B.I. Inleiding

In dit document wordt behandeld hoe de kwaliteit van het Proof-of-Concept product wordt gewaarborgd. Omdat in het geval van het Proof-of-Concept het systeem nog klein is wordt in dit document het Master Test Plan en het Detail Test Plan samengevoegd. Voor de uitbreidingsmodule zullen ze wel apart van elkaar worden gemaakt, omdat er dan meer te testen valt.

Als eerste zal er worden aangegeven welke onderdelen er wel en welke er niet zullen worden in de testdekking. Vervolgens zal de aanpak van het testen worden behandeld. Hierna komen de testgevallen en de criteria van de testen aan de orde. Als laatste zal de testplanning behandeld worden.

23.B.II. Testdekking

Om de kwaliteit van het product te kunnen waarborgen moet allereerst worden bepaald welke onderdelen er wel en welke niet worden getest. De lengte van het project staat echter niet toe dat er een dekkingsgraad van 100% gehaald zal worden. Om te zorgen dat er alsnog een zo hoog mogelijke dekking van het product zal zijn moet er naar de meest belangrijke functies worden gekeken.

Bij het Proof-of-Concept is dit de Actie-Reactie relatie tussen de Projectie en de Motion Detectie. Ook de afbakening van het herkenninggebied is een belangrijke functie, omdat dit bepaald of het systeem de vorm van de tafel kan herkennen.

Voor het systeem is een database een vereiste wat betekend dat ook deze getest moet worden. In deze database zitten de formules voor het bepalen van de tafel-vorm waarmee de afbakening van het herkenninggebied wordt afgehandeld. Verder wordt er ook bijgehouden hoeveel gebruikers er bij een specifieke tafel-vorm kunnen worden geregistreerd.

23.B.III. Aanpak

In dit hoofdstuk wordt behandeld hoe het testen van het Proof-of-Concept aan zal worden gepakt. Hiervoor is aangegeven van welke testsoorten er gebruik zal worden gemaakt.

Om te zorgen dat de dekkinggraad hoog is zullen op de twee producten, het Proof-of-Concept (PoC) en de Database (DB) testen worden uitgevoerd.

In de onderstaande tabel worden de te gebruiken testsoorten afgezet tegen de te testen producten. Er zal per testsoort worden aangegeven wat ermee bedoeld wordt en op welk product het van toepassing is, dit wordt met behulp van de kleuren Rood voor als het niet van toepassing is en Groen voor als het wel van toepassing is.

Naam Test	PoC	DB	Omschrijving Test
CRUD-Matrix			Create, Read, Update & Delete de vier basis acties voor een Relationale Database. White Box test.
Syntaxis Test			Word de mogelijkheid op foutieve invoer goed afgedekt? In deze test word bijvoorbeeld een cijfer ingevoerd waar alleen A-z mag om te kijken of het een fout teruggeeft.
State Transition			Kun je bij alle toestanden van het systeem bijvoorbeeld het versleutelen van het systeem
Systeem Test			Bij deze test wordt gekeken of het systeem aan de Requirements voldoen.

Tabel 1. Testsoorten

23.B.IV. Testgevallen

In dit hoofdstuk worden de testgevallen behandeld, deze testgevallen zijn de verwachte en de werkelijke uitkomsten van de tests. Voor de overzichtelijkheid zal er voor elke testsoort een sub hoofdstuk waardoor u direct naar de gezochte resultaten kan gaan. Ook zullen de criteria waaraan de producten voor elke test moeten voldoen aan worden gegeven.

23.B.IV.a. CRUD-Matrix Test

Een CRUD-matrix is een overzicht van de gebruikte acties van een Database. Er wordt per tabel gekeken welke methoden welke acties uitvoeren. In de onderstaande tabel 2 zijn respectievelijk de verwachte en de gevonden resultaten van de CRUD-matrix test te vinden.

Tabel	Resultaat	Methode 1	Methode 2	Methode 3	Methode 4
Gebruikers	Verwacht		C, R	R, U, D	
	Gevonden	N.V.T.	N.V.T.	N.V.T.	N.V.T.
Tafel_Gebruikers	Verwacht		C, R	R, U, D	
	Gevonden	N.V.T.	N.V.T.	N.V.T.	N.V.T.
Tafel	Verwacht	C, R	R, U	R, U	D
	Gevonden	N.V.T.	N.V.T.	N.V.T.	N.V.T.
Tafel_Afbakeningsdetectie	Verwacht		C, R, U	R, U	D
	Gevonden	N.V.T.	N.V.T.	N.V.T.	N.V.T.
Afbakeningsdetectie	Verwacht	C	R,U		D
	Gevonden	N.V.T.	N.V.T.	N.V.T.	N.V.T.

Tabel 2. Resultaten CRUD-matrix Test

Zoals u in de tabel hierboven ziet met de Niet van Toepassing (N.V.T.) wordt er in het uiteindelijke Proof-of-Concept geen gebruik gemaakt van de Database. Dit leidt ertoe dat deze test niet kan worden uitgevoerd.

23.B.IV.b. Syntaxis Test

Bij deze test wordt gekeken of het systeem onder anderen SQL Injectie toestaat. Er zal worden gekeken of je bij het invoeren van het aantal gebruikers van een tafel een woord kan invullen in plaats van een getal. In onderstaande tabel 3 en 4 staan, voor respectievelijk de GUI en de Database (DB), de verwachte en gevonden resultaten voor de test. Hierbij wordt eerst aangegeven wat de invoer was bij welke tabel en vervolgens wat het verwachte en gevonden resultaat is. Zoals u in de tabel hieronder ziet met de Niet van Toepassing (N.V.T.) wordt er in het uiteindelijke Proof-of-Concept geen gebruik gemaakt van de Database. Dit leidt ertoe dat dit onderdeel van deze test niet kan worden uitgevoerd.

Invoer in GUI	Betrokken onderwerp	Verwachte uitvoer	Gevonden uitvoer
'2'	Gebruikers	De Projectie en Motion Detectie is in 2 delen gesplitst	N.V.T.
'Tien'	Gebruikers	Foutmelding 'Geen getal'	N.V.T.
'10'	Gebruikers	Foutmelding 'Te groot aantal'	N.V.T.
'2-tal'	Gebruikers	Foutmelding 'Combinatie met non-getal'	N.V.T.
'5'	Tafel	Foutmelding 'Invoer is een getal moet de naam van een instelbare vorm zijn'	N.V.T.
'Rond'	Tafel	Detectie is afgesteld op een Ronde Tafel	N.V.T.
'Ster'	Tafel	Detectie is afgesteld op een 5punts Ster Tafel	N.V.T.
'5hoek'	Tafel	Foutmelding 'Naam tafelvorm mag enkel uit A-z bestaan'	N.V.T.
'Hexagon'	Tafel	Detectie is afgesteld op een Hexagon tafel	N.V.T.
'Y8:640x480:30'	Afbakening van detectie	De Detectie gebruikt een 480p resolutie Infrarood stream met 30 FPS	N.V.T.
'RGB32:1920x1080:60'	Afbakening van detectie	De Detectie gebruikt een 1080p resolutie RGB 32bit Kleur stream met 60 FPS	N.V.T.
'Depth:640x240:110'	Afbakening van detectie	De Detectie gebruikt 640x240 resolutie Diepte stream met 110 FPS	N.V.T.
'RGB'	Afbakening van detectie	Foutmelding 'onvolledige parameters'	N.V.T.
'RGB16:1920x1080:60'	Afbakening van detectie	Foutmelding 'verkeerd kleurspectrum'	N.V.T.
'Depth:1920x1080:110'	Afbakening van detectie	Foutmelding 'verkeerde resolutie en frame-rate'	N.V.T.

Tabel 3. Resultaten Syntaxis Test GUI

Invoer in DB	Betrokken tabel	Verwachte uitvoer	Gevonden uitvoer
'4'	Gebruikers	Nieuwe rij in tabel	Nieuwe rij in tabel
'Vier'	Gebruikers	error 'Not a number'	ERROR #1643 - Insert only non-zero number
'4-tal'	Gebruikers	error 'Not a number'	ERROR #1643 - Insert only non-zero number
'40'	Gebruikers	Error 'int greater than allowed'	Nieuwe rij in tabel
'5,1'	Tafel_Gebruikers	Nieuwe rij in tabel	Nieuwe rij in tabel
'm, k'	Tafel_Gebruikers	Error 'parameters must be (Tafel_ID, Gebruikers_ID)'	#1452 – Cannot add or update a child row: a foreign key constraint fails
'10'	Tafel	Error 'expected string not int'	ERROR #1643 - Expected text not number
'Ster'	Tafel	Nieuwe rij in tabel	Nieuwe rij in tabel
'5hoek'	Tafel	Nieuwe rij in tabel	Nieuwe rij in tabel
'Hexagon'	Tafel	Nieuwe rij in tabel	Nieuwe rij in tabel
'2,4'	Tafel_Afbakeningsdetectie	Nieuwe rij in tabel	Nieuwe rij in tabel
'gg'	Tafel_Afbakeningsdetectie	Error 'parameters must be (Tafel_ID, Detectie_ID)'	#1452 - Cannot add or update a child row: a foreign key constraint fails
'Y8: 640x480: 30'	Afbakeningsdetectie	Nieuwe rij in tabel	Nieuwe rij in tabel
'RGB32: 1920x1080: 60'	Afbakeningsdetectie	Nieuwe rij in tabel	Nieuwe rij in tabel
'Depth: 640x240: 110'	Afbakeningsdetectie	Nieuwe rij in tabel	Nieuwe rij in tabel
'RGB'	Afbakeningsdetectie	Nieuwe rij in tabel	Nieuwe rij in tabel
'RGB16: 1920x1080: 60'	Afbakeningsdetectie	Nieuwe rij in tabel	Nieuwe rij in tabel
'Depth: 1920x1080: 110'	Afbakeningsdetectie	Nieuwe rij in tabel	Nieuwe rij in tabel

Tabel 4. Resultaten Syntaxis Test DB

Zoals u hierboven in tabel 4 kunt lezen en zoals verwacht zijn de resultaten nagenoeg hetzelfde als van tevoren was bedacht.

23.B.IV.c. State Transition Test (STT)

Bij deze test wordt gekeken of het systeem alle toestanden en transities kan doorgaan. In het onderstaande diagram 1 ziet u welke toestanden er gelden voor het Proof-of-Concept (Groen) en welke er nu niet gelden. Verder ziet u in tabel 5 wat de verwachte en gevonden resultaten van de test zijn.

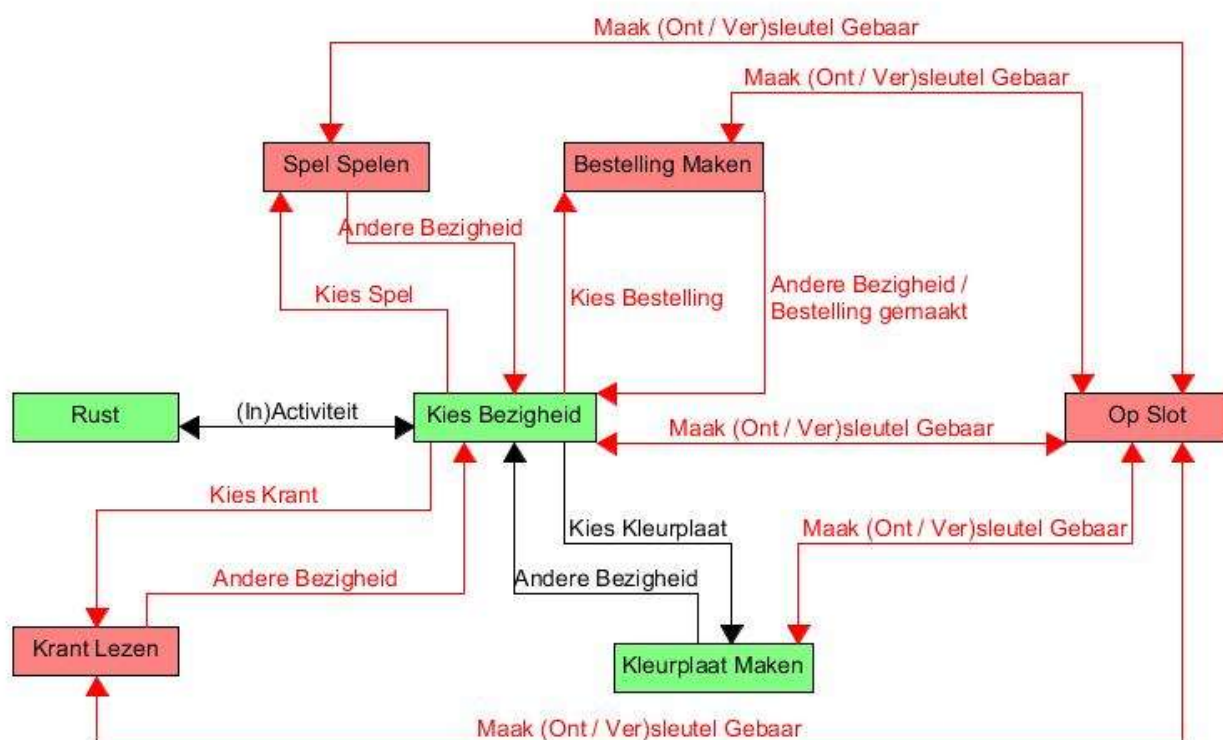


Diagram 1. State Diagram

Toestand	Verwacht	Gevonden
Rust	Toestand stabiel	Toestand stabiel
Rust → Kies Bezigheid	Transitie Geslaagd	Transitie Geslaagd
Kies Bezigheid	Toestand stabiel	Toestand stabiel
Kies Bezigheid → Rust	Transitie Geslaagd	Transitie Geslaagd
Kies Kleurplaat	Transitie Geslaagd	Transitie Geslaagd
Kleurplaat Maken	Toestand stabiel	Toestand stabiel
Andere Bezigheid	Transitie Geslaagd	Transitie Geslaagd

Tabel 5. STT Resultaten

Zoals u hierboven in tabel 5 kunt zien zijn de resultaten van de uitgevoerde test gelijk aan de verwachte resultaten.

23.B.IV.d. Systeem Test (ST)

Bij deze test wordt naar de Functionele Requirements gekeken die vooraf zijn bepaald. Vervolgens wordt gekeken of het systeem hier aan voldoet. Omdat bij dit project er geen eindgebruiker of klant beschikbaar is zal deze test in plaats daarvan door de Developer en de Opdrachtgever worden uitgevoerd.

In de onderstaande tabel 6 kunt u de Functionele Requirements terugvinden die betrekking hebben op het Proof-of-Concept. Verder wordt aangegeven wat de verwachte en gevonden resultaten zijn.

ID Requirement	Omschrijving Requirement	Verwacht Resultaat	Gevonden Resultaat
F01 Actie-Reactie	De Projectie en de Motion Detectie moeten reageren op elkaar	Er is Actie-Reactie tussen de Motion Detectie en de Projectie	Er is Actie-Reactie tussen de Motion Detectie en de Projectie
F04 Gebruikers verdeling	Er moet kunnen worden Aangegeven hoeveel gebruikers er aan een tafel zitten zodat de vorm wordt opgesplitst in genoeg gebieden	Projectie wordt in de juiste vorm en in het juiste aantal gebieden getoond	Niet gevonden in Proof-of-Concept
F05 Gebaren	Het systeem mag alleen op gebaren reageren die vooraf zijn ingesteld	Het systeem reageert alleen op de ingestelde gebaren	Niet gevonden in Proof-of-Concept
NF04	De kinderen moeten zichzelf met een kleurplaat kunnen intekenen	Je kan de projectie inkleuren	De projectie wordt ingekleurd met je bewegingen

Tabel 6. ST Test Resultaten

23.B.V. Planning

In dit hoofdstuk kunt u in onderstaande tabel 7 de startdatums en deadline zien voor het uitvoeren van de tests op het Proof-of-Concept.

Test	Startdatum	Deadline
CRUD-matrix	25-dec-2015	28-dec-2015
Syntaxis	25-dec-2015	28-dec-2015
STT	25-dec-2015	28-dec-2015
ST	25-dec-2015	28-dec-2015

Tabel 7. Planning Test Uitvoering

Testplan Uitbreidingsmodule

P.C. REINKING
STNR 08057575

23.C.I. Inleiding

In dit document wordt behandeld hoe de kwaliteit van de Uitbreidingsmodule product wordt gewaarborgd. Omdat ook in het geval van de Uitbreidingsmodule het systeem niet groot genoeg is wordt in dit document het Master Test Plan en het Detail Test Plan samengevoegd.

Als eerste zal er worden aangegeven welke onderdelen er wel en welke er niet zullen worden in de testdekking. Vervolgens zal de aanpak van het testen worden behandeld. Hierna komen de testgevallen en de criteria van de testen aan de orde. Als laatste zal de testplanning behandeld worden.

23.C.II. Testdekking

Om de kwaliteit van het product te kunnen waarborgen moet allereerst worden bepaald welke onderdelen er wel en welke niet worden getest. De lengte van het project staat echter niet toe dat er een dekkingsgraad van 100% gehaald zal worden. Om te zorgen dat er alsnog een zo hoog mogelijke dekking van het product zal zijn moet er naar de meest belangrijke functies worden gekeken.

Bij de Uitbreidingsmodule is dit de objectherkenning met leuke weetjes voor de objecten. Omdat de afbakening van het herkenningsgebied nog niet voltooid was in het Proof-of-Concept wordt die nu alsnog getest.

23.C.III. Aanpak

In dit hoofdstuk wordt behandeld hoe het testen van de Uitbreidingsmodule aan zal worden gepakt. Hiervoor is aangegeven van welke testsoorten er gebruik zal worden gemaakt.

Om te zorgen dat de dekkinggraad hoog is zullen op de Uitbreidingsmodule testen worden uitgevoerd.

In de onderstaande tabel worden de te gebruiken testsoorten afgezet tegen de te testen uitbreidingsmodule.

Naam Test	Omschrijving Test
State Transition	Kun je bij alle toestanden van het systeem bijvoorbeeld het versleutelen van het systeem
Systeem Test	Bij deze test wordt gekeken of het systeem aan de Requirements voldoen.
Informele test	Bij deze test verander ik de code om te kijken wat het resultaat is.

Tabel 1. Testsoorten

23.C.IV. Testgevallen

In dit hoofdstuk worden de testgevallen behandeld, deze testgevallen zijn de verwachte en de werkelijke uitkomsten van de tests. Voor de overzichtelijkheid zal er voor elke testsoort een sub hoofdstuk waardoor u direct naar de gezochte resultaten kan gaan. Ook zullen de criteria waaraan de producten voor elke test moeten voldoen aan worden gegeven.

23.C.IV.a. State Transition Test (STT)

Bij deze test wordt gekeken of het systeem alle toestanden en transities kan doorgaan. In het onderstaande diagram 1 ziet u welke toestanden er gelden voor de Uitbreidingsmodule (Groen) en welke er nu niet gelden. Verder ziet u in tabel 5 wat de verwachte en gevonden resultaten van de test zijn.

23.C.IV.a.iState Diagram

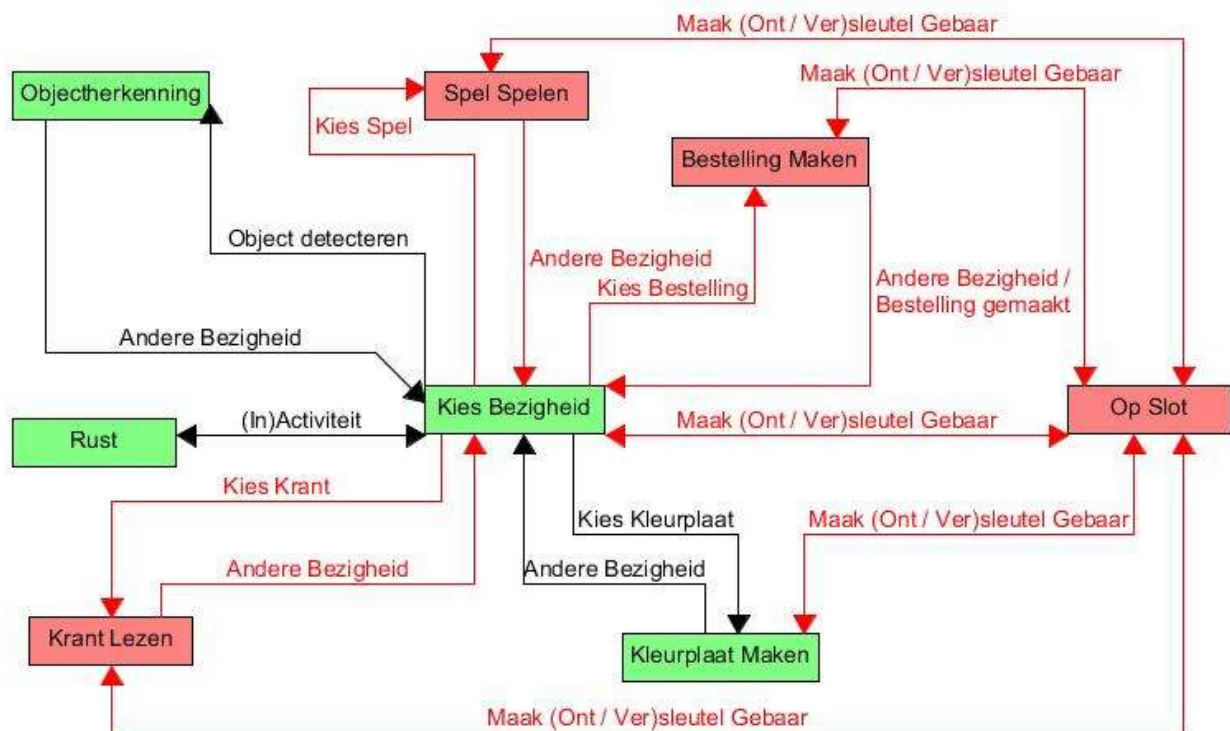


Diagram 1. State Diagram

23.C.IV.a.ii Testresultaten

Toestand	Verwacht	Gevonden
Rust	Toestand stabiel	Toestand stabiel
Rust → Kies Bezigheid	Transitie Geslaagd	Transitie Geslaagd
Kies Bezigheid	Toestand stabiel	Toestand stabiel
Kies Bezigheid → Rust	Transitie Geslaagd	Transitie Geslaagd
Kies Kleurplaat	Transitie Geslaagd	Transitie Geslaagd
Kleurplaat Maken	Toestand stabiel	Toestand stabiel
Andere Bezigheid	Transitie Geslaagd	Transitie Geslaagd
Object detecteren	Transitie Geslaagd	Transitie Geslaagd
Objectherkenning	Toestand stabiel	Toestand stabiel
Andere Bezigheid	Transitie Geslaagd	Transitie Geslaagd

Tabel 2. STT Resultaten

Zoals u hierboven in tabel 5 kunt zien zijn de resultaten van de uitgevoerde test gelijk aan de verwachte resultaten.

23.C.IV.b. Systeem Test (ST)

Bij deze test wordt naar de Functionele Requirements gekeken die vooraf zijn bepaald. Vervolgens wordt gekeken of het systeem hier aan voldoet. Omdat bij dit project er geen eindgebruiker of klant beschikbaar is zal deze test in plaats daarvan door de Developer en de Opdrachtgever worden uitgevoerd.

In de onderstaande tabel 6 kunt u de Functionele Requirements terugvinden die betrekking hebben op de Uitbreidingsmodule. Verder wordt aangegeven wat de verwachte en gevonden resultaten zijn.

ID Requirement	Omschrijving Requirement	Verwacht Resultaat	Gevonden Resultaat
F01 Actie-Reactie	De Projectie en de Motion Detectie moeten reageren op elkaar	Er is Actie-Reactie tussen de Motion Detectie en de Projectie	Er is Actie-Reactie tussen de Motion Detectie en de Projectie
F06 Objecten	Het systeem moet objecten kunnen detecteren en er een berichtje bij kunnen zetten	Het systeem herkent objecten die worden neergelegd	De objecten worden niet herkend omdat het product niet af is
NF05 Teken	De kinderen moeten zichzelf met een kleurplaat kunnen intekenen	Je kan de projectie inkleuren	De projectie wordt ingekleurd met je bewegingen

Tabel 3. ST Test Resultaten

23.C.IV.c. Informele Test

Ik heb ervoor gekozen om deze informele test te gebruiken omdat ik anders de code niet kan testen. Een andere manier van Code Testen zou de Unit Test zijn echter kan door hardware afhankelijkheid dit niet worden uitgevoerd.

Ik begrijp dat dit soort test geen officiële erkenning heeft en over het algemeen door de programmeurs in de ontwikkelomgeving worden gedaan. Ook nu is dit het geval echter moet ik er toch voor zorgen dat ik het systeem genoeg test en niet zonder testdekking het product oplever.

In de onderstaande tabel vindt u de resultaten van deze tests.

Module waarop wordt getest	Omschrijving Uitvoering Test	Verwacht Resultaat	Gevonden Resultaat
Kleurplaat	De tester plaatst zijn/haar hand in het afbakeningsgebied voor de (de)activering van het tekenen voor 10 seconden	De tekenfunctie wordt uit/aan gezet	De tekenfunctie wordt uit/aan gezet
	De tester plaatst zijn/haar hand in het afbakeningsgebied voor de (de)activering van het tekenen voor 5 seconden	De tekenfunctie blijft aan/uit	De tekenfunctie blijft aan/uit
	De tester verplaats zijn/haar handen in het gebied om te tekenen	De kleurplaat wordt ingekleurd	De kleurplaat wordt ingekleurd
Object herkenning	Keyboard en muis neergezet in het detectiegebied	Het systeem herkend het keyboard en de muis en geeft er een weetje bij	Geen herkenning
	Keyboard neergezet in het detectiegebied	Het systeem herkend het keyboard en de muis en geeft er een weetje bij	Geen herkenning
Object registratie	Keyboard en muis neergezet in het detectiegebied om te registreren	Twee Point Cloud's met het keyboard en de muis apart	Een Point Cloud met het keyboard, de muis en het stuk tafel waar ze op liggen
	Keyboard neergezet in het detectiegebied om te registreren	Een Point Cloud met alleen het keyboard	Een Point Cloud met het keyboard en het stuk tafel waar het op ligt

Tabel 4. Resultaten Informele Test

23.C.V. Planning

In dit hoofdstuk kunt u in onderstaande tabel 7 de startdatums en deadline zien voor het uitvoeren van de tests op de Uitbreidingsmodule.

Test	Startdatum	Deadline
STT	14-maart-2016	18-maart-2016
ST	14-maart-2016	18-maart-2016
Informele Test	11-januari-2016	11-maart-2016

Tabel 5. Planning Test Uitvoering