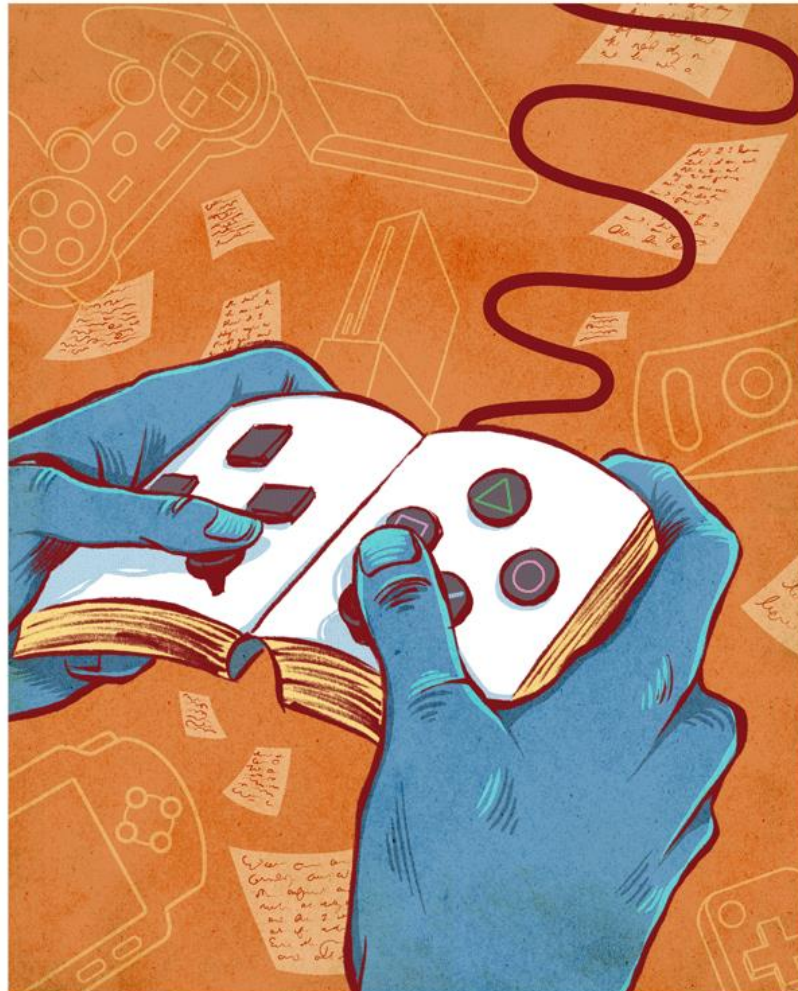


Onderzoeksrapport
Verhalen binnen de EMont visualiseren in een
interactieve 3d wereld



Afstudeerscriptie van Laurens Voncken
Studentnummer 00066711
In het kader van de opleiding HBO-ICT aan de
HZ University of Applied Sciences, Vliissingen

Stagebegeleider: Jolène Cijssouw
Bedrijfsbegeleider: Anton Bil
Datum: 23-1-2018

Onderzoeksrapport

Verhalen binnen de EMont visualiseren in een interactieve 3d wereld

[Afbeelding: interactieve vertelverhaal in een computerspel]

Afstudeerscriptie van Laurens Voncken
Studentnummer 00066711
In het kader van de opleiding HBO-ICT aan de
HZ University of Applied Sciences, Vlissingen

Stagebegeleider: Jolène Cijssouw
Bedrijfsbegeleider: Anton Bil
Datum: 23-1-2018

Versiebeheer

Versie	Auteur	Omschrijving
0.1	L.S. Voncken	Onderzoeksvoorstel, eigen concepten <ul style="list-style-type: none">- Herschreven Probleemstelling uit het startdocument op basis van eerste feedback van Mathieu Starink en Jolène Cijssouw.- Opgesteld conceptversie van de Methode en Materialen
0.2	L.S. Voncken	Onderzoeksvoorstel, eerste formele concept <ul style="list-style-type: none">- Herschreven Probleemstelling en Methode en Materialen op basis van feedback van Jolène Cijssouw. (Betere tekstuele samenhang; verwijdering van overbodigheden en concretisering van onderzoeksmethode en gebruikte begrippen)- Verbeterde APA-notatie door het gehele document.
1.3	L.S. Voncken	Onderzoeksvoorstel, eigen concepten <ul style="list-style-type: none">- Toevoeging van herschreven probleemstelling en Methode en Materialen op basis van tweede feedback van Jolène Cijssouw.
1.4	L.S. Voncken	Onderzoeksvoorstel, tweede formele concept <ul style="list-style-type: none">- Op basis van tweede feedback van Jolène Cijssouw:<ul style="list-style-type: none">• Taal en spellingsfouten verwijderd.• Theoretisch kader ingekort.• Afbeeldingen verplaatst naar bijlagen
2.0	L.S. Voncken	Onderzoeksrapport, eigen concepten <ul style="list-style-type: none">- Resultaten toegevoegd.
2.1	L.S. Voncken	Onderzoeksrapport, eerste formele concept <ul style="list-style-type: none">- Resultaten aangepast op basis van tweede feedback van Jolène Cijssouw. (Voornameeljk irrelevante tekst verwijderd)
2.2	L.S. Voncken	Onderzoeksrapport, eigen concepten <p>Discussie toegevoegd.</p>
2.3	L.S. Voncken	Onderzoeksrapport, tweede formele concept <ul style="list-style-type: none">- Resultaten en Discussie aangepast op basis van tweede feedback uit de mail van Jolène Cijssouw en Anton Bil's opmerkingen. (Ontbrekende adviezen en referenties naar andere theorieën toegevoegd)
3.0	L.S. Voncken	Onderzoeksrapport, definitief, eerste kans <ul style="list-style-type: none">- Taal in het rapport verbeterd.- Tabel en figuur referenties gerepareerd.

Voorwoord

In mijn afstudeerperiode heb ik onderzoek mogen doen naar de mogelijkheid om binnen een interactieve 3d wereld verhalen te vertellen die komen uit de Semantische MediaWiki's van het Expertise and Valorisation Management kenniscentrum (EVM) van de HZ. Het is voor mij een eer geweest om af te studeren bij het EVM. Het werken met actuele semantische technieken en begrippen, en de samenwerking met de experts van het EVM heeft mij toffe nieuwe ervaringen en kennis opgeleverd. Het is voor mij een leuke uitdaging geweest om deze omgeving te verbinden met de technologieën en begrippen uit de virtuele (3d) spelontwikkeling.

Ik zou graag al mijn collega's willen bedanken voor de onvergetelijke momenten die ik heb beleefd gedurende mijn afstudeerstage. Ik zou graag een speciaal bedankje willen geven aan mijn opdrachtgever Hans de Bruin en stagebegeleider Anton Bil voor hun goede begeleiding en ondersteuning bij mijn onderzoek. Zonder hen zou het onderzoek geen kans van slagen hebben gehad. Hans heeft mij enorm geholpen met het vinden van de meest cruciale informatie van mijn onderzoek. Anton heeft mijn onderzoeksmethodes en toegepaste beroepscompetenties steeds gecorrigeerd met een duidelijke en een professionele houding. Naast de medewerkers van het EVM zou ik ook een bedankje willen geven aan mijn stagedocente Jolène Cijssouw, oud stagedocent Mathieu Starink, mijn vader Sjef Voncken, mijn moeder Ineke Saarberg en mijn broer Martin Voncken. Zonder jullie mentale steun en taalkennis zou het mij niet gelukt zijn om een onderzoeksrapport met voldoende niveau te produceren.

Inhoud

I Samenvatting.....	9
II Abstract	10
1 Inleiding.....	11
1.1 Aanleiding.....	11
1.2 Probleemstelling.....	12
1.2.1 Doelstelling	12
1.2.2 Vraagstelling	13
1.3 Randvoorwaarden	13
1.4 Theoretisch kader	14
1.4.1 Expertise Management Methodology (EMM)	14
1.4.2 Expertise Management ontology (EMont) en Concept mapping.....	14
1.4.3 Narratives.....	16
1.4.4 Realisatie van een narrative in een Interactive 3d wereldmodel.....	17
2 Methode en materialen	19
2.1 Deelvraag 1.....	19
2.1.1 Vraag & oriëntatie.....	19
2.1.2 Benadering/Strategie	19
2.1.3 Meetinstrumenten en operationalisatie.....	19
2.1.4 Analysemethode	20
2.2 Deelvraag 2.....	21
2.2.1 Vraag & oriëntatie.....	21
2.2.2 Benadering/Strategie	21
2.2.3 Meetinstrument en operationalisatie.....	21
2.2.4 Analysemethode	22
2.3 Deelvraag 3.....	23
2.3.1 Benadering/Strategie	23
2.3.2 Meetinstrument en operationalisatie.....	23
2.3.3 Analysemethode	23
2.4 Deelvraag 4.....	24
2.4.1 Benadering/Strategie	24
2.4.2 Meetinstrumenten en operationalisatie.....	24
2.4.3 Analysemethode	25
2.5 Totaalbeeld.....	25
3 Resultaten	26

3.1	Dv1 – de EMont structuur uit interview en SMW-bronnenonderzoek.....	26
3.1.1	Resultaten.....	26
3.1.2	Analyse.....	28
3.2	Deelvraag 2 – Adviesstructuur voor interactieve 3d wereld	29
3.2.1	Resultaten.....	29
3.2.2	Analyse.....	30
3.3	Deelvraag 3 – Omzetting van de EMont gegevens naar de interactieve 3d wereld	31
3.3.1	Resultaten.....	31
3.3.2	Analyses	32
3.4	Deelvraag 4 – Valideren advies bevindingen met prototype.....	33
3.4.1	Resultaten.....	33
3.4.2	Analyses	33
4	Discussie.....	34
4.1	Deelvraag 1 – Analyse op het eerste interview en een SMW-wiki	34
4.1.1	Discussie.....	34
4.1.2	Deelconclusie.....	35
4.2	Deelvraag 2 – Adviesstructuur voor interactieve 3d wereld	35
4.2.1	Discussie.....	35
4.2.2	Deelconclusie.....	36
4.3	Deelvraag 3 – Adviesstructuur voor interactieve 3d wereld	36
4.3.1	Discussie.....	37
4.3.2	Deelconclusie	37
4.4	Deelvraag 4 – Valideren advies bevindingen met prototype.....	37
4.4.1	Discussie.....	38
4.4.2	Deelconclusie	38
5	Conclusie en aanbeveling.....	39
5.1	Vergelijking met ander onderzoek/theorie.....	40
5.2	Aanbeveling voor vervolgonderzoeken.....	40
5.2.1	Standaard voor een L2 van de EMont en de EMont toepassing binnen de huidige EVM-SMW	40
5.2.2	Aanvullend gebruikersinteractie onderzoek en validatie van het Aesthetic “narrative”	41
5.2.3	Toepassen van een semantische mapping framework (gebouwd op dotNetRDF) ...	41
6	Literatuur.....	42
	Bijlagen.....	45

Bijlage A.	Opsomming semiotiek in narratives	46
Bijlage B.	Compact overzicht van de drie tijdstructuren voor een interactief medium.....	47
Bijlage C.	Mechanics-Dynamics-Aesthetics (MDA)	48
Bijlage D.	Methode (aanpak en producten) van dv2.....	49
Bijlage E.	Zoekplan voor MDA-onderdelen	50
Bijlage F.	Zoekplan van het bronnenonderzoek van dv4.....	51
Bijlage G.	Game usability en accessibility design patterns	52
Bijlage H.	EMont en requirements interview (uit het analysedocument).....	55
Bijlage I.	Bronnenonderzoek naar statische de EMont structuur binnen een SMW	61
Bijlage J.	Bronnenonderzoek naar dynamische de EMont structuur binnen een SMW	67
Bijlage K.	Statische uitdrukking van de EMont binnen een EVM-SMW met feedback.....	71
Bijlage L.	Dynamische uitdrukking van enkele bestaande in de EMont vastgelegde narratives	72
Bijlage M.	Functionele eisen	73
Bijlage N.	Technische eisen	79
Bijlage O.	Validatie en feedback (uitbreiding bijlage Stap 4).....	81
Bijlage P.	Functionele en technische eisen (Prioriteiten).....	84
Bijlage Q.	Adviesrapport	86
Bijlage R.	Functionele ontwerp	109
Bijlage S.	Technische ontwerp	120
Bijlage T.	Klassendiagram van de interactieve 3d wereld.....	134
Bijlage U.	Prototype testen & Resultaten	136

I Samenvatting

Dit rapport beschrijft het afstudeeronderzoek voor de opleiding HBO-ICT. Dit onderzoek heeft zich gericht op het modelleren en visualiseren van verhalen uit een toepassing van de "Expertise Management ontology" (EMont) binnen een interactieve 3d wereld.

De Expertise Management Methode (EMM) is een probleemoplossingsmethode ontwikkeld door het "Expertise Management and Valorisation" kenniscentrum (EVM-kenniscentrum). De EMont is een ontologie die (menselijk) handelen beschrijft in acht verschillende concepten en wordt toegepast als structuur binnen Semantische MediaWiki's voor de centrale vastlegging van kennis binnen de EMM.

De "3d wereld" zal een 3d representatie zijn van de concepten uit de EMont. De interactie in deze 3d wereld zal bestaan uit te ondernemen activiteiten als meeslepende realistische navigatie door situaties. Deze situaties en activiteiten zijn concepten toegepast in de EMont. De interactieve 3d wereld zal bijdragen aan het leveren van gesimuleerde meegemaakte situaties van betrokkenen (narratives), zodat begrip kan worden gestimuleerd.

Het EVM-kenniscentrum wil met dit product voor alle betrokkenen begrip creëren voor een probleem dat lectoren aan de hand van de EMM proberen op te lossen.

In het onderzoek zijn literatuuronderzoeken uitgevoerd en interviews gehouden. Als resultaat van deze onderzoeksmethoden is een statisch model van de interactieve 3d wereld ontworpen. Het model is toegepast in een functioneel en technisch ontwerp dat is getest met acceptatie-, unit- en integratietesten.

De genoemde onderzoeksmethoden zijn uitgevoerd om de volgende hoofd- en deelvragen te beantwoorden:

Hoofdvraag:

Hoe kan de EMont worden verwerkt in een interactief 3d wereldmodel, zodat narratives kunnen worden verteld?

Deelvragen:

- Wat is de (statische en dynamische) structuur van de EMont binnen een EVM-SMW?
- Welk statisch model voor een interactieve 3d wereld kan de structuur van de EMont implementeren?
- Hoe kunnen de EMont gegevens worden omgezet in het statisch model van de interactieve 3d wereld?
- Is het mogelijk om aan de hand van de bevindingen een prototype op te stellen dat de EMont gegevens uit een EVM-SMW kan omzetten in een interactieve 3d wereld, zodat narratives kunnen worden verteld?

Het onderzoek heeft geconcludeerd dat het mogelijk is om de narratives uit de EMont te leveren in het gemaakte prototype. In de conclusie is ook een advies gegeven over de compleetheid en correctheid van de vastlegging van de EMont en hoe de 3d wereld de narratives meer overeenstemmend kan aanleveren met de wensen en eisen van de opdrachtgever Hans de Bruin. In de conclusie zijn vervolgonderzoeken geadviseerd. De vervolgonderzoeken gaan over de functionele en technische implementatie van de interactieve 3d wereld en de toepassing van de EMont binnen de bestudeerde casus.

II Abstract

This report discusses the graduation research for the HBO-ICT study program. This research focused on modelling and visualizing stories from an application of the "Expertise Management ontology" (EMont) within an interactive 3d world.

Expertise Management Method (EMM) is a problem-solving method developed by Expertise Management and Valorisation knowledge centre (EVM knowledge centre). The EMont is an ontology that describes (human) actions in eight different concepts and is applied as a structure within Semantic MediaWiki's used for the centralised storage of knowledge within the EMM.

The "3d world" will be a 3d representation of the concepts that are used in the EMont. The interaction within the world exists out of choosing activities as an immersive and realistic navigation through situations. These situations and activities are concepts defined within de EMont. The interactive 3d world will provide a realistic simulation of the experiences of the people that are involved in a social problem (narratives), so the EVM-knowledge centre can use this product to stimulate understanding of- and for each person that is involved in the problems that the lecturers try to solve using the EMM.

Literature studies and interviews were conducted in the study. Because of these research methods, a static model of the interactive 3d world has been designed. The model has been applied in a functional and technical design that has been tested with acceptance, unit and integration tests.

The research methods mentioned have been carried out to answer the following main and sub questions:

Main question:

How can the EMont be processed in an interactive 3d world model so that narratives can be told?

Sub-questions:

- What is the (static and dynamic) structure of the EMont within an EVM-SMW?
- Which static model for an interactive 3d world can implement the structure of the EMont?
- How can the EMont data be converted into the static model of the interactive 3d world?
- Is it possible to draw up a prototype based on the findings that EMont can convert data from an EVM-SMW into an interactive 3d world, so that narratives can be told?

The study concluded that it is possible to deliver the narratives from the EMont in the prototype that was made. The conclusion also includes an advisory report on the completeness and correctness of the recording of the EMont and describes how the 3d world can supply the narratives more in accordance with the wishes and requirements of the client Hans de Bruin. In the conclusion, follow-up studies were advised. The follow-up studies are about the functional and technical implementation of the interactive 3d world and the application of de EMont within the studied case.

1 Inleiding

In dit hoofdstuk wordt behandeld waarom dit onderzoek nodig is. In hoofdstuk twee zal per onderzoeksvraag worden besproken met welke onderzoeksmethode en materialen de vraag wordt beantwoord. In hoofdstuk drie zal worden verteld welke resultaten zijn verkregen uit de toepassing van de onderzoeksmethoden en materialen. In hoofdstuk 4 zullen er per onderzoeksvraag conclusies worden getrokken op basis van de resultaten en er zal gereflecteerd worden op de aanpak van het onderzoek, om aan te geven wat het betekent voor het onderzoek. Tenslotte zal in hoofdstuk 5 een afsluitende conclusie met advies worden gegeven. Dit hoofdstuk bespreekt de mate waarin de hoofdvraag is beantwoord, hoe dit is terug te koppelen aan bestaande theorieën en welke vervolgonderzoeken geadviseerd worden.

1.1 Aanleiding

Aan de HZ University of Applied Sciences bevindt zich het kenniscentrum Expertise and Valorisation Management (EVM). Het EVM heeft een aanpak ontwikkeld om praktijkgericht onderzoek uit te voeren, genaamd de Expertise Management Methodologie (EMM). Het praktijkonderzoek gebeurt in Living Labs. De Living Labs stimuleren de ontwikkeling van zogenoemde Coalition of the Willing: een samenwerkingsverband waarin de betrokkenen hun expertise inzetten om de meest haalbare en wenselijke oplossingen te vinden. Geleid door de EMM wordt kennis, opgedaan uit praktijkonderzoek, geanalyseerd en op een gemeenschappelijke plaats vastgelegd. De gemeenschappelijke vastlegging gebeurt in Semantische MediaWiki's (SMW's) van het EVM. (Going meta: the starting points of EMM, 2017)

Wanneer een lectoraat zijn onderzoek doet op een maatschappelijk probleem dan heeft dit te maken met een "wicked problem". Een wicked problem is een probleem dat moeilijk of onmogelijk is op te lossen. Dit als gevolg van onvolledige, tegenstrijdige en veranderende eisen die vaak moeilijk te herkennen zijn. Gewoonlijk worden wicked problems veroorzaakt door tegenstrijdige wereldbeelden van de betrokken deelnemers. Om deze wicked problems acceptabel en uitvoerbaar op te lossen met de EMM, structureert en visualiseert het EVM de relevante kennis in concept maps. Een concept map is een twee dimensionale weergave van probleem relevante kennis, die ook aangeeft hoe deze kennis is georganiseerd in bepaalde situaties (contexten). (Expertise and Valorisation Management, 2017) (Bruin, An Ontology about Expertise Management, 2016)

Een concept map vormt zich uit de Expertise Management ontology (EMont). De EMont is een universele, algemene visuele 'taal' om kennis, expertise en (menselijk) handelen binnen specifieke contexten te beschrijven en te structureren in een statische weergave. Zo wordt de EMont gebruikt als structuur binnen de EVM-SMW's voor het vastleggen van kennismodellen, casus studies en projecten. (Bruin, An Ontology about Expertise Management, 2016)

Nu wil het EVM een acceptabele en uitvoerbare oplossing vinden, door het creëren van begrip tussen de betrokken deelnemers. Dit maatschappelijke begrip zal volgens Hans de Bruin (EVM-medewerker) moeten worden gestimuleerd door elkaars situaties te begrijpen. Hans de Bruin wil het begrip stimuleren door de meegemaakte praktijkvoorbeelden, zoals ervaringen, van deelnemers te simuleren. Dit, door de narratives aan te leveren in een dynamisch te doorlopen 3d concept map, ofwel een interactieve 3d wereld.

De EMont uit een EVM-SMW definieert situaties met daarin activiteiten die een actor of rol kan uitvoeren. Dit vormt de basis voor de gewenste 3d wereld. De activiteiten kunnen verwijzen naar andere activiteiten. Door de gebruiker de activiteit(en) te laten uitvoeren kan deze worden begeleid door de situatie en vormt er een narrative. Door het meenemen van de, ook in de EMont te vinden, praktijkvoorbeelden kunnen de te kiezen activiteiten gelimiteerd worden binnen specifieke situaties. Zo kan de gebruiker worden begeleid bij het doorlopen van de activiteiten en het belanden in dezelfde situaties als de actoren of rollen die behoren tot de relevante praktijkvoorbeelden. Op deze manier kan er een specifieke narrative worden gevolgd.

Het EVM verlangt naar het inzetten van de interactieve 3d wereld bij alle Living Labs. Hiervoor is er expliciet gevraagd naar een mogelijkheid om een ontwerper invloed te laten hebben op het uiterlijk van de 3d wereld waarin de narratives afspelen. (Bruin, EVM problem explanation, 2017) (Hofstadter & Sander, 2013)

1.2 Probleemstelling

Het EVM-kenniscentrum gebruikt concept maps als visuele weergave van de vastgelegde probleemsituaties. Voor het oplossen van de complexe problemen is begrip vereist van de deelnemers. De huidige 2d concept maps tonen de situaties in een statisch overzicht dat geen praktijkvoorbeelden of ervaringen gebruikt uit de EMont. Door deze limieten van de concept map kunnen geen ervaringen van betrokkenen (narratives) worden verteld die het EVM kan gebruiken om begrip te stimuleren.

Het EVM vraagt naar een interactieve 3d wereld die narratives kan vertellen. Het probleem is dat er een ontwerp ontbreekt dat beschrijft hoe in een softwareapplicatie de EMont concepten gevisualiseerd kunnen worden als interacteerbare 3d objecten binnen een virtuele wereld. Het EVM wil dat de 3d objecten worden aangeleverd op basis van de door de gebruiker gekozen praktijkvoorbeelden, de ondernomen activiteiten en eventuele ontwerpkeuzes van een ontwerper voor de situatie waarin de gebruiker zich bevindt. (Bruin, An Ontology about Expertise Management, 2016)

1.2.1 Doelstelling

Het doel van het onderzoek is om een prototype van een applicatie op te leveren aan het EVM. Dit prototype levert de EMont concepten uit een EVM-SMW-dataset als interacteerbare 3d objecten. De interactie en weergave van de 3d objecten wordt bepaald aan de hand van de door de gebruiker gekozen praktijkvoorbeelden en ondernomen activiteiten en ontwerpkeuzes voor situaties, zodat er een narrative kan worden doorlopen. Het prototype zal worden ontworpen op basis van een gevalideerd standaard statisch model van een interactieve 3d wereld die is gebouwd uit de statische en dynamische structuur zoals deze is te vinden en in een EVM-wiki's.

1.2.1.1 Relevantie

1.2.1.1.1 *Bedrijfsrelevantie en Maatschappelijke relevantie*

Hans de Bruin, als lectoraat lid aan het EVM-kenniscentrum, kan de interactieve 3d wereld inzetten als simulatie van de verhalen van betrokkenen bij maatschappelijke problemen. Dit test de compleetheid van de EMont en ondersteunt de EMM met een nieuwe visualisatie mogelijkheid. Beide, de EMont en de EMM, zijn door Hans de Bruin ontwikkeld en ingezet binnen het EVM-kenniscentrum. Dit maakt het onderzoek relevant voor Hans de Bruin en het EVM-kenniscentrum. (Bruin, An Ontology about Expertise Management, 2016)

EMont en de EMM, dragen bij aan het leveren van acceptabele en uitvoerbare oplossingen voor de maatschappelijke problemen die de Living Labs in samenwerking met het EVM-kenniscentrum probeert op te lossen. (Expertise and Valorisation Management, 2017) De applicatie kan worden toegepast bij bijvoorbeeld het "Overleven met kanker" EVM-project dat zich bezighoudt met het ondersteunen van mensen met kanker. Dit bewijst dat het onderzoek maatschappelijk relevant is. (Overleven met Kanker, 2017)

1.2.1.1.2 *Kennisgebied en theoretische relevantie*

Het onderzoek is relevant voor het kennisgebied, omdat het een nog niet bestaande koppeling legt tussen het gebruik van de EMont als basis voor een interactieve 3d wereld. Het onderzoek zal ook aangeven in welke maten de EMont een adaptieve dynamiek aanbiedt voor het vertellen van verhalen in verschillende plotstructuren binnen applicaties.

1.2.2 Vraagstelling

1.2.2.1 Onderzoeksvraag (Ov) & Deelvragen (D.V.)

De vraag die dit onderzoek probeert te beantwoorden luidt:

Ov - *Hoe kan de EMont worden verwerkt in een interactief 3d wereldmodel, zodat narratives kunnen worden verteld?*

De onderzoeksvraag is op te delen in de volgende deelvragen:

Dv 1 - Wat is de (statische en dynamische) structuur van de EMont binnen een EVM-SMW?

Dv 2 - Welk statisch model voor een interactieve 3d wereld kan de structuur van de EMont implementeren?

Dv 3 - Hoe kunnen de EMont gegevens worden omgezet in het statisch model van de interactieve 3d wereld?

Dv 4 – Is het mogelijk om aan de hand van de bevindingen een prototype op te stellen dat de EMont gegevens uit een EVM-SMW kan omzetten in een interactieve 3d wereld, zodat narratives kunnen worden verteld?

1.3 Randvoorwaarden

Het onderzoek kent de volgende randvoorwaarden (Bruin, EVM problem explanation, 2017):

- 1) De (statische en dynamische) structuur zal worden herleid uit analyses op het web domein van de EVM-SMW: www.projectenportfolio.nl.
- 2) De EVM-SMW de EMont-gegevens die het product kan inlezen zullen komen uit de aangeboden exporteer mogelijkheden van een EVM-SMW.
- 3) Het statisch model van de interactieve 3d wereld zal zodanig worden opgesteld dat een ontwerper de 3d wereld kan vormen.
- 4) Het product dient minimaal twee narratives van diverse actoren of rollen te kunnen visualiseren in de interactieve 3d wereld.
- 5) De interactieve 3d wereld en bijbehorende interactie zal worden gebouwd binnen de Unity game engine.
- 6) Er zal binnen Unity alleen worden geprogrammeerd met C#.

1.4 Theoretisch kader

In het theoretisch kader zullen de bestaande relevante concepten en theorieën worden uitgelegd.

1.4.1 Expertise Management Methodology (EMM)

EMM is een methodiek die is ontwikkeld door Hans de Bruin. De essentie van de EMM is het op een gestructureerde wijze boven water halen van de wereldbeelden van belanghebbenden. De EMM draagt bij aan het analogie denken (ervaringsleer) en het bewustmaken van situaties. De EMM bestaat uit vier pijlers (EMM, de theorie en de praktijk - Projectenportfolio, 2017):

- Soft Systems Methodology (SSM), het binnen de EMM gebruikte raamwerk voor actieonderzoek.
- Expertise Management ontology (EMont), een ontologie voor het vastleggen van knowing-that (feiten en concepten) en knowing-how (skills) kennis.
- Concept mapping, een 2d visuele weergave van de kennisstructuren en het verbinden van nieuwe kennis met bestaande kennis. De structuur is beschreven met de concepten uit de EMont.
- Body of Knowledge & Skill: de Semantische wiki, een centrale semantische opslag voor gebruiker gegenereerde content. Voor het managen van de content worden MediaWiki's gebruikt met Semantische MediaWiki plug-in.

1.4.1.1 EMM en Soft System Methodologie (SSM)

EMM heeft te maken met de wicked problems (uitgelegd in de aanleiding). SSM is de methode die speciaal is ontwikkeld om deze problemen gestructureerd aan te pakken. De SSM-aanpak bestaat uit een iteratief proces van vier stappen (Steenbeek, 2015) (EMM, de theorie en de praktijk - Projectenportfolio, 2017):

1. Finding out: Het verdiepen in de situatie waar het probleem speelt.
2. Model building: Het modelleren van de situatie vanuit ieder zijn wereldbeeld of standpunt.
3. Discussing and debating: Discussieer over de diverse standpunten en kom zo tot een oplossing.
4. Taking action: Het ondernemen van actie op basis van de uitkomsten.

1.4.2 Expertise Management ontology (EMont) en Concept mapping

Een ontologie is typisch een datastructuur die alle relevante entiteiten en hun onderlinge relaties en regels binnen een domein bevat.

De Expertise Management ontology (EMont) biedt de mogelijkheid tot het beschrijven van de "knowing-how" (gedrag en de activiteiten) kennis en de wereldbeelden van menselijk handelen in specifieke situaties. De EMont modelleert de kennis en expertise in elementen als concepten en relaties. De concepten en relaties van de EMont kunnen worden gevisualiseerd met behulp van concept maps. (Expertise Management ontology, 2016)

De activiteiten (de expertise) van mensen en niet-mensen (zoals machines) kan worden vastgelegd met behulp van de PQR-formule. De PQR-formule is leidend voor de EMont en is afkomstig uit de SSM.

De PQR-formule bestaat uit de P en Q als activiteiten en R als het doel dat men wil bereiken. "P" beschrijft wat voor een activiteit of proces wordt uitgevoerd om een doel "R" te kunnen bereiken. "Q" beschrijft hoe de activiteit of proces "P" wordt gedaan. "Q" is een sub-activiteit of sub-proces van "P". (HZ University of Applied Sciences, 2017)

Het tweede niveau (ofwel L1) is “Domain-specific ontology”. In dit niveau worden instanties gemaakt van de foundation concepten. Deze instanties beschrijven samen specifieke situatie voor het menselijk handelen. De handelingen (ofwel activiteiten) binnen deze situaties heten handelingsperspectieven.

Het derde niveau (ofwel L2) is “Experiences”. In dit niveau worden de praktijkvoorbeelden beschreven. Een praktijkvoorbeeld beschrijft voor een context welke combinatie van uitgevoerde handelingsperspectieven in de praktijk, of in een actor zijn ervaring, positief of niet positief zijn geweest.

De wiki vormt de EMont door de concepten per niveau semantisch vast te leggen. De semantische vastlegging van de concepten gebeurt in de standaard Simple Knowledge Organization System (SKOS). (Bruin, EVM problem explanation, 2017)

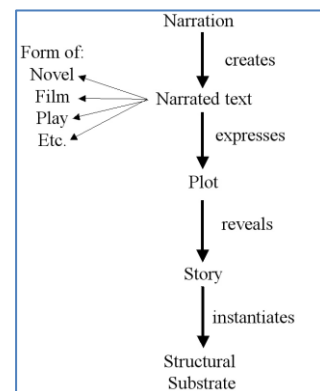
1.4.3 Narratives

Nu wil het lectoraat dat dit onderzoek kan ontdekken op welke manier de “narratives” kunnen worden gevisualiseerd in een interactieve 3d wereld.

De term “narrative” is een Engels woord voor “vertelverhaal”. De semantisch vastgelegde narratives vertellen de meegemaakte situaties aan de hand van de uitgevoerde activiteiten van de betrokken personen van een probleem. Narrative bestaan uit een zeven elementen. In het onderzoek zal moeten worden achterhaalt hoe de EMont toepassing deze elementen beschrijft.

Een narrative bestaat uit de volgende elementen: De omgeving, het thema, het plot, de karakters, het perspectief, de toon en de toevoeging van beeldwerk en symbolisme. (Kissner, 2006) Deze elementen zijn verwerkt in een algemene structuur van een narrative. Deze structuur is te zien in Figuur 3. De structuur bestaat uit (Lindley, 2005):

- **Narration (vertelling):** Is de handeling of het proces van de productie van de tekst. De tekst zelf is het verhaal. Hierin bevinden zich de narrative elementen “toon” en “perspectief”.
- **Narrated tekst:** Een instantie in tijd, gemaakt door de Narration, die verhalen vertelt. Het concept van de “tekst” omvat zowel auditieve als visuele media.
- **Het Plot:** Dit zijn narrative elementen die het verhaal onthullen door bepaalde verhaalde gebeurtenissen te benadrukken of af te leiden. Het plot wordt overgebracht aan de hand van de Narrative tekst.
- **De Story:** Dit wijst de vertelde gebeurtenissen aan, die uit hun dispositie in de tekst worden samengevat en in hun chronologische volgorde worden gereconstrueerd, samen met de deelnemers aan deze gebeurtenissen. Dit omvat de narrative elementen: “thema” en “karakters” en “beeldwerk”.
- **De Structural substrate:** de structurele denkwijze (taal en cultuur) waar het verhaal uit wordt opgebouwd.



Figuur 3 Algemene structuur voor semiotiek in narratives (Lindley, 2005)

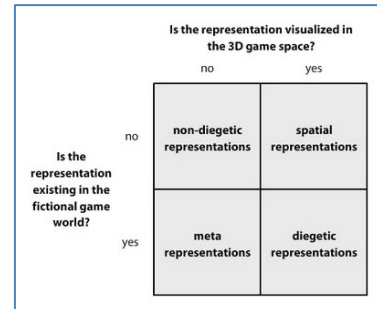
Het narrative element “symbolisme” is terug te vinden in elke van de Narration, de Narrated text, het Plot en de Story van de algemene narrative structuur. Dit kan namelijk metaforisch worden verwerkt in de vertelling (de narration) of als een Allegorie zijn verwerkt door het gehele verhaal (Story). (Thomas, 2017)

1.4.4 Realisatie van een narrative in een Interactive 3d wereldmodel

Het begrip “interactief medium” verwijst naar producten en diensten op digitale computer-gebaseerde systemen die reageren op de acties van de gebruiker door inhoud te presenteren. (Ploeg, 2010) De interactieve 3d wereld is een interactief medium. De structuur van een narrative in een interactief medium kent een gelijke structuur als die van een narrative in het algemeen, die is genoemd in 1.4.3. Aan de hand van verzamelde informatie is een verband te leggen tussen de twee structuren en vereiste ontwerpbeslissingen voor de interactieve 3d wereld. In Bijlage A is in Tabel 2 een opsomming gegeven van deze verbanden.

1.4.4.1 Functioneel ontwerp beslissingen narrative in een Interactive 3d wereld

In Tabel 2 is te zien dat de Narration, Narrated text en het plot onderdeel zijn van de functionele beschrijving van het systeem. Deze onderdelen van het narrative hebben te maken met de interactie tussen de gebruiker en het verhaal: Het zogenoemde graphical user interface (GUI). GUI houdt zich bezig met welke reactie het systeem moet geven aan de hand van de input van het gebruik en in welke vorm het systeem zijn output levert aan de gebruiker. In Figuur 4 is te zien dat een user interface bestaat uit vier verschillende ontwerpen: diëtetische, Niet Diëtetische-, Ruimtelijke – en Meta GUI-elementen. (Andrews , 2010)



Figuur 4 Termologie van Fagerholt, Lorentzon (2009) "Beyond the HUD - User Interfaces for Increased Player Immersion in FPS Games. (Andrews , 2010)

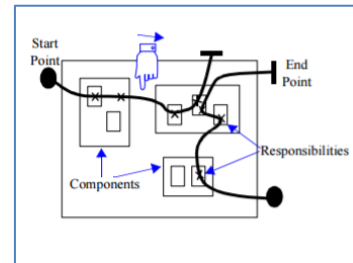
Barry Ip definieert “narrative descriptors” als de verzameling en het gebruik van GUI-elementen om het verhaal te communiceren naar de speler. Deze elementen zijn: tekstuele instructies, korte (cut scenes) en lange filmfragmenten, interface elementen (als besturing en condities), visuele elementen die het gezichtsveld van de speler omvatten (voor het vormen van perspectieven) en interactie tussen de elementen (dynamiek/gameplay). (Ip, 2010)

Het plot is het verloop van het verhaal. Dit wordt in een interactief medium doorlopen in een tijdstructuur of een combinatie van verschillende tijdstructuren. Deze tijdstructuren zijn opgesplitst in Narrative, Game en Simulation in een driehoeksverhouding. In Bijlage B in Tabel 19 is een opsomming te vinden van deze tijdstructuren. Binnen de driehoek kunnen de verschillende categorieën worden geïdentificeerd die het genre aangeven van de interactieve media. Functionaliteit voor een interactief medium kan worden herleid uit de kenmerken van het genre. De keuze voor het genre is afhankelijk van de gewenste implementatie van een tijdstructuur. De Simulatie tijdstructuur, gaat over de onderlinge interactie binnen de elementen uit de 3d wereld. Naarmate dat er meer interactie wordt gekozen tussen de gebruik en de wereld, zal er meer een Game of Narrative tijdstructuur worden toegepast. Het onderscheid tussen een Game en een Narrative tijdstructuur wordt gemaakt door de vrijheidsgraden, dit is het aantal verschillende interactiemogelijkheden die worden aangeleverd. Weinig tot geen vrijheidsgraden zijn kenmerken van een Narrative tijdstructuur, dit is voor een Game tijdstructuur tegenovergesteld.

Een Game tijdstructuur is op te delen in voortgang aan de hand van drie verschillende interactie structuren. De eerste interactie structuur is “Game Moves”, ofwel voortgang door toepassen van interactie. De tweede structuur is “Gameplay Gestalt”, dit gaat over progressie beslist door methode van omgaan met de interactie. De laatste variant is “Higher Level Temporal Structure”, dit is een systeem dat bepaalt wanneer er weer voortgang is, bijvoorbeeld aan de hand van een score of na bepaalde tijd. In een onderzoek van E. W. Adams wordt verteld dat het gebruikelijk is om de Game Move structuur af te stemmen op de doelgroep. Voor narrative maakt Adams onderscheid tussen verhaalgeoriënteerde spelertypes als doelgroep. De verhaalgeoriënteerde spelertypes bestaan uit: “Audience”, “Performer” en “Immresionist”. (Lindley, 2005) (Adams, 2005)

Bij een Narrative tijdstructuur wordt er onderscheid gemaakt tussen twee soorten van structuren. Een lineaire of interactieve narrative. Een lineair narrative kent als dynamiek ofwel plotstructuur een boom, voor gedefinieerde vertakkingen in het verhaal. Het interactie narrative, waaruit het interactief medium van het onderzoek zal vormen, kent de structuren: Exploratorium, Parallele plotstructuur, Knoop- of doodlopende structuur, Gemoduleerde of de dynamische labyrintstructuur en twee varianten van een openstructuur. (Lindley, 2005)

De dynamiek van de interactieve 3d wereld is te herleiden uit het vastgelegde verloop van de narratives binnen de semantische MediaWiki. In een artikel van Hans de Bruin is vermeld dat deze dynamiek kan worden vastgelegd in een Use Case Map (UCM). In Figuur 5 is de notatie en interpretatie van een UCM te zien. UCM is een modelleermethode om gedrag en structuur te koppelen. Hans de Bruin heeft vermeld dat de EMont contexten in UCM te noteren zijn met de componenten elementen en activiteiten als endpoints of responsibilities elementen. Volgens Hans de Bruin kan een EMont Actor of Rol (type Context) worden weergegeven als een startpoint dat in verbinding staat met de endpoints of responsibilities. (Amyot D. , 1999) (Bruin, An Ontology about Expertise Management, 2016)



Figuur 5 Basale notatie en interpretatie van een Use Case Map. (Amyot D. , 2001)

1.4.4.2 Technische ontwerp beslissingen narrative in een Interactive 3d wereld

In Tabel 2 van Bijlage B is te zien dat story wordt gezien als een model, ofwel simulatie. Dit is de wereld van entiteiten met onderlinge relaties en interactie. De statische structuur, ofwel statisch model van de wereld, is een representatie van de entiteiten en onderlinge relaties zoals deze zijn vastgelegd in de wereld van het verhaal. Dit betekent dat het statisch model van de wereld is te herleiden uit de statische structuur van de vastlegging van de EMont binnen de EVM-wiki die deze wereld zal representeren. Het statisch model van de wereld definieert niet alleen de fysieke objecten, maar ook de methode die wordt gebruikt om de representatie van het verhaal binnen de wereld te realiseren en definieert de interactie die kan worden gehouden tussen de in-wereldse objecten en de gebruiker. Dit gehele model wordt beschreven binnen het gekozen "Generative substrate". Het model van de te onderzoeken interactieve 3d wereld zal tevens rekening houden met de (data)transitie van gegevens uit de wiki en deze om te zetten in daadwerkelijke objecten binnen de wereld.

De Generative substrate is het technische systeem dat het mogelijk maakt een "Structural substrate", oftewel de achterliggende taal en cultuur, te kunnen realiseren in een digitale omgeving. In het onderzoek is als randvoorwaarde gesteld dat er gewerkt wordt met de Unity game engine. De Unity game engine is het systeem waarin de 3d wereld in opgezet en ingesteld zal worden. Dit is een cross-platform game engine ontwikkeld door het bedrijf Unity Technologies. Het combineert verschillende technologieën afgedekt onder een gebruiksvriendelijke interface voor definiëren van de 3d of 2d interactieve applicatie. Er zal hiervoor een softwarearchitectuur moeten worden bedacht om de narrative data mee te kunnen uitdrukken in Unity. De gevraagde architectuur voor de narratives is een representatie van de drie EMont niveaus zoals deze vastliggen in de semantische wiki's, maar dan in softwareklasse of objecten. (Bil & Bruin, 2017)

Het statische model van de EMont kan gemodelleerd worden met de standaard Unified Modeling Language (UML). Uit het UML schikt een klassendiagram voor het definiëren van entiteiten en hun onderlinge relaties die bevinden in de vastgelegde EMont (Fowler, 2003). In Figuur 1 is een eerder uitgewerkt klassendiagram van het LO niveau binnen de EMont te zien.

Het statische model zal moeten worden getransleerd naar een model met meegenomen architecturale vereisten en gevalideerde praktijktoepassingen van Unity producten.

2 Methode en materialen

In dit hoofdstuk zal er per deelvraag worden uitgelegd welke onderzoeksmethoden zullen worden toegepast om de deelvraag te beantwoorden. Dit wordt gedaan door eerst in een paragraaf “Vraag & oriëntatie” te verklaren wat voor een informatie nodig is en waarom dit relevant is. Vervolgens zal er onderbouwd worden verteld in de paragraaf “Benadering/Strategie” welke onderzoeksstrategie toepasbaar is voor het beantwoorden. Deze strategie krijgt een concrete invulling over het verzamelen van gegevens in de paragraaf “meetinstrumenten en operationalisatie” en een concrete invulling in de analyse van deze gegevens in de paragraaf “Analysemethode”.

2.1 Deelvraag 1

2.1.1 Vraag & oriëntatie

Dv 1 - Wat is de (statische en dynamische) structuur van de EMont binnen een EVM-SMW?

Voor het vinden van een passend statisch model voor een interactieve 3d wereld zal er duidelijk moeten zijn uit welke entiteiten de wereld minimaal zal moeten bestaan. Ook zal er duidelijk moeten zijn welke attributen deze entiteiten kennen. Dit zal worden herleid uit de statische structuur, zoals is vastgelegd in een EVM-SMW. Het model van de wereld definieert ook het verloop van het verhaal. Het verloop zal worden herleid uit de dynamische structuur van de narratives, zoals deze zijn vastgelegd in een EVM-SMW.

2.1.2 Benadering/Strategie

Deze vraag zal beantwoord worden door het bestuderen van een specifieke Semantische MediaWiki (SMW) van het EVM. Dit zal een casusstudie zijn die specifiek wordt toegepast op de EVM-SMW: www.Projectenportfolio.nl. Deze wiki kent net als de andere EVM-SMW's een verzameling van diverse wikipagina's, waaruit de statische en dynamische uitdrukking van de narratives in de EMont is te herleiden.

Om de kwaliteit van de resultaten te bewaken zullen de resultaten uit de analyses worden voorgelegd aan de Hans de Bruin wegens zijn kennis over de EMont en SMW's. De resultaten en ook de onderzoeksmethode zullen getoond worden aan de stagebegeleider. Hiermee zal de correctheid en compleetheid van de resultaten worden gegarandeerd.

2.1.3 Meetinstrumenten en operationalisatie

Er zal een interview worden gehouden op het HZ. Het interview doelt op het verkrijgen van inzicht over de individuele vastlegging van de verschillende niveaus van de EMont in de verschillende wiki. Dit interview zal semigestructureerd zijn, omdat enkel het bestaan van een statisch en dynamisch structuur van de EMont niveaus in de wiki bekend is. Uit het interview zal duidelijk moeten worden welke entiteiten, attributen, relaties en interacties er zijn uitgedrukt binnen de EVM-SMW's, of uit welke bron deze informatie gehaald kan worden.

De geïnterviewde zal over kennis beschikken van de EMont en de EVM Living Labs. Hans de Bruin heeft de EMont ontwikkeld en deelgenomen aan veel verschillende Living Labs van het EVM. Dit maakt Hans de Bruin de meest geschikte persoon voor het interview. In Tabel 1 staan de topics van het interview met gegeven onderbouwing voor hun relevantie.

Tabel 1 Topics van het semigestructureerde interview van deelvraag 1, met onderbouwing

Topic	Onderbouwing topic
De vorming van de EMont niveaus binnen een EVM-SMW	Met dit topic kan er worden herleid welke concepten van de EMont ontologie terug te vinden zijn binnen de wiki. Hieruit kunnen de EMont entiteiten gevonden worden.
De EVM-SMW-structuur in het algemeen	Met dit topic kan er worden herleid welke relaties te vinden zijn tussen de EMont concepten van het eerste topic.
De koppeling tussen narratives en handelingsperspectieven	Met dit topic kan duidelijk worden hoe een verloop van handelingsperspectieven zijn beschreven als meegemaakte situaties van een actor of rol binnen de EVM-SMW.
De toepassing van de narrative elementen in de vastgelegde narratives	Met dit topic kunnen inzichten worden verkregen over welke EMont elementen in de EVM-SMW een verband hebben met welk narrative element.
Het verloop van narratives	Met dit topic kunnen inzichten worden verkregen over bestaande narratives in de EVM-SMW en hoe deze zijn beschreven in de EMont.
Het bestaan van ondersteunende referenties over de EMont structuur binnen een EVM-SMW.	Met dit topic kunnen inzichten worden verkregen over de EMont structuur (entiteiten, attributen, relaties en interacties), zoals beschreven binnen een EVM-SMW.

Aanvullend op het interview zal een bronnenonderzoek plaatsvinden. Dit bronnenonderzoek zoekt naar de entiteiten, attributen en relaties waaruit de EMont is opgebouwd binnen de EVM-SMW: www.projectenportfolio.nl. De entiteiten zijn binnen de EVM-SMW vastgelegd als wikipagina's. Deze wikipagina's aangemaakt met behulp van formulieren. Via de formulieren worden ook de attributen en relaties voor een pagina ingevuld. Om de entiteiten, attributen en relaties te vinden zal elke wikipagina met formulier worden bekeken. (Bil & Bruin, 2017) De gevonden entiteit met attributen en relaties zal worden genoteerd in een tabel. In de tabel zal er per entiteit worden aangegeven uit welke wikipagina of formulier deze is herleid.

Indien de geïnterviewde refereert naar andere relevante bronnen, zal er een aanvullend bronnenonderzoek worden toegepast. Het bronnenonderzoek zal proberen uit deze bronnen nog niet gevonden entiteiten, attributen en relaties te verkrijgen. De zoektermen binnen deze bronnen zullen worden bepaald door de geïnterviewde.

2.1.4 Analysemethode

Het analyseren zal bestaan uit de volgende vijf stappen:

1. Het **coderen** (markeren van relevante zinnen en woorden binnen teksten) van het interviewtranscript op zoek naar de entiteiten, attributen, relaties tussen entiteiten, interactie tussen entiteiten en eventuele aangeraden te bestuderen bronnen. (Saunders, Lewis, Thornhill, Booi, & Verckens, 2004)
2. Het **visualiseren van de EMont statische structuur** van de bekeken EVM-SMW in een klassendiagram. Dit klassendiagram is op basis van de entiteiten, attributen en de relaties uit de coderingsresultaten. Het gebruik van een klassendiagram is onderbouwd in 1.4.4.2 van het theoretisch kader.
3. Het **visualiseren van de EMont dynamische structuur** van casussen uit de bekeken EVM-SMW in een Use Case Map. Deze Use Case Map is op basis het klassendiagram en de interactie uit de coderingsresultaten. De keuze voor de Use Case Map is gegeven in het theoretisch kader, zie 1.4.4.1.
4. Het **uitbreiden van het klassendiagram en Use Case Map** met eventueel gevonden resultaten ontbrekende entiteiten, relaties en interactie uit de door Hans de Bruin aangeraden bronnen.
5. Het **uitbreiden van het klassendiagram met attributen** dit door de tabel met de gevonden formulieren uit te breiden met de attributen die zijn terug te vinden in de opzet van de formulieren. De attributen worden vervolgens als attributen toegevoegd bij de entiteiten van het klassendiagram.
6. Het **valideren van het klassendiagram en de Use Case Map** bij Hans de Bruin, omdat Hans de Bruin narratives heeft onderzocht en een technische achtergrond heeft met de EVM-SMW's.

De analyses zullen worden gedocumenteerd in een gezamenlijk analysedocument.

2.2 Deelvraag 2

2.2.1 Vraag & oriëntatie

Dv 2 - Welk statisch model voor een interactieve 3d wereld kan de structuur van de EMont implementeren?

Het statisch model beschrijft de structuur van de interactieve 3d wereld. Deze structuur is een combinatie van de technische implementatie (mechaniek) van de "Narrative", "Game" en "Simulatie" tijdstructuur, genoemd in 1.4.4. In de eerste deelvraag is deze mechaniek van de Narrative tijdstructuur al uitgedrukt in een klassendiagram. Nu dient er uitgezocht te worden welke mechanieken uit de Game en Simulatie structuur bijdragen aan het model van de wereld. Tevens zal er worden onderzocht welke technische implementatie vereist is voor de gewenste functionaliteit van de wereld. Dit zal worden verwerkt in een adviesdocument. In het onderzoek naar het advies zullen requirements worden gevonden die zullen worden gedocumenteerd in een analysedocument.

2.2.2 Benadering/Strategie

Er zal een bronnenonderzoek worden gehouden naar de functionele werking van een interactieve 3d wereld met bijbehorende technische vereisten. Aan de hand van een interview met de opdrachtgever Hans de Bruin zullen de functionele eisen worden gelimiteerd, zodat enkel de technische implementatie van de relevante functionele werking van de wereld worden meegenomen in het advies voor een statisch model. Het ontwerpen van het advies zal worden gedaan met behulp van het MDA-raamwerk. Dit wordt nader uitgelegd in Bijlage C.

De kwaliteit van de (tussen)resultaten zal worden gegarandeerd, doordat deze iteratief zullen worden voorgelegd aan de opdrachtgever Hans de Bruin in interviews en demoesessies. Ook door het gebruik van de UML standaard voor modelleren en een ISO-model voor softwarekwaliteit. Ook zal de kwaliteit van de resultaten en onderzoeksmethode worden gegarandeerd, doordat deze worden voorgelegd aan de stagebegeleider.

2.2.3 Meetinstrument en operationalisatie

Er zijn twee varianten van onderzoeksmethode benodigd voor het vinden van een statisch model, dit zijn: bronnenonderzoeken en interviews. In Tabel 3 van Bijlage D is per MDA weergegeven welke onderzoeksmethodes gebruikt worden. Ook geeft de Tabel 3 aan in welk hoofdstuk van welk document het resultaat van de onderzoeksmethoden worden opgenomen.

De bronnenonderzoeken (aangeven als onderzoeksmethode 1, 2 en 3 uit Tabel 3) worden uitgevoerd aan de hand van een compact zoekplan. Dit zoekplan is gegeven in Tabel 4 van Bijlage E.

De gevonden bronnen uit ieder bronnenonderzoek worden genoteerd onder de paragraaf binnen het toegewezen document. De per bronnenonderzoek toegewezen documenten staan vermeld in Tabel 3 van Bijlage D. Op de vastgelegde bronnen zullen de analyses worden toegepast.

Elke interview zal semigestructureerd worden gehouden. Dit met Hans de Bruin als geïnterviewde, omdat hij de opdrachtgever is. Er wordt aan het begin van ieder interview een korte presentatie gehouden over de resultaten

van het bronnenonderzoek waar de topics van het interview overgaan. Beide interviews worden vastgelegd in een transcript.

Het eerste interview (aangegeven als onderzoeksmethode 2 in Tabel 3 van Bijlage D) heeft als onderwerp Aesthetics. Deze kent enkel de topic “emotionele reactie”. Hierin zal worden gevraagd naar Hans de Bruin zijn gedachte over de te stimuleren emotionele reactie met behulp van de dynamiek binnen de 3d wereld.

Het tweede interview (aangegeven als onderzoeksmethode 4 in Tabel 3 van Bijlage D) heeft als onderwerp functionele en technische eisen. Dit interview dient te achterhalen welke functionele en technische eisen Hans de Bruin stelt voor de interactieve 3d wereld. De functionele eisen worden achterhaald aan de hand van topics voor ieder relevant begrip ten aanzien van Dynamics (de functionele ontwerp relevante begrippen genoemd in Tabel 2 van Bijlage A) en een topic voor de gebruiksvriendelijkheid ontwerppatronen die veel worden gebruikt in interactieve media. De technische eisen zullen worden opgesteld door de officiële ISO-25010 softwaremodelkwaliteitscriteria als topics toevoegen aan het interview.

2.2.4 Analysemethode

De analyse van deze deelvraag bestaat uit een aantal stappen:

1. Het **bronnenonderzoek naar mogelijke Aesthetics** zal worden geanalyseerd door een samenvatting te maken van de verschillende soorten Aesthetics met betekenis, zoals deze is vermeld in de bronnen.
2. Het **interview naar relevante gewenste Aesthetics** zal worden geanalyseerd door het transcript te coderen naar Aesthetics genoemd door Hans de Bruin. Uit de codering zal de door Hans de Bruin gewenste Aesthetics herleid. Deze Aesthetics zullen worden gemarkeerd in de samenvatting van het bronnenonderzoek.
3. Het **bronnenonderzoek naar mogelijke Dynamics voor de Aesthetics** zal worden geanalyseerd door middel van het doorlezen van de bronnen op zoek naar de diverse varianten binnen het relevante begrip. Elke variant van een begrip krijgt een rij in een tabel met de kolommen “[variant van het relevante begrip]” en “Functionele toepassing”.
4. Het **interview naar functionele eisen (en technische eisen) voor het filteren relevante Dynamics** zal worden geanalyseerd door codering op het transcript naar woorden of zinnen die beschrijven wat het systeem moet doen. Deze zullen SMART worden vastgelegd volgens een de standaardmethode voor het vastleggen van functionele requirements. (Dijkgraaf & van Spall, 2007) Er zal ook codering worden gedaan naar de woorden die beschrijven wat Hans de Bruin concludeert over de kwaliteit van het systeem ten aanzien van de kwaliteitseisen van de officiële betrouwbare standaard van ISO 25010. De codering zal specifiek zoeken naar “Metrics”, ofwel de meetwaarde van kwaliteitskenmerken binnen het ISO 25010 model. (ISO/IEC, 2011) Deze worden omgezet naar SMART requirements op dezelfde manier als de functionele requirements. Zowel de functionele en technische requirements zullen worden voorgelegd aan de Hans de Bruin ter validatie.
5. Het **bronnenonderzoek naar de Mechanics voor een Unity applicatie, design patterns en wereld ontwerp mogelijkheid** zal worden geanalyseerd door de bronnen door te nemen en deze samen te vatten. De samenvatting zal verdeeld worden in diverse koppen die beschrijven: Welke modestructuur Unity en de EMont de basis vormt voor de Simulatie van de wereld, welke design patterns

(ontwikkelpatronen) de statisch model verbeteren en hoe er werelden apart ontwikkeld kunnen worden. Zo zijn alle relevante begrippen (zie theoretisch kader 1.4.4) afgevangen.

6. Het **bronnenonderzoek naar Mechanics voor de gefilterde Dynamics** zal worden gedaan door twee kolommen toe te voegen aan elk van de tabellen die zijn opgesteld in bronnenonderzoek naar mogelijke Dynamics (zie analyse 3, bronnenonderzoek naar mogelijke Dynamics). De eerst toegevoegde kolom beschrijft welke technische implementatie er nodig is om deze functionaliteit toe te passen. De tweede toegevoegde kolom beschrijft de functionele eisen waaraan deze voldoet. Zo wordt er duidelijk welke Mechanics van de Dynamics nodig zal zijn binnen het model van de interactieve 3d wereld. Samen met de uitkomsten van analyse 5 zal er een advies klassendiagram worden opgesteld voor een wereldmodel.

2.3 Deelvraag 3

Dv 3 - Hoe kunnen de EMont gegevens worden omgezet naar het statisch model van de interactieve 3d wereld?

Deze deelvraag zal het advies uit deelvraag 2 uitbreiden met een methode om de semantische vastgelegde narratives om te zetten naar het adviesmodel van deelvraag 3.

2.3.1 **Benadering/Strategie**

Het onderzoek kent als randvoorwaarde om de EMont-gegevens te verkrijgen uit de standaard exportermogelijkheden van het projectenportfolio.nl. Er zal met een bronnenonderzoek worden verkregen welke exportermogelijkheden er in de EVM-SMW www.projectenportfolio.nl worden aangeboden en in welke formaten deze aanleveren (Randvoorwaarde 2 in 1.3 Randvoorwaarden). Vervolgens zullen de gewenste exportermethoden en formaten worden uitgekozen door deze te filteren met de functionele en technische eisen. Het bronnenonderzoek zal opzoek gaan naar bestaande code die de exporteerformaten kan lezen en om te zetten in objecten binnen het adviesmodel van deelvraag 2. Validatie van de uitkomst wordt gedaan in deelvraag 4.

Om de kwaliteit van de resultaten te bewaken zullen de resultaten uit de analyses worden voorgelegd aan Hans de Bruin wegens zijn kennis over de EMont en zijn kennis over de wensen en eisen van de interactieve 3d wereld. De resultaten en ook de onderzoeksmethode zullen getoond worden aan de stagebegeleider. Hiermee zal de correctheid en compleetheid van de resultaten worden gegarandeerd.

2.3.2 **Meetinstrument en operationalisatie**

Het bronnenonderzoek zal worden gehouden aan de hand van een zoekplan. Het zoekplan is te vinden in Tabel 5 van Bijlage F. Het plan definieert de zoekmethode, zoektermen en bronnen die zullen worden gebruikt voor het verzamelen van de relevante informatie. De relevante bronnen zullen een vastgelegde referentie krijgen in het adviesrapport. De zoektermen zullen worden gebruikt in de Engelse en Nederlandse schrijfwijze van het begrip.

2.3.3 **Analysemethode**

Aan de hand van het zoekplan zijn er bronreferentie vastgelegd. Deze bronnen worden doorgenomen en verwerkt in een aanvulling op het bestaande adviesrapport. In het adviesrapport zal in het advies over de mechaniek worden verteld wat er zal moeten worden toegepast om een koppeling te leggen tussen het statische model uit deelvraag 2 en de gevonden methode voor het inladen en gebruiken van de EVM-SMW geëxporteerde gegevens. Deze koppeling zal bestaan uit objecten en relaties die worden meegenomen in het statische model van het uiteindelijke advies.

2.4 Deelvraag 4

Dv 4 - Is het mogelijk om aan de hand van de bevindingen een prototype op te stellen dat de EMont gegevens uit een EVM-SMW kan omzetten in een interactieve 3d wereld, zodat narratives kunnen worden verteld?

Dit is de laatste deelvraag van het onderzoek. Er dient te worden gevalideerd of het advies uit deelvraag 2 en 3 toepasbaar zijn.

2.4.1 Benadering/Strategie

In dit onderdeel van het onderzoek zullen de uitkomsten van het adviesdocument worden geplaatst in een functioneel ontwerp (FO) en technisch ontwerp (TO), die de opzet van een prototype interactieve 3d wereld beschrijven. De functionele compleetheid en correctheid zal worden getest met acceptatietesten die zijn beschreven in het functioneel ontwerp. De technische compleetheid en correctheid zal worden getest met unittesten voor ieder van de relevante componenten van het systeem. De ontwerpen zijn op basis van de eisen die al eerder zijn geformuleerd in onderzoeksmethode 4 van Deelvraag 2.

Om de kwaliteit van de resultaten te bewaken zullen de resultaten uit de analyses worden voorgelegd aan Hans de Bruin in verband met zijn kennis van de EMont en zijn kennis van de wensen en eisen van de interactieve 3d wereld. De resultaten en ook de onderzoeksmethode zullen getoond worden aan de stagebegeleider. Hiermee zal de correctheid en compleetheid van de resultaten worden gegarandeerd.

2.4.2 Meetinstrumenten en operationalisatie

Het experiment zal plaats vinden op de HZ. Hier zal er op de laptop van de onderzoeker een aantal fases worden doorlopen voor het ontwikkelen van het prototype. Deze fases zijn uit de Software Development LifeCycle (SDLC): de ontwerp-, de realisatie- en beheerfase. (Systems Development Life-Cycle Policy, 2017). In de ontwerpfase zullen twee ontwerpdocumenten worden opgesteld:

1. Het functionele ontwerp (FO): Een ontwerp met daarin de functionele beschrijving van de interactieve 3d wereld. Dit document bestaat uit de uitleg over de interface elementen en de dynamiek uitgelegd aan de hand van de Use Case Map en een klassendiagram. Deze onderdelen zijn op basis van de relevante begrippen ten aanzien van Dynamics (zie theoretisch kader 1.4.4). Ook bevat het FO een testplan beschrijvende acceptatietesten voor het testen van de functionele compleetheid.
2. Het technische ontwerp (TO): Een ontwerpdocument dat de technische specificaties en de ontwikkeling van het systeem beschrijft. Het TO bevat uitleg over de ontwikkeltools, de codeconventie, de entiteiten en de relaties. Het TO beschrijft deze onderdelen aan de hand van tekstuele uitleg, een sequentiediagram (technische beschrijving systeem interactie) en een klassendiagram.

Het realiseren van het product zal worden gedaan met de volgende softwareproducten:

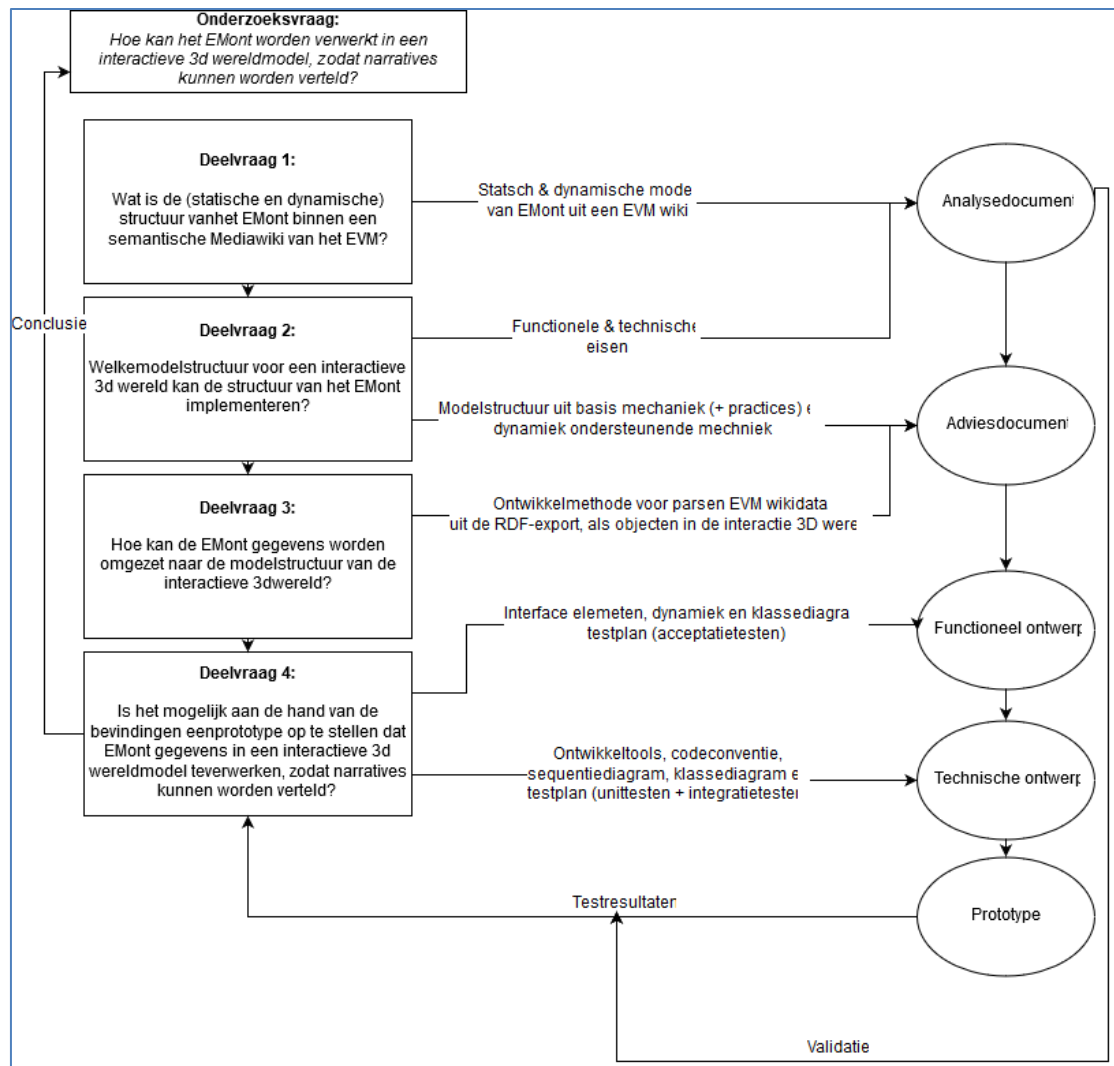
- De Unity – game engine voor het bouwen en testen van de interactieve 3d wereld.
- Microsoft Visual Studio 2017, Integrated development environment (IDE) voor het beheren en testen van de C# code voor de interactieve 3d wereld.
- Bitbucket, het versiebeheersysteem voor het beheren van de product programmering.

2.4.3 Analysemethode

Wanneer het prototype is gerealiseerd volgens de ontwerpkeuzes van het FO en TO zullen de testen worden uitgevoerd. Er zal worden gekeken of alle testresultaten een positieve uitkomst aangeven. Alleen positieve resultaten geven aan dat het ontwerp daadwerkelijk een prototype beschrijft dat de interactieve 3d wereld kan realiseren. Bij negatieve testresultaten zal het ontwerp, realisatie en testproces nogmaals worden doorlopen.

2.5 Totaalbeeld

Het totaalbeeld van het onderzoek is aangegeven in een diagram dat de producten en deelvragen aantoont en hier ook de bijbehorende beroepscompetenties aangeeft.



Figuur 6 Algemeen overzicht van de samenhang van deelvragen, resultaten en producten

3 Resultaten

In dit hoofdstuk worden de resultaten bij de deelvragen in chronologische volgorde besproken.

3.1 Dv1 – de EMont structuur uit interview en SMW-bronnenonderzoek

Wat is de (statische en dynamische) structuur van de EMont binnen een EVM-SMW?

3.1.1 Resultaten

Voor het beantwoorden van de deelvraag is een interview gehouden met Hans de Bruin. Het interview is gebruikt om zowel de dynamische als de statische structuur van de EMont binnen de EVM-SMW www.projectenportfolio.nl te herleiden. In Bijlage H is het interview transcript en de codering terug te vinden. In het interview is ook gevraagd naar alternatieve bronnen om deze structuur uit te herleiden. De alternatieve bronnen zijn individueel onderzocht en vastgelegd in aanvullende bronnenonderzoeken. Tussen deze bronnenonderzoeken naar de statische vastlegging (Bijlage I) en dynamische vastlegging (Bijlage J) is er een kort interview gehouden met Hans de Bruin, om de onduidelijkheden in de bronnen en het interview te verklaren. Dit interview is met vragen en antwoorden beschreven in stap 5 van Bijlage I.

3.1.1.1 *Statische structuur*

Resultaat van het interview

Met behulp van codering zijn uit het eerste interview transcript entiteiten, attributen en relaties gevonden die de EMont beschrijven. De beschrijving komt overeen met wat al van de EMont beschreven is in 1.4.2 van het theoretisch kader.

Dezelfde codering geeft aan dat de EMont wordt vastgelegd met pagina's binnen de SMW. In het interview heeft Hans de Bruin verteld dat iedere pagina uit een formulier bestaat. Het formulier beschrijft welke waardes de pagina's vasthouden. Ook heeft hij verteld dat de formulieren hun waardes krijgen van hun toegewezen sjabloon. Van deze pagina's zouden categoriepagina's en instantiepagina's zijn. Volgens Hans de Bruin zijn de categorieën gelijk aan softwareklassen en zijn de instantiepagina's vergelijkbaar met softwareobjecten (instanties) van de klassen. Deze beschrijving komt overeen met de theorie uit 1.4.4.2 van het theoretisch kader.

Ook uit de codering is een alternatieve bron gevonden. Hans de Bruin heeft verwezen naar de volgende link: <https://www.projectenportfolio.nl/wiki/index.php/Special:Category%C3%ABn>. Deze link zou verwijzen naar een webpagina waarop een overzicht van alle categoriepagina's te vinden is. Hij heeft geadviseerd om de structuur van de pagina's te doorlopen te beginnen bij de "EMont core" categoriepagina.

Resultaat bronnenonderzoek

Het aanvullende bronnenonderzoek naar de vastlegging van de entiteiten en attributen heeft het advies van Hans de Bruin opgevolgd. Door de categoriepagina's van "EMont core" te doorlopen is het mogelijk geweest een schematisch overzicht (Tabel 9 van Bijlage I) te maken van de verschillende klassen (categorie wikipagina's). Het schema toont de gevonden categoriepagina per EMont niveau. De "->" geeft aan op welke pagina (links van de pijl) de pagina rechts van de pijl overerft.

Het bronnenonderzoek heeft de gevonden categoriepagina's uit Tabel 9 geplaatst onder de "EMont elementen (entiteiten) + attributen" van een nieuw schema (Tabel 10 van Bijlage I). Vervolgens is van iedere categoriepagina het bijbehorende formulier en het sjabloon opgezocht. De sjablonen zijn bekeken, omdat deze de opbouw van een formulier beschrijven. De links van de formulieren en sjablonen zijn genoteerd onder de "Bron(nen)" kolom. Binnen de formulieren en sjablonen gevonden attributen zijn genoteerd onder de paginanaam in de eerste kolom. De notatie van de attributen is gedaan met de standaardnotatie van attributen in een klassendiagram. Referenties naar andere objecten pagina's binnen de SMW zijn rood gemarkeerd. Ook deze categoriepagina's zijn meegenomen. Het complete resultaat van de schematisering van de entiteiten en attributen is te vinden in Tabel 10 van Bijlage I.

De resultaten uit Tabel 10 van Bijlage I zijn gevisualiseerd in een eerste versie van een klassendiagram. Dit klassendiagram is te vinden in Figuur 8 van Bijlage I.

In een tweede interview (stap 5 van Bijlage I) is er gevraagd naar de correctheid van de eerste uitwerking van het statische model. Hier heeft Hans de Bruin enkele opmerkingen over gemaakt die zijn aangepast aan het klassendiagram. In Bijlage K bevindt zich het klassendiagram van (Figuur 10) en de opmerkingen. De EMont basiselementen zijn wit met zwart.

3.1.1.2 Dynamische structuur

Resultaat van het interview

Met behulp van codering zijn uit het eerste interview transcript de relatie tussen de narrative elements (uitgelegd in 1.4.3) en de EMont elementen gevonden. Hieruit is gebleken welk narrative element wordt beschreven met welke EMont elementen. Dit is vastgelegd in Tabel 11 (Bijlage J).

Uit de codering naar aanvullende bronnen is gebleken dat Hans de Bruin heeft verwezen naar de resultaten pagina van het "Overleven met Kanker (OmLK)" project, te vinden op het adres:

https://www.projectenportfolio.nl/wiki/index.php/PR_00019_-_resultaten.

Hans de Bruin heeft verteld dat de casussen met hun handelingsperspectieven, herkenbare en bestaande situaties te vinden zijn op deze pagina.

Resultaat bronnenonderzoek

Met behulp van de alternatieve bronnen van Hans de Bruin zijn vier casussen gevonden. Tijdens het verzamelen van de gegevens van de casussen ontstonden onduidelijkheden over de toepassing van de EMont binnen de EVM-SMW. Deze onduidelijkheden zijn besproken in een tweede interview (Stap 5 van Bijlage I). Hieruit zijn de volgende opmerkingen voortgekomen:

- In het statische model van Figuur 10 (Bijlage K) is te zien dat contexten een superContext hebben. Dit kan worden gebruikt om te herleiden binnen welke herkenbare situatie een bestaande situatie valt, maar deze koppeling was niet te vinden in de bestaande casussen. In het tweede interview heeft Hans de Bruin aangegeven dat de herkenbare situaties "includes" gebruiken voor het linken van de contexten. Dit is onderdeel van een onofficiële de EMont structuur.

- De attributen Decompositie types (van een Intentional Element) zijn “logical operations” bestaande uit AND (= Deze en een andere), XOR (= kiezen tussen het een of het ander, niet beide) en IOR (= kiezen tussen het een of het ander, of beide).

Per casus is een schema opgesteld van de bestaande situaties, herkenbare situaties, activiteiten en de decompositietype van de activiteiten. In Bijlage J zijn alle schematiseringen van de casussen weergegeven in Tabel 12, Tabel 13, Tabel 14 en Tabel 15.

De schema's van Claire en Mevrouw Schoonenbrug zijn gevisualiseerd in een Use Case Map, omdat ze beide in dezelfde herkenbare situaties zijn beland (te vinden in Figuur 11 van Bijlage L). De Use Case Map is gevalideerd in een korte meeting met Hans de Bruin. Hans de Bruin had geen fouten gezien in de narratives.

3.1.2 **Analyse**

Uit de resultaten van het interview, het bronnenonderzoek en de validatie interviews is het statische model (Figuur 10 van Bijlage K) en dynamische model (Figuur 11 van Bijlage L) gevormd.

3.1.2.1 **Statische structuur**

In het statische model in Figuur 10 (Bijlage K) is te zien dat een EVM-SMW de EMont statisch implementeert, zoals de EMont basis (L0) is beschreven in Figuur 1 in het theoretisch kader. Dit is te herleiden uit de zwart met witte entiteiten. Hiervan zijn de Intentional elements “Goal”, “Belief”, “Outcome” en “Condition” beschreven met de variabele “elementsType” en zijn de Intentional elements “Actor” en “Activity” aparte entiteiten. De relaties Depends, Contributes, Selects en Connects zijn echter gedefinieerd met een eigen entiteit en attributen, zoals aangegeven door de blauwe markering.

Uit de rode markering is af te leiden dat er diverse entiteiten zijn vastgelegd die overerven op de basis entiteiten van de EMont (L0 entiteiten van de EMont concepten). Deze vormen het L1 en L2 niveau. Van deze entiteiten heeft Hans de Bruin aangegeven dat alleen de dynamiek belangrijk is. Deze entiteiten kennen geen vaste benaming en maken gebruik van attributen die niet worden gebruikt als standaard voor de EMont vastlegging binnen de EVM-SMW.

Binnen de onofficiële structuur worden condities beschreven met “SSM” objecten. Dit in tegenstelling tot het basismodel (L0) van de EMont waarin er wordt gesproken over de entiteit “conditie”. De conditie zou een sub entiteit zijn van een intentional element.

3.1.2.2 **Dynamische structuur**

Uit het dynamische model is te zien hoe het verloop van de vastgelegde narratives is vastgelegd binnen de projectenportfolio EVM-SMW. De actoren Claire en Mevrouw Schoonenbrug belanden in diverse situaties afhankelijk van de keuze in handelingsperspectieven die ze ondernemen. De keuze voor de handelingsperspectieven is bepaald door de condities en decompositie types. Er wordt in 1.4.2.1 van het theoretisch kader verklaard dat practices gebruikt worden om aan te geven welke activiteiten ondernomen worden binnen situaties. Deze zijn niet gevonden in de narratives. De gekozen activiteiten zijn in chronologische terug te vinden in de vrije tekst van de casus. (Figuur 9 van Bijlage J).

3.2 Deelvraag 2 – Adviesstructuur voor interactieve 3d wereld

Dv 2 - Welk statisch model voor een interactieve 3d wereld kan de structuur van de EMont implementeren?

3.2.1 Resultaten

Voor het beantwoorden van deze deelvraag is er een onderzoek gedaan naar de Aesthetics, de gewenste Dynamics en de Mechanics die zijn vereist voor de interactieve 3d wereld. Deze zijn terug te vinden in het adviesrapport (Bijlage Q).

Keuze Aesthetics

Er is gezocht naar de verschillende varianten van Aesthetics die een interactief medium kan overbrengen. Dit is gedaan in een bronnenonderzoek dat ook Dynamics en Mechanics heeft opgezocht. Er zijn acht verschillende Aesthetics gevonden. De acht Aesthetics zijn onderdeel van een set die opgesteld is door Hunicke, LeBlanc en Zubec. De set bestaat uit (Abbott t. , 2010): Sensation, Fantasy, Narrative, Challenge, Fellowship, Discovery, Expression, Submission. (De set is met meer detail uitgelegd in Figuur 12 van Bijlage Q.

Nadat de Aesthetics zijn gevonden is er een interview gehouden met Hans de Bruin. Dit interview bevatte een opsomming van de gevonden Aesthetics, Dynamics en Mechanics voor een interactief medium en volgde het zoekplan dat is gegeven in Tabel 4 van Bijlage E. Als transcript voor het interview is er per aangeleverde slide opgeschreven welke wensen of eisen Hans de Bruin heeft kunnen bedenken met de nieuwe informatie. Uit het transcript (Bijlage O) is te herleiden dat Hans de Bruin bij slide nummer drie heeft gekozen voor het Aesthetics “narrative”, omdat het gaat over het uitleggen van problemen van mensen waar andere mensen zich in moeten kunnen plaatsen.

Keuze Dynamics

Voor het vaststellen van de dynamiek is net als bij de Aesthetics een bronnenonderzoek en een interview gehouden. Het bronnenonderzoek heeft gezocht naar de dynamische mogelijkheden. Er is gezocht op basis van het zoekplan dat is te vinden in Bijlage D. De volgende (functionele ontwerp) begrippen zijn opgezocht en aangevuld in de paragraaf 1.4.4.1 van het theoretisch kader: Graphical User Interface, Barry Ip’s narrative descriptors, Game timestructuur. De bevindingen zijn verwerkt als uitbreiding op het theoretisch kader, omdat in het theoretisch kader hier al over uitgelegd is. Ook heeft het zoekplan bepaald te zoeken naar de Game usability en accessibility design patterns. Deze zijn niet verwerkt in het theoretisch kader, maar samengevat in Bijlage G.

De bevindingen van het bronnenonderzoek naar de varianten van de functionele ontwerp begrippen zijn verwerkt in een presentatie waarvan de hand-out te vinden is in Bijlage O. De presentatie is in een interview voorgelegd aan Hans de Bruin. Dit is het eerdergenoemde interview dat de gevonden Aesthetics, Dynamics en Mechanics aan Hans de Bruin heeft uitgelegd. Uit het transcript (Bijlage H) van het interview zijn de eisen en wensen verzameld en omgezet naar SMART-geformuleerde functionele requirements. De opgestelde requirements zijn te vinden in Bijlage M.

De meest uitgeschreven versie van de SMART functionele requirements zijn gekopieerd in een tabel met twee kolommen. De linkse kolom is voor het tonen van de requirement. De rechter kolom geeft de prioriteit aan. De requirements zijn in een informeel interview voorgelegd aan Hans de Bruin. Hans de Bruin heeft requirements gevalideerd, en vervolgens geprioriteerd met de MoSCoW-methode. (Bruin, Feedback en prioritering van opgestelde eisen, 2017)

Keuze voor Mechanics

Voor het vinden van de juiste Mechanics is er wederom gebruik gemaakt van het zoekplan dat is te vinden in Bijlage E. Eerst is er met gezocht naar “Unity – Game Engine classdiagram” uit de termenlijst van het zoekplan. Aan de hand van de zoekterm is een klassendiagram gevonden van een Unity wereld. Het klassendiagram is te vinden in Figuur 13 van Bijlage Q. Dit vormt de basis voor het uiteindelijke statisch adviesmodel van de interactieve 3d wereld. (Figuur 27 van H4 van Bijlage Q).

Eerst is er een interview gehouden met Hans de Bruin. Dit is hetzelfde interview als het eerdergenoemde interview met de presentatie (Bijlage O). Aan het eind van het interview is het ISO-25010 kwaliteitsmodel erbij gepakt en is er gevraagd naar welke technische eisen Hans de Bruin heeft te stellen voor de gekozen functionele eisen. De wensen en eisen zijn wederom genoteerd per slide. Deze zijn te vinden in Bijlage O. Als analyse op het transcript is elke gevonden eis om gezet in een SMART requirement. Hiervoor is bij iedere eis genoteerd welke metric (meetbare waarde) er te halen is uit de eis of wens van Hans de Bruin. Aan de hand van deze waarde zijn SMART geformuleerde requirements opgesteld. Alle opgestelde technische eisen zijn te vinden in Bijlage N.

De technische requirements zijn in een informele meeting met Hans de Bruin geprioriteerd. De prioritering van de technische requirements is op eenzelfde manier uitgevoerd en vastgelegd als die van de functionele requirements. Vervolgens is er gezocht naar veel gebruikte Unity ontwerppatronen aan de hand van de zoekterm “Unity design patterns” uit het zoekplan van Bijlage E. In paragraaf 2.1 van het adviesrapport (Bijlage Q). Van de gevonden ontwerppatronen zijn bronreferenties genomen met een citaat van de korte beschrijving. De gevonden patronen zijn in eigen woorden samengevat in Tabel 21 van Bijlage Q. Hierbij is er gekeken of deze bijdragen aan niet-functionele eisen.

Het bronnenonderzoek naar Mechanics is verdergegaan met het zoeken naar de technische implementatie van de relevante Dynamics. In het adviesrapport (Bijlage Q) is er voor iedere Game tijdstructuur (Tabel 22, Tabel 23, Tabel 24 genoemd in 3.1), de GUI-elementen (Tabel 25 in 3.2), de narrative descriptors (Tabel 26 in 3.2) en de ontwerppatronen (Tabel 27 in 3.3) een aparte tabel opgesteld. In deze tabellen zijn de verschillende varianten onder elkaar gezet. Per variant is aangegeven welke functionele of technische eis gebruik maakt van de variant van een Dynamic.

De tabellen zijn doorgelopen en aangevuld met een beschrijving voor een technische implementatie van de Dynamic. De technische beschrijving is alleen opgezocht als er een functionele of technische eis is gekoppeld aan de Dynamic variant. Voor de tabel van de narrative descriptors is er in plaats van een technische beschrijving een referentie gelegd naar de GUI-elementen waar die gebruik van maken.

Ten slotte zijn de tabellen doorgelopen aan de hand van beschrijvingen van technische vereisten van de interactieve 3d wereld. Deze zijn in Tabel 28 (in 3.1 van Bijlage Q) verwerkt in een samengevat overzicht, waar bij iedere technische implementatie is gekoppeld aan de relevante eis.

De technische vereisten zijn verwerkt in een klassendiagram (Figuur 27 in H4 van Bijlage Q). Boven het klassendiagram is uitgelegd hoe deze vormt uit de bevindingen.

3.2.2 Analyse

In Figuur 27 (in H4 van Bijlage Q) is een uiteindelijke statisch adviesmodel weergegeven in een klassendiagram. Het model bestaat uit een zestien aantal entiteiten die horen bij een Unity wereld model (aangegeven met blauw). Om binnen de wereld objecten te plaatsten is er de MonoBehaviour entiteit. Dit de basisklasse voor een Unity

component dat kan worden gekoppeld aan een GameObject. Hiervan kunnen de variabelen worden bewerkt in de Unity Editor als deze public zijn. Uit deze entiteit zijn "Manager" entiteiten en View-componenten gevormd, deze bepalen samen de Game, Simulatie en een deel van de Narrative tijdstructuur.

De Manager-entiteiten beheren de staat en de limieten van de tijdstructuren. De Views zijn de weergaven van de GUI-elementen en de narrative descriptors. De GUI-elementen en de narrative descriptors kennen hun interactie door het versturen van events naar een gekoppeld Controller-component. Een event wordt verstuurd wanneer een waarde van de View wordt aangepast of als een (Unity event) methode wordt aangeroepen. Het event wordt opgevangen door de gekoppelde Controller. De Controller handelt het event af door het uitvoeren van de juiste reactie. De View en Controller samen leveren de interactiemogelijkheden die de gebruiker begeleiden door de Game- en Simulatietijdstructuur.

De Models zijn de instanties van de entiteiten uit de EMont data. Deze zijn vergelijkbaar met de pagina's in een EVM-SMW, omdat deze alleen informatie vasthouden. De Controllers maken gebruik van Models voor het vasthouden van de waarden van View objecten. Hierdoor kunnen View objecten worden vervangen of weggehaald bij het laden van een nieuwe wereld. Elk van de MVC-componenten zal een subklasse krijgen voor een semantische of niet-semantische entiteit. Uit een semantische MVC-entiteit vormt een SMW MVC-entiteit die de basis vormt voor MVC-entiteiten van de EMont entiteiten. Zo kan de EMont statisch worden vastgehouden als Models en door middel van een Controller een koppeling krijgen met een dynamisch te laden View object (dat zich in de wereld bevindt). Iedere MVC-entiteit kan worden gemaakt in de Factory-entiteit die is bedoeld voor dat specifieke MVC-component.

Het MVC-ontwikkelpatroon ondersteunt de wijzigbaarheid van het wereldmodel door een scheiding te maken tussen de statische modelgegevens van de EMont uit een Semantische Wiki (Models) en de objecten in de wereld (Views). Hierdoor draagt dit bij aan de technische eis: NFR3 – De uit de applicatie geëxporteerde werelden dienen te kunnen worden aangepast binnen de Unity Editor. Een nadeel van het MVC is dat het de complexiteit van de softwarearchitectuur verhoogt, maar dit heeft geen invloed op de andere technische eisen die zijn gesteld aan de interactieve 3d wereld. Het gebruik van het Factory-ontwerppatroon verhoogt testbaarheid van de individuele MVC-componenten, maar deze voldoet verder niet aan een gestelde kwaliteitseis.

De interactie die een Controller zal afhandelen is afhankelijk van de Statemachine variabelen van een View of Model. De variabele kent een enumeratie van mogelijke staten om zich in te begeven.

3.3 Deelvraag 3 – Omzetting van de EMont gegevens naar de interactieve 3d wereld

Dv 3 - Hoe kunnen de EMont gegevens worden omgezet in het statisch model van de interactieve 3d wereld?

3.3.1 Resultaten

Voor het verkrijgen van een statische uitbreiding op het adviesmodel van deelvraag 3 is er een bronnenonderzoek gehouden. Het bronnenonderzoek is op zoek gegaan naar de verschillende exportermogelijkheden van de EVM-SMW. Vervolgens is er voor het exportformaat (voor semantische data) gezocht naar bestaande libraries die dit kan inlezen en parsen naar softwareobjecten. De verschillende libraries zijn tegen elkaar afgewogen. De officiële online documentatie van de beste library is opgezocht en doorgenomen op zoek naar de statische structuur. Vervolgens is de statische structuur is toegevoegd aan het klassendiagram van deelvraag 2.

Wiki exporteerformaten

De gevonden exportmogelijkheden binnen de wiki zijn opgesomd in de Tabel 29 in H5.1.1 van het adviesrapport (Bijlage Q).

Bestaande code mogelijkheden voor het inlezen en parsen van het gekozen formaat

Aan de hand van het zoekplan in Tabel 5 van Bijlage F is er gezocht naar de term “RDF” (het relevante semantische exportformaat) en “Mapping”. Binnen de bron <https://www.nuget.org/> zijn de libraries gevonden die RDF-formaten kunnen parsen, beheren, queryen en schrijven. Deze zijn genoteerd in een schema binnen het adviesrapport (Tabel 30 in H5.2 van Bijlage Q).

Statische structuur van RDF parsing naar softwareobjecten met de gekozen library

Uit de gevonden bestaande code is de DotNetRDF library het meest geschikt gebleken (Tabel 30 in H5.2 van Bijlage Q). Het bronnenonderzoek heeft op de documentatiepagina uitleg gevonden over de statische structuur van RDF-data. De uitleg is omgezet in een klassendiagram (Figuur 28 in H5.3 van Bijlage Q).

De statische structuur van dotNetRDF (Figuur 28 in H5.3 van Bijlage Q) is gekoppeld aan de Models van het klassendiagram van deelvraag 2, door middel van een eigen implementatie. Een “SMWParser” entiteit die “Nodes” uit een Graph lezen om de categoriepagina’s (de klassen) als “SMWCategory” te kunnen mappen. Vanuit de SMWCategories kunnen de instantiepagina’s (de EMont instanties) worden gebouwd. In Figuur 29 (in H5.4 van Bijlage Q) is de koppeling te zien.

3.3.2 Analyses

Uit de resultaten is te zien dat de SMW’s van het EVM de pagina gegevens kunnen teruggeven in een XML of RDF-formaat. Enkel het RDF-formaat kan de semantiek van de pagina’s leveren, daarom is deze relevant gevonden. Door binnen de site van de NuGet (C# packetmanager) te zoeken naar RDF lezende en parsende libraries of frameworks zijn er vijf relevante libraries gevonden. Door het naast elkaar leggen van eigenschappen van de libraries is de conclusie getrokken dat dotNetRDF het meest geschikt is.

De dotNetRDF library is bekeken op implementatie en vervolgens visueel gemaakt in een statisch model. Het statisch model toont dat het RDF-bestand wordt ingelezen in een Graph. Deze Graph bevat een set van Nodes waarvan de relaties zijn aangegeven met Triples. Nodes zijn hier de objecten uit het RDF. Deze bevatten de Categorieën en instanties van de EMont.

De Graphs bevatten semantische objecten binnen het RDF als Node objecten. De relatie tussen de Nodes zijn binnen de Graphs aangegeven met Triple objecten. Deze drie Nodes van de Triple zijn een onderwerp object (subject node), een relatie (predicate node) en een object waarnaar de relatie wijst (object node). De Triples geven de variabelen van een subject node aan. Dit door het benoemen van de variabeltype met een predicate node en het benoemen van de variabelwaarde met een object node. Om deze Graphs met Nodes en Triples aan te maken gebruikt dotNetRDF een FileLoader en een RdfXmlParser object.

Er zijn diverse mogelijkheden gevonden voor het mappen van Nodes (uit een semantische Graph) als softwareobjecten, maar deze zullen beter bekeken moeten worden voordat deze worden toegepast. In plaats van deze methodes is er simpele koppeling gelegd door het meegeven van de node aan het bijbehorende softwareobject Figuur 29 (in H5.4 van Bijlage Q). De eigen gemaakte koppeling bestaat uit een SMWParser die

Graphs vasthoudt en hieruit de Nodes voor de SMW-categorieënpagina's uitleest en omzet in de SMWCategory objecten. De SMWCategory lezen de Nodes uit de Graph van hun Node, om zo alle SMW-instantiepagina nodes van de SMW-categorie te vinden. Voor deze SMW-instantiepagina nodes maakt de SMWCategory een instantie van een subklasse van SMWObject. Deze SMWObject kent zijn Node en kan via de Node zijn Graph bevragen om zijn Triples. Iedere subklasse van de Node dient zijn eigen methode te beschrijven voor het ophalen van de waardes. Ook bevatten de Graphs de relatie objecten als Depends, Connects en Contributes. Deze zullen worden opgehaald methode van de SMWParser.

3.4 Deelvraag 4 – Valideren advies bevindingen met prototype

Dv 4 – Is het mogelijk om aan de hand van de bevindingen een prototype op te stellen dat de EMont gegevens uit een EVM-SMW kan omzetten in een interactieve 3d wereld, zodat narratives kunnen worden verteld?

3.4.1 Resultaten

Voor het valideren van het statische adviesmodel van deelvraag 2 en 3 is er een prototype ontwikkeld dat het model uit deelvraag 2 en 3 implementeert. Het prototype is ontworpen aan de hand van een functioneel ontwerp (Bijlage R) en technisch ontwerp (Bijlage S). Het gerealiseerde product is getest aan de hand van acceptatietesten (acceptatietestplannen zijn genoemd in H6 van Bijlage R), Unittesten en integratietesten (voor het testen van de techniek achter de omzetting naar softwareobjecten). De testen zijn gefocust op het aantonen van de "Must" requirements.

Acceptatie testen

De interactieve 3d wereld is gestart binnen de Unity Editor en doorlopen, zoals de acceptatietesten deze hebben beschreven. De stappen zijn vastgelegd met een screenshot en beschrijving van de situatie. De vastlegging is te vinden in Bijlage U. In Tabel 31 (in 2 van Bijlage U) is een opsomming gegeven van de testresultaten.

Unittesten testen

Op basis van het technische ontwerp zijn er unittesten geschreven. De testen zijn uitgevoerd in de Unity Editor. De uitslagen van de testen zijn te vinden in Bijlage U.

3.4.2 Analyses

De acceptatietesten en de unittesten geven aan dat het product voldoet aan de "must"-eisen. De uitgevoerde acceptatietesten hebben de use case "UCM2 - Dynamische weergave van een simpele testcase narrative" (te vinden in Bijlage R) succesvol getest. Deze use case bevat alle wireframes en voldoet aan de eisen: FR1, FR2.1, FR5.2, FR6 en FR11 (Bijlage P). De unittesten hebben alle "Manager" objecten, de RDF-omzetting naar Models, de omzetting van Models naar Views en het bestaan van de RelationControllers (condities), ProjectMenu en het NarrativeMenu getest. Dit met een succesvol doorlopen automatisch uitgevoerde UCM2 use case. Het functionele en technische ontwerp lijken, dus werkend te zijn toegepast.

4 Discussie

In dit hoofdstuk zal per deelvraag in chronologische volgorde worden doorlopen wat er te zeggen valt over de kwaliteit en compleetheid van de onderzoeksresultaten en wat dit betekent voor de andere deelvragen.

4.1 Deelvraag 1 – Analyse op het eerste interview en een SMW-wiki

Dv 1 - Wat is de (statische en dynamische) structuur van de EMont binnen een EVM-SMW?

4.1.1 Discussie

Het gewenste resultaat van deze deelvraag is een beschrijving van de structuur van de EMont zoals deze nu vastligt in de EVM-SMW's. Dit resultaat is een statisch en dynamisch model, waarvan de entiteiten uit het dynamisch model zijn terug te vinden in het statisch model.

Om dit resultaat te bereiken is er een ongestructureerd interview gehouden met Hans de Bruin. Uit de codering van het eerste interviewtranscript is het mogelijk geweest een eerste indruk te krijgen van de EMont, zoals deze vast zou liggen binnen EVM-SMW's (uitgelegd in 1.4.2). Uit de codering is het ook mogelijk geweest om de koppeling tussen de narrative elements en de EMont elementen en ondersteunende informatiebronnen te verkrijgen voor het vinden van vastgelegde casussen en de LO EMont entiteiten (zie Tabel 11 van Bijlage J).

In een bronnenonderzoek naar de statische structuur zijn de categoriepagina's doorlopen vanaf de categorie "EMont core" uit de door Hans de Bruin geadviseerde bron met het overzicht van alle categorieën. De gevonden pagina's zijn omgezet in een overzicht dat het EMont niveau toont per EMont niveau binnen de Projectenportfolio EVM-SMW (zie Tabel 9). Deze eerste weergave heeft gevalideerd dat alle EMont niveau binnen de bron te vinden zijn. Vervolgens is elk formulier van deze pagina's geanalyseerd naar de attributen. Door het analyseren van de attributen binnen de SMW is het mogelijk geweest om nog niet bekende relevante pagina's te vinden. Alle gevonden pagina's en attributen zijn verwerkt in een overzichtelijk schema (zie stap 3 van Bijlage I). Dit schema is verwerkt in een klassendiagram. Het klassendiagram is op syntax gecontroleerd door ICT-docent Anton Bil. Ter validatie van de compleetheid en correctheid van de entiteiten, attributen en relaties is het klassendiagram voorgelegd aan Hans de Bruin in een semigestructureerd interview.

In het bronnenonderzoek naar de dynamiek is er gebruik gemaakt van de "Overleven met Kanker (OmLK)" projectenpagina, dit op aanraden van Hans de Bruin. Uit de codering van het interview is beschreven welke elementen een narrative vormen. Aan de hand van de bevindingen uit het eerste concept statische model (Figuur 8) is er gezocht binnen het Projectenportfolio EVM-SMW. Hierdoor zijn vaagheden ontstaan die zijn verklaard in een interview met Hans de Bruin. Deze bevindingen zijn meegenomen bij het vastleggen van de narratives in schema's. Twee schema's zijn gevisualiseerd in een Use Case Map en voorgelegd aan Hans de Bruin ter validatie.

Doordat de gegevens zijn verzameld binnen geadviseerde bronnen en binnen een actieve EVM-SMW zijn de uitkomsten relevant en betrouwbaar. Doordat de tussen- en eindresultaten van het de bronnenonderzoeken zijn besproken met experts is er hoge garantie van de kwaliteit en de compleetheid.

4.1.2 Deelconclusie

Op basis van de uitgevoerde analyses op de interviews en de bronnenonderzoeken is het mogelijk geweest om de structuur van de EMont uit te drukken in een klassendiagram (statisch structuur te zien in Figuur 10 van Bijlage K) en een Use Case Map (dynamisch structuur te zien in Figuur 11 van Bijlage L).

De statische structuur is op te delen in de L0, L1 en L2 niveaus, zoals deze zijn genoemd in 1.4.2.1 van het theoretisch kader. Het L0 is opgesteld met de entiteiten: IntentionalElement en Context. Een IntentionalElement is een superklasse van Actor en Activity en kent de andere intentional elements als een variabel met de enumeratie "Goal", "Belief", "Outcome" en "Condition". Dit is anders dan het L0 statische model uit Figuur 1 getoond in 1.4.2 van het theoretisch kader, want deze toont alle intentional elements als een eigen entiteit.

Uit de opgestelde statische structuur zijn het L1 en L2 van de EMont uitgedrukt met diverse gevonden entiteiten: HerkenbareSituaties, Proces, Case, SSMSituations, Handelingsperspectief, Tip, ContactPersoon, Organisatie en SSMWorldView. Deze zijn opgesteld in het klassendiagram als antwoord op deelvraag 1, maar zullen niet meegenomen worden in het interactieve 3d wereldmodel van deelvraag 2. Hans de Bruin heeft gezegd dat van die entiteiten enkel de dynamiek belangrijk is. Ook opvallend is dat de condities zijn vastgelegd in SSMWorldViews in plaats van de "Condition" entiteit en dat de link tussen de bestaande situaties en herkenbare situaties is aangegeven met een "include" binnen de vrije tekst van een SMW-pagina. Dit zijn twee onofficiële gebruikswijzen van de EMont. Volgens de opdrachtgever dient de interactieve 3d wereld alleengebruik te maken van de officiële EMont gebruikswijzen. Hierdoor zal het niet mogelijk zijn om de huidige narratives met de juiste dynamiek aan te leveren binnen de 3d wereld. Er zal voor deelvraag 4 eigen testdata moeten worden gegeven die wel gebruik maakt van de officiële EMont structuur.

Uit de opgestelde dynamische structuur is te herleiden dat Mevrouw Schoonenbrug en Claire handelingsperspectieven doorlopen, om zo in nieuwe situaties te belanden. Bij het opstellen van een modelstructuur voor de interactieve 3d wereld zal er rekening moeten worden gehouden met het afhandelen van de gebruikers interactie, om zo keuzes te maken tussen handelingsperspectieven.

De narratives zijn beschreven met subklassen van de basis EMont elementen (L0) zoals deze zijn beschreven in het opgestelde klassendiagram, deze zijn: Context (de subklasse voor de casussen, herkenbare situaties, bestaande situaties) en Activity (de subklasse voor de handelingsperspectieven). Er is geen gebruik gemaakt van de Practice entiteit uit het klassendiagram. De chronologische uitvoer van de activiteiten binnen een casus zijn vastgelegd in een tabel in de vrije tekst van een EVM-pagina. Hierdoor zal het niet mogelijk zijn om een vooraf vastgelegd verhaal semantisch op te halen en uit te lezen.

4.2 Deelvraag 2 – Adviesstructuur voor interactieve 3d wereld

Dv 2 - Welk statisch model voor een interactieve 3d wereld kan de structuur van de EMont implementeren?

4.2.1 Discussie

Het gewenste resultaat van deze deelvraag is een klassendiagram. Het diagram beschrijft de statische structuur van de EMont entiteiten (te zien in Figuur 10 van Bijlage K) als interactieve objecten van de interactieve 3d wereld, zodat de dynamiek van de narratives (de Use Case Map in Figuur 11 van Bijlage L) kan worden doorlopen.

Om dit resultaat te bereiken is er een bronnenonderzoek gedaan naar de mogelijkheden van de Aesthetics (over te brengen gevoel), Dynamics (dynamiek) en Mechanics (techniek). Het bronnenonderzoek heeft acht Aesthetics opgeleverd, een uitbreiding op 1.4.4.1 van het theoretisch kader, een verzameling van gebruiksvriendelijkheid en toegankelijkheid toepassingen, een beschrijving van Unity's basis statische structuur met toe te passen technische Unity ontwerppatronen en de mogelijkheid voor een ontwerper om de wereld aan te passen. Door het raadplegen van de expertise van professionals is er een zekerheid over een stabiele en logische dynamische en technische basis voor de interactieve 3d wereld.

Deze bevindingen zijn voorgelegd aan Hans de Bruin. Hierdoor heeft Hans de Bruin de mogelijkheid gekregen om zijn functionele en technische eisen te kunnen bedenken en stellen. Deze eisen en wensen zijn omgezet naar SMART-geformuleerde functionele en technische requirements. De SMART-requirements zijn gevalideerd bij Hans de Bruin.

De requirements zijn gebruikt voor het filteren van de relevante dynamische ontwerpkeuzes. Vervolgens is er gezocht naar de technische uitwerking van alleen het filtraat van de dynamische ontwerpkeuzes. Zo is er op een efficiënte manier gezocht naar nog vereiste technische vastlegging. De gevonden ontwerppatronen en technische vastlegging zijn samen verwerkt in een klassendiagram dat een statisch model definieert voor de interactieve 3d wereld.

Het ontwerpen van het statisch adviesmodel aan de hand van het MDA-raamwerk heeft ervoor gezorgd dat er een sterke onderbouwing is toegepast voor het ontwerp. Door de bevindingen af te stemmen is er een hoge garantie dat het model voldoet aan de eisen van Hans de Bruin.

4.2.2 **Deelconclusie**

Op basis van het de uitgevoerde analyses op de interviews en de bronnenonderzoeken is het mogelijk geweest om de structuur van de interactieve 3d wereld op te stellen waar de EMont in kan worden gevisualiseerd als interactieve 3d objecten. Het model (Figuur 27 in H4 van Bijlage Q) bestaat uit de basisstructuur van Unity die is uitgebreid met "Manager" objecten voor het beheren van de mogelijkheden voor de gebruiker en maakt gebruik van MVC voor het leggen van de koppeling tussen de 3d wereld en de statische vastlegging van de EMont. Ieder MVC-component kan worden gemaakt met een Factory object. De toepassing van het MVC-ontwikkelpatroon ondersteunt de wijzigbaarheid van het wereldmodel door een scheiding te maken tussen de statische modelgegevens van de EMont uit een Semantische Wiki (Models) en de objecten in de wereld (Views). Het gebruik van het Factory-ontwerppatroon verhoogt testbaarheid van de individuele MVC-componenten.

De interactie binnen de wereld kan worden beschreven met de Unity standaard componenten en eventuele eigen geschreven MonoBehaviour subklassen voor specifieke methode van menu's, interacteerbare objecten buiten om het adviesmodel en de EMont. Hiervoor bestaat een vrijheid in implementatie binnen het product. Een uitgebreid gebruiksvriendelijkheidsonderzoek dient de vereisten GUI-elementen specifiek uit te zoeken.

4.3 **Deelvraag 3 – Adviesstructuur voor interactieve 3d wereld**

Dv 3 - Hoe kunnen de EMont gegevens worden omgezet in het statisch model van de interactieve 3d wereld?

4.3.1 Discussie

Het gewenste resultaat van deze deelvraag is een beschrijving van een statische structuur als uitbreiding op het klassendiagram van deelvraag 3. De uitbreiding zou het mogelijk moeten maken om de EMont elementen uit het geëxporteerde bestand van een EVM-SMW om te zetten in softwareobjecten binnen de interactieve 3d wereld.

Om dit te verkrijgen is er een bronnenonderzoek gedaan naar de exporteer mogelijkheden van een EVM-SMW. Ieder van de mogelijkheden is gevonden op de officiële online documentatie van de MediaWiki en Semantische MediaWiki plug-in. Hierdoor zijn de resultaten erg betrouwbaar. Per gevonden exporteer mogelijkheid is genoteerd in welk format de data aangeleverd is. Het enige aangeboden schematische format was RDF, daarom is voor enkel dit formaat opgezocht welke bestaande code deze kan omzetten in softwareobjecten. De bestaande code is gevonden op NuGet, omdat dit een package manager is voor C# libraries en frameworks. Hans de Bruin wilde namelijk het liefst in dezelfde taal als Unity werken (C# of Javascript). Op de NuGet site is bij elk resultaat van een library of framework actuele informatie aangegeven. Dit maakt het ook een zeer betrouwbare bron. De gevonden libraries zijn de meest actuele en toegepaste libraries uit de resultaten van het NuGet. Deze zijn gefiltreerd met de filter mogelijkheden van de NuGet site, daarom zijn deze ook erg betrouwbaar. Het filteren van de gevonden libraries naar één is gedaan door het kijken naar de meest gunstige eigenschappen van iedere library afzonderlijk. Dit is betrouwbaar, maar kan beter worden uitgevoerd met een uitgebreide pakketselectie.

De technische toepassing van de gevonden library “dotNetRDF” is herleid uit de officiële documentatie. Dit maakt het ook erg betrouwbaar. De koppeling tussen deze technische toepassing en het adviesmodel uit deelvraag 3 kon worden gemaakt met mapping frameworks die gebruik maakten van de dotNetRDF library. Deze zijn nu niet gebruikt, omdat dit een aparte pakketselectie vereist. Hierdoor is er een eigen ontworpen koppeling gemaakt. Dit maakt de omzetting van de dotNetRDF wel mogelijk, maar mogelijk niet volgens effectievere en efficiëntere standaarden voor het mappen van semantische objecten. Om te valideren dat deze technische implementatie van het adviesmodel toepasbaar is zal dit in het prototype van deelvraag 4 aangetoond worden.

4.3.2 Deelconclusie

Op basis van bronnenonderzoeken is het mogelijk geweest om een technisch adviesontwerp te vinden voor een omzetting van de EMont-elementen uit de Semantische MediaWiki naar (3d) softwareobjecten binnen het advies van deelvraag 3. Het ontwerp (Figuur 29 in H5.4 van Bijlage Q) hiervoor bestaat uit een technische implementatie van de dotNetRDF library en een eigen ontworpen koppeling tussen het adviesmodel uit deelvraag 3 en dotNetRDF (Figuur 29 in H5.4 van Bijlage Q). De technische implementatie van dotNetRDF maakt gebruik van Graph objecten representerende de RDF-data. Door een te kort aan informatie over de mapping frameworks is er een eigen koppeling ontworpen om de Nodes uit de Graphs om te zetten in softwareobjecten voor SMW-categorieënpagina's (gerepresenteerd als SMWCategory objecten in Figuur 29), SMW-instantie Figuur 29 (gerepresenteerd als subklassen van SMWObjects in H5.4 van Bijlage Q) en de relaties: Depends, Connects en Contributes. In het prototype van deelvraag 4 zal aangetoond moeten worden of hiermee inderdaad de informatie uit een RDF in (3d) softwareobjecten kan worden omgezet. Er is wel aangeraden om te kijken naar een toepassing van de mapping frameworks voor het leggen van een mogelijk effectievere en efficiëntere omzetting van de RDF-export.

4.4 Deelvraag 4 – Valideren advies bevindingen met prototype

Dv 4 – Is het mogelijk om aan de hand van de bevindingen een prototype op te stellen dat de EMont gegevens uit een EVM-SMW kan omzetten in een interactieve 3d wereld, zodat narratives kunnen worden verteld?

4.4.1 **Discussie**

Deze deelvraag heeft als doel het valideren van de geadviseerde modelstructuren (uit deelvraag 2 en 3) in een prototype van een interactieve 3d wereld voor het vertellen van de verhalen. De ontwerpen bestaan uit een functioneel en technisch ontwerp die zijn opgesteld op basis van de SMART-geformuleerde eisen van de opdrachtgever Hans de Bruin en de adviesmodellen.

Het functionele ontwerp (FO, Bijlage R) beschrijft het doel van de applicatie en welke functionaliteit hieruit volgt. Het FO beschrijft welke EMont elementen met welke relaties te vinden zijn in de wereld in een klassendiagram. Deze komen uit de analyses op de resultaten van de eerste deelvraag. Uit de tweede deelvraag zijn functionele eisen gekomen. Deze eisen zijn opgenomen in het functionele ontwerp. Uit de eisen is te herleiden welke GUI-elementen er binnen de applicatie te vinden zullen zijn en welke situaties een gebruiker kan doorlopen (use cases). De GUI-elementen zijn aangegeven in wireframes bestaande uit screenshots van binnen Unity ontworpen objecten zonder verdere interactie of werkende code. Voor de use cases is er gebruik gemaakt van Use Case Maps van bestaande narratives binnen een EVM-SMW. Twee verschillende use cases zijn vastgelegd: de narratives van Claire en mevrouw Schoonenbrug, en een use case met daarin twee testbare narratives van twee verschillende rollen. De testbare narratives maken correct gebruik van de EMont, omdat de andere narratives geen geheel officiële vastlegging van de EMont gebruiken (gebleken uit de analyse van deelvraag 1). In het functioneel ontwerp zijn acceptatietesten beschreven voor het testen van de belangrijkste functionele eisen (de must-requirements). De acceptatietesten doorlopen de use case van de testbare narratives. Het functioneel ontwerp is op correctheid gevalideerd bij Hans de Bruin en Anton Bil. Hierdoor is het behalen van de acceptatietesten erg betrouwbaar voor het aangeven van een correcte werking van de interactieve 3d wereld.

Het technische ontwerp (TO, Bijlage S) heeft de achterliggende technische implementatie beschreven met een uitbreiding op het adviesmodel dat is opgesteld in deelvraag 2. De uitbreiding bestaat uit een toevoeging van specifieke objectklassen voor de GUI-elementen uit de wireframes en een klasse voor de interactie van de camera met de wereld. Het technische ontwerp is op basis van het functioneel ontwerp gemaakt en is op correcte syntax beoordeeld door Anton Bil. Hierdoor is het technische ontwerp van voldoende kwaliteit. Bij het technische ontwerp horen unittesten die het gerealiseerde product testen op de technische compleetheid. Deze testen of de verwachte waardes zijn gevonden tijdens het doorlopen van een narrative. De unittesten zijn opgesteld in Unity en uitgevoerd met de Unity testrunner.

De validatie van het prototype is gedaan met acceptatie- en unittesten en een demoesessie met de opdrachtgever. Via deze methodes is er met zekerheid te zeggen welke onderdelen van het prototype werken zoals deze zijn bedoeld. Dit geeft aan dat het prototype en de testresultaten betrouwbaar en compleet zijn.

4.4.2 **Deelconclusie**

Het doorlopen van de acceptatietesten en de unittesten heeft geresulteerd in het verkrijgen van positieve resultaten bij iedere test. Dit betekent dat het adviesmodel uit deelvraag 2 en 3 toepasbaar is voor een interactieve 3d wereld. (De testresultaten zijn te vinden in Bijlage U.)

Zoals beschreven in de conclusie van deelvraag 3 is er een vrijheid geboden voor het toepassen van de GUI-elementen en het soort interactie. Binnen het prototype is interacteren met de EMont Activiteiten en Menu's in de wereld gedaan met een RayCast van de FpsHudController van de te bestuderen Rol of Actor. Bij het interacteren met een activiteit wordt er een ObjectInteractionMenu getoond. Dit ObjectInteractionMenu kan net als het ProjectMenu en het NarrativeMenu alleen worden bediend met de RayCast van de computercursor.

5 Conclusie en aanbeveling

“Hoe kan de EMont worden verwerkt in een interactief 3d wereldmodel, zodat narratives kunnen worden verteld?”

Het is mogelijk om een interactieve 3d wereld, binnen Unity (Randvoorwaarde 5), te leveren voor de narrative uit een export van een Semantische MediaWiki van het EVM. Deze wereld werkt alleen voor narratives die zijn beschreven met de EMont concepten en bijbehorende attributen. Dit is aangetoond met een getest prototype. Het prototype is gemaakt op basis van het functioneel (Bijlage R) en technisch ontwerp (Bijlage S) die zijn te vinden op de EVM-Bitbucket (Git-versiebeheer): <https://bitbucket.org/expertisemanagement/narratieven-3d>.

De 3d wereld is een Unity scene gevuld met Game Objecten. De GameObjecten hebben toegewezen componenten (genaamd MonoBehaviours) voor het beschrijven van de eigenschappen. De combinatie van een GameObject met componenten beschrijven een object. Deze objecten worden gebruikt voor het vormen en regelen van de dynamiek binnen de wereld.

De vorming van de wereld gebeurt met objecten die bestaan uit een samenwerking tussen verschillende GameObjecten met Singleton “Manager”-componenten (subklassen van klasse “Manager” te zien Figuur 30 en Figuur 31 van Bijlage T). De Manager objecten werken met een MVC-implementatie van semantische en niet-semantische objecten, waarvan de View een component (subklasse van een Unity.MonoBehaviour is). De niet-semantische objecten bestaan uit instanties van klassen voor het beschrijven van een object (Object), een object dat kan worden bestuurd (ComplexObject), een camera (Camera) en een object dat alleen de besturing beschrijft van een bestuurbaar object (ObjectControl). De semantische objecten zijn instanties van de klasse SemanticObject (subklassen van een ComplexObject). Een SemanticObject kent een referentie naar een Node uit een Graph van de dotNetRDF. Een SemanticObject-type kan zijn attributen herleiden uit de geïndexeerde relaties tussen twee objecten (Triples) van een Graph.

De Graphs definiëren de categorieënpagina’s en instantiepagina’s uit een RDF-export van EVM-SMW als Nodes binnen de Graphs. Om de Managers te laten werken met de Graphs is er een SMWParser gemaakt voor het ophalen van de EMont concepten en de EMont relaties (Depends, Connects, Selects en Contributies te zien in Figuur 10). De SMWParser creëert voor iedere categoriepagina een SMWCategories. De SMWCategories gebruiken een ModelFactory voor het maken van de Models van de instantiepagina’s. De Models van instantiepagina’s zijn instanties van een subklasse van een SMWObject (subklasse van een SemanticObject). De SMWObject heeft een subklasse voor ieder gebruikt EMont concept binnen een EVM-SMW. Deze bestaat uit: een Context, Intentional Element, Activity en Actor. De Intentional Elements objecten hebben een attribuut “IntentionalElementType” om aan te geven of een Intentional Element een Activity, Actor, Goal, Outcome, Condition of een Belief is.

De WorldManager kan de wereld steeds aanpassen zonder de achterliggende EMont definitie van een EVM-SMW aan te passen. Dit is door de toepassing van de Models uit het MVC-ontwerppatroon. Het MVC-ontwerppatroon maakt het mogelijk voor een ontwerper om werelden in verschillende representaties op te slaan. Het ontwerp hoeft hiervoor de objecten (Views) in wereld op te halen en hiervan de waardes aan te passen. De verschillen de representatie kunnen worden ontwikkeld voor de Living Labs waarvoor deze ingezet zullen worden.

De dynamiek van de wereld bestaat uit het veranderen van de omgeving door interactie van de gebruiker. De interactie van de gebruiker wordt gedaan door het besturen van een wereldobject met de camera en het aangeven dat de gebruiker het bestuurde object wil laten interacteren met een ander object binnen de wereld. Het besturen van een object kan de gebruiker doen door het meegeven van input aan de Unity.InputManager. Een subklasse

van de InteractionManager kent het bestuurde object, leest de input van de gebruiker uit de InputManager, leest welke interactie mogelijk is en past de locatie en draaiing van het object aan. Overige interactie van de gebruiker met de andere objecten in de wereld heeft geen standaard technische vastlegging, maar kan worden toegepast op veel verschillende manieren. Het prototype heeft voor de interactie met GUI-elementen een specifiek component geschreven. Voor de interactie tussen de objecten is er een FpsHudController geschreven voor het detecteren van objecten waar de camera van het bestuurde object naar kijkt. Wanneer dit component een object detecteert en een gebruiker hiervoor de juiste InputManager.Input meegeeft kan deze een object laten verschijnen met een ObjectInteractionMenu. Het verschenen object is een menu met interactiemogelijkheden die de gebruiker kan kiezen. Wanneer de gebruiker deze interactie toepast bij een Activity dan kan hij deze uitvoeren met de "Interact" optie uit het menu. De uitgevoerde Activity wordt bijgehouden in een NarrativeManager die de progressie en protagonist van het verhaal beheert. De WorldManager (manager voor het beheren van de objecten in de wereld) past de vertoning van de objecten aan op basis van de informatie van de NarrativeManager. Wanneer een activiteit wordt uitgevoerd zal worden gekeken of de gebruiker in een andere Rol (een Context type) of situatie terecht komt waar de activiteit zich in bevindt. De WorldManager past wereld hierop aan.

Wanneer de WorldManager Intentional Elements laadt in de wereld creëert deze ook GameObject met RelationController componenten. Deze objecten kennen een relatie (Depends, Connects, Contributes) van een Intentional Element en maken deze visueel. De visualisatie wordt binnen het prototype gedaan met een getrokken lijn en tekst.

5.1 **Vergelijking met ander onderzoek/theorie**

In het theoretisch kader zijn diverse onderzoeken besproken. In het "An Ontology about Expertise Management" van H. de Bruin (Bruin, An Ontology about Expertise Management, 2016) is verklaard dat ervaringen van mensen kunnen worden vastgelegd met behulp van de EMont concepten. Hiervoor is er in het onderzoek een mogelijke technische implementatie beschreven die is weergegeven in Figuur 1 van het theoretisch kader. Al deze elementen zijn toegepast binnen het model van deze wereld. Het is in dit onderzoek aangetoond dat het mogelijk is met instanties van deze concepten door loopbare situaties aan te bieden binnen een 3d representatie van de wereld.

In 1.4.3 van het theoretisch kader zijn diverse onderzoeken samengevat voor het beschrijven van de structuur van een "narrative". In het prototype is aangetoond dat een Context (EMont concept) een situatie als protagonist kan beschrijven. De Intentional Elements en Contexten binnen een context vormen samen de setting. Het plot van het narrative wordt bepaald door de gekozen activiteiten van de gebruiker.

5.2 **Aanbeveling voor vervolgonderzoeken**

5.2.1 **Standaard voor een L2 van de EMont en de EMont toepassing binnen de huidige EVM-SMW**

De EMont beschrijft drie verschillende niveaus waarvan het eerste niveau (L0) alleen concepten zijn, het tweede niveau (L1) specifieke instanties zijn van de concepten voor het aangeven van bekende situaties en het derde niveau (L2) de ervaringen of waargebeurde verhalen zijn. Nu is in deelvraag 1 geconcludeerd dat er onofficiële (niet de EMont standaard) attributen zijn toegepast binnen de L2 instantie voor het aangeven van een chronologisch verloop van de meegemaakte situaties en het koppelen van meegemaakte situaties aan bestaande situaties. In het prototype is aangetoond dat het mogelijk is om situaties te kiezen en activiteiten uit te voeren, maar er is geen methode om ondernomen activiteiten van een gebeurd verhaal automatisch te doorlopen op basis van de EMont instanties uit de EVM-SMW.

5.2.2 **Aanvullend gebruikersinteractie onderzoek en validatie van het Aesthetic “narrative”**

Het onderzoek heeft geconcludeerd dat de GUI-elementen (waaronder de wereldobjecten) en interactie met deze GUI-elementen worden geleverd met GameObjecten en toegewezen componenten. In het theoretisch kader is aangegeven dat de vertelling van het plot wordt gedaan met narrative descriptors die bestaan uit een specifieke toepassing van GUI-elementen. Het model van de interactieve 3d wereld van het prototype is opgesteld op basis van functionele eisen die Hans de Bruin heeft bedacht na het hebben geleerd over de narrative descriptors en GUI-elementen. Voor het vinden van een meer toepasselijke dynamiek zal de gebruikersinteractie moeten worden geëvalueerd in een gebruikersvriendelijkheidsonderzoek. Dit onderzoek dient de functionele toepassing te verbeteren ten aanzien van de gekozen Aesthetic “narrative” uit het (MDA-framework).

5.2.3 **Toepassen van een semantische mapping framework (gebouwd op dotNetRDF)**

In deelvraag 3 van het onderzoek is aangegeven dat de EVM-SMW's de categoriepagina's (softwareklasse) en instantiepagina's (software instanties) kunnen exporteren in een RDF-formaat. Dit kan worden ingelezen met de dotNetRDF library. De dotNetRDF library definieert de categorieën en instanties als Nodes. Bij het zoeken van informatie over dotNetRDF zijn er mapping frameworks gevonden. De mapping frameworks kunnen bijdragen aan het leveren van een hogere kwaliteit code en software architectuur, maar dit zal moeten worden onderzocht in een apart onderzoek.

Wegens onzekerheid over de limitaties van de mapping frameworks binnen Unity is er een eigen technische oplossing bedacht voor het virtueel beschrijven van de categorieën (als dynamische klassen). De onzekerheid over de mapping frameworks speelt in de compatibiliteit van Unity en het kunnen werken met categorieën (denk aan de L2 van de EMont) uit de wiki die niet zijn beschreven in de architectuur van de interactieve 3d wereld.

6 Literatuur

- Abbott, t. (2010, December 12). *MDA Framework- Unconnected Connectivity*. Opgehaald van Gameasutra.com: https://www.gamasutra.com/blogs/TuckerAbbott/20101212/88611/MDA_Framework_Unconnected_Connectivity.php
- Abbott, T. (2010). *MDA Framework- Unconnected Connectivity*. Opgehaald van https://www.gamasutra.com/blogs/TuckerAbbott/20101212/88611/MDA_Framework_Unconnected_Connectivity.php
- Adams, W. E. (2005). *Interactive Narratives Revisited Ten Years of Research*. Opgehaald van http://www.designersnotebook.com/Lectures/Interactive_Narratives_Revisit/interactive_narratives_revisited.htm
- Ahmed, K. (2016, augustus 11). *github.com*. Opgehaald van dotNetRDF Documentation: <https://github.com/dotnetrdf/dotnetrdf/wiki>
- Amores, R. (2014, september 30). *Model View Controller pattern for Unity3d User Interfaces*. Opgehaald van Socialpont engineering: <http://engineering.socialpoint.es/MVC-pattern-unity3d-ui.html>
- Amyot, D. (1999). *Use Case Maps Quick Tutorial*. SITE, University Ottawa: University Ottawa. Opgehaald van <http://cserg0.site.uottawa.ca/ucm/pub/UCM/VirLibTutorial99/UCMtutorial.pdf>
- Amyot, D. (2001). *Use Case Maps as a Feature Description Notation*. University of Ottawa.
- Andrews, M. (2010, januari 23). *Game UI Discoveries: What Players Want*. Opgehaald van Gamasutra: http://www.gamasutra.com/view/feature/132674/game_ui_discoveries_what_players_.php
- Bil, A., & Bruin, H. (2017, Januari). Onderzoek oriëntatiegesprek. (L. Voncken, Interviewer)
- Borrel, A. (2015, november 25). *Unity's UI System in VR*. Opgehaald van Oculus | Developers: <https://developer.oculus.com/blog/unitys-ui-system-in-vr/>
- Bruin. (2016). *An Ontology about Expertise Management*. Vlissingen: Journal of Computer and Communications.
- Bruin. (2017, Maart 22). EVM problem explanation. (L. S. Voncken, Interviewer)
- Bruin. (2017, September). Feedback en prioritering van opgestelde eisen. (L. Voncken, Interviewer)
- Bruin. (2017). Kort feedback gesprek op de EMont dynamische model. (L. Voncken, Interviewer)
- Dijkgraaf, W. N., & van Spall, M. J. (2007). *Begin bij het eind met SMART requirements*. Synergio B.V.
- Dunstan, J. (2015, juni 29). *A Model-View-Controller (MVC) Pattern for Unity*. Opgehaald van JacksonDunstan.com: <https://jacksondunstan.com/articles/3092>
- EMM, de theorie en de praktijk - Projectenportfolio*. (2017, januari 22). Opgehaald van University of Applied Sciences: https://www.projectenportfolio.nl/wiki/index.php/PR_SSM_00065_-_resultaten
- Expertise and Valorisation Management*. (2017, februari 13). Opgehaald van HZ University of Applied Sciences: <https://hz.nl/lectoraten/expertise-valorisation-management>
- Expertise Management ontology*. (2016, oktober 9). Opgehaald van Projectenportfolio: https://www.projectenportfolio.nl/wiki/index.php/LC_00092
- Folmer, E. (2017, Mei 17). Designing Usable and Accessible Games with Interaction Design Patterns. *Gamasutra*, 6. Opgehaald van https://www.gamasutra.com/view/feature/129843/designing_usable_and_accessible_.php?page=5
- Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Boston, USA: Addison-Wesley Longman Publishing Co. Opgehaald van <http://dl.acm.org/citation.cfm?id=861282>
- Game programming patterns in Unity with C#*. (2017). Opgehaald van Habrador.com: <http://www.habrador.com/tutorials/programming-patterns/6-state-pattern/>
- Going meta: the starting points of de EMM*. (2017, Maart 23). Opgehaald van Deltaexpertise: https://www.deltaexpertise.nl/wiki/index.php/Going_meta:_managing_expertise_about_the_expertise_management_methodology_VN
- Hofstadter, D., & Sander, E. (2013). *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*. The Perseus Books Group.
- Hunicke, R., LeBlanc, M., & Zubek, R. (2017, mei 19). *MDA: A Formal Approach to Game Design and Game Research*. Opgehaald van <http://www.cs.northwestern.edu/~rob/publications/MDA.pdf>
- HZ University of Applied Sciences. (2017, maart 28). *PQR formula*. Opgehaald van Zeeweringwiki.nl: https://www.zeeweringenwiki.nl/wiki/index.php/PQR_formula
- Ip, B. (2010). *Games and Culture*. Sage publications. Opgehaald van http://www.few.vu.nl/~vbr240/onderwijs/pim/Games_and_Culture_Ip.pdf
- ISO/IEC. (2011). *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*.
- Kissner, E. (2006). *Summarizing, Paraphrasing, and Retelling: Skills for Better Reading, Writing, and Test Taking*. Heinemann.
- Lindley, C. A. (2005). *Story and Narrative Structures*. Opgehaald van designersnotebook.com: http://www.designersnotebook.com/Lectures/Interactive_Narratives_Revisit/Story_and_Narrative_Structures.doc
- Mathur, S., & Malik, S. (2010). Advancements in the V-Model. *International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 12*, 34. Opgeroepen op november 15, 2017, van <https://pdfs.semanticscholar.org/04ac/a97824d178d7ca3688bbd2118d0115dfaba.pdf>

- MediaWiki. (2017, december 4). *Help:Export*. Opgehaald van mediawiki.org: <https://www.mediawiki.org/wiki/Help:Export>
- Microsoft. (2008, oktober). *developer.microsoft.com*. Opgehaald van Introduction to Unity: <https://msdn.microsoft.com/en-us/library/ff649614.aspx>
- Montville, A. (2014, mei 11). *Implementing a replay system in Unity and how I'd do it differently next time*. Opgehaald van Gamasutra: https://www.gamasutra.com/blogs/AustonMontville/20141105/229437/Implementing_a_replay_system_in_Unity_and_how_I_d_do_it_differently_next_time.php
- MoSCoW-methode*. (2017). Opgehaald van Wikipedia.nl: <https://nl.wikipedia.org/wiki/MoSCoW-methode>
- Overleven met Kanker*. (2017, 12 28). Opgehaald van hz.nl: <https://hz.nl/projecten/overleven-met-kanker>
- Ploeg, T. (2010, juni 28). *interactieve media: iets met computers*. Opgehaald van theploeg.net: <http://www.theploeg.net/2010/06/28/interactieve-media-iets-met-computers/>
- Saunders, M., Lewis, p., Thornhill, A., Booij, M., & Verckens, J. (2004). *Methoden en Technieken van Onderzoek*. Pearson Education. Opgehaald van http://www.pearson.nl/Download/SAUNDERS_methoden_techn_editie7_H11.pdf
- Semantic MediaWiki. (2017, november 16). *API modules*. Opgehaald van semantic-mediawiki.org: <https://www.semantic-mediawiki.org/wiki/Help:API>
- Semantic MediaWiki. (2017, november 24). *Help:Maintenance script "dumpRDF.php"*. Opgehaald van semantic-mediawiki.org: https://www.semantic-mediawiki.org/wiki/Help:Maintenance_script_%22dumpRDF.php%22
- Steenbeek, M. (2015). *Iedereen aan de EMM*. Vlissingen: HZ University of Applied Sciences.
- Systems Development Life-Cycle Policy*. (2017, februari 14). Opgehaald van U.S. House of Representatives: <http://www.house.gov/content/cao/procurement/ref-docs/SDLCPOL.pdf>
- Thomas, S. (2017, mei 12). *What is Symbolism in Literature? - Definition, Types & Examples*. Opgehaald van study.com: <http://study.com/academy/lesson/what-is-symbolism-in-literature-definition-types-examples.html>
- Tulleken, H. (2012, juli 12). *50 Tips for Working with Unity (Best Practices)*. Opgehaald van Dev.Mag; A game development magazine: <http://devmag.org.za/2012/07/12/50-tips-for-working-with-unity-best-practices/>
- Tulleken, H. (2016, december 08). *50 Tips and Best Practices for Unity (2016 Edition)*. Opgehaald van Gamasutra; the art & business of making games: https://www.gamasutra.com/blogs/HermanTulleken/20160812/279100/50_Tips_and_Best_Practices_for_Unity_2016_Edition.php
- Unity. (2014, maart 3). *Persistence - Saving and Loading Data*. Opgehaald van unity3d.com: <https://unity3d.com/learn/tutorials/topics/scripting/persistence-saving-and-loading-data>
- Unity. (2017, mei 15). *AssetBundles*. Opgehaald van unity3d.com: <https://docs.unity3d.com/Manual/AssetBundlesIntro.html>
- Unity. (2017, November 29). *C# vs JS syntax*. Opgehaald van Unity3d.com: <https://unity3d.com/learn/tutorials/topics/scripting/c-vs-js-syntax>
- Unity. (2017, 11 27). *Character Controller*. Opgehaald van unity3d.com: <https://docs.unity3d.com/Manual/class-CharacterController.html>
- Unity. (2017, november 27). *Creating a World Space UI*. Opgehaald van unity3d.com: Creating a World Space UI
- Unity. (2017, 11 27). *Execution Order of Event Functions*. Opgehaald van unity3d.com: <https://docs.unity3d.com/Manual/ExecutionOrder.html>
- Unity. (2017, 11 27). *Game Managers*. Opgehaald van unity3d.com: <https://unity3d.com/learn/tutorials/projects/tanks-tutorial/game-managers>
- Unity. (2017, 11 27). <https://docs.unity3d.com/Manual/GameObjects.html>. Opgehaald van GameObjects: <https://docs.unity3d.com/Manual/GameObjects.html>
- Unity. (2017). *Image Effects: Overview*. Opgehaald van unity3d.com: <https://unity3d.com/learn/tutorials/topics/graphics/image-effects-overview>
- Unity. (2017, november 27). *Projector*. Opgehaald van Unity | Documentation: <https://docs.unity3d.com/Manual/class-Projector.html>
- Unity. (2017, november 27). *SceneManager*. Opgehaald van unity3d.com: <https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.html>
- Unity. (2017, november 27). *Time.timeScale*. Opgehaald van unity3d.com: <https://docs.unity3d.com/ScriptReference/Time-timeScale.html>
- Unity. (2017, november 27). *Variables and the Inspector*. Opgehaald van unity3d.com: <https://docs.unity3d.com/Manual/VariablesAndTheInspector.html>
- Wagner, C. (2004, februari 4). *Developing Your Own Replay System*. Opgehaald van Gamasutra: https://www.gamasutra.com/view/feature/2029/developing_your_own_replay_system.php
- Weissman, W. (2016, april 16). *Unity Shaderlab: Object Outlines*. Opgehaald van Will Weissman – Game Developer; Unity Game Development Linux Shaders Graphics: <https://willweissman.wordpress.com/tutorials/shaders/unity-shaderlab-object-outlines/>
- Wikipedia. (2017, November 29). Opgehaald van Wikipedia.nl: <https://en.wikipedia.org/wiki/Parsing>

- Wikipedia. (2017, juli 7). *Wikipedia.org*. Opgehaald van Object-relational mapping:
https://en.wikipedia.org/wiki/Object-relational_mapping
- Xuanyi, W. (2014, juli 06). *[Unity] Unity3d overview of basic concepts to look for beginners (a): GameObject*.
Opgehaald van Programering.com: <http://www.programering.com/a/MDO4UDNwATY.html>

Bijlagen

Bijlage A. Opsomming semiotiek in narratives

Tabel 2 Semiotiek in narratives in vergelijking tot de semiotiek van een interactief medium en daaruit gevormde ontwerp beslissingen.

Semiotiek in narratives (algemeen)	Semiotiek in narrative (Interactief medium)	Relevante concepten t.a.v. eisen voor narrative binnen de interactieve 3d wereld	
Narration/ verslaggeving (Het gebruik van Narrated text)	Narration/ verslaggeving (Het overbrengen van de narrative descriptors, ten aanzien van mogelijk een digitaal scherm, audio, haptische elementen, of een combinatie van de hier genoemde toestellen)	Graphical User Interface (GUI) (De door het systeem geleverde visuele ondersteuning voor communiceren met de gebruiker. Dit is uitgelegd in 1.4.4.1)	Functionele ontwerp
Narrated text (De elementen voor het overbrengen van het verhaal)	Narrated text/ Narrative descriptors (De aan de gebruiker aangeleverde elementen voor observeren en interacteren met het plot)	Barry Ip's narrative descriptors (Verzameling van GUI-elementen. Dit wordt uitgelegd in 1.4.4.1)	
Plot (De dynamische structuur van het verhaal)	Plot (Performance) (De voortgang; de gevormde digitale wereld en geschiedenis)	Narrative tijdstructuur: Dynamiek van de EMont narratives (Dit is uitgelegd in 1.1 & 1.4.4.1) & Game tijdstructuur: Gebruikersinteractie met de wereld (Dit is uitgelegd in 1.4.4.2)	
Story (verhaal) (Alle elementen in het verhaal. Deze elementen en hun onderlinge relaties zijn vastgelegd in de statisch model van de diverse de EMont niveaus in de EVM-wiki)	Story (Model/ simulation) (Wereld met historie en potentiële interactie)	Statische structuur story (EMont niveaus) & Statische vastlegging van Unity wereld met interactie en ontwerpers input (Het ontwerp dat met een prototype wordt gevalideerd in dv4)	Technische ontwerp
Structural substrate (Taal & cultuur)	Generative substrate (Ontwerpdocumenten, softwarecode, machinecode en gamegegevens)	EVM-SMW data (Een RDF-export gegeven als randvoorwaarden in 1.3) & Unity – Game Engine (Ontwikkelomgeving voor digitale spellen. Dit wordt uitgelegd in 1.4.4.2)	
	Structural substrate (Taal + cultuur)	Systeendenken (EMM) (Probleemoplossende methode beschreven in 1.4.1.1)	-

Bijlage C. Mechanics-Dynamics-Aesthetics (MDA)

Het Mechanics-Dynamics-Aesthetics (MDA) wordt als officieel raamwerk toegepast voor het gestructureerd ontwikkelen van een interactief medium, zoals de interactieve 3d wereld voor de narratives. Het raamwerk deelt de interactief medium op in drie te onderscheiden componenten:

- **Mechanics:** dit beschrijft de specifieke componenten van het medium, op het niveau van gegevensrepresentatie en algoritmen. Dit kan worden vastgelegd in een technisch ontwerp.
- **Dynamics:** dit beschrijft de mechanische werking die reageert op de gebruikers input en de systeemeigen uitvoer, gedurende de uitvoertijd. Dit kan worden vastgelegd in een functioneel ontwerp.
- **Aesthetics:** dit beschrijft de gewenste emotionele reacties opgeroepen door de speler, wanneer ze met het medium communiceert. Dit kan worden vastgelegd in het doel van de functioneel- of technisch ontwerp.

Bij het ontwerpen van een interactief medium wordt het MDA-raamwerk doorlopen A naar D. Dit komt overeen met de werkwijze uit de software development lifecycle (SDLC). Het vaststellen van het gewenste gevoel (Aesthetics), de functionele eisen (Dynamics) en de technische eisen (Mechanics) wordt vastgesteld in de elicitatiefase, ofwel analyse, fase van het SDLC. Vervolgens wordt er in de ontwerpfase van het SDLC een functioneel opgesteld, omvattende de Dynamics. In deze zelfde fase wordt op basis van het functionele ontwerp wordt een technisch ontwerp opgesteld dat de Mechanics beschrijft.

Bij het testen van het gerealiseerde interactief medium kan het MDA-raamwerk worden doorlopen van M naar A. Dit komt overeen met toegepaste V-testmodel in de testfase van het SDLC. In het V-model test eerst de mechaniek (Mechanics) aan de hand van integratietesten en Unittesten. Vervolgens test het model de functionele werking (Dynamics) aan de hand van acceptatietesten. De Aesthetics testen, staat los van het V-testmodel en dient met alternatieve test methode te worden uitgevoerd. Resultaten tot aanzien van de Aesthetics testen vormen feedback voor de functionele eisen. (Hunicke, LeBlanc, & Zubek, 2017) (Mathur & Malik, 2010)

Bijlage D. Methode (aanpak en producten) van dv2

Tabel 3 Onderzoeksmethode en deelproducten per MDA-onderdeel

MDA-onderdeel	Onderzoeksmethode(s)	Deelproduct/ Vastlegging
<p>Aesthetics</p> <p>De scope ligt alleen op de diverse emotionele responses van het model van Hunicke, LeBlanc en Zubec. (Abbott T. , 2010)</p>	<ol style="list-style-type: none"> 1. Een bronnenonderzoek naar mogelijke Aesthetics 2. Een interview naar relevante gewenste Aesthetics 	<p><u>Adviesdocument</u></p> <p>H1 – Functioneel & Technische ontwerp mogelijkheden</p> <ul style="list-style-type: none"> - § 1.1 Keuze voor Aesthetics <p>Bijlage x - Interview transcript</p>
<p>Dynamics</p> <p>De scope ligt alleen op de relevante begrippen ten aanzien van het functioneel ontwerp. Deze zijn terug te vinden in Tabel 2 van Bijlage A.</p>	<ol style="list-style-type: none"> 3. Een bronnenonderzoek naar mogelijke Dynamics voor de Aesthetics en daarvoor gebruikte Mechanics 4. Een interview naar functionele eisen (en technische eisen) voor het filteren relevante Dynamics (en Mechanics) 	<p><u>Adviesdocument</u></p> <p>H1 – Functioneel & Technische ontwerp mogelijkheden</p> <ul style="list-style-type: none"> - § 1.3 Keuze Dynamics en onderliggende Mechanics <p><u>Analysedocument</u></p> <p>H3 - Requirements</p> <p>Bijlage x - Interview transcript</p>
<p>Mechanics</p> <p>Scope ligt op de relevante begrippen ten aanzien van het technisch ontwerp. Dit is terug te vinden in Tabel 2 van Bijlage A.</p>	<ol style="list-style-type: none"> 5. Een bronnenonderzoek Mechanics voor een Unity applicatie, design patterns en wereld ontwerp mogelijkheid 6. Een bronnenonderzoek naar Mechanics voor de gefilterde Dynamics 	<p><u>Adviesdocument</u></p> <p>H1 – Functioneel & Technische ontwerp mogelijkheden</p> <ul style="list-style-type: none"> - § 1.2 Verzamelen Mechanics basis (Generative substrate, model en design patterns) - § 1.4 Advies statisch model

Bijlage E. Zoekplan voor MDA-onderdelen

Tabel 4 Zoekplan voor ieder van de MDA-onderdelen

MDA-onderdeel	Zoekmethode	Zoektermen, ofwel relevante begrippen (+ onderbouwing)	Bronnen (+ onderbouwing)
Aesthetics	Sneeuwbal methode + Uitbreiden en beperken	<ul style="list-style-type: none"> - MDA Aesthetics <p><u>Onderbouwing:</u> In game design Aesthetics heeft een specifieke betekenis binnen het MDA daarom de zoektermen "MDA" en "Aesthetics".</p>	<p>De resultaten van de Northwestern University (www.cs.northwestern.edu/)</p> <p>Onderbouwing: Northwestern University heeft een blogstelsel voor docenten, waaronder voor game research.</p>
Dynamics	"	<ul style="list-style-type: none"> - Graphical User Interface (GUI) interactive media/ games - Barry Ip's narrative descriptors - Game time structure - Game usability and accessibility design patterns <p><u>Onderbouwing:</u> Op basis van de van de relevante begrippen die zijn genoemd in theoretisch kader (§1.4.4). De game usability en accessibility design patterns zijn voor kwaliteit van gebruiksvriendelijkheid.</p>	<p>Officiële Unity website (https://unity3d.com/)</p> <p>Onderbouwing: Deze website bevat adviezen, verhalen en voorbeelden van Unity experts. Deze bevinden zich in het forum, de nieuwsartikelen en handleidingen.</p> <p>De resultaten van de Google Searchengine (http://Google.com)</p> <p>Onderbouwing: Het komt veel voor dat experts een blog schrijven over deze aandachtspunten. Deze blogs zijn niet te vinden binnen de Unity website, en zijn verdeeld over verschillende sites.</p> <p>Het theoretisch kader</p>
Mechanics	"	<ul style="list-style-type: none"> - Unity – Game Engine class diagram - Unity designers input - [Gevonden Dynamics] code example - Unity design patterns <p><u>Onderbouwing:</u> Op basis van de van de relevante begrippen die zijn genoemd in theoretisch kader (§1.4.4). De "unity design patterns" zijn voor softwarekwaliteit te kunnen garanderen.</p>	<p>Onderbouwing: In het theoretisch kader is er al wat verklaard over de Gametijdstructuur en de narrative descriptors</p>

Bijlage F. Zoekplan van het bronnenonderzoek van dv4

Tabel 5 Zoekplan voor het vinden van omzetsmethode

Zoekmethode	Zoektermen (+ onderbouwing)	Bronnen (+ onderbouwing)
Sneeuwbalmethode + Uitbreiden en beperken	<p>- Export [pages of categories]</p> <p>Onderbouwing: Er zal gezocht worden naar “exporteer” mogelijkheden, daarom wordt er gezocht met het Engelse woord “export”.</p> <p>De zoekterm van de “pages” of “categories” relevant, omdat deze de vastlegging van de EMont definiëren. Dit is verteld in 1.4.2.1 van het theoretisch kader.</p>	<p>Officiële MediaWiki website (www.mediawiki.org)</p> <p>Officiële Semantische MediaWiki website (www.semantic-mediawiki.org)</p> <p>Documentatie van de live EVM-SMW “Projectenportfolios.nl” (www.projectenportfolio.nl)</p> <p>Onderbouwing: Dit zijn officiële bronnen met documentatie over het SMW-systeem van de EVM-SMW.</p>
	<p>- C# reader parser [MW of SMW]</p> <p>Onderbouwing: Uitlezen of begrijpen in de computerwereld wordt ook wel gedaan met readers en parsers. (Wikipedia, 2017) De term “C#” is, omdat randvoorwaarde 6 van onderzoek alleen C# wil. Randvoorwaarde 2 eist dat de exporteer functies van de SMW worden gebruikt.</p> <p>- [Gevonden exporteer formaat] Mapping</p> <p>Onderbouwing: Er zal alleen worden gezocht naar het relevante exporteer formaat. Ook is het relevante om “mapping” als term te gebruiken, want die term staat voor het leggen van een koppeling tussen entiteiten uit data en entiteiten als softwareobjecten. (Wikipedia, 2017)</p>	<p>https://www.nuget.org/</p> <p>Onderbouwing: NuGet is een package manager voor .NET. Waar Unity gebruik van maakt. (Microsoft, 2008)</p>

Bijlage G. Game usability en accessibility design patterns

Hieronder bevindt zich een kleine samenvatting van de bevindingen van een artikel over diverse ontwerp patronen die zijn toe te passen als stimulans voor de gebruiksvriendelijkheid en (mentale en fysieke) toegankelijkheid van een interactief medium. Dit artikel is geschreven door Eelke Folmer, een assistent Professor aan de University of Nevada in Reno. (Folmer, 2017)

Usability (gebruiksvriendelijkheid) probleemgebieden

In het artikel van Folmer zijn vijf relevante probleemgebieden genoemd ten aanzien van de gebruiksvriendelijkheid van een interactief medium (Folmer, 2017):

- Prevent waiting: het voorkomen dat de gebruiker moet wachten.
- Prevent errors: het voorkomen dat de gebruiker te maken krijgt met gebruikers- of systeemfouten.
- Communicate status: het doorgeven van de staat van de gebruiker aan de gebruiker.
- Adapt tot de player: het aanpassen van de dynamiek of de wereld aan de gebruiker.
- Provide help: het bieden van ondersteuning van voor de gebruiker.

Accessibility (toegankelijkheid) type handicappen

Er zijn vier handicap varianten genoemd waar rekening mee kan worden gehouden (fysieke en mentale) ten aanzien van toegankelijkheid van een interactief medium (Folmer, 2017):

- Auditory disability: het ondersteunen van de doof of slecht horende gebruikers.
- Physical disability: het ondersteunen van gebruikers met een verlamming, neurologische aandoeningen, RSI (blessure door herhaalde belasting van spieren en gewrichten) en leeftijdsgebonden problemen.
- Visual disability: het ondersteunen van gebruikers met blindheid, slechtziendheid en kleurenblindheid.
- Cognitive disability: het ondersteunen van gebruikers met leerstoornissen, zoals dyslexie, dyspraxie, autisme en aandachtstekortstoornis.

Toepassingen van de ontwerp patronen voor Usability en Accessibility

In Tabel 6 worden enkele ontwerp patronen beschreven voor het verbeteren van de Usability en Accessibility.

Tabel 6 Toepassingsmogelijkheden voor de ontwerppatronen (Folmer, 2017)

Ontwikkelpatroon	Beschrijving van Ontwerptoepassing	Gebruiksvriendelijkheid probleemcategorie	Toegankelijkheid handicap
Seamless Gameworld	Het aanleveren van een naadloze wereld, ofwel een doorlopende wereld. Dit in tegenstelling tot het steeds opnieuw laden en plaatsten van de gebruiker in een geheel nieuwe wereld.	Prevent waiting	
Skippable Cutscenes	Het toestaan van om cutscenes over te slaan.	Prevent waiting	
Fast Forward	Het bieden van een mogelijkheid om door te spoelen door de tijdstructuur van het spel.	Prevent waiting	
Quick Save/Load	Het simpel en kort opslaan van voortgang van de gebruiker.	Prevent waiting	
Quick Start or Restart	Het simpel en snel kunnen herstarten.	Prevent waiting	
Pre Loader Game	Het inladen van de wereld in de achtergrond, zodat deze klaarstaat wanneer deze opgevraagd wordt.	Prevent waiting	
Arcade Mode	Het toestaan om de gebruiker te laten interacteren met de wereld zonder dat dit bijdraagt aan het doel.	Prevent waiting, Adapt to the player	
Slow	Het vertraag complexe situaties.	Prevent errors, Adapt to the player	Physical disability, Cognitive disability
Rewind	Het aanbieden van de mogelijkheid om een gebruiker een situaties opnieuw te beleven.	Prevent errors	
Auto save	Het automatisch opslaan.	Prevent errors	
Visual saves	Het aantonen dat er wordt opgeslagen.	Prevent errors	
Pause	Het mogelijk maken om het spel tijdelijk stil te zetten.	Prevent errors	

Free look	Het mogelijk maken voor de gebruiker, om door de wereld te navigeren met enorme bewegingsvrijheid.	Prevent errors	
Instant Replay	Het mogelijk maken voor de gebruiker om direct een terugblik te kunnen geven van een net doorlopen situatie.	Communicate status	
Game Progress	Het tonen aan de gebruiker wat zijn/haar voortgang is.	Communicate status	
Closed Captioning	Het meeleveren van een ondersteund visueel transcript van gebeurtenissen in een verhaal.	Adapt to the player	Auditory disability
Interaction Aids	Het bieden van ondersteuning bij interacteren.	Adapt to the player	Physical disability, Visual disability
Adaptive Difficulty level	Het aanpassen van de moeilijkheidsgraden.	Adapt to the player	Physical disability
Reconfigurable buttons	Het mogelijk maken voor de gebruiker om knoppen anders in te stellen.	Adapt to the player	Physical disability
Tutorial Agent	Het aanbieden van een ondersteuning door een virtuele begeleider.	Provide help	Cognitive disability
Interaction Aids	Het ondersteunen bij het interacteren met de wereld, door het gebruik van automatische afhandeling of het leveren van extra informatie over de handeling.	Provide help	Cognitive disability
Playground	Het aanbieden van een scenario binnen het spel waar bij de gebruiker zonder consequenties handelingen kan uitvoeren.	Provide help	
Journal	Het aan bieden van een logboek van uitgevoerde activiteiten.	Provide help	Visual disability

Bijlage H. EMont en requirements interview (uit het analysedocument)

Dit is een interview dat vragen heeft gesteld ten aanzien van de EMont structuur en hoe deze is verwerkt binnen de wiki. Om efficiënt gebruik te maken van de tijd van het onderzoek is als toevoeging aan het interview is ook gevraagd naar de functionele en eventuele technische eisen van de interactieve 3d wereld.

Behandelde Interview topics

De gebruikte interview topics voor het analyseren van de EMont structuur binnen de wiki's zijn genoemd in 2.1.3.

Zoals genoemd in 2.2.4 zal er worden gevraagd naar de gewenste toepassing van de gevonden informatie over de veel gebruikte functionele toepassingen. Zie tabel hieronder voor de vier functionele toepassingen met referentie naar gedetailleerde beschrijving.






Tabel 7 Functionele toepassingen met referentie naar detailbeschrijving

Functionele toepassing	Uitgebreide definitie
Graphical User Interface (GUI) interactive media/ games	Zie 1.4.4.1 van het theoretisch kader.
Barry Ip's narrative descriptors	
Game timestructuur	
Game usability en accessibility design patterns	Zie Bijlage F.

Interview transcript met codering

Het interview is vastgelegd in een transcript waarop een codering is toegepast. De codering is gedaan door de met kleuren aan te geven welke woorden combinaties relevant zijn. De relevante betekenis is terug te vinden in de legenda (Tabel 8).

Tabel 8 Legenda betekenis kleur relevantie voor het onderzoek

Legenda	
	EMont categorie (Entiteit)
	Wiki element (Entiteit)
	Narrative element/uitdrukking (Entiteit en relaties)
	Eisen aan narrative 3d
	Overige bruikbare informatie

[Introductie]

00:00 – 02:00

Laurens: “Is de EMont-model speciaal opgesteld voor de narratieven van het Overleven met Kanker Living Lab? Of werkt het universeel?”

Hans: “EMont is een ontologie en universeel op alle mogelijke gebieden, dat gaat voor dijkversterkingen, zoals we dat gebruik bij Building with nature, Aquacultuur, dus als dingen die in de Delta zitten. Maar het geldt ook voor Veerkrachtige samenlevingen, Scholen voor de toekomst. Het maakt niet uit. Alle domeinen eigenlijk.”

02:00 – 04:35

Laurens: “Dus voor elk Living Lab in het Projectenportfolio?”

Hans: “Ja, dat klopt.”

Laurens: “Ik heb een keer een onderzoek gedaan naar narratieve element, dit voor het algemene begrip narratief. Ik deze samengevoegd in een overzicht. Zijn deze elementen allemaal meegenomen in de EMont “ontologie?”

Hans: “Ik heb zelf ook een hele studie gedaan naar narratieven, een hele stapel boeken thuis. Door Middel hiervan ben ik tot de conclusie gekomen dat de EMont deze narratieven gewoon kan uitdrukken, dus alles wat je hier ziet kan worden meegenomen.” *Hij wijst naar het overzicht met de narratieve elementen*

Hans: “Om iets op te noemen, we gaan uit van een situatie, dit is te vergelijken met de setting (narratief element). Of de plaats. Dit dan geven in ... relatief van de andere settings. Dus dan heb je tijd, dus je gaat van de ene situatie settings. Dan heb je een opeenvolging van scenes, of hoe je het ook wil noemen. Dus dat zit er altijd wel in. Er spelen daar altijd randvoorwaarden in, dat zijn de condities, weerscondities of wat voor condities je daar ook in wil doen, maar ook sociale condities, ook de mood. Dus alles wat je hier inzet kun je in termen van condities beschrijven, dus de hele setting zit erin.”

04:35 – 06:00

Hans: “Binnen die setting heb je een aantal actoren die actief zijn. Die actoren zitten in samenhang. Die proberen een bepaalde doelstelling te doen/ een beweging maken om al niet een gemeenschappelijk doel of acteren uit hun concerns. Waar ze zich druk overmaken.”

Iedere actor heeft zijn eigen kijk op de zaak, dat is een perspectief. Je zou er boven kunnen hangen (kijker/omniscient/god). Maar je kan ook inzomen op een karakter/actor.

Laurens: “Is dat ook meegenomen in de wiki?”

Hans: “Dat is gewoon meegenomen in de ontologie. In de ontologie kan je dit allemaal beschrijven.

06:00 – 08:40

Hans: “Over het plot (narratieve element). Is gewoon de opeenvolging van scenes. Daarbij begin je altijd met een probleem, dat probleem gaat dan weer van kwaad naar erger, dan is er ook vaak iets aan de gang dat tot resolutie gaat komen (oplossingen). En dan zakt dat weer af. Dus eerst de eerste situatie de introductie, dan de beweging van de actoren, dat leidt tot een clash en dat gaat weer naar een resolutie. Dat is een standaard opeenvolging van een narratief.”

Laurens: “Dus dat is niet standaard zo meegenomen in de EMont?”

Hans: "Nee. Het is gewoon een ontologie waar je menselijke kennis en handelen in kan vastleggen, maar daar kun je dus ook het hele leven in beschrijven, inclusief narratieven. En dat je het zo gaat zeggen dat we dit zien als een narratief en dat zien als de toenemende actie (onderdelen van het narratieve element "plot"). Dat kan allemaal beschreven worden in de EMont, dus de willekeurige narratieven kunnen we zo beschrijven en dat doen we dan ook. Sterker nog, de karakters zitten verwerkt in de rollen. Die karakters kunnen ook ontwikkeld worden, dus naar aanleiding van condities, hij overgaat tot andere acties. He ik bedoel, je doet acties je doet acties, je bent in interactie met iemand anders. Dat kan leiden dat je tot andere overtuiging komt en dat je dan andere paden gaat kiezen. Dus dat zit ook opgesloten in de ontologie. Dus actoren kunnen ook een progressie door maken. Me andere woorden, als gevolg van interactie met andere een ander wereldbeeld krijgen (oftewel ander perspectief)."

08:40 – 09:50

Laurens: "Kan je dat zelf invullen?"

Hans: "Dat moet je modelleren in de EMont. Eigenlijk is Emont een foundation ontologie. Daarmee kun je concrete zaken modelleren, zoals hoe maak ik een duik. Dat kun je er allemaal meedoen, dus wat dat betreft is het een waanzinnige ontologie. Tijd heeft dat ons geleerd, want we gebruiken het al 4-5 jaar en he is nog niet veranderd in deze tijd, dus ook voor narratieven. Dit komt ook doordat ons uitgangspunt het menselijk handelen heeft genomen. Al deze (narratieve) elementen zitten er gewoon in."

00:09:50 – 00:16:50

Hans: "We maken een onderscheid in verschillende niveaus. Dit zijn drie niveaus. Het L0-niveau, dit is de foundation. Hier wordt er gedefinieerd wat een activiteit is, wat een doelstellingen, een context en alle elementen zitten daarin benoemd.

Vervolgens kun je dat toepassen op een verschillend domein (Thema's) als overleven met kanker of Dijkversterking. Bij dijkversterking kan dit zijn 'Hoe maak ik een dijk' en voor Overleven met Kanker 'Hoe voer ik een slecht nieuws gesprek'.

Deze zijn te vinden op:

"Toont website met url: [https://www.projectenportfolio.nl/wiki/index.php/PR_00019 - resultaten](https://www.projectenportfolio.nl/wiki/index.php/PR_00019_-_resultaten)"

Dat zijn de hoe doe je dat dit zijn de handelingsperspectieven. Bij een slechtnieuwsgesprek zitten dan wat actoren hieronder vallen artsen en een specialist, plus patiënten en eventueel een partner die er bij een rol speelt zit. In die setting gebeurt dat eigenlijk. Bij "Hoe maak ik een dijk" hoe gaan we dus zodanig met stenen aan de weer dat we uiteindelijk een dijk construeren, nou. Dit noemen we dan handelingsperspectieven. Afhankelijk van omstandigheden kan een oplossing wel werken en een ander niet. En als gevolg daarvan doe je bepaalde dingen dus wel of niet, maar het kan ook zijn dat je allemaal gefixeerde ideeën hebt, als "Alle Moslims het land uit". Maakt niet uit wat ze doen. Het is een gefixeerd idee. Ik ben tegen dat, dan is iedereen weg. Dan hebben we echt een belief, dat is wat we dat noemen. (een overtuiging). In onze optiek ligt een overtuiging gewoon vast. JE raakt er ook mede bij dat de ene in een bepaalde situatie dit doen en de andere totaal niet. Dat kan afhankelijk zijn van je overtuiging.

Een conditie hebben we ook dat is een maat van iets, dat kan van alles zijn. Het kan het klimaat zijn, het zonnetje schijnt, het regent. Zo'n conditie kun je eigenlijk weer opsplitsen in meerdere condities, die in totaal weer iets wat zeggen over het weer, dus de hitte, warmte, neerslag, de wind (op dat moment). Dus de samengesteld conditie. En zo kun je iets over het weer gaan zeggen, maar je kan ook wat zeggen het sociale domein. Je kan wat zeggen over de sociale conditie. Van hoe staat jou mentale conditie ervoor, ben je in de war, ben je dit, ben je zus, ben je zo.

Laurens: "En dat wordt meegenomen in de vraag?"

Hans: "Hier (Niveau L0 in de EMont) hebben we het over het algemene term conditie, maar op dit niveau (L1) wordt dat geconcretiseerd, dan spreek je van mentale condities, of we praten van de conditie van het hoge water.

Dus dat ga je hier concretiseren afhankelijk van het domein waarin je loopt te opereren, dit noemen we ook wel practices/handelingsperspectieven van hoe je in bepaalde domeinen het best kan opereren. Daar hebben we methode voor, dat is de Expertise Management Methodologie, om te kijken van "Jij doet het zo", "Jij doet het zo". Met dan de vraag: "Wat kunnen we dan het best doen met elkaar?" Zo kom je tot die practices. Die practices komen meestal tot stand op basis van experiences. Een experience is iets wat echt meegemaakt is, een echt verhaal van iemand. Die

interactie die je daar gezien hebt met clair dat is iets wat claire daadwerkelijk heeft meegemaakt. Een experience is een sequentie van situaties.

Laurens: "Dat zijn de casussen die ik zag staan in de wiki?"

Hans: "Ja, dat klopt. En die schrijf je gewoon in de EMont allemaal. Wat je dan ziet. Hier lees je altijd een instantie van.

Tekent een lijntje aangevende het ontstaan van de instantie in L1 uit L0

"En hetzelfde zie je ook hier"

Tekent een lijntje aangevende het ontstaan van de instantie in L2 uit L1

Zo zie je ook verschillende niveaus. **Die drukken we uit door middel van een instantie van.** Net als je OO-programmeren (klassen en instanties).

Een object is een geconcretiseerd geval die je als blauwdruk hebt. Dus hier kunnen we in activiteit tegen activiteit zeggen we gooien stenen neer, voor een duik. Dat is dan een instantie voor een duik. Oké, op die manier. Hetzelfde doen we ook hier. Ik heb hier een situatie, bijvoorbeeld een slecht nieuwsgesprek. Claire voert een slechtnieuwsgesprek met een arts (L2). Is een instantie van "Hoe voer ik nou een slechtnieuwsgesprek?", soort voorbeeld van."

16:50-21:24

Hans: "Waar wij dus mee bezig zijn is/jij mee bezig bent, is dit niveau, het L2 niveau. Jij bent daarin een verhaal aan het vertellen van een concreet iemand. We kunnen geen abstracte verhalen vertellen. En dat je het kan abstraheren van dit zijn goede gesprekken en dit zijn slechte gesprekken, dat is wat je leert uit de experiences. Net als in een Roman zijn deze mensen geen zombies of aliens. *lacht* Dit is wel een overdreven geval.

We praten over mensen van vlees en bloed! Het is dus de bedoeling die diepere laag te ontdekken. Van oké, dit is de algemene les, die hier speelt. Net als de functie van een Roman. Dus met andere woorden: ze zien dit, maar denken dit.

Het belangrijk onderdeel van dit is het analogisch denken. Dat is wel een heel belangrijk begrip. Mensen handelen niet zomaar, dat is een op basis van wat ze in het verleden hebben meegemaakt. Dat is analogisch denken.

In zekere zin kun jij in de toekomst kijken. Daar ben je niet uniek in dat kan iedereen. Dat kan doordat jouw ervaringen kun je voorspellen wat er gaat gebeuren. Nu blijkt door dat jij door schade en schande jij bent wie je bent. Dat zijn de **beliefs** in de EMont. Dit kan veranderen door bepaalde **condities en acties**.

Het belangrijke is dus dat mensen zich kunnen inbeelden in jouw situatie en zo kunnen voorspellen wat jij gaat doen. En in dat samenspel kun je lessen allemaal lessen leren die anderen daarin meegeven. Jij kan als ware, als je een Roman leest, de diepere lagen ontdekken.

Wat je wilt bereiken met het analogie denken is dat je naar bekende situatie wilt gaan waar je zo iemand kan inleven. **Maar deze situaties zijn zo rijk dat je die nauwelijks kan beschrijven!** Maar als je dat bilden kan uitdrukken, of in een Roman met plotten in zinnen kan doen. Dan kun je meer zeggen dan lijsten met bulletpoints.

Maar deze mensen vullen deze plotten met hoge mate zelf in met eigen ervaringen en dan kom je tot herkenbare situaties. En dan begrijp je "aha, ik snap hoe de situatie werkt!" Dus je helpt zo om de situatie te begrijpen en te doorgronden, dat is ook precies de functie van een film of een boek.

Je helpt zo mensen tot andere denkbeelden en inzichten te komen"

00:21:24 – 00:44:00

Laurens: "Hoe kan ik dat terugvinden in de wiki? Ligt de EMont in een model uitgetekend?"

Hans: "Ja"

Laurens: "En ik neem aan dat alle instanties vastliggen in de wiki?"

Hans: “Ja, dat klopt. de EMont ligt gewoon vast. In het artikel wordt het haar fijn uitgelegd”

toont wiki pagina: <http://www.scirp.org/Journal/PaperInformation.aspx?PaperID=73672>

Laurens: “Is dat exact dezelfde als die ze nu gebruiken?”

Hans: “Ja dit model is vastgelegd in de Semantische MediaWiki (SMW). Binnen de SMW heb je categorieën, dit zijn gewoon klassen. In term van Object-georiënteerdheid. En een wikipagina is een instantie van een categorie van. Dus een willekeurige webpagina is een instantie van de categorie ... *zoekt voorbeeld op de wiki*... bijvoorbeeld context, deze kan geïnterpreteerd worden als een situatie of een rol.

Pakt een voorbeeld Deze pagina is een instantie oftewel een object van de klasse light context. En dan gaan we naar de categorie zelf (light context) en dan zie je welke instanties er van zijn.

Wat je ook ziet is dat een categorie “light context” een subklasse is van de categorie “context”.

De toewijzing van een pagina aan de EMont core is gewoon gemakkelijk voor gebruik. Dit toont aan dat een pagina aan meerdere klasse kan worden toegewezen. In Java heb je dat ook. Dus het hele de EMont ligt geconcretiseerd vast in de SMW's.”

Laurens: “Je sprak van properties, zijn dit de soort van condities waar je het eerder over had?”

Hans: “Ja, dat kunnen ook condities zijn. Nou, nee. Condities zijn ook weer instanties van categorieën. Dus nee niet helemaal. Properties zijn gewoon eigenschappen. Dus als je naar de light context gaat kijken, hier hebben we een voorbeeld van Wicked Problems, als je gaat kijken naar het bewerken van het formulier, dan zie je dat toedichten aan deze pagina. Je bouwt een hiërarchie van light contexten. Dan zie je de top context is dit *wijst aan*. Je hebt een volgnummer en een context titel. Dit zijn de properties die iets zeggen over de context. Dat is gewoon te vergelijken met een attribuut in een klasse. En wat je doet met dit soort dingen is het vastleggen van de relaties met andere objecten. Zo maak je een heel semantisch netwerk dat bestaat uit het element die worden gedefinieerd in de ontologie. Dat zijn dus allemaal instanties van ontologische elementen. Zo zie je een groot Object-georiënteerd programma gevisualiseerd. Het is echt een hele krachtige combinatie. Het is enerzijds heel veel structuur, een semantische database, en anderzijds gewoon puur tekst wat erin staat. Dus je kan gewoon in tekst van alles neer kan zetten over wicked problems en wat dan ook, maar tegelijkertijd bouw je een hele grote semantische database.”

Wat minder belangrijk rest is niet zo belangrijk

Laurens: “Waar kan ik de condities vinden?”

Hans: “Nee die worden nog niet aangegeven. Als we praten over weer of neerslag, dan weten wij waar we het over hebben. Helaas weet de computer. Daarom hebben we de semantische wiki nodig. Er is de mogelijkheid om aan objecten meer informatie toe te voegen. Dit kan door het gebruik van een thesaurus. Wat we doen is we maken een piramide (data→info→kennis→wijsheid). Wij maken onderscheid tussen knowing that en know how. Knowing that is feiten. Knowing how is hoe doe ik dingen. Bij data/informatie (knowing a.k.a feiten) is het bijvoorbeeld een ziekenhuis. Wat is een ziekenhuis. Maar bij de knowing how gaat het over het hoe behandelen we patiënten in het ziekenhuis.

Wijst naar de data en informatie laag in de getekende piramide Dit is de know that kennis, deze feiten leggen we vast in het SKOS (standaard). Dit is een gestructureerde woordenlijst (plat gezegd). Dit kan zijn weer, maar deze kan weer bestaan uit temperatuur, en wind of een zeewering deze kan weer onderverdeeld worden in duin en dijk. Thesaurus definieert element waar je meer aan toe kan voegen. |

Wijst naar de formatie laag in de getekende piramide Dit is de know that kennis, deze feiten leggen we vast in het SKOS (standaard).”

00:44:00 – 00:46:00

Hans: “Dit is het gebeuren en zo kan je het zien. En wat jij kan doen is op basis van de abstracte redelijk begrippen, als wij maken gebruik van **condities** in een bepaald domein kun je zo'n conditie invullen. Dan praat je over mentale conditie, weersconditie, of wat dan ook. Maar als jij zo vind van dit wil ik ook kunnen visualiseren in mijn wereld, ik wil iets met weer kunnen doen. Dan is het de bedoeling dat je deze elementen gaat koppelen aan de bestaande elementen uit de

ontologie, die we hebben. Zo zit het semantisch web in elkaar, iedereen mag wat zeggen van een bepaald begrip. Zo kan iedereen pragmatische zaken voor jou te doen zonder dat je het hoeft te gaan vissen.”

Hans: “Is het een beetje helder?”

Laurens: “Ja, ja. Ik heb al eerder het document doorgelezen wat u net liet zien.”

00:46:00 – 00:44:00

Laurens: “Hoe zie je exact dat het wordt uitgevoerd? Applicatie die op alle platformen werkt en dat iedereen kan gebruiken? En dat je een casus kiest die dan wordt geselecteerd en zo kan worden doorlopen?”

Ik zag namelijk vele verschillende situaties, maar hoe maak ik dit dan interactieve beleving dan?”

Hans: “Er is niet een soort gebruiker.”

pakt een papiertje Je moet de wegen zien die we eerder hebben getekend. (soort van persoonlijke levenslijnen bestaande die door bepaalde situaties lopen, ook in een situatie die een ander ook meemaakt)

Hans: “Stel dat we twee actoren hebben a.k.a een levenslijn. En maken eigen situatie mee. Soort van toenemende actie uit je narratieve elementen is terug te vinden in de eerste keer dat de paden kruisen van de twee actoren. En dan kunnen de lijnen weer uit elkaar gaan. Dan komen er andere actoren bij en zo kun je per situatie kiezen vanaf wel perspectief je het verhaal wilt bekijken. Dit kan zijn de “omniscient, firstperson en thirtperson”. Maar ook hier weer boven kun je de situatie zien als een super situatie. Dus een context kun je opdelen in situatie en rollen. Dus je moet even de begrippen goed doorlezen. Daar staat alles in beschreven.

In zo’n situatie voeren de actoren/rollen. Daar hoort een bepaald gedrag bij, bepaalde doelstellingen, maar het kan ook zijn dat de interactie hier en wereld overtuiging van je kan veranderen. Dus als hier een bepaalde interactie plaatsvindt dan kan er in plaats van deze keuze een andere keuze wordt gemaakt.”

Laurens: “Maar in de wiki is een situatie vastgelegd in casussen.”

Hans: “De situatie (het plot) ligt vast, maar er zijn vrijheidsgraden gedefinieerd.”

Laurens: “Zijn er nog eisen waaraan er moet worden gedaan?”

Hans: “Ik wil wat ik net definieert voor de 3d beleving. De methode hoe deze worden uitgebeeld, dat is aan jou. Jij moet voorstellen voorleggen en deze zo nodig overleggen met de rest van de ”

[Wat irrelevante informatie]

Hans: “We horen graag de voorstellen die jij hebt voor het uiteindelijke product.”

[Afsluiting]

Bijlage I. Bronnenonderzoek naar statische de EMont structuur binnen een SMW

Stap 1 – Handmatige doorlopen van de wiki pagina's en sjablonen

In de volgende stap zijn de resultaten vastgelegd. Zie de stap hieronder.

Stap 2 – de EMont elementen, per niveau, binnen de wiki

Tussen deze pagina's bevinden een zich de volgende interessante pagina's, die hun interesse danken aan hun onderliggen relatie. Deze relaties zijn gevonden door:

- [Activity](#) (8 items)
- [Actor](#) (6 items)
- [Context](#) (26 items)
- [Context VN](#) (14 items)
- [EMont core](#) (406 items)
- [Handelingsperspectief](#) (23 items)
- [InfKrt Contact Person](#) (8 items)
- [InfKrt Organization](#) (57 items)
- [Intentional Element](#) (2 items)
- [OLmK](#) (11 items)
- [OLmK Case](#) (6 items)
- [OLmK Case VN](#) (6 items)
- [OLmK Context](#) (39 items)
- [OLmK Context VN](#) (39 items)
- [Pattern](#) (2 items)
- [SSM](#) (27 items)
- [SSM Situation](#) (56 items)
- [SSM Situation VN](#) (51 items)

Tabel 9 Bestaande EMont categorieën verdeelt onder het juiste de EMont niveau

Level 0: de EMont "Foundation"		Level 1: de EMont "top level ontology"	Level 2: "upper ontology"
Actor	EMont core -> Intentional Element -> Actor	Actor -> InfKrt Contact Person Actor -> InfKrt Organization	- Niet gevonden
Activity	EMont core -> Intentional Element -> Pattern -> Activity	Activity -> Handelingsperspectief	
Context	EMont core -> Context	Context -> OLmK Context	Context -> OLmK Case
			Context -> SSM Situation
Goal	EMont core -> Intentional Element	-	-
Condition	EMont core -> Intentional Element	-	-
Belief	EMont core -> Intentional Element	-	-
Outcome	EMont core -> Intentional Element	-	-
Practice		EMont core -> Practice	-

Stap 3 – Gevonden attributen (en relaties) per de EMont element

Tabel 10 Entiteiten en relaties herleidt uit de categoriepagina's en formulieren van deze pagina's

EMont element (Entiteit) + Attributen (rood = relaties)	Bron(nen)
IntentionalElement + intentionalElementType: Enum + decompositionType: Enum + concerns: String + contributes: Contributes + depends: Depends + partOf: IntentionalElement + context: Context	Wiki sjabloon en fomulier van de IntentionalElement pagina uit de projectenportfoliowiki van het EVM-lectoraat: 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Intentional_Element 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:Intentional_Element
Depends + note: String + link: IntentionalElement	
Contributes + contributionValue: Enum + note: String + link: IntentionalElement	
Context + description: String + contextType: Enum + summery: String + supercontext: Context	Wiki sjabloon en fomulier van de Context pagina uit de projectenportfoliowiki van het EVM-lectoraat: 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Context 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:Context
ThemaContext [Overerving van Context]	Wiki sjabloon en fomulier van een context van een Living Lab thema pagina uit de projectenportfoliowiki van het EVM-lectoraat: 1. Sjabloon context "OLmK": https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:OLmK_Context 2. Formulier context "OLmK": https://www.projectenportfolio.nl/wiki/index.php/Formulier:OLmK_Context
Case [Overerving van Context] + familySituation: type + lifecycle: String + educationAndWork: String + outlookOnLife: String + socialNetwork: String + medicalNetwork: String	Wiki sjabloon en fomulier van een "case" pagina uit de projectenportfoliowiki van het EVM-lectoraat: 1. Sjabloon casus "OLmK": https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:OLmK_Case 2. Formulier casus "OLmK": https://www.projectenportfolio.nl/wiki/index.php/Formulier:OLmK_Case

<p>SSMSituation</p> <p>[Overerving van Context]</p> <ul style="list-style-type: none"> + summary: String + startdate: timestamp + enddate: timestamp + concerns: String + situationType: String + ssmCommonGoal: String + ssmSucces: String + ssmRichImage: String + ssmIntervention: String + ssmInterventionEffect: String + previous: SSMSituation + ssmWorldviews: SSMWorldview[] 	<p>Wiki sjabloon en fomulier van de SSM Situatie pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:SSM_Situation 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:SSM_Situation
<p>SSMWorldview</p> <ul style="list-style-type: none"> + ssMrole: String + ssMstakeholder: String + ssMwhat: String + ssMhow: String + ssMwhy: String + ssMcondition: String + ssMbelief: String + ssMsuccess: String + ssMworldviewBackLink: String 	<p>Wiki sjabloon en fomulier van de SSM Worldview pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:SSM_worldview 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:SSM_Situation
<p>Practice</p> <ul style="list-style-type: none"> + practiceType: Enum + partOf: Practice 	<p>Wiki sjabloon en fomulier van de practice pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Practice 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:Practice
<p>Selects</p> <ul style="list-style-type: none"> + selectionType: Enum + selection contribution: Enum + selectionLinkNote: IntentionalElement 	<p>Wiki sjabloon en fomulier van de selects pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Selects 2. Formulier: - Niet gevonden -
<p>Actor</p> <p>[Overerving van IntentionalElement]</p> <ul style="list-style-type: none"> + organisation: Organisation + concern: String + responsibility: String 	<p>Wiki sjabloon en fomulier van de Actor pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Actor 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:Actor

<p>ContactPerson</p> <p>[Overerving van Actor]</p> <ul style="list-style-type: none"> + name: String + address: String + function: String + Address: String + houseNumber: Int + zipcode: String + location: String + phone: Int + email: String + website: String + info: String 	<p>Wiki sjabloon en fomulier van de Contact Person pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:InfKrt_Contact_Person 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:InfKrt_Contact_Person
<p>Organisation</p> <p>[Overerving van Actor]</p> <ul style="list-style-type: none"> + name: String + address: String + houseNumber: Int + zipcode: String + location: String + phone: Int + email: String + website: String + info: String 	<p>Wiki sjabloon en fomulier van de Organisation pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:InfKrt_Organiz_ation 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:InfKrt_Organiz_ation
<p>Activity</p> <p>[Overerving van IntentionalElement]</p> <ul style="list-style-type: none"> + decompositionType: Enum + partOf: Activity + concerns: Activity + consumes: Activity + produces: Acitivity 	<p>Wiki sjabloon en fomulier van de Activity pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Activity 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:Activity
<p>Handelingsperspectief</p> <p>[Overerving van Activity]</p> <ul style="list-style-type: none"> + tips[]: Tip 	<p>Wiki sjabloon en fomulier van de Handelingsperspectief pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Handelingsperspectief 2. Formulier: https://www.projectenportfolio.nl/wiki/index.php/Formulier:Handelingsperspectief
<p>Tip</p> <ul style="list-style-type: none"> + targetAudience: String + description: String 	<p>Wiki sjabloon en fomulier van de Tip pagina uit de projectenportfoliowiki van het EVM-lectoraat:</p> <ol style="list-style-type: none"> 1. Sjabloon: https://www.projectenportfolio.nl/wiki/index.php/Sjabloon:Tip

Zijn niet gebruikt in de Overleven met Kanker wiki.

4. Wat betekenen hoe zijn de SSM properties terug te vinden in de EMont? En wat betekenen de individueel?

De elementen beschrijven de situatie context.

5. Waar vind ik alle Decompositie types van de handelingsperspectieven? En wat betekenen ze allemaal?
(Voorbeeld: https://www.projectenportfolio.nl/wiki/index.php/OLmK_HP_Aanspreekpunt)

Er zijn drie verschillen de compositie types: IOR, XOR en AND. Als je naar de pagina eigenschappen gaat, dan vindt je het "Intentional Element decomposition type". Klik hierop voor meer uitleg.

AND = Deze en een andere

XOR (exclusive-OR) = kiezen tussen het één of het ander, niet beide.

IOR (inclusive-OR) = kiezen tussen het één of het ander, of beide.

7. Hoe haal ik de citaten uit de VNpagina's? (Zie voorbeeld:

https://www.projectenportfolio.nl/wiki/index.php/OLmK_Naasten_-_Zorg_voor_de_kinderen_VN)

Deze zijn in vrijeteksten in de wiki gezet

8. Klopt het gemaakte klassendiagram?

Opmerkingen van Hans de Bruin:

- Het model hoort geen aparte klassen te hebben voor thema's als "Case", "Proces", "HerkenbareSituaties", "ThemaContext", "Organisatie", "ContextPersoon", "Handelingsperspectief" en "tip". Dit zijn geen officiële onderdelen van de EMont.*

Bijlage J. Bronnenonderzoek naar dynamische de EMont structuur binnen een SMW

1. Verband tussen de EMont elementen en de entiteiten binnen de EVM-SMW

Tabel 11 Per Narrative element beschreven hoe de EMont elementen deze vormen.

Narratief element	EMont elementen	bron(nen)
De setting/omgeving	Activity, Actor, Belief, Concept, Condition, Context, Goal, Product	Bijlage H 00:09:50 – 00:16:50
Het thema	[Thema] Context (bijv. OLmK Context) met alle onderliggende casussen.	Bijlage H 00:09:50 – 00:16:50
Het plot	De ondernomen handelingsperspectieven van de bestaande situaties, of de ondernomen handelingsperspectieven door de gebruiker, binnen de mogelijke herkenbare situaties van een thema	Bijlage H 06:00 – 08:40 + 00:09:50 – 00:16:50
De karakters	Actors of Rollen (Contexten)	Bijlage H 06:00 – 08:40
Het perspectief	Situatie (Context) + Worldview (conditions)	Bijlage H 00:46:00 – 00:44:00
De ton	De verandering in het wereldbeeld van de actoren en de situaties aan de hand van de ondernomen handelingsperspectieven.	Bijlage H 02:00 – 04:35
Overige literaire toestellen	- Niet gevonden -	-

2. Statische vastlegging van de narratives

Tabel 12 Schematische weergave van de casus van Claire

Casus Claire			
Bestaande situatie	Herkenbare situatie(s)	Activiteit(en)	Decompositie type
S: Niet pluis, ziekenhuis en weer thuis	OLmK Psychisch en fysiek welzijn - Eenzaamheid VN	OLmK HP Eenzaamheid	IOR
S: Angst voor terugkeer	OLmK Psychisch en fysiek welzijn - Angst voor terugkeer VN	OLmK HP Angst voor terugkeer	IOR
S: Ziekte in combinatie met werk	-	-	-
S: Omgaan met sterfgevallen aan kanker	-	-	-
S: Begrip van omgeving	OLmK Naasten - Er niet over kunnen of durven praten VN	OLmK HP Er niet over kunnen of durven praten	IOR

S: De helft van je genezing is weten wat je zelf wilt.	OLmK Proces - Durven artsen tijdig door te verwijzen? VN	OLmK HP Durven artsen tijdig door te verwijzen?	IOR
S: Toekomstperspectief	OLmK Psychisch en fysiek welzijn - Grens aan aandacht VN	OLmK HP Grens aan aandacht	IOR
S: Hulp accepteren en hulp vragen	-	-	-
S: Verhaal kunnen vertellen	OLmK Psychisch en fysiek welzijn - Grens aan aandacht VN	OLmK HP Grens aan aandacht	IOR

Tabel 13 Schematische weergave van de casus van Aline van Oosten

Casus Aline van Oosten			
Bestaande situatie	Herkenbare situatie(s)	Activiteit(en)	Decompositie type
S: Slecht nieuws	-	-	-
S: Second opinion	OLmK Proces - Inzicht in traject VN		
S: Aline wordt behandeld	-	-	-
S: Steun van naasten	OLmK Naasten - Invloed op relaties VN	-	-
S: Partner en diens kinderen	-	-	-
S: Lotgenotencontact	-	-	-
S: Andere keuzes	-	-	-
S: Werk	-	-	-
S: Genezen en dan?	OLmK Psychisch en fysiek welzijn - Angst voor terugkeer VN	-	-

Tabel 14 Schematische weergave van de casus van Mevrouw Schoonenbrug

Casus Mevrouw Schoonenbrug			
Bestaande situatie	Herkenbare situatie(s)	Activiteit(en)	Decompositie type
S: Niet goed	OLmK Proces - Slecht nieuws gesprek VN	OLmK HP Slechtnieuwsgesprek	IOR
	OLmK Psychisch en fysiek welzijn - Vermoeidheid VN	OLmK HP Vermoeidheid	IOR

	OLmK Psychisch en fysiek welzijn - Niet pluis gevoel VN	OLmK HP Niet-pluisgevoel	IOR
S: Ziekenhuis	OLmK Proces - Inzicht in traject VN	OLmK HP Inzicht in traject	IOR
S: Eigen kamer	OLmK Financiële aspecten - Verzekering en vergoeding VN	OLmK HP Verzekering en vergoeding	IOR
S: Leven met de dag	-	-	-
S: Vertrouwen	OLmK Proces - Wie kun je vertrouwen? VN	OLmK HP Wie kun je vertrouwen?	IOR
S: Kwetsbaarheid	OLmK Psychisch en fysiek welzijn - Kwetsbaarheid en onwetendheid VN	OLmK HP Kwetsbaarheid en onzekerheid	IOR
S: Aandacht	OLmK Naasten - Invloed op relaties VN	OLmK HP Invloed op relaties	IOR
S: Partner en kinderen	OLmK Naasten - Er niet over kunnen of durven praten VN	OLmK HP Er niet over kunnen of durven praten	IOR

Tabel 15 Schematische weergave van de casus van de heer Raaymakers

Casus de heer Raaymakers			
Bestaande situatie	Herkenbare situaties	Activiteit(en)	Decompositie type
S: Diaree	OLmK Proces - Slecht nieuws gesprek VN	OLmK HP Slechtnieuwsgesprek	IOR
S: Eigen regie	OLmK Proces - Inzicht in traject VN	OLmK HP Inzicht in traject	IOR
S: Aanspreekpunt	-	-	-
S: Beren op de weg	OLmK Proces - Wie kun je vertrouwen? VN	OLmK HP Wie kun je vertrouwen?	IOR
S: Ervaringen delen	OLmK Naasten - Er niet over kunnen of durven praten VN	OLmK HP Er niet over kunnen of durven praten	IOR
S: Lotgenoten	-	-	-
S: Ziektegeld	OLmK Financiële aspecten - Verzekering en vergoeding VN	OLmK HP Verzekering en vergoeding	IOR
S: Moe	OLmK Psychisch en fysiek welzijn - Angst voor terugkeer VN	OLmK HP Angst voor terugkeer	IOR

3. Vastlegging van de practices (ervaringen) van de narratives

Casus Claire

Portfolio's > Zeeuwse Huiskamer > DZH Overleven met kanker > DZH Overleven met kanker - resultaten > OLmK Casus Claire

Biografie en sociaal-medisch netwerk

Biografie en sociaal-medisch netwerk[Uitvouwen]

OLmK Casus Claire

Claire is een vrouw van 57. Op veertigjarige leeftijd kreeg ze borstkanker en werd haar borst geamputeerd. Nu zijn er uitzaaiingen in de longen en in de hersenen en kan ze niet meer genezen van deze ziekte.

Samenvatting

Overkoepelende context [DZH Overleven met kanker - resultaten](#)

Situaties

Creër een situatie

Let op, iedere pagina moet een unieke naam hebben. Het advies is de drie puntjes te vervangen door een korte omschrijving van de situatie.

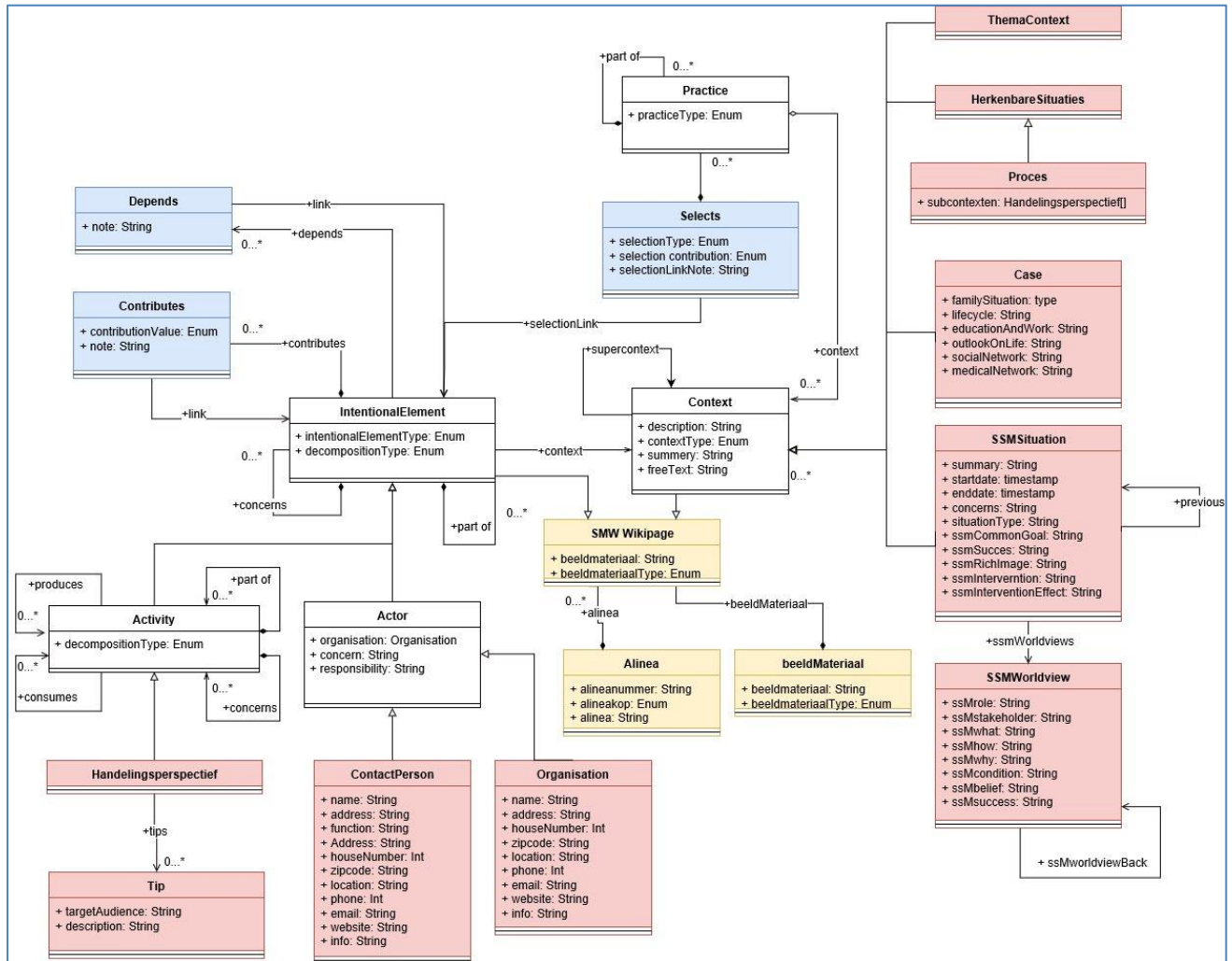
x
Creër een situatie

Bestaande situaties:

	Startdatum	Einddatum	Samenvatting
Niet pluis, ziekenhuis en weer thuis	1 januari 2016	2 januari 2016	Claire voelde eerst al een tijdje steken in haar borst maar ging pas later naar de dokter. Ze had ook lang borstvoeding gegeven en dacht dat het daarmee te maken had.
Angst voor terugkeer	2 januari 2016	3 januari 2016	Claire was erg bang dat het terug zou komen maar langzamerhand lukte het om dat naar de achtergrond te laten schuiven.
Ziekte in combinatie met werk	3 januari 2016	4 januari 2016	Ze wil graag werken, maar op het werk ondervindt ze weinig begrip voor haar situatie.
Omgaan met sterfgevallen aan kanker	4 januari 2016	5 januari 2016	Terwijl Claire zelf ziek is, sterft een vriendin aan kanker.
Begrip van omgeving	5 januari 2016	6 januari 2016	Veel mensen in haar omgeving zijn bang van de ziekte en nemen geen direct contact met haar op.

Figuur 9 Schermopname van de Casus van Claire (instantie van een de EMont Context).

Bijlage K. Statische uitdrukking van de EMont binnen een EVM-SMW met feedback

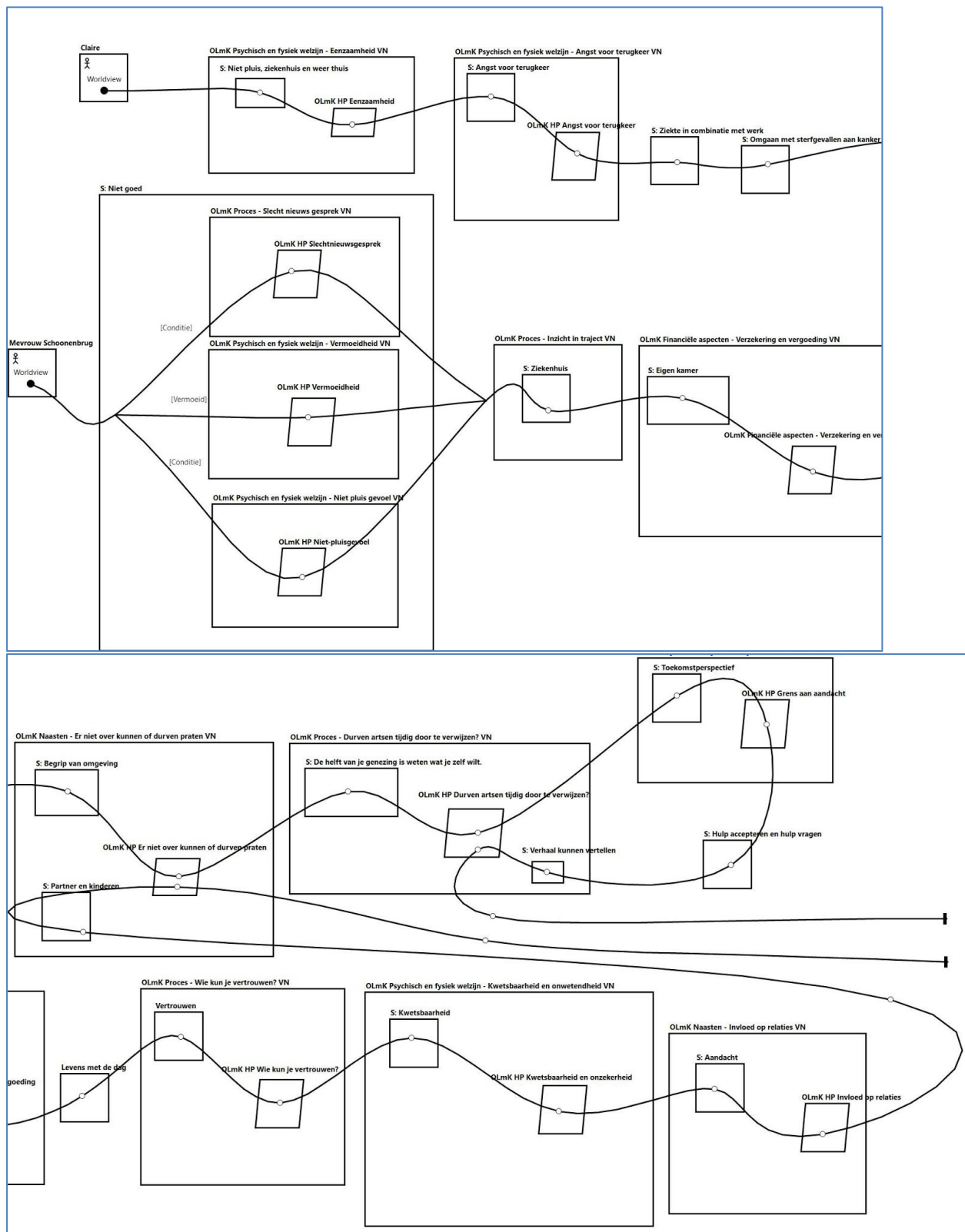


Figuur 10 Klassendiagram van de EMont binnen in de projectenportfolio wiki

Opmerkingen door Hans de Bruin voor bij Figuur 10:

- De EMont basiselementen zijn wit met zwart.
- De **blauwe** entiteiten zijn geen de EMont elementen, maar entiteiten voor het aangeven van relaties tussen de EMont elementen.
- De **gele** entiteiten zijn de basis categorieën, ofwel klassen, waaruit de EMont is gedefinieerd.
- De **Rode** entiteiten zijn de L1 en L2 de EMont elementen. Hiervan is alleen de dynamiek van belangrijk.

Bijlage L. Dynamische uitdrukking van enkele bestaande in de EMont vastgelegde narratives



Figuur 11 Ingezoomde weergaves van het dynamische model voor de narratives (de lijnen geven het verloop aan van de situaties waarin de actor beland)

Bijlage M. Functionele eisen

Door in de codering van het eerste interview (Bijlage H) onderzoek te gaan naar eisen zijn de wensen van de opdrachtgever Hans de Bruin te herleiden. Op basis van deze wensen zijn SMART geformuleerde functionele en technische eisen vastgesteld.

FR 1 – Leveren 3d wereld

Wens:

Het systeem kan op basis van de semantische wikidata een interactieve 3d wereld leveren.

SMART-eis(en):

FR1 - Het systeem kan de [EMont] en [wikidocumentatie] instantiëren binnen de game engine als gevisualiseerd 3d objecten.

EMont:

- De verschillende niveaus van het vastgelegde EMont, bestaande uit contexten, intentional elements.

Wikidocumentatie:

- De wiki documentatie bij de EMont elementen, bestaande uit afbeeldingen, video-audio opnames, websites en teksten.

FR1.1 - Het systeem kan de [EMont] instantiëren binnen de game engine als gevisualiseerd 3d objecten.

FR1.2 - Het systeem kan de [wikidocumentatie] instantiëren binnen de game engine als gevisualiseerd 3d objecten.

Bron(nen):

- Bijlage 2: de EMont en requirements interview (uit het analysedocument) 06:00 – 08:40

FR 2 – Vormen van de wereld

Wens:

De 3d wereld moet vormen naar de ondernomen de EMont activiteiten, situatie (context) en actor of rol condities en eventuele ontwerpers keuzes.

SMART-eis:

FR2 - Het systeem kan de [3d wereld] vormen aan de hand van de gebruikers [interactie] met de EMont activiteiten, de [condities] van de in de wereld bestaande actoren of rollen en de eventuele ontwerpers wereldontwerp keuzes, als deze zijn gevraagd door de gebruiker.

3d wereld:

- De plaatsing en weergave van alle intentional elements en niet de EMont wikidocumentatie in de 3d virtuele ruimte van Unity. Die verwijzen naar het basis de EMont wereld element.

interactie:

Het doorgeven aan het systeem dat de activiteit is gekozen om uit te voeren.

condities:

De Depends, Connects, Selects en Contributes waarde van relaties tussen de intentional elements.

FR2.1 - Het systeem kan de 3d wereld vormen aan de hand van de gebruikers interactie met de EMont activiteiten.

FR2.2 - Het systeem kan de 3d wereld vormen aan de hand van de [condities] van de in de wereld bestaande actoren of rollen.

FR2.3 - Het systeem kan de 3d wereld vormen aan de hand de eventuele ontwerpers wereldontwerp keuzes, als deze zijn gevraagd door de gebruiker.

Bron(nen):

- Bijlage 2: de EMont en requirements interview (uit het analysedocument) 06:00 – 08:40
- Inleiding

FR 3 – Limiteren vrijheidsgraden in activiteit keuze

Wens:

“Karakters kunnen ook ontwikkeld worden, dus naar aanleiding van condities, hij overgaat tot andere acties.”

SMART-eis:

FR3 - Het systeem kan op basis van het [compositie type] van de [relevante intentional elements] en [relevante practices], de [vrijheidsgraden] binnen de huidige context limiteren.

compositie type:

- Een variabele van een intentional element (of subcategory van) die met IOR, XOR of AND aangeeft welke uitgevoerd mag worden. (Uitgelegd in Bijlage 3: Bronnenonderzoek van statische de EMont structuur binnen de wiki (Stap 4 – Validatie en feedback)

relevante intentional elements:

- Alle intentional elements die zijn bestempeld als relevante. Dit is voor activiteiten als deze zijn uitgevoerd, voor geloof (belief) en doel (goal) als deze zijn bepaald en product als deze is gemaakt.

relevante practices:

- De door de gebruiker gekozen practices die gaan over de huidige context. Deze definiëren welke activiteiten goed of slecht zijn binnen de situaties. (Uitgelegd in het Theoretisch kader Expertise Management ontology (EMont) en Concept mapping)

vrijheidsgraden:

- De activiteiten binnen een context. Die zijn toegestaan om uit te voeren.

FR3.1 - Het systeem kan op basis van het compositie type van de relevante intentional elements, de vrijheidsgraden binnen de huidige context limiteren.

FR3.2 - Het systeem kan op basis van relevante practices, de [handelingsperspectieven] als vrijheidsgraden limiteren.

Bron(nen):

- Bijlage 2: de EMont en requirements interview (uit het analysedocument 06:00 – 08:40)

FR 4 – Gebruiker heeft controle over tijd binnen de EMont

Wens(en):

“...De visuele navigatie zal worden toegepast als een “exploratorium”, doordat een speler op eigen tempo door de verhalen heen kan navigeren, door de aparte situaties. Er is geen universele wereld voor de situaties..”

“...hoe snel en hoever een cutscene, video of dialoog is met presenteren.”

SMART-eis:

FR4 - Het systeem geeft de gebruiker een volledige controle de datum, tijd en het tijdsverloop van de interactieve 3d wereld.

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

FR 5 – Gebruiker kan zelf het perspectief kiezen

Wens(en):

“...In een situatie kan de speler in een perspectief van een eerste persoon of god persoon, de handelingen uitvoeren of te aanschouwen.”

SMART-eis:

FR5 - Het systeem kan de gebruiker zelf laten kiezen met welk [perspectief] er naar een situatie wordt gekeken en de [gebruiker interactiemogelijkheid] met wereld elementen aanpassen aan de hand van het gekozen perspectief.

perspectief:

- Actoren of rollen gezien vanuit omniscient
- Actor of rol te gezien vanuit eerste persoon

gebruiker interactiemogelijkheid:

- De gebruiker kan zelf handelen
- De gebruiker kan alleen toeschouwen

FR5.1 - Het systeem kan de gebruiker zelf laten kiezen met welk perspectief er naar een situatie wordt gekeken.

FR5.2 - Het systeem kan de gebruiker interactiemogelijkheid aanpassen aan de hand van het gekozen perspectief.

Bron(nen):

- Bijlage 2: de EMont en requirements interview (uit het analysedocument) 06:00 – 08:40

FR 6 – Inlezen van exports van EVM-SMW's

Wens(en):

““De EVM-SMW de EMont-gegevens die het product kan inlezen zullen komen uit de aangeboden exporteer mogelijkheden van een EVM-SMW”

“Het EVM verlangt naar het inzetten van de interactieve 3d wereld bij alle Living Labs.”.”

SMART-eis:

FR6 - Het systeem kan de EMont uitlezen uit een geëxporteerde formaat van de door [EVM-SMW's] aangeboden exporteer functie.

EVM-SMW's:

- Door het EVM-kenniscentrum opgestelde MediaWiki's die de EMont hebben vastgelegd aan de hand van de Semantic MediaWiki plug-in.

Bron(nen):

- Inleiding

FR 7 – Automatisch bijhouden van progressie van de narratives

Wens(en):

“... Branching gebeurt wel degelijk in de story en het plot, dit is terug te vinden in de wiki.”
“Ook maken de narratives gebruik van de “parallel plot structure”, doordat de verschillende levenslijnen naast elkaar afspelen.”

SMART-eis:

FR7 - Het systeem kan automatisch de [voortgang] van alle narratives bijhouden afhankelijk van de progressie van de gebruiker.

Voortgang:

- De ondernomen acties door de actor en rollen in uit de EMont

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

FR 8 – Het werken van de applicatie voor dat het gebouwd is

Wens(en):

“De ontwerper moet in Unity de wereld kunnen ontwerpen”

SMART-eis:

FR8 - Het systeem kan worden gedraaid binnen de Unity Editor

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

FR 9 – Exporteren van werelden

Wens(en):

“De ontwerper moet in Unity de wereld kunnen ontwerpen”

SMART-eis:

FR9 - Het systeem moet geladen werelden kunnen exporteren in een formaat dat Unity kan bewerken.

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

FR 10 – Inladen van werelden

Wens(en):

“De ontwerper moet in Unity de wereld kunnen ontwerpen”

SMART-eis:

FR10 - Het systeem moet binnen Unity bewerkte wereldexporten kunnen in laden als wereld.

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

FR 11 – De gebruiker kan de condities lezen

Wens(en):

“De ontwerper moet in Unity de wereld kunnen ontwerpen”

SMART-eis:

FR11 – De gebruiker van kan inzien wat de [condities] zijn van de wereld en te volgen actor of rol.

condities:

- Het overzicht aanwezige de EMont elementen in de huidige context en positieve of negatieve gevoelswaarde van de actor of rol, gevormd uit de gevolgde practices.

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

FR 12 – De gebruiker de wereld

Wens(en):

“..de gebruiker dient visueel te worden ondersteund, wanneer deze de mogelijkheid heeft te interacteren met de wereld objecten en aan te geven welke soort interactie mogelijk is.”

SMART-eis:

FR12 – De gebruiker krijgt te zien wanneer [interactie] mogelijk is met wereld objecten en welke soort interactie uitgevoerd kan worden.

Interactie:

- Een handeling binnen de dynamiek van de wereld, die door de gebruiker kan worden uitgevoerd.

FR12.1 – De gebruiker krijgt te zien wanneer interactie mogelijk is met wereld objecten.

FR12.2 – De gebruiker krijgt te zien welke soort interactie uitgevoerd kan worden.

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

FR 13 – De Droste-effect navigatie

Wens(en):

“...aan de hand van een “Droste-effect” navigeren door de situaties in de EMont.”

SMART-eis:

FR13 – Het systeem kan de gebruiker laten navigeren door de verschillende [contexten] en [in context gelegen overige de EMont elementen en wikidata], aan de hand van interacteerbare objecten binnen de 3d wereld of een, ieder moment op te brengen, Droste-effect navigatie.

context:

- Een thema
- Een herkenbare situatie
- Een bekende situatie
- Een rol

in context gelegen overige de EMont elementen en wikidata:

- Intentional elements en worldviews (condities uit de intentional element gekoppelt aan bestaande situaties)
- Wikidocumentatie (De wiki documentatie bij de EMont elementen, bestaande uit afbeeldingen, video-audio opnames, websites en teksten)

Bron(nen):

- Bijlage 9: Validatie en feedback (uitbreiding bijlage Stap 4)

Bijlage N. Technische eisen

ISO-25010 Hoofdcategorieën	ISO-25010 Kwaliteitseigenschap	Metric	Requirement	Bron(nen) / Onderbouwing
Functionele geschiktheid	Functionele compleetheid	Het voldaan hebben van functionele eisen	NFR1 – De applicatie dient minimaal aan FR1 en FR2.1 te voldoen.	Bijlage O: “De applicatie dient minimaal werelden te laden aan de hand van de EMont en door contexten te lopen door het kiezen van handelingsperspectieven.” Randvoorwaarden van het onderzoek: “De EVM-SMW de EMont-gegevens die het product kan inlezen zullen komen uit de aangeboden exporteer mogelijkheden van een EVM-SMW” Inleiding onderzoek: “Het EVM verlangt naar het inzetten van de interactieve 3d wereld bij alle Living Labs.”
“	“	Het aantal te doorlopen narratives met een andere actor of rol	NFR2 – De applicatie kan de narratives van diverse actoren of rollen te kunnen visualiseren in het 3d wereldmodel.	Randvoorwaarden van het onderzoek: “Het product dient de minimaal narratives van diverse actoren of rollen te kunnen visualiseren in het 3d wereldmodel.”
Onderhoudbaarheid	Wijzigbaarheid	De mogelijkheid om een wereld aan te passen in Unity.	NFR3 – De uit de applicatie geëxporteerde werelden dienen te kunnen worden aangepast binnen de Unity Editor.	Bijlage O: “De ontwerper moet in Unity de wereld kunnen ontwerpen”
Bruikbaarheid	Leerbaarheid	Het aantal hulpbronnen dat de gebruiker raadpleegt	NFR4 – De applicatie dient zo te werken dat de gebruiker geen bronnen hoeft raad te plegen buiten het systeem, om aan de hand van activiteiten uitvoer in een nieuwe context te komen.	Bijlage O: “het spel dient te worden begrepen zonder een tutorial te moeten volgen. Door de gebruiker te begeleiden met instructie op basis van zijn/haar doel.”
Betrouwbaarheid	Volwassenheid	De geladen wereld soorten bij het volgen van narratives	NFR5 - De applicatie laadt alleen contexten als wereld basis, als de gebruiker een [narrative volgt] van een actor of rol. narrative volgt:	Bijlage O: “De applicatie dient zich te herstellen bij het laden van niet bedoelde werelden. Dit door het laden van een wel bedoelde

			* activiteiten uitvoert als een actor of rol	wereld van waar een gebruiker weer kan handelen.”
“	Herstelbaarheid	Succesratio van goed geladen werelden	<p>NFR6 – De applicatie dient een 100% van het voorkomen een [fout geladen wereld] te herstellen met een [toegestane wereld].</p> <p>fout geladen wereld:</p> <ul style="list-style-type: none"> • Een wereld waarvan de basis geen context is, wanneer een gebruiker een narratief volgt <p>Goed geladen wereld:</p> <ul style="list-style-type: none"> • Een wereld waarvan de basis een context is, wanneer een gebruiker een narratief volgt 	<p>Bijlage O:</p> <p>“De applicatie dient zich te herstellen bij het laden van niet bedoelde werelden. Dit door het laden van een wel bedoelde wereld van waar een gebruiker weer kan handelen.”</p>
Onderhoudbaarheid	Herbruikbaarheid	Succesvol inladen van in EVM-SMW de EMont data in de klassen van de applicatie.	NFR7 - De klassen dienen 100% van de inlaad pogingen het EVM-SMW RDF-formaat export te kunnen bouwen als objecten binnen de wereld.	<p>Bijlage O:</p> <p>“De klassen van de applicatie dienen te kunnen werken met de verschillende semantische RDF-exporten van EVM-SMW’s”</p>
“	“	Gebruikte taal van de library of framework voor de omzetting van de EMontdata	NFR8 - De applicatie dient volledig te worden geschreven in de programmeertaal C#.	<p>Bijlage O:</p> <p>“Voor de omzetting is handig in eenzelfde taal te programmeren”</p> <p>Randvoorwaarden 6 (alleen programmeren in C#)</p>

Bijlage O. Validatie en feedback (uitbreiding bijlage Stap 4)

Als validatie van het statische en dynamische model van de EMont in een EVM-SMW is er nogmaals een interview gehouden. Dit interview is een combinatie van het interview genoemd in deelvraag 1 en 2. Het interview leverde vooraf een presentatie die Hans de Bruin heeft geïnformeerd over enkele bevinding uit het theoretisch kader van het onderzoek en usability en accessibility ontwerppatronen voor computerspellen. Uit deze presentatie zijn nieuwe eisen ontstaan. Andere vragen gesteld in dit interview zijn genoemd in stap 5 van Bijlage I.

Interview presentatie

Narrative in interactive 3D wereld – Functioneel Adviezen

1

Deel vraag 2+3

2

Keuze Aesthetics

Hunicke, LEBLANC en Zubeck definiëren acht diverse Aesthetics:

1. **Participation** (Game as sense of place): Player experiences something completely unfamiliar.
2. **Participation** (Game as mode of play): Imaginary world.
3. **Participation** (Game as drama): A story that draws the player to keep coming back.
4. **Participation** (Game as electronic novel): Urge to master something. Based on game's replayability.
5. **Participation** (Game as social phenomenon): A community where the player is an active part of it.
6. **Participation** (Game as multiplayer game):
7. **Participation** (Game as uncharted territory): Urge to explore game world.
8. **Participation** (Game as self-discovery): Open-ended. For example, creating character, re-arranging play with own avatar.
9. **Participation** (Game as quest): Connection to the game, as a whole, despite all constraints.

3

Narrative termen

4

Meer vrijheidsgraden	Minder vrijheidsgraden	Meer vrijheidsgraden
Game	Tetris	Computer Chess
Game Moves	Game-play Gestalts	Higher Level Temporal Structure
Game-play Gestalts	Higher Level Temporal Structure	Simulation
Simulation	Simulation	Simulation

5

6

Generative substrate – Simulation

"Een simulatie kan worden gedefinieerd als: een weergave van de functie, werking of eigenschappen van een proces of systeem door het gebruik van een ander." - C. A. Lindley, januari 2005

Simulatie omvat de laagste niveaus van tijdsstructuur, met auteursrechtelijke principes die bepalen hoe de tijd van frame tot frame op basis van fysica, de weergave van spelobjecten en hun gedrag, en discrete gebeurtenissen simulatie ontwikkelen.

In het Generative substrate "Games" creëert simulatie zich uit tot simulatie van cognitieve en emotionele toestanden en gedragingen van spelkarakters.

Voor "Game" wereld kan het simulatiemodel een model zijn van een systeem of wereld die een fictie of fabricage is. Sommige spellichaamen zijn simulaties van echte fysieke systemen.

Dit is niet het niveau van game moves, maar het niveau van de implementatie van de game moves.

7

Generative substrate – Game (1)

"Een spel is een doelgerichte en competitieve activiteit uitgevoerd binnen een kader van overeengekomen regels." - C. A. Lindley, januari 2005

Variante 1: Games Moves

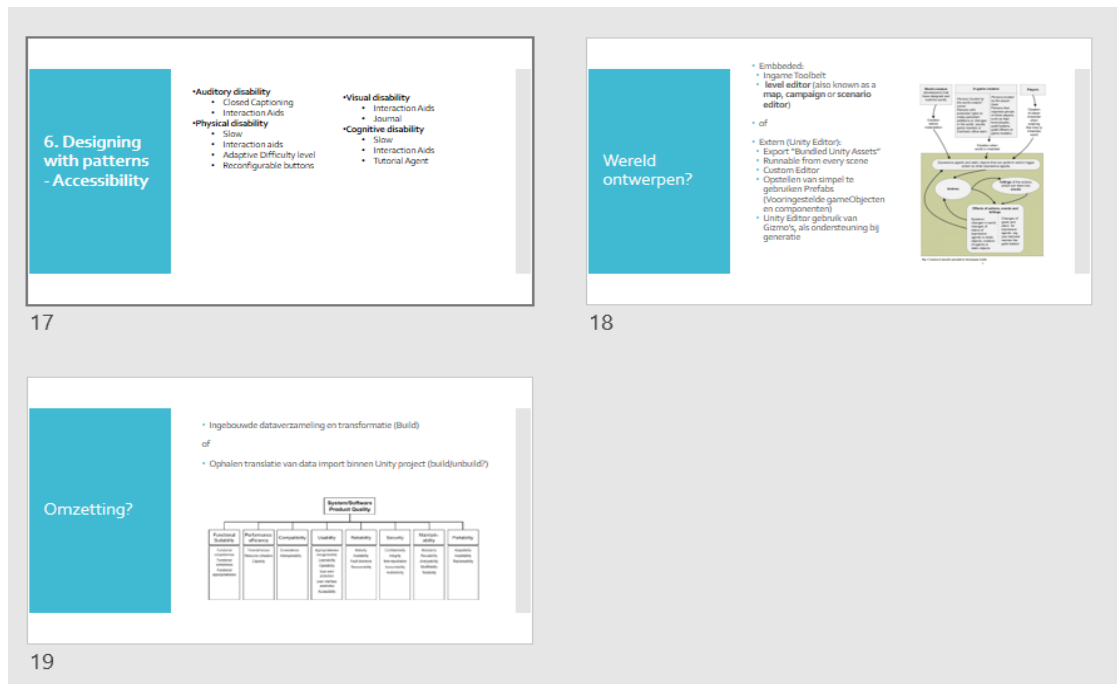
A move within a game is an abstraction over player action, mapping action to a specific significance within the rule set and independent of local, personal and idiosyncratic variations in performance; a move is a conception of a physical action allowed and facilitated by the semantic framing of the game

The definition of game moves represents an intermediate level of temporal design in a game system. Moves represent the player's space of possible in-game significant actions, and it is in the role of a chooser and initiator of moves that the player is constituted as a player.

Variante 2: Game-play Gestalts As Patterns of Moves

Variante 3: Higher Level Temporal Structure in Games

8



Tabel 16 Wensen en eisen uit het tweede interview met Hans de Bruin

Slide	Wens (van de Hans de Bruin)
3	“Natuurlijk moet het overkomen als een narrative. De gebruiker zal zich moeten kunnen plaatsten in het aangeleverde verhaal.”
4	“... branching gebeurt, wel degelijk in de story en het plot, dit is terug te vinden in de wiki.”
5	“Uit de voorbeelden van computerspellen kan ik herleiden dat mijn idee valt onder een multipath movie.”
8	“...de gebruiker kan de handelingen uitvoeren of aanschouwen.”
9	
13	“...De visuele navigatie zal worden toegepast als een “exploratorium”, doordat een speler op eigen tempo door de verhalen heen kan navigeren, door de aparte situaties. Er is geen universele wereld voor de situaties.”
15	“...hoe snel en hoever een cutscene, video of dialoog is met presenteren.”
	“De gegenereerde narratives zullen visueel maken wat er in de wiki pagina van de context staat, zo dat de gebruiker zelf door deze data kan navigeren. Denk aan video’s of foto’s bekijken.”
16	“...Geen tutorial, het moet van zelfsprekend zijn.”
	“...de gebruiker dient visueel te worden ondersteund, wanneer deze de mogelijkheid heeft te interacteren met de wereld objecten en aan te geven welke soort interactie mogelijk is.”
	“De gebruiker heeft de mogelijkheid vrij te navigeren door de wereld, zonder dat deze er invloed op heeft.”
	“De gebruiker moet terug kunnen gaan naar vorige situaties.”
	“De gebruiker moet het spel kunnen pauzeren.”
18	“Voor de omzetting is handig in eenzelfde taal te programmeren”
19	“De ontwerper moet in Unity de wereld kunnen ontwerpen”
	“het spel dient te worden begrepen zonder een tutorial te moeten volgen
	“De applicatie dient zich te herstellen bij het laden van niet bedoelde werelden. Dit door het laden van een wel bedoelde wereld van waar een gebruiker weer kan handelen.”
	“De klassen van de applicatie dienen te kunnen werken met de verschillende semantische RDF-exporten van EVM-SMW’s”
	“De applicatie dient minemaal werelden te laden aan de hand van de EMont en door contexten te lopen door het kiezen van handelingsperspectieven.”

Bijlage P. Functionele en technische eisen (Prioriteiten)

Per diepste niveau requirement is er met Hans de Bruin besproken welke prioriteit deze requirement heeft. De prioritering volgt de MoSCoW-methode. (Bruin, EVM problem explanation, 2017) Volgens Wikipedia bestaat deze uit (MoSCoW-methode, 2017):

M - must haves: deze eisen (requirements) moeten in het eindresultaat terugkomen, zonder deze eisen is het product niet bruikbaar;

S - should haves: deze eisen zijn zeer gewenst, maar zonder is het product wel bruikbaar;

C - could haves: deze eisen zullen alleen aan bod komen als er tijd genoeg is;

W - won't haves: deze eisen zullen in dit project niet aan bod komen maar kunnen in de toekomst, bij een vervolgproject, interessant zijn.

Functionele requirements

Tabel 17 Alle uitgeklaapte versies van de functionele eisen en hun prioritering

Requirement ID	Prioriteit
FR1.1 - Het systeem kan de [EMont] instantiëren binnen de game engine als gevisualiseerd 3d objecten.	Must
FR1.2 - Het systeem kan de [wikidocumentatie] instantiëren binnen de game engine als gevisualiseerd 3d objecten.	Could
FR2.1 - Het systeem kan de 3d wereld vormen aan de hand van de gebruikers interactie met de EMont activiteiten.	Must
FR2.2 - Het systeem kan de 3d wereld vormen aan de hand van de [condities] van de in de wereld bestaande actoren of rollen.	Should
FR2.3 - Het systeem kan de 3d wereld vormen aan de hand de eventuele ontwerpers wereldontwerp keuzes, als deze zijn gevraagd door de gebruiker.	Could
FR3.1 - Het systeem kan op basis van het compositie type van de relevante intentional elements, de vrijheidsgraden binnen de huidige context limiteren.	Should
FR3.2 - Het systeem kan op basis van relevante practices, de [handelingsperspectieven] als vrijheidsgraden limiteren.	Should
FR4 - Het systeem geeft de gebruiker een volledige controle de datum, tijd en het tijdsverloop van de interactieve 3d wereld.	Should
FR5.1 - Het systeem kan de gebruiker zelf laten kiezen met welk perspectief er naar een situatie wordt gekeken.	Should
FR5.2 - Het systeem kan de gebruiker interactiemogelijkheid aanpassen aan de hand van het gekozen perspectief.	Should

FR6 - Het systeem kan de EMont uitlezen uit een geëxporteerd formaat van de door [EVM-SMW's] aangeboden exporteer functie.	Must
FR7 - Het systeem kan automatisch de [voortgang] van alle narratives bijhouden afhankelijk van de progressie van de gebruiker.	Should
FR8 - Het systeem kan worden gedraaid binnen de Unity Editor.	could
FR9 - Het systeem moet geladen werelden kunnen exporteren in een formaat dat Unity kan bewerken.	could
FR10 - Het systeem moet binnen Unity bewerkte wereldexporten kunnen in laden als wereld.	Could
FR11 - De gebruiker van kan inzien wat de [condities] zijn van de wereld en te volgen actor of rol.	Must
FR12.1 - De gebruiker krijgt te zien wanneer interactie mogelijk is met wereld objecten.	Should
FR12.2 - De gebruiker krijgt te zien welke soort interactie uitgevoerd kan worden.	Should
FR13 - Het systeem kan de gebruiker laten navigeren door de verschillende contexten en in context gelegen overige de EMont elementen en wikidata, aan de hand van interacteerbare objecten binnen de 3d wereld of een, ieder moment op te brengen, Droste-effect navigatie.	Could

Niet-Funtionele requirements (technische requirements)

Tabel 18 Alle niet-functionele eisen en hun prioritering

Requirement ID	Prioriteit
NFR1 – De applicatie dient minimaal aan FR1 en FR2.1 te voldoen.	Could
NFR2 – De applicatie kan de narratives van diverse actoren of rollen te kunnen visualiseren in het 3d wereldmodel.	Must
NFR3 – De uit de applicatie geëxporteerde werelden dienen te kunnen worden aangepast binnen de Unity Editor.	Should
NFR4 – De applicatie dient zo te werken dat de gebruiker geen bronnen hoeft raad te plegen buiten het systeem, om aan de hand van activiteiten uitvoer in een nieuwe context te komen.	Could
NFR5 - De applicatie laadt alleen contexten als wereld basis, als de gebruiker een [narrative volgt] van een actor of rol.	Should
NFR6 – De applicatie dient een 100% van het voorkomen een [fout geladen wereld] te herstellen met een [toegestane wereld].	Should
NFR7 - De klassen dienen 100% van de inlaad pogingen het EVM-SMW RDF-formaat export te kunnen bouwen als objecten binnen de wereld.	Should

Bijlage Q. Adviesrapport

Functionele & technische Practices voor narratives in een interactieve 3d wereld in Unity

In dit hoofdstuk zal worden uitgezocht welke functionele en technische implementatie er vereist is om in een (3d) wereld narratives te vertellen. Dit door uit te zoeken wat functioneel en technisch mogelijk is, welke ontwerpen voldoen aan de gewenste eisen.

Het te ontwikkelen product is een spel. Voor het ontwikkelen/ontwerpen van spellen is er een standaard model bedacht, het Mechanics-Dynamics-Aesthetics (MDA). Dit model deelt een spel op in drie verschillende elementen, Mechanics (technische componenten), de dynamics (de functionele componenten; interacterbaarheid of wel dynamiek) en de Aesthetics (het gecreëerde gevoel). Bij het ontwerpen van een spel wordt er eerst gedefinieerd wat de gewenste Aesthetics zullen zijn. Vervolgens zal er worden gekeken welke Dynamics hiervoor gebruikt zullen worden. Als laatst wordt er gekeken welke Mechanics de Dynamics nodig hebben. Voor het testen en ontwikkelen van het spel wordt het model tegenovergesteld afgelopen ten aanzien van het ontwerpen (, dus van Mechanics naar Dynamics naar Aesthetics).

De Aesthetics uit het MDA wordt gevormd door de Mechanics en Dynamics. Er zal eerst moeten worden bepaald welke Aesthetics bereikt wilt worden, dit voor het specifieke speler type. Vervolgens zal er worden gekeken welke narratieve Dynamics er bekend zijn, die bijdragen aan het product van deze casus. Als laatst zal er ook worden gekeken welke Mechanics bijdragen aan de Dynamics en het creëren van de narratieve elementen in het algemeen.

1. Keuze voor Aesthetics

Het uiteindelijke product richt op het overbrengen van een waargebeurd verhaal dat de gebruiker stimuleert met volle interesse het verhaal te doorlopen om te zien hoe het verliep en waar het toe heeft geleid, dit om begrip te kweken. Dit valt onder de Aesthetic: 'narrative' (oftewel drama). In Figuur 12 zijn de Aesthetics weergegeven.

Hunicke, LeBlanc en Zubec definiëren acht diverse Aesthetics:

1. Sensatie (Spel als zintuiglijk genot): Speler ervaart iets dat totaal onbekend is.
2. Fantasy (Spel om te laten geloven): denkbeeldige wereld.
3. **Narrative (Spel als drama): een verhaal dat de speler ertoe brengt terug te blijven komen.**
4. Challenge (Spel als hindernisbaan): drang om iets te beheersen. Vergroot de herspeelbaarheid van een game.
5. Fellowship (Game als sociaal raamwerk): een community waar de speler actief deel van uitmaakt.
6. Discovery (Spel als niet in kaart gebracht territorium): drang om de gamewereld te verkennen.
7. Expression (Spel als zelfontdekking): Eigen creativiteit. Bijvoorbeeld, het creëren van karakter dat lijkt op de eigen avatar van de speler.
8. Submission (Game as pastime): Verbinding met het spel als geheel, ondanks beperkingen.

Figuur 12 De acht verschillende Aesthetics (Abbott t. , 2010) (Bruin, Feedback en prioritering van opgestelde eisen, 2017):

2. Verzamenen Mechanics basis (Generative substrate, model en design patterns)

2.1 Keuze generative substrate

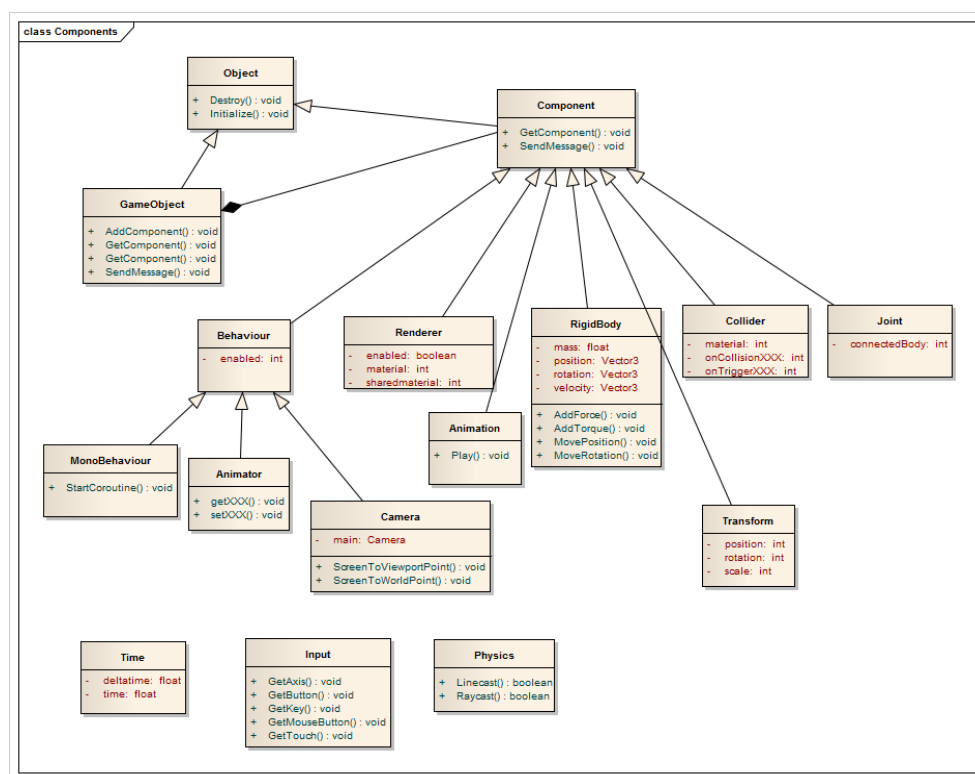
Project randvoorwaarden: Unity – game engine

De Unity game engine is het systeem waarin de 3d wereld in opgezet en ingesteld zal worden. Dit is een cross-platform C# en Unity Script (javascript) game engine ontwikkeld door het bedrijf Unity Technologies. Het combineert een verschillende technologieën afgedekt onder een gebruiksvriendelijk interface voor makkelijke definiëren en van het 3d/2D interactieve applicatie, aan de hand van de beschikbaar gestelde Unity API.

De Unity Editor, het door Unity aangeboden digitale 2D interface waarin in ontwerpers een 3d wereld kunnen ontwerper. De 3d werelden zijn Scenes, een object binnen Unity dat met behulp van GameObjects een wereld vormt. GameObjects kunnen alles zijn binnen de wereld afhankelijk van de ingestelde Componenten. Componenten zijn objecten die eigenschappen definiëren voor het toegewezen game object. Ieder component is een C# of Unity script (javascript) script.

2.2 KEUZE MODEL (UNITY + EMONT)

Met Unity als generative substrate wordt er gebruik gemaakt van de Unity model/simulatie basis voor het virtualiseren van data. Dit model/simulatie bestaat uit een Scene met daarin GameObjecten met toegewezen componenten, die Unity API aanspreken (Figuur 13). (Xuanyi, 2014)



Figuur 13 Algemene structuur binnen Unity, aangeboden door de Unity API (Xuanyi, 2014)

Uit het analysedocument is gebleken dat de dynamiek van de EMont narratives wordt gedefinieerd aan de hand van drie verschillende dynamische structuren, deze kennen standaard statische structuren voor het vastleggen in

een applicatie, zie tabel hieronder. Deze statische structuren zijn onderdeel van het model van de applicatie en zijn te tonen in een UML-diagram (Figuur 11 van Bijlage I)

Tabel 19 verhaalstructuren uit “Story and Narrative Structures in Computer Games” door Craig A. Lindley. (Lindley, 2005)

Dynamische structuur	Statische structuur
Exploratorium	<p>Een exploratorium is een sequentie met knooppunten die sets van vertellende componenten zijn; Als de toegankelijkheid van componenten binnen elke knooppunt moet worden gemodelleerd als een grafiek, dan is het een volledige grafiek, dwz. Elk object binnen een knooppunt is verbonden met elk ander object binnen dat knooppunt.</p> <p>Terug te vinden in als klasse “Context” binnen het de statische structuur van de implementatie van de EVM-SMW, zie figuur hieronder.</p>
Parallele plotstructuur	<p>Een parallele plotstructuur bestaat uit parallele sequenties met koppelingen tussen die sequenties, die een die een graph vormen</p> <p>Terug te vinden in als klasse “SSM Situatie”, zie figuur hieronder.</p>
Een open structuur geassocieerd met verschillende plaatsen	<p>Een open structuur misschien ofwel hetzelfde als een exploratorium, behalve dat de structuur op hoog niveau een vertakking grafiek is, in plaats van een sequentie, of een volledige grafiek.</p>

2.3 UNITY BEST PRACTICE TECHNISCHE DESIGN PATTERNS

2.3.1 Best practice: Singleton Manager

29. Use singletons for convenience. The following class will make any class that inherits from it a singleton automatically:

```
public class Singleton<T> : MonoBehaviour where T : MonoBehaviour
{
    protected static T instance;

    /**
     * Returns the instance of this singleton.
     */
    public static T Instance
    {
        get
        {
            if(instance == null)
            {
                instance = (T) FindObjectOfType(typeof(T));

                if (instance == null)
                {
                    Debug.LogError("An instance of " + typeof(T) +
                        " is needed in the scene, but there is none.");
                }
            }

            return instance;
        }
    }
}
```

Singletons are useful for managers, such as `ParticleManager` or `AudioManager` or `GUIManager`.

- Avoid using singletons for unique instances of prefabs that are not managers (such as the Player). Not adhering to this principle complicates inheritance hierarchies, and makes certain types of changes harder. Rather keep references to these in your `GameManager` (or other suitable God class 😊)
- Define static properties and methods for public variables and methods that are used often from outside the class. This allows you to write `GameManager.Player` instead of `GameManager.Instance.player`.

Figuur 14 Screenshot van de uitleg over de practice "Singleton Manager" in Unity (Tulleken, 50 Tips for Working with Unity (Best Practices), 2012)

2.3.2 Best practice: Toepassen van MVC als scheiding van data en model

31. Separate interface from game logic. This is essentially the MVC pattern.

Any input controller should only give commands to the appropriate components to let them know the controller has been invoked. For example in controller logic, the controller could decide which commands to give based on the player state. But this is bad (for example, it will lead to duplicate logic if more controllers are added). Instead, the Player object should be notified of the intent of moving forward, and then based on the current state (slowed or stunned, for example) set the speed and update the player facing direction. Controllers should only do things that relate to their own state (the controller does not change state if the player changes state; therefore, the controller should not know of the player state at all). Another example is the changing of weapons. The right way to do it is with a method on Player `SwitchWeapon(Weapon newWeapon)`, which the GUI can call. The GUI should not manipulate transforms and parents and all that stuff.

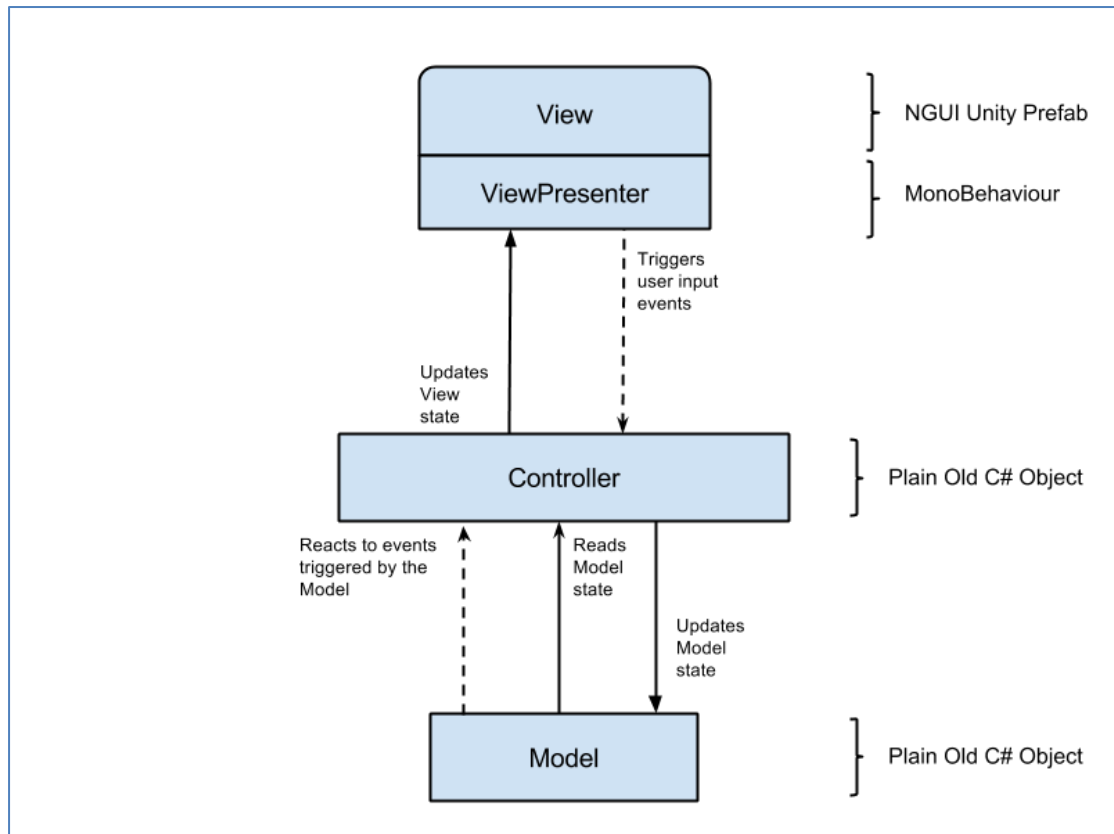
Any interface component should only maintain data and do processing related to its own state. For example, do display a map, the GUI could compute what to display based on the player's movements. However, this is game state data, and does not belong in the GUI. The GUI should merely display game state data, which should be maintained elsewhere. The map data should be maintained elsewhere (in the `GameManager`, for example).

Gameplay objects should know virtually nothing of the GUI. The one exception is the pause behaviour, which is may be controlled globally through `Time.timeScale` (which is not a good idea as well... see). Gameplay objects should know if the game is paused. But that is all. Therefore, no links to GUI components from gameplay objects.

In general, if you delete all the GUI classes, the game should still compile.

You should also be able to re-implement the GUI and input without needing to write any new game logic.

Figuur 15 Screenshot van de uitleg over het toepassen van MVC in Unity (Tulleken, 50 Tips for Working with Unity (Best Practices), 2012)



Figuur 16 Schermopname van de uitleg over het toepassen van MVC in Unity (Amores, 2014)

```

using System;

using UnityEngine;

// Dispatched when the enemy's position changes
public class EnemyPositionChangedEventArgs : EventArgs
{
}

// Interface for the model
public interface IEnemyModel
{
    // Dispatched when the position changes
    event EventHandler<EnemyPositionChangedEventArgs> OnPositionChanged;

    // Position of the enemy
    Vector3 Position { get; set; }
}

// Implementation of the enemy model interface
public class EnemyModel : IEnemyModel
{
    // Backing field for the enemy's position
    private Vector3 position;

    public event EventHandler<EnemyPositionChangedEventArgs> OnPositionChanged = (sender, e) => {};
  
```

```

public Vector3 Position
{
    get { return position; }
    set
    {
        // Only if the position changes
        if (position != value)
        {
            // Set new position
            position = value;

            // Dispatch the 'position changed' event
            var eventArgs = new EnemyPositionChangedEventArgs();
            OnPositionChanged(this, eventArgs);
        }
    }
}

```

Figuur 17 C# Code voorbeeld beschrijvende de opzet van een model. Deze beschrijving toont een klasse met een “backingfield” en toegewezen compact genoteerde assessors (get & set) en het te verzenden event bij aanpassing van het backingfield via de assessor methode. (Dunstan, 2015)

```

using System;

using UnityEngine;

// Dispatched when the enemy is clicked
public class EnemyClickedEventArgs : EventArgs
{
}

// Interface for the enemy view
public interface IEnemyView
{
    // Dispatched when the enemy is clicked
    event EventHandler<EnemyClickedEventArgs> OnClicked;

    // Set the enemy's position
    Vector3 Position { set; }
}

// Implementation of the enemy view
public class EnemyView : MonoBehaviour, IEnemyView
{
    // Dispatched when the enemy is clicked
    public event EventHandler<EnemyClickedEventArgs> OnClicked = (sender, e) => {};

    // Set the enemy's position
    public Vector3 Position { set { transform.position = value; } }

    void Update()
    {
        // If the primary mouse button was pressed this frame
        if (Input.GetMouseButtonDown(0))
        {

```

```

        // If the mouse hit this enemy
        var ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        RaycastHit hit;
        if (
            Physics.Raycast(ray, out hit)
            && hit.transform == transform
        )
        {
            // Dispatch the 'on clicked' event
            var eventArgs = new EnemyClickedEventArgs();
            OnClicked(this, eventArgs);
        }
    }
}

```

Figuur 18 C# Code voorbeeld beschrijvende de opzet van een view. Deze beschrijving toont een klasse met een “backingfield” en toegewezen compact genoteerde assessors (get & set) en het te verzenden event bij aanpassing van het backingfield via de assessor methode. (Dunstan, 2015)

```

using UnityEngine;

// Interface for the enemy controller
public interface IEnemyController
{
}

// Implementation of the enemy controller
public class EnemyController : IEnemyController
{
    // Keep references to the model and view
    private readonly IEnemyModel model;
    private readonly IEnemyView view;

    // Controller depends on interfaces for the model and view
    public EnemyController(IEnemyModel model, IEnemyView view)
    {
        this.model = model;
        this.view = view;

        // Listen to input from the view
        view.OnClicked += HandleClicked;

        // Listen to changes in the model
        model.OnPositionChanged += HandlePositionChanged;

        // Set the view's initial state by syncing with the model
        SyncPosition();
    }

    // Called when the view is clicked
    private void HandleClicked(object sender, EnemyClickedEventArgs e)
    {
        // Do something to the model
        // This example just moves the model +1 along the X axis
    }
}

```

```

        model.Position += new Vector3(1, 0, 0);
    }

    // Called when the model's position changes
    private void HandlePositionChanged(object sender, EnemyPositionChangedEventArgs e)
    {
        // Update the view with the new position
        SyncPosition();
    }

    // Sync the view's position with the model's position
    private void SyncPosition()
    {
        view.Position = model.Position;
    }
}

```

Figuur 19 C# Code voorbeeld beschrijvende de opzet van een controller. Deze beschrijving toont een klasse met een methode voor het afvangen van de view en model events (bijv. "HandlePositionChanged()") en de methode die de logica van de afhandelingmethode kent (bijv. "SyncPosition()"). (Dunstan, 2015)

2.3.3 Best practice: onderscheid tussen de state van het object en de savedata

32. Separate state and bookkeeping. Bookkeeping variables are used for speed or convenience, and can be recovered from the state. By separating these, you make it easier to

- save the game state, and
- debug the game state.

One way to do it is to define a `SaveData` class for each game logic class. The

```

[System.Serializable]
PlayerSaveData
{
    public float health; //public for serialisation, not exposed in inspector
}

Player
{
    //... bookkeeping variables

    //Don't expose state in inspector. State is not tweakable.
    private PlayerSaveData playerSaveData;
}

```

Figuur 20 Screenshot van de uitleg over het onderscheiden van state en savedata in Unity (Tulleken, 50 Tips for Working with Unity (Best Practices), 2012)

2.3.4 Best practice: Toepassen state machines voor het gemakkelijk afhandelen van acties in verschillende situaties

38. Use state machines to get different behavior in different states or to execute code on state transitions. A light-weight state machine has a number of states, and for each state allows you to specify actions to run when entering or existing the state, and an update action. This can make code cleaner and less error prone. A good sign that you could benefit from a state machine is if your Update method's code has an if- or switch-statement that changes what it does, or variables such as `hasShownGameOverMessage`.

```
public void Update()
{
    if(health <= 0)
    {
        if(!hasShownGameOverMessage)
        {
            ShowGameOverMessage();
            hasShownGameOverMessage = true; //Respawning resets this to false
        }
    }
    else
    {
        HandleInput();
    }
}
```

With more states, this type of code can become very messy; a state machine can make it a lot cleaner.

Figuur 21 Schermopname van de uitleg over het onderscheiden van state en savedata in Unity (Tulleken, 50 Tips and Best Practices for Unity (2016 Edition), 2016)

```
enum CarFSM
{
    Forward,
    Brake,
    Reverse,
    Park
}
//Default state
CarFSM carMode = CarFSM.Park;

//And then you change the state with something like
if (carMode == CarFSM.Park && IsPressingGasPedal())
{
    carMode = CarFSM.Forward;
}
else if (carMode == CarFSM.Forward && IsPressingBrakePedal())
{
    carMode == CarFSM.Brake;
}
//...and so on
```

Figuur 22 Schermopname van de uitleg over het onderscheiden van state en savedata in Unity (Game programming patterns in Unity with C#, 2017)

2.3.5 Best practice: Toepassen observer bij veel conditie veranderingen binnen de wereld

40. Use the observer pattern to detect when a field value changes. The problem of executing code only when a variable changes crops up frequently in games. We have baked a general solution of this in a generic class that allows you to register for events whenever the value changes. Here is an example with health. Here is how it is constructed:

```
/*ObservedValue*/ health = new ObservedValue(100);  
health.OnValueChanged += () => { if(health.Value <= 0) Die(); };
```

You can now change it everywhere, without doing checking in each place where you check it, for example, like this:

```
if(hit) health.Value -= 10;
```

Whenever the health hits a point below 0, the Die method is called. For further discussion and an implementation, see this [post](#).

Figuur 23 Screenshot van de uitleg over het onderscheiden van state en savedata in Unity (Tulleken, 50 Tips and Best Practices for Unity (2016 Edition), 2016)

2.3.6 Best practice: Factory voor het model, de view en de controller van het MVC

“This takes care of the three main pieces of the pattern, but it’s often a good idea to create factory or builder classes to build one or all of the parts. Here are some example factories:”

```
// Interface for the model factory  
public interface IEnemyModelFactory  
{  
    // Get the created model  
    IEnemyModel Model { get; }  
}  
  
// Implementation of the model factory  
public class EnemyModelFactory : IEnemyModelFactory  
{  
    public IEnemyModel Model { get; private set; }  
  
    // Create the model  
    public EnemyModelFactory()  
    {  
        Model = new EnemyModel();  
    }  
}
```

Figuur 24 Code voorbeeld van een Factory voor het maken van model voor het MVC binnen Unity (Dunstan, 2015)

```

// Interface for the view factory
public interface IEnemyViewFactory
{
    // Get the created view
    IEnemyView View { get; }
}

// Implementation of the view factory
public class EnemyViewFactory : IEnemyViewFactory
{
    public IEnemyView View { get; private set; }

    // Create the view
    public EnemyViewFactory()
    {
        var prefab = Resources.Load<GameObject>("Enemy");
        var instance = UnityEngine.Object.Instantiate(prefab);
        View = instance.GetComponent<IEnemyView>();
    }
}

```

Figuur 25 Code voorbeeld van een Factory voor het maken van views voor het MVC binnen Unity (Dunstan, 2015)

```

// Interface of the view factory
public interface IEnemyControllerFactory
{
    // Get the created controller
    IEnemyController Controller { get; }
}

// Implementation of the controller factory
public class EnemyControllerFactory : IEnemyControllerFactory
{
    public IEnemyController Controller { get; private set; }

    // Create just the controller
    public EnemyControllerFactory(IEnemyModel model, IEnemyView view)
    {
        Controller = new EnemyController(model, view);
    }

    // Create the model, view, and controller
    public EnemyControllerFactory()
        : this(new EnemyModel(), new EnemyView())
    {
    }
}

```

Figuur 26 Code voorbeeld van een Factory voor het maken van views voor het MVC binnen Unity (Dunstan, 2015)

2.4 (Extra) Mechanic dynamisch laden van ontworpen werelden

In de randvoorwaarden van het onderzoek wordt er gesproken van een mogelijkheid om de interactieve 3d wereld aan te passen. In het artikel van "Story Construction and Expressive Agents in Virtual Game Worlds" wordt er gesproken over twee verschillende manieren voor het ontwerper van werelden een World creator en een In-game World creator. De technische relevante wereld bewerkmogelijkheden zullen worden opgezocht. Een bewerkmogelijkheid is pas relevant als er een eis over is gesteld.

Tabel 20 Mogelijke technische implementatie van een mogelijkheid om werelden te bewerken

Editortype	Technisch	Eis
World creator	Exporteren van Unity Scenes met gebruikte projectbestanden in assetbundles.	FR10

(in Unity Editor)	<p>“Asset Bundles are a collection of assets, packaged for loading at runtime. With Asset Bundles, you can dynamically load and unload new content into your application. AssetBundles can be used to implement post-release DLC.</p> <p>They can be used to reduce the amount of space on disk used by your game, when first deployed. It can also be used to add new content to an already published game.” - (Unity, 2017)</p> <p>De scenes kunnen binnen de Unity applicatie worden ingeladen met behulp van de SceneManager. (Unity, 2017) De assetbundles kunnen worden gebruikt om de scenes in te laden in de Unity applicatie. (Unity, 2017)</p>	
In-game World creator (in interactive 3d wereld)	-Niet relevant-	

2.5 Samenvatting van de Unity practices in eigenwoorden

Tabel 21 Unity practices in eigenwoorden

Impact op het basismodel	Relevante eisen
<p>Singleton (“Manager”) component: Een object in de Unity Scene met daaraan een Singleton component dat een specifiek onderdeel regelt van de applicatie.</p>	N.v.t
<p>Observer: Het definiëren van “subject” objecten die verandering in waardes door geeft aan een “observer” object, door middel van events. Het observer object kan reageren op de waarde aangegeven door het verstuurd event van de subject verandering. Subjects kunnen zelf bepalen of deze gekoppeld zijn aan een observer, of willen ontkoppelen.</p>	N.v.t
<p>MVC: Het vervangen van de controller componenten aan een object in de Unity Scene door een view component, een controller object en een model object. De view kent geen waardes. De view stuurt alle events door naar de bijbehorende controller. De controller handelt alle Monodevelop-events af en regelt update de waardes van het model en de componenten het view object. Het model is een object dat alle waardes van de view vasthoudt.</p> <p>De attributen die de view vereist te hebben zal bestaan uit een “backed” variabele met een ophaal- (Getter) en instelmethode (Setter). De backed variabele is de variabele zoals deze is opgeslagen en kan alleen worden aangepast via de Getter en Setter methode. De Setter methode verstuurt een event naar de gekoppelde controller als deze de backed variabele heeft aangepast.</p>	NF3
<p>State machines: Het vooraf definiëren van de staten van een object, zodat enkel een verandering van staat genoeg is om het object anders te laten werken. De staten worden aangeleverd in een ENUM variabele.</p>	N.v.t
<p>Factory (voor ieder gebruikt MVC-component): Een specifieke klasse voor het creëren van de MVC-componenten. Factories houden bij welke objecten deze heeft gemaakt.</p>	N.v.t

3. Verzamelen Mechanics uit Dynamics

3.1 Verzamelen toepassingen voor Dynamics (Het plot)

3.1.1 Plot aan de hand van "Game"

3.1.1.2 Mogelijke dynamiek met "Game Moves" tijdstructuur

Tabel 22 Mogelijke dynamiek en technische implementatie met "Game Moves" tijdstructuur

Type Story-oriented player	Gameplay	Technische implementatie	Functionele eisen
The audience:	Onderneemt acties voor het lezen van narratief materiaal in het spel, zoals het openen en lezen van virtuele boeken waarin het spelscenario en het fictieve spel wereldgeschiedenis materiaal wordt gepresenteerd, en beweegt voor conversieve interactie met NPC's, die allemaal bij RPG's gebruikelijk zijn. (cutsenes, zijn niet in te bewegen.)	Er is geen technische implementatie die specifiek behoort tot deze spelertypes. Wel zijn er algemene practices gevonden die van toepassing zijn op al deze spelertypes: - Een bestuurbaar karakters beschrijven door individuele GameObject te definiëren in Unity met ieder een eigen "controller" component, voor het afhandelen van de input van de gebruiker. (Unity, 2017) - Interactie met andere objecten kan worden registreert met behulp van de executie volgorde van de Unity script eventfuncties. (Unity, 2017)	FR2, FR3, FR4, FR5
The performer:	Bewegingen die verband houden met de uitdrukking van vooraf gedefinieerde karakterrollen voor spelers karakters. Verplaatsingen voor de selectie (of niet) van in-game quests kunnen ook als prestatiegerichte bewegingen worden beschouwd.	- GameObject met Singleton "GameManager" component dat de doelen van het spel bij houdt. (Unity, 2017)	FR2, FR3, FR5, FR7
The immersionist:	Het is minder duidelijk welke specifieke functies en bewegingen een speler ervaring ondersteunen die zich verdiept van de prestatie van een karakter tot de ontwikkeling van een persona. Eladhari suggereert dat het ontwijken van een persona vrijheid vereist om een karakter te personaliseren; In de huidige context stelt dit voor dat er meer vrijheid is dan nodig is voor prestatiegericht spel, hoewel onderdompeling ook een lange aanwezigheid in de spelwereld vereist door de speler.		-Niet van toepassing-

3.1.1.3 Mogelijke dynamiek met "Gestalts As Patterns of Moves" tijdstructuur

Tabel 23 Mogelijke dynamiek met technische implementatie met “Gestalts As Patterns of Moves” tijdstructuur

Toepassing	Gameplay	Technische implementatie	Functionele eis(en)
Binnen het genre “Strategie Games”	Bestel boeren, stuur naar het werk, bestel soldaten, stuur naar perimeters, herhaal terwijl de perimeters langzaam worden uitgebreid (tot op het punt van catastrofale win/lose);	GameObject met Singleton “GameManager” component dat de doelen van het spel bij houdt. (Unity, 2017)	-Niet van toepassing-
In Algemeen:	Confronteer barrière, sla de barrière over, herlaad en probeer het als mislukt		-Niet van toepassing-

3.1.1.4 Mogelijke dynamiek met “Higher level temporal structure” tijdstructuur

Tabel 24 Mogelijke dynamiek met technische implementatie met “Higher level temporal structure” tijdstructuur

Toepassing	Gameplay	Technische implementatie	Functionele eis(en)
“principle of accrual of results”	Accurate eindstand	GameObject met Singleton “GameManager” component dat de doelen van het spel bij houdt. (Unity, 2017)	FR11
“elimination events”	Eliminatie van tegemoetkomingen		FR2, FR3
“principles of handicap”	Het aanpassen van de handicap		FR5

3.1.1.4 Mogelijke dynamiek met “Higher level temporal structure” tijdstructuur

3.2 Verzamelen toepassingen voor Dynamics (Narration)

3.2.1 Verzamelen Interface elementen voor Dynamics

Tabel 25 Interface elementen voor Dynamics met technische implementatie

Userinterface design	Verskillende toepassingen	Mechanics Unity
Diëtetisch (video, audio of haptisch met zichtbare bron)	Binnen de wereld geplaatste gereedschap	De aanwezigheid van een Unity GameObject met de componenten (Unity, 2017): <ul style="list-style-type: none"> • <u>Transform</u> (werelds positie, rotatie en grote) • Meshrenderer (visueel maken) • Scripts (voor dynamiek van het object) • (Optioneel) Andere componenten voor andere eigenschappen

	Highlighting van wereld elementen	Aan de hand van een script het toepassen van een shader (effect op de textuur) van een Materiaal van een Unity GameObject met minimaal een Meshrenderer en Transform component. (Weissman, 2016)
Meta	Filters op camera	Een Unity GameObject met een Transform en Camera component, waaraan een script wordt toegevoegd voor het bewerken van de opnames van de camera. (Unity, 2017)
	Metaforische 2D GUI-elementen, representerende binnen de wereld bevindende 3d object interactie	-Geen gevonden implementaties in Unity-
Ruimtelijk	Projectie van GUI-elementen op in wereldse objecten	Het toevoegen van een projector component op het op te projecteren GameObject. (Unity, 2017)
	Een GUI-element dat is geplaatst op de X-, Y- en Z-as van de wereld	Een Unity GameObject met componenten gemaakt van de standaard Canvas Prefab, deze bij het "Render mode" variabel ingesteld als "world space". (Unity, 2017)
Niet-diëtetisch (video, audio of haptisch met niet-zichtbare bron)	Menus	Een Unity GameObject met componenten gemaakt van de standaard EventSystem Prefab. (Deze dient gebruikers input te kunnen lezen) Een verzameling van Unity GameObjecten, met een Graphical Raycaster script component (deze laten) (Borrel, 2015)

3.2.2 Verzamelen narrative descriptors van Dynamics

Tabel 26 Narrative descriptors en bij behorende GUI-onderdelen

Narrative descriptor	Relevante user interface design	Functionele eis(en)
Tekstuele instructies	Alle	FR11, FR12
Korte (cut scenes) en lange filmfragmenten	Alle	FR5
Interface elementen (als besturing en condities)	Niet-diëtetisch	FR5, FR11, FR13
Visuele elementen dat het gezichtsveld van de speler omvat (voor het vormen van perspectieven)	Ruimtelijk	FR1, FR2
Interactie tussen de elementen (dynamiek/gameplay)	Alle	FR2, FR4, FR12

3.3 Verzamelen gewenste usability en accessibility design patterns met Mechanics

Tabel 27 Technische implementatie van de relevante Usability en Accessibility patronen (op basis van de functionele en technische eisen)

ID	Ontwikkelpatroon	Relevante eis	Technische implementatie
1	Seamless Gameworld	N.v.t	-
2	Skippable Cutscenes	FR4, FR5	Scene tijd manipuleren met Time.timeScale van de Unity API. (Unity, 2017)
3	Fast Forward	FR4	Scene tijd manipuleren met Time.timeScale van de Unity API. (Unity, 2017)
4	Quick Save/Load	N.v.t	-
6	Quick Start or Restart	N.v.t	-
7	Pre Loader Game	N.v.t	-
8	Arcade Mode	N.v.t	-
9	Slow	FR4	Scene tijd manipuleren met Time.timeScale van de Unity API. (Unity, 2017)
10	Rewind	FR4	<p>“Save state replay system”: het registreren van de oorspronkelijke toestand van de spelsystemen op het eerste frame en alleen de speler invoer tijdens de gameplay.</p> <p>Design pattern(s):</p> <ul style="list-style-type: none"> • Singleton Manager component (GamePlayManager) <p>(Wagner, 2004) (Montville, 2014)</p>
11	Auto save	FR7	<p>Automatisch opslaan van in het geheugen bevonden toestanden van objecten.</p> <p>Design pattern(s):</p> <ul style="list-style-type: none"> • Update Method (per GameObject) • Een gescheiden klasse voor enkel de data van een object • Gebruik van System.IO.FileStream() voor het open en bewerken van bestanden. • Gebruik van System.Runtime.Serialization.Formatters.Binary.BinaryFormatter voor het omzetten van runtime objecten naar een systeembestand.

			(Game programming patterns in Unity with C#, 2017) (Unity, 2014)
12	Visual saves	FR7	Zie toepassing van Auto save (ontwikkelpatroon 11) met technische implementatie van tekstuele instructies (zie GUI-mogelijkheden §2.2).
13	Pause	FR4	Scene tijd manipuleren met Time.timeScale van de Unity API. (Unity, 2017)
14	Free look	FR5	Een GameObject te met ieder een eigen "controller" component, voor het afhandelen van de input van de gebruiker (Unity, 2017) met een door Unity standaard aangeboden Camera component (Zie GUI-mogelijkheden §2.2)
15	Instant Replay	N.v.t	-
16	Game Progress	N.v.t	-
17	Closed Captioning	N.v.t	-
18	Interaction Aids	FR12	(Zie GUI-mogelijkheden §2.2)
19	Adaptive Difficulty level	N.v.t	-
20	Reconfigurable buttons	N.v.t	-
21	Tutorial Agent	N.v.t	-
23	Playground	N.v.t	-
24	Journal	FR11	(Zie GUI-mogelijkheden §2.2)

3.4 Samenvatting van de Mechanics uit de gewenste Dynamics

Tabel 28 Samenvatting van de Mechanics uit de gewenste Dynamics

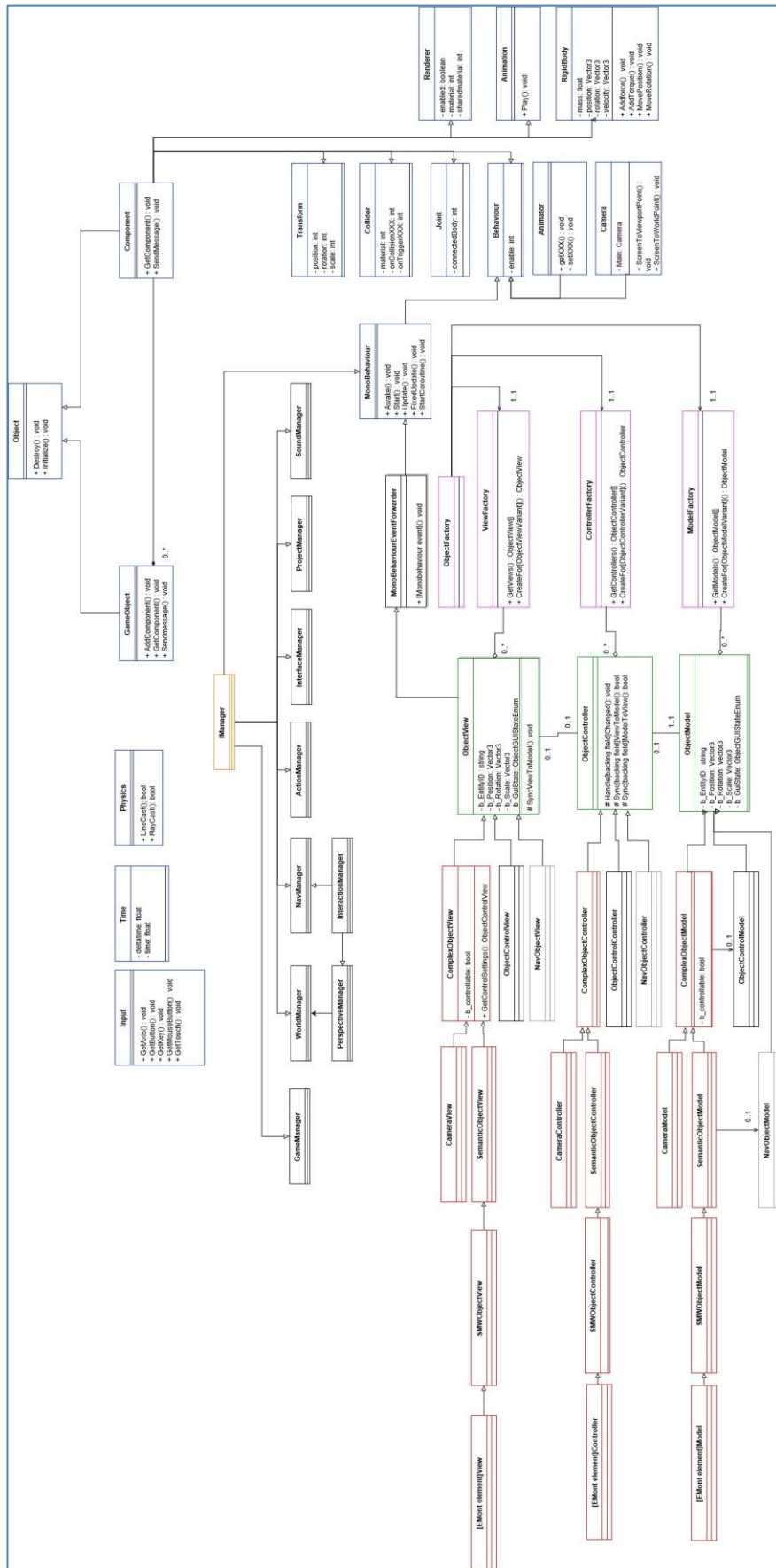
Impact op het basismodel	Relevante eisen
Singleton "GameManager" als god-object voor het afhandelen van de spelwerking.	FR2, FR3, FR4, FR5, FR7
Controller component voor het afhandelen van interactie tussen objecten in de wereld	FR2, FR3, FR4, FR5, FR7
Singleton "GameManager" component voor het beheren van de spelstaat	FR2, FR3, FR5, FR11
Gebruik van het standaard "Canvas", "Canvas Scaler", "Canvas Raycaster" en "Canvas Renderer" Unity component voor menu's	FR5, FR11, FR13
Tijd manipulatie door gebruik van Unity API "Time.timeScale"	FR4, FR5
Singleton component voor serialisatie van objecten	FR7

4. Advies voor een statisch model van een interactieve 3d wereld

In Figuur 27 bevindt zich een het advies voor statisch model van de interactieve 3d wereld. Het model is opgesteld op basis van de volgende onderbouwing:

- De **blauw** gemarkeerde entiteiten zijn de Unity basis waaruit de wereld vormt. Dit is behandeld in het hoofdstuk 2 van het adviesrapport. Het zijn de entiteiten uit het klassendiagram Figuur 13.
- De **bruin** gemarkeerde entiteit is een Singleton klasse genaamd "Manager". Dit is op basis van de "Singleton Manager" practice genoemd in 2.3.1 van het adviesrapport. De practice volgt de maak van verschillende eenmalig voor komende objecten die worden elk een eigen subsysteem binnen de wereld beheersen. De gemarkeerde "Manager" klasse wordt toegepast als basis klasse voor de volgende managers:
 - GameManager: voor het beheer van de hoogste niveau afhandeling als interactie tussen de managers en staten van de wereld (Genoemd in de technische implementatie van de Game Move en de Usability en Accessibility ontwerppatronen 3.1 en 3.3 en ondersteund de requirements FR7)
 - WorldManager: voor het beheren van alle 3d wereld objecten die onderdeel zijn van de 3d simulatie van de EMont elementen. (Dit is vereist voor het beheren van de in requirement FR1 en FR2 beschreven simulatie. Dit ondersteunt ook de technische eis NFR1)
 - NavManager: voor het beheren van alle Navigatie objecten die onderdeel zijn van een Droste-effect navigatie browser (Dit is vereist voor de afhandeling van de Dorste-navigatie van FR13).
 - ActionManager: voor het beheren van de ondernomen en te ondernemen activiteiten (Zo wordt er voldaan aan FR3).
 - InterfaceManager: voor het beheren van GUI-elementen. (Dit is de GUIManager die is genoemd in de beschrijving van de 2.3.1 "Singleton Manager" practice van het adviesrapport)
 - ProjectManager: voor het beheren van projecten. Deze toont de locatie van de mogelijke en te gebruiken projecten bestanden, zoals de EVM-SMW-exports en de ontworpen werelden. (Dit is te herleiden uit de requirements: FR1, FR2, FR6)
 - SoundManager: voor het afhandelen van geluiden. (Uit eigen inzicht: de werelden kunnen wegladen worden door het laden van andere werelden. Bij het managen van geluiden op een centrale plek is het mogelijk om deze overgang te begeleiden.)
 - PerspectiveManager: voor het beheren van het te beleven perspectief (en de besturing) van de objecten in de simulatie wereld. Deze erft de besturingsmogelijkheden over van de InteractionManager. (Dit dient de controle over perspectieven aan te leveren, zoals FR5 dit vraagt)
 - InteractionManager: voor het besturen van 3d wereld objecten. (Uit eigen inzicht: Zowel de PerspectiveManager en NavManger zullen objecten besturing moeten afhandelen)

- De **groen** gemarkeerde entiteiten zijn MVC-objecten van de Unity practice MVC practice (2.3.2). Deze bestaan uit de volgende onderdelen:
 - o Een ObjectView is de basis van alle View objecten in de wereld. Een ObjectView erft over op MonoBehaviourForwarder. De MonoBehaviourForwarder is een aanpassing op het MonoBehaviour component zorgt dat de Unity Events worden doorgegeven naar de Controller. De Views zijn de componenten aan de GameObjects in de wereld, doordat ze komen uit MonoBehaviour. De View legt zijn variabelen vast in een backing field (een private variabele) met een toegekende public accessors (getter en setter). Bij het aan roepen van de accessor zal een event worden gestuurd naar de controller. Om rekening te houden met dat de wereld ontworpen (te herleiden uit NFR3 en FR2.3) moet worden zullen de backingfields public moeten zijn, zodat kunnen ze bewerkt worden in de Unity Inspector. Dit blijkt uit documentatie van de Unity Editor. (Unity, 2017)
 - o Een ObjectController is de basis van alle Controller objecten in de wereld. Deze legt de verbinding tussen de View en de Model. De controller regelt alle interactie met de view (De beschrijving in 2.3.2 zegt dat alle interactie moet worden afhandelen in de controller en volgt ook de “controller” uit de technische implementatie van de beschrijving van de Game Moves). Ook regelt de controller de synchronisatie tussen de variabelen van de Model en de View. De waarden van de View worden eenmalig gesynchroniseerd naar de Model als de View komt van een ontworpen wereld.
 - o Een ObjectModel is het meest actuele savedata object met dat de waarden van een View vasthoudt. De controller gebruikt de Model data voor de interactie. Dit volgt het ontwikkelpatroon 11. Savedata, aangegeven in Tabel 27 van 3.3.
- De **Roze of Paarse** kleur toont de klassen die de objecten maken (Factories) voor ieder van de MVC-(sub)klasse. Dit is op basis van het de practice 2.3.6. Naar eigen voorkeur zijn de Factories zijn te benaderen via een Singleton ObjectFactory.
- Er is gekozen voor een opsplitsing van de MVC-componenten in de volgende vertakkingen:
 - o De (**rood** gemarkeerde) ComplexObject[MVC-Component] klasse: een ComplexObject is een object dat kan worden bediend. Dit splitst op in een Camera en SemanticObject. SemanticObject is bedoeld voor het weergeven van semantische dataobjecten. Een SemanticObject splits in SMWObject wat de basis vormt voor de EMont element MVC-componenten. Deze de EMont elementen zijn de basis de EMont entiteiten die zijn aangegeven als zwart met witte entiteiten in Figuur 10 van Bijlage K.
 - o De ObjectControl[MVC-Component] klasse: een object dat enkel de beschrijft welke bewegingen een ComplexObject kan maken. Het beschrijft de bewegingsrichting en kracht over de x-, y- en z-as.
 - o De (**grijs** gemarkeerde) NavObject[MVC-Component] klasse: een niet te bedienen object als onderdeel van de Droste-Effect navigatie. (Nodig voor FR13)
- Een View object kent de variabele “b_GuiState”. De variabele is een enumeratie bestaande uit mogelijke grafische vertoning. Dit is genoemd in 2.3.4 als het “Statemachines” Unity practice.



Figuur 27 Advies statische model van de interactieve 3d wereld (Kleuren marking onderbouwd de keuzes)

5. Advies uitbreiding statisch model met translatie van RDF-format de EMont naar Objecten

5.1 Exportdata voor de narratives uit de wiki

Na het bestuderen van de Semantische MediaWiki (SMW) en MediaWiki (MW) documentatie is er gebleken dat er vijf verschillende exporteer mogelijkheden worden aangeboden (**Rode markering geeft aan dat ze niet relevant zijn**):

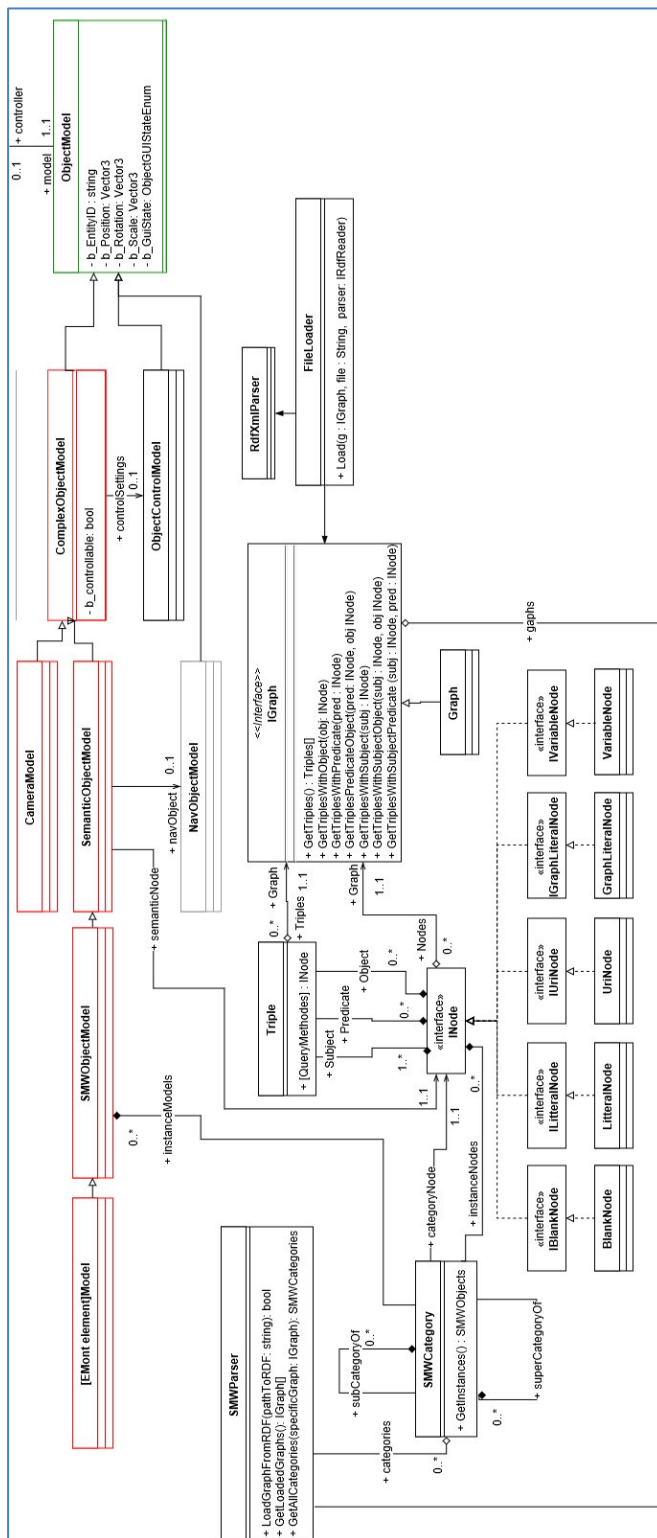
1. De [Special:Export](#), een speciaal interface aangeboden in een aparte pagina binnen de wiki. Deze pagina retourneert een XML met de pagina inhoud van de iedere van de in de meegegeven paginanamen, die bestaan binnen de wiki. Ook aan te roepen via de url: [www.mediawiki.org/wiki/Special:Export/\[FULLPAGENAME\]](http://www.mediawiki.org/wiki/Special:Export/[FULLPAGENAME]). Deze export maakt gebruik van de wiki API en levert resultaten in JSON, PHP en XML. (MediaWiki, 2017) (Semantic MediaWiki, 2017)
2. Het backupscript [dumpBackup.php](#) stort alle wikipagina's in een XML-bestand. [dumpBackup.php](#) werkt alleen op MediaWiki 1.5 of nieuwer. Er is vereist direct toegang te hebben tot de server om dit script te kunnen uitvoeren. Dumpen van mediawiki-projecten worden regelmatig ter beschikking gesteld op <https://dumps.wikimedia.org/>. Opmerking: [AdminSettings.php](#) zal mogelijk moeten worden geconfigureerd om de [dumpBackup.php](#) succesvol te kunnen uitvoeren. (MediaWiki, 2017)
3. Het [OAI-PMH](#)-interface om regelmatig pagina's te halen die sinds een bepaalde tijd zijn gewijzigd. Voor Wikimedia-projecten is deze interface niet openbaar beschikbaar; Zie de feed service van Wikimedia Update. OAI-PMH bevat een wrapper formaat rond de werkelijke geëxporteerde artikelen. (MediaWiki, 2017)
4. Het gebruik van het [Python Wikipedia Robot Framework](#), dit is een framework om wiki's te onderhouden library en verzameling tools die werk automatiseren van een wiki. (MediaWiki, 2017)
5. Op basis van de semantische **notities van de artikelen**, produceert Semantic MediaWiki machine leesbare documenten in OWL / RDF-formaat, die kunnen worden opgehaald via de Special: ExportRDF (zelfde als [Special:Export](#), maar dan voor RDF). Deze export maakt gebruik van het serverside onderhoudsdocument "[dumpRDF.php](#)" dat het mogelijk maakt om RDF triples uit te exporteren. Er is ook een API aanspreekbaar met op URL: [\[wikidomein\]/index.php?title=Speciaal:RDFExporteren&page=\[Paginatitel\]&backlinks=\[true = 1 of false = 0\]&recursive=\[true = 1 of false = 0\]](#). (MediaWiki, 2017) (Semantic MediaWiki, 2017)

5.1.1 Opsomming van de gevonden exporteer mogelijkheden

Tabel 29 Opsomming van de gevonden exporteer mogelijkheden

EVM-SMW Exportmethode	Formaat van export	Bron(nen)
"Exporteren" voor van informatie uit een specifieke set pagina's https://www.projectenportfolio.nl/wiki/index.php/Speciaal:Exporteren	XML	(MediaWiki, 2017) (Semantic MediaWiki, 2017)
"Pagina's exporteren naar RDF" voor de semantische vastlegging van een specifieke set pagina's. https://www.projectenportfolio.nl/wiki/index.php/Speciaal:RDFExporteren	RDF	(MediaWiki, 2017) (Semantic MediaWiki, 2017)

5.4 Koppeling tussen dotNetRDF parsing en de Models van het interactief 3d wereldmodel



Figuur 29 Koppeling tussen dotNetRDF parsing en de Models van het interactieve 3d wereldmodel.

Bijlage R. Functionele ontwerp

Het functioneel ontwerp is online te vinden op <https://bitbucket.org/expertisemanagement/narratieven-3d/wiki/Functioneel%20ontwerp>. Voor een offline versie zie de afbeeldingen heironder:

Dit is een functioneel ontwerp. In dit document wordt uitgelegd hoe het "Narrative 3D" product zal werken. Dit document limiteert zich tot enkel het vertellen van: het doel van het product, het beschrijven van de architectuur op hoog niveau, het aangeven van de functionele eisen/gebruikssituaties en acceptatietesten (testen voor het pijlen van de functionele compleetheid).

Inhoudsopgave

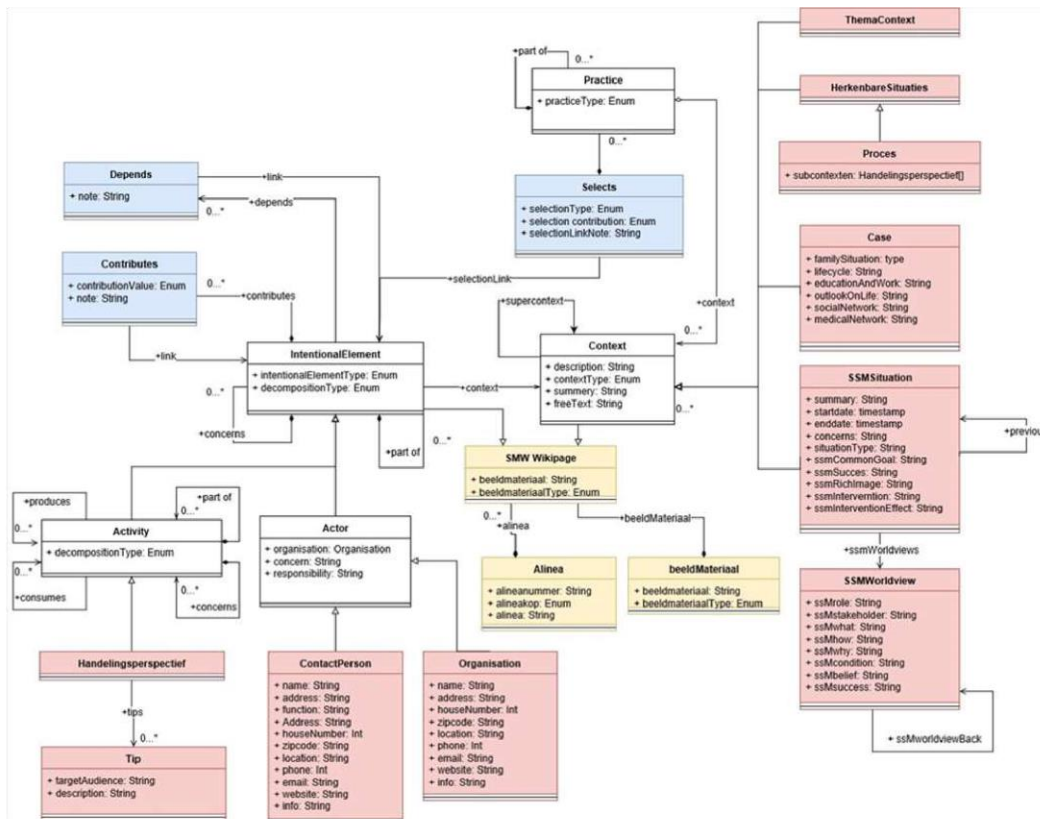
- 1. Doel
- 2. Structuur van de EMont binnen een SMW
- 3. Functionele eisen
- 4. Data
- 5. Gebruikssituaties (Use cases)
 - 5.1 Use Case - Wireframes (W)
 - 5.1.1 W1 - Selecteren van een project
 - 5.1.2 W2 - Selecteren van een Graph en Context
 - 5.1.1 W3 - Tonen van de condities
 - 5.1.1 W3 - Keuzen van een handeling (op een intentional element)
 - 5.2 Use Case - Use Case Map (UCM) tonende de dynamiek van de verhalen
 - 5.2.1 UCM1 - Dynamische weergave van casussen (narratives) van Claire en Mevrouw Schoonenbrug
 - 5.2.2 UCM2 - Dynamische weergave een simpele testcase narrative
- 6. Acceptatietesten (AT)
 - 6.1 AT1 - Doorlopen van de simpele test case
 - 6.2 AT2 - Uitvoeren van de activiteiten
 - 6.3 AT3 - Bekijken van condities

1. Doel

Het product doelt op het aanleveren van een 3d wereld die vormt uit de RDF-export van een met de Expertise and Management ontology (EMont) opgebouwde Semantische MediaWiki. Binnen deze wereld zal de gebruiker kunnen navigeren door situaties door het uitvoeren van activiteiten. Hierdoor zal de gebruiker een "narrative", ofwel drama, ervaren dat de gebruiker begrip zal helpen kweken voor de rollen (situatie met toegekende activiteiten) waar de gebruiker mee te maken krijgt.

2. Structuur van de EMont binnen een SMW

De EMont bestaat uit de 2 hoofdelementen: Context, Intentional Element. Een context is een situaties waar binnen een andere context of meerdere intentional elements kunnen plaats vinden. Een intentional element kent 7 verschillende varianten: Een activiteit, actor, goal, beliep, outcome, condition en beliep. De elementen beschrijven de wereld zoals deze in elkaar zit. In Figuur 41 is de statische vastlegging van EMont gemodeleerd in een klasse diagram. Dit klasse diagram is een structuur van de EMont zoals deze is aangeleverd in een RDF-export van de www.projectenportfolio.nl EVM-SMW. In het klass diagram is te zien dat ieder variant van een intentional element kent eigen relaties met een ander. De EMont basiselementen zijn met wit en zwart aangegeven in het diagram. Deze zullen in elke EVM-SMW-export het zelfde zijn. De blauwe entiteiten zijn geen EMont elementen, maar entiteiten voor het aangeven van relaties tussen EMont elementen. Deze gebeven de condities aan. De gele entiteiten zijn de basis categorieën van een SWM, ofwel klassen, waaruit EMont is gedefinieerd. De Rode entiteiten zijn de L1 en L2 EMont elementen. Hiervan is alleen de dynamiek van belangrijk. De Rode entiteiten zijn verschillend binnen de EVM-SMW's.



Figuur 1: Klasse diagram van de EMont structuur binnen de www.projectenportfolio.nl EVM-SMW

3. Functionele eisen

In overleg met Hans de Bruin zijn er functionele eisen gesteld aan het systeem, zie Tabel 1. Deze beschrijven welke interactie er zal worden aangeboden. Per eis, ofwel requirement, is aangegeven welke prioriteit hiervoor is gesteld. De “Must” dienen te zijn toegepast voor het succes vol werken van de interactiviteit. De “Should” en “Could” prioriteiten zijn voor uitbreidingsdoelenden. De requirements kennen vage begrippen (aangegeven tussen vishaken) die worden verklaard in een begrippenlijst in Tabel 2.

Tabel 1 Functionele eisen voor het functioneel ontwerp

Requirement ID	Prioriteit
FR1.1 - Het systeem kan de [EMont] instantiëren binnen de game engine als gevisualiseerd 3d objecten.	Must
FR1.2 - Het systeem kan de [wikidocumentatie] instantiëren binnen de game engine als gevisualiseerd 3d objecten.	Could
FR2.1 - Het systeem kan de 3d wereld vormen aan de hand van de gebruikers interactie met EMont activiteiten.	Must
FR2.2 - Het systeem kan de 3d wereld vormen aan de hand van de [condities] van de in de wereld bestaende actoren of rollen.	Should

Requirement ID	Prioriteit
FR2.3 - Het systeem kan de 3d wereld vormen aan de hand de eventuele ontwerpers wereldontwerp keuzes, als deze zijn gevraagd door de gebruiker.	Could
FR3.1 - Het systeem kan op basis van het compositie type van de relevante intentional elements, de vrijheidsgraden binnen de huidige context limiteren.	Should
FR3.2 - Het systeem kan op basis van relevante practices, de [handelingsperspectieven] als vrijheidsgraden limiteren.	Should
FR4 - Het systeem geeft de gebruiker een volledige controle de datum, tijd en het tijdsverloop van de interactieve 3D wereld.	Should
FR5.1 - Het systeem kan de gebruiker zelf laten kiezen met welk perspectief er naar een situatie wordt gekeken.	Should
FR5.2 - Het systeem kan de gebruiker interactiemogelijkheid aanpassen aan de hand van het gekozen perspectief.	Should
FR6 - Het systeem kan EMont uitlezen uit een geëxporteerde formaat van de door [EVM-SMW's] aangeboden exporteer functie.	Must
FR7 - Het systeem kan automatisch de [voortgang] van alle narratives bijhouden afhankelijk van de progressie van de gebruiker.	Should
FR8 - Het systeem kan worden gedraaid binnen de Unity Editor.	could
FR9 - Het systeem moet geladen werelden kunnen exporteren in een formaat dat Unity kan bewerken.	could
FR10 - Het systeem moet binnen Unity bewerkte wereldexporten kunnen in laden als wereld.	Could
FR11 - De gebruiker van kan inzien wat de [condities] zijn van de wereld en te volgen actor of rol.	Must
FR12.1 - De gebruiker krijgt te zien wanneer interactie mogelijk is met wereld objecten.	Should
FR12.2 - De gebruiker krijgt te zien welke soort interactie uitgevoerd kan worden.	Should
FR13 - Het systeem kan de gebruiker laten navigeren door de verschillende contexten en in context gelegen overige EMont elementen en wikidata, aan de hand van interacteerbare objecten binnen de 3D wereld of een, ieder moment op te brengen, Droste-effect navigatie.	Could

Tabel 2 Begrippenlijst voor de functionele eisen

(Vage) begrip	Concrete beschrijving
EMont	De verschillende niveaus van het vastgelegde EMont, bestaande uit contexten, intentional elements.
wikidocumentatie	De wiki documentatie bij de EMont elementen, bestaande uit afbeeldingen, video-audio opnames, websites en teksten.
condities	De relaties (depends, selects en connents) tussen de intentional elements.

(Vage) begrip	Concrete beschrijving
handelingsperspectieven	Een instantie van een activiteit, ofwel een specifieke activiteit.
EVM-SMW	Door het EVM-kenniscentrum opgestelde MediaWiki's die EMont hebben vastgelegd aan de hand van de Semantic Media Wiki plug-in.
voortgang	De gemaakte keuzes door een gebruiker.

4. Data

Het ontwerp van de wereld vormt uit een RDF-export van de paginastructuur binnen een [Semantische MediaWiki \(SMW\)](#). De geëxporteerde RDF-gegevens liggen vast in een lokaal op de computer geplaatst ".xml" of ".rdf" bestand. Deze bestanden komen uit een speciale "Speciaal:RDFExporteren" exporteerfunctie van een SMW van het EVM. Een voorbeeld van deze exporteerfunctie is hier te vinden:

<https://www.projectenportfolio.nl/wiki/index.php/Speciaal:RDFExporteren>. Parameters voor de exporteerfunctie zijn [Category](#) en instantie (onderliggend aan een Category) pagina's. Hier volgt het RDF-format (Figuur 3) dat vormt komt uit de exporteer methode met parameters die zijn aangegeven in Figuur 2.

Pagina's exporteren naar RDF

Deze pagina maakt het mogelijk gegevens te verkrijgen van een pagina in RDF-formaat. Voer in het onderstaande invoerveld paginanamen in om pagina's te exporteren. Iedere paginanaam op een eigen regel.

```

Categorie:Project
PR_00181

Categorie:Context
IN_Situation
IN_Role_1
IN_Role_2

Categorie:Activity
IN_Activity_1
IN_Activity_2
IN_Activity_3
IN_Activity_4

```

Ook alle pagina's exporteren die verwijzen naar de te exporteren pagina's. Genereert door te bladeren RDF.

Figuur 2: Exporteermethode en parameters



Figuur 4: Selecteren van een project (Toont FR5.2 en FR6)

5.1.2 W2 - Selecteren van een Graph en Context

Step 1 - Select Graph(s) to look in

VDS.RDF.Graph

Step 2 - Select Context

IN_Situation

Step 3 - Select an Actor/Roll

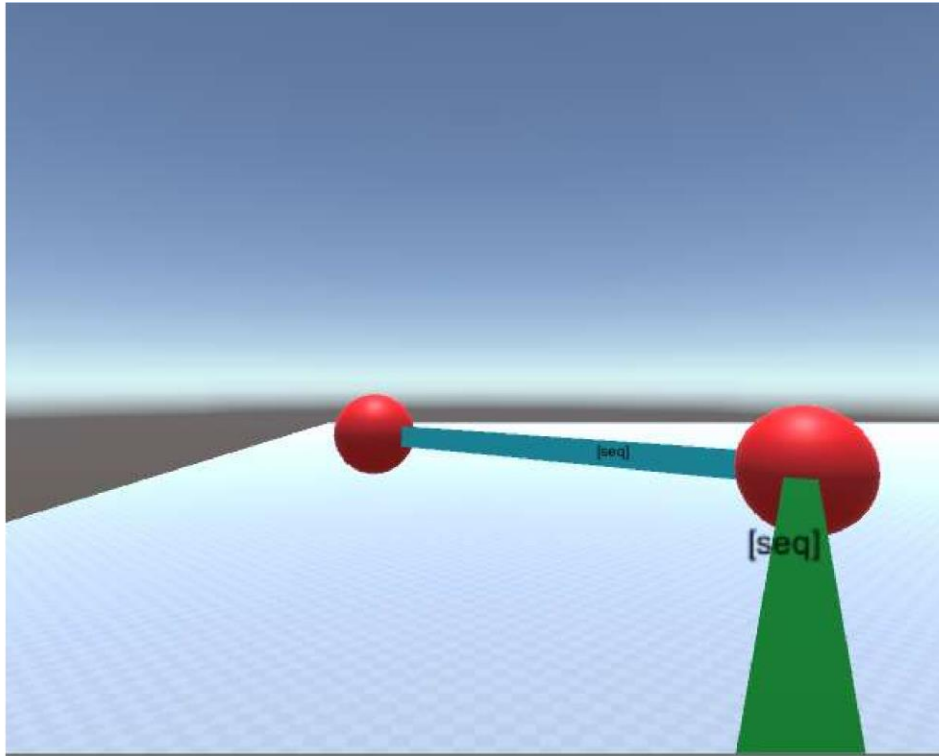
IN_Role_1

Step 4 - Initiate adventure!

Initiate here!

Figuur 5: Selecteren van een Graph uit een RDF-Export en een te beleven context(Toont FR5.2 en FR6)

5.1.1 W3 - Tonen van de condities



Figuur 6: Weergave van condities (Toont FR11)

5.1.1 W3 - Keuzen van een handeling (op een intentional element)



o

e

8

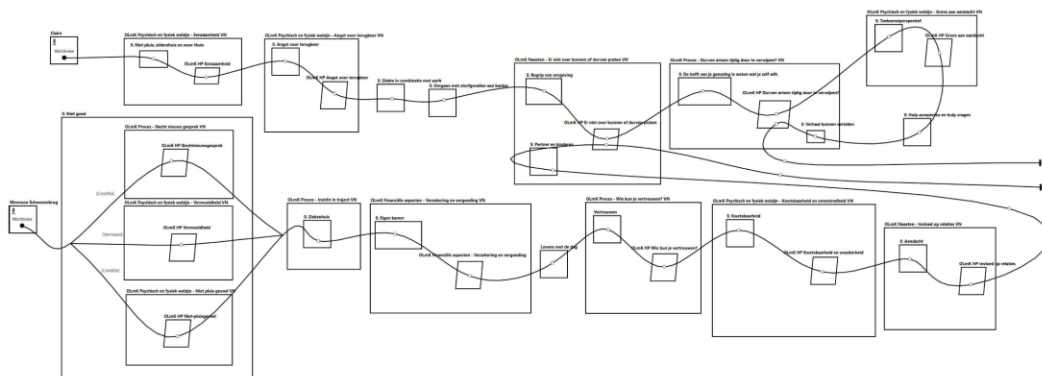
Figuur 7: Keuze menu voor het handelen (Toont FR11)

5.2 Use Case - Use Case Map (UCM) tonende de dynamiek van de verhalen

Voor het begrijpen hoe de gebruiker door een verhaal heen navigeert zal er een voorbeeld worden gegeven van de verhalen zoals deze zijn beschreven in de het "Overleven met kanker (OLmK)" van de www.projectenportfolio.nl EVM-SMW. Ook zal er een casus worden gegeven van een simpele test scenario komende van een [speciale test omgeving](http://www.projectenportfolio.nl) binnen de www.projectenportfolio.nl EVM-SMW.

5.2.1 UCM1 - Dynamische weergave van casussen (narratives) van Claire en Mevrouw Schoonenbrug

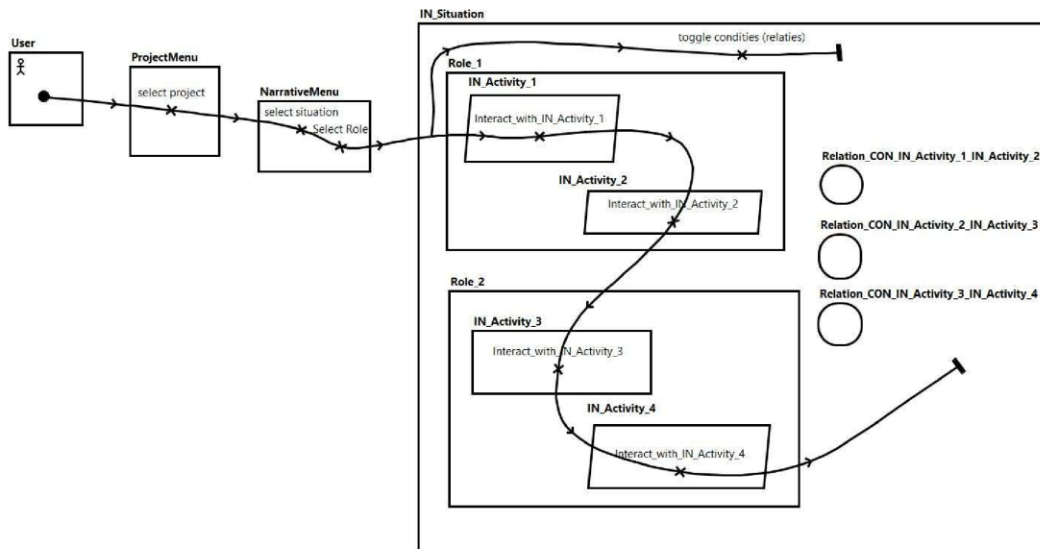
In Figuur 8 staat een weergave van een Use Case Map. In deze weergave zijn de situaties te zien als blokjes, deze kunnen zich bevinden en in andere situaties. Deze gebruiker zal als een rol-situatie (aangegeven in een blokje met een poppetje) navigeren door de diverse situaties (aangegeven met een levenslijn). Binnen de situaties zijn activiteiten (aangegeven met blokjes met een rondje) getoond. De activiteiten kunnen gekozen/uitgevoerd worden afhankelijk van de condities (aangegeven met tekst bij een splitsende levenslijn).



Figuur 8: Dynamische weergave van casussen (narratives) van Claire en Mevrouw Schoonenbrug

5.2.2 UCM2 - Dynamische weergave een simpele testcase narrative

In Figuur 9 is een weergave van een testbare casus opgesteld door Hans de Bruin.



Figuur 9: Dynamiek van een narrative in de interactieve 3d wereld (Bovenste is deel 1 en onderste deel 2 van het gehele plaatje)

6. Acceptatietesten (AT)

6.1 AT1 - Doorlopen van de simpele test case

Testsoort = straattest

Te testen functionele eisen = FR1, FR6

Chronologische volgorde van wireframe = W1 -> W2 -> W3 of W4

Relevante Narrative = UCM2

Testomgeving = Het product zal worden getest binnen de Unity Editor (deze is platform onafhankelijk).

Uitvoer = Het product zal worden gestart. Vervolgens zal de gebruiker een project selecteren in het projectenmenu (W1). Vervolgens zal de gebruiker een willekeurige Graph selecteren. Nu zal ook een optie verschijnen om een Context (Situaties) te kiezen. De gebruiker zal een situaties kiezen tot dat hij/zij een Role_1 of Role_2 kan kiezen. De gebruiker kiest een rol en klikt op de knop om de situatie te laden.

Verwachte resultaat = De gebruiker beland in de 3d wereld met een te besturen rol object. De wereld zal vol zitten met 3d objecten van EMont elementen die zich bevinden in de zelfde situatie. (Role_1, Role_2, IN_Activiteit_1, IN_Activiteit_2, IN_Activiteit_3 en IN_Activiteit_4)

6.2 AT2 - Uitvoeren van de activiteiten

Testsoort = straattest

Te testen functionele eisen = FR2.1

Chronologische volgorde van wireframe = W1 -> W2 -> W3

Relevante Narrative = UCM2

Testomgeving = (Gelijk aan die van AT1)

Uitvoer = Eerst zal AT1 worden uitgevoerd. Vervolgens zal de gebruiker interacteren met IN_activiteit_1 en dan IN_Activity_2.

Verwachte resultaat = De stappen van AT1 zullen lukken. Het resultaat van AT1 zal zichtbaar worden. Wanneer de gebruiker interacteerd met IN_activiteit_2 dan zal deze bij de uitvoer van "IN_Activity_2" de gebruiker plaatsen in "Role_2".

6.3 AT3 - Bekijken van condities

Testsoort = straattest

Te testen functionele eisen = FR11

Chronologische volgorde van wireframe = W1 -> W2 -> W4

Relevante Narrative = UCM2

Testomgeving = (Gelijk aan die van AT1)

Uitvoer = Eerst zal AT2 worden uitgevoerd. Vervolgens zal de gebruiker relaties in de wereld ophalen.

Verwachte resultaat = De stappen van AT2 zullen lukken. Het resultaat van AT2 zal zichtbaar worden. Bij het opvragen van de relaties zullen de condities van de activiteiten zichtbaar worden.

Bijlage S. Technische ontwerp

Het functioneel ontwerp is online te vinden op <https://bitbucket.org/expertisemanagement/narratieven-3d/wiki/Technisch%20ontwerp>. Voor een offline versie zie de afbeeldingen heironder:

Dit is een technisch ontwerp. In dit document wordt uitgelegd hoe het "Narrative 3D" product in elkaar zit. Dit document limiteert zich tot enkel het vertellen van: het beschrijven van de architectuur laag niveau, het aangeven van de technische eisen, te gebruiken ontwikkelpatronen/algortime en unittesten (testen voor het pijlen van de technische compleetheid).

Inhoudsopgave

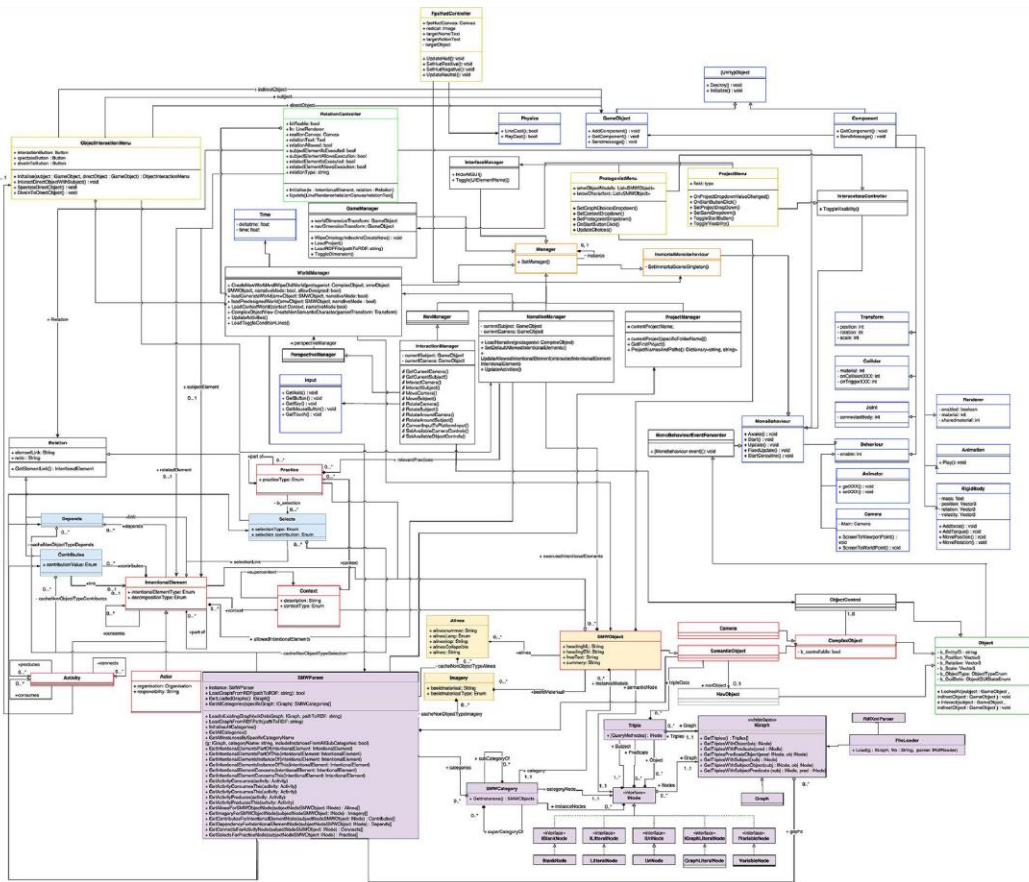
- 1. Doel
- 2. Statische structuur van de interactieve 3d wereld
 - 2.1 De "Manager" klassen
 - 2.2 De wereldobjecten
 - 2.2 De ObjectControllers
 - 2.3 De niet-wereldobjecten (relaties)
 - 2.4 De specifieke Controllers (uitbreiding op interactie binnen de wereld),
 - 2.5 De RDF omzettingklassen
- 3. Technische eisen
- 4. ontwikkelpatronen / algoritmes
 - 4.1 statisch model structuur met ontwikkelpatronen
- 5 Dynamische structuur van de interactieve 3d wereld
 - 5.1 Het laden van een inladen van een Project via het ProjectMenu
 - 5.2 Het het kiezen van een te bekijken narrative
 - 5.3 Het besturen van een ComplexObject wereldobject
 - 5.4 Het interacteren een gebruiker bestuurd wereldobject met een andere wereldobject, via de camera van het bestuurd wereldobject
 - 5.5 Het kiezen van een interactie uit het ObjectInteractieMenu
 - 5.6 Het laden van een wereld
- 6. Ontwikkelomgeving
 - 6.1 Tools
 - 6.2 Opzet van de Bitbucket
 - 6.3 Folderstructuur & Scene sctructuur

1. Doel

Het product doelt op het aanleveren van een 3d wereld die vormt uit de RDF-export van een met de Expertise and Management ontology (EMont) opgebouwde Semantische MediaWiki. Binnen deze wereld zal de gebruiker kunnen navigeren door situaties door het uit voeren van activiteiten. Hierdoor zal de gebruiker een "narrative", ofwel drama, ervaren dat de gebruiker begrip zal helpen kweken voor de rollen (situatie met toegekende activiteiten) waar de gebruiker mee te maken krijgt.

2. Statische structuur van de interactieve 3d wereld

De interactieve 3d wereld bestaat uit zes verschillende soorten klassen: de Managers, de wereldobjecten, ObjectControllers, de niet-wereldobjecten (relaties), specifieke Controllers (uitbreiding op interactie binnen de wereld), de RDF omzettingklassen. Deze zijn verwerkt in het klasse diagram (Figuur 1)



Figuur 1: Statische modelstructuur van de interactieve 3d wereld.

2.1 De "Manager" klassen

De "Managers" klassen zijn voor het beheren van de "Higher temporal structure" (vormt en beschrijft de wetten) van het interactief medium. De Manager klassen zijn:

- De GameManager, een Singleton voor het afhandelen van de hoogste niveau keuzes, zoals welk project geladen zal worden en welke semantische data en verhalen uit het project zullen worden gehaald.
- De ProjectManager, een Singleton voor het opslaan/ophalen van projecten en hun bestanden.
- De WorldManager, een Singleton voor het beheren van de wereld voor een te beleven narrative.
- De InteractionManager, een Singleton voor het besturen van een wereldobject (subject) en een camera. Dit vormt het perspectief van binnen een wereld.
- De PerspectiveManager, een subklasse van de InteractionManager voor het besturen van objecten binnen de wereld van de WorldManager.
- De NavManager, een Singleton voor het beheren van de wereld voor de Droste-navigatie. Dit is een subklasse van de InteractionManager.
- De NarrativeManager, een Singleton voor het beheren van de condities van de EMont elementen. In andere worden kan de klasse beschreven worden als de klasse voor het verkrijgen van informatie over de voorgang en mogelijkheden van een narrative.

2.2 De wereldobjecten

De interactieve 3d wereld is waarneembaar met de wereldobjecten. Deze bestaan uit een GameObject met een klasse van de superklasse "Object". Het Object kent de variabele "ObjectType" voor het beschrijven van het soort wereldobject. Een wereldobject komt voor in de volgende varianten:

- De Object, een object.
- De ComplexObject, een bestuurbaar object.
- De Camera, een ComplexObject variant representeerde een Camera.
- De SemanticObject, een object dat terug is te vinden als een Node in een Graph.
- De SMWObject, een semantische object dat beschreven als een instantie binnen een Semantische MediaWiki.
- De EMont elementen (genoemd in het [H2 van het FO](#)), de wereldobject voor het beschrijven menselijkhandelen.

2.2 De ObjectControllers

Voor het besturen van wereldobjecten van het type "ComplexObject" zal het GameObject van een ComplexObject een ObjectController component moeten hebben. Dit Object beschrijft de mogelijke, niet-mogelijke en vereiste input voor een soort beweging. De soorten bewegingen zijn: Het bewegen over het draaien draaien om: de y-as, x-as en z-as. De ObjectController kent twee varianten:

- De ObjectController, een index van mogelijke bewegingsmogelijkheden van een ComplexObject.
- De CameraController, een index van mogelijke bewegingsmogelijkheden van een Camera.

2.3 De niet-wereldobjecten (relaties)

De niet-wereldobjecten zijn de klasse voor het beschrijven van de relaties tussen de EMont elementen en de subobjecten van object binnen RDF (deze subobjecten kennen geen categorie of klasse in het RDF). De niet-wereldobjecten zijn aangegeven met de blauwe blokken binnen Figuur 1.

2.4 De specifieke Controllers (uitbreiding op interactie binnen de wereld),

Om binnen tussen wereldobjecten interactie mogelijk te maken zijn specifieke componenten geschreven die aan alle GameObjecten kunnen worden geplaatst. Een wereldontwerper kan eigen klassen hiervoor schrijven. Binnen de interactie 3d wereld zijn de volgende Controllers (aangegeven binnen Figuur 1 als klasse met gele randen):

- De FpsCameraController, een component aan een Camera van een bestuurbaar wereldobject. Het het schieten van een RayCast (een recht uit vliegende lijn) die bij aanraking van een wereldelement een interactie scherm kan openen. Ook toont het informatie over het object waar de RayCast mee in aanraking komt.
- De ObjectInteractionMenu, een component aan een wereldobject voor tonen van de mogelijke interactie tussen het bestuurd wereldobject en het wereldobject waar het menu aan vastzit. Kunnen enkele worden bedient met de 2DRayCast van de cursor.
- De RelationController, een component aan een GameObject voor tonen van de mogelijke de relaties tussen wereldelementen met een niet-wereldobject (relatie) en het naar gerelateerde object.

D NarrativeMenu, een component aan een GameObject voor het bieden van de gebruiker's keus voor een te volgen narrative.

- De ProjectMenu, een component aan een GameObject voor het bieden van de gebruiker's keus voor een te gebruiken project.

2.5 De RDF omzettingsklassen

De klassen voor het inlezen van een EVM-SMW-export (in RDF-formaat) en deze om te zetten als softwareobjecten. De klassen bestaan uit de [Nodes, Triples en Graphs uit de dotNetRDF library](#) en de SMWCategory voor het representeren van een Categoriepagina uit de semantische wiki en een SMWParser als Singleton voor het ophalen een aanmaken van de instanties van de SMW-instantiepagina's als (niet-)wereldobjecten en SMW-categorieën als SMWCategories.

3. Technische eisen

Requirement ID	Prioriteit
NFR1 – De applicatie dient minimaal aan FR1 en FR2.1 te voldoen.	Could
NFR2 – De applicatie kan de narratives van diverse actoren of rollen te kunnen visualiseren in het 3d wereldmodel.	Must
NFR3 – De uit de applicatie geëxporteerde werelden dienen te kunnen worden aangepast binnen de Unity Editor.	Should
NFR4 – De applicatie dient zo te werken dat de gebruiker geen bronnen hoeft raad te plegen buiten het systeem, om aan de hand van activiteiten uitvoer in een nieuwe context te komen.	Could
NFR5 - De applicatie laadt alleen contexten als wereld basis, als de gebruiker een [narrative volgt] van een actor of rol.	Should
NFR6 – De applicatie dient een 100% van het voorkomen een [fout geladen wereld] te herstellen met een [toegestane wereld].	Should
NFR7 - De klassen dienen 100% van de inlaad pogingen het EVM-SMW RDF-formaat export te kunnen bouwen als objecten binnen de wereld.	Should

4. ontwikkelpatronen / algoritmes

In een onderzoek (naar een interactieve 3d wereld voor het vertellen van verhalen) is er gezocht naar veel gebruikte ontwikkelpatronen voor een interactieve 3d wereld binnen Unity. Hieruit zijn een aantal verschillende ontwikkelpatronen gevonden, deze zijn getoond in Tabel 1. In dit onderzoek is ook gekeken naar de technische ontwerp vereisten die zijn te herleiden uit de [functionele eisen](#).

Impact op het basismodel	Relevante eisen
Singleton (Manager) component: Een object in de Unity Scene met daaraan een Singleton component dat een specifiek onderdeel regelt van de applicatie.	N.v.t

Impact op het basismodel	Relevante eisen
Observer: Het definiëren van subject objecten die verandering in waardes door geeft aan een observer object, doormiddel van events. Het observer object kan reageren op de waarde aangegeven door het verstuurd event van de subject verandering. Subjects kunnen zelf bepalen of deze gekoppeld zijn aan een observer, of willen ontkoppelen.	N.v.t
MVC: Het vervangen van de controller componenten aan een object in de Unity Scene door een view component, een controller object en een model object. De view kent geen waardes. De view stuurt alle events door naar de bijbehorende controller. De controller handelt alle Monodevelop-events af en regelt update de waardes van het model en de componenten het view object. Het model is een object dat alle waardes van de view vasthoudt. De attributen die de view vereist te hebben zal bestaan uit een backed variabel met een ophaal- (Getter) en instelmethode (Setter). De backed variabel is de variabel zoals deze is opgeslagen en kan alleen worden aangepast via de Getter en Setter methode. De Setter methode verstuurd een event naar de gekoppelde controller als deze de backed variabel heeft aangepast.	N.v.t
State machines: Het vooraf definiëren van de staten van een object, zodat enkel een verandering van staat genoeg is om het object anders te laten werken. De staten worden aangeleverd in een ENUM variabel.	N.v.t
Factory (voor ieder gebruikt MVC-component): Een specifieke klasse voor het creëren van de MVC-componenten.Factories houden bij welke objecten deze heeft gemaakt.	N.v.t

Tabel 1: Technische ontwerp vereisten voor de te gebruiken ontwikkelpatronen

Impact op het basismodel	Relevante eisen
Singleton GameManager als god-object voor het afhandelen van de spelwerking.	FR2, FR3, FR4, FR5, FR7
Controller component voor het afhandelen van interactie tussen objecten in de wereld	FR2, FR3, FR4, FR5, FR7
Singleton GameManager component voor het beheren van de spelstaat	FR2, FR3, FR5, FR11
Gebruik van het standaard Canvas, Canvas Scaler, Canvas Raycaster en Canvas Renderer Unity component voor menu's	FR5, FR11, FR13
Tijd manipulatie door gebruik van Unity API Time.timeScale	FR4, FR5
Singleton component voor serialisatie van objecten	FR7

Tabel 2: Technische ontwerp vereisten voor de functionele eisen

4.1 statisch model structuur met ontwikkelpatronen

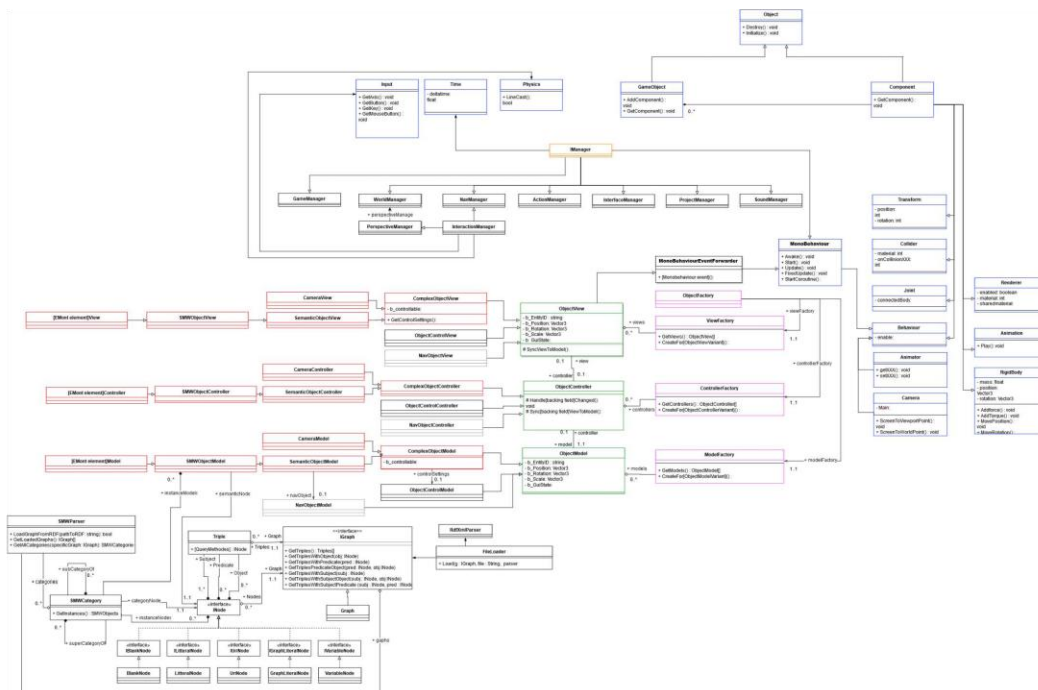
- De blauw gemarkeerde entiteiten zijn de Unity basis waaruit de wereld vormen (Figuur 1).

D bruin gemarkeerde entiteit is een Singleton klasse genaamd Manager. Dit is op basis van de Singleton Manager practice. De practice volgt de maak van verschillende eenmalig voor komende objecten die worden elk een eigen subsysteem binnen de wereld beheersen. De gemarkeerde Manager klasse wordt toegepast als basis klasse voor de volgende managers:

- **GameManager**: voor het beheer van de hoogste niveau afhandeling als interactie tussen de managers en staten van de wereld. Dit komt uit het Manager ontwikkelpatroon. (Dit ondersteunt de requirements FR7)
- **WorldManager**: voor het beheren van alle 3d wereld objecten die onderdeel zijn van de 3d simulatie van de EMont elementen. (Dit is vereist voor het beheren van de in requirement FR1 en FR2 beschreven simulatie. Dit ondersteunt ook de technische eis NFR1)
- **NavManager**: voor het beheren van alle Navigatie objecten die onderdeel zijn van een Droste-effect navigatie browser (Dit is vereist voor de afhandeling van de Dorste-navigatie van FR13).
- **ActionManager**: voor het beheren van de ondernomen en te ondernemen activiteiten (Zo wordt er voldaan aan FR3).
- **InterfaceManager**: voor het beheren van GUI-elementen.
- **ProjectManager**: voor het beheren van projecten. Deze toont de locatie van de mogelijke en te gebruiken projecten bestanden, zoals de EVM-SMW-exports en de ontworpen werelden. (Dit is te herleiden uit de requirements: FR1, FR2, FR6)
- **SoundManager**: voor het afhandelen van geluiden. (Uit eigen inzicht: de werelden kunnen wegladen worden door het laden van andere werelden. Bij het managen van geluiden op een centrale plek is het mogelijk om deze overgang te begeleiden.)
- **PerspectiveManager**: voor het beheren van het te beleven perspectief (en de besturing) van de objecten in de simulatie wereld. Deze erft de besturingsmogelijkheden over van de InteractionManager. (Dit dient de controle over perspectieven aan te leveren, zoals FR5 dit vraagt)
- **InteractionManager**: voor het besturen besturen van 3d wereld objecten. (Uit eigen inzicht: Zowel de PerspectiveManager en NavManger zullen objecten besturing moeten afhandelen)
- De groen gemarkeerde entiteiten zijn MVC-objecten van de Unity practice MVC practice. Deze bestaan uit de volgende onderdelen:
 - Een ObjectView is de basis van alle View objecten in de wereld. Een ObjectView erft over op MonoBehaviourForwarder. De MonoBehaviourForwarder is een aanpassing op het MonoBehaviour component zorgt dat de Unity Events worden doorgegeven naar de Controller. De Views zijn de componenten aan de GameObjects in de wereld, doordat ze komen uit MonoBehaviour. De View legt zijn variabelen vast in een backing field (een private variabele) met een toegekende public accessors (getter en setter). Bij het aan roepen van de accessor zal een event worden gestuurd naar de controller. Om rekening te houden met dat de wereld ontworpen (te herleiden uit NFR3 en FR2.3) moet worden zullen de backingfields public moeten zijn, zodat kunnen ze bewerkt worden in de Unity Inspector. Dit blijkt uit documentatie van de Unity Editor.
 - Een ObjectController is de basis van alle Controller objecten in de wereld. Deze legt de verbinding tussen de View en de Model. De controller regelt alle interactie met de view (De beschrijving in de MVC practice zegt dat alle interactie moet worden afhandelen in de controller. Ook regelt de controller de synchronisatie tussen de variabelen van de Model en de View. De waardes van de View worden eenmalig gesynchroniseerd naar de Model als de View komt van een ontworpen wereld.

- o Een ObjectModel is het meest actuele savedata object met dat de waardes van een View vasthoudt. De controller gebruikt de Model data voor de interactie. Dit volgt het ontwikkelpatroon voor savedata.
- o De Roze of Parse kleur toont de klassen die de objecten maken (Factories) voor ieder van de MVC-(sub)klassen. Dit is op basis de MVC factory practice. Naar eigen voorkeur zijn de Factories zijn te benaderen via een Singleton ObjectFactory.
- Er is gekozen voor een opsplitsing van de MVC-componenten in de volgende vertakkingen:
 - o De (rood gemarkeerde) ComplexObject[MVC-Component] klasse: een ComplexObject is een object dat kan worden bediend. Dit splitst op in een Camera en SemanticObject. SemanticObject is bedoeld voor het weergeven van semantische dataobjecten. Een SemanticObject splits in SMWObject wat de basis vormt voor de EMont element MVC-componenten. Deze EMont elementen zijn de basis EMont entiteiten die zijn aangegeven als zwart met witte entiteiten.
 - o De ObjectControl[MVC-Component] klasse: een object dat enkel de beschrijft welke bewegingen een ComplexObject kan maken. Het beschrijft de bewegingsrichting en kracht over de x-, y- en z-as.
 - o De (grijs gemarkeerde) NavObject[MVC-Component] klasse: een niet te bedienen object als onderdeel van de Droste-Effect navigatie. (Nodig voor FR13)

-Een View object kent de variabele b_GuiState. De variabele is een enumeratie bestaande uit mogelijke grafische vertoning. Dit is een toepassing van de Statemachines Unity practice.



Figuur 3: Klasse diagram van het uiteindelijke statische model van interactieve 3d wereld (bevat inlezen, omzetten en visualiseren)

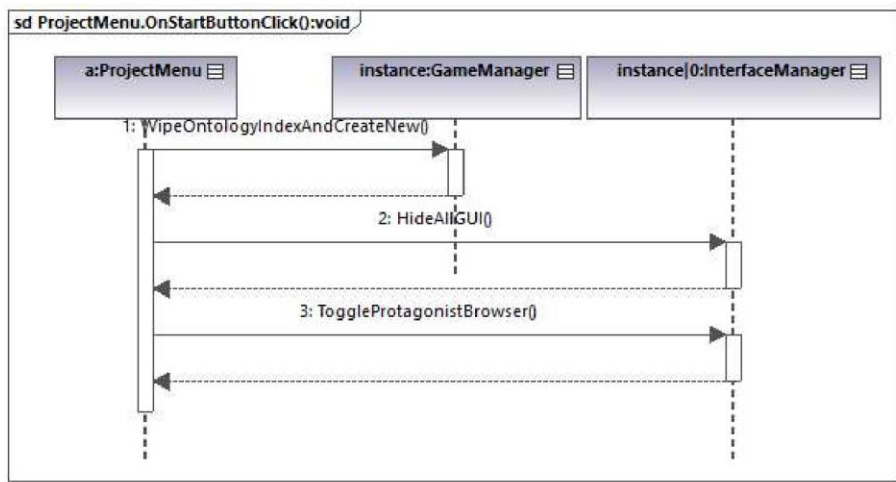
5 Dynamische structuur van de interactieve 3d wereld

De dynamische structuur van de interactieve 3d wereld is op te delen in:

- Het laden van een inladen van een Project via het ProjectMenu (zie 2.4.1)
- Het kiezen van een te bekijken narrative (zie 2.4.2)
- Het besturen van een ComplexObject wereldobject (zie 2.4.3)
- Het interacteren een gebruiker bestuurd wereldobject met een andere wereldobject, via de camera van het bestuurd wereldobject (zie 2.4.4)
- Het kiezen van een interactie uit het ObjectInteractieMenu (zie 2.4.5)
- Het laden van een wereld (zie 2.4.5)

5.1 Het laden van een inladen van een Project via het ProjectMenu

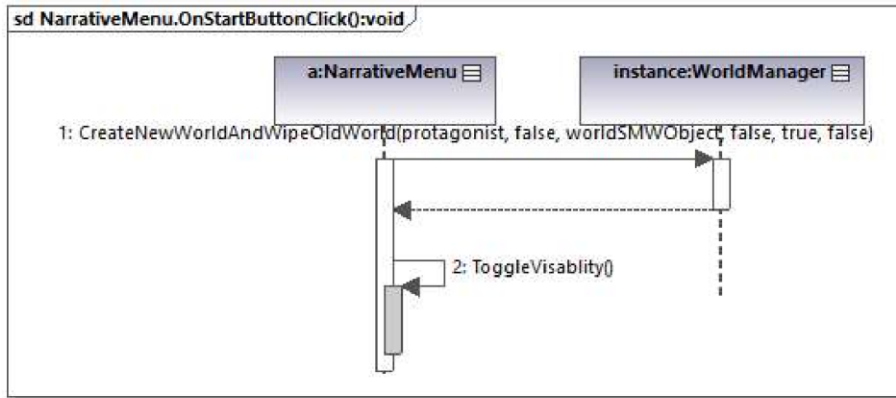
In de use case UCM2 van het [functioneel ontwerp](#) wordt beschreven dat de gebruiker een project kiest. Om het project te kiezen zal de gebruiker een aanroep doen op de ProjectMenu.OnStartButtonClick() methode. Deze aanroep wordt uitgevoerd zoals is aangegeven in figuur 4.



Figuur 4: SequenceDiagram van het kiezen van een project - ProjectMenu.OnStartButtonClick()

5.2 Het kiezen van een te bekijken narrative

In de use case UCM2 van het [functioneel ontwerp](#) wordt beschreven dat de gebruiker een narrative kiest. Om het narrative te kiezen zal de gebruiker een aanroep doen op de NarrativeMenu.OnStartButtonClick() methode. Deze aanroep wordt uitgevoerd zoals is aangegeven in figuur 5.



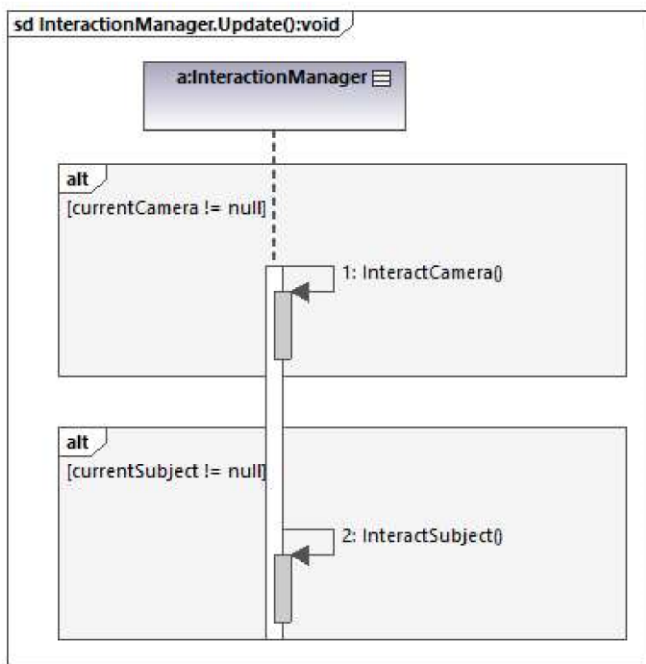
Generated by UModel

www.altova.com

Figuur 5: SequenceDiagram van het kiezen van een narrative - NarrativeMenu.OnStartButtonClick()

5.3 Het besturen van een ComplexObject wereldobject

In de use case UCM2 van het [functioneel ontwerp](#) wordt beschreven dat de gebruiker een verplaatst binnen de situaties. Om het object te besturen zal de gebruikers Unity.InputManager.Input worden gelezen in de InteractionManager.Update() methode. Deze aanroep wordt uitgevoerd zoals is aangegeven in figuur 6.



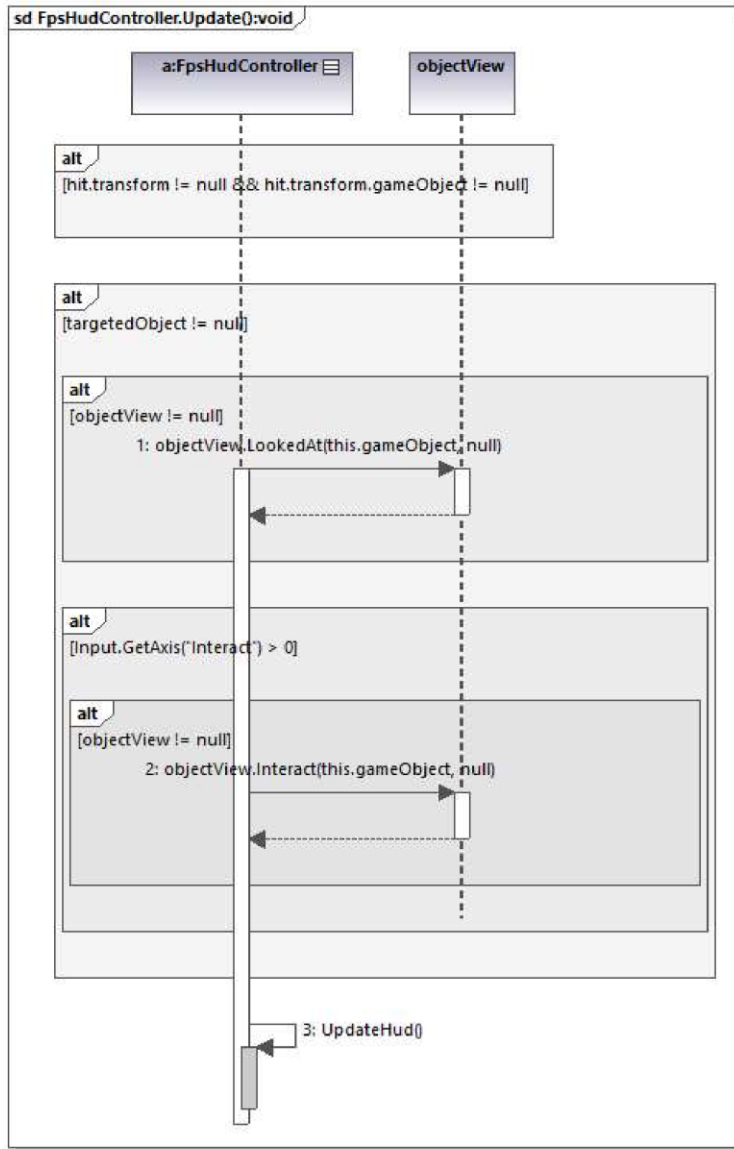
Generated by UModel

www.altova.com

Figuur 6: SequenceDiagram van het kiezen van een ComplexObject wereldobject - InteractionManager.Update()

5.4 Het interacteren een gebruiker bestuurd wereldobject met een andere wereldobject, via de camera van het bestuurd wereldobject

In de use case UCM2 van het [functioneel ontwerp](#) wordt beschreven dat de gebruiker kan interacteren met wereldobjecten. Om met een object te kunnen interacteren wordt er vanaf de FpsCameraController van een bestuurd Camera een RayCast (lijn die detecteert impact) geschoten die bij aanraking met een wereldobject interactie kan afhandelen. Deze interactie komt voor uit de aanroep van de FpsCameraController.Update() methode. Deze aanroep wordt uitgevoerd zoals is aangegeven in figuur 7.



Figuur 7: SequenceDiagram van het kiezen van een ComplexObject wereldobject - InteractionManager.Update()

0

- **MS Visual Studio 2017:** Integrated development environment (IDE) voor het schrijven van de C# code voor de interactieve wereld.
- **Webbrowsers: Edge, Chrome, Opera, FireFox:** de navigatie middel voor het doorlopen van het web. Gebruikt voor het exporteren van de RDF-data.

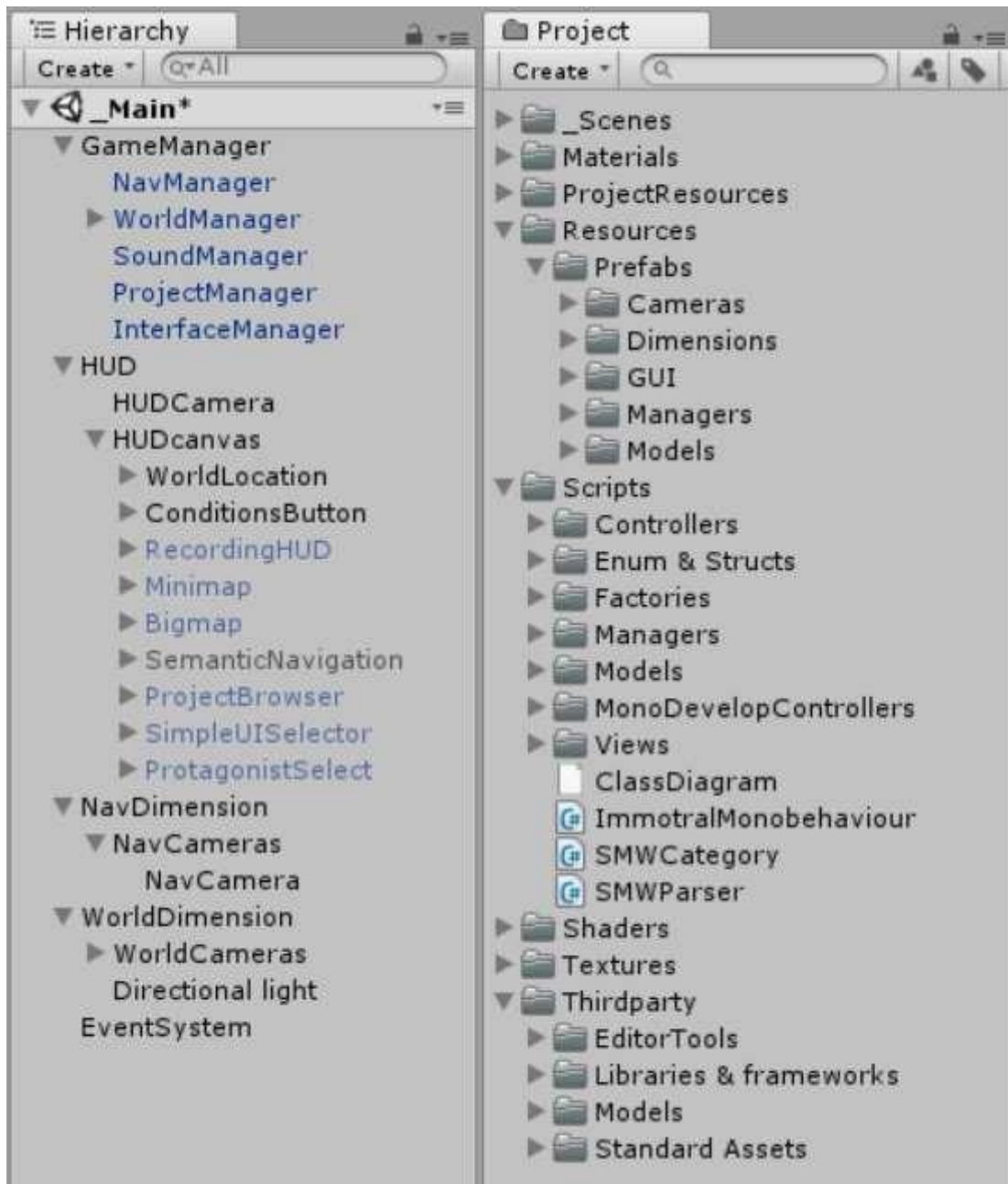
6.2 Opzet van de Bitbucket

De interactieve 3d wereld wordt bij gehouden in een [Git-versiebeheer](#) op een [Bitbucket](#) van een door Bitbucket aangeboden [Atlassian Suite](#). Binnen deze Git wordt er gebruik gemaakt van een "[Feature branching](#)" workflow. Binnen deze structuur bevindt zich een master en development branch als basis. In de development branch zal er worden gewerkt met de meest complete stabiele ontwikkelde versie van het product. Wanneer de development slaagt voor zijn testen zal deze worden [gebouwd met de Unity Editor](#). Als de bouw van het product is geslaagd dan wordt deze gecommited en gepushed naar de master branch.

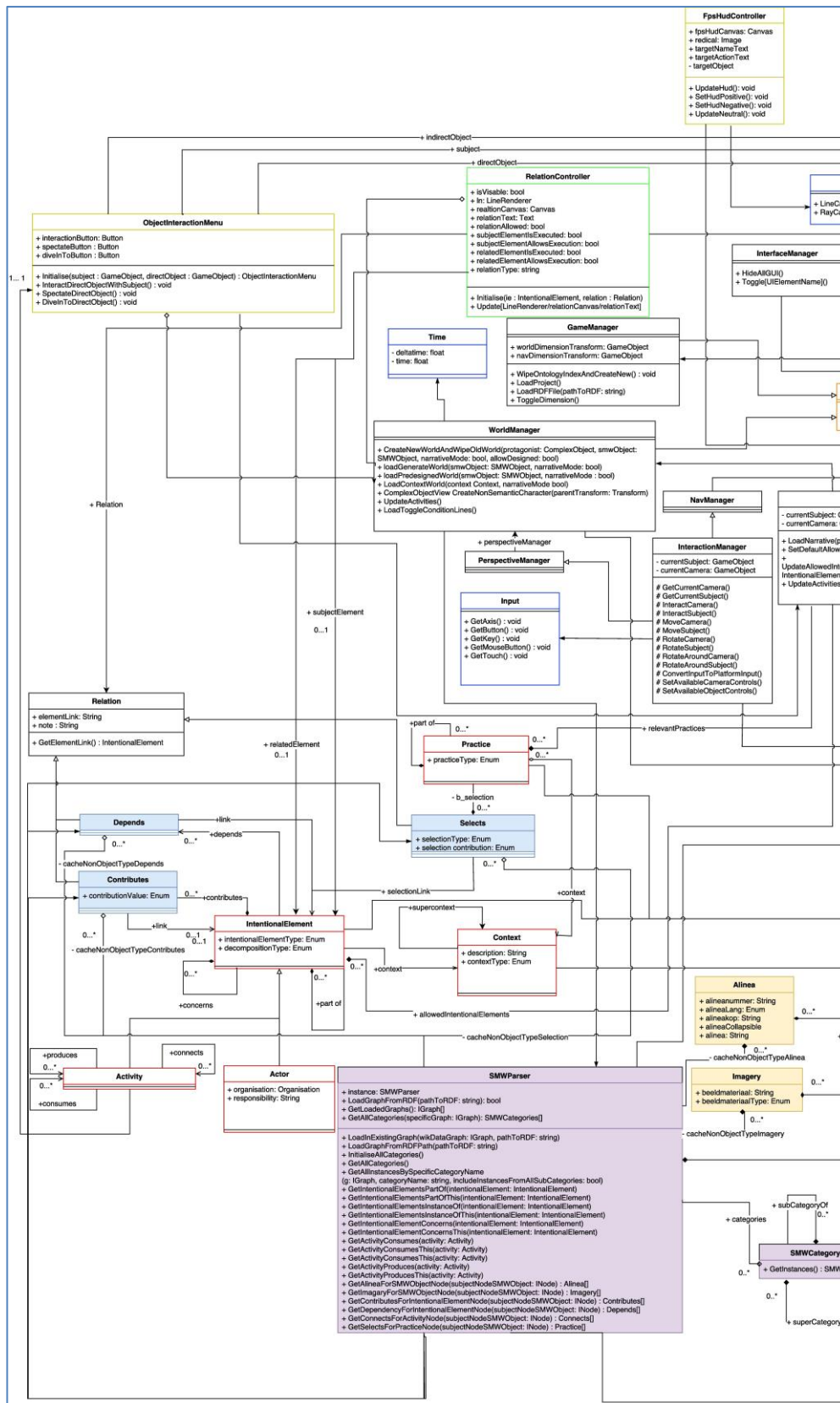
6.3 Folderstructuur & Scene structuur

De interactieve 3d wereld is gebouwd in Unity. Hiervoor is binnen het Unity project een bestandenstructuur bedacht. De bestandenstructuur kent de hoofdmappen: `_Scenes` (hierin staan de werelden), `Materials` (hierin staan de materialen beschreven voor de gebruikte 3d objecten), `ProjectResources` (ontwerpersproject te lezen in 5.4), `Resources` (hierin zitten de [Prefabs](#)), `Scripts` (hierin bevindt zich alle code), `Shaders` (hierin zitten eigen ontwikkelde [shaders](#) voor voor de materials), `Textures` (hierin staan de texturen voor materialen), `Thirdparty` (hierin bevinden zich de externe code of code die niet met de interactieve 3d wereld werkt).

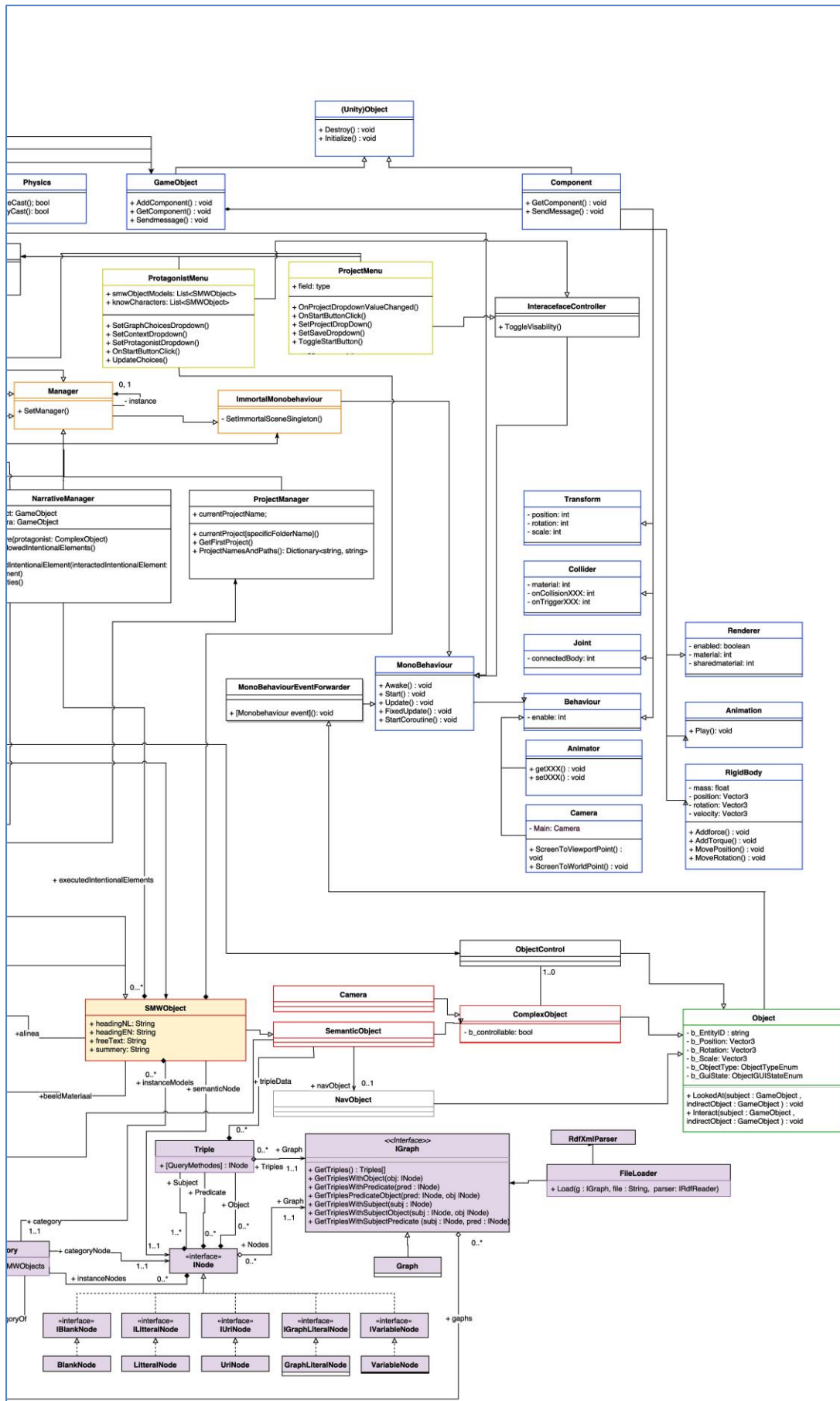
De wereld maakt gebruik van een scene om de wereld in te laden. De structuur van de scene is bestaat uit: `GameManager` object met de andere managers als kinderen, een `HUD` object met als kinderen een `Unity canvas` (met alle HUD-elementen) en een camera die het canvas kan lezen, `NavDimension` object dat alle navigatie object vasthoudt als kinderen en een toegewezen camera heeft voor het kunnen waarnemen van de navigatie objecten, een `WorldDimension` object dat de EMont elementen vasthoudt als kinderen en ook een camera object bevat dat de kinderen kan zien. Ook is er een `eventsystem` object dat vereist is voor het detecteren van interactie op de canvassen.



Bijlage T. Klassendiagram van de interactieve 3d wereld



Figuur 30 Klassendiagram interactieve 3d wereld (deel 1)

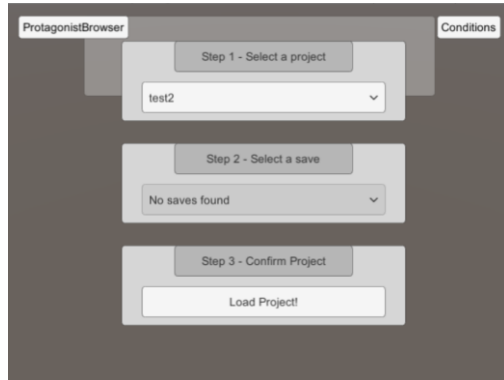


Figuur 31 Klassendiagram interactieve 3d wereld (deel 2)

Bijlage U. Prototype testen & Resultaten

1. Functionele testen (acceptatietesten)

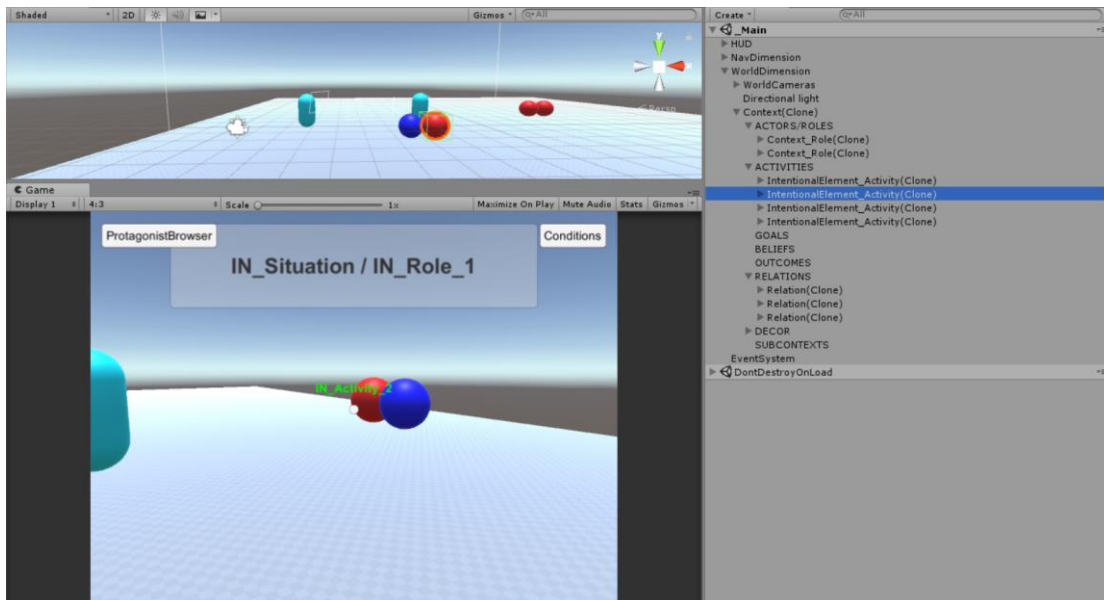
1.1 **[Geslaagd]** AT1 - Doorlopen van de simpele test case



Figuur 34 Gebruiker's keuzes in het Project browser

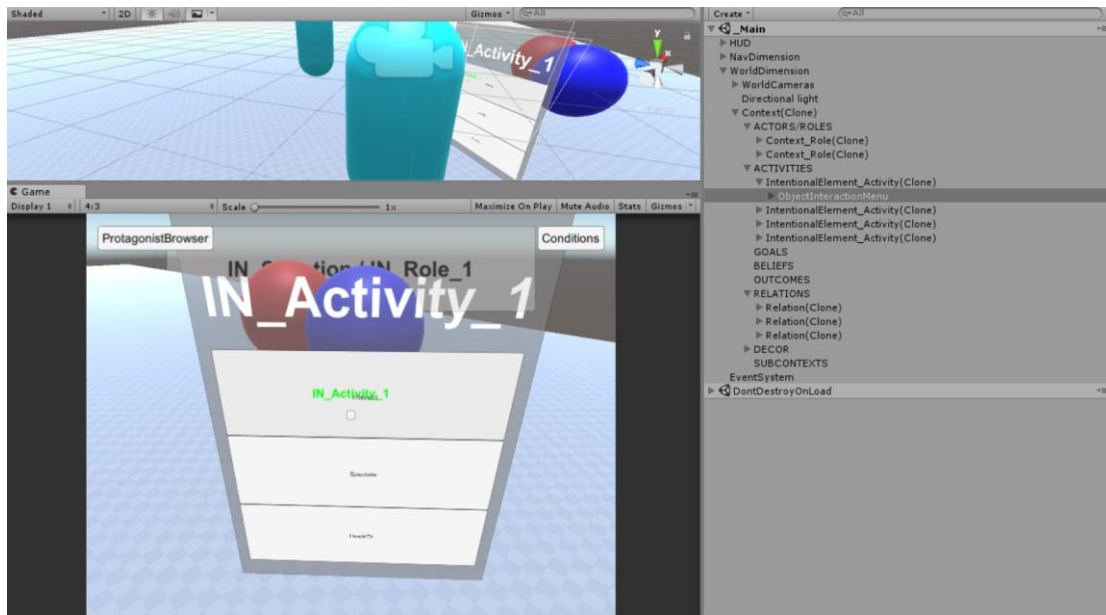


Figuur 32 Gebruiker's keuzes in het Protagonist browser

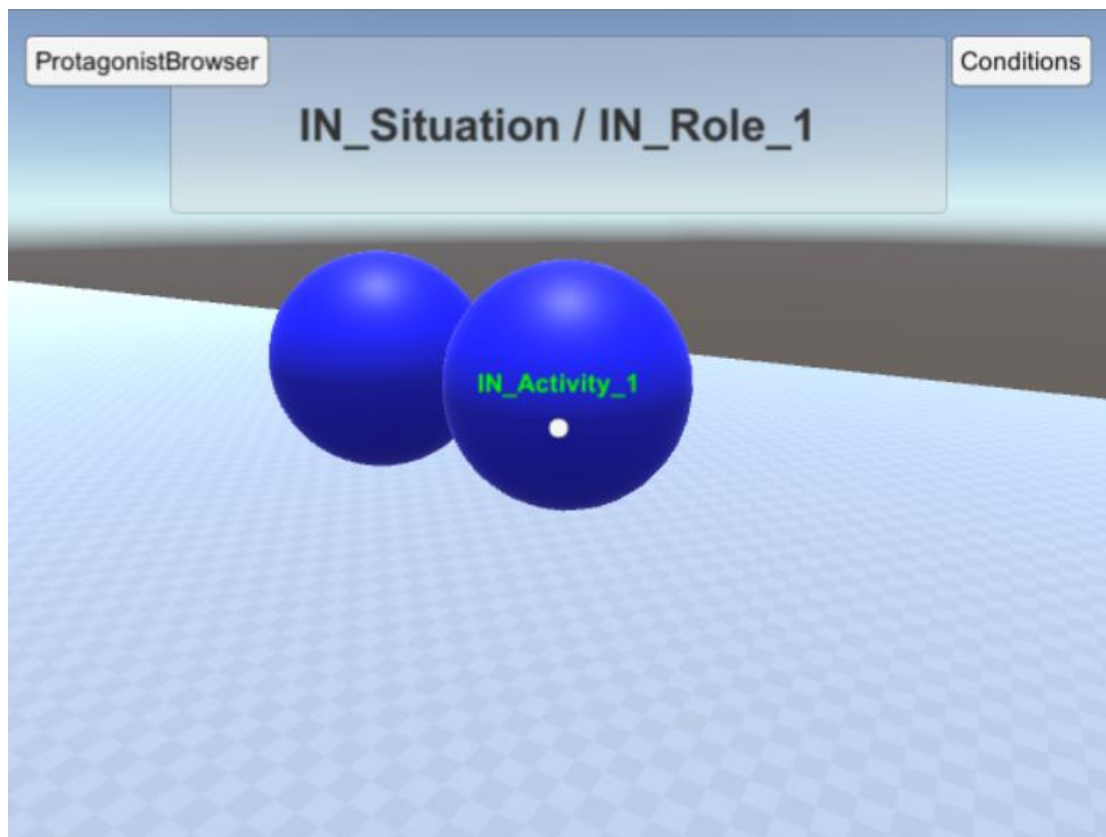


Figuur 33 Gebruiker is "Role_1" in de "IN_Situation" wereld

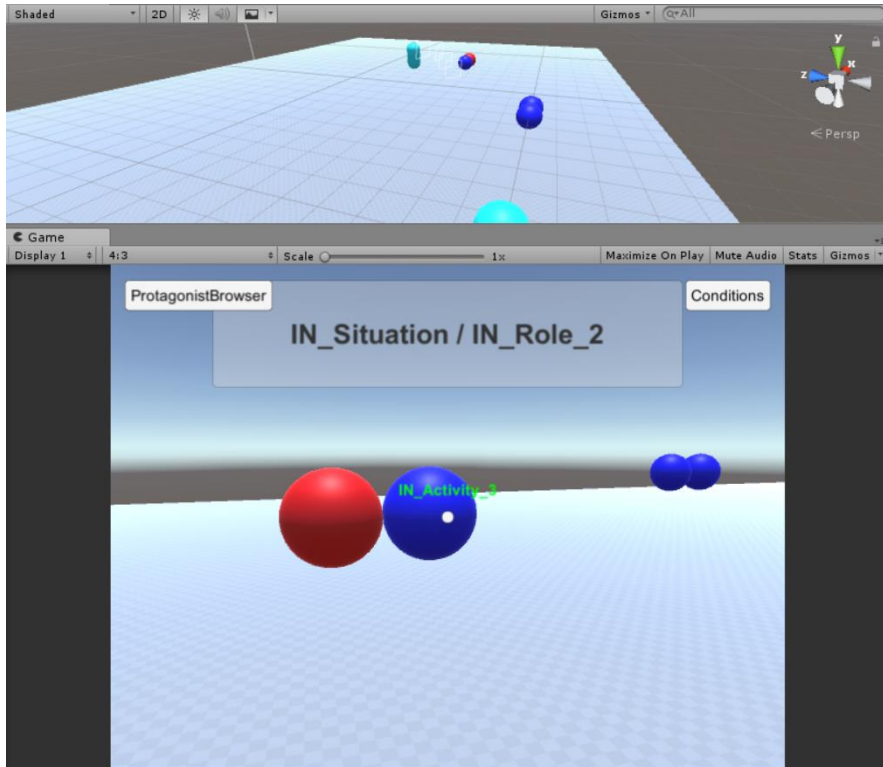
1.2 [Geslaagd] AT2 - Uitvoeren van de activiteiten



Figuur 36 Gebruiker heeft geïnteracteed met "IN_Activiteit_1", waardoor het interactiemenu van de activiteit zichtbaar is

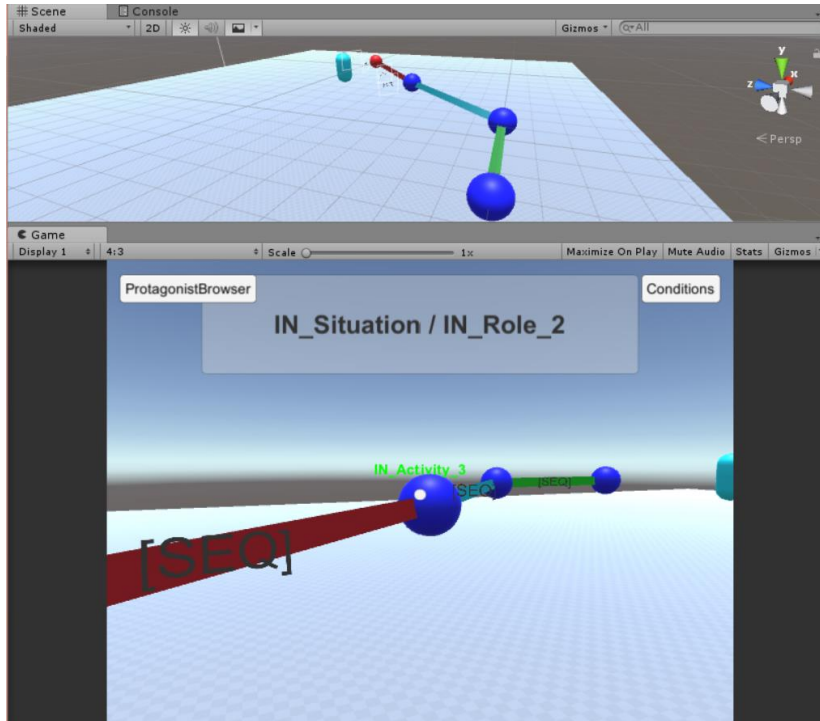


Figuur 35 Gebruiker heeft "Interact" gekozen uit het interactiemenu van "IN_Activity_1", hierdoor is als gevolg "IN_Activity_2" beschikbaar gesteld



Figuur 37 Herhalen van de toegepaste interactie op "IN_Activity_1", maar dan op "IN_Activity_2". Hierdoor is de gebruiker terecht gekomen in "Role_2"

1.3 [Geslaagd] AT3 - Bekijken van condities



Figuur 38 De gebruiker heeft aangegeven dat hij/zij de condities zou willen zien. Hierdoor zijn visuele relaties ontstaan. De weergegeven lijnen zijn de "Connects" entiteiten uit de EMont statische model. groen = uitgevoerd, blauw is mogelijk, en rood = mogelijk na interactie). De tekst op de relatielijn is de verbindingstype

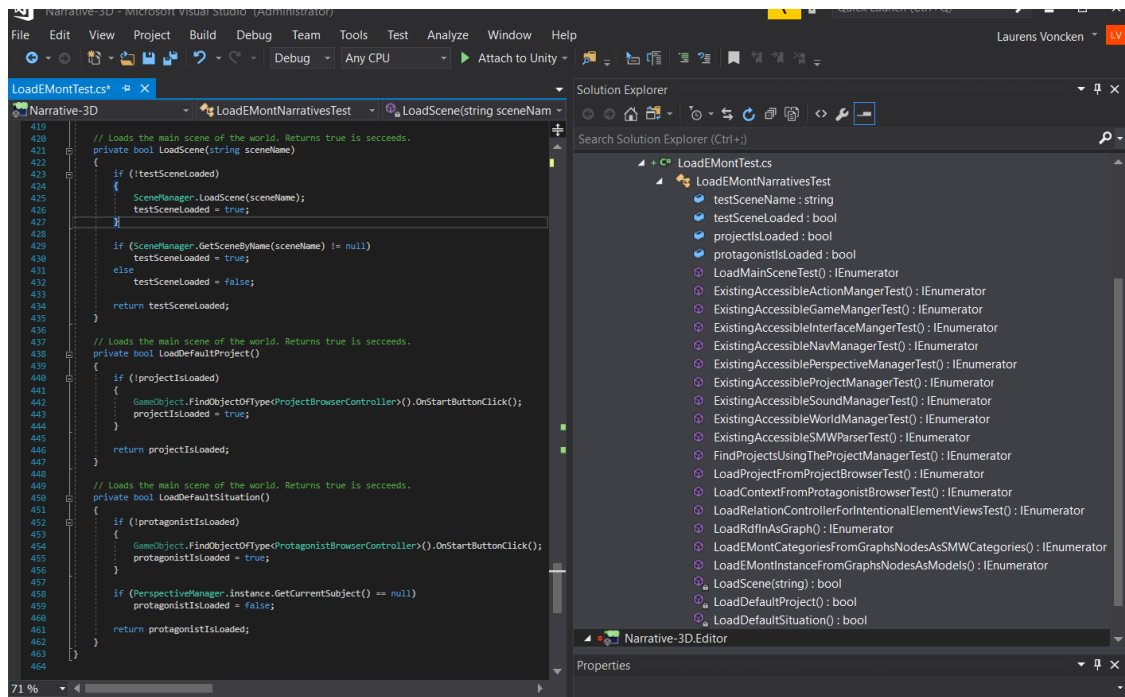
2. Opsomming van de acceptatietest resultaten

Tabel 31 Opsomming van de acceptatietest resultaten

Nummer	Korte beschrijving	Waarneming	Conclusie
AT1	Doorlopen van de simpele test case: <ol style="list-style-type: none">1. Kiest een project2. Kiest een situatie en rol3. Bestaat als gekozen rol in gekozen 3d situatie	De stappen 1 en 2 zijn doorlopen en gebruiker beland in een 3d wereld van de situatie met de gekozen rol.	Behaald
AT2	Uitvoeren van de activiteiten: <ol style="list-style-type: none">1. (Stappen van AT1)2. Interacteert met een IN_Activity_1 en dan IN_Activity_2 (voert deze uit)3. Beland na interactie met IN_Activity_2 in "Role_2"	Gelijke waarneming van AT1 gevolgd door het openen van een interactiemenu waarbij het kiezen van de optie "Interact" de gebruiker na "IN_Activity_2" heeft geplaatst in "Role_2".	Behaald
AT3	Bekijken van condities: <ol style="list-style-type: none">1. (Stappen van AT2)2. Vraagt om condities3. Ziet de condities tussen de intentional elementen in de wereld.	Gelijke waarneming van AT2 gevolgd door het zien van conditie lijnen tussen de EMont wereldobjecten.	Behaald

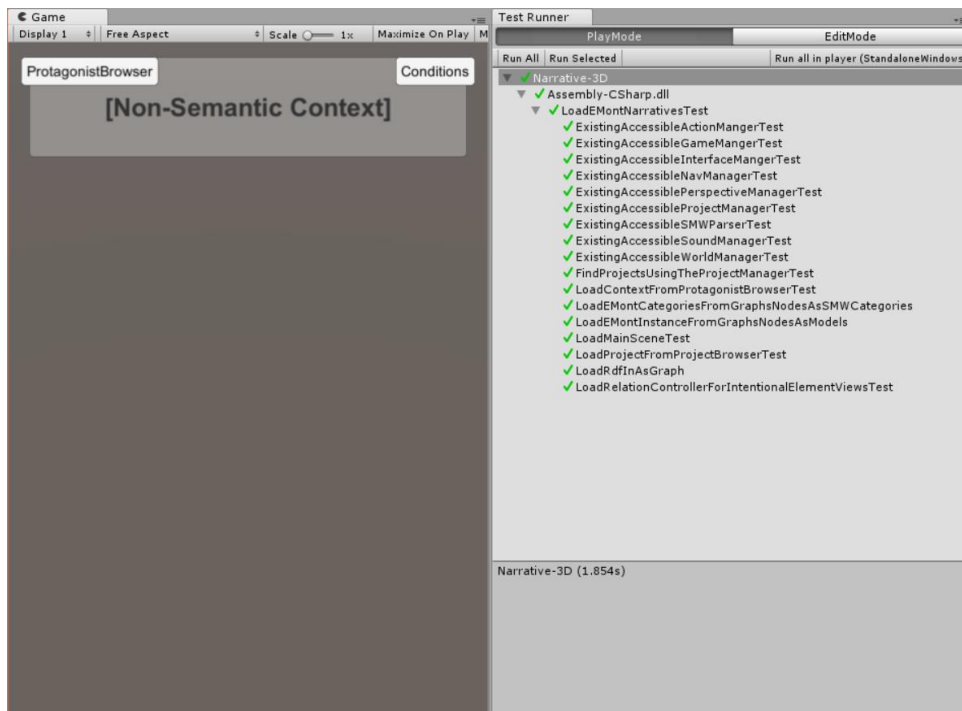
3. Technische testen (unittesten & integratietesten)

3.1 Overzicht van de testresultaten



Figuur 39 Overzicht alle opgestelde unittesten in Microsoft Visual Studio. Unittesten spelen achterelkaar af en maken gebruik van dezelfde geladen scene (wereld), project en gekozen actors

3.2 Overzicht van de testresultaten



Figuur 40 Testresultaten van het automatische unittest executieprogramma "Test runner"