

SyncVR Medical



# Implementeren van wearables binnen Virtual Reality Rehabilitation Programs

David H.W. Bruschke

4-6-2021



Onder begeleiding van

R. Brouwer

V. Broeren

D.R. Stikkolorum

**DE HAAGSE**  
HOGESCHOOL

## Documenthistorie

Datum	Versie	Beschrijving	Auteurs
09-02-201	0.1	Eerste opzet Afstudeerdossier. Formatering toegepast en veldcodes geplaatst.	D.H.W. Bruschke
11-02-2021	0.2	Plan van Aanpak bijgevoegd. Eerste opzet over de theoretische verkenning.	D.H.W. Bruschke
18-03-2021	0.3	High Concept Documents bijgevoegd. Theoretische verkenning uitgewerkt met patterns en Bluetooth Low Energy.	D.H.W. Bruschke
23-04-2021	0.4	Feedback vanuit het concept bespreking verwerkt. Meer ingegaan op persoonlijke ontwikkeling en delen van reflectie bijgewerkt.	D.H.W. Bruschke
05-05-2021	0.5	Nieuwe sprints en Blender renders toe gevoegd voor meer context bij hoofdstuk Realiseren.	D.H.W. Bruschke
13-04-2021	1.0	Tussentijds Assessment feedback verwerkt.	D.H.W. Bruschke
02-06-2021	1.1	Feedback verwerkt, spellingscontrole en stijlfouten verbeterd.	D.H.W. Bruschke
04-06-2021	1.2	Bedrijfsmentor evaluatie toegevoegd. Laatste styling aanpassingen en correctie datums van bijlages.	D.H.W. Bruschke

## Voorwoord

Voor u ligt een digitale kopie van mijn afstudeerdossier. Na zeventien weken te hebben gedocumenteerd, ontwikkeld en te hebben voorbereid op het volgende stadium in mijn leven, ben ik trots op wat ik uiteindelijk heb neergezet als eind- en tussenproducten.

Momenteel studeer ik af aan de Haagse Hogeschool op Zoetermeer, waar ik eerst drie jaar Software Engineering heb gestuurd. Daarna ben ik doorgroeid naar Game Development & Simulation, wat ik een ontzettend leuk en interessant gebied van programmeren vind. Ook geeft deze branch een goeie outlet om creatief te zijn en mezelf niet te limiteren tot alleen programmeren. Ik vind het erg leuk om in Blender 3D modellen te maken en deze als concept te gebruiken. Ook heb ik interesse in het schrijven van shaders in talen zoals HLSL.

Ik heb in het verloop van zeventien weken een probleem toegetakeld die voorkwam bij het bedrijf SyncVR Medical. Tijdens dit traject hebben ze mij begeleid en geholpen om dit probleem op te lossen. Hier heb ik over gedocumenteerd en zo veel mogelijk mijn keuzes in bepaalde technieken en toepassingen verantwoord.

Hierbij wil ik een aantal partijen met wie ik heb samengewerkt bedanken, en die er mede voor hebben gezorgd dat ik mijn afstudeertraject heb kunnen afleggen. Hieronder vallen mijn examinatoren, expert en begeleidend; met hun hulp hebben zij mij door de moeilijke eerste en laatste fase van het traject zeer geholpen.

Mijn medestudenten en ouders die me hebben bijgestaan tijdens lastige momenten waarin motivatie moeilijk te vinden was.

En tot slot mijn bedrijfsmentor Robbert Brouwer en de medewerkers binnen SyncVR Medical, wie mij hebben begeleid binnen het bedrijf en mij met open armen hebben opgenomen binnen de SyncVR bedrijfscultuur.

Ik wens u veel leesplezier toe.

David Bruschke

Hillegom, 4 juni 2021

## Referaat

David H.W. Bruschke (17023785), “Implementeren van wearables binnen Virtual Reality Rehabilitation Programs”, Utrecht, SyncVR Medical.

Binnen dit afstudeerdossier worden activiteiten, handelingen en besluiten, welke zijn uitgevoerd gedurende het afstudeertraject, beschreven en ondersteund. Deze zijn afgelegd in de periode tussen februari 2021 en juni 2021. De opdracht is opgesteld en uitgevoerd door een Game Development & Simulation student aan de Haagse Hogeschool ter Zoetermeer.

De opdracht betreft het implementeren van Bluetooth Low Energy wearables in Unity, communicatie tussen de wearable en Virtual Reality Android headsets en het ontwerpen en ontwikkelen van een Virtual Reality Android applicatie als Proof of Concept.








SyncVR Medical heeft het SyncVR Fit applicatie ontwikkeld dat patiënten binnen ziekenhuizen helpt om te rehabiliteren en terug in beweging te brengen met behulp van Virtual Reality applicaties (Virtual Reality Rehabilitation Programs).

De patiënten kunnen deze applicaties gebruiken door Virtual Reality headsets die worden ondersteund door het Android Operating System. Deze Virtual Reality headsets beschikken over Bluetooth, dat gebruikt wordt om te communiceren met de bijgevoegde controllers.

De opdracht draait om deze Bluetooth mogelijkheden te gebruiken om zodoende data, zoals hartslag of stappen, op te halen uit wearables en deze data te gebruiken binnen SyncVR Fit applicaties om het gedrag van de applicatie te beïnvloeden.

Het afstudeertraject is afgelegd onder de begeleider V.E. Broeren en onder de expert examiner D.R. Stikkolorum.

### Descriptoren

-  Unity
-  C#
-  Virtual Reality
-  Android
-  Java
-  Bluetooth (Low Energy)
-  Virtual Reality Rehabilitation Programs

## Inhoudsopgave

<b>1</b>	<b>INLEIDING.....</b>	<b>6</b>
<b>2</b>	<b>HET BEDRIJF .....</b>	<b>7</b>
	2.1 SYNCVR MEDICAL .....	7
	2.2 WERKSFEER .....	8
<b>3</b>	<b>OPDRACHTOMSCHRIJVING.....</b>	<b>9</b>
	3.1 AANLEIDING .....	9
	3.2 PROBLEEMSTELLING .....	10
	3.3 DOELSTELLING.....	10
	3.4 BEGINSITUATIE .....	10
	3.5 EINDPRODUCTEN.....	11
<b>4</b>	<b>THEORETISCH KADER.....</b>	<b>12</b>
	4.1 BLUETOOTH LOW ENERGY.....	12
	4.2 UNITY ANDROID JNI.....	15
	4.3 DESIGN PATTERNS .....	16
<b>5</b>	<b>METHODES EN WERKWIJZEN.....</b>	<b>18</b>
	5.1 KANBAN.....	18
	5.2 MODELEREN MET UML .....	20
<b>6</b>	<b>REQUIREMENTS.....</b>	<b>21</b>
	6.1 VERGAREN VAN REQUIREMENTS .....	21
<b>7</b>	<b>REALISEREN.....</b>	<b>23</b>
	7.1 ONTWERPEN VAN CONCEPTEN.....	23
	7.2 OPZETTEN VAN BLE .....	25
	7.3 COMMAND STACK.....	33
	7.4 HEART DRUMMING .....	37
	7.5 TESTEN EN FEEDBACK LOOP.....	38
<b>8</b>	<b>EINDPRODUCT .....</b>	<b>42</b>
<b>9</b>	<b>AANBEVELING .....</b>	<b>43</b>
	9.1 GIT EN GITHUB .....	43
	9.2 UNITY .....	44
<b>10</b>	<b>EVALUATIE EN REFLECTIE .....</b>	<b>45</b>
	10.1 EVALUATIE .....	45
	10.2 REFLECTIE .....	47
	10.3 BEROEPSTAKEN.....	48
<b>11</b>	<b>VERWIJZINGEN.....</b>	<b>51</b>
	11.1 FIGUREN.....	51
	11.2 TABELLEN.....	51
	11.3 AFKORTINGEN.....	52
<b>12</b>	<b>BIBLIOGRAFIE.....</b>	<b>53</b>

<b>13</b>	<b>BIJLAGEN .....</b>	<b>55</b>
13.1	UNITY BLE PACKAGE EN ANDROID BLE LIBRARY UML DESIGN (FASE 2) .....	56
13.2	UNITY BLE PACKAGE EN ANDROID BLE LIBRARY UML DESIGN (FASE 3) .....	57
	<b>PLAN VAN AANPAK .....</b>	<b>58</b>
	<b>HIGH CONCEPT DOCUMENTS .....</b>	<b>70</b>
	<b>AFSTUDEERPLAN.....</b>	<b>79</b>
	<b>EVALUATIE - BEDRIJFSMENTOR .....</b>	<b>85</b>

# 1 Inleiding

Dit is een afstudeerverslag voor het afstudeertraject van David Bruschke bij het bedrijf SyncVR Medical. Dit afstudeertraject en project komt voort uit de vraag om wearables te combineren met Virtual Reality Rehabilitation Programs voor de fitness georiënteerde afdeling onder SyncVR Medical, genaamd SyncVR Fit. Binnen SyncVR Fit is al vaker gekeken naar het combineren van wearables en Virtual Reality mogelijkheden; deze wearables hebben al als doel om Virtual Reality te ondersteunen.

De focus van het afstudeertraject is om wearables, zoals hartslagmeters en stappentellers, te laten communiceren met een in Unity ontwikkelde fitness georiënteerde toepassing die Virtual Reality Rehabilitation ondersteunt.

Dit document bevat gemaakte keuzes die de opgestelde eind- en tussenproducten betreffen, vindingen die deze producten ondersteunen, bijlages met aanvullende informatie, een reflectie en evaluatie over de gemaakte producten, en het proces dat hieraan gekoppeld is.

Het document begint met een omschrijving van het bedrijf, waarna de theoretische vindingen worden uitgelegd. Zo wordt er eerst duidelijk gemaakt waar het afstudeertraject is afgelegd en welke technieken er zijn gebruikt om de eindproducten te realiseren. Dit hoofdstuk bevat ook de requirements voor deze producten.

Afsluitend wordt er gereflecteerd over het volledige afstudeertraject, de keuzes die zijn gemaakt voor de eind- en tussenproducten en de persoonlijke ontwikkeling in de vorm van beroepstaken die gekoppeld zijn aan de Game Development & Simulation HBO opleiding.

Veel termen worden afgekort binnen dit document. Ter verduidelijking worden deze afkortingen en termen genoemd en verklaard in hoofdstuk *11 Verwijzingen* (p. 51).

## 2 Het Bedrijf

Om een idee te geven hoe het afstudeertraject is verlopen, behandel ik in dit hoofdstuk: het bedrijf, de bedrijfsmentor en de werksfeer binnen SyncVR Medical.

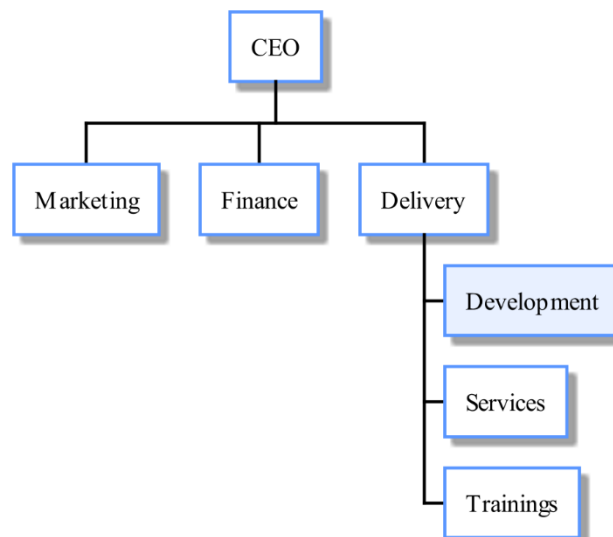
### 2.1 SyncVR Medical

Al jaren ontwikkelt SyncVR Medical verschillende toepassingen voor medische doeleindes. Elke toepassing richt zich op een eigen doel. Zo is SyncVR Relax & Distract gericht op het verminderen van acute pijn of angst tijdens medische interventies, SyncVR Fit voor het spelenderwijs beweging stimuleren, en nog vele andere toepassingen<sup>1</sup>.

Het SyncVR Medical platform maakt het mogelijk om Virtual Reality (hierna te noemen als VR) te implementeren in het gehele ziekenhuis: van hardware, software en workshops tot afspraken met de ondersteunende diensten en een hygiëne-protocol. Met 75+ aangesloten ziekenhuizen, waaronder Zuyderland, Spaarne en het Erasmus MC, en meer dan 3000 patiënten die zijn behandeld met VR, zijn zij het grootste VR platform van de medische zorg in Europa. Door middel van applicaties die zijn ontwikkeld met de Unity game engine, willen ze de kwaliteit van de medische zorg verbeteren.

De toepassingen van SyncVR Medical zijn niet alleen gebaseerd op inzichten van patiënten en zorg-professionals, maar ook op wetenschappelijk onderzoek. Deze bestaan uit internationale- en door SyncVR Medical/SyncVR Medical partners uitgevoerde onderzoeken.<sup>2</sup>

SyncVR Medical bestaat uit een compact team dat onderverdeeld kan worden in drie verschillende lagen. Deze lagen worden visueel uitgebeeld in het organogram in *Figuur 1*, waarin het lichtblauw gekleurde onderdeel de sub-laag is waarin het afstudeertraject is afgelegd. De *Delivery* laag houdt zich bezig met het ontwikkelen van nieuwe producten, het bieden van services, zoals trouble shooting wanneer een product niet werkt of er problemen zijn met de geleverde VR headsets, en het geven van cursussen en trainingen aan ziekenhuizen die voor het eerst worden geïntroduceerd aan VR en VRRP.



Figuur 1: Organogram van het bedrijf SyncVR Medical

<sup>1</sup> [Homepagina](#)

<sup>2</sup> [Onderzoek](#)



De heer R. Brouwer, de bedrijfsmentor voor het traject, is onderdeel van de CEO laag en houdt zich bezig met onderdelen van alle lagen daaronder, zo ook de *Development* laag. Door te starten met een aantal gesprekken over het bedrijf en de bedrijfscultuur binnen SyncVR Medical, legt de heer Brouwer een goede basis voor het verdere verloop van het traject. Ook leidt hij de wekelijkse team-meetings van SyncVR Fit, waarin hij de leider rol aanneemt om overzicht te krijgen waar elk teamlid die week aan werkt en geeft hier eventueel feedback op wanneer sommige taken een hogere prioriteit hebben dan andere.

## 2.2 Werksfeer

Bij SyncVR Medical hangt een ongedwongen werksfeer. Er wordt op een informele manier met elkaar gesproken, wat het makkelijk maakt om met andere collega's te praten. Het team van SyncVR Medical is vrij groot, en de vele stagiaires zijn verspreid over de verschillende bedrijfslagen. Er zijn acht vaste werknemers binnen SyncVR Medical en hebben een selectie aan tien à vijftien stagiaires. Het kantoor is relatief klein, en de afstand vanaf Hillegom naar Utrecht en de lockdowns belemmerden een frequent bezoek.

## 3 Opdrachtomschrijving

In dit hoofdstuk wordt uitgelegd welke opdracht er tijdens het afstudeertraject wordt behandeld, hoe deze tot stand is gekomen en welke activiteiten waren uitgevoerd om dit probleem op te lossen.

### 3.1 Aanleiding

Virtual Reality Rehabilitation Programs (hierna te noemen als VRRP) is de techniek achter het aanbieden van traditionele rehabilitatie aan patiënten in de vorm van VR. Door bewegingsoefeningen die gewoonlijk door een therapeut worden overzien, worden deze oefeningen in VRRP aangeboden als oefeningen binnen Virtual Reality. Dit kan gedaan worden door bijvoorbeeld een spel uit te voeren waarbij patiënten een bepaalde beweging herhalen om net als een reguliere (niet VR) bewegingsoefening, de juiste spieren en bewegingen te simuleren.

SyncVR Medical biedt een pakket aan voor ziekenhuizen en andere zorginstanties om deze VRRP te integreren door middel van een pakket van VR oefeningen gebundeld in een applicatie, genaamd 'SyncVR Fit'. Deze applicatie wordt beschikbaar gesteld voor verschillende soorten VR headsets, zoals de Oculus Quest, HTC Vive Focus en de Pico Neo 2. Dit zijn VR headsets die geen externe apparatuur nodig hebben om hun 3-Dimensionale positie te bepalen binnen de fysieke omgeving. Dit is ontzettend handig voor ziekenhuis en zorginstanties. Deze hoeven dan namelijk geen apparatuur op of ruimtes beschikbaar te stellen voor dit soort apparatuur, maar kunnen de VR headsets gebruiken zonder dat er veel moeite gedaan hoeft te worden om deze gereed te maken.

Iets wat deze (en andere) VR headsets niet ondersteunen, is het vergaren van data zoals de hartslag van de gebruiker, hoeveel stappen hij heeft gezet of hoeveel calorieën er zijn verbrand tijdens het gebruik van de headset. Dit is data wat van pas kan komen voor therapeuten en andere zorginstanties, aangezien deze data aangeeft of een gebruiker veel moeite heeft met een bepaalde oefening, of dat deze te makkelijk is en er opgeschaald kan worden naar andere bewegingen.

Hierdoor wil SyncVR Medical deze VR headsets combineren met wearables die dit soort data wél van de gebruiker kan uitlezen. Deze wearables kunnen dan de data van de gebruiker verzamelen of direct uitlezen en deze direct naar de VR headset sturen. Zo kan er tijdens de oefening iets gedaan worden met de data, zoals het terugschalen van de moeilijkheidsgraad van de oefening als de hartslag van een patiënt te hoog raakt.

Hierbij was de opdracht om wearables te kunnen verbinden met de SyncVR Fit VRRP en andere SyncVR VR applicaties, welke gebouwd zijn in de Unity Game Engine. Ook moet deze oplossing compatibel zijn met de VR headsets die worden aangeboden door SyncVR Medical (Oculus Quest, HTC Vive Focus, Pico Neo (2) ). Hierbij wordt er ook een VRRP oefening ontwikkeld om te gebruiken binnen SyncVR Fit, waarin de verbinding met een wearable mogelijk is en waarvan de combinatie met de wearable een toevoeging kan geven aan de gameplay.

## 3.2 Probleemstelling

VR headsets hebben van nature geen integratie met wearables en kunnen dus ook niet direct verbinding maken met een wearable. Unity3D ondersteunt geen out-of-the-box oplossing voor verbinding met wearables; dus een connectie maken met een wearable is in de huidige staat bij SyncVR Medical en haar applicaties niet mogelijk.

Origineel was er een plan opgesteld om dit probleem aan te pakken door de huidige stand van VRRP, wearables en eventuele connecties tussen VR en wearables te analyseren en te onderzoeken. Deze resultaten zouden dan het product ondersteunen. Dit product is dan een oplossing voor het verbinden van VR headsets en wearables. Dit plan was opgesteld en ter controle aan de afstudeermentor afgeleverd, maar werd in de eerste week van het afstudeertraject aangepast. Het aangepaste plan omvat het doen van deskresearch en prototyping onderzoeksmethodes om de eindproducten te ontwikkelen en hierbij VR headsets de mogelijkheid geven om verbinding te maken met een wearable.

## 3.3 Doelstelling

Het doel van het afstudeertraject en de eindproducten, is om SyncVR Medical applicaties die ontwikkeld zijn in Unity ondersteuning te geven om een verbinding te maken met een wearable en de data die de wearable verzamelt, uit te lezen.

Hierbij wordt een VRRP oefening ontwikkeld voor SyncVR Fit als Proof of Concept (vaak afgekort als POC) die de mogelijkheid heeft om de voorgenoemde oplossing voor het verbinden van een wearable te gebruiken, om een onderdeel binnen de VRRP oefening te manipuleren of te ondersteunen.

## 3.4 Beginsituatie

De oorspronkelijke staat bij SyncVR Medical waarbij het gaat om het koppelen van SyncVR Fit op Virtual Reality headsets met wearables, was nog niet opgezet. SyncVR Fit is een team dat de SyncVR Fit applicatie ontwikkelt. Binnen deze applicatie zijn acht verschillende oefeningen die VRRP om patiënten in ziekenhuizen of andere zorginstanties te helpen en beweging bij deze patiënten te stimuleren. Deze oefeningen zijn samengevoegd tot één applicatie, zodat deze makkelijker te distribueren valt en eventuele Unity assets (zoals Textures, Modellen of Shaders) binnen hetzelfde project gedeeld kunnen worden. Dit project staat dan ook op GitHub, waarbij nieuw ontwikkelde oefeningen een eigen branch binnen de repository hebben.

Voor het afstudeertraject is SyncVR Medical begonnen met het ontwikkelen van een Bluetooth Agent, waarmee het scherm van de VR-headset gedeeld wordt met een Spectator applicatie. Ook kunnen knoppen binnen de applicatie ingedrukt worden met behulp van deze Spectator applicatie, zodat toezichthouders die met de patiënt de SyncVR Fit oefeningen ondernemen kunnen ingrijpen als dat nodig is. Deze Bluetooth Agent maakt geen gebruik van Bluetooth Low Energy, maar heeft wel invloed op de uiteindelijke requirements voor dit afstudeertraject.

Ook werd er binnen de SyncVR Fit repository omgeschakeld naar een andere plugin om de applicaties te compileren naar verschillende VR headsets. Origineel werd er per VR headset een aparte plugin gebruikt om naar dat platform te compileren, maar door dit te vervangen met de Unity XR plugin is er maar één plugin nodig om te compileren naar de verschillende platforms. Deze verandering was al ingezet vóór het afstudeertraject en is tijdens het afstudeertraject gemerged naar een nieuwe SyncVR Fit repository.

### 3.5 Eindproducten

Binnen het afstudeertraject is er een aantal deliverables opgesteld. Dit zijn producten die aan het eind van, of tijdens, het afstudeertraject worden opgeleverd aan het bedrijf. Deze eindproducten zijn tijdens het traject omgevormd. Origineel wordt er in het Plan van Aanpak (hierna te noemen als PVA) beschreven dat er een Unity package wordt opgeleverd met communicatie tussen Unity en BLE op Android. Dit is tijdens het traject omgevormd naar het opleveren van twee verschillende producten, de *Unity Bluetooth Low Energy Package* en de *Android Bluetooth Low Energy Library*. Wat deze libraries precies omvatten wordt beschreven in *Tabel 1 (p. 11)*. In *Tabel 2 (p. 11)* worden de tussenproducten beschreven. Om te voorkomen dat de eindproducten in verschillende vormen genoemd worden binnen het Afstudeerdossier, worden ze afgekort weergegeven. De origineel opgestelde deliverables zijn terug te vinden in *Bijlage A: Plan van Aanpak (p. 58)*.

*Tabel 1: Tabel en afkortingen voor eindproducten*

Afkorting	Volledige Benaming	Beschrijving
<i>UBLEP</i>	Unity Bluetooth Low Energy Package	De code en onderdelen die vallen onder de Unity Bluetooth Low Energy Package. C# scripts die communicatie leggen tussen Unity en de Android Bluetooth Low Energy Library.
<i>ABLEL</i>	Android Bluetooth Low Energy Library	De code en onderdelen die vallen onder de Android Bluetooth Low Energy Library. Java scripts die de onderliggende Bluetooth Low Energy protocollen van Android aanroept, gebruikt en doorgeeft aan de Unity Bluetooth Low Energy Package.
<i>HEDR</i>	Heart Drumming	Het Proof of Concept dat wordt gebouwd om de mogelijkheden van UBLEP weer te geven. Dit is een Unity applicatie voor SyncVR Fit met de mogelijkheid om UBLEP te gebruiken.

*Tabel 2: Tabel en afkortingen voor tussenproducten*

Verkorting	Volledige Benaming	Beschrijving
<i>Music Editor</i>	Music Unity Editor Window	Een Unity EditorWindow die het makkelijk maakt om nieuwe nummers te voorzien van een vooraf ingestelde beat. Ook geeft deze de opties om een delay aan de start toe te voegen en de BPM te bepalen.
<i>PVA</i>	Plan van Aanpak	Een document waarin vooraf is beschreven wat er wordt opgeleverd, hoe deze tot stand wordt gebracht en een algehele planning van het verloopt van het afstudeertraject.

## 4 Theoretisch Kader

In dit hoofdstuk worden theoretische vindingen beschreven die tijdens het afstudeertraject aan bod kwamen. Binnen dit theoretisch kader vallen nieuwe termen, technieken en apparaten die extra uitleg behoeften, voordat deze begrepen en gebruikt konden worden. Ook leveren de individuele paragrafen informatie die nodig is om de rest van de onderdelen binnen het afstudeerdossier te begrijpen.

### 4.1 Bluetooth Low Energy

De keuze voor Bluetooth Low Energy (ook aangeduid als BLE) viel vrijwel direct. BLE heeft een ontzettend groot marktaandeel vergeleken met andere draadloze communicatie middelen (Søderholm, 2016), waardoor het vrijwel dé standaard is geworden voor een wearable, om BLE te ondersteunen. Andere communicatiemiddelen zijn bijvoorbeeld Wi-Fi, NFC en ANT, waarvan de laatste twee zijn ontworpen voor communicatie op korte afstanden. Op Wi-Fi na, worden deze communicatiemiddelen niet ondersteund op de VR headsets die binnen SyncVR Medical worden gebruikt.

De VR headsets die worden gebruikt bij SyncVR Medical worden ondersteund door het Android Operating System (hierna te noemen als OS) en beschikken over een Bluetooth chip om te communiceren met de controllers die bij de headset horen. Android ondersteunt BLE natively sinds Android versie 4.3, waardoor de VR headsets met de ingebouwde Bluetooth chip ook BLE ondersteunen. Dit maakt de BLE de perfecte communicatie protocol om een wearable te verbinden met een VR headset.

BLE is een techniek dat is ontwikkeld om dezelfde mogelijkheden van klassiek Bluetooth aan te bieden, ingericht als een instantie van Bluetooth dat weinig energy verbruikt. De ontwikkeling van BLE was gericht op apparaten die draadloos communiceren, maar gelimiteerd zijn door de stroom die ze kunnen verbruiken en de ruimte die ze bieden. Dit zijn bijvoorbeeld wearables, draadloze camera's of telefoons, die allemaal gelimiteerd zijn door hun grootte ofwel de bijgaande batterij (gebrek aan een constante oplader). Hiervoor zijn de kleine en energie efficiënte BLE chips een goed alternatief. Ook is er voldoende ondersteuning vanuit de Operating Systems (ook als OS afgekort), met de meest gebruikte OS's die standaard BLE ondersteunen, zoals Windows (Phone, 8 en 10), Android, IOS en Linux.

#### 4.1.1 Generic Attribute Profile

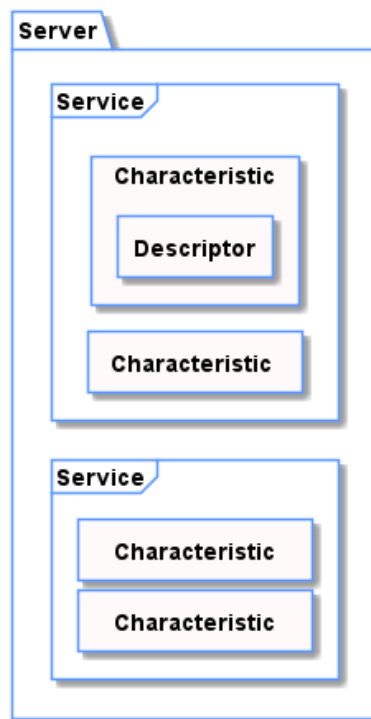
Binnen Bluetooth bestaan verschillende profielen om communicatie te standaardiseren en hiermee verschillen te voorkomen, zodat diverse apparaten met elkaar kunnen communiceren via hetzelfde Bluetooth protocol. Binnen BLE wordt de Generic Attribute profiel (afgekort als GATT) gebruikt. GATT is een profiel dat definieert op wat voor wijze het BLE apparaat data verstuurt en ontvangt naar zijn services en characteristics (Townsend, 2021). Onder GATT ligt nog een protocol dat de adressen van deze services en characteristics opslaat in een tabel, genaamd Attribute Protocol. Dit protocol zorgt ervoor dat de cliënt die verbinding maakt met het BLE apparaat, direct een lijst heeft van de verschillende kanalen die deze kan aanroepen. Een beperking van GATT is dat er slechts één apparaat verbonden kan zijn. Nadat een apparaat verbonden is met de GATT peripheral, stopt de peripheral met het zenden van advertisements om verbinding te maken, wat de reden is dat er geen andere apparaten aan verbonden kunnen worden.

Heart Rate Measurement Service: 0000 [180D] -0000-1000-8000-00805F9B34FB  
↳ characteristic: 0000 [2A37] -0000-1000-8000-00805F9B34FB

*Figuur 2: Bluetooth Low Energy Heart Rate Measurement profiel z'n unieke identifiers*

Binnen het GATT zijn er ook sub-profielen en standaarden voor het communiceren van gelabelde data (Bluetooth SIG Proprietary, 2020). Dit zijn regels die aangeven onder welke specifieke Universally Unique Identifier (verder te noemen als UUID) een service en een characteristic valt. Ook binden deze regels een limiet aan wat voor soort data een characteristic kan bevatten en onder welke service deze characteristic valt. Zo bestaat er een Heart Rate Measurement profiel dat aangeeft dat de service waarin de characteristics staan om hartslag data uit te lezen, een UUID heeft dat zich identificeert met '180D' en de characteristic '2A37' (Silicon Labs, 2020). Deze 16-bit UUID's zijn onderdeel van een totale 128-bit lange identifier. Dit is een UUID voor services en characteristics die wordt gedekt door een BLE profiel. In *Figuur 2* worden de UUID's van de Heart Rate Measurement profiel aangegeven, waarbij de gekaderde onderdelen van de UUID de unieke onderdelen zijn die de service en characteristic identificeren. Ook valt te zien dat de UUID's hetzelfde zijn, met het gekaderde onderdeel als enige verschil.

Deze profielen komen van pas tijdens het afstudeertraject, omdat ze een direct overzicht geven tot een bepaald data type waartoe characteristics behoren. Zo beschrijven deze profielen op welke characteristic hartslag data standaard zit, maar ook in welke vorm deze data teruggeven moet worden aan gebruikers (Little-Endian, Big-Endian, Hexadecimaal, etc.). Zo kan er in UBLEP een makkelijke functie zijn die automatisch een characteristic aanroept van een bepaald profiel.



*Figuur 3: Opzet van een BLE apparaat met een Generic ATtribute Profile*

In *Figuur 3* staat een Component diagram dat het BLE GATT profiel in kaart brengt. Het GATT profiel is onderverdeeld onder meerdere services, waarbij services weer bestaan uit meerdere characteristics.

In de Component diagram is tevens een descriptor weergegeven. descriptors zijn optioneel binnen GATT, maar zoals de naam in het Engels aangeeft, beschrijven ze het gedrag van een characteristic (Description (EN) = Beschrijving (NL)). characteristics kunnen zich namelijk op negen verschillende manieren gedragen (Kevin Townsend, 2014), maar de manier hoe ze zich gedragen en wat voor permissies ze hebben, staat in de descriptor. In *Tabel 3 (p. 14)* is een lijst aan type gedrag weergegeven, dat een characteristics kan hebben die relevant waren en aan bod kwamen tijdens het afstudeertraject.

*Tabel 3: Relevant gedrag van BLE characteristic en de data binnen z'n descriptor*

Aanduiding	Beschrijving
<i>Read</i>	De waarde van de characteristic waar de descriptor meta-data over bevat kan uitgelezen worden. De descriptor bevat meta-data over welk formaat waarde de characteristic bevat (Hexadecimaal, Binair, Decimaal).
<i>Write</i>	De waarde van de characteristic waar de descriptor meta-data over bevat kan naar geschreven worden. De descriptor bevat meta-data over welk formaat waarde de characteristic bevat.
<i>Notification / Indication</i>	<p>Wanneer de waarde van deze characteristic verandert, kan een apparaat dat verbonden staat met de GATT server een notificatie ontvangen dat de waarde van de characteristic is gewijzigd of wanneer de peripheral zelf een notificatie wil geven over de waarde in de characteristic.</p> <p>De descriptor bevat meta-data over welk formaat waarde de characteristic bevat en om hoeveel tijd de characteristic automatisch zijn waarde doorgeeft.</p>

Aangezien de relevante wearables voor het afstudeertraject allemaal werken met BLE, wordt het GATT profiel gebruikt, om zo min mogelijk energie te gebruiken. Zo wordt er ook gebruik gemaakt van de services, characteristics en descriptors dit bij het GATT profiel horen.

Deze verschillende soorten gedrag van characteristics heeft ook betrekking op de uiteindelijke functies die ondersteund kunnen worden binnen de eindproducten ABLEL en UBLEP, aangezien deze een aantal basis functies moeten ondersteunen. Vanuit UBLEP moet de mogelijkheid bestaan om deze functies van een descriptor aan te roepen, of automatisch naar een bepaald gedrag zetten, aan de hand van de command dat binnen komt in ABLEL.



#### 4.1.2 Toepassen van Wearables binnen SyncVR Fit

Het uiteindelijke doel van dit stagetraject is om een oplossing te vinden om wearables te verbinden met een VRRP applicatie op een VR headset. Uit deze wearable kan dan data uitgelezen worden en in de applicatie gebruikt worden. Door een kader te trekken rond welke wearables geschikt kunnen zijn om dit doel te bereiken, is er gezocht naar een wearable waarmee verbinding gemaakt kan worden met BLE, die hartslag data kan uitlezen en makkelijk gedragen kan worden.

Na een aantal wearables te hebben genoteerd, bezit *Xiaomi Band 3* (hierna te noemen als MiBand 3) alle functies die hierboven zijn beschreven en blijkt het goedkoopst. Ook bestaan er veel artikelen over de werking van de MiBand 2 en 3, waardoor het eenvoudig was om informatie te vinden over de achterliggende werking van deze wearables.

Uiteindelijk heeft de keuze voor een specifieke wearable niet veel invloed op het stagetraject, als er maar data uit de wearable gelezen kan worden en deze verbonden kan worden met de VR headset. Aangezien de gekozen communicatie wijze BLE is geworden, wordt het een wearable dat communiceert via BLE. Qua data is het handig als dit zinvolle data is voor een ziekenhuis of andere zorginstelling, waardoor hartslag een goede databron is om te gebruiken. De MiBand 3 beschikt over al deze functionaliteiten.

## 4.2 Unity Android JNI

Unity beschikt over een API waarmee direct gecommuniceerd kan worden vanaf Unity naar de Android Java Virtual Machine (hierna te noemen als JVM) dat de Unity Android applicatie ondersteunt. De techniek om te communiceren vanuit een (voor Java-) onbekende taal naar een Java code dat draait binnen de Java Runtime, wordt de Java Native Interface (verder te noemen als JNI) genoemd (Bealdung, 2020). Dit houdt in dat talen zoals C, C#, Javascript en andere programmeertalen kunnen communiceren met Java code terwijl de Java code aan het draaien is binnen de Java Runtime. Met JNI kunnen deze programmeertalen methodes, classes en andere libraries binnen de Java Runtime aanroepen.

Omdat Unity ook intern protocollen en andere methodes van Android aanroept, beschikt een gecompileerde Unity applicatie over een JNI, wat de mogelijkheid geeft om geschreven code binnen Unity te laten communiceren met eventuele Java libraries die mee gecompileerd zijn. Dit biedt Unity aan met een C# wrapper om deze JNI heen, zodat de JNI makkelijker is aan te roepen binnen de code dat in Unity geschreven wordt.

Ondanks de goede kanten van JNI, zijn er wel problemen met performance geconstateerd (Arpe, 2020). Deze problemen met performance zijn er vanwege: de verschillen tussen de manieren waarop programmeertalen hun data in het geheugen opslaan en limitaties die de JVM biedt, en het feit dat native Java geen Just-In-Time gecompileerde taal is op Android. Dit zorgt ervoor dat wanneer er naar een native methode binnen Java wordt gevraagd via de JNI, de JVM de positie van de native methode buiten zijn normale (JIT) werkomgeving moet vinden (Bealdung, 2020), maar binnen Ahead-Of-Time native libraries. Dit veroorzaakt performance problemen bij vraag naar native Java methodes die alleen binnen al-gecompileerde libraries aanwezig zijn. Dit komt omdat de JVM moet zoeken naar de positie van de native methode om deze aan te roepen.

Het gebruik van de JNI maakt het mogelijk om BLE te integreren in een Unity applicatie en zal ook gebruikt worden tijdens dit stagetraject. Hoe de JNI is geïntegreerd in het eindproduct UBLEP en ABLEL wordt verder toegelicht in het hoofdstuk 7.2 *Opzetten van BLE* (p. 25).



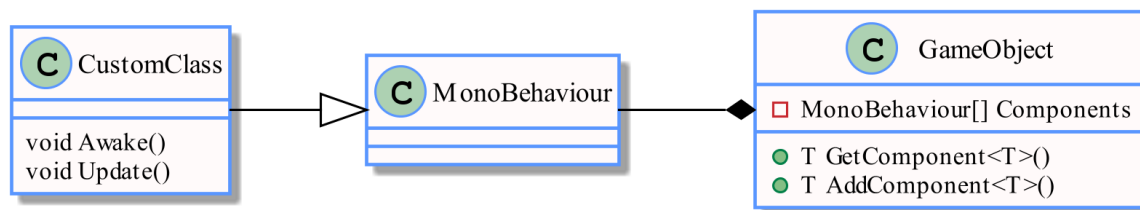
## 4.3 Design Patterns

Een design pattern is een herbruikbare oplossing voor een terugkomend probleem binnen software design. Omdat design patterns universeel in elke programmeertaal te gebruiken zijn, zijn ze makkelijk toe te passen binnen elk software ontwikkelingsproject. Tijdens het afstudeertraject is gebruikgemaakt van een aantal software design patterns. Ook is er een aantal patterns die gebruikt moest worden, omdat de manier waarop sommige applicaties hun services aanbieden en bepaalde manier van programmeren, forceert.

### 4.3.1 Composite Pattern

Standaard wordt binnen Unity gewerkt met een class genaamd *MonoBehaviour*. Als een nieuwe class binnen Unity wordt aangemaakt dan erft deze automatisch over van de *MonoBehaviour* class. Na wat deskresearch te hebben gedaan naar de innerlijke werking van deze class en waarom Unity deze toepast op nieuw aangemaakte classes, was de conclusie dat dit een Composite Pattern is. Een Composite pattern houdt in dat een object uit verschillende sub-objecten bestaat met elk een eigen taak om te vervullen (Refactoring Guru, 2020). Samen (het object in context met alle sub-objecten) wordt dit gezien als één object. Uitgetekend zou dit resulteren in een boom structuur, waarbij de stam verschillende blad-componenten heeft die allemaal een eigen functie hebben binnen de algehele structuur.

Unity implementeert Composite Pattern door een basis class -genaamd *MonoBehaviour*- te hebben, waar elk sub-object van overerft. Alleen classes die overerven van de *MonoBehaviour* class kunnen toegevoegd worden aan een *GameObject*. Dit maakt *GameObject* de stam en elk *MonoBehaviour* class die wordt toegevoegd aan deze *GameObject*, een blad-component van de *GameObject*. Binnen het voorbeeld van *Figuur 4* valt dit te zien in de vorm van een UML diagram. Hierbij is een voorbeeld class aangemaakt, genaamd *CustomClass*. Dit is een fictionele class die als voorbeeld dient. *CustomClass* overerft van de *MonoBehaviour* class, waardoor het als onderdeel toegevoegd kan worden aan *GameObject*. Classes die niet overerven van Unity's *MonoBehaviour* class kunnen niet als onderdeel worden toegevoegd aan *GameObject*.

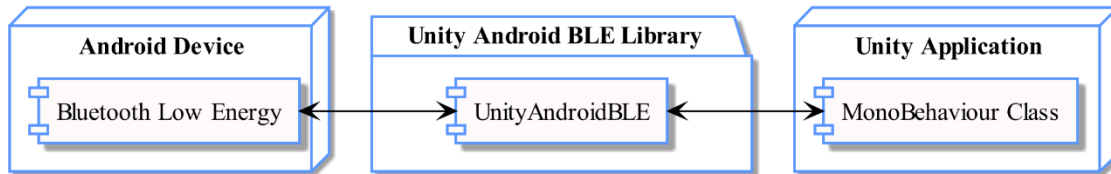


Figuur 4: UML van Unity's Composite Pattern implementatie

Er kan ook zonder de *MonoBehaviour* class geprogrammeerd worden, maar dit zou in de weg zitten tijdens het stagetraject, vanwege de manier waarop ABLEL communiceert met UBLEP. De manier waarop gecommuniceerd wordt tussen deze twee producten wordt in detail uitgelegd in 7.2 *Opzetten van BLE* (p. 25), maar is kort samengevat: ABLEL communiceert met UBLEP door gebruik te maken van een methode binnen de *MonoBehaviour* class, wat er voor zorgt dat informatie tussen classes gestuurd kan worden die geen referentie naar elkaar hebben, behalve de naam van de *GameObject* waar de class aan gekoppeld is en de naam van de methode die wordt aangeroepen.

### 4.3.2 Adapter Pattern

Omdat Unity niet direct met de Bluetooth Low Energy protocollen van Android kan communiceren, en er wederzijds geen communicatie kan ontstaan over deze protocollen, is gebruikgemaakt van het Adapter pattern om Unity met BLE te laten communiceren. Door een Android Java library (ABLEL) te schrijven tussen Unity en Android's BLE protocollen, ontstaat wederzijdse communicatie, wat ervoor zorgt dat BLE informatie vanuit Android naar Unity gecommuniceerd kan worden.



Figuur 5: Simpele notatie van het Adapter Pattern

Ook binnen Unity wordt gebruikgemaakt van Adapter Patterns, namelijk in de Android JNI (zie 4.2 *Unity Android JNI* (p. 15)). Met de Android JNI kan de gecompileerde code van Unity communiceren met de native Android Java code, of eventuele aanwezige Java libraries die zijn mee gecompileerd met de applicatie. Deze JNI maakt het mogelijk om de hiervoor benoemde Adapter Pattern te laten werken, omdat de JNI zorgt voor de communicatie tussen Unity en de Unity Android BLE Library (ABLEL).

### 4.3.3 Observer Pattern

In het vorige hoofdstuk is er uitgelegd hoe BLE werkt, en de werking van de verschillende soorten gedrag dat een characteristic kan tonen, waaronder het Notification / Indication gedrag. Dit gedrag is te vergelijken met het Observer Pattern; er gaat een notificatie af zodra een actie wordt ondervonden op hetgeen dat geobserveerd wordt. In het geval van BLE wordt er een notificatie of indicatie gegeven wanneer de waarde van de characteristic verandert. Dit design pattern wordt uiteindelijk ook toegepast binnen UBLEP en ABLEL, omdat deze notificatie characteristics de hartslag metingen van een wearable bevatten.

Ook buiten de het gedrag van de characteristics wordt het Observer pattern gebruikt. Om alle BleCommands te laten werken in het eindproduct UBLEP wordt het Observer pattern gebruikt om de eventuele events door te geven wanneer er iets gebeurt binnen ABLEL. Zo wordt er een methode uitgevoerd wanneer een wearable wordt verbonden en wanneer de verbinding hiervan is verbroken. Dat roept dan de *DeviceConnected* en *DeviceDisconnected* methode aan. Door dit pattern toe te passen kunnen er onderdelen afgehandeld worden zodra een event plaatsvindt.

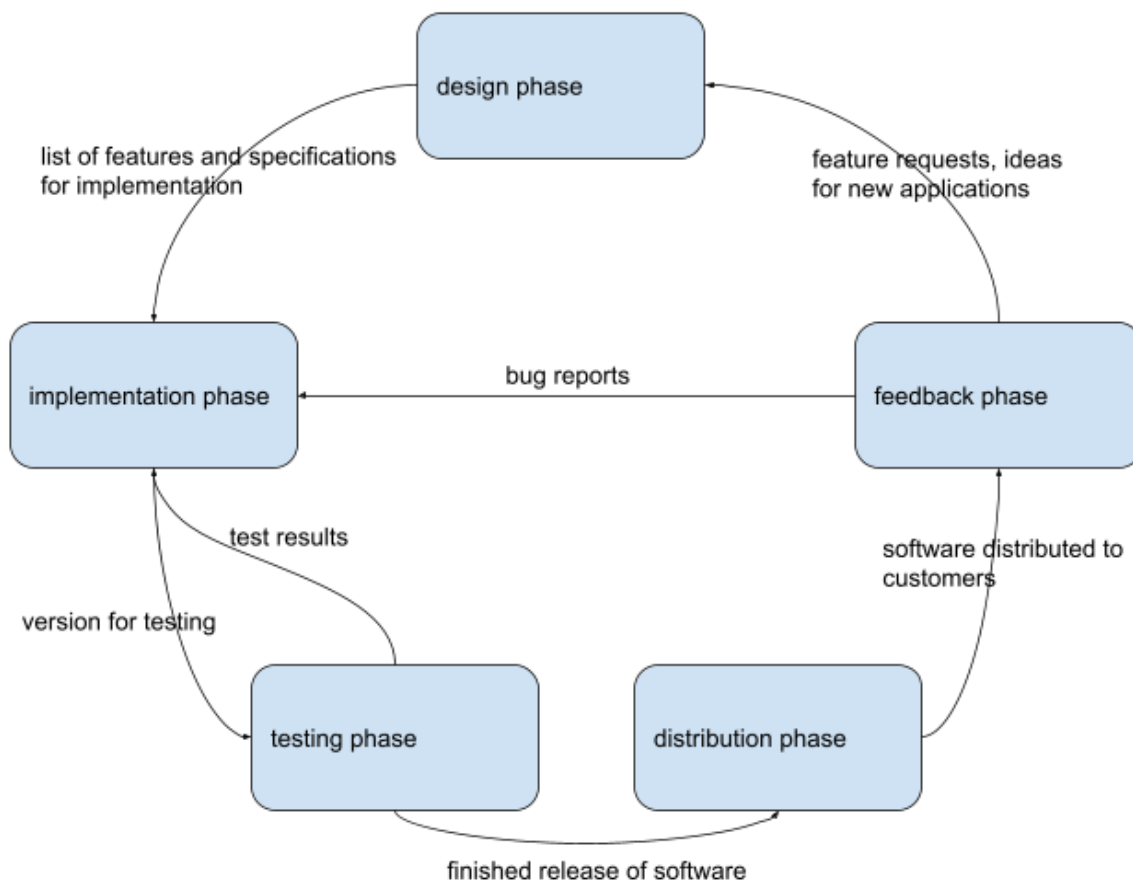
## 5 Methodes en Werkwijzen

In dit hoofdstuk worden de gebruikte methodes en werkwijzen besproken. Hierbij worden verschillende technieken, softwareontwikkelingsmethodes en versiebeheer in detail uitgelicht.

Ook wordt er gesproken over de gebruikte softwareontwikkelingsmethode; op welke manier deze is toegepast en waarom juist déze is toegepast. Hierbij wordt versiebeheer meegenomen, omdat de gebruikte softwareontwikkelingsmethode goed aansluit op het platform waarop versiebeheer plaatsvindt.

### 5.1 Kanban

Kanban is een methode om het ontwikkelproces op te delen in eenvoudige taken en deze te verdelen over verschillende kolommen die context geven aan de huidige staat van de taak. In combinatie met een Agile ondersteund softwareontwikkelingsmethode, sluit dit goed aan bij de manier van werken bij SyncVR Medical. Bij SyncVR Medical wordt namelijk gewerkt in verschillende ontwikkelingsfases, waarbij elke fase zijn eigen taken heeft. Hierdoor vallen de taken makkelijk op te delen in hun eigen kolom met een eigen voorstelling van ontwikkelingsfase.



*Figuur 6: Fases voor het ontwikkelen van applicaties binnen SyncVR Medical  
Schematic of software development process, By SyncVR Medical, 2020*

In *Figuur 6* worden de ontwikkelingsfasen visueel weergegeven. Deze fasen worden meerdere keren herhaald tijdens het ontwikkelen van een applicatie of feature binnen SyncVR Medical. Dit komt omdat ziekenhuizen en andere zorginstanties andere eisen hebben aan applicaties of features, waardoor er soms snel een feature aangepast moet worden. Door constant deze fasering aan te houden blijft de ontwikkeling van de applicatie gaande en kunnen features worden weggehaald als deze onnodig zijn of uitgebreid worden, als het ziekenhuis hier behoefte aan heeft.

Deze fasering wordt ook toegepast tijdens het afstudeertraject, mede omdat het ontwikkelingsplatform van SyncVR Medical al is ingericht rond het ontwikkelen om deze methode, maar ook omdat het past binnen de stappen die nodig zijn om de deliverables over te kunnen dragen. Er wordt gewerkt binnen het SyncVR Fit team, maar omdat door ieder lid van dit team een eigen oefening of onderdeel voor de SyncVR Fit applicatie wordt ontwikkeld, past Kanban in combinatie met Agile goed bij het ontwikkelproces.

Als er in een groep gewerkt zou worden dan zou Scrum als iteratieve softwareontwikkelingsmethode beter werken, omdat het bij Scrum belangrijk is om gedeeltes van het product per iteratie op te leveren om gezamenlijk het volledige project te ontwikkelen. Andere softwareontwikkelingsmethodes zoals de watervalmethode zou niet kunnen werken binnen het ontwikkelen in SyncVR Medical, omdat met deze methode ervan uit wordt gegaan dat technieken en methodes niet veranderen over de periode waarin je ontwikkelt, maar omdat er constant feedback vanuit het team en ziekenhuizen komt en hierbij dus technieken of onderdelen binnen de applicatie aangepast moet worden, past de watervalmethode (en afgeleide van) niet bij deze faseringsmethode dat wordt toegepast binnen het ontwikkelen bij SyncVR Medical en daarbij ook niet bij het afstudeertraject.

### 5.1.1 Versiebeheer

Onder versiebeheer valt het bewaren van wijzigingen in bestanden om op deze manier verschillende versies van dat bestand, of meerdere bestanden, bij te houden. De meest gebruikte vorm van versiebeheer is Git, een systeem dat sinds 2005 wordt ontwikkeld (Gehman, 2019). Binnen SyncVR Medical wordt ook Git gebruikt op het platform genaamd GitHub. Qua features loopt GitHub achter op de Git platform market als het gaat om features die het platform zelf ter beschikking stelt. Met bijvoorbeeld een gebrek aan een ingebouwd iteratie systeem voor project management in de vorm van Scrum of ondersteuning voor het managen van grootschalige teams.

Door deze gebreken presenteert GitHub zichzelf als een keuze voor het managen van startups en kleinere projecten, wat goed past bij SyncVR Medical. Voor het afstudeertraject kan niet worden afgehaakt van Git, om bijvoorbeeld Mercurial te gebruiken, omdat GitHub geen ondersteuning heeft voor andere versiebeheer systemen dan Git. Dit maakt het koppelen van andere versiebeheer systemen alleen mogelijk met technieken om het compatibel te maken met GitHub.

Wel biedt GitHub een Kanban board aan in de vorm van GitHub Boards, waar aan een repository verschillende boards toegevoegd kunnen worden. Een board kan worden opgedeeld in verschillende kolommen om zodoende taken te organiseren en de huidige staat ervan te labelen. Taken die op het board gezet worden, kunnen een taakbeschrijving hebben, eventueel een checklist en een label gekoppeld krijgen. Ook worden taken gekoppeld aan pull requests, zodat deze taken als 'Done' worden gelabeld, wanneer een pull request met taken wordt gemerged naar de *main* of *development* branch. Op dit board wordt de Kanban ontwikkelproces toegepast die in de vorige alinea (5.1 Kanban) is beschreven.

## 5.2 Modeleren met UML

Unified Modeling Language (verder te noemen als UML) is een van de meest gebruikte methodes om de visies van systeemontwikkelaars begrijpelijk en gestandaardiseerd vast te leggen (Cruysberghs, 2002). Door hier een standaard van te maken, kan UML gebruikt worden om context te geven aan verschillende programmeertalen, zonder dat het van bepaalde technieken afhangt, die exclusief zijn voor een bepaalde taal. Dit maakt het een handige methode binnen het afstudeertraject om de interactie tussen de verschillende onderdelen van ABLEL en UBLEP weer te geven, omdat ze beide in verschillende talen geschreven zijn (Java en C#). Ook wordt het gebruikt om nieuwe features of implementaties van design patterns, op een overzichtelijke manier weer te geven.

Om veranderingen in code en het design achter code bij te houden en visueel weer te geven, wordt gebruikgemaakt van PlantUML. PlantUML is een syntax waarmee met leesbare tekst en commands verschillende soorten UML diagrammen gegenereerd kunnen worden. Omdat het leesbare tekst en commands zijn, kunnen de geschreven diagrammen worden bijgehouden met versiebeheer systemen zoals Git. Git heeft namelijk moeite met het bijhouden van versies van binair- of ander niet-mens-leesbare tekst bestanden, waardoor save-bestanden die gegenereerd worden door UML modeler software zoals Microsoft Visio, niet bijgehouden kunnen worden. In tegenstelling daarmee genereert software zoals StarUML en PlantUML geen binaire bestanden, waardoor versiebeheer toegepast kan worden. Maar wat PlantUML effectiever maakt dan StarUML, een ander populair UML modeler programma, is dat het geen extra aanwijzingen nodig heeft om een UML te genereren, terwijl StarUML informatie opslaat over de posities van de verschillende onderdelen.

Ook is PlantUML cross-platform, omdat het gelimiteerd is tot een enkele Java package, waardoor het UML kan genereren op elk platform dat Java ondersteunt. Dit, en de combinatie met versiebeheer ondersteuning, maakt het een goede tool voor het bijhouden van verschillende versies van het design van een codebase, het noteren van activiteiten diagrammen en het generen van use-case scenario's.

## 6 Requirements

Dit hoofdstuk gaat dieper in op de requirements die aan het afstudeertraject zijn gesteld. Dit kunnen requirements zijn die aangeven op welke manier er wordt geprogrammeerd, hoe de uiteindelijke tussen- en eindproducten werken met andere al-ontwikkelde applicaties of requirements die er voor zorgen dat het product makkelijker te gebruiken is binnen een ziekenhuis omgeving.

### 6.1 Vergaren van Requirements

De manier van programmeren binnen SyncVR Medical is vrij los en conformeert zich niet aan al te veel regels die vooraf gesteld zijn. Dit zat in de weg tijdens het ontwikkelen, omdat vooraf afgestemde requirements het overzichtelijk maakt om te programmeren. Door later regels toe te voegen, kan dat ervoor zorgen dat oude code, keuzes en oplossingen hervormd moeten worden om aan deze nieuwe requirements te voldoen. Door na te vragen bij de bedrijfsmentor en andere ontwikkelaars binnen SyncVR Medical zijn er toch een aantal requirements opgesteld aan het begin van het afstudeertraject. Hier zijn naderhand meer requirements aan toegevoegd door de manier van ontwikkelen binnen SyncVR Medical, namelijk continue fasering in ontwikkelen waar Agile een grote rol in speelt. Na elke gegeven demo over het Proof of Concept, kan feedback gegeven worden over de applicatie zelf, of over de toepassing van BLE binnen het Proof of Concept. Deze kunnen genoteerd worden als requirement zodra het als regel kan worden toegepast, of als use-case diagram over de manier waarop een gebruiker interacteert met het Proof of Concept, of hoe een ontwikkelaar de UBLEP of ABLEL kan gebruiken.

De beschreven requirements kunnen ook van toepassing zijn op het Unity project. Als voorbeeld: het gebruiken van een aantal prefabs, wegens restricties die worden gelegd door packages. Zo wordt er standaard de *Unity XR Rig* geplaatst in elke scene waarin VR een rol speelt, zodat gebruikers zich niet hoeven aan te passen aan verschillende soorten instellingen in VR, maar een standaard model voor handen hebben, standaard instellingen voor de hoogte van de bril in vergelijking met de grond, etc.

De requirements zijn onderverdeeld in twee verschillende categorieën: BLE package (*Tabel 4*) requirements en requirements die invloed hebben op het ontwikkelen van een SyncVR Fit applicatie (*Tabel 5*). Hiervan hebben de requirements van de BLE package een extra categorie gelabeld om de invloed weer te geven op welk tussen- of eindproduct dit invloed heeft (*Kolom 3 van de requirements tabellen*).

*Tabel 4: Requirements ter het ontwikkelen van de BLE package*

Afkorting	Beschrijving	Categorie
R1	Hou de data dicht tot realtime. Data mag er niet <b>te lang</b> (0.5s) over doen om vanuit de wearable invloed te hebben op de applicatie. Creëer geen systeem waarin data de hele tijd over een weer wordt gestuurd, dat voor meer vertraging zorgt om de data naar het doel te brengen.	UBLEP / ABLEL
R2	SyncVR Medical wilt dat de oplossing eenvoudig te gebruiken is binnen ziekenhuizen. De hardware dat gebruikt wordt voor de oplossing moet praktisch zijn en moeiteloos te implementeren zijn binnen zorginstanties.	ABLEL
R3	SyncVR Medical wilt dat er een POC komt dat de oplossing implementeert om wearables te verbinden met een VR headset.	HEDR
R4	Ontwikkel een Bluetooth Low Energy adapter dat kan samenwerken met de al bestaande SyncVR Bluetooth Spectating Agent. Zorg ervoor dat beide Bluetooth instanties kunnen werken op dezelfde VR bril.	UBLEP / ABLEL
R5	SyncVR Medical wilt dat de oplossing generiek is. Zorg ervoor dat de oplossing te gebruiken is binnen elk project binnen SyncVR Medical zonder dat er een aparte oplossing geschreven hoeft te worden.	UBLEP / ABLEL

*Tabel 5: Technische requirements ter het ontwikkelen van een SyncVR Fit applicatie (POC)*

Afkorting	Beschrijving	Categorie
UR1	Gebruik altijd de 'XR Rig' prefab als er binnen de scene gebruikgemaakt wordt van Virtual Reality. Deze prefab bevat al de tools die nodig zijn voor een patiënt om zich te wenden binnen de scene in Virtual Reality.	HEDR
UR2	Om verschillende moeilijkheidsgraden te maken wordt gebruikgemaakt van een <i>ScriptableObject</i> dat overerft van de 'Difficulty' class.	HEDR
UR3	Voordat tekst op UI wordt getoond, wordt altijd gebruikgemaakt van een gelokaliseerde string. Deze heeft een unieke en relevante identificeer code die gekoppeld staat aan de ontwikkelde oefening.	HEDR
UR4	Om verschillende moeilijkheidsgraden te laten zien binnen de oefening, wordt dit gedaan met de 'Exercise Control Canvas – XR'. Deze laat benodigde knoppen zien en blijft dezelfde stijl houden door alle scenes heen, wat het makkelijker maakt voor patiënten om de menu en zijn functie te herkennen.	HEDR
UR5	Er wordt gebruikgemaakt van een aantal packages, namelijk... Die zijn nodig om de verschillende gebruikte headsets te ondersteunen. Deze packages moeten aanwezig zijn en mogen dus niet verwijderd worden. Ook mogen nieuw toegevoegde packages geen conflicten hebben met al geïmporteerde packages.	HEDR / UBLEP



## 7 Realiseren

Tijdens het afstudeertraject is gewerkt aan enkele eindproducten die binnen dit afstudeerdossier zijn beschreven. Deze eindproducten zijn vooraf afgesproken bij het voorleggen van het Plan van Aanpak (*Bijlage A: Plan van Aanpak (p. 58)*) en hebben ook hun eigen verwijzingen (*3.5 Eindproducten (p. 11)*). In de tekst wordt uitleg over concrete onderwerpen zo veel mogelijk vermeden en er wordt alleen nieuwe leerstof, keuzes en overwegingen genoteerd als deze aan bod kwamen opgedaan tijdens het werken aan de desbetreffende onderwerpen. Uitleg over concrete onderwerpen zijn vooraf beschreven en toegelicht in het hoofdstuk Theoretisch Kader (*4 Theoretisch Kader (p. 12)*).

De onderwerpen binnen dit hoofdstuk zijn chronologisch, waarbij het eerste hoofdstuk het begin van het stagetraject beschrijft. Ondanks dat de onderwerpen in chronologische volgorde staan, betekent dit niet dat het realisatie proces is verlopen onder een waterval methode. Er wordt ook binnen de hoofdstukken terug verwezen wanneer elementen worden uitgebreid, vervangen of verbeterd in een latere periode binnen het afstudeertraject.

Problemen tijdens dit traject worden eerst geanalyseerd, daarna wordt het originele design aangepast of een nieuwe ontworpen. Met dit design worden de aanpassingen aan de producten verwerkt. Zo blijft er iteratief gewerkt worden aan de uiteindelijke eindproducten.

### 7.1 Ontwerpen van Concepten

Het begin van het afstudeertraject stond in het thema van opzetten. Binnen de eerste twee sprints was het dus ook belangrijk om een branch in de SyncVR Fit GitHub repository te hebben, een Kanban board en een set aan requirements om te starten met het bestuderen en toepassen van BLE binnen Unity en Android. SyncVR voerde aan om een aantal concepten uit te schrijven voor de uiteindelijke Proof of Concept, zodat feedback- over, en een preferentie naar, de concepten geleverd kon worden. Hiervoor zijn uiteindelijk drie concepten uitgewerkt aan de hand van het High Concept Document (Adams, 2008), waarmee binnen een paar hoofdstukken en punten duidelijk wordt gemaakt wat het doel van het concept is, welke core features het de gebruiker biedt en hoe het uitgewerkte concept staat in relatie tot de competitie. De keuze om het High Concept Document te gebruiken in tegenstelling tot andere formaten om concepten van games te presenteren -zoals een Game Design Document- is omdat het High Concept Document snel en gericht het concept van de game uitlegt, zonder te diep of te gedetailleerd over het verhaal achter de game in te gaan, of over de personages die erin voorkomen. Ook geeft het High Concept Document de mogelijkheid om snel aanpassingen te maken aan het concept, omdat de punten die het eindproduct beschrijven globaal en oppervlakkig zijn beschreven, waardoor het eenvoudig is om aanpassingen te vinden. Daarnaast behoeft het verhaal of de personages nauwelijks diepgang voor de oefeningen die ontwikkeld worden binnen SyncVR Fit. Dit tref je wel in andere games buiten SyncVR, zoals *Silent Hill*<sup>3</sup> of *Doom*<sup>4</sup>. In tegenstelling tot een game treatment, bevat het High Concept Document slechts een globaal en “hoog” zicht op wat de game gaat worden. Bij een game treatment wordt de inhoud van het High Concept Document uitgewerkt met meer details, mockups, concept art en wordt gedetailleerder wat karakters en wereld betreft (Vincent, 2013).

---

<sup>3</sup> [Silent Hill Design Document](#)

<sup>4</sup> [Doom Design Document](#)



De uitgewerkte High Concept Documents staan in *Bijlage B: High Concept Documents* (p. 70). De inhoud van deze concepten is niet belangrijk; de drie concepten focussen op het laten bewegen van de speler om zo Virtual Reality Rehabilitation toe te passen en het integreren van een wearable. Deze wearables hebben een rol binnen alle games en kan invloed hebben op hoe de game zich verder ontwikkelt, bijvoorbeeld het stoppen- of het langzamer maken van de oefening wanneer de hartslag boven een bepaalde drempel komt. Bij *Heart Drumming* en *Shape Your Body* wordt de hartslag functie van een wearable gebruikt om de snelheid van de oefening te bepalen. *March Your Step* gebruikt de stappenteller binnen een wearable om de gebruiker naar voren te laten bewegen binnen de game, zonder dat de gebruiker fysiek naar voren hoeft te bewegen, alleen op dezelfde plek stap bewegingen hoeft te maken.

Persoonlijk was er onzekerheid over het gebruik maken van *March Your Step* als concept, een gebrek aan de juiste (voor)kennis op het gebied van stappentellers en hoe deze intern functioneren. Als deze namelijk een bepaalde techniek gebruiker om stappen te tellen waardoor het niet werkt als de gebruiker niet fysiek naar voren beweegt, dan kan het uitgewerkte concept niet werken. Zodra het SyncVR Fit team een voorkeur had voor *March Your Step*, werd hier deskresearch toegepast om te achterhalen of het integreren werkt met de Xiaomi Band 3's (verder te noemen als MiBand 3) stappenteller, of dat hier een andere techniek voor moet worden toegepast. Nadat het concept was opgesteld, werd deze voorgelegd aan het SyncVR Fit team om feedback te krijgen over de effectiviteit van de games als Virtual Reality Rehabilitation applicatie en over de manier waarop de wearables en hun functionaliteiten worden gebruikt binnen de concepten.

Uiteindelijk is er gekozen voor Heart Drumming, omdat het uitgewerkte concept goed werkt als Proof of Concept. Door de hartslag een rol te laten spelen binnen de oefening waaraan een wearable wordt gekoppeld, functioneert het als een goede POC voor het verbinden van een wearable aan een VR headset. Ook biedt het zichzelf aan als een Virtual Reality Rehabilitation program, vanwege de vele armbewegingen die nodig zijn om de drumsets binnen de game te raken. Om eventuele veranderingen van concept in de toekomst te kunnen dekken, wordt gebruikgemaakt van de MiBand 3 als wearable peripheral. Deze wearable ondersteunt verscheidene doeleindes, zoals het registreren van een hartslag, de beschikking hebben over een stappenteller en het communiceren via BLE.

## 7.2 Opzetten van BLE

Eerst werd gekeken naar de mogelijkheden om Android's BLE te gebruiken binnen Unity. De mogelijkheden zijn gelimiteerd tot het gebruik van Unity Asset Store packages, uitbreiding van bestaande BLE frameworks om de MiBand 3 te ondersteunen of het ontwikkelen van een BLE library onder SyncVR Medical. Na een bespreking met stagebegeleider R. Brouwer is er gekozen om een BLE library te ontwikkelen onder SyncVR Medical. Dit geeft de mogelijkheid om het moeiteloos uit te breiden, biedt meer controle over de source code en is niet afhankelijk van derde partijen om de code te onderhouden. Een asset gebruiken van de Unity Asset Store is ook een valide mogelijkheid, alleen hiervoor zijn de packages gelimiteerd en weinig packages BLE ondersteunen voor de Android operating system.

Er is overwogen om te kiezen om bestaande BLE frameworks uit te breiden voor ondersteuning voor de MiBand 3, daarvoor echter zijn de bestaande BLE frameworks zeer gelimiteerd: dit omdat er weinig vraag is naar een BLE framework voor Android in Unity, maar ook omdat de frameworks die wel hiervoor zijn gemaakt, gelimiteerd zijn aan een set aan peripheral(s) (die niet de MiBand 3 ondersteunen) of waar de code al een tijd niet onderhouden is. Door een nieuwe library te schrijven voor ondersteuning van Android BLE binnen Unity, is er meer controle over de code en kan een beter persoonlijk kennisveld over BLE worden ontwikkeld.

Voordat er werd ontwikkeld aan de BLE library, werd er eerst een stappenplan gemaakt om meer duidelijkheid te genereren over SyncVR Medical en over welke activiteit er eerst gemaakt moeten worden voordat waardes uit een wearable gelezen zou kunnen worden. Dit stappenplan hielp ook op persoonlijk vlak vanwege het overzicht over de splitsing planning en het ontwikkelen van BLE. Ook geeft het inzicht wanneer een stap een bepaalde onderzoeksmethode nodig heeft (te zien in *Tabel 6: Stappenplan ter voorbereiding van het opzetten van Bluetooth Low Energy*, Kolom 2). Hierbij wordt bij elke *Academische* stap deskresearch toegepast en bij *Technisch* wordt gebruikgemaakt van de prototyping onderzoeksmethode. Door praktisch de opgedane kennis toe te passen die gerealiseerd zijn door deskresearch, wordt de geleerde stof beter opgepakt. Ook de stagebegeleider vanuit SyncVR Medical stimuleert deze methode van informatie vergaren, omdat er dan op een effectievere manier resultaten kan worden getoond. Dit resulteert in minder theoretische echter wel veel praktisch geïmplementeerde kennis, die kan worden overgedragen aan SyncVR. De actiepunten in het stappenplan zijn opgesteld aan de hand van enige voorkennis over BLE en nieuwe kennis die is opgedaan door het uitvoeren van deskresearch. Deze deskresearch omvatte het lezen van artikelen over BLE, reverse-engineering artikelen over de MiBand 3 en het bestuderen van bestaande BLE frameworks in C#, React Native en Android Java code. Door toepassing van deskresearch werd duidelijk welke techniek te gebruiken om communicatie tussen Unity en Android's BLE plaats te laten vinden. Dit is weergegeven in *Tabel 6: Stappenplan ter voorbereiding van het opzetten van Bluetooth Low Energy* en wordt sequentieel en chronologisch uitgevoerd. De tijd die hiervoor wordt genomen staat in de planning in de *Plan van Aanpak* onder het onderwerp *Planning* (p. 68) en beslaat ongeveer zes weken.

Tabel 6: Stappenplan ter voorbereiding van het opzetten van Bluetooth Low Energy

#	Stap	Type	Doel
1	Unity en Bluetooth	<i>Academisch</i>	Uitzoeken met welke tools binnen Unity gewerkt kan worden met Bluetooth, en hoe dit systeem Bluetooth Low Energy wordt ondersteund.
2	BLE peripheral vinden	<i>Technisch</i>	Met behulp van een Android BLE protocol peripherals achterhalen, zoals een Media Access Control <sup>5</sup> (verder te noemen als MAC), adres en naam van de peripheral.
3	Verbinding maken met peripheral	<i>Technisch</i>	Door de MAC adres te gebruiken die vergaard kan worden door de vorige stap, verbinding maken met een peripheral via de standaard BLE protocollen.
4	BLE characteristic vinden	<i>Academisch / Technisch</i>	Met een BLE characteristic kan gestart worden met uitlezen van data. Als een characteristic gevonden wordt, dan kan ook een descriptor gevonden worden voor het toepassen van Read, Write of Subscribe / Notification gedrag.
5	BLE characteristic waardes uitlezen	<i>Technisch</i>	Als een BLE characteristic gevonden wordt en de descriptor het type waarde kan doorgeven, dan is het mogelijk dat waardes binnen de characteristic uitgelezen kunnen worden.
6	BLE descriptor Subscribe / Notificatie gedrag laten vertonen	<i>Academisch / Technisch</i>	Door een specifieke waarde voor een type gedrag te schrijven naar een BLE descriptor die het ondersteunt, vertoont de descriptor dit gedrag. Hierbij is de bedoeling om de volgende stap voor te bereiden, door de op de descriptor aan te geven dat het veranderende data uit de characteristic moet doorsturen.
7	BLE characteristic Subscribe waardes uitlezen	<i>Technisch</i>	Wanneer de descriptor op het juiste gedrag is afgesteld, kan deze veranderende data doorgeven naar de verbonden server aan de peripheral.

Na het uitvoeren van stap 7 zou hartslag waardes vanuit een BLE peripheral naar Unity gestuurd moeten worden. Deze waardes kunnen vanaf dit punt gebruikt worden om het Proof of Concept (Heart Drumming) te ondersteunen.

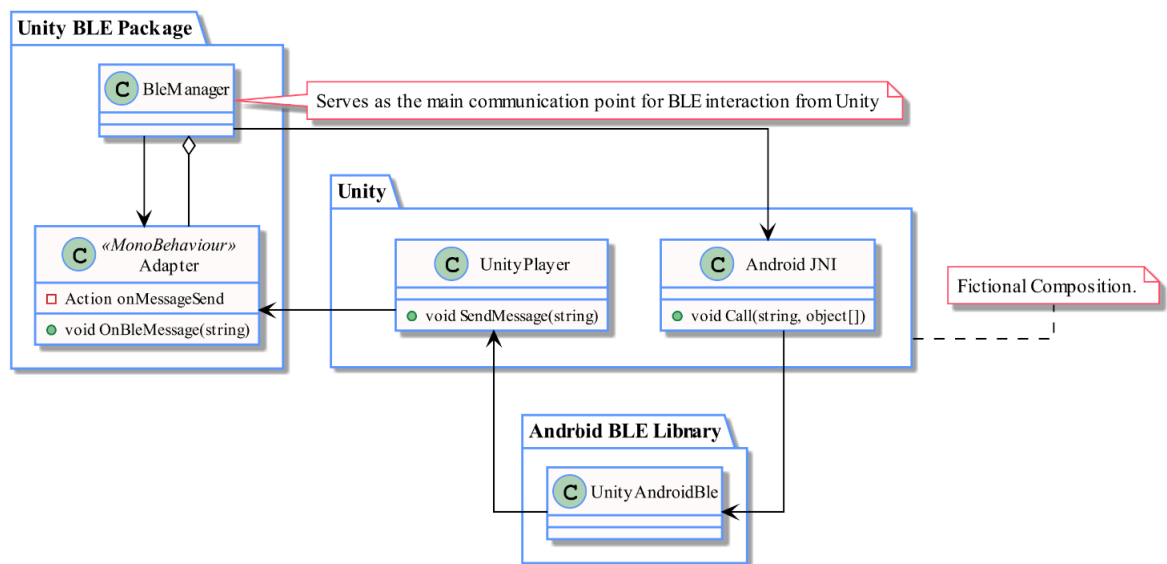
Stap één: Unity ondersteunt van oorsprong geen Bluetooth. C#, de primaire programmeertaal binnen Unity, daarentegen ondersteunt Bluetooth wel, maar alleen binnen het eigen platform van Microsoft, zoals UWP en Xamarin. De mogelijkheden buiten deze platforms zijn zeer gelimiteerd en geen daarvan ondersteunen Bluetooth op Unity gecompileerd voor Android. Hieruit kwam de conclusie dat een nieuwe oplossing gevonden moest worden. De ideeën in een artikel van J.C. Murray (ontwikkelaar binnen het bedrijf *Blackberry*) brachten nieuwe manieren om dit probleem aan te pakken. In het artikel wordt er een library in C geschreven om op een BlackBerry Bluetooth Low Energy protocollen aan te roepen en deze door te sturen naar Unity via de `UnitySendMessage` methode (Murray, 2014). Deze C library wordt mee gecompileerd met het Unity project, waardoor interne protocollen van de BlackBerry apparaat BLE aanspreken en dit communiceren naar Unity, met de `UnitySendMessage` methode.

<sup>5</sup> [MAC-Adres](#)

Zonder ervaring te hebben in C, is gezocht naar een alternatieve programmeertaal die hetzelfde principe volgt en ondersteund wordt door Unity. Unity heeft een lijst<sup>6</sup> aan ondersteunde plug-ins die werken met Unity en mee worden gecompileerd naar een Android applicatie. Binnen deze lijst bestaat er ook de mogelijkheid om een JAR (afkorting voor Java Archive) plug-in te compileren en deze binnen het Unity project te gebruiken. JAR is een methode om verschillende files te binden tot één file<sup>7</sup>. Binnen Unity wordt dit gebruikt om verscheidene Java files te binden tot één file. Deze JAR files kunnen extern van Unity gemaakt worden, hiervoor is gebruikgemaakt van Android Studio om de Android specifieke JAR library voor BLE te schrijven. Deze JAR library roept de interne Android BLE protocollen aan en geeft waarden terug aan Unity met behulp van de UnitySendMessage methode die JAR plug-ins kunnen aanroepen wanneer deze mee gecompileerd zijn in de Unity Android applicatie. Via C# scripts kan er dan interactie plaatsvinden met de JAR library vanuit Unity, door de JNI te gebruiken om de methodes binnen de JAR library aan te roepen.

De manier waarop UnitySendMessage werkt is door een naam van een GameObject binnen de huidige scene te sturen, de naam van de methode die uitgevoerd moet worden en een *string* met de informatie die de methode kan gebruiken om een actie uit te voeren. Door een Adapter te bouwen binnen Unity waarvan de JAR library bekend is met de naam en de methode om uit te voeren, kan informatie vanuit de JAR library worden doorgestuurd naar Unity.

In *Figuur 7* is het eerste ontwerp te zien dat met behulp van het voorgenoemde artikel is opgesteld. Dit ontwerp laat de wijze van het versturen van data zien, door de UnitySendMessage methode te visualiseren. Binnen dit ontwerp dient de *Unity* package als fictieel onderdeel. Omdat de werking binnen de Android JNI en de UnityPlayer niet bekend is noch hoe deze eventueel met andere classes communiceert, zijn er hiervoor dummy classes ontworpen. In het ontwerp dient *Adapter* als de class die alle data vanuit de *Android BLE Library* binnen krijgt en deze door kan sturen naar andere classes, zoals de *BleManager*. De classes die de BLE informatie willen ontvangen, kunnen dan een methode toevoegen aan de *onMessageSend*, hetgeen wordt aangeroepen wanneer nieuwe informatie vanuit de *Android BLE Library* wordt doorgestuurd.



Figuur 7: Eerste ontwerp voor de BLE implementatie binnen Unity

<sup>6</sup> [Unity – Plugins For Android](#)

<sup>7</sup> [JAR](#)

Door de *Android JNI* te gebruiken, zou er ook wederzijdse communicatie moeten ontstaan. Door de *Call* functie te gebruiken, kunnen methodes in de Android BLE Library aangeroepen worden, waarbij eventueel parameters meegegeven kunnen worden.

Alternatieven voor het versturen van BLE informatie naar Unity zijn moeilijk te vinden, omdat dit een niche is waar weinig interesse vanuit de Unity gemeenschap naar is. Een aantal alternatieven die zelf zijn bedacht zijn: het opstellen van een eigen wearable met behulp van Arduino, Teensy of andere microchip systemen die gekoppeld zijn aan een ander protocol, zoals met behulp van het internet communiceren door het gebruik van chip ESP8266. Hierbij wordt dan geen Bluetooth of BLE meer toegepast, maar wordt informatie tussen de VR headset en de wearable uitgewisseld via het netwerk. Deze module zou dan data uit kunnen lezen uit modules die ermee verbonden zijn, zoals een hartslagmonitor, en het via internet naar de VR-headset sturen. Zo is er meer controle over wát gestuurd wordt en hóe deze data wordt ontvangen, omdat een gepersonaliseerde manier van communiceren geïmplementeerd kan worden. Doordat deze requirement *R1* (*'Hou data dicht tot realtime...'*) en *R2* (*'De oplossing moet praktisch zijn...'*) overtreedt, is dit bepaald geen implementeerbaar alternatief. Het is ook mogelijk dat er problemen ontstaan met het versturen van informatie over een ziekenhuis-netwerk, of dat dit niet is toegestaan is met onbeveiligde informatie, of dat hierdoor juist allerlei soorten verbindingen worden geblokkeerd.

Binnen .NET bestaan er libraries en DLL's die het mogelijk maken om BLE te gebruiken. Deze libraries en DLL's zijn niet aanwezig op Android of Unity, omdat deze gelimiteerd zijn tot Microsofts eigen frameworks. Dit zorgt ervoor dat deze tools niet gebruikt kan worden in een mobiele VR-headset omgeving binnen Android. Ook andere frameworks die BLE voor .NET ondersteunen zijn gericht op deze frameworks, niet voor .NET in combinatie met Unity.

Er is nog kort overwogen om te kiezen voor een bepaalde programmeertaal, ondanks dat er al gekozen was om een Library te schrijven waarmee met behulp van de Android JNI interactie kon plaatsvinden. Die Library kon worden geschreven in Java, Kotlin (voor een JAR plugin) of C++ (voor native Android plugins). Als nieuwe ervaring en ter voorbereiding voor de toekomst zou C++ of Kotlin een goede keuze zijn en zou Java een goede keuze zijn voor het aansterken van Java programmeer kennis. Aangezien andere Android plugins binnen SyncVR Medical in Java worden geprogrammeerd, wordt er (toch) ook gebruikgemaakt voor ABLEL de Java programmeertaal.

### 7.2.1 Centraliseren van Communicatie

Om communicatie tussen UBLEP en ABLEL gecentraliseerd te maken, is gekozen om het Singleton pattern te implementeren om deze communicatie te regelen binnen een enkele instantie. Alhoewel het implementeren van een Singleton pattern kan leiden naar een anti-pattern (Bhardwaj, 2017), zijn de meeste punten die gebruikt worden om een Singleton als anti-pattern te beschrijven niet van toepassing op de manier hoe het is geïmplementeerd binnen het eindproduct UBLEP. De voornaamste reden waarom het Singleton pattern een anti-pattern wordt genoemd, is omdat het onderdelen binnen de SOLID principes verbreekt. Binnen SOLID bestaan twee principes welke Singleton overtreedt als deze fout wordt geïmplementeerd, namelijk de *Single Responsibility Principle* en de *Open/Closed Principle*.

Ondanks de nadelen aan het verkeerd implementeren van het Singleton design pattern, is het twee keer geïmplementeerd binnen het eindproduct. Omdat BLE niet optimaal werkt wanneer meerdere schrijf- en lees commands worden gestuurd (Ong, 2019) kan daardoor informatie verloren gaan, is het belangrijk om een wachtrij te realiseren voor de verschillende commands die naar het BLE wearable wordt gestuurd. Om overzicht te houden op deze wachtrij (van commands), moet er een implementatie zijn die deze commands uitvoert wanneer ze beschikbaar zijn en in een wachtrij houden zolang de vorige command nog niet volledig is afgehandeld. Hiervoor is het Singleton design pattern gebruikt om slechts één instantie te hebben van deze class; deze instantie houdt overzicht over de commands die uitgevoerd en afgehandeld moeten worden. Zo wordt maar één instantie aangemaakt die commands opstelt, deze in de wachtrij zet en dit consequent achter elkaar uitvoert.

In het design en binnen UBLEP wordt deze Singleton implementatie met zijn doel gebruikt, maar omdat UBLEP niet wordt gecompileerd naar een library zoals een DLL, betekent het dat elke SyncVR Medical-developer die UBLEP gebruikt, de code kan aanpassen. Dit kan er dus voor zorgen dat andere gebruikers de Singleton implementatie gaan overerven, wat het doel van de Singleton weer verbreekt. Dit resulteert erin dat het toch een anti-pattern wordt. Echter, door de Singleton class een *sealed* class te maken, wordt ervoor gezorgd dat geen andere class de Singleton kan overerven. In combinatie met de Command Stack dat het Command pattern toepast, houdt men zich ook aan de *Single Responsibility Principle*, wat in detail wordt uitgelegd in 7.3 *Command Stack* (p. 33). Op deze manier wordt het Singleton pattern geïmplanteerd op een manier dat zich houdt aan de SOLID, waardoor het overkomt als een anti-pattern dat geïmplementeerd is.

### 7.2.2 Informatie Versturen

Met de manier van communiceren tussen Unity en de Android library opgezet, kan de JAR library worden ontwikkeld. Deze library zal alle BLE protocollen uitvoeren, de informatie uit die protocollen vergaren en deze doorsturen naar de adapter binnen Unity. De Adapter heeft dan verschillende C# Actions die worden uitgevoerd aan de hand van de ontvangen command. Een Action in C# kan referenties houden naar methodes; dit houdt in dat wanneer de command *DiscoveredDevice* door de Adapter wordt ontvangen, alle methodes die de Action refereert worden uitgevoerd. Hierbij worden ook de desbetreffende parameters doorgestuurd met diezelfde Action, zodat ook deze gebruikt kunnen worden. Bijvoorbeeld wanneer een BLE peripheral wordt gevonden, dan wordt deze met zijn MAC adres en naam doorgestuurd naar Unity door middel van het *DiscoveredDevice* command. Een voorbeeld hiervan is te zien in *Figuur 8*, waarin de MiBand 3 wordt afgebeeld als *DiscoveredDevice* command. Met behulp van zo'n command kan er dus een actie uitgevoerd worden binnen Unity.

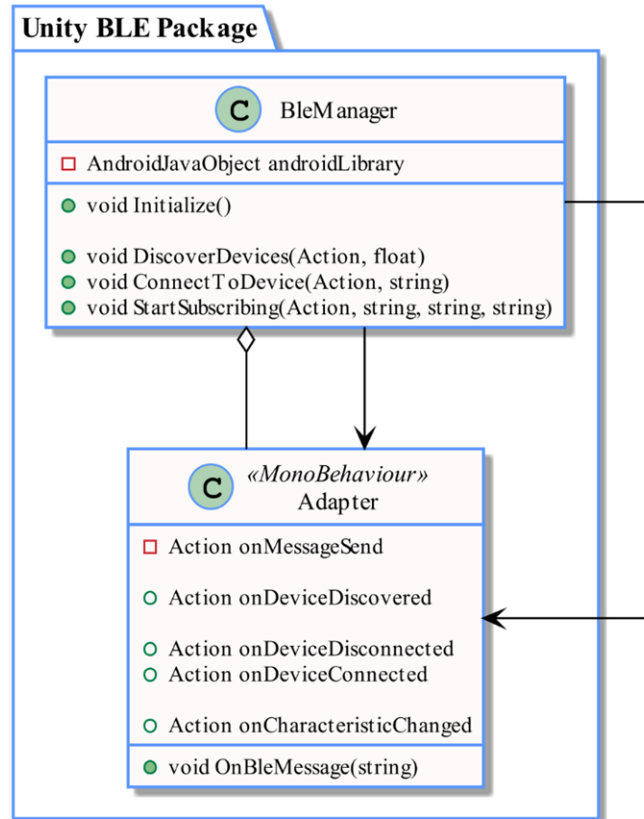
DiscoveredDevice~E0:9A:31:5A:2D:CF~Xiaomi Band 3		
⌞ Command	⌞ Device MAC Adres	⌞ Device Naam

*Figuur 8: DiscoveredDevice informatie dat vanuit ABLEL wordt verstuurd naar UBLEP*

Alle parameters die binnen een doorgestuurd command vallen, zijn gescheiden door middel van een enkel karakter; het '~' teken dient hiervoor. Door de C# *string.Split* functie te gebruiken, kan de command gescheiden worden van de parameters, en worden de parameters verpakt in een aparte array. Deze parameters worden daarna gebruikt om Actions, zoals *OnConnected*, meer informatie te geven zoals met welke peripheral een verbinding gemaakt is.



Figuur 9 toont een onderdeel van de huidige staat van het UML design, van de manier van communiceren tussen UBLEP en ABLEL. Het volledige design staat in de Bijlagen, onder *13.1 Unity BLE Package en Android BLE Library UML Design (Fase 2)* (p. 56), hiervan is het onderdeel in het figuur linksboven te vinden binnen het volledige design. Dit onderdeel is geïmplementeerd binnen UBLEP en ABLEL. Dit zorgt ervoor dat de informatie zoals hiervoor genoemd, wordt doorgestuurd en ontvangen in de *BleManager*.

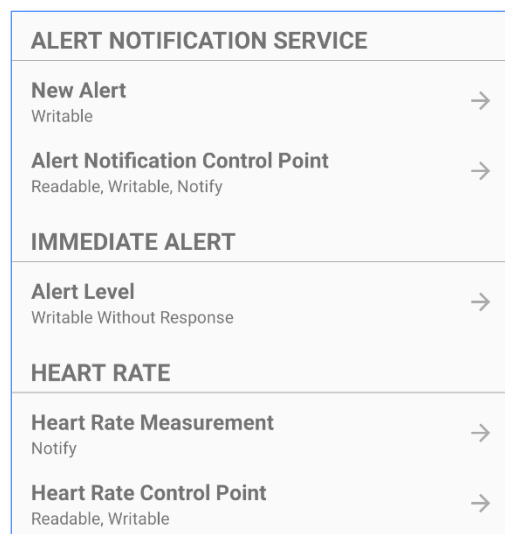


Figuur 9: Onderdeel van het UML design voor communicatie tussen UBLEP en ABLEL

In de *Adapter* zijn *OnBleMessage* methode staat een *switch* logica dat aan de hand van het eerste onderdeel uit de ontvangen informatie (de command, zie eventueel *Figuur 8*) de juist Action aanroept. Dus wanneer een *string* binnenkomt zoals in *Figuur 8*, dan wordt de action *onDeviceDiscovered* uitgevoerd. Deze Actions zijn vooraf gezet door de *BleManager* zodra een desbetreffende methode is aangeroepen. Achteraf bleek dit geen goede aanpak, aangezien dit ervoor zorgt dat nieuwe commands en hun logica in de *switch* case verwerkt moesten worden. Ook is nadere kennis nodig om te kunnen bepalen welke commands binnen kunnen komen vanuit de ABLEL, om zodoende een command toe te voegen al dan niet te verwijderen. Dit kan voor verwarring zorgen en kan makkelijk opgelost worden door middel van een design-pattern, namelijk het Command Pattern. Dit pattern is in een later stadium binnen het afstudeertraject toegepast, en wordt verder toegelicht in hoofdstuk 7.3 *Command Stack* (p. 33).

### 7.2.3 Authenticeren via BLE

Tijdens het ontwikkelen van UBLEP en ABLEL werd gebruikgemaakt van een test applicatie wat ontwikkelt is voor een Android applicatie. Dit is een relatief eenvoudige applicatie om de protocollen en methodes van ABLEL te testen, zonder VR toe te passen. Hiervoor werd een smartphone met Android gebruikt dat BLE ondersteunt. Door middel van deze applicatie kon er een verbinding worden gelegd met de MiBand 3. Om te zien wat voor services, characteristics en descriptors de MiBand 3 bevat, werd met behulp van de applicatie 'LightBlue' hiervan een overzicht gegenereerd. Aan de hand van dit overzicht -zie *Figuur 10* - bleek dat de MiBand 3 zich houdt aan drie gestandaardiseerde BLE profielen: de *Immediate Alert service*, *Alert Notification*- en de *Heart Rate Profile*. Aan de hand van deze informatie kan er via de applicatie hartslag waardes uitgelezen worden via de standaard Heart Rate characteristic. Deze ligt op de service 180D en characteristic 2A37 (Silicon Labs, 2020) en de waardes hiervan kunnen uitgelezen worden door op de characteristic te Subscriben (zie *Figuur 10*).



ALERT NOTIFICATION SERVICE	
New Alert Writable	→
Alert Notification Control Point Readable, Writable, Notify	→
IMMEDIATE ALERT	
Alert Level Writable Without Response	→
HEART RATE	
Heart Rate Measurement Notify	→
Heart Rate Control Point Readable, Writable	→

Figuur 10: Schermopname van de LightBlue applicatie over de MiBand 3

Via de smartphone werkte deze applicatie, maar wanneer dezelfde code en libraries van UBLEP en ABLEL werden gebruikt binnen een VR applicatie op de Oculus Quest, kon er wel met de MiBand 3 verbonden worden, maar was het niet mogelijk om hartslag waardes uit te lezen. Ook werd de verbinding automatisch verbroken na ongeveer dertig seconden. Na het lezen van artikelen van '*I hacked MiBand 3*'<sup>8</sup> en '*MiBand 3 and React Native*'<sup>9</sup> die over het verbinden van verschillende platforms met de MiBand 3 gaan, bleek dat een authenticatie proces wordt gebruikt om een apparaat te verbinden met de MiBand 3. Dit authenticatie proces zorgt ervoor dat een apparaat de waardes uit de MiBand 3 z'n characteristics mag lezen en langer verbonden mag zijn. Dit authenticatie proces wordt ook wel 'pairing' genoemd en wordt binnen Bluetooth gebruikt als term om twee apparaten met elkaar te verbinden en eventueel authenticatie stappen hierbij uit te voeren.

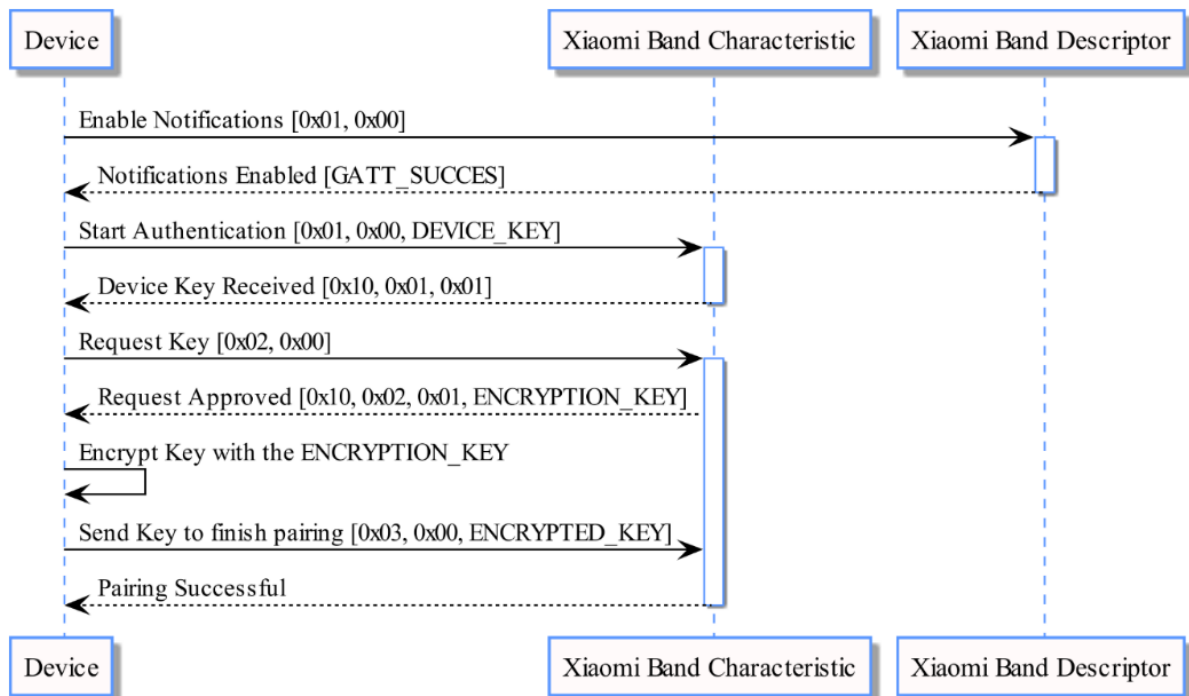
<sup>8</sup> [I Hacked MiBand 3](#) – Ohja Yogesh

<sup>9</sup> [MiBand 3 and React Native](#) – Alex K.



De reden dat de applicatie op de smartphone wel verbinding kon maken met de MiBand 3 en hartslag waarden kon uitlezen, is omdat deze was ge-authentiseerd via de Xiaomi Mi Fit applicatie. Tijdens dit proces is de smartphone ge-authentiseerd met de MiBand 3 en herkent de MiBand 3 het apparaat dat verbinding probeert te maken, waarbij het apparaat een tijdelijke connectie geeft om te authenticeren terwijl deze connectie niet de bevoegdheid geeft om de waarden van characteristics uit te lezen. Het authenticeren van een apparaat op een MiBand 3 heeft een proces dat zelf geprogrammeerd moet worden, omdat dit niet standaard wordt aangeboden binnen Android. Het proces om te authenticeren staat visueel beschreven in *Figuur 11*.

Bij dit diagram zijn de 'Xiaomi Band characteristic' en 'Xiaomi Band descriptor' een authenticatie characteristic (UUID: 0009) en descriptor (UUID: 2902). De aangegeven waarden binnen de [ ] zijn byte waarden in een array. Hierbij kan de array nog een voor gedefinieerde naam hebben, zoals [GATT\_SUCCES] dat zich vertaalt naar de array [0x00], of een array met bytes dat niet van tevoren voorspeld kan worden, zoals [ENCRYPTION\_KEY].



Figuur 11: MiBand 3 authenticatie proces.

Door te beginnen met de descriptor van de Authenticatie characteristic op Subscribe te zetten, worden updates binnen de descriptor doorgegeven aan het apparaat dat verbinden wilt. Hierdoor kan bijgehouden worden in welke stap binnen het authenticatie proces het apparaat zich bevindt, doordat de characteristic een nieuwe waarde doorgeeft om aan te geven dat het authenticatie proces een stap verder is gegaan, of dat iets is fout gegaan. Na aangegeven te hebben dat de descriptor op Subscribe moet staan, geeft de MiBand 3 door dat het gelukt is om de characteristic op Subscribe te zetten, waardoor er gestart kan worden met stap twee. Hierbij wordt een *DEVICE\_KEY* naar de characteristic geschreven. Deze *DEVICE\_KEY* is een 16 byte array met willekeurige bytes welke uniek zijn voor het apparaat waarmee je naar de MiBand 3 wilt verbinden. In combinatie met de byte *0x01* en *0x00* wordt deze *ENCRYPTED\_KEY* naar de characteristic verstuurd, waarna de characteristic een Subscribe notificatie geeft om te laten weten dat de sleutel succesvol is ontvangen. Vanaf dit punt is het apparaat dat met de MiBand 3 wilt verbinden ge-authentiseerd, en kunnen de waardes uit voorgaand-versleutelde characteristics uitgelezen worden. De volgende keer dat het apparaat verbinding wilt maken met de MiBand 3, wordt dit proces herhaald.

Deze stappen zijn nodig om de MiBand 3 te authenticeren, maar zijn zeer specifiek en tijdens het prototypen van dit authenticatie proces in de ABLEL wordt alles ge-hardcode omdat de UBLEP geen mogelijkheid biedt om dit proces te versimpelen in een C# class. Dit brengt nadelen met zich mee, omdat de package dan niet uitgebreid kan worden naar eventueel andere wearables die hun eigen authenticatie proces hebben. Dit gaat tegen requirement *R6* (*'Maak de adapter zo generiek mogelijk...'*) in. Dit, en voorgenoemde redenen, gaf de doorslag om een betere oplossing te vinden om de Unity kant van de Adapter te regelen. Het proces hiervan wordt beschreven in de volgende paragraaf.

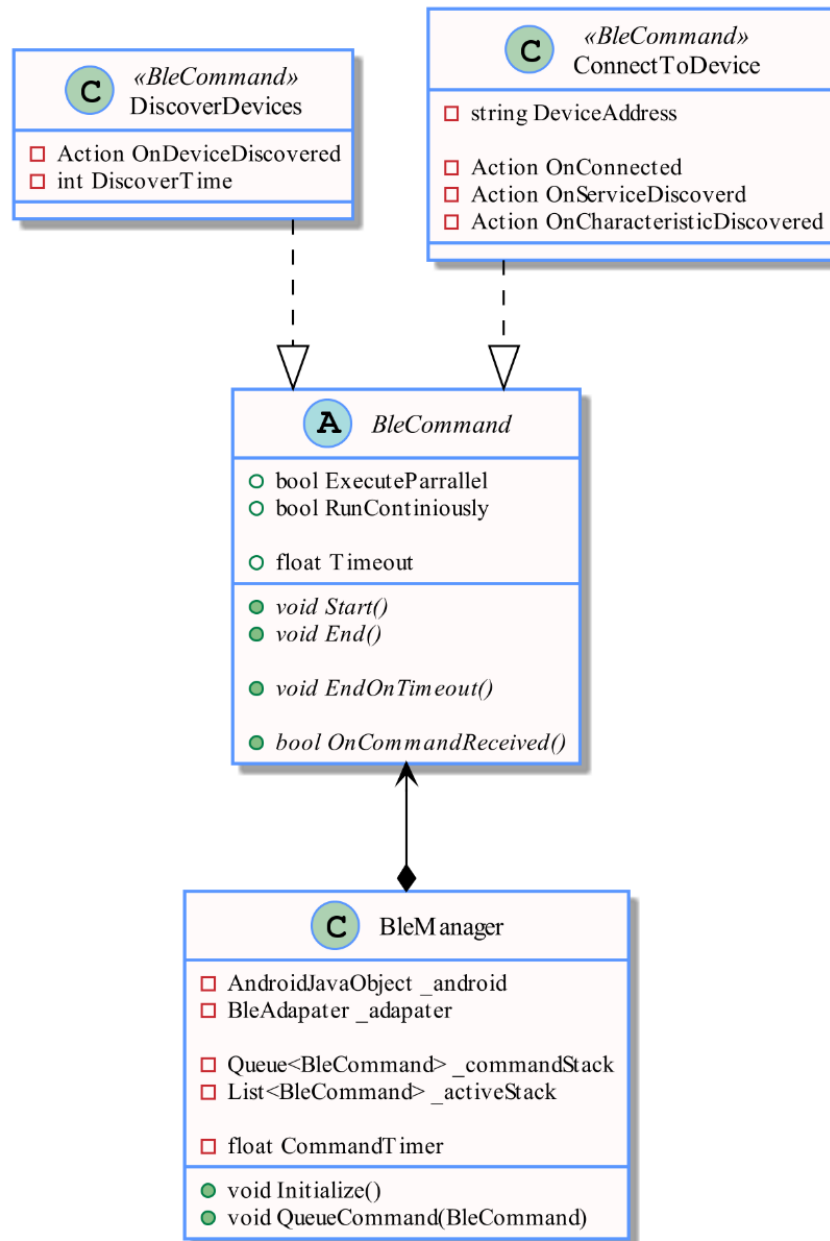
## 7.3 Command Stack

Om ervoor te zorgen dat UBLEP dynamisch blijft om eventuele nieuwe authenticatie protocollen voor andere peripherals -niet de MiBand 3 zijnde-, of andere features toe te voegen, wordt UBLEP omgezet naar een library die wordt ondersteund door het Command Pattern. Deze verandering van werking in de backend van UBLEP wordt de *Command Stack* genoemd, vanwege de manier waarop de Commands worden vergaard en sequentieel als een stapel worden uitgevoerd.

Het Command pattern wordt vooral gebruikt als een verbetering van de manier waarop UBLEP BLE instructies doorgeeft aan ABLEL. Omdat BLE in het algemeen niet goed werkt als veel instructies tegelijkertijd of snel achter elkaar worden gestuurd, wordt het command pattern gebruikt om verschillende instructies in een wachtrij te zetten en deze sequentieel achter elkaar uit te voeren. Het command pattern maakt dit makkelijk omdat deze de BLE instructies opdeelt in aparte classes met hun eigen functionaliteiten, maar blijven niet generiek genoeg zodat de centrale BLE Singleton manager deze kan aanvangen en afhandelen.

Het idee van de Command Stack is dat er één abstracte basis class is die de benodigde velden en methoden bevat om commands en informatie vanuit ABLEL op te vangen en hier weer een actie over uit te voeren. Deze class wordt binnen een *BleManager* class sequentieel uitgevoerd, of parallel als de class een actie uitvoert dat data op irreguliere frequenties ontvangt (zoals het Subscriben aan een characteristic, waarnaar alleen nieuwe data wordt gestuurd wanneer deze beschikbaar is). Door een class aan te maken voor verschillende interacties met BLE, zoals het verbinden van een peripheral, het schrijven naar een characteristic of het Subscriben naar een characteristic -dat de abstracte basis class overerft- kunnen deze aangemaakt worden en toegevoegd worden aan de *BleManager* class, waarna de *BleManager* class de parallel uitvoert of in een *Queue* zet om later uit te voeren.

Zoals in *Figuur 12* wordt weergegeven, hebben alle *BleCommand* classes een *Timeout* property. De *BleManager* houdt intern een timer bij en kan zo commands die er te lang over doen om uitgevoerd te worden afsluiten zodra dit nodig is. Als een *BleCommand* de *RunContinuously* property op *true* heeft, dan zal deze niet worden afgesloten zodra de interne timer hoger is dan de *Timeout*, en kan de command blijven draaien totdat deze extern wordt afgesloten. Een voorbeeld van een command die continue moet blijven draaien is een *SubscribeToCharacteristic* command. Deze moet blijven luisteren naar nieuwe waarden die binnenkomen en kan dus alleen afgesloten worden als er geen behoefte meer is om deze waarden op te vangen. Hiermee kan de *End()* methode worden aangeroepen om de command te stoppen.



Figuur 12: Command Stack weergegeven in een UML Diagram

De keuze om de Command Stack zo op te stellen wordt vooral gebaseerd op uitbreidbaarheid. Door de *BleCommand* op te stellen als een generieke abstracte class, kan er snel nieuwe commands worden toegevoegd die direct compatibel zijn met de *BleManager*. Het hoeft alleen zijn eigen gedrag te beschrijven en aan te geven of deze parallel- en/of continue blijft werken. Het UML diagram laat maar een beperkt aantal commands zien (*DiscoverDevices* en *ConnectToDevice*). Dit UML diagram is opgesteld om alvast een start op te stellen over hoe de Command Stack te werk zal gaan en ter voorbereiding voor abstractie.

Door de Command Stack toe te passen, wordt de verantwoordelijkheid voor verschillende acties over BLE opgesplitst over verschillende classes. Zo wordt de code overzichtelijker en wordt de *Single Responsibility Principle* uit de *Solid* toegepast op de *BleManager* en verbreekt het hierdoor het antipatern van de *Singleton*.

```
public void OnDeviceConnected(string deviceId)
{
    var command = new SubscribeToCharacteristicBleCommand (
        deviceId, "1802", "2A06",
        (data) => { _heartRate = (int)data; },
        typeof(int)
    );
    BleManager.QueueCommand(command);
}
```

Figuur 13: Code om te verbinden met Hartslag characteristic wanneer de wearable verbonden is

In *Figuur 13* wordt de kracht van de Command Stack weergegeven. Omdat de *ConnectToDevice* BLE command een trigger heeft wanneer de applicatie succesvol is verbonden met een wearable, kan deze gebruikt worden om direct een nieuwe command in de wachtrij van de *BleManager* te zetten. In het figuur is dit een *SubscribeTocharacteristic* command die parameters meegeeft om te verbinden met een hartslag characteristic. Wanneer data binnenkomt dat aansluit aan de characteristic waaraan ge-subscribed is, zal deze het *\_heartRate* veld aanpassen naar de waarde dat is binnen gekomen.

### 7.3.1 Informatie in JSON

Om de Command Stack beter uit te breiden en het verwisselen van informatie vanuit ABLEL naar UBLEP eenvoudiger te maken, wordt het uitwisselen van informatie in JSON formaat gedaan in plaats van het splitten van *strings*. De JSON informatie wordt vanuit de ABLEL gestuurd en wordt binnen UBLEP opgevangen en omgezet naar een class dat de informatie van de JSON bevat. Deze JSON data bevat genoeg data om UBLEP te voorzien van de benodigde data om zijn actieve commands van informatie te voorzien.

De keuze om te veranderen naar JSON is omdat dit makkelijk te sturen valt als een *string*, wat de primaire bron van informatie versturen is tussen UBLEP en ABLEL. Ook heeft Unity ingebouwde functies om JSON strings om te zetten naar een class dat voor Unity serializeerbaar is, wat er voor zorgt dat er geen externe JSON parsers nodig zijn om deze oplossing te laten werken. Een alternatief voor JSON zou zijn om XML te gebruiken om data te versturen van ABLEL naar UBLEP, maar XML heeft veel meer features die niet worden gebruikt om te communiceren vanaf ABLEL naar UBLEP. Ook is Unity's performance voor het serialiseren en deserialiseren van JSON objecten ontzettend snel (Dunstan, 2015) vergeleken met andere JSON libraries. Dit maakt JSON een goede datastructuur om de *string* split functionaliteit, wat voorheen werd gebruikt om data te versturen te vervangen.

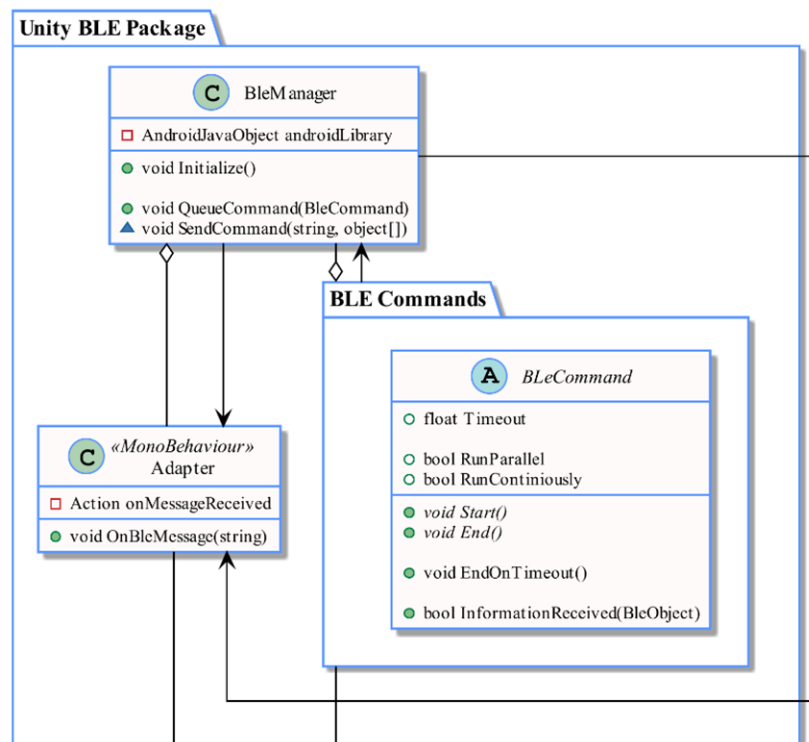
```
{
  "command": "CharacteristicValueChanged",
  "device": "E0:9A:31:5A:2D:CF",
  "name": "Xiaomi Band 3",

  "service": "0000fee1-0000-1000-8000-00805F9B34FB",
  "characteristic": "00000009-0000-3512-2118-0009af100700",

  "base64_message": "EAE=",
  "has_error": false,
  "error_message": ""
}
```

Figuur 14: JSON data dat vanuit ABLEL naar UBLEP wordt gestuurd

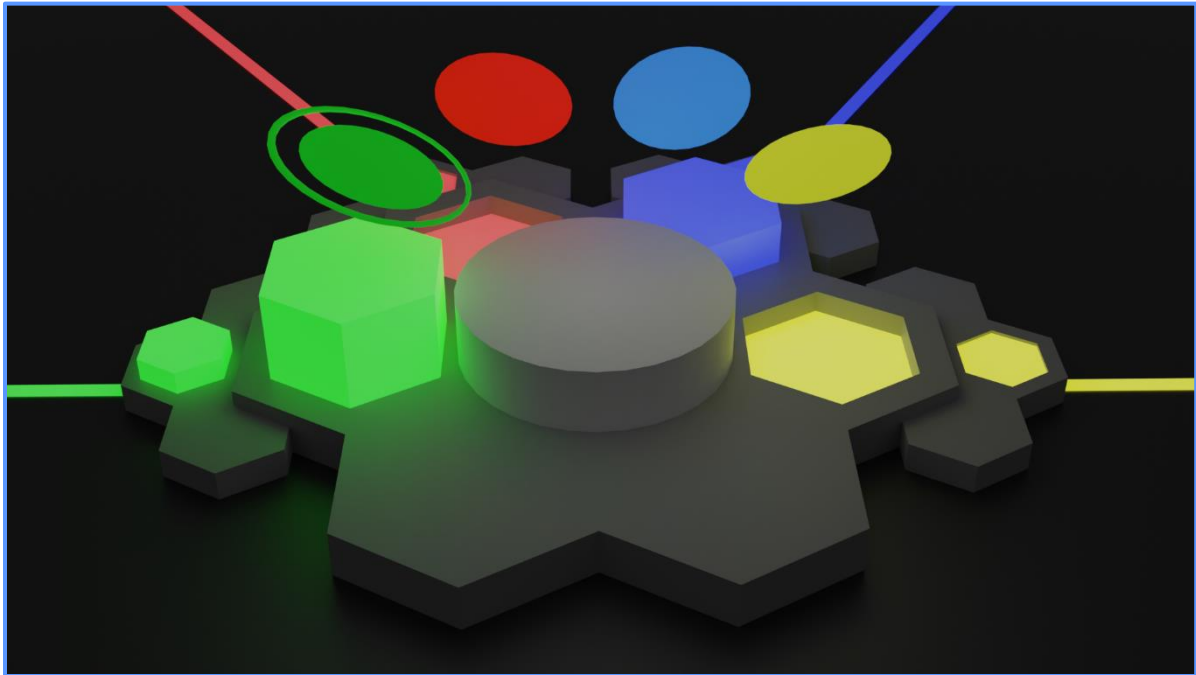
De omslag naar JSON en de implementatie van de Command Stack hebben ervoor gezorgd dat het muterende design van de communicatie tussen UBLEP en ABLEL richting zijn finale vorm gaat. Een onderdeel van dit design is te vinden in *Figuur 15*, waarin de *BleManager* nu niet meer beschikt over elke manier om interactie plaats te laten vinden met ABLEL, maar dit deels gesplitst is naar de *BleCommand* class en classes die hiervan overerven. Niet in het figuur te zien, maar wel te zien in de volledige versie van deze versie van het UML design (*13.2 Unity BLE Package en Android BLE Library UML Design (Fase 3)* (p. 57)) is de omschakeling naar het gebruik van JSON. Wel is hier een deel van te zien, namelijk in de *BleCommand*, waarin de *InformationReceived* methode een *BleObject* als parameter neemt. De *BleObject* is de class waarin de JSON data wordt ge-deserializeerd, wat direct wordt gedaan wanneer de *Adapter* vanuit ABLEL informatie doorgestuurd krijgt.



Figuur 15: Uitgelicht onderdeel van de BleManager met de Command Stack

## 7.4 Heart Drumming

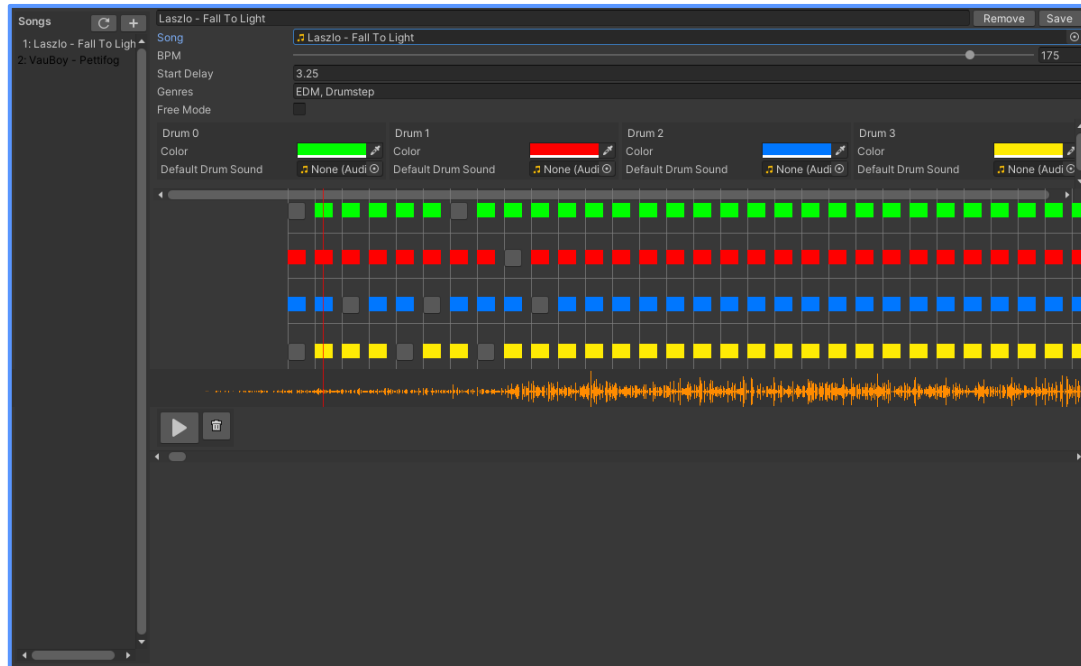
Als uiteindelijke POC is er gekozen om het concept *Heart Drumming* uit te werken. Dit is een concept dat zich voordoet als VRRP applicatie binnen SyncVR Fit. Het doel van Heart Drumming is om armbewegingen te stimuleren, door gebruikers een actie uit te laten voeren op (de maat van) muziek. Hiervoor wordt een virtueel drumstel, dat uit vier hihats bestaat (voor gemakkelijke annotatie wordt dit beschreven als een ‘drum’, hoewel het de vorm heeft van een hihat), voor de gebruiker neergezet en krijgt deze twee drumsticks aangereikt waarmee op de drums geslagen kan worden. Door met een krimpande cirkel aan te geven dat er op de drum geslagen moet worden, kan de gebruiker tijdig reageren en op de drum slaan. Dit bleek na testen geen goede manier van aangeven te zijn en zal worden aangepast naar een nieuw systeem; meer hierover is te lezen in *7.5 Testen en Feedback Loop* (p. 38). Om SyncVR Medical en het SyncVR Fit team een beter beeld te geven bij het concept Heart Drumming, zijn verschillende renders gemaakt in het 3D-modelerings programma Blender. Vanuit het team kwam weinig feedback over de concepten, maar de stagementor was enthousiast over dit idee. In *Figuur 16* valt zo’n concept render te zien, waarbij de groene drum (links) omcirkeld wordt. Dit is een krimpande cirkel die aangeeft dat er op de drum geslagen moet worden wanneer deze het dichtst bij de drum is (hierna te noemen als ‘anticipatie cirkel’).



Figuur 16: Concept render van Heart Drumming

### 7.4.1 Music Editor Window

Om Heart Drumming in de toekomst uit te breiden met meer muziek nummers, of om bepaalde patronen -die gebruikers met hun armen moeten volgen- aan een nummer toe te voegen, is hiervoor een Unity Editor Window opgezet. Deze Editor Window maakt ScriptableObjects aan met een referentie naar het te gebruiken muziknummer, de Beats Per Minute, de time-out voordat het nummer begint, eventuele genres en een grid, om een beat van het nummer een anticipatie cirkel neer te zetten. De afgeronde Editor Window staat in *Figuur 17* (p. 38), beschikt over een ‘Free Mode’. Deze optie maakt gebruik van RNG om een patroon te genereren aan de hand van de moeilijkheidsgraad. Zo hoeft er niet voor elk nummer een vooraf aangegeven patroon gemaakt te worden, wat tijd scheelt.



Figuur 17: Music Editor Window voor Heart Drumming

Voor voorgaande projecten, buiten het afstudeertraject om, is er al ervaring opgedaan met het maken van Unity Editor Windows, waardoor het proces van het ontwikkelen van de Music Editor vrij gemakkelijk ging. Unity biedt standaard een systeem om UI elementen binnen de Editor te maken door middel van de *EditorGUILayout* class, dat methodes aanbiedt om velden aan te maken binnen de window. Door deze slim te combineren en de serializatie van Unity goed te doorgronden, kunnen dit soort Editor Windows snel gemaakt worden.

## 7.5 Testen en Feedback Loop

Tijdens het ontwikkelingsproces is er om de twee weken een feedback-moment. Hierbij presenteert je de huidige staat van de applicatie en vertel je welke taken zijn afgerond en welke taken na deze feedback worden opgepakt. Elke medewerker van elk SyncVR Medical team geeft deze presentatie. Zo kan ook feedback verzameld worden vanuit andere teams over bijvoorbeeld het design van de applicatie, vormgeving van nieuwe features of andere dingen die je presenteert. Omdat je ook aangeeft welke taken je oppakt voor de weken na het feedback moment, kunnen andere teams ook aangeven wat hogere of lagere prioriteit moet krijgen.

Deze feedback-momenten hebben bijgedragen aan het krijgen van actieve feedback over de POC, want de gecompileerde APK wordt ook rondgedeeld binnen SyncVR Medical, zodat ook andere teams kunnen testen. Voor de Corona pandemie konden ontwikkelde applicaties getest worden binnen aangesloten ziekenhuizen om zo feedback te krijgen over de applicatie in de omgeving waar het ook daadwerkelijk gebruikt gaat worden. Tijdens de pandemie kon dit niet meer en werden alleen nieuwe releases van applicaties getest door aangesloten ziekenhuizen. Dit zorg ervoor dat test applicaties alleen getest kunnen worden binnen de teams van SyncVR Medical.

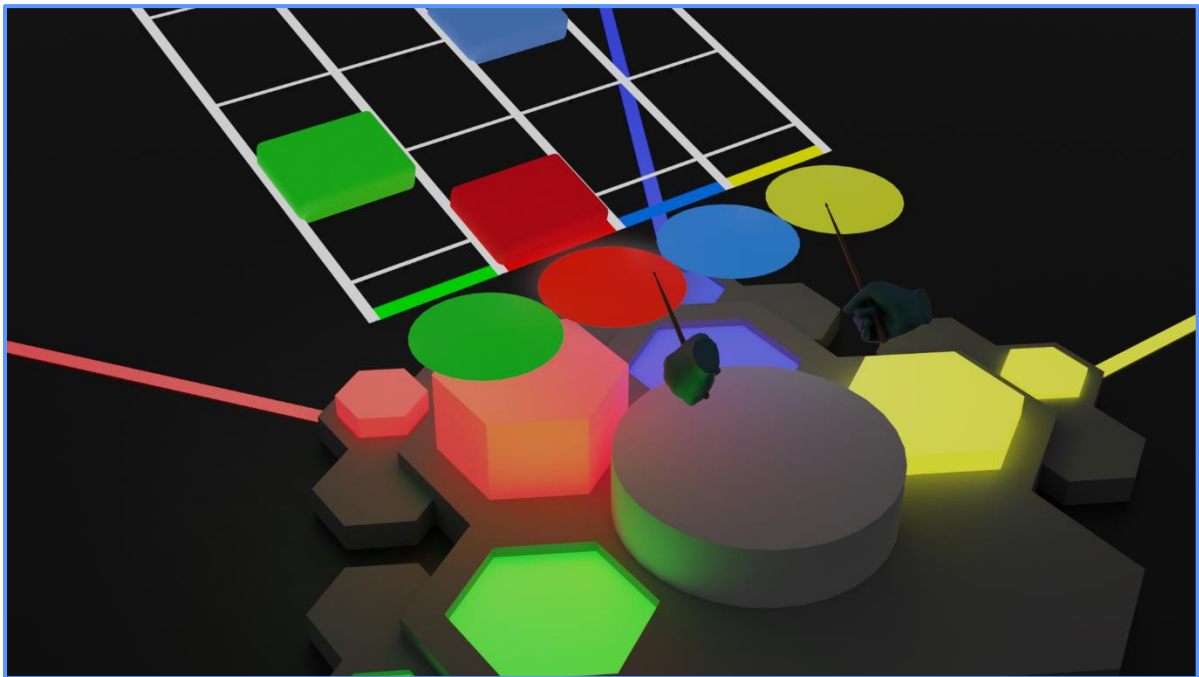
Binnen dit hoofdstuk worden grote veranderingen binnen Heart Drumming genoteerd en voorzien van een uitleg waarom deze veranderingen hebben plaatsgevonden. Al deze veranderingen hebben betrekking op feedback die vergaard is vanuit de teams van SyncVR Medical of uit persoonlijke kringen die Heart Drumming als losse applicatie konden testen.



### 7.5.1 Transportband

Vanuit SyncVR Fit en uit persoonlijke kringen waarin Heart Drumming werd getest kwam feedback dat het zien van de anticipatie cirkel moeilijk wordt vanwege de vele cirkels die tegelijkertijd krimpen (bij hogere moeilijkheidsgraden). Hierdoor kan er mentaal geen volgorde worden gemaakt om op de drums te slaan en raken de gebruikers enigszins in de war, waardoor ze de drum niet- of verkeerd slaan en hierdoor punten missen. Om dit te voorkomen en om het eenvoudiger te maken konden er twee keuzes gemaakt worden: de hogere moeilijkheidsgraad overzichtelijker maken of het systeem aanpassen zodat het eenvoudiger wordt om te anticiperen wanneer een drum geslagen moet worden.

Deze kwestie is met de SyncVR Fit team besproken, waarbij de conclusie was om het systeem aan te passen om het te verduidelijken -naar de gebruikers toe- om op de juiste beat op de drums te slaan. Er is inspiratie opgedaan zoals van games zoals Osu! Mania en Beat Saber, waarbij de timing om op de beat een actie uit te voeren duidelijk wordt aangegeven door een blok naar de speler toe te laten komen. Bij Osu! Mania is dit 2D, bij Beat Saber is dit 3D. Door de eigenschappen van deze twee games te combineren, is uiteindelijk gekozen voor het maken van een soort transportband waarbij de blokken aangeven wanneer de speler op de beat een drum, door deze naar de speler toe te laten komen. Zodoende wordt de volgorde makkelijker te zien, omdat de blokken in een sequentiële volgorde naar de speler toe bewegen. Deze nieuwe manier van aangeven, wordt in *Figuur 18* uitgebeeld als Blender render. Hierbij worden de blokken naar de drums toe vervoert op de witte lijnen die naar beneden bewegen op de maat van de muziek.

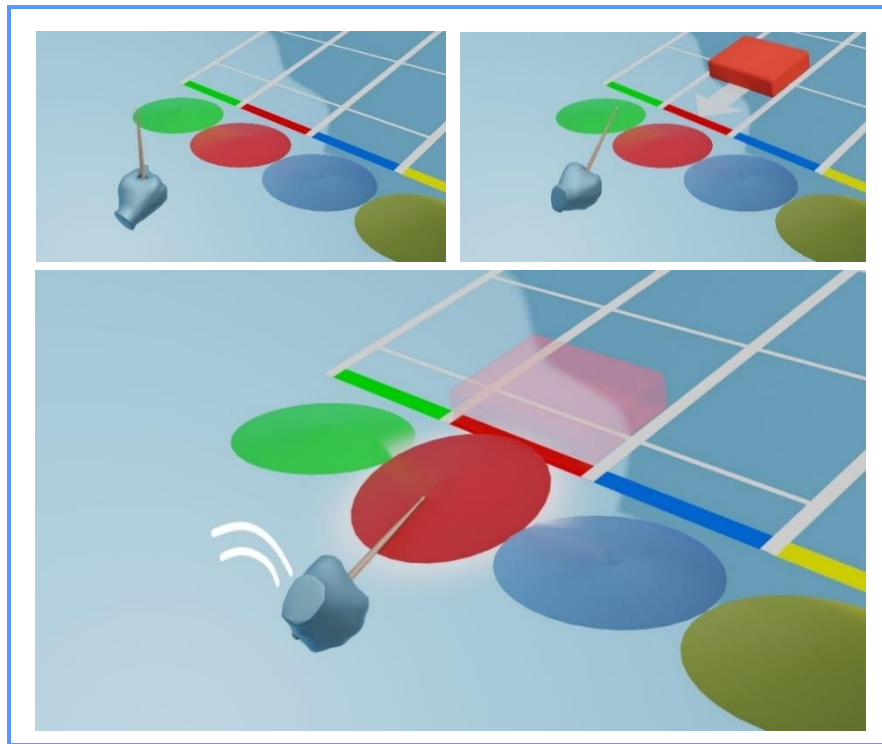


*Figuur 18: Concept render van loopband techniek voor Heart Drumming*



### 7.5.2 Tutorial

Aangezien de gebruikers van de SyncVR Fit applicatie waarschijnlijk niet bekend zijn met andere VR applicaties buiten SyncVR Medical z'n applicaties, zoals Beat Saber, kwam de vraag vanuit de bedrijfsmentor om een tutorial te maken zodat nieuwe spelers Heart Drumming makkelijker op konden pakken. Hiervoor werden de assets genomen die al waren ontwikkeld voor de concept renders voor Heart Drumming en werden de handelingen om de oefening uit te voeren weergegeven aan de hand van drie renders. Deze renders zijn terug te vinden in *Figuur 19*, waarin ze in chronologische volgorde staan weergegeven (van links naar rechts).



*Figuur 19: Blender renders voor Heart Drumming tutorial*

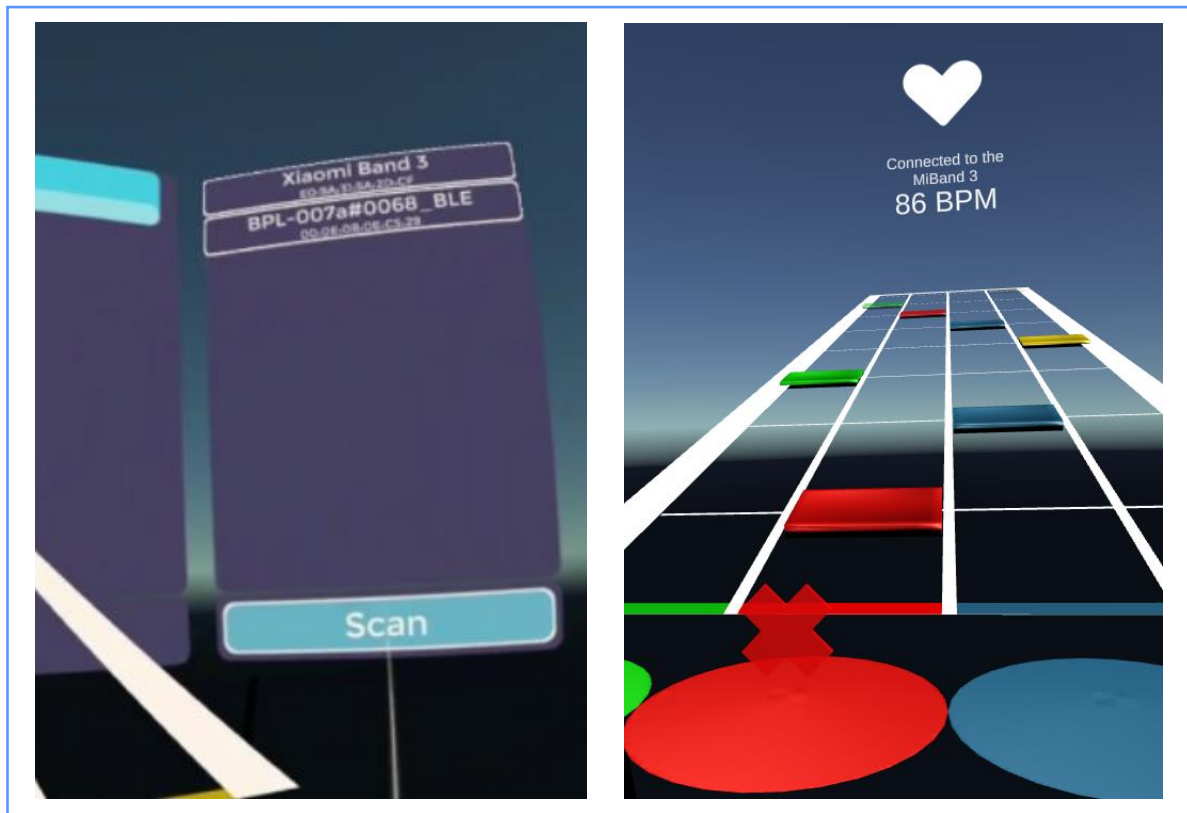
Deze renders zijn in Heart Drumming verwerkt onder een 'Handleiding' knop in het menu. Ook hebben alle renders eigen teksten meegekregen die de lokalisatie-functies gebruiken van Unity. Deze bevatten Noorse, Duitse en Engels vertalingen van de Nederlandse handleiding. Zo kan Heart Drumming ook internationaal gebruikt worden in ziekenhuizen en zorg-instanties buiten Nederlands sprekende landen.

### 7.5.3 Combineren van HEDR en BLE

Aan het eind van het stagetraject werd nog tijd genomen om de twee opgestelde eindproducten te combineren, zodat Heart Drumming als POC voor UBLEP en ABLEL kan staan. Hiervoor werd een aparte branch opgesteld zodat de toevoegingen niet de Heart Drumming branch of de main branch in SyncVR Fit verstoren. Binnen deze branch werd de POC gecombineerd met UBLEP en ABLEL om te laten zien dat de werkingen van BLE te combineren zijn met een SyncVR Fit oefening. Met een naderende deadline werden manieren om BLE invloed uit te laten oefenen op Heart Drumming qua complexiteit vereenvoudigd. Eerst was het plan om de oefening te laten stoppen of om de moeilijkheid van de oefening naar beneden te schalen wanneer de hartslag van de gebruiker boven een bepaalde

drempel komt. Nu is dit vereenvoudigd door het tonen van de hartslag aan de hand van een icoon in de wereld en een veld dat de hartslag laat zien.

Om dit te implementeren en de gebruiker de optie te geven om een wearable uit te laten zoeken binnen de oefening, wordt dit aan de hand van een menu met opties voor BLE weergegeven en eventuele BLE apparaten die in de buurt zijn gevonden. Als de wearable is verbonden verdwijnt het configuratie scherm en wordt de naam van de wearable getoond boven de oefening. Wanneer de wearable is verbonden wordt er ook ge-Subscribed op de hartslag characteristic van de wearable (als deze aanwezig is) en wordt de data die deze characteristic doorgeeft, weergegeven. In *Figuur 20* zijn er screenshots weergegeven hoe deze panels eruit zien, met links een lijst aan gevonden BLE apparaten in de buurt van de VR headset. Hiervoor worden apparaten 10 seconden gescand, waarna de gebruiker op de 'Scan' knop kan drukken als het gewenste apparaat niet in de lijst staat. Als het apparaat is verbonden dan wordt de naam van het apparaat getoond met de waarde van de hartslag characteristic daaronder (rechts te zien op *Figuur 20*).



*Figuur 20: Screenshots van Heart Drumming in combinatie met BLE*

*Links: Configuratie scherm voor BLE om te verbinden met een gevonden device*

*Rechts: BLE informatie over de hartslag dat wordt uitgelezen uit de MiBand 3*

Door UBLEP en ABLEL te combineren met Heart Drumming, dient Heart Drumming als een POC om weer te geven dat BLE werkt op een VR headset. Met deze branch is aangegeven dat een SyncVR Fit oefening die als VRRP oefening toe doet BLE ondersteuning kan krijgen met de opgestelde eindproducten.

## 8 Eindproduct

In dit hoofdstuk wordt het resultaat van het afstudeertraject vastgelegd. De eindproducten worden hier genoemd en er wordt geconstateerd of de doelstelling is bereikt en daarmee de probleemstelling is opgelost met de ontwikkelde producten.

Door de POC Heart Drumming te combineren met BLE, is het mogelijk gemaakt voor SyncVR Medical om BLE te combineren met ontwikkelde oefeningen in Unity. Door de combinatie van UBLEP en ABLEL te gebruiken, kunnen wearables verbonden worden met ontwikkelde VRRP en andere soorten oefeningen binnen SyncVR Medical.

De gebruikte manier om Unity applicaties op VR headsets BLE ondersteuning te geven is door de Android JNI te gebruiken in combinatie met een Android Java library dat communiceert met Android's Bluetooth protocollen. De BLE informatie wordt dan doorgegeven via de UnityPlayer naar de applicatie. Deze informatie wordt daarna opgevangen door een command wat een actie uitvoert aan de hand van de geleverde JSON informatie. Op die manier kan een Unity applicatie op een VR headset communiceren met externe BLE wearables en de informatie daarvan uitlezen.

Het eindproducten UBLEP en ABLEL houden zich nog niet aan alle vastgestelde requirements, waardoor deze methode nog niet volledig te gebruiken is binnen SyncVR Medical om BLE functionaliteiten te gebruiken binnen applicaties die in Unity zijn ontwikkeld. Wel tonen ze aan dat het mogelijk is om te communiceren vanaf een Unity applicatie naar een wearable via BLE, maar dat deze nog niet volledig getest zijn aan de hand van de gegeven requirements.

Met deze eindproducten is het mogelijk dat er succesvol een wearable (MiBand 3) verbonden wordt met verschillende VR headsets (*Pico Neo 2* en *Oculus Quest*) door gebruik te maken van de onderliggende Android BLE functionaliteiten. Ook kan er data verstuurd worden naar de wearable en data ontvangen worden vanuit de wearable. characteristics kunnen gebruikt worden om data te schrijven naar de wearable en om data op bepaalde manieren te ontvangen (*Read* en *Subscribe / Notification*).

Eindproducten UBLEP en ABLEL worden gecombineerd in een Unity Package, wat een archief is voor één of meerdere bestanden in een enkel bestand. Unity kan met behulp van zo'n Package de bestanden op de juiste plek verplaatsen (UBLEP onder *Scripts/BLE* en ABLEL onder *Plugins/Android*), zo worden de bestanden op de juiste plek gezet zodat deze correct mee worden gecompileerd en gebruikt kunnen worden.

## 9 Aanbeveling

Tijdens het afstudeertraject is er een aantal punten naar voren gekomen waar een aanbeveling op gegeven kan worden. Dit zijn aanbevelingen aan de hand van de technieken waarmee is gewerkt, de manier waarop er gewerkt wordt binnen SyncVR Fit en de structuur waarop platforms zijn ingedeeld binnen SyncVR Medical.

### 9.1 Git en GitHub

Alhoewel Git in combinatie met GitHub mogelijkheden biedt om verschillende projecten te managen, is het platform niet volledig geschikt om dit te doen. GitHub biedt een *Enterprise* optie aan voor bedrijven om betere tools aan te bieden om producten te ontwikkelen binnen grotere bedrijven. SyncVR Medical beschikt niet over zo'n Enterprise optie en onderhoudt alles via de Teams functie, een manier om gebruikers en repositories te binden aan een bepaald Team dat onder een GitHub organisatie hoort.

Binnen SyncVR Medical wordt er in verschillende teams gewerkt, maar is er één generieke organisatie opgezet voor alle teams, waarin elk team zijn eigen (of meerdere) repository heeft. Dit zorgt ervoor dat alle leden van deze organisatie allemaal individuele rechten hebben die handmatig geregeld moeten worden (bijvoorbeeld schrijfrechten voor een bepaalde repository, of het ontnemen hiervan). Hiervoor heeft GitHub zogenoemde 'child teams', een functie binnen een organisatie om de leden hiervan op te splitsen in kleinere teams, waarvan de rechten per groep bepaald kunnen worden.

Ook hebben de repositories verschillende soorten standaarden wanneer het aankomt op het noemen van branches, het regelen van pull requests en de manier waarop branches gebruikt worden. Bij SyncVR Fit bijvoorbeeld, heeft elk lid zijn eigen branch voor een bepaalde oefening die zij aan het ontwikkelen zijn. Deze hebben allemaal een ander soort naming convention. Dit is op zichzelf geen groot probleem, maar dit wordt het wel wanneer veel leden verschillende soorten branches gaan aanmaken die niet verwijderd worden. Zo kan er geen overzicht gehouden worden welke branches bij welke oefening horen en onder welk lid deze valt, of het feature branches zijn of dat de branch stale is en al gemerged is.

Ook al zou elk lid een feature branch aanmaken, dan wordt het uiteindelijk ook niet overzichtelijk. Als elk lid voor elke feature een nieuwe branch aanmaakt, dan kan het aantal branches gemakkelijk oplopen tot zo'n tien tot twintig verschillende branches. Een aanbeveling hiervoor zou zijn om stagiaires en leden van teams forks te maken voor een nieuwe oefening waar degene zich op dat moment mee bezig houdt. Zoals het maken van een pull request wanneer er een grote ontwikkeling is binnen een bepaalde oefening. Deze forks kunnen zijn eigen naming convention aanhouden, zonder dat de main repository oploopt met verschillende soorten branches.

## 9.2 Unity

Binnen de SyncVR Fit repository is de folder structuur voor Unity niet overzichtelijk. De *Assets* folder is gevuld met verschillende plugins en packages die zijn geïmporteerd, maar niet zijn geordend of verplaatst naar de juiste folder. Als er meerdere developers werken aan hetzelfde project is het handig om een folder structuur te gebruiken, zodat elke developer weet waar nieuwe files geplaatst moeten worden en waar andere files staan. Van origine heeft SyncVR Medical wel een folder structuur voor Unity en andere projecten, maar de SyncVR Fit repository gebruikt deze folder structuur niet. Wel wordt er een vorm van een folder structuur aangehouden in een folder binnen de *Assets* folder. Binnen deze folder wordt wel de SyncVR Medical Unity folder structuur gehanteerd.

Ook wordt er geen gebruikgemaakt van namespaces. Namespaces worden gehanteerd om een bepaald object te identificeren binnen de codebase om zo objecten te structureren en groeperen binnen het project. Omdat de meeste code dat al geschreven is en geschreven wordt onderdeel wordt van de *global* namespace, zodat functies en classes die onderdeel zijn van de *global* namespace overal aangeroepen kunnen worden.

Het niet toevoegen van namespaces gaat hand in hand met het gebrek aan een code conventie. Door het SyncVR Fit project worden verschillende stijlen gebruikt qua code conventie, wat het onoverzichtelijk maakt en er een mogelijkheid bestaat dat dit niet conform OOP gaat (wegens de manier waarop Unity properties van een class laat zien binnen de editor). Een aanbeveling hiervoor is om te kijken naar code standaarden en om deze te combineren met een software kwaliteit standaard, zoals de software kwaliteit levels van de ‘*Institute for Software Quality*’<sup>10</sup> (IfSQ). Wanneer ontwikkelaars zich aan een standaard houden, wordt geschreven code gestructureerd en overzichtelijker.

---

<sup>10</sup> [ifSO Standards](#)

## 10 Evaluatie en Reflectie

Door te reflecteren op de gemaakte producten en hoe mijn proces is verlopen, wordt er voor mij persoonlijk een beter beeld gegeven hoe ik in de toekomst processen en het ontwikkelen van producten kan verbeteren.

### 10.1 Evaluatie

Tijdens het afstudeertraject is er een aantal producten opgeleverd, waaronder tussen- en eindproducten. Deze zijn aan het eind of binnen in het stagetraject overgeleverd aan het bedrijf of aan andere partijen waar de producten betrekking bij hebben.

#### 10.1.1 Unity Bluetooth Low Energy Package (UBLEP)

Ik zie C# als mijn beste taal als het om programmeren gaat, vooral omdat ik hier veel mee gewerkt heb tijdens een minor Game Development & Simulation, waar ervaring is opgedaan met de Unity 3D engine. Ook wordt er voornamelijk met C# gewerkt binnen de opleiding Software Engineering op De Haagse Hogeschool Zoetermeer. Dit betekent niet dat mijn ervaring met C# goed is, maar ik weet er genoeg van af om zonder problemen een goede package te schrijven die mensen buiten mijn directe omgeving kunnen gebruiken. Voor UBLEP heb ik een aantal design patterns gebruikt die nieuw zijn, maar waarvan ik de basis al een keer heb meegekregen. Door deze patterns goed te bestuderen en UML diagrammen op te stellen hoe zo'n pattern toegepast zou kunnen worden in UBLEP, zorgde ik ervoor dat de patterns zo goed mogelijk zijn toegepast.

Graag had ik Unit Tests toe willen voegen om UBLEP te kunnen testen en om te zien of er methodes of classes zijn die beter geoptimaliseerd kunnen worden. Hier was geen tijd meer voor, omdat UBLEP met vaart getest moest worden en vast bleef zitten in de ontwikkelingsfase van SyncVR Medical, waarbij feedback vergaard wordt en per iteratie wordt toegepast. Hier kwam het gebrek aan requirements bij, omdat er geen vastgezet punt was waarop gezegd kon worden dat UBLEP afgerond en klaar om getest te worden was. Ook zou er dan veel tijd en aandacht naar de Unit Tests gaan, omdat het afhangt van een externe library (ABLEL). Hiervoor zou dan een Stub opgesteld moeten worden om de library na te bootsen zodat deze getest kon worden. Anders worden de Unit Tests te dicht opgesteld aan de beschikbare MiBand 3, omdat er geen andere generieke BLE peripheral was om mee te testen.

Het combineren van UBLEP en een Use-Case diagram zou ook hebben geholpen om een beter beeld te krijgen welke functies er allemaal beschikbaar moeten zijn om genoeg interactie te hebben met BLE. Door deze diagrammen op te stellen zou het makkelijker zijn om te zien welke methodes nog aangemaakt moeten worden, in plaats van dat deze worden aangemaakt wanneer ik er behoefte aan had tijdens het ontwikkelproces. Dit kwam samen met het feit dat er niemand beschikbaar was voor het testen of gebruiken van UBLEP en ABLEL om feedback te krijgen over hoe het is om met UBLEP te ontwikkelen.

Omdat GATT werkt met profielen (zoals het hartslag monitor profiel), zou het handig zijn geweest als er ook een provider binnen UBLEP was waar verschillende profielen aangevraagd kunnen worden. Op die manier kan de informatie van veelgebruikte GATT profielen direct worden opgevraagd, zoals de UUID's van de characteristics van het opgevraagde profiel.

### 10.1.2 Android Bluetooth Low Energy Library (ABLEL)

Alhoewel Java en C# op elkaar lijken als het gaat om mogelijkheden (zoals OOP, overerving en grotendeels dezelfde syntax), heb ik weinig ervaring met het schrijven van effectieve Java code. Effectieve Java code houdt in dat ik geen rekening heb gehouden met eventuele optimalisatie technieken die niet zijn meegegeven in de artikelen die zijn gebruikt om de producten te ontwikkelen. Dit heeft ervoor gezorgd dat ondanks het functioneren van de library, er geen technieken zijn toegepast die de library eventueel sneller laat werken, om te compenseren voor de bottleneck van de Android JNI.

Ook ben ik niet bekend met de code conventie van Java of Android en welke er wordt toegepast op een Android Java Library voor Unity, dus is de code geschreven aan de hand van de code conventie van Android. Dit kan voor problemen in de toekomst zorgen wanneer een ontwikkelaar het project moet uitbreiden met nieuwe functies en juist de Java conventie wil gebruiken.

### 10.1.3 Heart Drumming (HEDR)

Het idee van Heart Drumming is heel leuk, omdat het lijkt op andere muziek-gebaseerde VR games. Deze games heb ik zelf ook vaak gespeeld, dus het ontwikkelen van een soortgelijk spel was tof om te doen. Zelf vind ik dat de gameplay goed is geïmplementeerd en dat met behulp van de tutorial het makkelijk is om de oefening te begrijpen en zelf uit te voeren.

Het liefst had ik een klein deskresearch of onderzoek gedaan over VRRP om beter grip te krijgen op wat zich een VRRP kan noemen en hoe dat toegepast kan worden om SyncVR Fit oefeningen. Hier zijn research documenten over, met links op de website van SyncVR Medical, die ik door had willen lezen. Dit lukte niet tijdgebrek en voorbereiding. Wel geloof ik dat Heart Drumming een oefening is dat past binnen SyncVR Fit en VRRP oefeningen.

### 10.1.4 Music Editor Window (Music Editor)

Aangezien er weinig tijd in een stagetraject zit, kon de Music Editor Window niet volledige gebruikt worden waarvoor het was ontworpen. Origineel is de Music Editor gemaakt om patronen bij stukken van nummers toe te voegen om zo een bepaalde beweging te stimuleren. Ook hier was geen tijd voor om deze patronen te maken en aan nummers toe te voegen. Wel heb ik het kunnen gebruiken om nieuwe nummers compatibel te maken voor Heart Drumming door de juiste parameters aan de nummers te koppelen en een Free Mode toe te voegen.

Graag zou ik willen dat Heart Drumming een blijvende oefening binnen SyncVR Fit blijft en dat de Music Editor een gebruikt onderdeel wordt voor de workflow van Heart Drumming. De tool maakt het namelijk eenvoudig om nieuwe nummers toepasbaar te maken voor Heart Drumming.



## 10.2 Reflectie

Ik zal starten met dat ik al een tijd lang niet goed in mijn vel heb gezeten. Door de Corona pandemie en verschillende lockdowns is dit verergerd naar dat ik mezelf moeilijk emotioneel in stand kon houden. Door een moeilijke start met een volledig nieuw proces zoals afstuderen, voelde ik me dus ook gelijk niet goed en was ik bang dat ik het proces fout zou doorlopen en mijn afstudeertraject hierdoor niet kon afmaken.

Door dit te bespreken met mijn begeleidend en expert examiner werd ik de goede kant op geduwd om het zelf op te lossen met het bedrijf en werd duidelijk dat dit een onderdeel van het proces was. Al vond ik dit ontzettend moeilijk om te begrijpen, heb ik door de zure appel heen gebeten en ben ik doorgegaan met het vast- en samenstellen van mijn afstudeerplan om een HBO waardige afstudeer stage af te leggen. Zonder mijn vrienden en mede studiegenoten had ik dit niet kunnen doen, zij hebben mij zeker geholpen door mij moed en inspiratie te geven te volharden en om het traject af te maken.

Daarnaast kijk ik terug op een vrij hectisch afstudeertraject. Met vele lockdowns en communicatie dat voornamelijk online plaatsvindt, heb ik de planning die in het PvA was opgesteld behoorlijk moeten versoepelen. Zo heb ik de BLE package afgeschreven in week tien in plaats van week zes dat origineel was opgesteld in het plan. Dit had echter geen negatieve impact op de eindproducten, omdat alle producten uiteindelijk volledig zijn afgerond en in een overdraagbare staat aan SyncVR Medical zijn afgedragen.

Door deze lockdowns was het ook lastig communiceren. Het contact verliep via Rocket Chat of Google Hangout meetings, wat het aan het begin van het afstudeertraject moeilijk maakte om op dezelfde lijn te zitten met de bedrijfsmentor. Uiteindelijk is dit goed gekomen, maar zelf denk ik dat contact waarbij je elkaar kan zien (zonder webcam) een stuk effectiever is (ook door non-verbale communicatie bv).

Naast deze enkele negatieve ervaringen, ben ik trots op wat ik heb ontwikkeld, hoe ik het probleem heb aangepakt en de eind producten die uit dit afstudeertraject zijn voortgekomen! Ik vind het erg leuk dat ik deze unieke mogelijkheid om een onderbelicht onderwerp aan te pakken en een bijpassende oplossing hiervoor te vinden heb kunnen nemen. Ik zag enigszins tegenop om te beginnen voor een commercieel bedrijf, maar ik ervaar dit nu juist als een positieve ervaring.

Het proces was aan het begin wennen. Tot nu toe zijn projecten waar er vaste regels aan het begin van het traject waren gesteld de projecten waar ik het meest comfortabel mee was. Daarom had ik moeite met de start binnen SyncVR Medical; er zijn wel guidelines geschreven, maar weinig requirements uit deze guidelines waren voorhanden. Ook was mijn bedrijfsmentor vrij nonchalant in de oplossing die ik voor het probleem kon ontwikkelen, waardoor het krijgen van requirements vrij stroef verliep. Uiteindelijk heb ik het faseringsproces van SyncVR Medical begrepen en heb ik mezelf aangepast aan de manier van ontwikkelen binnen het bedrijf. De requirements kwamen uiteindelijk vanzelf naarmate ik meer had ontwikkeld.

Voor de planning had ik voor een ruimere indeling van de onderdelen moeten kiezen. De tijd voor realisatie van de VR Headset om met een wearable te communiceren, heeft uiteindelijk meer tijd gekost dan gepland. Hieruit kwam ik tot de conclusie dat ik mijn tijd beter had moeten verdelen voor het ontwikkelen van Heart Drumming en UBLEP/ABLEL. Door de planning ruimer te maken of beter in te delen zou er ook meer tijd ontstaan om aan meer requirements te voldoen of om de eindproducten beter te testen.



## 10.3 Beroepstaken

Gedurende het stagetraject is er een aantal beroepstaken van ICT, Software Engineering en Game Development & Simulation behandeld. Deze beroepstaken zijn opgesteld aan de hand van het afstudeerplan dat voor het stagetraject is opgesteld. Tijdens het afstudeertraject kunnen plannen wijzigen waarbij de onderbouwing om te voldoen aan een beroepstaak gewijzigd. Als zo'n situatie zich voordeed is dit aangekaart met de examinatoren en is er aan de hand van de situatie een aangepast afstudeerplan opgesteld. Om weer te geven dat er is voldaan aan elk opgestelde beroepstaak, worden ze in deze paragraaf onderbouwd waarom aan de beroepstaak is voldaan en wat de ervaring was om aan deze beroepstaak te voldoen.

### A1 – Analyseren probleemdomein & opstellen probleemstelling

Omdat er weinig bekend is over het combineren van VR headsets en wearables, moest het probleem van SyncVR Medical goed geanalyseerd worden, zodat eventuele resultaten een contributie kunnen maken aan het vorderen van deze ontwikkeling (3 *Opdrachtomschrijving* (p. 9)). Daarom is het probleem goed bestudeerd en is er gekeken hoe de wearables te verbinden met VR headsets en het uitlezen van de data.

Hieruit is uiteindelijk gekomen dat er een BLE oplossing komt, waarbij alternatieven zijn besproken met de bedrijfsmentor en BLE als beste oplossing voorkwam (4.112 *Bluetooth Low Energy* (p. 12)). Ook omdat BLE voldoet aan de requirements vergaart en zijn opgesteld, wat het een goede keuze is geweest om te gebruiken voor het verbinden van VR headsets met wearables.

### C9 – Ontwerpen game, computermodel en simulatie

Ter voorbereiding van het ontwikkelen van een Proof of Concept op het BLE is er een aantal concept documenten opgesteld om het idee van BLE, een SyncVR Fit oefening en een wearable te combineren. Hierbij is het High Concept Document Template gebruikt om deze concepten zo effectief mogelijk te omschrijven binnen een aantal hoofdstukken. Aan de hand van zo'n opgesteld concept document is een SyncVR Fit oefening ontwikkeld binnen Unity dat zich als game voordoet. De ontwikkelde oefening bevat game elementen die beschreven zijn in het concept document. Het ontwikkelproces wordt beschreven in 7.4 *Heart Drumming* (p. 37) en gaat in op het toevoegen van nieuwe elementen en het uiteindelijke koppelen met wearables.

### D1 – Realiseren van software

Tijdens het afstudeertraject zijn er drie eindproducten opgeleverd die als software gezien worden, namelijk UBLEP, ABLEL en Heart Drumming (7 *Realiseren* (p. 23)). Deze stukken aan software hebben alle drie hun eigen doel en zijn elk ontwikkeld aan de hand van een andere techniek. Voor Heart Drumming is er gebruikgemaakt van een High Concept Document om het ontwikkelen van de applicatie voor te bereiden. Ook zijn concept afbeeldingen ontwikkeld door middel van Blender om een beter beeld te geven bij de applicatie. Voor onderdelen van UBLEP en ABLEL zijn er UML diagrammen uitgeschreven, en zijn gegenereerd in PlantUML om een beter overzicht te krijgen: wat er precies ontwikkeld wordt en hoe sommige onderdelen met elkaar gaan samenwerken. Uiteindelijk zijn UBLEP en ABLEL ontwikkeld aan de hand van deze UML diagrammen en zijn de diagrammen uitgebreid en aangepast naarmate er nieuwe requirements werden opgesteld of wanneer er nieuwe kennis was opgedaan om het product te verbeteren.

### Ga – Effectief (internationaal) communiceren

Het realiseren van producten tijdens het stageproject zijn ontwikkeld binnen het SyncVR Fit team. Met dit team werd elke maandag een meeting gehouden om te weten met welke taken de leden binnen het team mee bezig waren en of er eventueel andere taken verdeeld kunnen worden binnen SyncVR Fit, zoals het verwerken van feedback over al-ontwikkelde applicaties. Hierbij was de bedrijfsmentor aanwezig om overzicht te houden op welke taken het team mee bezig was en om eventueel sturing te geven als dit nodig is. Bij deze meetings werd er goed gecommuniceerd om een overzicht te kunnen maken van de taken die de teamleden die week zouden uitvoeren, en werden genoteerd door een teamlid.

Aan het begin van het stagetraject was het veel observeren om mijn plek binnen SyncVR Medical te vinden en om een goede start te maken aan UBLEP, ABLEL en Heart Drumming. Hierbij kwamen er problemen met verschillende packages die worden gebruikt binnen de SyncVR Fit applicatie waar nog niet eerder mee is gewerkt. Dit kon ik aankaarten bij de technische begeleider binnen het SyncVR Fit team die, wanneer een goede omschrijving van het probleem is gestuurd, een goede uitleg kon geven over waarom het probleem zich voordeed en hoe deze te omzeilen valt.

Ook was er aan het begin van het stagetraject een probleem in de communicatie tussen het stagebedrijf en de school-examinatoren. Hierbij ging het om verschillende aanpassingen van het stageplan om de huidige stand van zaken te kunnen verwerken, waarbij de originele plannen die in het stageplan uiteen waren gezet, aangepast of omgegooid moesten worden. Dit vond plaats binnen de eerste twee weken van het stagetraject en werd op tijd aangekaart met de expert- en begeleidend examinator, die vervolgens elk, onafhankelijk, een meeting met mij hebben aangemaakt om hierover te praten. Zo heb ik mijn persoonlijke ervaring over de situatie kunnen delen met de examinatoren en bleek dat het niet erg was dat het plan zou worden aangepast, als er maar goed met de examinatoren en het bedrijf zou worden gecommuniceerd wát er aangepast moest worden. N.a.v. van deze meeting met de examinatoren is er nog een meeting ingepland met de bedrijfsmentor, waarbij ik liet weten hoe ik er op dat moment in stond met het afstuderen, en de vele wijzigen van de originele plannen die in het afstudeerplan waren beschreven.

Door deze gesprekken was ik enorm gerustgesteld dat ik gewoon mijn afstudeertraject kon afmaken met wat wijzingen aan het originele afstudeerplan. Deze is dan ook aangepast en gestuurd naar mijn examinatoren voor feedback en goedkeuring.

### Gc – Kritisch, onderzoekend en methodisch werken

Om de eindproducten te ontwikkelen is er gewerkt met compleet nieuwe technologieën waar ik nog geen ervaring mee had (*4 Theoretisch Kader (p. 12)*). Het combineren van Virtual Reality, de Bluetooth mogelijkheden die de VR headsets beschikken en die te combineren met Unity en wearables is een vrij unieke combinatie waar geen specifieke documenten of artikelen over te vinden zijn. Dit zorgde ervoor dat er geen complete stap-voor-stap uitleg was over hoe Virtual Reality te combineren is met Bluetooth Low Energy. Door deskresearch te doen en de vindingen hiervan te noteren als een eigen stappenplan, kon ik constateren of er per stap een bepaalde onderzoeksmethode gebruikt kon worden.

## Gf – Leren leren: voorbereiden op volgende studiefase en beroep

Voor het afstuderen had ik voor mezelf als doel gesteld dat ik een opdracht zou vinden waarmee ik in de toekomst aan de slag wil. Hierbij sloot SyncVR Medical ontzettend goed aan, omdat ze daar werken met Unity en C#, twee technieken waar ik in de toekomst ook graag mee wil werken; dit is wat je niet op school kunt leren, de mate van variatie en uniciteit van een bedrijf en de mate van de geleerde technieken toe te kunnen passen binnen hun omgeving.

Het toepassen van al geleerde technieken (Unity, C# en Android development) in een nieuwe bedrijfsomgeving was ontzettend leerzaam en geeft mij een doel om deze technieken nog beter te kennen in verschillende omgevingen, om zo een groter kennisveld op te bouwen.

Ook het combineren van Unity en Android, welke ik allebei apart heb geleerd, was ook een unieke ervaring. In plaats van dat Unity alleen compileert naar Android en de engine alles regelt, moet er een deel van de werking binnen Android zelf ontwikkeld worden. Dit is bijzonder interessant en brengt mij ook meer bij over het combineren van verschillende technieken en welke synergie deze kunnen bereiken.

Het combineren van deze technieken werd ondersteund door de medewerkers binnen SyncVR Medical, aan wie ik vragen kon stellen (voor of tijdens het aandienen van een probleem). Mijn collega's binnen SyncVR Medical konden me dan verder helpen binnen het proces en mij advies geven die ik weer kon toepassen. Hierbij heb ik geleerd dat communicatie belangrijk is om problemen op te lossen en te bespreken met mensen die meer kennis hebben of mogelijkheden hebben om die kennis aan mij over te kunnen dragen.

Ook heb ik geleerd om eerder naar collega's te stappen die meer kennis of ervaring hebben met bepaalde onderwerpen waar ik tegenaan liep. Zo heb ik vaak zelf naar oplossing gezocht voor kleine problemen met bijvoorbeeld het compileren naar een bepaalde VR headset. Dit heb ik een aantal keer gedaan, omdat ik onzeker was over het vragen van zo'n kleine bug aan collega's. Maar na de eerste keer dat ze het probleem sneller hebben kunnen verhelpen, heb ik mezelf aangepast om eerder naar collega's te stappen met vragen.

## 11 Verwijzingen

Binnen het afstudeerdossier worden binnen de teksten verwijzingen gemaakt naar figuren, tabellen of andere veel voorkomende- of afgekorte benamingen. Deze zijn in dit hoofdstuk opgenomen voor snelle verwijzingen naar de desbetreffende figuur, tabel of andere verwijzingen.

### 11.1 Figuren

Figuur 1: Organogram van het bedrijf SyncVR Medical .....	7
Figuur 2: Bluetooth Low Energy Heart Rate Measurement profiel z'n unieke identifiers .....	13
Figuur 3: Opzet van een BLE apparaat met een Generic ATtribute Profile .....	13
Figuur 4: UML van Unity's Composite Pattern implementatie.....	16
Figuur 5: Simpele notatie van het Adapter Pattern .....	17
Figuur 6: Fases voor het ontwikkelen van applicaties binnen SyncVR Medical .....	18
Figuur 7: Eerste ontwerp voor de BLE implementatie binnen Unity.....	27
Figuur 8: DiscoveredDevice informatie dat vanuit ABLEL wordt verstuurd naar UBLEP .....	29
Figuur 9: Onderdeel van het UML design voor communicatie tussen UBLEP en ABLEL .....	30
Figuur 10: Screenshot van de LightBlue applicatie over de MiBand 3 .....	31
Figuur 11: MiBand 3 authenticatie proces. ....	32
Figuur 12: Command Stack weergegeven in een UML Diagram .....	34
Figuur 13: Code om te verbinden met Hartslag characteristic wanneer de wearable verbonden is....	35
Figuur 14: JSON data dat vanuit ABLEL naar UBLEP wordt gestuurd.....	36
Figuur 15: Uitgelicht onderdeel van de BleManager met de Command Stack.....	36
Figuur 16: Concept render van Heart Drumming .....	37
Figuur 17: Music Editor Window voor Heart Drumming.....	38
Figuur 18: Concept render van loopband techniek voor Heart Drumming.....	39
Figuur 19: Blender renders voor Heart Drumming tutorial .....	40
Figuur 20: Screenshots van Heart Drumming in combinatie met BLE .....	41
Figuur 21: Planning over de tijdsperiode van het afstudeer traject.....	68
Figuur 22: Concept- and example render of Heart Drumming .....	74

### 11.2 Tabellen

Tabel 1: Tabel en afkortingen voor eindproducten .....	11
Tabel 2: Tabel en afkortingen voor tussenproducten .....	11
Tabel 3: Relevant gedrag van BLE characteristic en de data binnen z'n descriptor.....	14
Tabel 4: Requirements ter het ontwikkelen van de BLE package .....	22
Tabel 5: Technische requirements ter het ontwikkelen van een SyncVR Fit applicatie (POC).....	22
Tabel 6: Stappenplan ter voorbereiding van het opzetten van Bluetooth Low Energy .....	26
Tabel 7: Afkortingen en de benamingen hiervan binnen het Afstudeerdossier .....	52
Tabel 8: Personen en betrokkenen binnen het afstudeertraject .....	64
Tabel 9: Risico's binnen het afstudeertraject .....	65
Tabel 10: Feature Cards template van SyncVR Medical .....	67
Tabel 11: Overzicht tabel voor academische deadlines (Hague University of Applied Sciences) .....	69

## 11.3 Afkortingen

Hieronder vindt u een tabel die de verschillende afkortingen die genoemd worden binnen dit document overzichtelijk weergeeft. Achter de volledige benaming van de afkorting, staat het hoofdstuk waarin de afkorting voor het eerst wordt vermeld. Dit is om eventueel meer informatie over de afkorting op te zoeken, of om verwarring te voorkomen als de afkorting voor het genoemde hoofdstuk is gebruikt.

Tabel 7: Afkortingen en de benamingen hiervan binnen het Afstudeerdossier

Afkorting	Volledige Benaming
<i>ABLEL</i>	Android Bluetooth Low Energy Library
<i>AOT</i>	Ahead-Of-Time
<i>API</i>	Application Programming Interface
<i>BLE</i>	Bluetooth Low Energy
<i>DOF</i>	Degrees of Freedom
<i>GATT</i>	Generic ATtribute
<i>HEDR</i>	Heart Drumming
<i>HMD</i>	Head Mounted Display
<i>JAR</i>	Java Archive
<i>JAR</i>	Java ARchive
<i>JIT</i>	Just-In-Time
<i>JNI</i>	Java Native Interface
<i>JVM</i>	Java Virtual Machine
<i>MiBand 3</i>	Xiaomi Band 3
<i>OS</i>	Operating System
<i>POC</i>	Proof of Concept
<i>PVA</i>	Plan Van Aanpak
<i>SOLID</i>	Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion <sup>11</sup>
<i>UBLEP</i>	Unity Bluetooth Low Energy Package
<i>UML</i>	Unified Modeling Language
<i>UUID</i>	Universally Unique Identifier
<i>VR</i>	Virtual Reality
<i>VRRP</i>	Virtual Reality Rehabilitation Program

<sup>11</sup> [SOLID](#)

## 12 Bibliografie

- Adams, E. W. (2008). *High Concept Template*. Opgehaald van Skolan För Elektroteknik Och Datavetenskap: <http://www.csc.kth.se/utbildning/kth/kurser/DH2640/grip08/HighConceptTemplate-Inl4.pdf>
- Arpe, K. (2020, December 5). *What makes JNI calls slow?* Opgeroepen op Maart 16, 2021, van Stackoverflow: <https://stackoverflow.com/questions/7699020/what-makes-jni-calls-slow/7809300#7809300>
- Bældung. (2020, October 4). *Guide to JNI (Java Native Interface)*. Opgeroepen op Maart 16, 2021, van Bældung: <https://www.baeldung.com/jni>
- Bhardwaj, K. (2017, Oktober 12). *When Singleton Becomes an Anti-Pattern*. Opgeroepen op March 3, 2021, van DZone: <https://dzone.com/articles/singleton-anti-pattern>
- Bluetooth SIG Proprietary. (2020, December 15). *GATT Specification Supplement 3*. Opgeroepen op Maart 12, 2021, van Bluetooth: <https://www.bluetooth.com/specifications/specs/gatt-specification-supplement-3/>
- Cruysberghs, S. (2002, Juni). *What is UML?* Opgeroepen op Mei 12, 2020, van Scip.be: [http://www.scip.be/PDF/Wat\\_is\\_UML.pdf](http://www.scip.be/PDF/Wat_is_UML.pdf)
- Dunstan, J. (2015, December 14). *JSON Performance Benchmarks*. Opgeroepen op Mei 3, 2021, van JacksonDunstan.com: <https://www.jacksondunstan.com/articles/3294>
- Gehman, C. (2019, January 9). *Git vs. Mercurial: How Are They Different?* Opgeroepen op April 26, 2021, van Perforce: <https://www.perforce.com/blog/vcs/git-vs-mercurial-how-are-they-different>
- K., A. (2019, August 13). *MiBand 3 and React Native*. Opgeroepen op April 1, 2021, van DZone: <https://dzone.com/articles/miband-3-and-react-native-partnbspl>
- Katsandres, J. (2017, Februari 15). *Bluetooth Low Energy - It Starts with Advertising*. Opgeroepen op March 3, 2021, van Bluetooth: <https://www.bluetooth.com/blog/bluetooth-low-energy-it-starts-with-advertising/>
- Kevin Townsend, C. C. (2014, May 12). Chapter 4. GATT (Services and Characteristics). In C. C. Kevin Townsend, *Getting Started with Bluetooth Low Energy* (Vol. IV, p. 180). United States of America: O'Reilly Media. Opgeroepen op March 13, 2021, van O'REILLY: <https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html>
- Marchionne, F. (2019, December 17). *Virtual Reality Rehabilitation - The Future Of Physical Therapy*. Opgeroepen op Mei 19, 2021, van Imotions: <https://imotions.com/blog/virtual-reality-rehabilitation/>
- Mathieu, A. (2019, May 23). *What Is BLE*. Opgeroepen op Maart 12, 2021, van ELA Innovation: <https://elainnovation.com/what-is-ble.html>
- Murray, J. (2014, Februari 20). *Diary of Aunity 3D Newbie - Bluetooth Low Energy Plugins*. Opgeroepen op Februari 4, 2021, van BlackBerry | Developers Blog: <https://devblog.blackberry.com/en/2014/02/diary-of-a-unity-3d-newbie-bluetooth-low-energy-plugins>

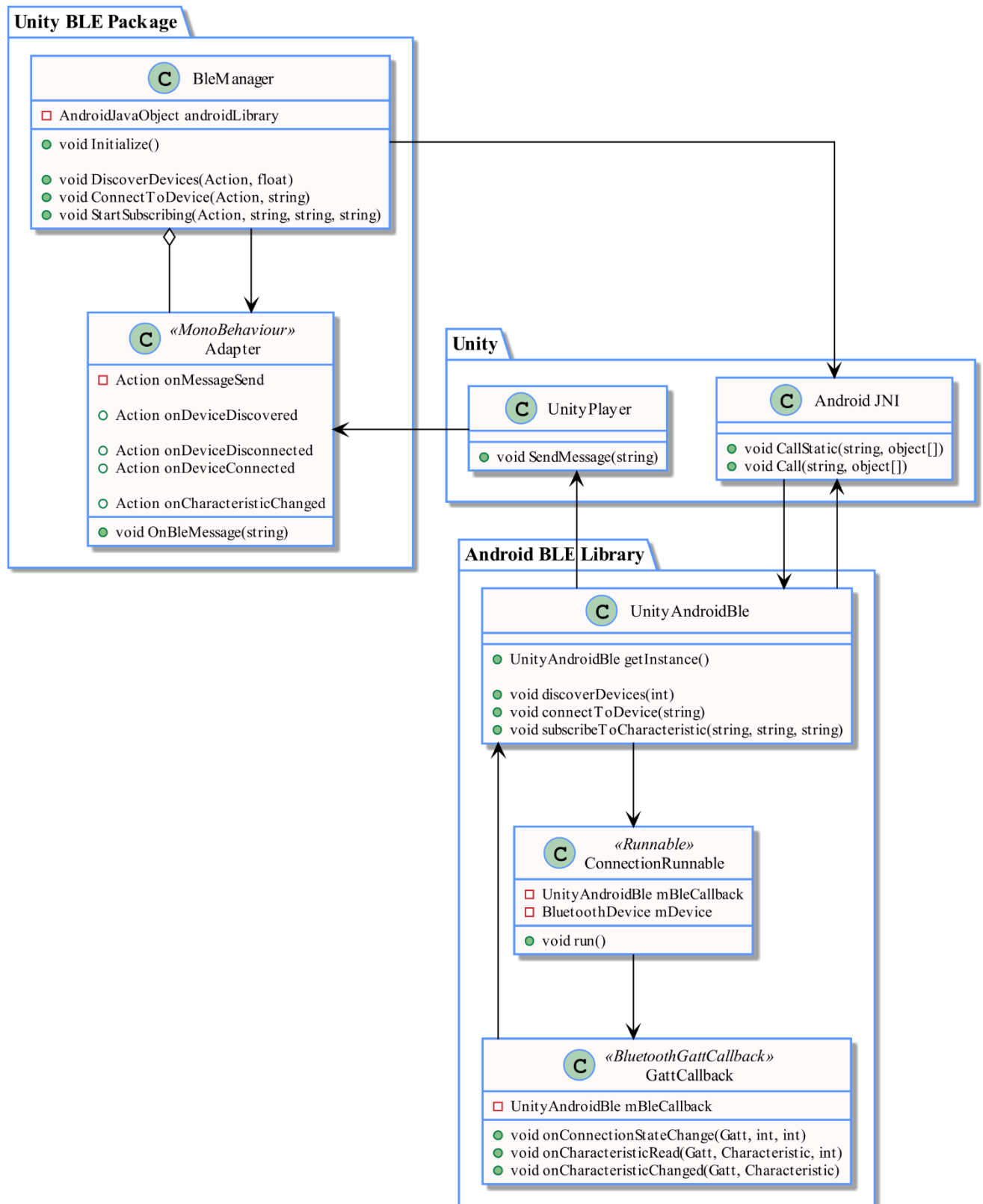
- Ojha, Y. (2018, November 24). *I hacked MiBand 3, and here is how I did it. Part 1*. Opgeroepen op April 1, 2021, van Medium: <https://medium.com/@yogeshojha/i-hacked-xiaomi-miband-3-and-here-is-how-i-did-it-43d68c272391>
- Ong, C. Y. (2019, September 10). *4 Tips to make Android BLE Actually Work*. Opgeroepen op Maart 3, 2021, van Punchthrough: <https://punchthrough.com/android-ble-development-tips/>
- Refactoring Guru. (2020, Januari 1). *Composite*. Opgeroepen op Maart 5, 2021, van Refactoring Guru: <http://gyanendushekhhar.com/2016/09/01/composite-design-pattern-c/>
- Silicon Labs. (2020, December 2). *AN983: Bluetooth® 4.0 Heart Rate Sensor*. Opgeroepen op Maart 12, 2021, van Silicon Labs: <https://www.silabs.com/documents/public/application-notes/AN983-Bluetooth-4.0-Heart-Rate-Sensor.pdf>
- Søderholm, T. (2016, May 24). *6 wireless technologies for wearables*. Opgeroepen op Mei 21, 2021, van Nordic Semiconductor: <https://blog.nordicsemi.com/getconnected/wireless-technologies-for-wearables>
- Townsend, K. (2021, March 12). *Introduction to Bluetooth Low Energy*. Opgeroepen op Maart 12, 2021, van Adafruit: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>
- Vincent, S. (2013, Winter). *Game Design: Design Documents*. Opgeroepen op Mei 28, 2021, van Steve Vincent's Web Site: [https://stevevincent.info/CAP211\\_2012GameDesign4.htm](https://stevevincent.info/CAP211_2012GameDesign4.htm)

## 13 Bijlagen

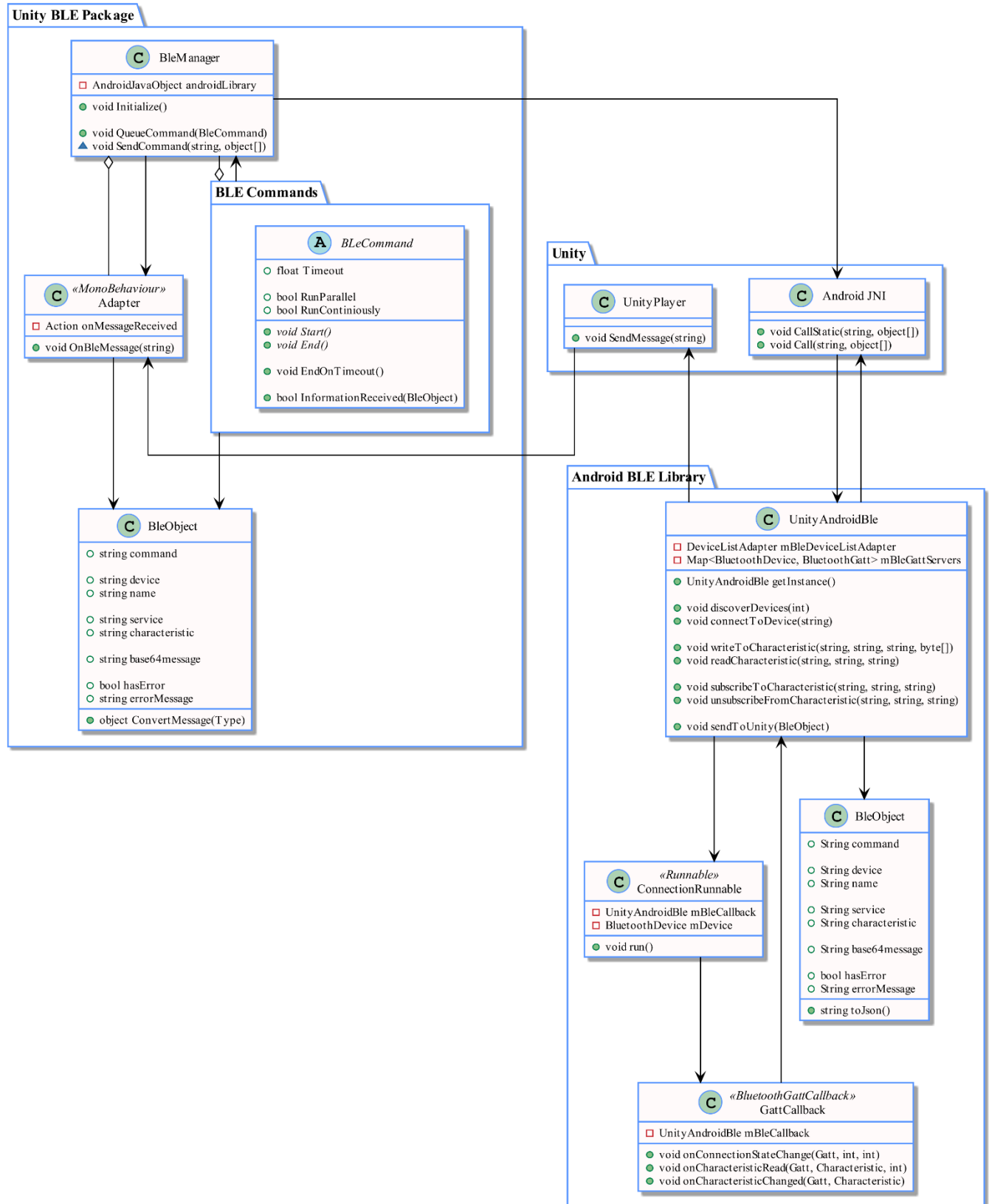
In de bijlagen zijn de UML designs meegenomen. Hierop volgen een aantal documenten, zoals de *Plan van Aanpak* (Bijlage A) en de *High Concept Documents* (Bijlage B).



## 13.1 Unity BLE Package en Android BLE Library UML Design (Fase 2)



## 13.2 Unity BLE Package en Android BLE Library UML Design (Fase 3)



SyncVR Medical

# Plan van Aanpak

Implementeren van wearables binnen Virtual Reality  
Rehabilitation Programs

David H.W. Bruschke  
16-03-2021



## Documenthistorie

<b>Datum</b>	<b>Versie</b>	<b>Beschrijving</b>	<b>Auteurs</b>
09-02-201	<i>0.1</i>	Eerste opzet	D.H.W. Bruschke
11-02-2021	<i>0.2</i>	Opstellen Risico-Matrix	D.H.W. Bruschke
11-02-2021	<i>0.3</i>	Noteren van projectmethodes	D.H.W. Bruschke
12-02-2021	<i>0.4</i>	Opstellen van planningstabel	D.H.W. Bruschke
16-02-2021	<i>1.0</i>	Opmaak	D.H.W. Bruschke

# 1 Inhoudsopgave

<b>INLEIDING .....</b>	<b>61</b>
<b>OPDRACHTOMSCHRIJVING .....</b>	<b>62</b>
AANLEIDING .....	62
PROBLEEMSTELLING .....	62
DOELSTELLING .....	62
ACTIVITEITEN .....	62
<b>DELIVERABLES .....</b>	<b>63</b>
<b>PROJECTORGANISATIE .....</b>	<b>64</b>
CONTACTGEGEVENS .....	64
<b>RISICO'S .....</b>	<b>65</b>
<b>PROJECTMEHODE .....</b>	<b>67</b>
VERSIEBEHEER .....	67
<b>PLANNING .....</b>	<b>68</b>
DEADLINES .....	69

## 2 Inleiding

Dit is een Plan van Aanpak voor het afstudeertraject van David Bruschke bij het bedrijf SyncVR Medical. In dit document wordt beschreven: het probleemdomain, de aanleiding, probleemstelling, deliverables en planning voor deze periode van veertien weken.

Dit afstudeertraject en project komt voort uit de vraag om wearables te combineren met Virtual Reality Rehabilitation Programs voor de fitness georiënteerde afdeling onder SyncVR Medical, genaamd SyncVR Fit. Binnen SyncVR Fit is al vaker gekeken naar het combineren van wearables en Virtual Reality mogelijkheden, maar deze wearables hebben al als doel om Virtual Reality te ondersteunen.

De focus van het afstudeertraject is om wearables, zoals hartslagmeters en stappentellers, te laten communiceren via Bluetooth met een Unity3D fitness georiënteerde toepassing dat Virtual Reality Rehabilitation Programs ondersteunt.

## 3 Opdrachtomschrijving

Binnen dit hoofdstuk komen de aanleiding, probleemstelling en doelstelling van het afstudeertraject aan bod. Hierbij wordt de focus gelegd om de opdracht uit te lichten en het doel van het afstudeertraject op te zetten.

### 3.1 Aanleiding

SyncVR Medical houdt zich bezig met het ontwikkelen van Virtual Reality gerichte applicaties voor medische doeleinden. Verscheidene afdelingen binnen SyncVR Medical houden zich bezig met verschillende vormen van Virtual Reality Rehabilitation Programs toe te passen binnen hun applicaties. Bij SyncVR Fit wordt de focus gelegd om spelenderwijs beweging te stimuleren bij patiënten.

Voor het ondersteunen- en toevoegen van nieuwe gameplay-elementen voor applicaties binnen SyncVR Fit wordt er gezocht naar een methode om fitness wearables te laten communiceren met de applicaties. Hierbij kunnen de wearables informatie doorgeven aan de applicatie en kunnen deze waarden gebruikt worden voor verschillende doeleinden. Onder deze doeleinden kan vallen dat er informatie geüpload kan worden naar het dashboard van SyncVR Medical, het gebruiken van de waarden van wearables om objecten te manipuleren binnen Unity3D en andere doeleinden die tijdens het afstudeertraject aan bod komen.

### 3.2 Probleemstelling

Voor aankomende projecten wil de afdeling SyncVR Fit dat er gekeken wordt naar de mogelijkheden om wearables te combineren met bewegingsoefeningen die worden uitgevoerd in Virtual Reality. Echter binnen SyncVR Medical is er weinig bekend over hoe wearables toegepast kunnen worden binnen deze applicaties en hoe wearables kunnen communiceren met Virtual Reality HMD (Head Mounted Display) apparaten.

### 3.3 Doelstelling

Het doel van het afstudeertraject is om een effectieve communicatie op te zetten tussen een wearable en een HMD. Hierbij wordt er een oefening opgezet als Proof of Concept waarbij een wearable wordt gebruikt om gameplay-elementen toe te voegen binnen een SyncVR Fit applicatie.

### 3.4 Activiteiten



Het uitzoeken van de werking van BLE binnen wearables



Binnen Unity data uitlezen van een wearable



Concept optellen voor SyncVR Fit applicatie



SyncVR Fit applicatie concept realiseren



## 4 Deliverables

Aan het eind van het afstudeertraject zullen de volgende producten worden opgeleverd:

- Plan van aanpak.
- Document dat verschillende concepten bevat, opgesteld aan de hand van de High Concept Document template<sup>12</sup>.
- Unity package voor communicatie tussen een wearable en een Unity applicatie.
- SyncVR Fit applicatie met ondersteuning voor een wearable (Proof of Concept).

Deze producten zullen worden opgeleverd in het formaat dat het best past voor het product zelf. Voor de Plan van Aanpak geldt hiervoor dat deze wordt verwerkt binnen het afstudeerdossier in de vorm van een Word document. De SyncVR Fit applicatie zal worden opgeleverd op twee verschillende manieren:

- 1) Binnen de SyncVR GitHub repository in de vorm van on-gecompileerde code.
- 2) Als gecompileerde APK die compatibel is met de geleverde Oculus Quest en andere HMD's, mede deze als Operating System Android hebben en Bluetooth Low Energy ondersteunen.

---

<sup>12</sup> [High Concept Document Template](#)

## 5 Projectorganisatie

De projectorganisatie bestaat uit drie delen: uitgangssituatie (al beschreven: de opdracht), groepsmanagement en administratie. Deze zijn verder onderverdeeld in taakverdeling en contactgegevens. Dit hoofdstuk beschrijft hoe daarvan invulling wordt gegeven tijdens het afstudeertraject.

### 5.1 Contactgegevens en rolverdeling

Tabel 8 bevat alle contactgegevens van de relevante personen die betrokken zijn bij het afstudeerproject en de rolverdeling onderling deze betrokkenen. Contact zal voornamelijk via Google Meet en Rocket Chat plaatsvinden. Zodra er een mogelijkheid ontstaat buiten de lockdown door het Covid-19 virus, dan zal hierbij ook mondeling contact plaatsvinden.

Tabel 8: Personen en betrokkenen binnen het afstudeertraject

Naam	Rol	Faculteit	Contactgegevens
D.H.W. Bruschke	<i>Developer</i>	Game Development & Simulation	+31 6 831 25 985 17023785@student.hhs.nl
Robbert Brouwer	<i>Opdrachtgever &amp; Product Owner</i>	-	+31 6 536 73 653 robbert@syncvr.tech
V.E. Broeren	<i>Begeleider</i>	Docent Haagse Hogeschool ter Zoetermeer	v.e.broeren@hhs.nl
D.R. Stikkolorum	<i>Expert-Begeleider</i>	Docent Haagse Hogeschool ter Zoetermeer	d.r.stikkolorum@hhs.nl

## 6 Risico's en maatregelen

De risico's zijn opgesteld aan de hand van de *what-if* methodiek. Hierbij wordt uitgegaan van een risico die zich voor kan doen tijdens het afstudeertraject dat impact kan hebben op de snelheid, opdracht of omvang van de werkzaamheden aan het afstudeer opdracht. Hierbij is er een schaalverdeling toegepast genaamd de kans x impact methode. Deze schaalverdeling is onderverdeeld tussen 3 onderdelen: de kans, impact en het totaal. Kans en impact lopen van een schaal tussen 0 t/m 10 (0 is hierbij het laagste- en 10 het hoogste niveau) en totaal is de kans keer de impact. Deze risico's zijn opgesteld in een risico-matrix, te vinden in *Tabel 9*.

Bij het opstellen van het risico-matrix is er rekening gehouden met het voor-nu aanwezige Covid-19 virus. Risico's 4 en 6 zijn hier voorbeelden van. Bij deze risico's is er rekening gehouden dat de kans van het risico hoger ligt en hierbij ook negatieve gevolgen heeft op de impact.

*Tabel 9: Risico's binnen het afstudeertraject*

<b>Risico (what-if)</b>	<b>Impactbeschrijving</b>	<b>Maatregel</b>
1. Wat als de Virtual Reality device niet wordt geleverd?	Hierdoor kan er geen applicatie ontwikkeld worden die op Virtual Reality berust.  <i>Kans: 1 Impact: 8 Totaal: 8</i>	<ul style="list-style-type: none"> <li>• Wachten tot deze geleverd wordt.</li> <li>• Alternatieven vinden om Virtual Reality device te simuleren.</li> </ul>
2. Wat als er geen manier is om wearables te combineren met Virtual Reality devices?	Hierdoor kan er geen applicatie ontwikkeld worden die op wearables berust.  <i>Kans: 3 Impact: 8 Totaal: 21</i>	<ul style="list-style-type: none"> <li>• Proof of Concept wordt door ontwikkeld zonder wearable.</li> <li>• Doel van wearables simuleren binnen de applicatie.</li> </ul>
3. Wat als de Virtual Reality device defect is?	Hierdoor halteert de voortgang aan het ontwikkelen van het Proof of Concept.  <i>Kans: 1 Impact: 4 Totaal: 4</i>	<ul style="list-style-type: none"> <li>• Wachten tot de opdrachtgever een vervanging kan vinden.</li> </ul>
4. Wat als ik langdurig ziek ben?	Vertraging in het project.  <i>Kans: 3 Impact: 4 Totaal: 12</i>	<ul style="list-style-type: none"> <li>• Concept bijsnijden en requirements bijstellen.</li> </ul>
5. Wat als blijkt dat tijdens development het gekozen concept voor het Proof of	Het Proof of Concept kan niet goed de integratie van wearables reflecteren.  <i>Kans: 2</i>	<ul style="list-style-type: none"> <li>• Overleggen met opdrachtgever of het Proof of Concept verder ontwikkeld wordt, of dat er een ander concept</li> </ul>

<i>Concept niet goed de opdracht kan reflecteren?</i>	<i>Impact: 8 Totaal: 16</i>	wordt uitgekozen om te ontwikkelen.
6. Wat als ik niet gemotiveerd blijf tijdens het afstudeertraject?	Vertraging in het project en communicatie.  <i>Kans: 3 Impact: 5 Totaal: 15</i>	<ul style="list-style-type: none"> <li>• Door een standaard 9:00-17:00 planning te volgen blijf ik in een standaard patroon.</li> <li>• Door vaak te bellen met medestudenten wordt er motivatie opgedaan.</li> </ul>

## 7 Projectmethode

Binnen dit hoofdstuk wordt de te-gebruiken projectmethodes beschreven. SyncVR Medical gebruikt voornamelijk Kanban als softwareontwikkelingsmethode en neemt hierbij een Agile-geïnspireerd softwareontwikkelingsproces. Hierbij wordt er in verschillende fases gewerkt om nieuwe features toe te voegen, feedback te verwerken en om nieuwe versies te distribueren. Het Kanban-board is opgedeeld in drie kolommen: *To-Do*, *Doing* en *Done*. De features in de *To-Do* kolom worden geprioriteerd, waarbij de belangrijkste feature aan de bovenkant van de kolom wordt gezet.

Wanneer er een bug-report binnenkomt over de applicatie wordt deze direct afgehandeld of zo snel mogelijk aangekaart. Bug-reports worden genoteerd op een Kanban board met een hogere prioriteit dan normale features zodat deze sneller opgepakt- en afgehandeld wordt.

Nieuwe features volgen een standaard template voor een feature card. Hierbij gaat het om een *titel*, *achtergrond* van de feature, *beschrijving* en eventueel een *definition of done* en *tech spec*. Deze zijn terug te vinden in *Tabel 10*.

Tabel 10: Feature Cards template van SyncVR Medical

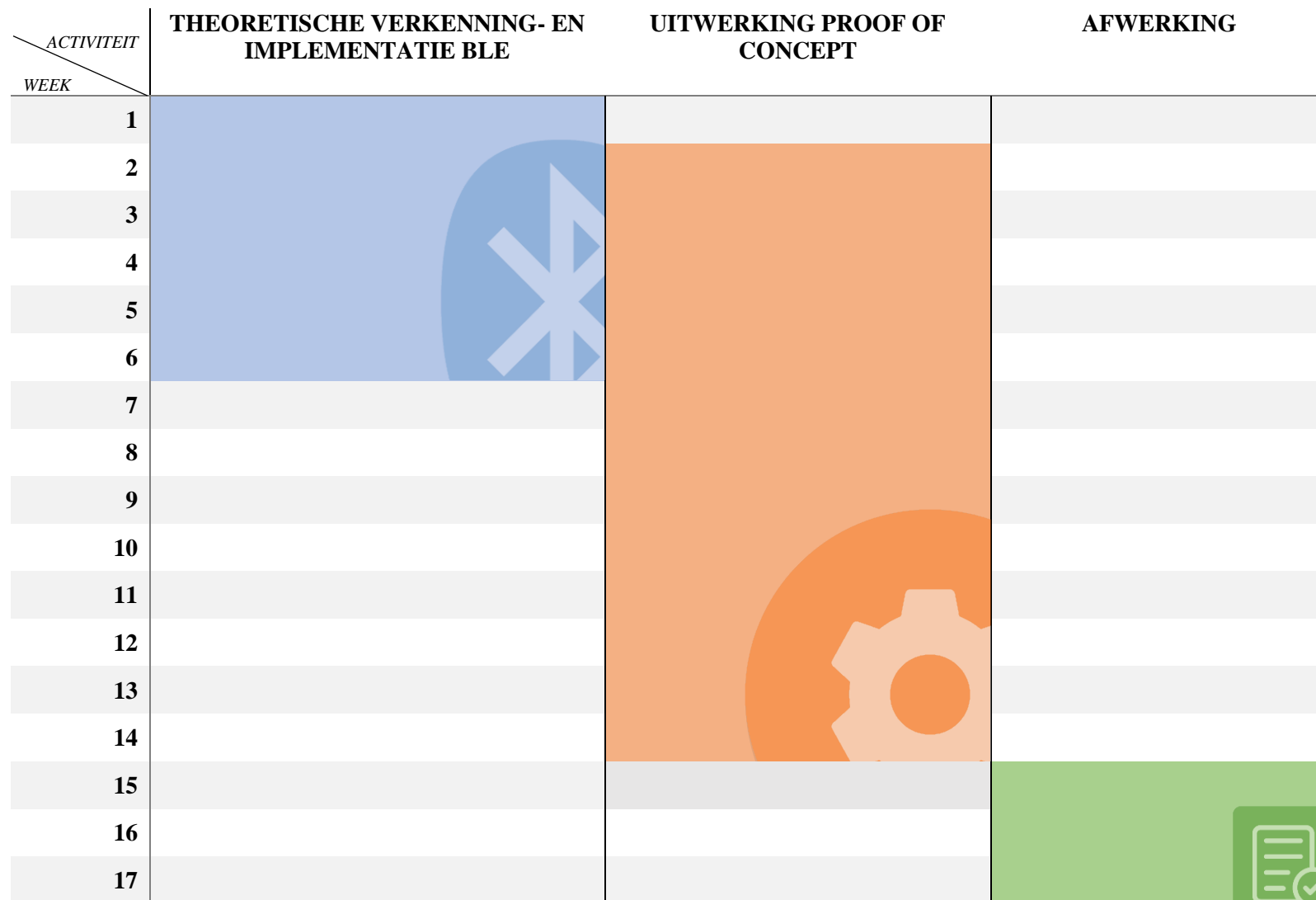
Title	Descriptive title of this feature.	Mandatory
Background	Reason why this feature is desired (e.g. concrete customer feedback).	Mandatory
Description	More in-depth description of the desired feature.	Mandatory
Tech Spec	A technical specification of this feature, if the feature is complex enough to warrant this.	Optional
Definition of Done	A detailed description of when this feature is considered <i>done</i> , if this feature is complex enough to warrant this and needs explicit scoping.	Optional

### 7.1 Versiebeheer

Versiebeheer wordt gebruikt om verschillende fases bij te houden van de projecten die worden uitgevoerd tijdens het afstudeertraject. Hierbij gaat het om het bijhouden van verschillende versies van de code die wordt geschreven voor het implementeren van Bluetooth Low Energy en de uitwerking van het Proof of Concept. SyncVR Medical heeft hun versiebeheer opgezet binnen GitHub; elke afdeling heeft zijn eigen repository waarin de medewerkers van die afdeling nieuwe branches aan kunnen maken. Elke medewerker heeft zijn eigen branch die uiteindelijk gemerged worden naar de ‘develop’ branch.

Er zal een nieuwe branch aangemaakt worden waarin de code van het Proof of Concept van dit afstudeerproject ge-pushed wordt. Lokaal zal er met git met feature branches gewerkt worden die worden gemerged met de nieuwe branch voor het Proof of Concept, maar zullen niet ge-pushed worden naar de repository van SyncVR Fit. Dit voorkomt dat er te veel nieuwe branches worden aangemaakt binnen de online repository.

## 8 Planning



Figuur 21: Planning over de tijdsperiode van het afstudeer traject

## 8.1 Deadlines

In *Figuur 21* staat een tabel waarin in grote lijnen de planning over het afstudeer traject. Binnen de eerste zes weken van het traject wordt er een theoretische verkenning gedaan over hoe Bluetooth Low Energy ge-implementeert kan worden binnen Unity i.c.m. de VR device Oculus Quest. Tijdens deze theoretische verkenning wordt het Proof of Concept uitgewerkt. Hierbij behoort de voorbereidingsfase, waaronder het opstellen van verschillende concepten aan de hand van het High Concept Document template (zie [Deliverables](#)). Tijdens het uitwerken van het Proof of Concept zullen nieuwe features opgesteld worden aan de hand van feedback, waaronder feedback van de opdrachtgever, medewerkers binnen SyncVR Medical en eventueel patiënten als hier een mogelijkheid tot valt. Bij het afwerken wordt vooral gefocust op de staat van overdracht van de deliverables. Hierbij wordt tijd besteed aan het verder uitwerken van het afstudeerdossier, het packagen van eventuele Unity Packages en worden de documenten verzameld en overgedragen aan SyncVR Medical.

In *Tabel 11* staat een tabel met academische deadlines. Deze zijn bedoeld voor de student, maar kunnen ook van invloed zijn op de opdrachtgever binnen SyncVR Medical.

*Tabel 11: Overzicht tabel voor academische deadlines (Hague University of Applied Sciences)*

Evenement	Datum
Bedrijfsbezoek	(01-12)/03/2021
Bespreken concept afstudeerdossier	(12-16)/04/2021
Tussentijds assessment	Voor 07/05/2021
Inleveren afstudeerdossier	04/05/2021
Zittingen	21/05/2021 – 07/06/2021



SyncVR Medical

# High Concept Documents

Document containing concepts that are ready for a proposal to SyncVR Fit.  
These concepts are based on a type of wearable.

David H.W. Bruschke  
09-02-2021



# 1 Index

<b>DRUM TO YOUR HEART'S CONTENT (<i>HEART DRUMMING</i>) .....</b>	<b>72</b>
FEATURES.....	72
PLAYER MOTIVATION.....	72
GENRE.....	72
LICENSE.....	72
TARGET CUSTOMER .....	73
COMPETITION .....	73
UNIQUE SELLING POINTS.....	73
TARGET HARDWARE .....	73
CONCEPT ART .....	74
<b>SHAPE YOUR BODY .....</b>	<b>75</b>
FEATURES.....	75
PLAYER MOTIVATION.....	75
GENRE.....	75
TARGET CUSTOMER .....	75
COMPETITION .....	76
UNIQUE SELLING POINTS.....	76
TARGET HARDWARE .....	76
INSPIRATIONAL ART .....	76
<b>MARCH YOUR STEP .....</b>	<b>77</b>
FEATURES.....	77
PLAYER MOTIVATION.....	77
GENRE.....	77
TARGET CUSTOMER .....	77
COMPETITION .....	77
UNIQUE SELLING POINTS.....	78
TARGET HARDWARE .....	78

## 2 Drum to your Heart's Content (*Heart Drumming*)

A virtual reality (further in this document regarded as VR) rhythm game based on the heartbeat of the user, where the user must hit the right piece of the drum set to score points. By scaling the beats per minute (further in this document regarded as BPM) and progression of the song based on the heartbeat of the user, the game will automatically progress to later stages with a proper cooldown.

### 2.1 Features

- Challenges a user's sense of rhythm based on a familiar beat that's always with them: their heartbeat.
- By hitting the right parts of the drum set with the Virtual Reality provided drumsticks, the user gets more points and can progress to further stages of the song if the game sense a certain threshold in their heart rate.
- Integrates heart rate like not fitness application has done before. By syncing the song's BPM to the user's heartbeat, the game will provide a unique rhythm experience.

### 2.2 Player Motivation

By stimulating the player visually and motorically (controller vibrations) by providing feedback when hitting drums and showing the progression into the song based on pretty visuals, the users are immersed and motivated to reaching the end of the song (and therefore also the exercise).

### 2.3 Genre

By combining VR and Music, the genre is quite specific towards the bullet points set below. While it isn't particularly a clear genre, the game also falls under Fitness games, also known as Exergames (portmanteau of "exercise" and "game").

- Virtual Reality (VR)
- Action
  - Rhythm Game
- Exergame

### 2.4 License

While it's possible to make the music in-house, there exists an opportunity that creating in-house music will take too much time. In this case samples and music will be used with the [Creative Commons License](#) online. If credit is required by these artists, then they shall be provided within the *Drum to your Heart's Content* credit tab shown at the end of a song.

## 2.5 Target Customer

Patients that require heavy arm exercises or patients looking to get some sweat on. Since the movement is focused on the arms, this exercise can be played both sitting and standing up.

## 2.6 Competition

Other movement-based music VR-based games. Below is a list of music VR-based games that can pose as competition for the game *Drum to your Heart's Content*. Attached to each bullet point is a link to its trailer which features gameplay of the game.

- [Beat Saber](#)  
While *Beat Saber* has the same core gameplay, the focus lies on different parts. In *Beat Saber* it's the user's capability to quickly see in which way to slice the cubes, focusing more on the user's coordination. *Drum to your Heart's Content* focuses more on the user's reflexes based on hitting the different parts of the drum set seated (virtually) in front of them.
- [Audica](#)  
*Audica* is another VR-rhythm based game based on firing guns towards the targets on the song's BPM. While also an intense arm-based VR movement game, the focus lies on the accuracy of hitting the targets and how well the user aims down sight (stretching the arm forward).
- [Beats Fever](#)  
*Beats Fever* is heavily based on *Beat Saber*, where it differs is that it has streaks of targets to hit. While *Beat Saber* only has single hit targets, *Beats Fever* has the user follow streaks of targets to hit.

## 2.7 Unique Selling Points

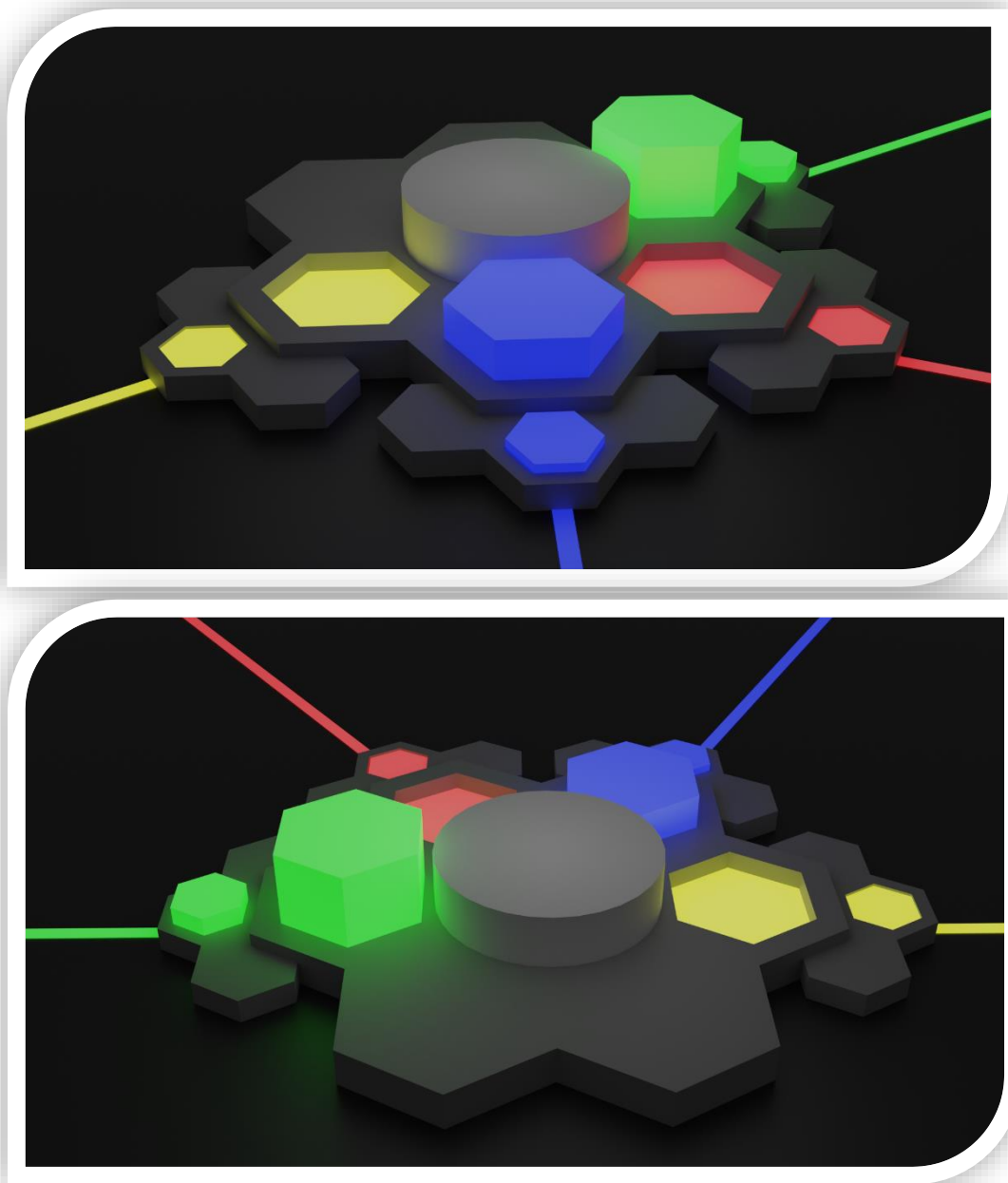
- First VR game that integrates Bluetooth Low Energy (further in this document regarded as BLE) wearables to support gameplay.
- Creates an upper body workout that can be adjusted with different levels of intensity.
- A visually fitting VR game that is based on the user's heartbeat.

## 2.8 Target Hardware

Standalone Head Mounted Display (further in this document regarded as HMD) with controller support. The game requires at least 2 controllers to function properly.

## 2.9 Concept Art

Provided are renders done in Blender to show the general idea of how the game will look.



*Figuur 22: Concept- and example render of Heart Drumming*

## 3 Shape Your Body

Shape your body into the way the shapes in the walls guide you. A VR game where your character stands still while walls with shapes come onto your path. Hold your controllers and head in the right position to go through the shapes or be stopped by the wall and start all over again.

### 3.1 Features

- Challenges a user to quickly get their body in to the right shape by holding the controllers and headset at an indicated pose.
- Features fast gameplay, scattered across different visual zones.
- By syncing the game to the player's heartbeat, the user challenges themselves if their heart rate goes up.

### 3.2 Player Motivation

Reach the end of the exercise. After a set amount of time (either syncs to a song or depending on the difficulty) and correctly shaped body shapes, the player wins the exercise.

### 3.3 Genre

By combining VR and (potentially) Music, the genre is quite specific towards the bullet points set below. While it isn't particularly a clear genre, the game also falls under Fitness games, also known as Exergames (portmanteau of "exercise" and "game").

- Virtual Reality (VR)
- Action
  - Rhythm Game
- Exergame

### 3.4 Target Customer

Patients that require heavy arm- or upper-body exercises or patients looking to get some sweat on. Since the movement is focused on the arms and head, this exercise can be played both sitting and standing up, but since there can be difficult poses to perform while sitting down, standing up is recommended.

## 3.5 Competition

Other pose-based games. Below is a list of pose based games that can pose as competition for the game *Shape Your Body*. Attached to each bullet point is a link to its trailer which features gameplay of the game.

- [Muscle March](#)

Since *Muscle March* is the main inspiration for *Shape Your Body*, it serves as the right competition for the game. Since this game is a WiiWare only game, the game runs on the Nintendo Wii, therefore limits the user into simple poses that can easily be distinguished by the Wii controllers. With VR there is a greater amount of poses possible, combining both the head and the arms.

## 3.6 Unique Selling Points

- First VR game that integrates BLE wearables to support gameplay.
- Creates an intense arm workout with a speed that can be adjusted with different levels of intensity.
- A fun mini game that can still provide quite some exercise.

## 3.7 Target Hardware

Standalone HMD with controller support. The game requires at least 2 controllers to function properly.

## 3.8 Inspirational Art



## 4 March your Step

By marching in a group, you follow the movements of the characters in front of you. By using a pedometer (function that counts your steps inside wearables), you need to march on spot to follow the group. When the character (Non-Playable-Character) at the front does a move, the 2 characters behind them will copy that move with an interval each of 1 second (or beat) until it's the player's turn to copy the move.

### 4.1 Features

- Counts the user's steps that they perform on the spot, giving an extra element to immerse the user into VR by moving their character forward with every step.
- Short-, but heavy exercise keeping the player moving during the whole exercise.
- A cheering audience on the side will judge how well the player performs, motivating the player to continue.

### 4.2 Player Motivation

Reach the end of the exercise while performing poses in front of an audience. After a set amount of time (either synced to a song or depending on the difficulty), the player will end the exercise and get a grade based on their scores.

### 4.3 Genre

By combining VR and (potentially) Music, the genre is quite specific towards the bullet points set below. While it isn't particularly a clear genre, the game also falls under Fitness games, also known as Exergames (portmanteau of "exercise" and "game").

- Virtual Reality (VR)
- Action
  - Rhythm Game
- Exergame

### 4.4 Target Customer

Patients that require heavy arm exercises or patients looking to get some sweat on. Since the movement is focused on the arms and head, this exercise can only be played by standing up.

### 4.5 Competition

Other pose-based games. Below is a list of pose based games that can pose as competition for the game *Shape Your Body*. Attached to each bullet point is a link to its trailer which features gameplay of the game.

- [Muscle March](#)

Since *Muscle March* is the main inspiration for *Shape Your Body*, it serves as the right competition for the game. Since this game is a WiiWare only game, the game runs on the Nintendo Wii,



therefore limits the user into simple poses that can easily be distinguished by the Wii controllers. With VR there is a greater amount of poses possible, combining both the head and the arms.

- [Rhythm Fever](#)

Although not a direct competitor, *Rhythm Fever* focuses a lot on precise time-rhythm based actions. If in *March Your Step* a form of rhythm is implemented in the way the player needs to perform poses, it'll have a slight correlation with *Rhythm Fever*.

## 4.6 Unique Selling Points

- First VR game that integrates BLE wearables to support gameplay
- Creates an intense arm- and leg-workout with a speed that can be adjusted with different levels of intensity.
- A short, but fun mini game that provides plenty of exercise.

## 4.7 Target Hardware

Standalone HMD with controller support. The game requires at least 2 controllers to function properly.

SyncVR Medical

# Afstudeerplan

David H.W. Bruschke  
09-02-2021



Afstudeerblok	2020-2021
Startdatum uitvoering afstudeeropdracht	04-2-2021
Inleverdatum afstudeerdossier volgens jaarrooster	04-6-2021

Studentnummer	17023785
Achternaam	Bruschke
Voorletters	D.H.W.
Roepnaam	David
Adres	
Postcode	
Woonplaats	Hillegom
Telefoonnummer	
Mobiel nummer	
Privé emailadres	

Differentiatie	SE
Afstudeerprogramma	GDS
Domein*	GDS / SE
Locatie	Zoetermeer
Variant	Voltijd

Naam studieloopbaanbegeleider	Sebastiaan Rieter
Naam begeleidend examiner	Vincent Broeren
Naam expert examiner	Dave Stikkolorum

#### Bedrijf

Naam	SyncVR Medical
Afdeling	SyncVR Fit
Bezoekadres	Padualaan 8
Postcode	3584 CH
Plaats	Utrecht
Telefoonnummer	+31 6 127 49 279
URL	<a href="https://syncvr.tech/">https://syncvr.tech/</a>

#### Opdrachtgever

Achternaam	Brouwer
Voorletters / Voornaam	Robbert
Titel / Opleiding	Directeur
Functie	COO
Telefoonnummer (werk)	
Emailadres (werk)	
Is de opdrachtgever ook de bedrijfsmentor?	Ja

# Titel afstudeeropdracht

Hoe kunnen wearables toegepast worden met Virtual Reality Rehabilitation Programs binnen oefeningen of games gemaakt binnen Unity3D?

## Opdrachtomschrijving

### 1 Bedrijf & Probleemdomein

Al jaren ontwikkeld SyncVR Medical (verder te noemen als SyncVR) verschillende toepassingen voor medische doeleinden. Elke toepassing richt zich op een eigen doel. Zo is SyncVR Relax & Distract gericht op het verminderen van acute pijn of angst tijdens medische interventies, SyncVR Fit voor het spelenderwijs beweging stimuleren, en nog vele andere toepassingen<sup>13</sup>.

Het SyncVR Medical platform maakt het mogelijk om VR te implementeren in het gehele ziekenhuis: van hardware, software en workshops tot afspraken met de ondersteunde diensten en een hygiëne-protocol. Met 75+ aangesloten ziekenhuizen, waaronder Zuyderland, Spaarne en het Erasmus MC, en meer dan 3000 patiënten die zijn behandeld met VR, zijn zij het grootste virtual reality platform van de medische zorg in Europa. Doormiddel van applicaties binnen Unity3D, willen ze ervoor zorgen om de kwaliteit van de zorg te ondersteunen.

De toepassingen van SyncVR Medical zijn niet alleen gebaseerd op inzichten van patiënten en zorg-professionals, maar ook op wetenschappelijk onderzoek: internationale- en door SyncVR Medical/SyncVR Medical partners geïntanceerde onderzoeken.<sup>14</sup>

---

<sup>13</sup> [Homepagina](#)

<sup>14</sup> [Research](#)

## 2 Aanleiding & probleemstelling

Het afstudeertraject is een uitbreiding op projecten die gaande zijn op een afdeling binnen SyncVR Medical, genaamd **SyncVR Fit**. Op deze afdeling worden applicaties ontwikkeld om spelenderwijs beweging te stimuleren. Hierbij zijn kern-toepassingen kinderfysiotherapie, geriatrische revalidatie en post-IC revalidatie. Hierbij zijn ingerichte games en oefeningen gemaakt binnen Unity3D in combinatie met VR. Gebruikers kunnen deze oefeningen en games uitvoeren met behulp van de Oculus Quest Quest of HTC Vive Focus Plus, dit zijn draadloze VR-headsets en controllers die werken in 6dof (6 Degrees of Freedom<sup>15</sup>). Het probleem is dat er weinig onderzoek data bestaat over Virtual Reality Rehabilitation Programs (verder te noemen als 'VRRP'), laat staat het combineren van wearables om het gebied van VRRP te bevorderen. Een oplossing hiervoor zou zijn om onderzoek te doen naar het toepassen van wearables, zoals hartslagmeters, op het gebied van VRRP en hiervan een koppeling te maken naar Unity3D.

Wearables kunnen data doorgeven aan de toepassing dat een standaard Oculus Quest of HTC Vive Focus Plus niet direct kan aanbieden. Hierbij kan gedacht worden aan hartslagmeters, stappentellers, Apple Watches, etc. Deze wearables kunnen informatie doorgeven aan de toepassing, waar de toepassing dan op kan reageren, bijvoorbeeld door de oefening langzamer te laten verlopen als de hartslag van de gebruiker omhoog gaat, of juist hierop inspelen en de moeilijkheid van de oefening schalen met de hartslag.

Het doel van het afstudeertraject is om onderzoek te doen naar welke wearables relevant kunnen zijn voor VRRP en hoe deze gebruikt kunnen worden in combinatie met toepassingen binnen **SyncVR Fit**. Deze toepassingen worden gemaakt binnen Unity3D, maar kunnen ook gebruikt worden voor monitoring binnen hun dashboard.

## 3 Doelstelling

Met het eindproduct van deze afstudeerstage heeft SyncVR Medical een package voor Unity waarmee HMD's die gebruikt worden binnen SyncVR Medical kunnen communiceren met een wearable dat gebruik maakt van Bluetooth Low Energy.

Tijdens de laatste fase van het afstudeertraject, wordt er een Proof of Concept (hierna te noemen als 'PoC') gebouwd aan de hand van de vindingen van het onderzoek. Dit PoC zal elementen bevatten binnen Unity3D die reageren aan de hand van de data dat uit een wearable gelezen kan worden. Dit kan gebouwd worden aan de hand van een simulatie, SyncVR-fit game toepassing of een adapter om de data uit een wearable relevant te maken voor Unity3D als game engine.

## 4 Concrete werkzaamheden en producten

Om het eindproduct te kunnen bereiken, is het traject opgedeeld in 2 fases. Hierbij gaat het om:

### *Fase 1: Ontwikkeling en Toepassen*

Tijdens deze fase wordt gekeken naar het ontwikkelen van een Bluetooth Low Energy package om met een HMD te communiceren met een wearable. Hierbij wordt er gezorgd dat er waardes vanuit de wearable uitgelezen kunnen worden met behulp van de package. Bij deze fase komt het uitwerken van de BLE package, maar ook het ontdekken en ontwikkelen met een (voor mij) nieuwe technieken. Hierbij komt persoonlijke- en technische ontwikkeling bij kijken.

---

<sup>15</sup> [Six DOF](#)

Ook wordt er tijdens deze fase de basis-benodigheden opgezet om aan de slag te gaan binnen SyncVR Medical, zoals Git, eventuele documenten of contracten ondertekenen, kennismaking en het opstellen en presenteren van een plan van aanpak.

### *Fase 2: Toepassen en Creëren (ET: 12 weken)*

Op basis van de ontwikkelde package die is opgesteld tijdens **Fase 1**, wordt er een PoC ontwikkeld. Hierbij staan resultaten centraal.

Binnen deze PoC zal er gebruikgemaakt worden van een onderzochte wearable. Deze wearable zal gebruikt worden voor een VRRP applicatie die zich houdt aan de SyncVR Fit richtlijnen. De bedoeling is dat de gemaakte PoC als eventuele voorbeeld kan dienen voor toekomstige projecten binnen SyncVR Fit.

### *Technieken*

Tijdens deze fases worden een aantal technieken, softwareontwikkelmethodes en programma's gebruikt om de eindresultaten te behalen. Vooraf kunnen een aantal hiervan opgesteld worden, maar tijdens het project kunnen hier nieuwe aan worden toegevoegd. Deze worden bijgehouden en beschreven in het uiteindelijke stage (overdracht) document.

### *Programma's*

**Unity3D.** Applicaties en toepassingen binnen **SyncVR** worden geschreven in de Unity3D Editor. In deze Editor wordt er geschreven met de programmeertaal C# (.NET 4.\*). Omdat er verschillende Unity3D versies zijn, wordt er tijdens het afstudeertraject gekozen voor de meest recente **LTS** versie.

**Microsoft Visual Studio.** Visual Studio is één van de meest gebruikte Integrated Development Environment (te noemen als IDE) voor het ontwikkelen met Microsoft gerelateerde ontwikkel methodes (C#, F#, TypeScript, UWP, Winform, etc.). Het is een krachtige editor dat ook directe ondersteuning biedt voor het schrijven van code binnen Unity3D en het debugging van de gemaakte code binnen de Editor.

**Git.** Voor versiebeheer wordt er gebruikgemaakt van Git in combinatie met de services van **BitBucket**, mede omdat BitBucket integratie biedt met **Jira**. Hierbij kunnen taken gemaakt in Jira automatisch verbonden worden aan nieuwe branches binnen BitBucket, en kunnen de taken gesloten, in progressie of een andere status krijgen aan de hand van de status van de gerelateerde Git branch.

### *Softwareontwikkelmethodes*

**Kanban** in combinatie met **Jira**. Er zal niet direct gewerkt worden binnen het huidige **SyncVR Fit SCRUM** team, maar er wordt wel direct mee gecommuniceerd. Hierdoor is er een sterk softwareontwikkelingsmethode nodig dat ervoor zorgt dat er mogelijk is om nieuwe taken toe te voegen en categoriseren om een goed overzicht te houden binnen, maar ook buiten het project. Dit wordt gedaan met de **Kanban** softwareontwikkeling methode. Met deze methode gaan taken door een proces voordat het als klaar wordt gezien. Hierbij wordt er een online board gebruikt om deze taken en het proces bij te houden, namelijk **Jira**. Ook kunnen nieuwe taken toegevoegd worden aan de hand van de feedback van het SyncVR personeel en de bedrijfsmentor. Hierdoor wordt er dus ook **Agile** gewerkt.

## 5 Beroepstaken

Gedurende het stagetraject worden er een aantal beroepstaken van ICT/SE/GDS behandeld. De beroepstaken waar direct aan gewerkt wordt staan hieronder beschreven.

### A1 Analyseren probleemdomein & opstellen probleemstelling

Bij het maken van het stageplan wordt er een probleemdomein opgesteld om aan te werken tijdens het stagetraject. Deze is meegenomen in [Aanleiding & Probleemstelling](#).

Tijdens **Fase 1** (*Onderzoek*, beschreven in **4. Concrete werkzaamheden en producten**) wordt er een analyse gedaan naar de limitaties en mogelijkheden van wearables binnen VRRP. Hierbij wordt er een probleemdomein opgesteld en hoe dit probleem behandeld kan worden.

### C9 Ontwerpen game, computermodel en simulatie

Om de resultaten uit **Fase 1** te ondersteunen, wordt er een game ontwikkeld die gebruikt kan worden op de SyncVR Fit afdeling die gebruik maakt van een wearable. Hierbij wordt er een game ontworpen en geïmplementeerd dat zich bezighoudt met het laten bewegen van de eindgebruiker met VRRP. Hierbij worden de controllers van de VR-headsets gebruikt om virtuele objecten aan te sturen en zo interactie te laten vormen met de game.

### D1 Realiseren van software

In **Fase 2** (*Toepassen en Creëren*) worden er ontwikkeld met de toepassing die is ontwikkeld binnen **Fase 1**. Hier komt code bij kijken voor het verbinden van wearables aan een XR device met Unity3D. Deze implementaties kunnen worden gebruikt als voorbeeld voor toekomstige applicaties, waarbij een wearable wordt gecombineerd met toepassingen voor VRRP. Door het realiseren van **Fase 2** en daarmee de resultaten uit **Fase 1** ondersteund, wil ik aantonen dat ik beroepstaak D1 heb behaald.

### Ga Effectief (internationaal) communiceren

Bij Fase 1 (*Ontwikkeling en Toepassen binnen BLE*) is er informatie nodig om te weten wat de eventuele requirements zijn voor het onderzoek. Door Fase 1 en 2 af te ronden en een onderzoeksrapport op te stellen wil ik laten zien dat ik effectief gecommuniceerd heb.

Ook wordt er tijdens het stagetraject veel gecommuniceerd met de stagementor binnen de afdeling SyncVR Fit, met behulp van wekelijkse standups.

### Gc Kritisch, onderzoekend en methodisch werken

Tijdens **Fase 1** (*Ontwikkeling en Toepassen binnen BLE*) wordt er ontwikkeld met een nieuwe techniek voor een package dat binnen SyncVR Medical gebruikt gaat worden. Hiervoor moet deze techniek correct worden toegepast aan de hand van betrouwbare technieken en methodes. Deze moeten getoetst en kritisch bekeken worden voordat deze worden geïmplementeerd binnen het eindproduct. Hierbij gaat het om betrouwbare bronnen, bekende design patterns en artikelen die relevant zijn voor de technieken.

### Gf Leren leren: voorbereiden op volgende studiefase en beroep

Een afstudeerstage bij **SyncVR** bereid me voor op het werken in een beroep binnen het **Game Development & Simulation** werkveld van ICT. Ook zal het toepassen van een onderzoek en volgen van de regels die verbonden zijn aan programmeren vanuit het HHS richtlijnen geven aan het volgen van een beroep binnen dit werkveld.

SyncVR Medical

# Evaluatie - Bedrijfsmentor

David H.W. Bruschke  
4-6-2021





### **Evaluatieformulier afstuderen**

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: David Bruschke

Periode: 2021 januari - juni

Bedrijf c.q. instelling: SyncVR Medical

Bedrijfsmentor: Robbert Brouwer

Plaats: Utrecht

Datum: 04-06-2021

**1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?**

David heeft zich vanaf moment één opgesteld als een zeer professionele en fijne afstudeerder en collega. Altijd bereikbaar, betrokken en open voor feedback.

**2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?**

Zeer goed. David heeft hierin een hele mooie leercurve doorgemaakt. In het begin was er wat onduidelijkheid/gebrek aan communicatie over de richting van de afstudeeropdracht, deze feedback heeft David heel goed en open opgepakt en daarvan geleerd. In het vervolg heeft David dit enorm goed toegepast, dit bewijst zijn lerend vermogen en het toepassen van feedback. Ook heeft David tijdens de wekelijkse meetings en twee-wekelijkse demo's altijd goede en heldere presentaties/updates gegeven.

### 3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

- Qua verantwoordelijkheid **goed** / voldoende / matig / onvoldoende
- Qua zelfstandigheid **goed** / voldoende / matig / onvoldoende
- Qua planmatig werken **goed** / voldoende / matig / onvoldoende
- Qua creativiteit **goed** / voldoende / matig / onvoldoende
- Qua productiviteit **goed** / voldoende / matig / onvoldoende
- Qua samenwerken met collega's **goed** / voldoende / matig / onvoldoende
- Qua draagvlakontwikkeling **goed** / voldoende / matig / onvoldoende
- Qua inspelen op bedrijfscultuur **goed** / voldoende / matig / onvoldoende
- Qua rekening houden met de  
specifieke context van het bedrijf **goed** / voldoende / matig / onvoldoende
- Qua het op gang brengen van de  
nodige veranderingen goed / **voldoende** / matig / onvoldoende

### 4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

Zeer goed, zelfs zo goed dat David dit zelf nog niet altijd realiseert. Onze CTO heeft een keer met David op kantoor gewerkt en was echt enorm onder de indruk van het geleverde werk tot dan toe. Een verbeterpunt is dat David zelf nog meer moet realiseren hoe goed hij is ;)

### 5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

Zeer goed, David heeft een kwalitatief zeer goede opdracht opgeleverd.

### 6. Bent u tevreden over het opgeleverde (eind)product?

100%

- In hoeverre heeft u gekregen wat is afgesproken?
  - i. Volledig
- In hoeverre voldoet het (eind)product aan uw verwachtingen?
  - i. Volledig

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**
  - i. Zeer goed
- **Wat gebeurt er met het opgeleverde (eind)product?**
  - i. Wordt geïmplementeerd in ons product SyncVR Fit
- **Kunt u direct met het opgeleverde product aan de slag?**
  - i. Ja

7. **Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?**

Nee