

# Geïntegreerde Proef

**Evi Boelen**

**6 ITN**

**IT & Netwerken**

**Schooljaar 2018-2019**





# Voorwoord

Het samenstellen van dit eindwerk was geen makkelijke taak. Meermaals zag ik door de bomen het bos niet meer. Andere ogenblikken waren mijn gedachten vol goede ideeën die ik kon gebruiken om mijn GIP tot een goed einde te brengen.

Maar natuurlijk had ik dit allemaal niet kunnen verwezenlijken zonder de professionele hulp van de heer Frank van Geirt, mijn stagebegeleider. Daarnaast wil ik alle leerkrachten bedanken voor hun medewerking en begeleiding bij het maken van de GIP.

Ik wil ook mijn ouders bedanken die mij in dit laatste jaar nog harder hebben gesteund dan ooit tevoren.

Dankzij Windows 10, de bijhorende Microsoft Office 2016 software (Word, Excel en Microsoft Powerpoint), Unity, Visual studio, Oculus en natuurlijk mijn HP DeskJet 360 (printer) kon ik deze proef realiseren.

Om een zo professioneel mogelijk eindwerk te verkrijgen werden de NBN-normen strikt opgevolgd en toegepast.

Het werken aan de geïntegreerde proef vond ik boeiend en interessant. Dit werk heeft mij veel geleerd over het aanpakken van een groter project. Deze ervaring zal me zeker en vast bijblijven.



## Inleiding

Mijn geïntegreerde proef bestaat uit een opdracht die ik op mijn stageplaats voorgeschoteld kreeg. De opdracht bestaat uit het ontwikkelen van een Virtual Reality game.

Het doel van de geïntegreerde proef was voor mij vooral om te kijken of een job in de informatica wel weg gelegd was voor mij. Dit kon ik ervaren aan de hand van de stage die gepaard ging met de geïntegreerde proef. Mijn stageplaats was in een game bedrijf. Dit was voor mij enorm voordelig. Na het hopelijke behalen van een bachelor diploma in de toegepaste informatica zou ik graag nog voor een master diploma in game development bij studeren. Mijn stageplaats kon me al een eerste blik geven op de job die ik later graag zou gaan uitoefenen.

Tijdens de eerste dag van mijn stage werd er beslist wat voor game we effectief zouden gaan maken. Dit hebben we gedaan tijdens een kleine vergadering. Tijdens de vergadering werd mijn idee verkozen tot het idee dat uitgewerkt zou gaan worden. Dit was extra motiverend voor mij aangezien ik het idee ook grotendeels zelf zou uit werken.

In de loop van het schooljaar hebben we enkele opdrachten gekregen. Deze opdrachten gingen over onze stage. Eén van de opdrachten was een gedetailleerd verslag over wat we allemaal precies hebben gedaan op onze stage. Maar dat is niet de enige opdracht die we kregen. Voor de vakken Engels, Nederlands, Frans, Hardware en Bedrijfsbeheer kregen we ook nog enkele opdrachten. Deze hadden allemaal te maken met verschillende aspecten van het bedrijf waar we de stage bij deden, en ons project.

Al deze opdrachten heb ik samengebundeld tot het GIP-werk dat u nu leest. Heel de bundel is opgesteld volgens de BIN-normen.

Deze geïntegreerde proef heeft me veel geleerd. Ik heb geleerd dat deadlines enorm belangrijk zijn en dat ik mij altijd goed voor moet bereiden. Ik heb ook geleerd dat dingen plannen heel belangrijk is. Als je dingen vergeet in te plannen vergeet je ze uiteindelijk ook te maken en in te dienen. Vervolgens leerde ik ook dat ik enorm graag in een bedrijf zoals mijn stagebedrijf zou willen werken. Een kleiner bedrijf is veel leuker. Iedereen kent elkaar, er heerst een gezellige sfeer en de communicatie verloopt veel beter dan bij een groot bedrijf.

Tot slot kan ik met trots mededelen dat ik mijn (virtuele) hoogtevrees heb overwonnen dankzij het ontwikkelen van deze game.



# Inhoudstabel

<b>Voorwoord .....</b>	<b>1</b>
<b>Inleiding .....</b>	<b>4</b>
<b>Inhoudstabel .....</b>	<b>6</b>
<b>1 Kennismakingsmail .....</b>	<b>10</b>
<b>2 Recherche, CV et entretien d’embauche .....</b>	<b>11</b>
<b>3 Informaticaonderzoek van het stagebedrijf .....</b>	<b>12</b>
3.1 Omschrijving van het aanwezige netwerk.....	12
3.1.1 stagebedrijf.....	12
3.2 Opleiding van het personeel.....	12
3.3 Gebruikte softwarepakketten .....	12
3.4 Aanwezigheid op het internet .....	13
3.5 Beveiliging van het netwerk .....	13
3.6 Back-up strategie .....	13
3.7 Ergonomie .....	13
3.8 Noodbatterij .....	14
3.9 Toekomstvisie .....	14
3.10 Marketingmix van de stageplaats.....	15
<b>4 Probleemanalyse van de stageopdracht .....</b>	<b>16</b>
4.1 Het bedrijf.....	16
4.2 Mijn stageopdracht .....	16
4.2.1 De game.....	16
4.2.2 Mijn eigen versie .....	16
4.2.3 De benodigdheden .....	17
4.2.4 Voorbereiding.....	17
4.2.5 2D game.....	18
4.2.6 3D game.....	18
<b>5 Technisch verslag van de stageopdracht .....</b>	<b>20</b>
5.1 Het voorbereidend werk .....	20
5.1.1 De game.....	20
5.1.2 Programma’s .....	20
5.1.3 HTC VIVE .....	21
5.2 De eerste bewegingen.....	23
5.2.1 VR Camera .....	23
5.2.2 VR Controllers.....	24
5.2.3 Controller acties inlezen .....	24
5.3 Een vlotte zachte beweging.....	25
5.4 Een leeg object aan de speler koppelen.....	26



5.5	De grond detecteren .....	27
5.5.1	De werking van een Raycast .....	28
5.6	Virtual Reality headset traceren .....	28
5.7	De speler kantelt door platformen .....	29
5.8	Turbo functie .....	29
5.9	Enums .....	30
5.10	Het menu .....	31
5.11	Het design .....	32
5.12	Script ringen .....	33
5.13	Een realistische zee .....	34
5.14	Aanpassen van de bewegingen .....	35
5.15	Positie van de ringen .....	35
5.16	Het volledige menu .....	36
5.17	Eindsituatie .....	38
<b>6</b>	<b>Evaluatie van de leerervaringen .....</b>	<b>39</b>
6.1	Wat heb ik geleerd? .....	39
6.2	Hoe zou ik het een tweede keer aanpakken? .....	39
6.3	De toepassing .....	40
<b>7</b>	<b>Stageverslag .....</b>	<b>41</b>
<b>8</b>	<b>Work Experience Project .....</b>	<b>45</b>
8.1	The assignment .....	45
8.2	Preparation .....	45
8.3	The internship .....	47
8.4	Evaluation .....	47
<b>9</b>	<b>Fiche entreprise .....</b>	<b>48</b>
<b>10</b>	<b>Zelfevaluatie .....</b>	<b>50</b>
<b>11</b>	<b>Dankbrief .....</b>	<b>53</b>
<b>12</b>	<b>Uitnodiging mondelinge verdediging .....</b>	<b>54</b>
<b>13</b>	<b>Code .....</b>	<b>55</b>
13.1	Playerbehaviour script .....	56
13.2	FollowSmooth Script .....	57
13.3	Turbo script .....	58
13.4	Float script .....	59
13.5	Controllertype (enum) script .....	59
13.6	DetectGround script .....	60
13.7	ItemPickUp script .....	61
13.8	VRUIItem .....	62

13.9	VRUIInput script.....	63
13.10	ModeSelect script.....	64
13.11	DeadPanel.....	64
13.12	StartPanel script .....	65
13.13	EndPanel script .....	65
<b>Besluit</b>	.....	<b>66</b>
<b>Bronvermelding</b>	.....	<b>67</b>
<b>Bijlagen</b>	.....	<b>68</b>



# 1 Kennismakingsmail



Naam: Evi Boelen  
Klas: 6ITN  
Datum: 13 oktober 2019

---

## Kennismakingsmail

<p><b>Van:</b> evi.boelen@icloud.com <b>Datum:</b> 13 oktober 2019 <b>Aan:</b> info@firewolf.be <b>Onderwerp:</b> Kennismakingsmail</p>
---

Geachte meneer Van Geirt

Als 6dejaars leerling in de richting IT&N (informaticabeheer) van de secundaire school Campus HAST te Hasselt, moet ik in het kader van mijn geïntegreerde proef stage doen bij een informaticabedrijf. Hiervoor koos ik voor Firewolf Studios.

Ik heb namelijk vernomen van een voormalige stagiair dat jullie werken met het programma Unity. Ik zou hier zelf heel graag mee leren werken. Naar aanleiding van ons telefonisch gesprek van gisteren verstuur ik u deze mail met verdere details over de stage.

De stage zal doorgaan van maandag 18 februari tot en met vrijdag 1 maart. De bedoeling is dat ik in deze periode een stageopdracht krijg van jullie die aansluit bij mijn richting. Ik ben bij voorkeur iedere woensdag namiddag vrij om verdere details hierover te bespreken. Ik zou graag een afspraak maken voor een verkennend gesprek. Dit kan bij voorkeur op woensdag namiddag of op andere weekdays van 16:30.

Ik kijk uit naar een persoonlijke ontmoeting en hoop snel wat terug te horen.

Met vriendelijke groeten

Evi Boelen

## 2 Recherche, CV et entretien d'embauche

### CURRICULUM VITAE

**Boelen Evi**

Stiemberbeekstraat 29

3600 GENK

0489/501/321

evi\_boelen@hotmail.com

09/09/1998

Nationalité belge

**Formation**IT&N  
Hast Hasselt

2016 – présent

Sciences de l'ingénierie sociale  
Regina Mundi Genk

2015 – 2016

La formation d'aide logistique  
Don Bosco Genk

2010 – 2015

**Compétences linguistiques**néerlandais : langue maternelle  
L'anglais: parler couramment**Compétences informatiques**Word: très bonnes connaissances  
Excel: bonnes connaissances  
C# : connaissance de base  
HTML : connaissance de base  
CSS : connaissance de base**Expérience professionnelle****PXL Hasselt**

2014 - 2015

- Aide de cuisine  
(stage)

**Kinderkampen Kadee! Hasselt**

2015 - présent

- Animateur

**VZW Olijfboom**

2014 -2015

- Aide logistique  
(stage)

**Points fortes****Points faibles**

Être capable de respecter des délais	Je peux paraître stressée. Mais au fond, c'est un moteur pour continuer de me dépasser et atteindre les objectifs fixés.
Être capable de travailler en équipe et progresser au sein de celle-ci	Je suis une personne sensible, mais je maîtrise de mieux mon émotivité.
Être capable d'apprendre de comprendre et d'appliquer de nouvelles technologies	Je suis un peu orgueilleuse, mais j'essaie d'y faire attention et je suis capable de me remettre en question.

## **3 Informaticaonderzoek van het stagebedrijf**

### **3.1 Omschrijving van het aanwezige netwerk**

#### **3.1.1 stagebedrijf**

In firewolf studios bevinden zich geen servers. Ze werken met een wifi router. Deze werkt gewoon met een abonnement zoals je deze bij een provider zoals Telenet zou nemen. Ze hebben wel servers nodig voor het hosten van hun website en voor hun database. Ze maken gebruik van een extern hosting bedrijf genaamd hostYou. Hier kunnen ze tegen betaling hun servers laten onderhouden. Het voordeel hiervan is de prijs. Als ze met servers zouden werken moeten ze de hardware aankopen en een beheerder aanwerven. Dit geeft alleen maar extra kosten. Hierdoor is het goedkoper om hun servers te laten beheren door hostYou. Firewolf investeert liever in mensen die mee kunnen werken aan projecten (games). Deze mensen leveren hun geld op terwijl een serverbeheerder hun alleen maar geld zal kosten. Een eigen server wordt pas interessant als er servers opgezet moeten worden voor gameprojecten (bijvoorbeeld multiplayer games). Er wordt samengewerkt aan de hand van een extern programma genaamd SourceTree. In dit programma kan elke programmeur of designer apart werken en hierna zijn bewerkingen in het eigenlijke project opslaan.

### **3.2 Opleiding van het personeel**

De werknemers van Firewolf Studios scholen zich dagelijks bij door op te zoeken wat ze nodig hebben. Hier hoort zelden een extern opleidings-moment bij. Het internet is de grootste informatiebron. Het is financieel niet mogelijk voor Firewolf Studios om werknemers naar specifieke opleidingen te sturen. Online cursussen worden soms wel aangeboden. Het belangrijkste voor dit bedrijf is dat de werknemers geïnteresseerd zijn in hun werk en dat ze hier een goede kennis van hebben. Er wordt geen verschil gemaakt tussen een bachelor of een master diploma. De cursussen om een certificaat te behalen in Unity kan je terug vinden op de website [certification.unity.com](http://certification.unity.com). De certificaten verieren van gewoon programmeur tot gameplay programmeur of technisch artiest shading & effects.

### **3.3 Gebruikte softwarepakketten**

De voornaamste software die het bedrijf gebruikt is Unity. Dit is een game engine die gebruikt wordt om hun games in te ontwikkelen. Visual studio wordt gebruikt als scripting editor. Hier wordt de nodige codering voor de games in gedaan. Gianluca de 3D-ontwerper maakt gebruik van Adobe photoshop en 3D blender voor het ontwikkelen van 3D objecten en het animeren van deze objecten. Frank gebruikt Eclipse PHP voor het scripten in PHP. Dit wordt gebruikt om de website te maken. PhpMyAdmin wordt gebruikt voor het werken in databases. In de plaats van het veel gebruikte Microsoft office pakket maakt Firewolf Studios gebruik van Libre office. Dit is een gratis pakket. Het bedrijf maakt ook gebruik van dropbox als back-up. Voor het delen van bestanden wordt gitlab gebruikt. Dit programma geeft je de mogelijkheid bestanden te delen met elkaar aan de hand van een link. SourceTree is een software die gebruikt wordt om samen te kunnen werken aan projecten. Hiermee kun je tegelijk aan hetzelfde project werken. De software wordt gebruikt

door degene die het nodig heeft. Zo wat elke computer of laptop in het kantoor beschikt over een versie van Unity. De 3D ontwikkelaar maakt gebruik van de programma's die hij nodig heeft, de ontwikkelaars gebruiken Visual Studio en zo voort. Het bedrijf probeert zo veel mogelijk met open source (gratis) programma's te werken. De rest is gebaseerd op een abonnement en word dus maandelijks betaald.

### **3.4 Aanwezigheid op het internet**

Firewolf is zowel aanwezig op het internet als op social media met als doel klanten te bereiken met het bedrijf en eventuele promotie te voeren. Op het internet hebben ze een website met informatie over het bedrijf. De social media accounts bestaan uit een account op facebook, instagram en LinkedIn. De website wordt onderhouden door Frank Van Geirt. De social media accounts worden gezamenlijk beheerd door de werknemers van Firewolf Studios. De enige richtlijn die het bedrijf toepast op de social media platformen is dat de communicatie in het engels gebeurt.

### **3.5 Beveiliging van het netwerk**

Het netwerk word beveiligd door de internet provider waar ze een abonnement bij hebben. Dit gebeurt bij hun enkel met een wachtwoord. Het internet zelf word beveiligd met een wifi wachtwoord. Omdat er geen servers aanwezig zijn is er geen nood aan firewalls, ... . Zowel de werknemers als de bezoekers van het bedrijf mogen gebruik maken van het internet.

### **3.6 Back-up strategie**

Er is geen specifieke back-up procedure of systeem. De rede hiervan is dat ze dropbox gebruiken. Gitlab word gebruikt om projecten te delen. Gitlab dient als back-up voor de projecten waar aan gewerkt word. Elke wijziging die de gebruiker doet aan de game, en hierna toepast op de SourceTree software word automatisch ook toegepast op het project in gitlab.

De back-up word beveiligd door de firma's van de programma's zelf. Ze zorgen er voor dat de documenten veilig staan. Voor de rest word hier geen back-up bijgehouden. Zelf zou ik dit een beetje anders aanpakken. Ik zou elke dag een kopie van de gewerkte projecten op een externe harde schijf plaatsen. Zo ben je zeker dat je nooit je werk kwijt raakt. Op deze manier ben je wel niet beveiligd tegen het cryptolocker virus aangezien de data direct op de schijf wordt gezet.

### **3.7 Ergonomie**

Volgens de ergonomische richtlijnen zou het beeldscherm van hoogte verstelbaar moeten zijn, de stoel moet eveneens volledig verstelbaar zijn. De kamer moet voldoende verlucht zijn. En de werknemer moet ook vaak genoeg de benen strekken om medische problemen te voorkomen.

Firewolf Studios doet zijn best om te voldoen aan deze richtlijnen. Ze gebruiken verstelbare schermen zodat deze altijd op de juiste hoogte staan. Hun meest recentelijke aankoop bestond uit volledig verstelbare ergonomische stoelen.

Jaarlijks krijgt Firewolf Studios controle van de IDEWE, een organisatie waar bedrijven lid van moeten zijn die zich ontfemen over de ergonomie in bedrijven. Ze komen elk jaar langs om te kijken of ze het bedrijf ergens kunnen bijsturen of helpen. De volgende stap die het bedrijf wil zetten is het aanschaffen van verstelbare werktafels. Als de hoogte van de tafel aanpasbaar is, draagt dit bij tot de ergonomische houding van de werknemer. Nog een volgende stap is een uurtje van de dag recht staan aan deze verstelbare bureaus. Als je lang stil zit kun je last krijgen van bloedklonters in je benen. Het recht staan zal dit tegen gaan.

### **3.8 Noodbatterij**

Een noodbatterij slaat elektriciteit op om dit later te kunnen gebruiken. De noodbatterij wordt ingeschakeld op het moment dat de frequentie van het signaal voor meer dan 6 seconden 6 procent onder de gebruikelijke frequentie valt. Als dit gebeurt zal de noodbatterij het overnemen van de gewone stroomtoevoer. Deze werkt geen uren, maar kan er voor zorgen dat je tijdens een stroomuitval alle bestanden veilig kan opslaan. Firewolf maakt geen gebruik van een noodbatterij, dit is niet hun verantwoordelijkheid. Ze betalen voor de elektriciteit van de Corda Campus. Als de elektriciteit uit valt in de Corda Campus wordt er een back-up plan voorzien door Corda Campus zelf. Een groot risico hiervan is dan wel weer dat bij een stroomuitval de vooruitgang aan de projecten waar ze die dag aan hebben gewerkt verloren gaat. Maar één werkdag verloren is minder erg dan bijvoorbeeld een hele week.

### **3.9 Toekomstvisie**

Het stagebedrijf maakt voornamelijk gebruik van het besturingssysteem windows10. Op één van de computers wordt gewerkt met windows7. De reden hiervoor is dat het nog niet nodig is geweest om te upgraden naar een nieuwere versie. Één van de werknemers maakt gebruik van een iOS systeem op een iMAC. Als je games wil ontwikkelen die ondersteund worden door iOS moeten deze ook gebouwd worden vanaf een Apple toestel.

Het voordeel aan het gebruik van Windows is dat alle courante pakketten beschikbaar zijn. Het nadeel is dan weer dat je volledig gebonden bent aan Microsoft. iOS heeft dan weer het voordeel dat iOS eenvoudiger is om mee te werken. Maar het is ook technisch meer dicht getimmerd.

Voor de cloudservice wordt er gebruik gemaakt van SaaS (Dropbox en gitlab). In de toekomst zal dit zo blijven. Het bedrijf maakt al gebruik van externe servers die ze niet zelf moeten beheren.

Als toekomst visie voor het bedrijf hebben ze nog geen concreet idee. Op dit moment wordt er gekeken naar wat er vandaag en morgen nodig is. Er kan gekeken worden naar wat er over een jaar nodig is maar de prioriteiten in het bedrijf veranderen van dag tot dag. Het doel van Frank Van Geirt (stagebegeleider en co-eigenaar van FirewolfStudios) is het uitgroeien naar een bedrijf van ongeveer 20 mensen. Maar voor dat daar aan gewerkt kan worden zijn er eerst andere dingen die nu belangrijk zijn.



### **3.10 Marketingmix van de stageplaats**

Firewolf Studios biedt zijn klanten learning en advising games. Dit betekent dat ze games maken die de klant kan gebruiken om hun eigen bedrijf te promoten. Voor Firewolf is het belangrijk dat de klanten het product kunnen gebruiken in hun voordeel. De klant moet het drievoud van de betaalde prijs er terug kunnen uithalen. De game wordt volledig ontwikkeld naargelang de wensen en de eisen van de klant.

Het bedrijf is gevestigd op de Corda Campus in Diepenbeek. Een voordeel van deze vestigingsplaats is de grote mogelijkheid tot contacten met andere bedrijven. Dit komt omdat de Corda Campus boordevol zit met bloeiende IT bedrijven.

De games worden gemaakt voor de klant en ze verkopen deze vervolgens ook aan deze klant. Voor de games zelf maken ze dus geen reclame of promotie. De werknemers van Firewolf Studios gaan wel naar beurzen en maken zich zo bekend. Social media platformen spelen ook een rol in het verspreiden van hun bekendheid. Zo krijgen ze meer potentiële klanten.

De prijs van de game hangt af van hoe uitgebreid de eisen van de klant zijn, hoe lang er aan gewerkt wordt, en hoe gedetailleerd de game zal zijn.

## 4 Probleemanalyse van de stageopdracht

### 4.1 Het bedrijf

Tijdens mijn stage zal ik ingeschakeld worden in het bedrijf Firewolf Studios. Firewolf Studios is een software bedrijf dat zich bevindt op de Corda Campus in Diepenbeek. Hun bedrijf creëert educatieve en adverterende games op aanvraag van klanten. Met de slogan “Level up your business” maken ze potentiële klanten duidelijk dat ze de game ontwikkelen zodat de klant hier het maximale kan uit halen.

### 4.2 Mijn stageopdracht

#### 4.2.1 De game

Tijdens deze stage zal ik een game mogen ontwikkelen. De game zal mijn eigen versie zijn van de game “Richies Plank Experience”. Deze stelt zich voor als een VR game. Een VR game (of Virtual Reality game) is exact naar wat de naam verwijst. Je zet een VR-bril op, en zodra je door de lenzen van de bril kijkt, stap je een virtuele wereld binnen (naargelang het spel dat je speelt). In Richies Plank Experience stap je een lift in zodra de game begint, deze brengt je naar boven. Eenmaal boven aangekomen gaat de lift open en moet je over een dunne plank lopen en hierna van af springen.



*Richies Plank Experience*

#### 4.2.2 Mijn eigen versie

Mijn eigen versie van deze game start in een vliegtuig. Eerst was het plan om hier verplicht uit te moeten springen, maar uiteindelijk hebben we besloten om de speler zelf aan te laten geven wanneer/of hij wil springen door op de start knop te drukken. In de plaats van over een plank te moeten wandelen, spring je uit een vliegtuig en glijd je met de lucht mee naar beneden. Tijdens de vlucht moet je zo veel mogelijk ringen verzamelen door er doorheen te zweven. Het spel eindigt als je op een bootje in een meer land. Deze game gaat uiteindelijk dienen voor Firewolf Studios om op beurzen te presenteren. Hierdoor kunnen de mensen een kleine Virtual Reality ervaring beleven.

### 4.2.3 De benodigheden

De game zal ontwikkeld worden in de game engine Unity. Dit is een software die gemaakt is om games in te ontwerpen. Visual Studio zal gebruikt worden om in te scripten. Scripten is het automatiseren van processen. In een script worden er instructies gegeven aan objecten of variabelen die uitgevoerd worden wanneer het script opgeroepen word. Visual Studio gebruiken we ook op school, dit is een softwareontwikkeling omgeving om websites, apps, mobiele apps, ... te ontwikkelen.



HTC VIVE VR-headset

Om de game in een VR-omgeving te kunnen zetten is er een VR-bril nodig. Tijdens mijn stage zal ik gebruik gaan maken van de HTC Vive. De HTC Vive is een VR-bril van het merk HTC. Om deze aan te kunnen roepen en te koppelen in Unity zal ik de software SteamVR nodig hebben. De koppeling word gelegd door SteamVR downloaden en te importeren naar het project via de Unity AssetStore. Deze software zit vol met functies die je kan aanroepen in een script in Unity. Hierdoor is het mogelijk om acties aan de controllers en de VR-bril te koppelen

### 4.2.4 Voorbereiding

Als voorbereiding op de opdracht heeft mijn mentor mij gevraagd om me te verdiepen in enkele aspecten van Unity. Het belangrijkste was dat ik het scripten een beetje onder de knie kreeg. Om dit te doen heb ik enkele tutorials gevolgd op de officiële website van Unity.

Onder anderen de Roll a Ball en de 3D Game Kit tutorial. Na dat ik de tutorials heb gevolgd, heb ik een groot deel van de scripting tutorials bekeken om mijn kennis over scripting taal bij te schaven. YouTube hielp me ook met mijn voorbereiding. Dit was echter altijd voor heel specifieke dingen. Dit waren te veel bronnen om te vermelden. Hierna ben ik zelf begonnen aan een game. Eerst een kleine 2D game, maar omdat 3D een grotere prioriteit was ben ik begonnen aan een 3D game. Hier werk ik nu nog af en toe aan.

## Projects



### Interactive Tutorials (4)

Get Started with Unity.



### Roll-a-ball tutorial (9)

Build your first simple game and Learn to code in C#



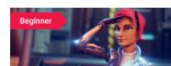
### Unity Playground (10)

A gentle introduction to Unity for the very beginner. Perfect for teachers!



### 2D Game Kit (38)

Create your own 2D platform game without writing any code.



### 3D Game Kit (34)

Explore and create in this 3D game kit without writing any code.



### Space Shooter tutorial (19)

Blast some Asteroids!



### Survival Shooter tutorial (12)

They mostly come at night..



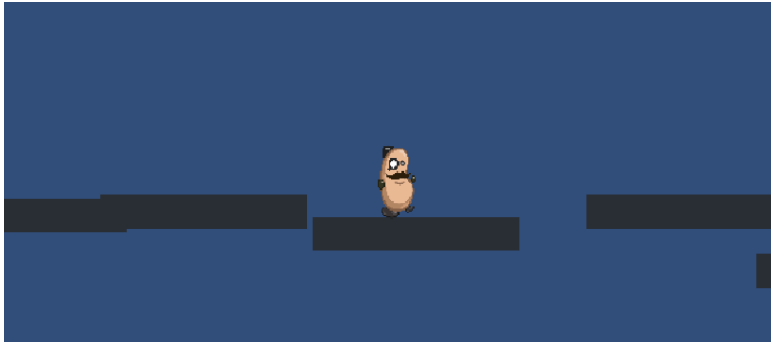
### 2D UFO tutorial (9)

New? Want to make 2D games? Start here.

*Tutorials op de Unity website*

#### 4.2.5 2D game

Mijn 2D voorbereiding bestond uit een kleine game (met een aardappel als hoofdkarakter). De speler moet binnen een tijds spanne zo snel mogelijk op platformen springen die op een willekeurige plaats verschijnen. De game is nog niet verder uitgewerkt dan dit. De reden hiervan is omdat een 3D game voor mij een hogere prioriteit had. Mijn stageopdracht zou namelijk bestaan uit een Virtual Reality game en deze wordt ontwikkeld in een 3D omgeving.



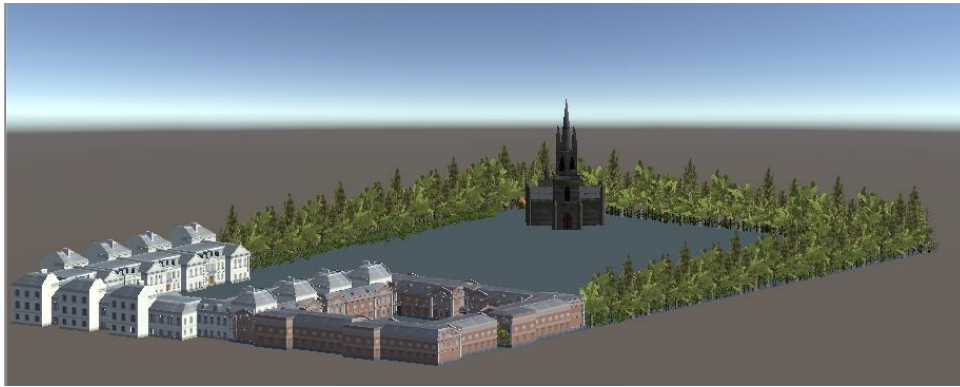
*Print screen uit een onafgewerkte game als voorbereiding van mijn stage*

#### 4.2.6 3D game

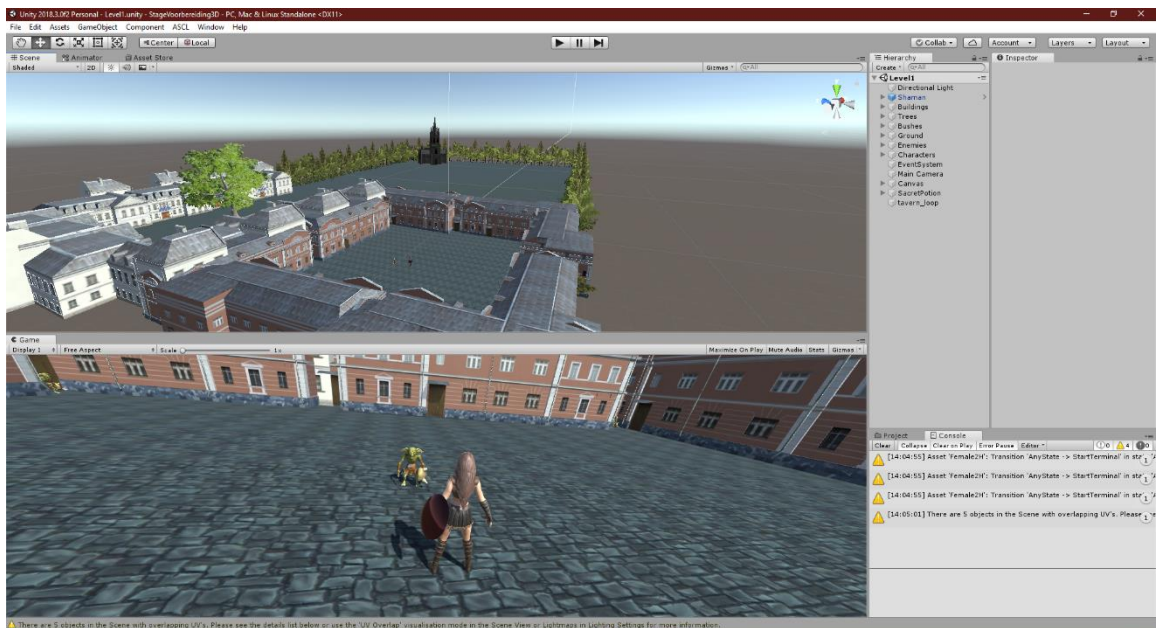
Op de onderstaande afbeelding ziet u mijn speler (Vrouwelijke krijger) en een vijand (goblin). Om me voor te bereiden op mijn stage heb ik er voor gezorgd dat ik wist hoe ik bepaalde acties aan verschillende objecten kon koppelen. De vrouwelijke krijger kan de goblin aan vallen, na deze 5 keer te raken sterft de goblin en word er een punt bij de kill score van de speler gezet. In de game zijn er ook enkele quests beschikbaar. Één daarvan is het zoeken van een magisch drankje om het stervende woud te redden.



*Print screen uit een onafgewerkte game als voorbereiding van mijn stage*



Print screen uit een onafgewerkte game als voorbereiding van mijn stage



Print screen uit een onafgewerkte game als voorbereiding van mijn stage

## 5 Technisch verslag van de stageopdracht

### 5.1 Het voorbereidend werk

#### 5.1.1 De game

Tijdens mijn eerste dag werd er besloten wat mijn stageopdracht concreet zou zijn. Dit gebeurde tijdens een kleine vergadering bij aanvang van de stage. Mijn stageopdracht werd een Virtual Reality parachute simulator. Virtual Reality is zoals de naam al verklaart een virtuele realiteit. Deze kan je betreden aan de hand van een VR-bril met bijhorende VR-controllers. Tijdens mijn stage maakte ik gebruik van de HTC Vive. Dit is een Virtual Reality bril van het merk HTC.

In de game begin je hoog in de lucht staand in een vliegtuig. De deur van het vliegtuig staat open. Je kan pas uit het vliegtuig springen als je op de startknop drukt. Eenmaal gedrukt op de startknop kan je uit het vliegtuig springen (of gewoon blijven staan) en moet je je navigeren doorheen ringen in de lucht. De game eindigt als je landt op het daarvoor voorziene bootje, het water of het terrein. Als je kan landen op het bootje krijg je je score te zien (aantal ringen waar je doorheen vloog), samen met de keuze om het spel te herstarten of om het spel af te ronden.. Als je op het terrein of het water land krijg je als melding dat de missie gefaald is en dan krijg je eveneens de keuze om te herstarten of het spel af te ronden.

#### 5.1.2 Programma's

De nodige programma's om de opdracht te realiseren:

- Unity
- Visual Studio
- Gitlab
- SourceTree
- SteamVR



Unity werd gebruikt als game engine. Dit betekent dat de game die ik zal ontwikkelen in deze engine gemaakt zal worden. Unity ondersteunt scripts in de programmeertaal C# en C++.

Scripts zijn lijsten van commando's die uitgevoerd worden door een script-engine. Scripts worden gebruikt om processen of acties te automatiseren.

Visual Studio is de ontwikkelaarsomgeving waar scripts in ontwikkeld worden. Visual Studio biedt een mogelijkheid om programma's te ontwikkelen in verschillende programmeertalen. Enkele van de ondersteunde programmeertalen zijn C, C++, C#, Python en Node.js.

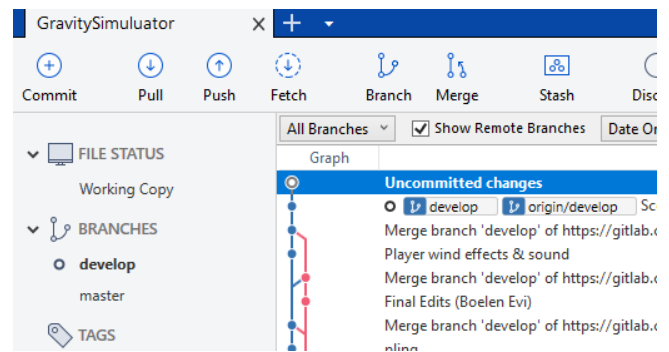
Gitlab is een platform dat je de mogelijkheid geeft om samen te programmeren. Je kan projecten delen met elkaar aan de hand van een link. Je upload een programma op Gitlab, en er word een link gecreerd. Deze link kan je ingeven als



een nieuw project op de SourceTree. De veranderingen die worden gedaan worden direct toegepast op het programma. Als een andere gebruiker ook aan het programma werkt, kan je SourceTree gebruiken om je veranderingen volledig vast te leggen. Pas als je op de SourceTree “pusht” (je veranderingen opslaat), en natuurlijk als de andere gebruiker deze gepushte dingen ook “pulled” (veranderingen van anderen op halen) worden de veranderingen ook toegepast. In de SourceTree werk je namelijk op de link van Gitlab. Als je pusht komen je aanpassingen dus ook terecht in het eigenlijke project.

SourceTree wordt gebruikt om samen te werken aan projecten en als een kleine

backup. Op SourceTree kan je de link die je van Gitlab af haalde (van het programma waar gezamenlijk aan gewerkt word). In dit programma kan je op verschillende afdelingen werken waardoor er geen conflicten ontstaan als er twee programmeurs aan één project werken op het zelfde moment. Als dezelfde code is aangepast kan er gekozen worden tussen welke van de twee veranderingen toegepast zal worden. Zoals ik eerder zei word SourceTree ook gebruikt als back-up. Dit komt omdat je met SourceTree de veranderingen pas toepast (op Gitlab en SourceTree) als je hier toestemming voor geeft.



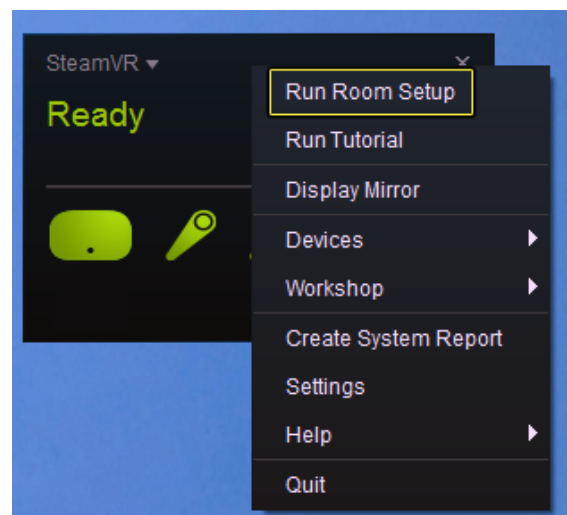
*Print screen op SourceTree van mijn stageproject, een voorbeeld van hoe het programma er uit ziet.*

SteamVR is een Virtual Reality software die word gebruikt als ondersteuning voor de HTC Vive

### 5.1.3 HTC VIVE

Om een Virtual Reality game te ontwikkelen moet je beschikken over een VR-bril. Deze kan je aan de hand van de net vermelde SteamVR software koppelen aan Unity. De steamVR software zit boordevol met scripts en andere leuke functies. Deze kan je gebruiken om de VR-bril samen met de controllers een functie te geven.

SteamVR zal eerst een “room setup” uitvoeren. Hier kan je je speelterrein markeren en je plaats in dit terrein aangeven.



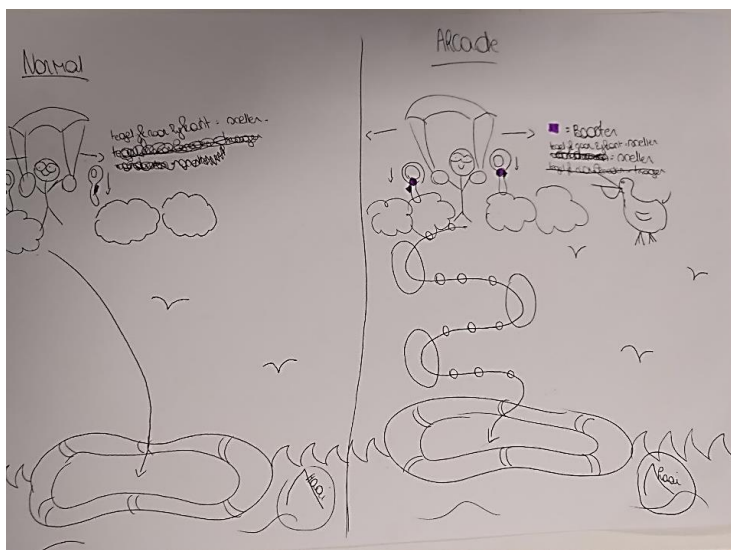
*Steam VR setup*

In deze setup kan je ook aangeven waar de grond zich bevind.



Steam VR setup

Als de HTC Vive klaar is voor gebruik, is het belangrijk om eerst mijn visie van het spel op papier uit te tekenen. Hieronder vind u een kleine schets van het spel. Op onderstaande schets zie je een “Normal” en een “Arcade” gamemode. Het verschil hierin is dat je in de Normal mode enkel naar beneden moet vliegen. In de Arcade mode kan je doorheen ringen en obstakels vliegen.



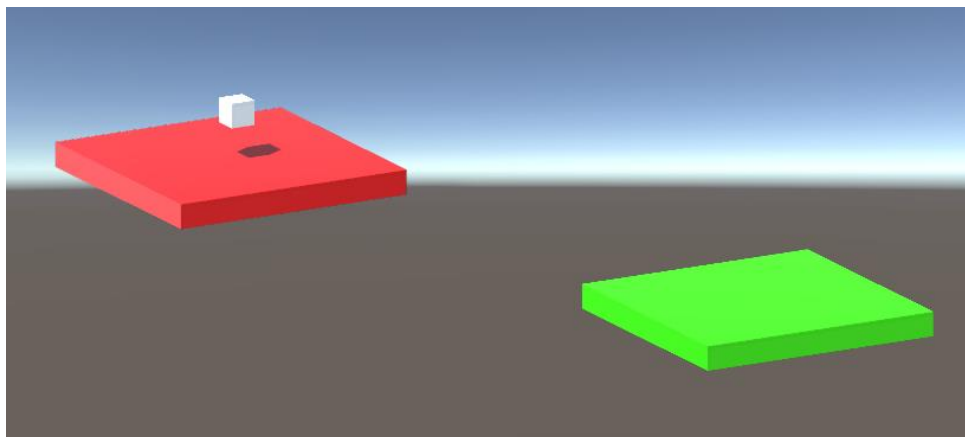
Schets van het programma

Om te beginnen word eerst de Arcade mode uitgewerkt. Om dit tot een goed einde te brengen is het belangrijk dat ik in enkele scripten de bewegingen van de speler zal schrijven. De grond zal moeten gedetecteerd kunnen worden. De speler zal doorheen ringen moeten kunnen vliegen, en dit moet gedetecteerd en opgeteld worden. Ik zal een menu moeten maken waarmee de speler opnieuw kan spelen indien hij dit wilt.



## 5.2 De eerste bewegingen

Vooraleer er aan het uiterlijk en het Virtual Reality aspect van de game gewerkt word, is het belangrijk om een beginsituatie te creëren. Voor mijn game bestond deze uit een startplatform (rood), een landplatform (groen) en een speler (wit). Alle objecten in deze game zijn op dit moment vierkanten, de platformen wat groter dan de speler.

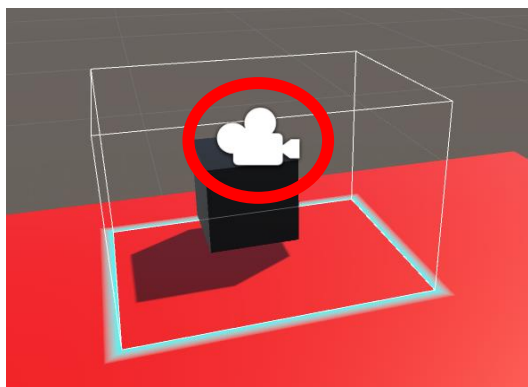


*Een vroeg voorbeeld van de game zonder interface*

De bewegingen van dit object werden gecontroleerd door een klein script. Dit script vind je terug op pagina 24. Hierin werd gekeken naar de invoer van de gebruiker. Hiermee word bedoeld dat er gekeken word naar op welke knop de speler drukt. Op basis daarvan werd er een beweging uitgevoerd. Als er op het pijltje op het klavier naar links gedrukt werd, beweegt de speler naar links, als het rechter pijltje ingedrukt werd, bewoog deze naar rechts. Het script op pagina 24 maakt wel al gebruik van de HTC VIVE controllers.

### 5.2.1 VR Camera

Om een VR game te ontwikkelen moet je gebruik kunnen maken van een VR camera (die weergeeft wat je ziet in de VR-bril). De camera is een functie die je uit het SteamVR pakket kan halen door hem gewoon op een object kan slepen. De camera bevat een 'head camera, een 'eye camera' en een vierkante box. De head camera verwijst naar de headset en de eye camera verwijst naar het scherm dat je op de VR-bril ziet. Het vierkant is je speelomgeving.



*VR camera in de game*

## 5.2.2 VR Controllers

Vervolgens moeten de VR controllers ingesteld worden. Dit moest ik doen aan de hand van een externe website. Deze website is enkel toegankelijk vanuit een editor die toegankelijk is vanuit Unity. De link kan ik hier dus helaas niet in plaatsen. In de editor koppel je acties aan de controller die je dan vervolgens kan oproepen in een script om hier een concrete beweging of actie aan toe te voegen. Een Button, trackpad, trigger en een grip zijn knoppen die je op de controller terug vindt. Aan de hand van deze editor kan je dus functies geven aan de knoppen.



### HTC VIVE controller instellingen

Hier ondervond ik wat hinder bij. De SteamVR software was pas vernieuwd. Het systeem dat hierboven vermeld staat hoort bij de nieuwere versie. Hier was zeer weinig documentatiemateriaal over te vinden om mij op te baseren. Hier heb ik bijna een volledige dag opzoekwerk over gedaan. Helaas tevergeefs. Het uitzoeken van het nieuwe systeem ging te lang duren dus er werd me geadviseerd om opzoek te gaan naar een oudere versie van SteamVR. Hier was meer dan genoeg documentatiemateriaal over te vinden op websites zoals het Unity forum, Circuitstream, Youtube en nog heel veel andere. De rede dat ik van dit documentatiemateriaal geen links heb is omdat dit gewoonweg te veel was.

## 5.2.3 Controller acties inlezen

In de game zal de speler de triggers van de HTC Vive controller gebruiken om te sturen doorheen de lucht. Als deze ingedrukt worden moet de speler dus naar de gevraagde kant bewegen. Dit deed ik aan de hand van onderstaande lijn code. Hier word er op gekeken of er één van de triggers (knoppen op de HTC VIVE controller) gedrukt word. Met deze functie kan de speler nu naar voor gaan. Later word er een

extra uitbreiding gedaan waarmee er gedetecteerd wordt welke trigger er gebruikt wordt en zo zal de speler kunnen draaien.

```
if (Controller.GetHairTriggerDown())
```

*Script: Hoe het indrukken van een trigger gedetecteerd wordt*

Dit werd gezet in een 'if-statement'. Een if-statement wordt gebruikt om na te gaan of iets aan een bepaalde waarde of actie voldoet. Zo ja zal de code die tussen de accolades er onder staat uitgevoerd worden.

De code die tussen de accolades staat is in dit geval:

```
if (Controller.GetHairTriggerDown())
{
    moveDirection = new Vector3(0, moveValue, 0);
    trackedObj.transform.Translate(moveDirection * Time.deltaTime);
}
```

*Script: De speler gaat naar voor als de trigger gebruikt wordt*

De moveDirection stelt zich voor als een Vector 3. Dit betekent dat hier 3 waarden ingevoerd kunnen worden. Een vector 3 wordt gebruikt als je in 3D werkt. Je kan ook met een Vector 2 werken (x en een y waarde), maar dit zal natuurlijk enkel de waarden op de x en de y as kunnen aanpassen. De waarden worden toegepast op 3 assen. De eerste waarde op de x-as, de tweede waarde op de y-as en de derde waarde op de z-as. Hier wordt zowel de beweging op de x-as als die op de z-as op 0 gezet. Dit komt omdat we willen dat de speler zal dalen als er op de trigger gedrukt wordt. Op de trigger drukken simuleert het trekken aan de parachute zodat je sneller daalt naar de kant waar je aan trekt. De y-as krijgt de waarde van de moveValue. De moveValue staat publiek en is dus aanpasbaar. Dit is een waarde die je kan veranderen in de inspector van Unity. Dit maakt het gemakkelijker om de waarden te veranderen om te testen want je kan deze waarden ook veranderen als je de game opstart in Unity.

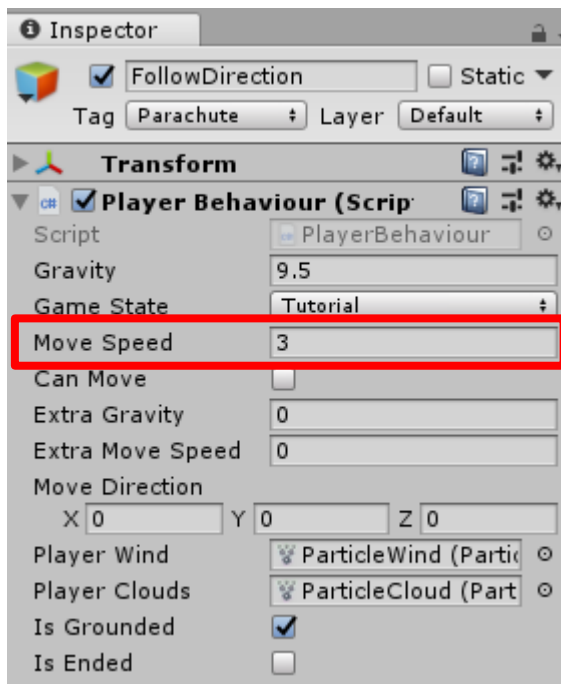
### 5.3 Een vlotte zachte beweging

Nu de speler instaat is om te bewegen wordt het tijd dat deze ook kan draaien. Dit wordt gedaan met een procedure. Er wordt een variabele doorgegeven vanuit het hoofdprogramma. Deze wordt ingevuld in de procedure. In dit geval is de variabele movementValue. Dit is een aanpasbare variabele die ik zelf heb aangemaakt. In de inspector kan de waarde van de MovementValue veranderd worden. Hier wordt de movementValue op de z-as gezet. Dit komt omdat de speler zal moeten roteren naar de gevraagde kant. In de plaats van 'translate' te gebruiken zoals in het vorige

```
//Glide in the air
public void Dive(float movementValue)
{
    //The player object rotates on the Z-axis
    moveDirection = new Vector3(0, 0, movementValue);
    transform.Rotate(moveDirection);
}
```

script maken we nu gebruik van Rotate. Deze laat het object roteren. Op de volgende pagina is hier een voorbeeld van.

*Script: De speler draait rond met de 'Rotate' functie*



*Voorbeeld MovementValue*

De beweging is nu a wat vlotter maar nog niet perfect. Wouter Spaas (collega bij Firewolf Studios) hielp me om de beweging optimaal te maken. Hij legde me uit dat ik gebruik kon maken van een 'Lerp' functie. Deze zorgt voor een vlotte beweging.

```
//The player object tracks the movement of the chosen target (target.position)
//The player object follows this movement with a speed assigned as Dampine (with delay)
transform.position = Vector3.Lerp(transform.position, target.position, positionDamping * Time.deltaTime);
```

*Script: Voorbeeld van de Lerp functie*

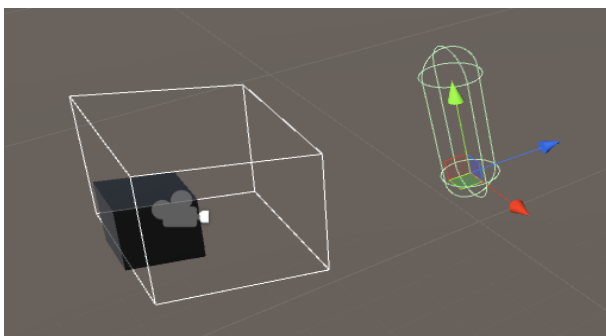
In de bovenstaande code gebruik ik de Lerp functie. Deze zorgt ervoor dat de beweging die uitgevoerd word, verzacht wordt. Hierdoor krijg je dus een mooie vlotte zachte beweging. In de Lerp functie worden 3 variabelen gebruikt. Transform.Position, Target.Position en een PositionDamping. De Transform position geeft het beginpunt weer en de Target.Position geeft het eindpunt weer. De PositionDamping zal er voor zorgen dat de speler van het beginpunt naar het eindpunt reist met een vertraging van de PositionDamping. De PositionDamping is een variabele die net zoals de MovementValue aangepast kan worden in de inspector.

Deze functie zal staan in een FixedUpdate(). Dit is een functie die opgeroepen wordt aan de hand van elke stap die fysieke objecten maken in de game. In ons geval is dit de speler. Het script word gekoppeld aan de speler.

## 5.4 Een leeg object aan de speler koppelen

De speler heeft nu een VR-camera, controllers die ingelezen worden, kan vooruit bewegen en ook roteren. Tijdens het testen valt me op dat de speler redelijk stroef beweegt. Ik had geen idee hoe ik dit moest oplossen. Wouter Spaas schoot me te hulp. Hij liet me weten dat een object dat een ander object volgt veel zachter beweegt. Ik moest dus een object maken, deze koppelen aan de speler en dan het object de bewegingen laten uitvoeren. Ik maakte het lege object aan, koppelde

deze aan de speler en koppelde het script dat zorgt voor bewegingen aan het lege object.



*De camera volgt een leeg object*

Om de speler naar het lege object te laten kijken en mooi te laten roteren in de richting van de beweging van het lege object moest ik mijn bewegingsscript aanpassen.

```
//The player object tracks the movement of the chosen target (target.position)
//The player object follows this movement with a speed assigned as Damping (with delay)
transform.position = Vector3.Lerp(transform.position, target.position, positionDamping * Time.deltaTime);

//Makes the player object look at the chosen target and follow it smoothly
var rotation = Quaternion.LookRotation(target.position - transform.position);
transform.rotation = Quaternion.Slerp(transform.rotation, rotation, Time.deltaTime * positionDamping);
```

*Aanpassing in het bewegingsscript*

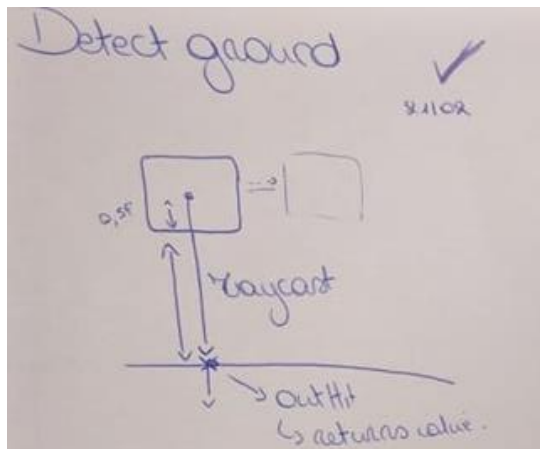
De eerste regel (niet //) van de hierboven geselecteerde code zorgt er voor dat er een nieuwe variabele aan gemaakt word die de waarde van de positie van je target (gevolgde object) verminderd met de waarde van jouw positie om een vaste minimum afstand van jou en je gevolgde object te berekenen. De regel daar onder geeft aan dat jouw rotatie mee gaat met de rotatie van het object met de nodige vertraging (positionDamping). Deze code kreeg ik cadeau van Wouter Spaas. De Quaternion zal er voor zorgen dat het object waar het script aan vast zit heel zacht zal gaan roteren rond de euler angles.

## 5.5 De grond detecteren

Om de grond of andere oppervakken te detecteren moet je gebruik maken van een Raycast. Deze lost het probleem op dat de speler eerst door de grond zou zakken. Dankzij de Raycast word er gedetecteerd dat de speler op een voorwerp staat.

### 5.5.1 De werking van een Raycast

Een Raycast is een functie die je kan gebruiken in je Unity scripts. Deze stuurt een straal vanuit het middelpunt van het object waar het script aan hangt, in de richting die je zelf kan aangeven in de functie. De straal detecteert of er objecten met colliders in de buurt zijn. Een collider zorgt voor botsingen tussen objecten. Als er een collider gedetecteerd wordt, stuurt de Raycast een waarde 'true' terug. Indien er geen collider gedetecteerd wordt, stuurt deze 'false' terug.



Schets van een raycast

In onderstaand script wordt er gekeken of de collider die geraakt wordt de tag "Ground" heeft. Indien dit zo is zal er aangegeven worden dat de speler op de grond staat. Deze zal dan niet meer bewegen. Indien de Raycast een object met een andere tag detecteert, zal de speler geen actie ondernemen en gewoon door doen met de actie die uitgevoerd werd.

```
//declaration raycast (detects if there is a collision)
RaycastHit hit;
//The raycast starts at the position of the player object
//The raycast casts a ray downwards and calculates if there has been a collision
//The raycast sends a value out to that is true when there is a collision and false when there is not
if (Physics.Raycast(transform.position, Vector3.down, out hit, distanceToGround + 0.1f))
{
    //if the gameObject with the tag "Ground" is hit, it returns that the player is on the ground
    if (hit.transform.gameObject.tag == "Ground")
    {
        playerBehaviour.isGrounded = true;
    }
    //else it returns that the player is not on the ground
    else
    {
        playerBehaviour.isGrounded = false;
    }
}
```

Script: Raycast

In bovenstaand script wordt er met de raycast functie een straal naar beneden gestuurd. Als deze straal in aanraking komt met een collider van een object wordt de naam of de 'tag' doorgegeven aan de 'hit'. Dan wordt er in de if-statements gekeken of de tag die de straal opnam gelijk is aan de "ground". Als dit zo is wordt de waarde 'isgrounded' gelijk gezet aan true. Dit betekent dat in het Playerbehaviour script de beweging gestopt wordt.

## 5.6 Virtual Reality headset traceren

Om de beweging van de speler te traceren moet de headset de effectieve speler voorstellen. Er moet gedetecteerd worden wanneer de speler van het platform af springt. Dit wordt eveneens gedaan met het eerder vermelde Raycast script. Het script wordt gehangen aan de 'head camera'. Deze stelt dus de speler voor.

Er word nu gedetecteerd wanneer de speler geen grond meer detecteert via de Raycast. Als de grond niet meer gedetecteerd wordt dan wordt de movespeed terug op de waarde gezet waar deze eerst gelijk aan was en zal de speler gaan bewegen.

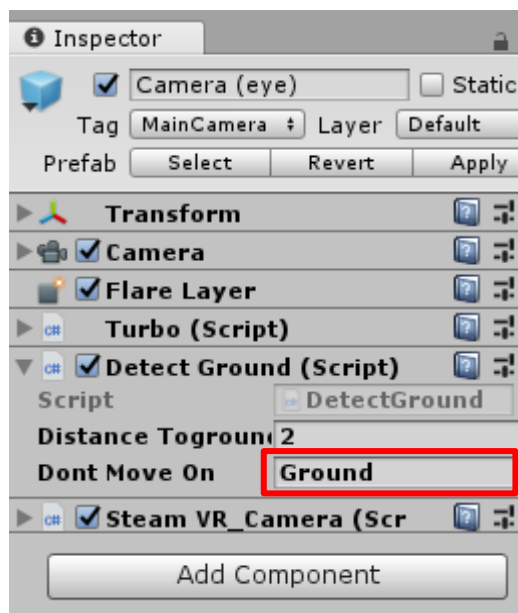
## 5.7 De speler kantelt door platformen

Met camera die als speler gezien word, kantelt de speler nu door het landplatform als deze hier op landt. Dit werd opgelost door een kleine aanpassing te maken in het script dat de grond detecteert. In de plaats van dat de tag van het object waar de speler op landt in de code gezet wordt, wordt deze vervangen door een string.

```
//if the gameobject with the tag "Ground" of "RubberBoeat is hit, it returns that the player is on the ground
// player stops moving
if (hit.transform.gameObject.tag == DontMoveOn)
{
    playerBehaviour.isGrounded = true;
    //playerBehaviour.moveSpeed = 0;
    transform.Translate(new Vector3(0, 0, 0));
}
```

*Schript: De grond wordt gedetecteetrd en de speler stopt met bewegen*

Dit een object met als waarde een **tekst**. De tekst kan in de inspector veranderd worden. Deze kan op elk moment aangepast worden door een andere waarde in het rood gemarkeerde veld in te geven. Het script zal kijken of de tag van het object waar de speler mee botst gelijk is aan de waarde in het gemarkeerde vak.



*De DontMoveOn tag kan aangepast worden*

## 5.8 Turbo functie

Nu de speler goed zacht kan bewegen, de headset gedetecteerd word en de oppervlakken waar je op land gedetecteerd worden is het tijd om een extra functie toe te voegen. Dit doe je aan de hand van een script. Het doel van het onderstaande script is het versnellen in de richting van de controller waar de trigger



van ingedrukt word. Indien beide controllers de trigger indrukken versnelt de speler naar voor.

In onderstaande code staat er bij elke lijn uitgelegd wat er gedaan word.

```
PlayerBehaviour playerBehaviour;
public float extraGravity;
public float extraMoveSpeed;

private bool isRightTriggered;
private bool isLeftTriggered;
//playerBehaviourscript gets called
private void Awake()
{
    playerBehaviour = FindObjectOfType<PlayerBehaviour>();
}
//procedure returns boolean and the type of controller that has been triggered
//if the controller that had been used is left, istriggerleft = true, same counts for right
public void ControllerTriggered(bool isTriggered, ControllerType controllerType)
{
    if (controllerType == ControllerType.Left){
        isLeftTriggered = isTriggered;
    }
    else if (controllerType == ControllerType.Right){
        isRightTriggered = isTriggered;
    }
}
private void Update()
{
    //if controller trigger left and right are pressed --> 2 * extra speed forward (turbo)
    //if controller trigger left and right are pressed --> 2 * extra gravity (downward)
    if (isLeftTriggered && isRightTriggered){
        playerBehaviour.extraGravity = extraGravity * 2;
        playerBehaviour.extraMoveSpeed = extraMoveSpeed * 2;
    }
    //if left OR right trigger is pressed --> extra speed forward (faster in 1 direction)
    //if left OR right trigger is pressed --> extra gravity (downward)
    else if (isLeftTriggered || isRightTriggered){
        playerBehaviour.extraGravity = extraGravity;
        playerBehaviour.extraMoveSpeed = extraMoveSpeed;
    }
    else{
        //no extra speed or gravity
        playerBehaviour.extraGravity = 0;
        playerBehaviour.extraMoveSpeed = 0;
    }
}
```

Script: De turbo functie

## 5.9 Enums

Om bovenstaand script te kunnen gebruiken is er eerst nog een extra script nodig. Een enum script. Enums kan je ook in je hoofdprogramma gebruiken, deze moet je dan bovenaan in je programma declareren. Maar het was makkelijker om deze in een apart script te zetten. In een enum script kan je alle mogelijke waarden die aangeroepen kunnen worden declareren. In ons geval stellen we een enum script op om na te gaan welke controller we gebruiken. In

```
public enum ControllerType
{
    Left = 0,
    Right = 1
}
```

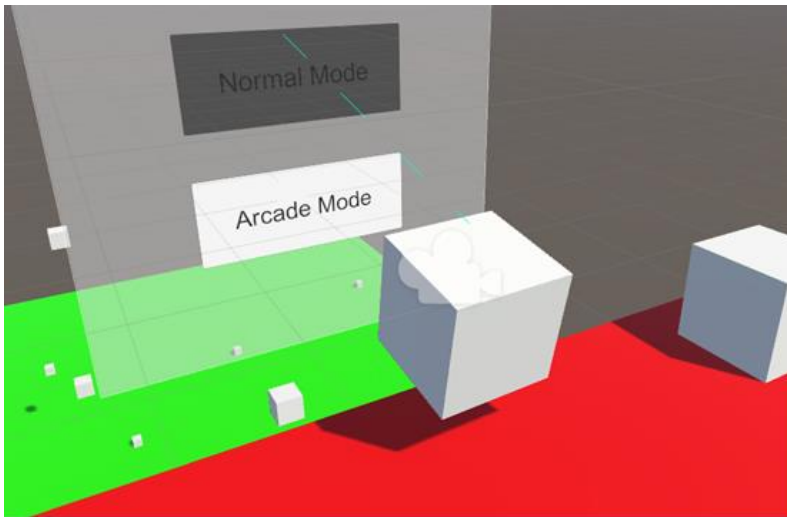
Script: Enum



bovenstaand script roepen we ons enum script op in een procedure en erna in een Update(). Een Update() word elke frame uitgevoerd dus elke frame zal gekeken worden of de controllers ingedrukt worden.

## 5.10 Het menu

Een game moet natuurlijk ook beschikken over een menu. In het menu zou de speler eerst de keuze krijgen om een gamemode te kiezen. Maar twee gamemodes uitwerken in deze korte stageperiode was niet haalbaar. Hier onder kun je het prototype van het menu zien.



*Voorbeeld van het menu zonder uitgebreide interface*

Het menu bestaat uit een canvas met daarop een paneel. Op dit paneel staan twee knoppen. De knoppen kunnen aangeklikt worden met de SteamVR Laserpointers. Deze kan je gebruiken dankzij de SteamVR die eerder gedownload is uit de UnityAssetstore. Eenmaal je SteamVR hebt gedownload uit de Assetstore krijg je de optie deze te importeren naar je project waarna je een map vind genaamd SteamVR. Hierin vind je alle functies terug die je nodig hebt in je VR-game. Je gaat opzoek naar de Laserpointer-scripts en deze koppel je aan beide controllers.

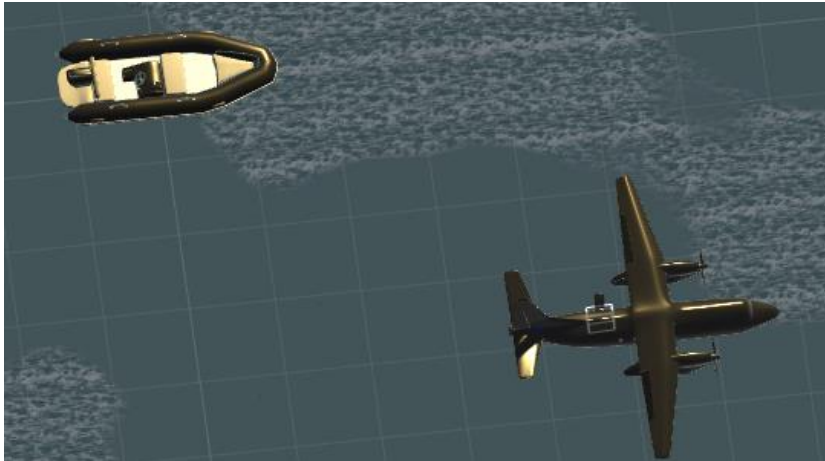
Als er op een knop geklikt word, zal het canvas verdwijnen en de laserpointer uitgeschakeld worden zodat de speler kan beginnen met het spel. Dit wordt aan de hand van de onderstaande gemarkeerde code gedaan.

```
public class ModeSelect : MonoBehaviour
{
    public Button btnMode;
    public Canvas cnv;
    private void Start()
    {
        GetComponent<SteamVR_LaserPointer>().enabled = true;
    }
    public void Click(string mode)
    {
        Debug.Log("Selected game mode: " + mode);
        cnv.gameObject.SetActive(false);
        GetComponent<SteamVR_LaserPointer>().enabled = false;
    }
}
```

*Script: de laser wordt uitgezet wanneer er een knop gebruikt word.*

## 5.11 Het design

Nu de bewegingen volledig af waren en je de keuze had tussen twee gamemodes, kon ik de ontwerpen van Gialuca (Functie: 3D-ontwerp) toevoegen in het project. Ik kon de designs gewoon kopiëren en in de game plakken. Door de designs direct te plakken in de hiërarchie van de game, bleef de plaats hetzelfde dan waar Gianluca de objecten geplaatst had. In de plaats van dat de speler begint op een rood oppervlak begint deze nu in een vliegtuig. In de plaats van dat de speler landt op een groen platform, kan deze landen op een bootje. Er werd ook een zee toegevoegd. Het resultaat van deze toevoegingen zie je hier onder.



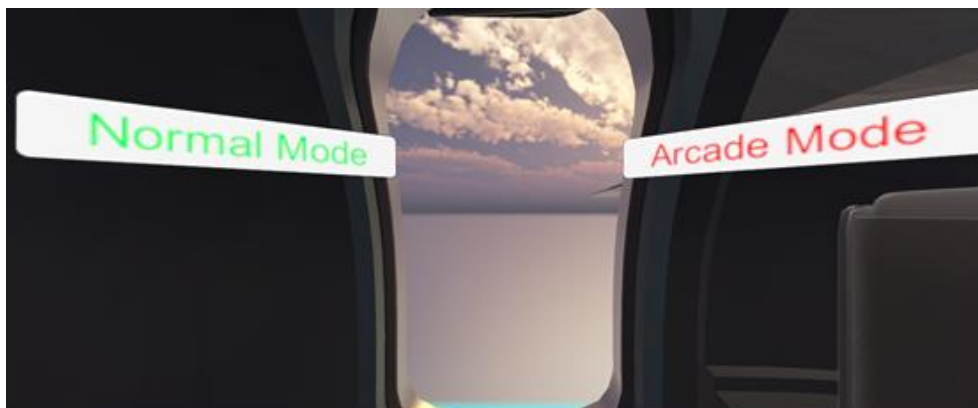
*Het design van de game*

Een kijkje binnenin het vliegtuig (Dit kan de speler ook zien voordat hij uit het vliegtuig springt):



*De binnenkant van het vliegtuig*

Het menu werd ook aangepast:



*Het menu met interface*

## 5.12 Script ringen

In de arcade mode moet je doorheen ringen vliegen, deze worden opgeteld en de score wordt afgebeeld op je scherm als je landt op de rubberen boot. De ringen worden manueel in de game gezet. Je kan de ringen plaatsen op je eigen gewenste manier.

In onderstaand script wordt er gedetecteerd wanneer de collider van de speler (met de tag 'player') in contact komt met de collider van de ring. Indien dit zo is, word de ringCount plus één gedaan. De ringCount staat gedeclareerd in het playerBehaviour script. Dit is het script dat voor de beweging van de speler zorgt. De ringCount staat niet gedeclareerd in het onderstaande script omdat dit script word gekoppeld aan elke ring. Als deze aan elke ring hangt, zal elke ring zichzelf op tellen en niet hoger dan 1 geraken. Als deze vanuit een ander script word aangeroepen zal elke ring opgeteld worden en word de waarde in het andere script bijgehouden.

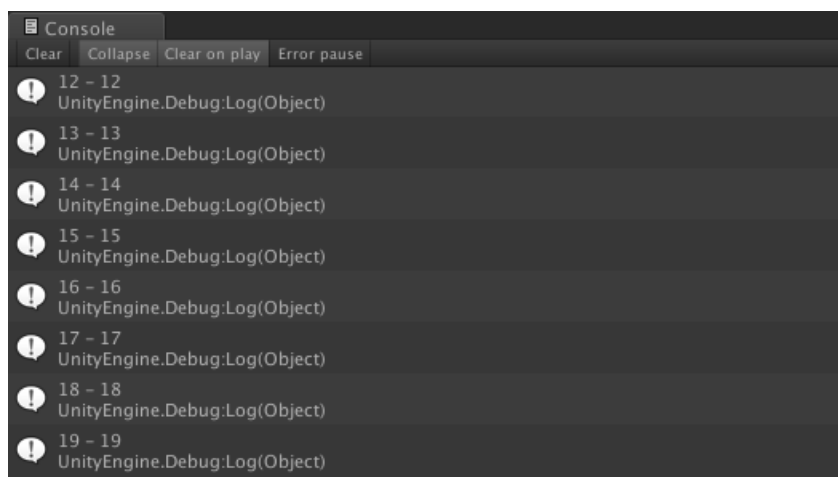
```
public class ItemPickUp : MonoBehaviour
{
    PlayerBehaviour playerBehaviour;

    // Use this for initialization
    void Start()
    {
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();
    }

    void OnTriggerEnter(Collider other)
    {
        //Debug.Log("1 . CollisionRing");
        if (other.GetComponent<Collider>().tag == "Player")
        {
            Debug.Log("2: CollisionRing");
            playerBehaviour.ringCount++;
            Debug.Log("+ 1");
            Destroy(gameObject);
            Debug.Log("Destroy");
        }
    }
}
```

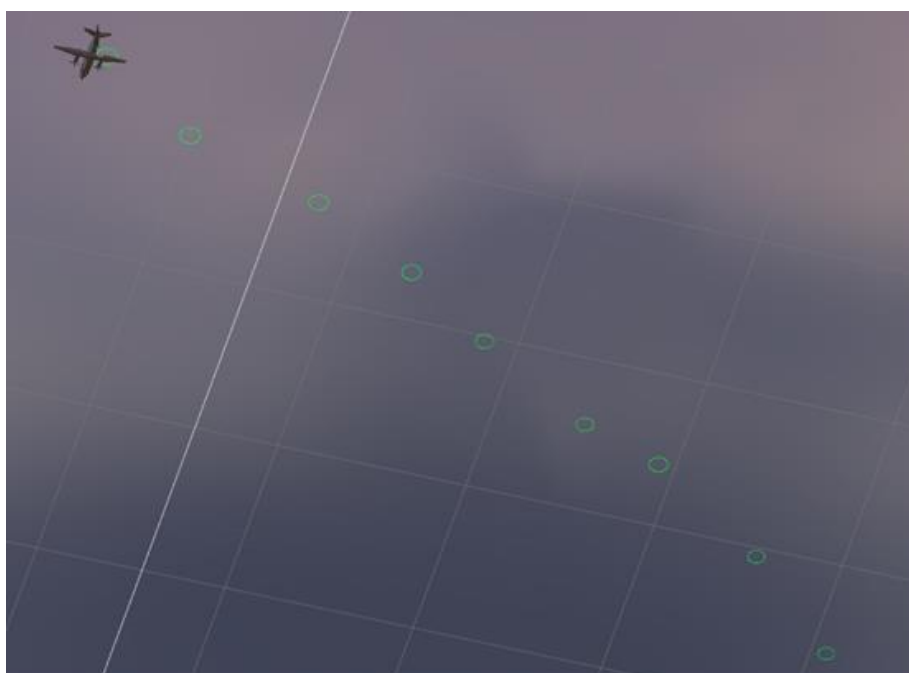
*Script: Er word gedetecteerd of de speler door een ring gaat*

Een `Debug.Log("");` gebruik je om te testen of alles uitgevoerd word. Zodra de code uitgevoerd word, verschijnt de debug in de console window van Unity.



*Debug*

Om dit script uit te testen plaats je de ringen (groen) op een rechte lijn zoals het voorbeeld hieronder.



*Voorbeeld van de ringen in de game*

## 5.13 Een realistische zee

Om een realistische zee te kunnen gebruiken kocht Firewolf Studios een zee in de Unity AssetStore. In de Unity AssetStore vind je 3D-ontwerpen, scripts, opmaak, etc. Je hebt zowel gratis 'assets' als betalende 'assets' in de AssetStore.

Helaas is deze zee enkel bruikbaar in een nieuwere versie van Unity. We zijn dus overgeschakeld naar een nieuwere versie. Hier bleek dat de oudere SteamVR software die we gebruikte niet compatibel was met deze nieuwere Unity versie. We moesten dus gebruik maken van de nieuwere SteamVR software. Hier had ik

eerder in mijn stage problemen mee, vandaar de terug schakeling naar de oudere versie. Na een volledige dag opzoekwerk en hulp van collega's werd er beslist om de zee niet te gebruiken en terug te schakelen naar de vorige Unity en SteamVR versie. Deze dag was leerrijk voor mij. Ik heb die dag niet veel programmeerwerk gedaan, maar veel opgezocht. Dit heeft mij veel kennis bijgebracht.

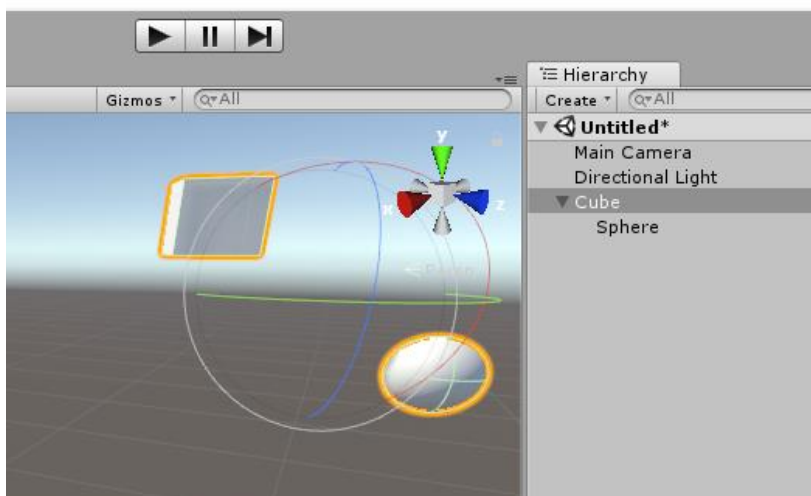
## 5.14 Aanpassen van de bewegingen

De bewegingen zijn helaas nog niet perfect. Dit kan aangepast worden door met de nummers in de scripts te spelen. De draaisnelheid, beweegsnelheid en de zwaartekracht van de speler werd aangepast zodat de bewegingen veel vlotter en realistischer waren voor de speler.

## 5.15 Positie van de ringen

De ringen zet je telkens op de juiste plaats zodat de speler hier net aan kan. Dit is echter niet altijd even makkelijk. Bij mij stonden de ringen te kort op elkaar. Dit moest aangepast worden. De speler moest makkelijk bij de ring geraken, maar hij zou ook bij volgende ringen moeten geraken als hij er ééntje miste. Dit was niet haalbaar bij mijn opstelling van de ringen. Na even uitzoeken hoe ik dit kon oplossen schoot Wouter mij te hulp. Hij begon uit te leggen hoe ik dit het beste aanpakte. Eerst snapte ik zijn uitleg niet, maar na er even over na te denken snapte ik het. Je moet de 'parent' van de ring (dit is een leeg object) op de startpositie van de speler zetten. Hierna test je de perfecte afstand van de speler tot de ring zodat de ring niet te hoog of te laag hangt. De speler moet niet constant de turbo functie gebruiken om een ring te kunnen halen. Als de perfecte positie is gevonden kopieer je de volledige ring (inclusief de parent). Deze plak je opnieuw in de game. Nu zet je de lege parent op de positie van de vorige ring. Hierdoor is de afstand tussen de twee ringen hetzelfde als de startpositie en de ring. De ring kan nog geroteerd worden zodat je niet in een rechte lijn naar beneden gaat.

Hieronder vind je een screenshot van hoe dit er uit ziet met normale objecten.



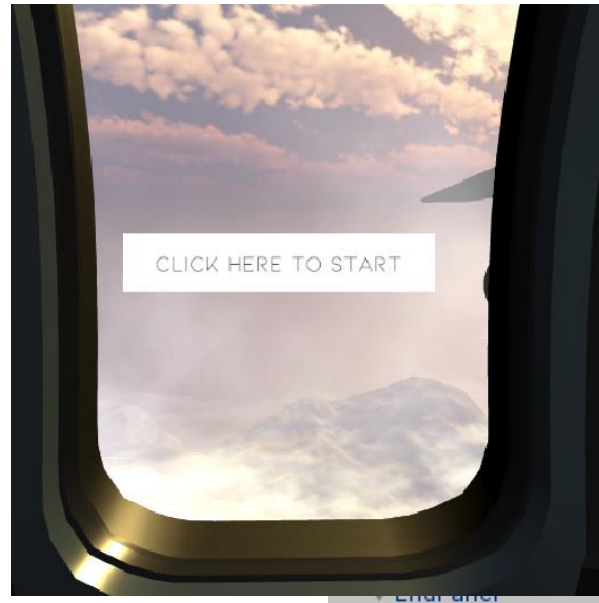
*2 objecten die aan elkaar vast hangen en bewegen met elkaar*

## 5.16 Het volledige menu

Het menu hangt vast aan de head camera. Zoals eerder vermeld stelt deze de VR-bril voor. Dit betekent dat het menu altijd voor de speler zal verschijnen.

Als de speler op de startknop drukt (met de laserpointers) zal het spel starten en kan de speler van het platform af springen. Als er niet op start gedrukt word, zal de speler niet kunnen springen.

Aan het menu canvas zijn drie panelen gevestigd. Één paneel dat enkel zichtbaar is wanneer je start (het startpaneel). Één paneel dat enkel zichtbaar word als je op het bootje land. Hierop staan twee knoppen. Een cancel knop die je de game laat afsluiten en een try again knop die je terug naar het vliegtuig stuurt. Je score (aantal ringen waar je door ging) word eveneens zichtbaar. Het laatste paneel is enkel zichtbaar als je op het water of de bergen land. Hier krijg je de melding dat de missie niet geslaagd is.



Startknop op het menu



Indeling van het menu

De volgende scripts zijn aan de panelen gekoppeld.

```
public class EndPanel : MonoBehaviour {  
  
    public Text points;  
    PlayerBehaviour playerBehaviour;  
  
    //The panel is hidden when the game starts  
    //The PlayerBehaviour script is called  
    void Awake()  
    {  
        Hide();  
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();  
    }  
    //In this procedure the panel is set active  
    public void Show()  
    {  
        gameObject.SetActive(true);  
    }  
    //In this procedure the panel is hidden  
    public void Hide()  
    {  
        gameObject.SetActive(false);  
    }  
    //The collected rings are put into a text  
    //This text is assigned to a public textbox (on the panel)  
    public void RingText()  
    {  
        points.text = " You have collected " + playerBehaviour.ringCount + " rings.";  
    }  
}
```

Script: Menu dat gezien word als je land op het bootje

```

public class StartPanel : MonoBehaviour {

    //The panel is hidden when the game starts
    void Awake()
    {
        Show();
    }

    //In this procedure the panel is set active
    public void Show()
    {
        gameObject.SetActive(true);
    }

    //In this procedure the panel is hidden
    public void Hide()
    {
        gameObject.SetActive(false);
    }
}

```

*Script: Menu dat gezien word als je de game start*

```

public class DeadPanel : MonoBehaviour {

    //The panel is hidden when the game starts
    void Awake()
    {
        Hide();
    }

    //In this procedure the panel is set active
    public void Show()
    {
        gameObject.SetActive(true);
    }

    //In this procedure the panel is hidden
    public void Hide()
    {
        gameObject.SetActive(false);
    }
}

```

*Script: Menu dat gezien word als je niet op het bootje land*

Aan het menu canvas zit onderstaand script gekoppeld, Hier worden de functies aan de knoppen gegeven.

```

public class ModeSelect : MonoBehaviour
{
    public Canvas cnv;
    public StartPanel startPanel;
    public DeadPanel deadPanel;
    public EndPanel endPanel;

    PlayerBehaviour playerBehaviour;
    SteamVR_LaserPointer laserPointer;

    VRUIInput laser;

    //The start panel is active when the game starts
    //The playerbehaviour script is called
    void Start()
    {
        startPanel.Show();
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();
    }

    //When the start button is clicked the panel is hidden
    //ClickedStart = true, this means the player is able to move
    //this function is used in the PlayerBehaviour script
    public void IsClickedStart(string mode)
    {
        startPanel.Hide();
        playerBehaviour.isClickedstart = true;
    }

    //when the try again button is pressed the scene is reloaded
    //The ring count falls back to 0
    public void IsClickedTryAgain()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        playerBehaviour.ringCount = 0;
    }

    //When the cancel button is clicked you exit the game
    public void IsClickedCancel()
    {
        Application.Quit();
    }
}

```

*Script: Alle functies die je kan aanroepen in de menu scripten*



De functies worden aangeroepen in de inspector. Het mode select script word aangeroepen, met de functie `IsClickedStart` (uit het vorige script). Als er op de knop gedrukt word, zal de `IsClickedStart` functie uitgevoerd worden. In dit geval betekend dit dat het paneel waar de knop op staat zal worden verborgen en de speler kan vanaf dan ook uit het vliegtuig springen.



*De functie die nodig is als de knop ingedrukt is word aangeroepen*

## 5.17 Eindsituatie

Nu zijn alle acties, functies, procedures en procedures toegepast. De ringen hangen op zijn plaats. Het aantal ringen word opgeteld. De speler kan zijn parachute besturen om door de lucht te vliegen. De speler kan landen op het daarvoor voorziene bootje. Hij krijgt de keuze om opnieuw te beginnen of het spel af te sluiten. De punten worden getoond. De speler kan ook landen op het terrein of het water en krijgt dan de keuze om opnieuw te beginnen of het spel af te sluiten.

Voor een blik op het eindproduct, verwijs ik u verder naar bijlage nummer 1, 2 en 3.



## 6 Evaluatie van de leerervaringen

### 6.1 Wat heb ik geleerd?

Tijdens mijn stageperiode heb ik geleerd hoe het werkt om in een game bedrijf mee te draaien. Dit was het oorspronkelijke doel van mijn stage. Maar ik heb veel meer geleerd dan alleen dat. Ik heb geleerd dat er veel mogelijkheden zijn om samen te werken aan een project in een bedrijf.

Firewolf Studios maakt gebruik van het programma SourceTree om dit te verwezelijken. In dit programma kan je werken op verschillende versies van je project, en als je klaar bent met wat je gedaan hebt kan je het opslaan op het eigenlijke project. Hierdoor kan er aan verschillende delen van het project tegelijk gewerkt worden. Mijn opdracht was dus een VR game ontwikkelen in de game engine Unity. Ik heb hier de volledige twee weken van de stage aan gewerkt. Elke dag besloot ik bij aankomst welke functies ik die dag zou gaan programmeren. Dit deed ik aan de hand van enkele schetsen waarmee ik duidelijk kon zien wat ik die dag moest doen. Het ontwerp van de game werd gedaan door Gianluca Sorrentino. Hij is de 3D-ontwerper van Firewolf Studios. Verder kreeg ik hulp van Wouter en Frank als ik vast zat.

Ik ben verschillende problemen tegen gekomen tijdens mijn stage. Één van deze problemen was de werking van de controllers van een VR-bril in Unity. Ik heb veel onderzoek moeten doen naar de werking van de controllers samen met de nieuwe SteamVR (software die er voor zorgt dat je Unity aan de VR-bril kan linken) software. Ik zocht het hele internet af naar video's, handleidingen en andere handige artikels die mij verder konden helpen in verband met de controllers. Maar hier was heel weinig documentatie over, waardoor ik weinig tot geen informatiebronnen had om mij op te baseren. Dit is uiteindelijk opgelost geraakt door een oudere versie van de SteamVR software te downloaden. Het uitzoeken van de werking van de nieuwe software zou te veel tijd in beslag nemen. De stage was immers maar twee weken. Mijn andere problemen heb ik allemaal kunnen oplossen met enkele scripts. Deze zijn samen met de uitleg terug te vinden in mijn technisch verslag.

### 6.2 Hoe zou ik het een tweede keer aanpakken?

Als ik deze opdracht opnieuw zou moeten uitvoeren zou ik direct anders beginnen. Ten eerste zou ik het mezelf al wat makkelijker maken door alle objecten in het midden van de game te plaatsen (op positie (0,0,0)). In mijn project heb ik namelijk de objecten op posities gezet die niet zo voor de hand liggend waren. Een voorbeeld hiervan is bijvoorbeeld positie (12.6, 150, 154). Dit is niet handig omdat posities zoals deze het moeilijker maken om andere objecten op een juiste plaats te zetten. Vervolgens zou de benaming van mijn scripts veranderd worden. De werkvolgorde zou ik ook veranderen. Ik zou al de bewegingen en acties al programmeren als het ontwerp nog niet toegevoegd is aan de game. Hierdoor zou ik een veel beter overzicht krijgen over de game en de inhoud van mijn scripts. Ik zou veel gestructureerde gaan werken. Het eindproduct kan misschien wel hetzelfde zijn, maar de game moet overzichtelijk genoeg zijn om hem makkelijk uit te breiden.

### **6.3 De toepassing.**

Er is nog geen concreet idee voor het gebruiken van mijn toepassing. Één van de mogelijkheden is het verkopen aan een bedrijf (Dit is niet zeker dus de naam van het bedrijf kan ik nog niet doorgeven). Dit bedrijf bied leuke ervaringen aan voor klanten. Enkele hiervan zijn een diner in de lucht, speleoboxen, een klim muur, en nog enkele extreme games. Mijn parachute ervaring zou daar gebruikt gaan worden om mensen een virtuele parachute sprong te laten beleven. Een andere mogelijkheid is dat de game op een beurs getoond zou gaan worden om daar potentiële klanten een VR ervaring te geven. Geen van deze mogelijkheden is officieel.

## 7 Stageverslag

In het kader van mijn gip kreeg ik tijdens de laatste twee weken van februari (2019-02-18 – 2019-03-01) de kans om stage te lopen bij het bedrijf Firewolf Studios. Firewolf Studios is een software bedrijf gevestigd op de Corda Campus in Diepenbeek. Ze houden zich bezig met het ontwikkelen van (leerrijke) games op aanvraag van klanten. In het bedrijf werd ik begeleid door zowat iedereen. Ik werd vooral begeleid door Wouter Spaas (technical artist en programmeur) en Frank Van Geirt (programmeur en co eigenaar van Firewolf Studios). Voor mijn stageopdracht werkte ik samen met Gianluca Sorrentino, hij zorgde voor het uiterlijk van de game (3D-ontwerp).

Op mijn stage kreeg ik de mogelijkheid om als stageopdracht een Virtual Reality game te ontwikkelen. Een Virtual Reality game is een game waar je (zoals de naam het zegt) een virtuele realiteit binnenstapt. Dit is mogelijk gemaakt door een VR-bril. Als je een VR-bril op hebt kun je de virtuele wereld rondom je zien en hierin (afhankelijk van het spel dat je speelt) rondlopen, vechten, ... Het thema van de game werd beslist door een kleine vergadering die plaatsvond wanneer mijn stage begon. Ik had voor mezelf al een klein idee wat ik wilde doen met de game. Ik stelde dit voor aan mijn stagebegeleider en hij was direct aan boord met mijn idee. Dit gold ook voor de rest van het Firewolf team. Het concept van de game zou een parachute simulator worden. Je springt vanuit een vlieger op 1200 meter hoogte naar beneden met een parachute. Je moet je navigeren doorheen ringen die gepositioneerd staan in de lucht, en hierna landen op een bootje in een revier in een dal met bergen. Het doel van de game is een stapje dichterbij het overwinnen van hoogtevrees. Dit kwam goed uit voor mij omdat ik enorm last heb van hoogtevrees. Na die vergadering kon ik beginnen aan de opdracht. Mijn opdracht bestond uit het programmeren van de verschillende functies die de speler kon doen. De functies bestonden onderanderen uit het besturen van de parachute (rond glijden doorheen de lucht), doorheen ringen gaan tijdens het afdalen, ... De game zal gemaakt worden in de game engine Unity. Unity is een programma waarin je games kan ontwikkelen.

In de eerste paar dagen hield ik me bezig met het samenstellen van enkele scripts (instructies die je kan koppelen aan objecten in de game) die er voor zorgen dat de speler met zachte bewegingen kon navigeren doorheen de lucht. Dit deed ik door een leeg object aan de speler te koppelen. Aan het leeg object worden de scripts die zorgen voor de beweging gekoppeld. De rede hiervan was dat de beweging telkens hikkelig uitgevoerd word door het object dat voor de beweging uitvoert. Door de speler aan een leeg object die de beweging uitvoert te koppelen volgt de speler het leeg object. Waardoor de beweging dan zacht verloopt.

Hier kwam ik het eerste probleem al tegen. Ik moest namelijk in de scripts die voor beweging zorgen, de HTC VIVE (de VR-bril die we gebruikte) controllers hun functie geven. Dit liep moeilijk omdat de software wat we hiervoor gebruikte (SteamVR) onlangs geüpdatet was. Omdat de nieuwe versie nog niet lang beschikbaar was, was het documentatiemateriaal hierover ook beperkt. Na een hele dag uitpluizen hoe dit werkte zonder resultaat besloten we een oudere versie van SteamVR te gebruiken. Hier was genoeg documentatiemateriaal over te vinden. Om bovenstaande scripts te perfectioneren heb ik van maandag tot

woensdag nodig gehad. In de loop van de rest van de stage heb ik hier af en toe nog enkele aanpassingen in gemaakt.

Donderdag ondervond ik een tweede probleem. Namelijk dat mijn speler niet goed detecteerde of hij op de grond stond. Dit loste ik ook op met het opstellen van een script. In het script werd er gebruik gemaakt van een raycast. Een raycast stuurt een straal vanuit het midden van het object waar het script aan gekoppeld is, naar de gewenste richting (deze kun je aangeven in het script). Deze straal detecteert dan of er objecten in de weg staan en verstuurt deze waarde terug. Zo kon de speler detecteren of deze op de grond stond. Hier werden dan ook enkele voorwaarden aan toegevoegd. Als de speler (met het script) een object detecteert, zal deze hier op blijven staan. De speler zal ook stoppen met bewegen zodra de speler op het object staat. Dezelfde dag kwam ik nog een tweede probleem tegen. Zonder duidelijke aanleiding werden de controllers van de VR-bril niet meer gevonden door de game. De speler kon dus niet bewegen. Na heel wat zoekwerk naar mogelijke oorzaken zonder resultaat besloot ik om de mogelijkheid tot het gebruiken van de pijltjes op het toetsenbord toe te voegen. Deze werden dan gebruikt om toch nog verder te kunnen werken aan de game. Miraculeus werden de controllers later terug gedetecteerd en kon ik deze terug gebruiken.

Vervolgens moest ik er voor zorgen dat je fysiek uit het vliegtuig kon stappen. Dit deed ik door het eerder vermelde script met de raycast aan de VR-camera te hangen. De VR-camera stelt de fysieke VR-headset voor, maar dan in de game. Door het script aan de camera te hangen, ging het script kijken wanneer de speler zelf uit het vliegtuig wandelt. Zo krijg je dus een echte spring ervaring.

Vrijdag was mijn taak het toevoegen van een “turbo functie” in het spel. Deze zorgt er voor dat als je op beide knopjes van de VR-controllers drukt, je sneller naar voor beweegt. Dit deed ik met een script dat detecteerde welke knop er wanneer ingedrukt werd. Hier maakte ik ook voor het eerst kennis met “enums”. Dit is een soort script waar je alle verschillende waardes in kan neertypen die je mogelijk zou nodig hebben voor een functie. In mijn geval was dit ‘Left’ en ‘right’. Left kreeg als waarde nul en right kreeg als waarde één. Het enum script wordt gebruikt in het script dat zorgt voor de “turbo functie”. Aan de hand hiervan kun je makkelijk aangeven welke controller er wanneer gebruikt wordt en welke acties er dus uitgevoerd moeten worden. In het technisch verslag volgt hier meer uitleg en documentatiemateriaal over. Na week één zijn de acties van de speler bijna volledig af.

De volgende maandag begon ik met het opstellen van een menu. Hier op stonden eerst twee knoppen. Dit komt omdat er normaal twee modes van de game zouden zijn. Eentje waar je gewoon naar beneden vliegt en eentje waar je doorheen de ringen moet navigeren. Uiteindelijk heb ik het gehouden bij één gamemode omdat er voor een tweede niet genoeg tijd was. Eens je met je controller een gamemode selecteert, verdwijnt het menu. Dit werkt goed. Door het opstellen van het menu leerde ik werken met de User Interface die ik kon gebruiken in Unity. Om dit succesvol te laten werken heb ik veel opzoekwerk moeten doen. Hieruit heb ik veel geleerd. Vervolgens kreeg ik te horen van Gianluca (3D-ontwerp) dat hij klaar was met het ontwerpen van het vliegtuig en een stuk water met hierop een boot. Deze kon ik nu dus in de game zetten. Hierdoor zag de game er al direct veel mooier uit.

Dinsdag bestond mijn taak uit het samenstellen van een script dat er voor zal zorgen dat het aantal ringen waar de speler doorvliegt opgeteld wordt. Dit was een

kort en eenvoudig script. De rest van de dag hadden we te maken met een groot probleem. Gianluca had een mooie zee in de Unity AssetStore gekocht. De AssetStore is een online winkel waar je veel dingen kan vinden die je kan gebruiken in Unity. Er zijn gratis dingen maar ook duurdere. De zee uit de AssetStore zorgde voor het probleem. Deze was namelijk niet bruikbaar in de versie van Unity waar we op dat moment gebruik van maakte. Er werd besloten om te veranderen van Unity versie. Hierdoor hadden we echter een groot probleem. De verouderde versie van SteamVR waar we op dat moment gebruik van maakte, kon niet gebruikt worden in de nieuwe versie van Unity. We moesten terug gebruik maken van de nieuwe versie van SteamVR.

Hierdoor hield de rest van mijn dinsdag veel opzoekwerk in. Helaas was dit opzoekwerk tevergeefs. Ik had de mogelijkheid om in een aangrenzend bedrijf hulp te vragen. Het bedrijf was fulltime bezig met het ontwerpen van VR-games en kon me hier dus zeker bij helpen. Één van hun softwareontwikkelaars hielp me met veel enthousiasme. Maar zelfs hij kreeg het probleem niet opgelost. Uiteindelijk hebben we dan op het einde van de dag besloten de zee uit de AssetStore niet te gebruiken en terug over te schakelen naar de oudere versie van Unity en SteamVR.

Woensdag was mijn voornaamste taak het positioneren van de ringen in de game. Hierdoor moest de speler navigeren. Het positioneren van de ringen nam veel tijd in beslag omdat ik er voor moest zorgen dat er genoeg plaats tussen de ringen was, maar ook dat de speler aan elke ring kon geraken. Aan elke ring werd een script gekoppeld. Dit script had ik dinsdag opgesteld. Nu kon ik het testen met de ringen. Ik merkte dat het aantal ringen waar de speler doorheen vloog niet opgeteld werden. Na even zoeken merkte ik dat de ringen niet getagd stonden als 'pickup'. Dit was de oorzaak van het probleem. Toen ik de tag veranderde werden de ringen wel opgeteld.

Donderdag hield ik me bezig met het opstellen van een menu dat je op het einde van het spel ziet als je op het bootje landt. En van een menu dat je ziet als je dood bent gegaan (als je bent geland op het water of op de bergen). Dit kon ik makkelijk doen met mijn UI-kennis van de afgelopen dagen. Jammer genoeg werkte de menu's niet. Hierbij kreeg ik hulp van Frank Van Geirt. Dit is de co-eigenaar van Firewolf Studios, samen met Stijn Mommen. Met het volledig opstellen van alle menu's was mijn volledige donderdag gevuld.

Vrijdag was mijn laatste stagedag, tot mijn spijt moest ik op het einde van deze dag afscheid nemen van mijn leuke stageplaats. Natuurlijk kwam ik 's ochtends op mijn stageplaats aan met een kleine verrassing als bedankje. Ik had namelijk de dag ervoor zelf brownies gemaakt, en deze waren ook nog eens super goed gelukt. Ik begon mijn dag met opmaken van de menu's in de game. Voor de rest werkte ik verschillende laatste details bij en kreeg ik van Gianluca het landschap doorgestuurd. Deze voegde ik toe in de game. Het zag er wonderbaarlijk uit. De rest van mijn dag bestond uit het aanpassen van kleine dingen in de game. Om 15:00 was mijn game klaar. Gianluca maakte nog enkele aanpassingen aan het uiterlijk van de game voordat we deze veranderde in een echte applicatie die je op de computer kan openen.

Firewolf Studios liet me echter niet vertrekken zonder enkele geschenken waaronder een MediaMarkt gift card en een VR-bril waar je je gsm in kan steken voor mobiele VR-games. Ze wouden me bedanken voor de aanwinst die ik in hun

team was. Ze lieten me ook weten dat als ik later hulp nodig heb met iets, ik zeker bij hun mocht langskomen.

Deze stage heeft me laten inzien dat ik thuishoor in de IT-sector. Dit betekent dat ik door het vuur zal gaan om later een job zoals deze te kunnen bemachtigen. Ik stond elke dag met een lach op, ging met een lach naar de stageplaats en kwam met een lach terug thuis. De stage heeft niet alleen maar invloed gehad op mijn opleiding maar ook op mijzelf als persoon.

Ik heb geleerd dat een programmeur zijn job bestaat uit veel opzoekwerk, elke dag bijleren, en enorm veel uittesten. De stage was heel nuttig voor me, en ik zal deze ervaring zeker meedragen in de rest van mijn leven.

## 8 Work Experience Project

### 8.1 The assignment

In the context of my GIP, I received an assignment at my internship. The assignment consisted of developing a Virtual Reality game. Virtual reality (as the name explains) gives you the opportunity to enter a virtual world. This can be in the form of an interactive game, a movie, or something else. To be able to play a Virtual Reality game you have to use a VR-headset (the screen that's in front of your eyes) and VR-Controllers.

Before the internship started, I sat together with Frank Van Geirt. Co-owner of Firewolf Studios. He explained that I was going to have to develop my own take on a game called "Richie's plank experience". The game is a Virtual Reality game where you have to walk across a small wooden plank on the top floor of a sky-high building. I was free to think of anything that I wanted. I took my time to think about it for a while and eventually I came up with the idea to develop a game where you have to jump out of an airplane. I was going to have to use the game engine Unity. This is a program that has numerous possibilities when it comes to designing a game.

I have worked with Unity a year ago so I was familiar with the software. This gave me a small advantage to prepare for my assignment. My knowledge of C# (a programming language) was also at my advantage. All the code that I would have to develop was going to be written in this programming language.

My goal to reach before starting my internship, was being able to write small scripts (more explanation about this in the next subsection). Another thing I really wanted to accomplish was making one or two small games as a preparation for my internship.

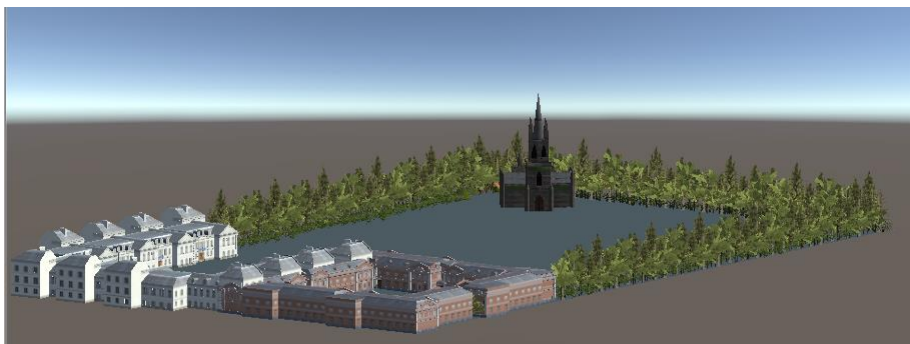
### 8.2 Preparation

To prepare myself for my assignment, I had to watch a lot of small video courses on how to script in Unity. I had to learn how to develop scripts to use in my game. By scripting, you add actions or give instructions to the object that the script is attached to. Scripts are responsible for actions, movements, and all other kinds of things inside of the game. Visualize that you have to pick up an item when playing a game by using a certain button. This is most likely done by a script that says "when you press this button, do this". I learned how to make scripts by watching a lot of videos on the official Unity website. They have got a whole page full of script lessons. Each lesson is a couple minutes long. Some of them are also interactive, so you can actually write code together with the host of the video, and see how your code is reacting when you write it. Every line of code is explained briefly. I learned a lot by watching these videos. I also followed some tutorials (also listed on the official Unity website). These tutorials consisted of the basic 'roll a ball' tutorial, where you learn the basics of Unity, and the 3D gamekit tutorial. The 3D gamekit tutorial was kind of a letdown, because it consisted of downloading an asset pack in the Unity AssetStore (an online website where you can get everything and more to use in your game). After downloading and unpacking the gamekit, the game was already fully built. There was no challenge in this at all.

As a further preparation I created two games. I started with a small 2D game. This game consisted of a small character (in the form of a potato), and a couple of automatically spawning platforms for the player to jump on. When I was developing this small game I had to put my new knowledge to the test. I learned a lot from it. During this learning process I had to look up every error and every mistake that I encountered.



After some thinking I realized that I'd better start with a 3D game instead of a 2D game. The reason for this was because I had to develop a Virtual Reality game, and this is a 3D environment. I put the 2D project on hold. I started a small "Starting level" for an open world game. The level consisted of a character, some enemies, a couple of quests and items to pick up.



I haven't completed this game because this was a way to use a lot of functions in one project. It is very messy and is clearly a test project. I am planning to restart this project from scratch, write down a full plot for the game and then I will be developing the game again, using the knowledge I have picked up at my internship.



This game has been a good preparation for my assignment. Both of these games were made during this schoolyear.

### **8.3 The internship**

During my internship I was able to develop my game thanks to my preparation and the help from my co-workers. At my first day at the internship, we discussed my idea about jumping out of an airplane. The idea was approved, but slightly adapted. Instead of just jumping out of the airplane into the water, the player jumps out of the airplane, and navigates himself (by using the controllers) through the air to collect rings and land on a boat in the water. When the player lands it gets to see his score (count of rings), and has the option to try again or cancel the game. I didn't think that I was going to be able to complete the game, but I was. At the end of the internship I was able to deliver a game where the player can jump out of the airplane after pressing the start button. The player can glide smoothly through the air and is able to collect rings by flying through them. Last of all the previous options are displayed in front of the player when landing on the boat. When the player lands into the water, or the surrounding terrain, the screen turns black. A message with "mission failed" is displayed together with the option to restart the game or quit it.

### **8.4 Evaluation**

I am really happy with the end result. I was able to realize more than I initially thought I would be able to. This is all thanks to my preparation, the help that I got from my co-workers during the internship and a lot of research before and during the internship. I was able to solve a lot of problems by googling. Google is the main information source for a programmer because everything you need is accessible via google. The only requirement is that you know what you're looking for and how you have to type it in your sentence to find exactly what you need.. There were also some things that I couldn't find that easily by googling it. Those are the problems that my co-workers assisted me with. They almost never gave me the answer, but they pushed me into the right way to find it myself. I am utterly grateful for that, because by doing a lot of research you automatically learn a lot of things. You may not need all of those things right now, but some day you will and then you know how to look for it.

## 9 Fiche entreprise

Nom	Firewolf Studios
Forme juridique	Société anonyme
Date de fondation	2012
Organigramme	En annexe (4)
Maitre de stage	Frank Van Geirt
Siège social	Hasselt
Adresse	Corda campus bâtiment 6 grand salle A
Accessibilité	Accessible par le transport public Accessible par la route de Kempische steenweg
Services	Des jeux Animations Visualisation
Concurrence	Lugus Studios Bazookas Curious Cats
Principaux fournisseurs	La société n'a pas de fournisseurs
Entendue du marché	L'amusement
Filiales	1
Chiffre d'affaires	Pas disponible
Nombre de salariés	6 ( + Stagiaires)
Slogan	"Level up your business"

**Logo**



**Espace de travail**



## 10 Zelfevaluatie

### Self-Evaluation Form for Employees

Employee Name: Boelen Evi

Job Title or Position: Game Developer

Job Description (max. 10 words): Developing code that handles actions in games.

Date: 18 February 2019 – 1 march 2019

What do you consider to be the three to five priorities of your job as you understand them?

I believe that in this job the top 4 priorities are being passionate about it, being flexible, being able to work together in a team, and having a lot of patience. You have to be passionate about this job because this makes you better at it. Being passionate about a job is always a priority. You have to be flexible and adapt to different kinds of environments. You also have to be able to step out of your comfort zone. A game developer also has to be able to work in a team because most of the time you'll work together with other people to develop a game. Last of all you have to have a lot of patience. This is because you have to do a lot of testing and you have to be able to keep going even when your code keeps malfunctioning.

What particular part(s) of your job did you enjoy most, and why?

The thing I enjoyed most about my internship is that I was able to see myself making progress. At the beginning of the internship I needed help a lot of the time. When the days went by I was able to solve problems on my own, using the internet. At the end of the week I was even able to help a colleague with a small issue. The reason I enjoyed this most is because I could see that I was learning new things.

What did you find most straightforward or easy in your job, and why?

I didn't find anything in my internship particularly easy. My job consisted of a lot of research, coding, testing and designing (placing objects in my game). Even looking up something on Google was not always easy or straightforward. This was because you always have to type exactly what you need in the right context to be able to get aid on the topic you need.

What do you find most difficult in your job, and why? Were there things that may have hindered your performance?

The most difficult for me was changing the way I think. I had to place myself into an environment that wasn't real and program mechanics to make the player move as realistically as possible. To do this I had to step

out of the “I’m developing a game” way of thinking and right in to the “I’m creating a world” way of thinking.

What do you see as your greatest accomplishments or successful efforts over this past review period?

My greatest accomplishment in my internship is the fact that I was able to “complete” my game in two weeks. This was not an easy task, and I thought that I would never be able to finish it in such a small period of time. Yet I did.

Complete the following sentence: I believe that my greatest contribution to the department and to my company is: ...

I believe that my greatest contribution to the department and to my company is the game that I have created.

In what area or areas would you like to gain more experience, training or education? Do you feel like your school training was insufficient in certain areas? Which areas? Why?

I would like to gain more experience in the field of game development. This is a world that I’ve gotten a scent from at my internship and I want to gain experience in this field. Our school training was a good base for my internship. But I have noticed that a lot of the times we are taught things just because they have to be that way, without an explanation. When given an explanation why things (code) have to be used a certain kind of way it is way easier to understand it.

What could you do to perform similar job duties and assigned tasks more efficiently in the future?

If I would have to perform similar job duties I would try to learn as much as I can find about the topic that I’m working on. This makes the job go way smoother.

What can your supervisor or co-workers do to assist you in becoming more efficient in the future?

The best way for my co-workers to assist me in becoming more efficient in the future is letting me do research before helping. This is something that the people at my internship did. That is the reason that I have learned so much in this short period.

Write a ‘review’ about the company where you did your internship. Did you enjoy working there? What were the positive and negative aspects of this company for you (at least 4 aspects in total). Use at least 150 words.

I have enjoyed my internship at Firewolf Studios a lot. I was welcomed into their team with open arms. My co-workers were always at my disposal and were always very happy to assist me when I needed it. I have learned a lot about the working environment and the job. In these two weeks I have realized that I really want to be able to work as a game developer. I have developed as a person in these two small weeks. Literally the only negative thing about this internship is that it only lasted for two weeks. I hope that I can work in a company like this (or perhaps this company) in the future after I graduate. Another good thing about this internship is that it has taught me what a job as a game developer is. I have also learned how to use Google properly. This is something that an programmer has to be able to do.

## 11 Dankbrief

Evi Boelen  
Stiernerbeekstraat 29  
3600 GENK  
Tel. 0489501321

04 februari 2019

Firewolf Studios  
Kempische steenweg 303, bus 16  
3500 HASSELT

### Dankbief

Beste

Dit jaar kreeg ik de kans om stage te lopen in jullie bedrijf Firewolf Studios. Dit was een enorm leerrijke ervaring voor mij. Niet alleen maar leerrijk vanwege de bergen aan nieuwe kennis die ik kon verwerken maar ook voor mijzelf als persoon. Ik heb in de afgelopen twee stageweken ingezien dat ik thuis hoor in een omgeving zoals die op mijn stageplaats. Hiervoor zou ik graag alle werknemers van Firewolf Studios bedanken.

Ik zou graag Wouter Spaas en Gianluca Sorrentino bedanken om me elke dag bij te staan in het ontwikkelen van mijn game die zou gaan dienen als mijn stageproject. Wouter Spaas heeft me meer inzicht gegeven in het coderen dat achter een game zit. Hij heeft me bij elk probleem dat ik tegen kwam de kans gegeven om het zelf op te lossen. Hierdoor heb ik leren opzoeken op de juiste manier. Dit gaf me ook de kans om zelfstandig te werken waardoor ik nu op mijzelf een soortgelijke game zou kunnen maken. De softwarekennis die Wouter me bij bracht kan ik ook toepassen in de game waar ik voor mijn stage al aan werkte. Gianluca Sorrentino zou ik graag bedanken voor het bijbrengen van kennis over 3D-ontwerp. Het was leerrijk om te zien hoe een game ontworpen wordt. Vervolgens zou ik graag Frank Van Geirt bedanken. Hij heeft me de kans gegeven om stage te lopen bij Firewolf Studios. Deze ervaring zal ik levenslang bij me dragen en heeft me veranderd als persoon. Zijn kennis over programmeren en een job als programmeur heeft me veel bijgeleerd over mijn mogelijke toekomst.

Jo Claes zou ik graag bedanken voor het inzicht dat hij me gaf over het zakelijke aspect van een bedrijf zoals Firewolf Studios. Hem wil ik ook graag bedanken voor enkele levenslessen die ik voor altijd bij zal dragen. Met Stijn Mommen heb ik niet veel tijd kunnen doorbrengen. Dit kwam omdat hij vaak naar vergaderingen en andere activiteiten moest gaan. Toch zou ik hem graag bedanken voor het geloven in me als stagiair en de gezellige sfeer die altijd aanwezig was.

Ik zou graag nog eens heel Firewolf Studios bedanken voor het aanbieden van deze geweldige stage.

Hoogachtend

Evi Boelen

## 12      Uitnodiging mondelinge verdediging

Evi Boelen  
Stiernerbeekstraat 29  
3600 GENK  
Tel. 0489501321

24 april 2019

Firewolf Studios  
Kempische steenweg 303, bus 16  
3500 HASSELT

### **Uitnodiging**

Beste

Tijdens de periode van 18 februari tot en met 1 maart kreeg ik de kans om stage te lopen bij jullie in Firewolf Studios. Ik wil alle werknemers van Firewolf Studios graag bedanken voor deze leuke en leerrijke weken. De sfeer zat altijd goed, ik kreeg voldoende de kans om zelfstandig te werken (waar ik heel veel uit heb geleerd), en ik heb veel geleerd over mezelf en mijn toekomstvisie.

Nu dat al mijn gip-opdrachten stilaan afgewerkt geraken komt de mondelinge verdediging elke dag dichterbij. Ik zou jullie graag uitnodigen om mijn mondelinge verdediging bij te wonen op woensdag 5 juni. Dit zal plaats vinden om 08.30 uur op de Campus Hast gelegen in de Kleine Breestraat 7 in Hasselt.

De bijwoning van deze mondelinge verdediging is volledig vrijblijvend en dus zeker niet verplicht. Maar als u interesse hebt om deze mondelinge verdediging bij te wonen kan dit.

Bedankt dat ik de kans kreeg om mijn stage bij jullie te doen, en hopelijk tot op de mondelinge verdediging.

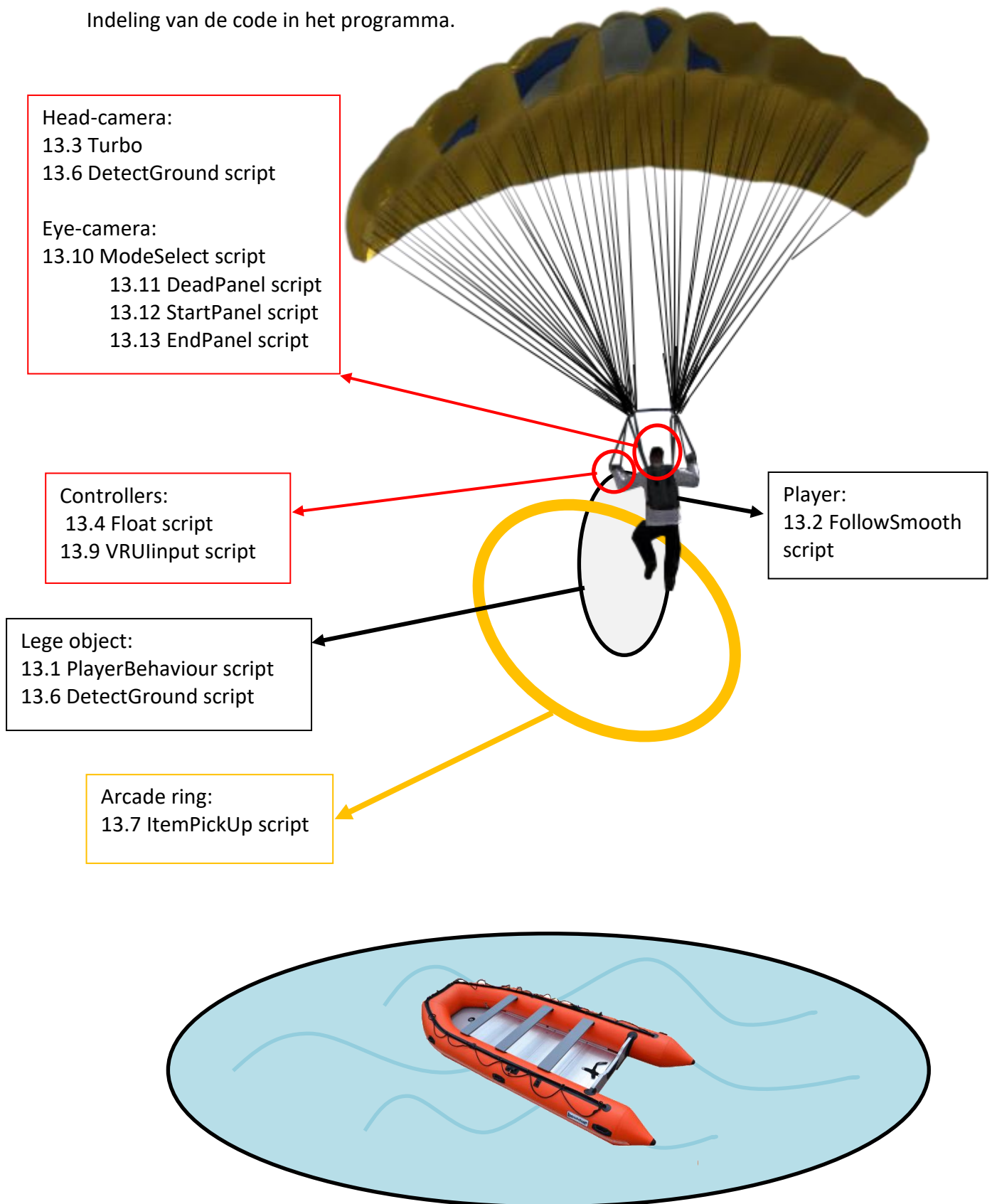
Vriendelijke groet

Boelen Evi



## 13 Code

Indeling van de code in het programma.



## 13.1 Playerbehaviour script

Dit script bestuurt de bewegingen van het lege object. De speler volgt het lege object (voor een zachte beweging). Dit script kijkt ook of de speler op de grond staat of niet, en als de speler al op start gedrukt heeft of niet.

```
public class PlayerBehaviour : MonoBehaviour
{
    public float gravity;
    public float rotateSpeed;
    public float moveSpeed;

    public int ringCount;

    public float extraGravity;
    public float extraMoveSpeed;

    public Vector3 moveDirection;
    public Transform cameraHead;

    public Animator planeAnim;
    public ParticleSystem playerWind;
    public ParticleSystem playerClouds;

    public bool isGrounded = false;
    public bool isEnded = false;
    public bool isClickedstart = false;

    public AudioSource powerup;

    //rotate
    public void RotatePlayer(float rotationValue)
    {
        transform.Rotate(Vector3.up, rotationValue, Space.World);
    }
    void FixedUpdate()
    {
        if (isEnded) return;

        moveDirection = Vector3.zero;
        if (isClickedstart == true)
        {
            if (!isGrounded)
            {
                moveDirection = Vector3.forward * GetMoveSpeed();
                moveDirection.y -= GetGravity();
                transform.Translate(moveDirection * Time.deltaTime, Space.Self);
                planeAnim.SetTrigger("Jumped");
                playerWind.Play();
                playerClouds.Play();
            }
            else
            {
                .
            }
        }
    }
}
```

```

        else
        {
            moveSpeed = 0;
            playerWind.Stop();
            playerClouds.Stop();
        }
    }

}

public float GetMoveSpeed()
{
    return moveSpeed + extraMoveSpeed;
}

private float GetGravity()
{
    return gravity + extraGravity;
}

```

## 13.2 FollowSmooth Script

Dit script zorgt er voor dat de speler het lege object zacht volgt.

```

public class FollowSmooth : MonoBehaviour
{
    // The target we are following
    [SerializeField]
    private Transform target;
    public float positionDamping;

    // Update is called once per frame
    void LateUpdate()
    {
        if (!target)
            return;

        //The player object tracks the movement of the chosen target (target.position)
        //The player object follows this movement with a speed assigned as Damping (with delay)
        transform.position = Vector3.Lerp(transform.position, target.position, positionDamping * Time.deltaTime);

        //Makes the player object look at the chosen target and follow it smoothly
        var rotation = Quaternion.LookRotation(target.position - transform.position);
        transform.rotation = Quaternion.Slerp(transform.rotation, rotation, Time.deltaTime * positionDamping);
        transform.rotation = Quaternion.Euler(0, transform.eulerAngles.y, 0 /*transform.eulerAngles.z*/);
    }
}

```

## 13.3 Turbo script

Het turbo script zorgt er voor dat de speler bij het indrukken van een trigger, sneller naar beneden vliegt naar de kant van de trigger die ingedrukt werd. Als beide triggers ingedrukt worden, zal de speler sneller naar voor glijden.

```
public class Turbo : MonoBehaviour
{
    PlayerBehaviour playerBehaviour;
    public float extraGravity;
    public float extraMoveSpeed;

    private bool isRightTriggered;
    private bool isLeftTriggered;
    //playerBehaviourscript gets called
    private void Awake()
    {
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();
    }
    //procedure returns boolean and the type of controller that has been triggered
    //if the controller that had been used is left, istriggerleft = true, same counts for right
    public void ControllerTriggered(bool isTriggered, ControllerType controllerType)
    {
        if (controllerType == ControllerType.Left){
            isLeftTriggered = isTriggered;
        }
        else if (controllerType == ControllerType.Right){
            isRightTriggered = isTriggered;
        }
    }
    private void Update()
    {
        //if controller trigger left and right are pressed --> 2 * extra speed forward (turbo)
        //if controller trigger left and right are pressed --> 2 * extra gravity (downward)
        if (isLeftTriggered && isRightTriggered)
        {
            playerBehaviour.extraGravity = extraGravity * 2;
            playerBehaviour.extraMoveSpeed = extraMoveSpeed * 2;
        }
        //if left OR right trigger is pressed --> extra speed forward (faster in 1 direction)
        //if left OR right trigger is pressed --> extra gravity (downward)
        else if (isLeftTriggered || isRightTriggered ){
            playerBehaviour.extraGravity = extraGravity;
            playerBehaviour.extraMoveSpeed = extraMoveSpeed;
        }
        else{
            //no extra speed or gravity
            playerBehaviour.extraGravity = 0;
            playerBehaviour.extraMoveSpeed = 0;
        }
    }
}
```

## 13.4 Float script

Onderstaand script werkt samen met het PlayerBehaviour script en het Turbo script. Het script leest in of er een controller gebruikt word en geeft deze waarde dan door aan het Turbo script.

```
public class Float : MonoBehaviour
{
    public ControllerType controllerType;
    private SteamVR_TrackedObject trackedObj;
    private PlayerBehaviour playerBehaviour;
    private Turbo turbo;

    public float moveValue;
    public float rotateValue;

    private SteamVR_Controller.Device Controller
    {
        get
        {
            return SteamVR_Controller.Input((int)trackedObj.index);
        }
    }

    private void Awake()
    {
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();
        turbo = FindObjectOfType<Turbo>();
    }

    void Start()
    {
        trackedObj = GetComponent<SteamVR_TrackedObject>();
    }
    // Update is called once per frame
    void FixedUpdate()
    {
        if (Controller.GetHairTrigger())
        {
            playerBehaviour.RotatePlayer(rotateValue);
            turbo.ControllerTriggered(true, controllerType);
        }
        else
        {
            turbo.ControllerTriggered(false, controllerType);
        }
    }
}
```

## 13.5 Controllertype (enum) script

```
public enum ControllerType
{
    Left = 0,
    Right = 1
}
```

## 13.6 DetectGround script

Dit script kijkt of de speler de grond raakt. Dit script zit zowel op het lege object als op de speler vast. Het script hangt op het lege object zal eerst de grond raken als de speler landt. Het script hangt op de speler (head-camera) omdat er zo gedetecteerd kan worden wanneer de speler uit het vliegtuig wandelt.

```
public class DetectGround : MonoBehaviour
{
    PlayerBehaviour playerBehaviour;
    public float distanceToGround = 0.5f;
    public string DontMoveOn, stopMoving;

    public EndPanel endPanel;
    public DeadPanel deadPanel;

    private void Awake()
    {
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();
    }

    // Update is called once per frame
    void FixedUpdate()
    {
        //declaration raycast (detects if there is a collision)
        RaycastHit hit;

        //The raycast starts at the position of the player object
        //The raycast casts a ray downwards and calculates if there has been a collision
        //The raycast sends a value out to that is true when there is a collision and false when there is not
        if (Physics.Raycast(transform.position, Vector3.down, out hit, distanceToGround))
        {
            Debug.Log(hit.transform.gameObject);
            //if the gameObject with the tag "Ground" of "RubberBoeat is hit, it returns that the player is on the ground
            // player stops moving
            if (hit.transform.gameObject.tag == "Plane")
            {
                playerBehaviour.isGrounded = true;
                playerBehaviour.moveSpeed = 0;
                //transform.Translate(new Vector3(0, 0, 0));
            }
            else if (hit.transform.gameObject.tag == "RubberBoat")
            {
                playerBehaviour.isGrounded = true;
                playerBehaviour.isEnded = true;
                endPanel.Show();
                endPanel.RingText();
            }
            else if (hit.transform.gameObject.tag == "Water" || hit.transform.gameObject.tag == "Envoirement")
            {
                playerBehaviour.isGrounded = true;
                playerBehaviour.gravity = 0;
                deadPanel.Show();
            }
            //else it returns that the player is not on the ground
            else
            {
                playerBehaviour.isGrounded = false;
                playerBehaviour.moveSpeed = 3;
            }
        }
        else
        {
            playerBehaviour.isGrounded = false;
            playerBehaviour.moveSpeed = 3;
        }
    }
}
```

## 13.7 ItemPickUp script

Dit script detecteert wanneer de speler in contact komt met een ring. Elke keer als dit gebeurd word dit opgeteld. Er speelt zich ook een klein geluidje af en de ring verdwijnt.

```
public class ItemPickUp : MonoBehaviour
{
    PlayerBehaviour playerBehaviour;

    public ParticleSystem ring1;
    public ParticleSystem ring2;

    // Use this for initialization
    void Start()
    {
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.GetComponent<Collider>().tag == "Player")
        {
            playerBehaviour.ringCount++;
            playerBehaviour.powerup.Play();
            ring1.Stop();
            ring2.Play();
        }
    }
}
```

## 13.8 VRUIItem

Dit script hangt vast op knoppen die je kan gebruiken in de game. Het script geeft je de functie om te klikken op de knop.

```
[RequireComponent(typeof(RectTransform))]  
public class VRUIItem : MonoBehaviour {  
  
    private BoxCollider boxCollider;  
    private RectTransform rectTransform;  
  
    public Button btnChoice;  
  
    private void OnEnable()  
    {  
        ValidateCollider();  
    }  
    private void OnValidate()  
    {  
        ValidateCollider();  
    }  
  
    private void ValidateCollider()  
    {  
        rectTransform = GetComponent<RectTransform>();  
  
        boxCollider = GetComponent<BoxCollider>();  
        if (boxCollider == null)  
        {  
            boxCollider = gameObject.AddComponent<BoxCollider>();  
        }  
  
        boxCollider.size = rectTransform.sizeDelta;  
    }  
}
```



## 13.9 VRUIInput script

Dit script registreert of er op een laser gedrukt wordt (met de trigger van de HTC VIVE console). Dit script zorgt er ook voor dat het mogelijk is om acties te verrichten met de lasers zoals op knoppen drukken.

```
public class VRUIInput : MonoBehaviour
{
    public SteamVR_LaserPointer laserPointer;
    private SteamVR_TrackedController trackedController;

    private void OnEnable()
    {
        laserPointer = GetComponent<SteamVR_LaserPointer>();
        laserPointer.PointerIn -= HandlePointerIn;
        laserPointer.PointerIn += HandlePointerIn;
        laserPointer.PointerOut -= HandlePointerOut;
        laserPointer.PointerOut += HandlePointerOut;

        trackedController = GetComponent<SteamVR_TrackedController>();
        if (trackedController == null)
        {
            trackedController = GetComponentInParent<SteamVR_TrackedController>();
        }
        trackedController.TriggerClicked -= HandleTriggerClicked;
        trackedController.TriggerClicked += HandleTriggerClicked;
    }

    private void HandleTriggerClicked(object sender, ClickedEventArgs e)
    {
        if (EventSystem.current.currentSelectedGameObject != null)
        {
            ExecuteEvents.Execute(EventSystem.current.currentSelectedGameObject,
                new PointerEventData(EventSystem.current), ExecuteEvents.submitHandler);
            laserPointer.active = false;
        }
    }

    private void HandlePointerIn(object sender, PointerEventArgs e)
    {
        var button = e.target.GetComponent<Button>();
        if (button != null)
        {
            button.Select();
            Debug.Log("HandlePointerIn", e.target.gameObject);
        }
    }

    private void HandlePointerOut(object sender, PointerEventArgs e)
    {
        var button = e.target.GetComponent<Button>();
        if (button != null)
        {
            EventSystem.current.SetSelectedGameObject(null);
            Debug.Log("HandlePointerOut", e.target.gameObject);
        }
    }
}
```

## 13.10 ModeSelect script

In dit script wordt het Start scherm, eindscherm en het “missie gefaald” scherm aangeroepen. Het Menu hangt vast aan de Eye-camera van de speler. Zo komt het menu altijd voor de ogen van de gebruiker terecht.

```
public class ModeSelect : MonoBehaviour
{
    public Canvas cnv;
    public StartPanel startPanel;
    public DeadPanel deadPanel;
    public EndPanel endPanel;

    PlayerBehaviour playerBehaviour;
    SteamVR_LaserPointer laserPointer;

    VRUIInput laser;
    //private void Start()
    //{
    //    GetComponent<SteamVR_LaserPointer>().enabled = true;
    //}
    void Start()
    {
        startPanel.Show();
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();
    }
    public void IsClickedStart(string mode)
    {
        startPanel.Hide();
        playerBehaviour.isClickedstart = true;
    }
    public void IsClickedTryAgain()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        playerBehaviour.ringCount = 0;
    }
    public void IsClickedCancel()
    {
        Application.Quit();
    }
}
```

## 13.11 DeadPanel

```
public class DeadPanel : MonoBehaviour {

    void Awake()
    {
        Hide();
    }
    public void Show()
    {
        gameObject.SetActive(true);
    }
    public void Hide()
    {
        gameObject.SetActive(false);
    }
}
```

## 13.12 StartPanel script

```
public class StartPanel : MonoBehaviour {  
  
    void Awake()  
    {  
        Show();  
    }  
    public void Show()  
    {  
        gameObject.SetActive(true);  
    }  
    public void Hide()  
    {  
        gameObject.SetActive(false);  
    }  
}
```

## 13.13 EndPanel script

```
public class EndPanel : MonoBehaviour {  
  
    public Text points;  
    PlayerBehaviour playerBehaviour;  
  
    void Awake()  
    {  
        Hide();  
        playerBehaviour = FindObjectOfType<PlayerBehaviour>();  
    }  
    public void Show()  
    {  
        gameObject.SetActive(true);  
    }  
    public void Hide()  
    {  
        gameObject.SetActive(false);  
    }  
    public void RingText()  
    {  
        points.text = " You have collected " + playerBehaviour.ringCount + " rings.";  
    }  
}
```

## Besluit

Het was een zeer interessante periode met interessante opdrachten waaruit ik veel heb bijgeleerd. Het ging niet altijd even goed, en soms zag ik het even niet meer zitten omdat sommige opdrachten veel energie vroegen. Maar ik heb niet opgegeven en ik ben ook heel blij dat ik dit niet heb gedaan. De kennis en de ervaring die ik op heb gedaan tijdens deze stage zal ik voor altijd mee dragen in mijn rugzak terwijl ik rondtrek doorheen mijn leven.

Ik volgde mijn stage bij Firewolf Studios. Ik heb hier vooral geleerd hoe een dag van een programmeur in hun bedrijf er uit ziet. Tijdens deze stage heb ik ook ingezien hoe belangrijk onderzoek doen naar een probleem is. Als ik een probleem tegenkwam kreeg ik altijd zelf eerst de kans om op zoek te gaan naar een oplossing. Als ik meerdere oplossingen geprobeerd had en het lukte nog steeds niet dan kreeg ik hulp of vroeg ik hulp. In het bedrijf heb ik ook gezien dat niemand alles weet. Iedereen moet regelmatig wat opzoeken of af en toe hulp vragen aan iemand anders. Mensen zijn natuurlijk geen robots.

Ik kijk enorm uit naar mijn GIP-verdediging. Dit zal een moment zijn dat ik mondeling al mijn ervaringen met de leerkrachten en de externe jury kan delen. Dit doe ik met veel enthousiasme. Ik ben er ook wel wat zenuwachtig voor, maar dat is normaal. Ik hoop dat ik zowel de leerkrachten als de leden van de externe jury ook wat bij kan leren.

Ik zou als laatste graag nog eens iedereen bedanken die mij heeft geholpen om hier te geraken. De werknemers van Firewolf Studios bedank ik graag nog eens een extra keer om mij in te laten zien wat ik met mijn toekomst wil doen.

## Bronvermelding

*Scripting*. Geraadpleegd op 2018-10-21 via <https://unity3d.com/learn/tutorials/s/scripting>

*Roll a ball tutorial*. Geraadpleegd op 2018-12-12 via <https://learn.unity.com/project/roll-a-ball-tutorial>

*Unity Tutorial*. Geraadpleegd op 2019-01-11 via <https://unity3d.com/learn/tutorials>

3D game-kit tutorial. Geraadpleegd op 2019-01-11 via <https://unity3d.com/learn/tutorials/s/3d-game-kit>

## Bijlagen

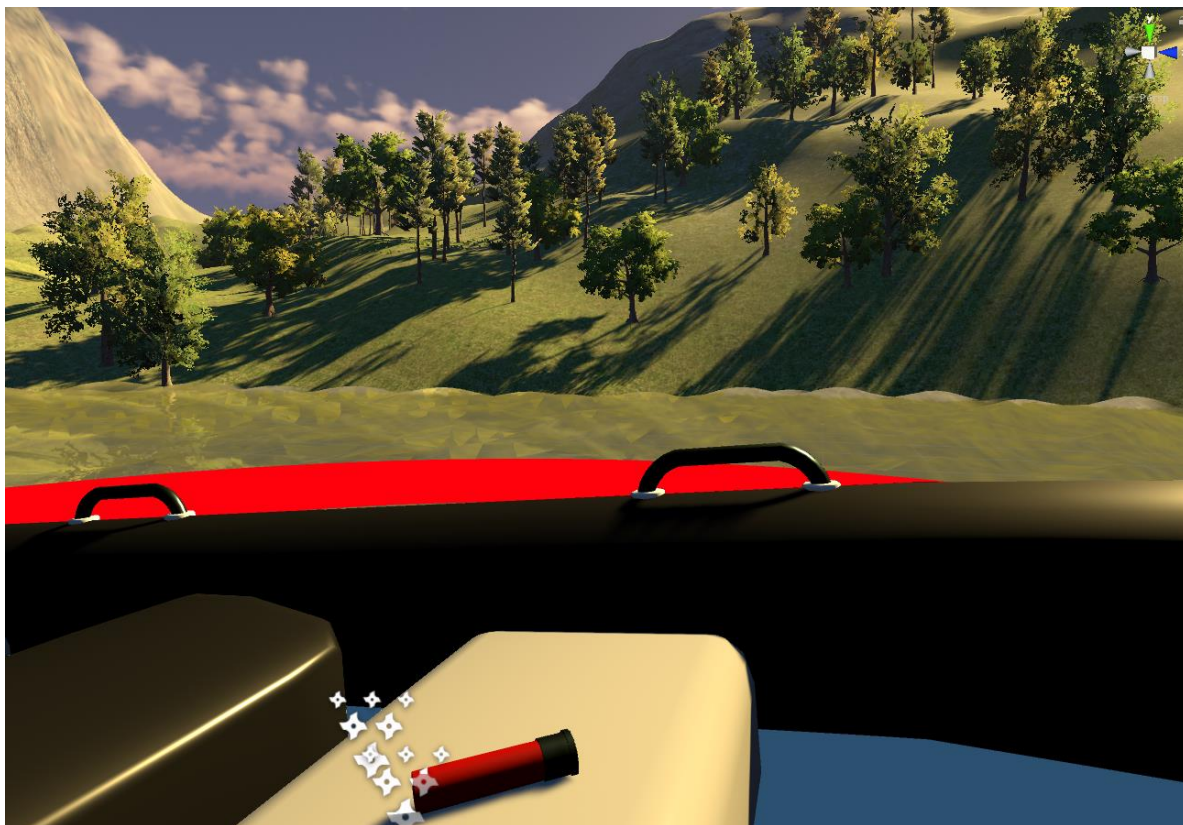
- 1 De omgeving van de game
- 2 Binnenkant van het vliegtuig
- 3 Het uitzicht vanuit de boot
- 4 Organigramme



1. Omgeving



2. Interieur vliegtuig



3. Uitzicht vanuit het bootje



# Organigramme

