23-2-2022

Rotterdam in VR

INFPRJ02D

Mohamad Almarawi,

Parwesh Bhaggan,

Moesab Ajdid,

Dylan Visser,

Darshan Alakhramsing,

Bas Weidmann

# Voorwoord

Voor het vak Project D hebben wij de taak om het probleem van een PO (Product Owner) op te lossen. Wij als groep 2 zullen hier de juiste oplossing voor gaan vinden doormiddel van onderzoeken wat is vastgelegd in dit document. De verschillende taken zullen verdeeld worden tussen de 6 leden uit onze groep. Hierbij maken wij gebruik van de scrum-methodiek om alles in een gestructureerde en overzichtelijke wijze te laten verlopen.

Wij willen graag de Stakeholders Peter van Waart, Rob van Willigen, Elise van Beurden, Jack Hoefnagel en onze project docenten Natnaël Tefera en Elise Bijl bedanken voor hun hulp bij het maken van dit verslag. Door feedback te hebben gekregen op de verschillende onderwerpen hebben wij ons onderzoek beter kunnen uitvoeren.

Inhoud

[Voorwoord 1](#_Toc1945964909)

[Inleiding 4](#_Toc1108084805)

[Inhoud van het rapport: 5](#_Toc295568184)

[Beginnen met onderzoek naar oplossingen 5](#_Toc1378652550)

[Onderzoek & ontwerp 5](#_Toc57248397)

[Test & validatie 5](#_Toc1706048004)

[Leren & verbeteren 7](#_Toc1146955963)

[3D Modellen Rotterdam 7](#_Toc486301252)

[Onderzoek & ontwerp 7](#_Toc522155674)

[Test & validatie 7](#_Toc269716464)

[Leren & verbeteren 8](#_Toc42518739)

[Unreal Engine 4 8](#_Toc1485228138)

[Onderzoek & ontwerp 8](#_Toc1767903279)

[Test & validatie 8](#_Toc1149398077)

[Leren & verbeteren 10](#_Toc1861807906)

[CryEngine 5.6 11](#_Toc1861507532)

[Onderzoek & ontwerp 11](#_Toc1681921205)

[Test & validatie 11](#_Toc1098918246)

[Leren & verbeteren 12](#_Toc1796137949)

[Unity 12](#_Toc1316790768)

[Onderzoek & ontwerp 12](#_Toc2131075031)

[Test & validatie 13](#_Toc934885452)

[Leren & verbeteren 13](#_Toc1113808463)

[Keuze maken voor een ontwikkelomgeving 13](#_Toc763211242)

[Onderzoek & ontwerp 13](#_Toc375475970)

[Test & validatie 13](#_Toc2106650812)

[Leren & verbeteren 13](#_Toc1312228189)

[Onderzoek Real Time RET-dienstregeling 13](#_Toc920792929)

[Onderzoek & ontwerp 13](#_Toc1493952185)

[Test & validatie 13](#_Toc1013348428)

[Unity Colliders 13](#_Toc935965737)

[Test & validatie 13](#_Toc1567654072)

[Leren & verbeteren 13](#_Toc1298177929)

[Checkpoints 13](#_Toc916102317)

[Leren & verbeteren 13](#_Toc1447589679)

[Werken met Protobuf en GTFS 13](#_Toc2067776009)

[Onderzoek & ontwerp 13](#_Toc2084914448)

[Test & validatie 13](#_Toc578599592)

[Leren & verbeteren 13](#_Toc441564951)

[Code implementeren in Unity 13](#_Toc1993945502)

[Onderzoek & ontwerp 13](#_Toc164920049)

[Test & validatie 13](#_Toc657099240)

[GitHub bij iedereen werkend krijgen 13](#_Toc1301076741)

[Onderzoek & ontwerp 13](#_Toc883133747)

[Test & validatie 13](#_Toc1711652241)

[Leren & verbeteren 13](#_Toc1871909490)

[KV78Turbo-OVAPI 13](#_Toc1727684291)

[Onderzoek & ontwerp 13](#_Toc7532290)

[Test & validatie 13](#_Toc1683865453)

[Leren & verbeteren 13](#_Toc983582146)

[NavMesh 13](#_Toc2088007329)

[Onderzoek & ontwerp 13](#_Toc705223919)

[Test & validatie 13](#_Toc1251821544)

[Leren & verbeteren 13](#_Toc1961536136)

[Conclusie: 13](#_Toc565716980)

[Referenties: 13](#_Toc2118779783)

# Inleiding

De PO heeft verteld dat er online data beschikbaar is van het Digital Twin project van de gemeente Rotterdam. De PO wil graag dat wij deze data combineren met de real time RET-dienstregeling. De real time RET-dienstregeling moet terug te zien zijn in een virtuele omgeving van station Beurs.

# Inhoud van het rapport:

## Beginnen met onderzoek naar oplossingen

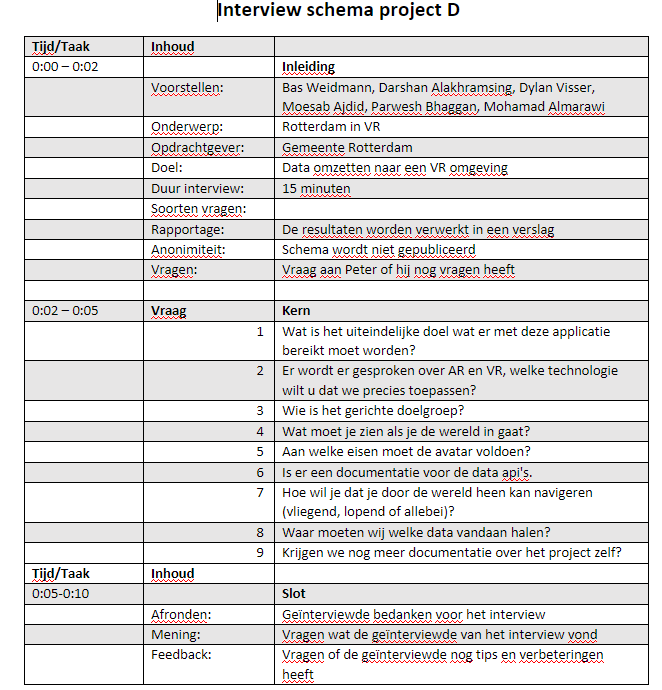
Doelstelling: Een duidelijk beeld krijgen van het probleem om oplossingen te zoeken.

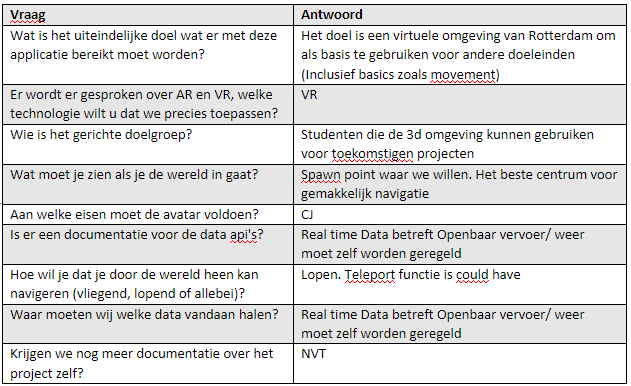
### Onderzoek & ontwerp

Uit het interview dat wij hebben gehouden met Peter hebben wij al bepaalde dingen kunnen vaststellen. Peter die wil graag dat je door Rotterdam centrum zou kunnen lopen in een VR omgeving. Als dit gemaakt is zou hij graag willen dat er extra opties bij de applicatie toegevoegd wordt.

### Test & validatie

Wij hebben een interviewschema uitgewerkt om semigestructureerd het interview te kunnen houden en binnen een respectabel tijdsduur te blijven.





### Leren & verbeteren

Het inplannen van het interview heeft twee tot bijna drie weken gekost. Als team vinden wij dat dit te lang heeft geduurd en zullen proberen in de toekomst sneller interviews in te plannen.

## 3D Modellen Rotterdam

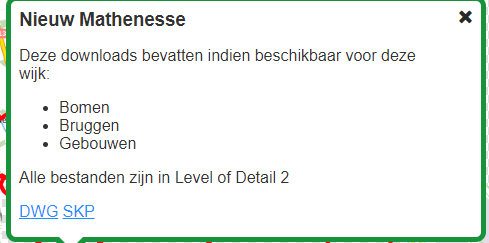
Doelstelling: Ophalen van de bestaande 3D gegevens van Rotterdam

### Onderzoek & ontwerp

Tijdens de eerste presentatie die Peter heeft gegeven die hij met ons heeft gedeeld staan er linkjes naar 3DRotterdam.nl. Hier hebben wij 3D modellen kunnen downloaden en geïmporteerd in Unity (Gemeente Rotterdam, n.d.).

### Test & validatie

We hebben de SKP-modellen gedownload van 3DRotterdam.nl en geprobeerd in een sketchup tool te openen. Dit is uiteindelijk niet gelukt. Vervolgens hebben wij de file formaat geprobeerd om te zetten wat ons na meerdere pogingen ook niet lukte. Deze hebben we uiteindelijk in een oudere versie van Blender kunnen openen met behulp van een externe add on. Hier liepen wij helaas tegen een ander probleem aan. De locatie van de modellen stonden namelijk nog op de juiste coördinaten wat ze niet zichtbaar maakte in grafische engines zoals sketchup of Unity. Dit is opgelost door alle coördinaten te verminderen in relatie met elkaar. Ook dit is gedaan in Blender doordat andere engines zoals Unity dit niet uit te box kunnen.



### Leren & verbeteren

De SKP-modellen worden momenteel nog met de hand naar de juiste locatie omgezet. Ook hebben de modellen geen texturen en hebben we nog geen terrein weten te verkrijgen.

## Unreal Engine 4

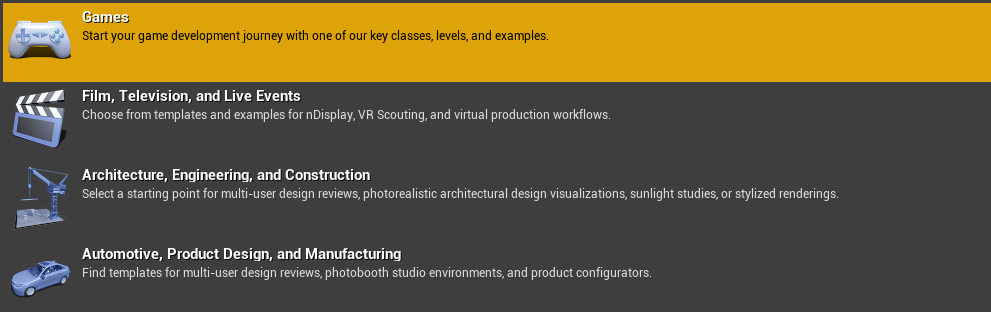
Doelstelling: Ontwikkelomgeving vinden die VR engines en SKP-modellen ondersteunt

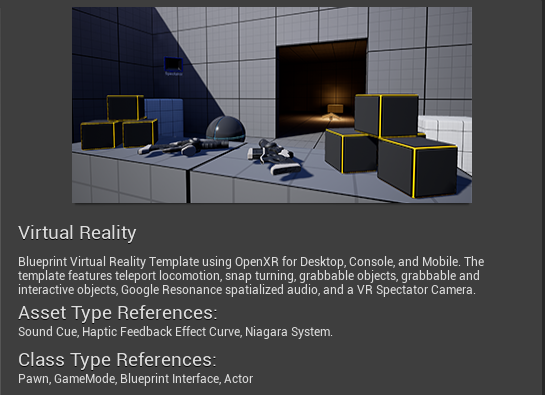
### Onderzoek & ontwerp

Unreal Engine 4 is één van de meest gebruikte game engines die er op dit moment op de markt is. Vele bekende spellen draaien op deze engine zoals: Batman, Unreal Tournament, Fortnite, Gears of War en nog veel meer. Het aantrekkelijkste aan Unreal Engine 4 is ook dat voor bepaalde individuen het complete gratis is om in te ontwikkelen (Unreal Engine, 2022).

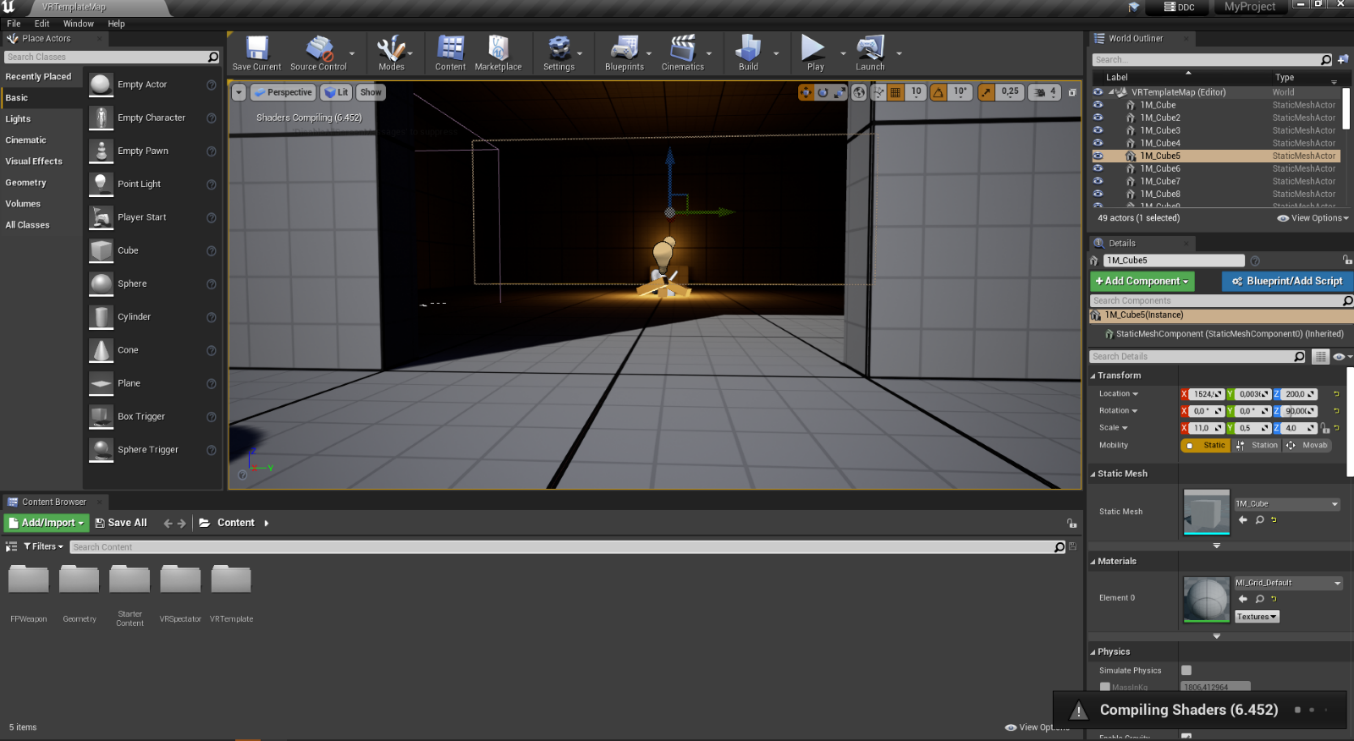
### Test & validatie

Om gebruik te maken van de Unreal Engine moet eerst de Epic Games client geinstalleerd worden. Daar moet je apart een account voor aanmaken. Vervolgens wanneer je met je account bent ingelogd kan je Unreal Engine installeren. Het aanmaken van een VR project in Unreal Engine geeft een paar opties aan om een project te starten één van hun eigen templates.

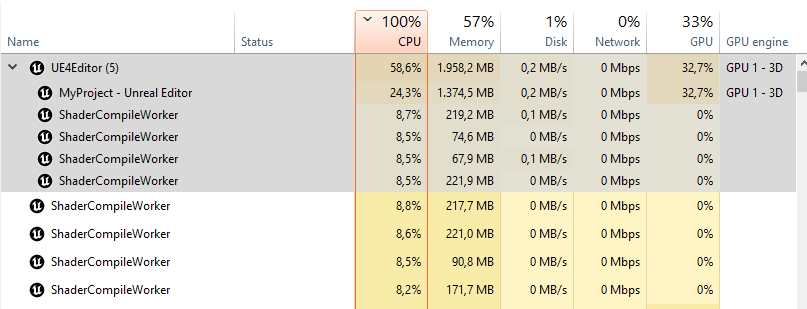
De enige VR template die er is, staat bij de games sectie



Zodra je een VR project hebt aangemaakt kom je dit allemaal te zien. Je krijgt de optie om een tutorial te volgen over hoe de UI allemaal in elkaar zit en ook de optie om uitleg te krjgen over wat alle hulpmiddelen doen waar je gebruik van kan maken.



Applicatie loopt vast tijdens het proberen op te starten van de template. Het kan misschien komen omdat niet alle shaders geladen zijn en het daarom gebeurt.



Shaders compilen in UE4 maakt heel wat gebruik van de CPU resources. Na het compilen van alle shaders duurt het nog steeds lang om zelfs de template alleen al in te laden. Ik denk niet dat UE4 development gedaan moet worden op laptops ten minste niet met laptops die mijn specificities hebben en lager. Voor UE4 moet je een externe code editor gebruiken om zelf scripts te schrijven zodat, UE4 ze vervolgens kan compilen.

### Leren & verbeteren

Unreal Engine 4 is iets meer user friendly als in de zin van, overal waar je klikt krijg je gelijk een tutorial van hoe het werkt. Als je besluit scripts te gaan schrijven voor bepaalde objecten, dan moet er wel gewerkt worden met C++. C++ is een iets gevorderde programmeertaal en lasting op te pakken voor iemand die er nog nooit iets mee heft gedaan. Unreal Engine is daar naast ook een heel zwaar programma. Op laptops duurt het opstarten/laden van de shaders veels te lang. Dit is geen ontwikkelomgeving waar je gebruik van wilt maken op laptops. Als laatste maar belangrijkste punt, het heeft geen ondersteuning om direct SKP modellen te importeren. Daarvoor is er een aparte plug-in nodig die niet gratis te krijgen is.

## CryEngine 5.6

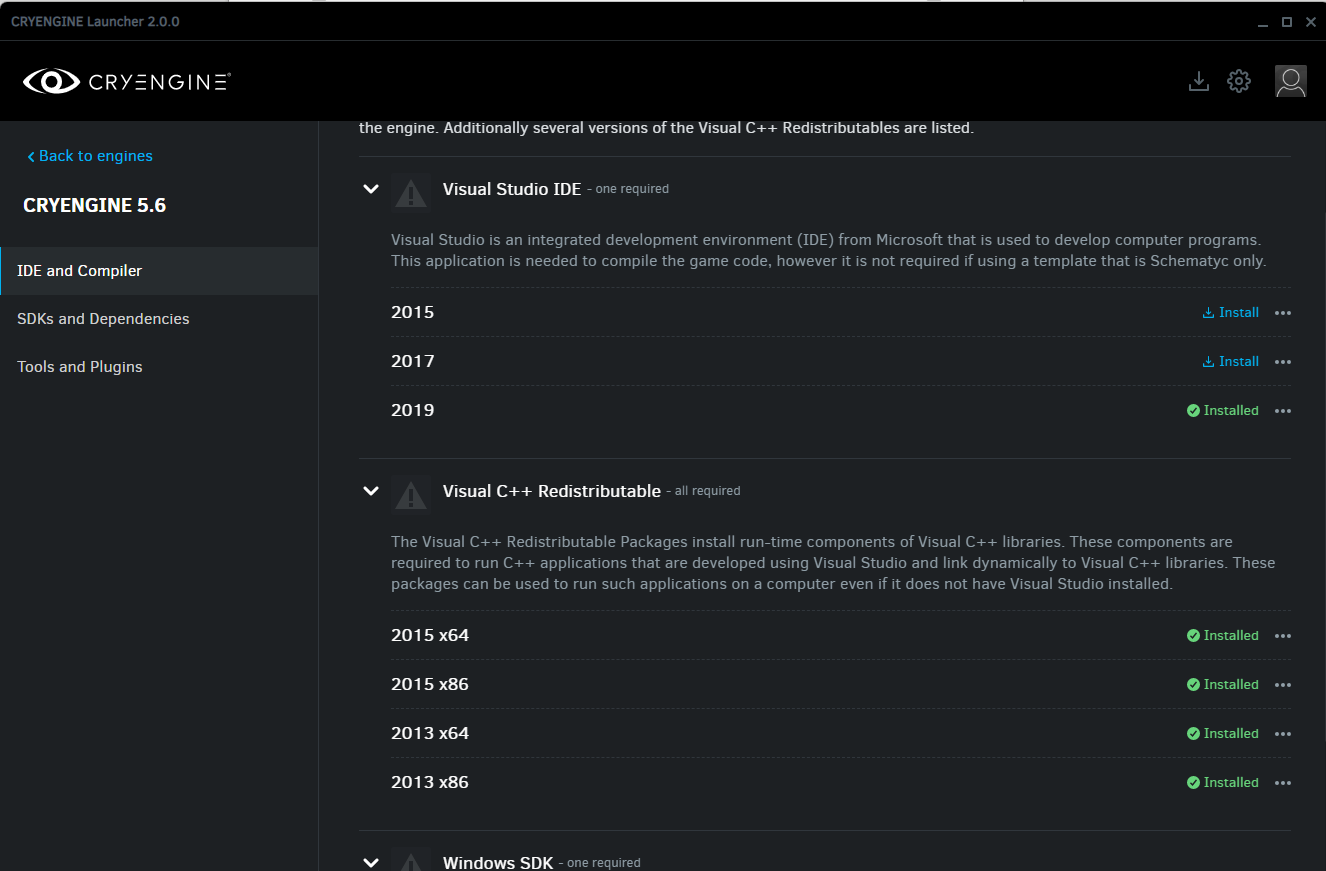
Doelstelling: Ontwikkelomgeving vinden die VR engines en SKP-modellen ondersteunt

### Onderzoek & ontwerp

CryEngine is een bekende game engine die gebruikt wordt voor spellen zoals: Far Cry, Crysis, Sniper Ghost Warrior en nog vele meer. Ook geeft het de mogelijkheid om voor kleine projecten een gratis variant om kennis te maken met de mogelijkheden die CryEngine aan te bieden heeft (CryEngine, n.d.)

### Test & validatie

Om te beginnen moet je de client installer downloaden van hun website. Wanneer dit gedaan is moet je een account maken om gebruik te kunnen maken van die features die de cliet aanbied. Eenmaal ingelogd zal je eerst zelf de engine moeten installeren om daarna een project te kunnen starten. Wanneer je de engine is geinstalleerd is krijg je een pop up met een overzicht van alles wat je nodig hebt om een project te kunnen starten. Ook kan je hier zien wat je wel geinstaleerd hebt en wat niet.



Na wat onderzoek te doen heb ik gezien op Cry Engine hun eigen site en documentatie dat ze verschillende vr platformen ondersteunen. Zelf kan ik niet zien waar ik een VR project in Cry Engine 5  kan starten. Ik heb een tutoiral gevonden die laat zien hoe je zo een project kan aanmaken maar, mijzelf is het niet gelukt om de stappen succesvol te volgen. VR support werd geintroduceerd met Cry Engine 5.5 en de laatste Cry Engine release is 5.6

### Leren & verbeteren

Op dit moment is CryEngine niks voor VR ontwikkeling (op dit moment) omdat, het gewoon niet lukt om zelfs een VR project te starten zelfs met behulp van guides en tutorials. Op hun website staat dat het SKP-modellen zou ondersteunen, maar dat kan nu lastig zelf gecontroleerd worden.

## Unity

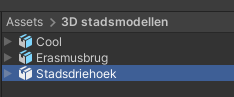
Doelstelling: Ontwikkelomgeving vinden die VR engines en SKP-modellen ondersteunt

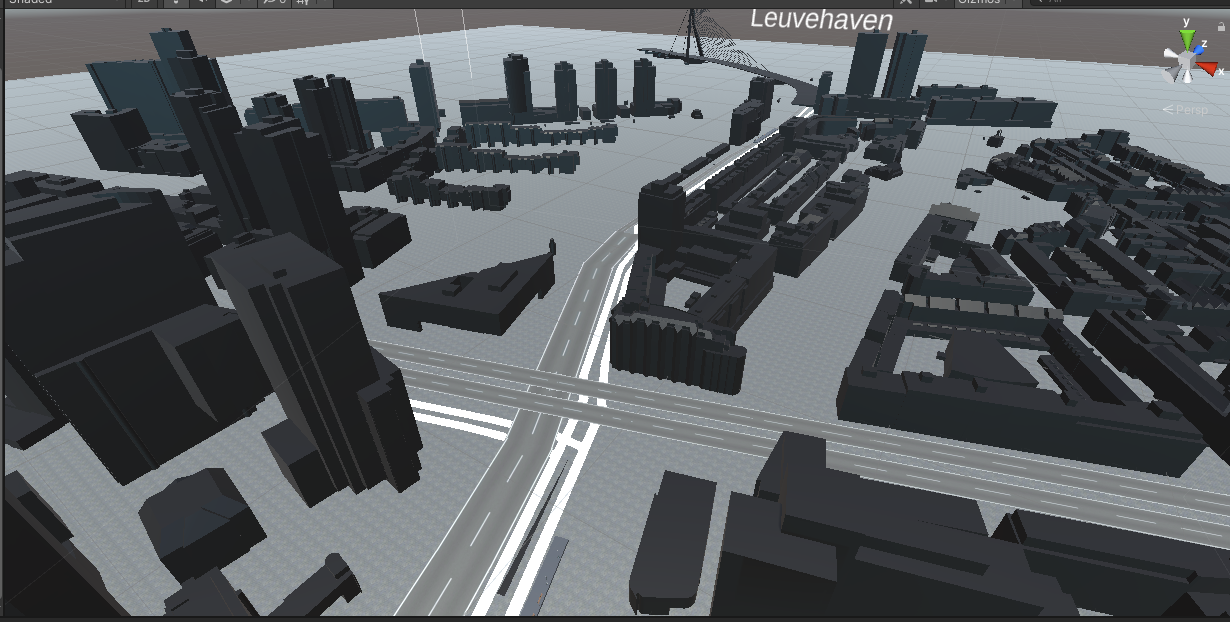
### Onderzoek & ontwerp

Unity is de meest gebruikte game engine voor zowel games en simulaties. Dit komt grotendeels doordat het gemakkelijk te gebruiken is en het vroegtijdig beschikbaar werd gemaakt voor iedere hobbyist. Voorbeelden van meer bekendere games gemaakt in Unity zijn; Pokemon Go, Fall Guys, Among us, en Escape from tarkov. Het is compleet gratis voor gebruik met als enige kosten een kleine precentage afhankelijk van inkomsten boven de 10.000 euro. Daarnaast gebruikt het ook nog C#.

### Test & validatie

Het instaleren van Unity en laden van een template is snel gedaan. Het is erg gebruiks vriendelijk en geeft ons de mogelijkheid om direct de SKP-modellen te importeren, als we ze eerst aan het assets mapje toevoegen.





A screenshot of a computer

Description automatically generated with medium confidence

### Leren & verbeteren

Unity is goed voor ontwikkelen op een laptop, sinds het een lichte applicatie is. Het direct inladen van de SKP-modellen bespaard veel tijd met het zelf converteren van de modellen naar een ander object. Naast deze punten maakt het gebruik van C#. Het is een programmeertaal die makkelijk op te pakken is voor programmeurs die al eerder met een C taal hebben gewerkt.

## Keuze maken voor een ontwikkelomgeving

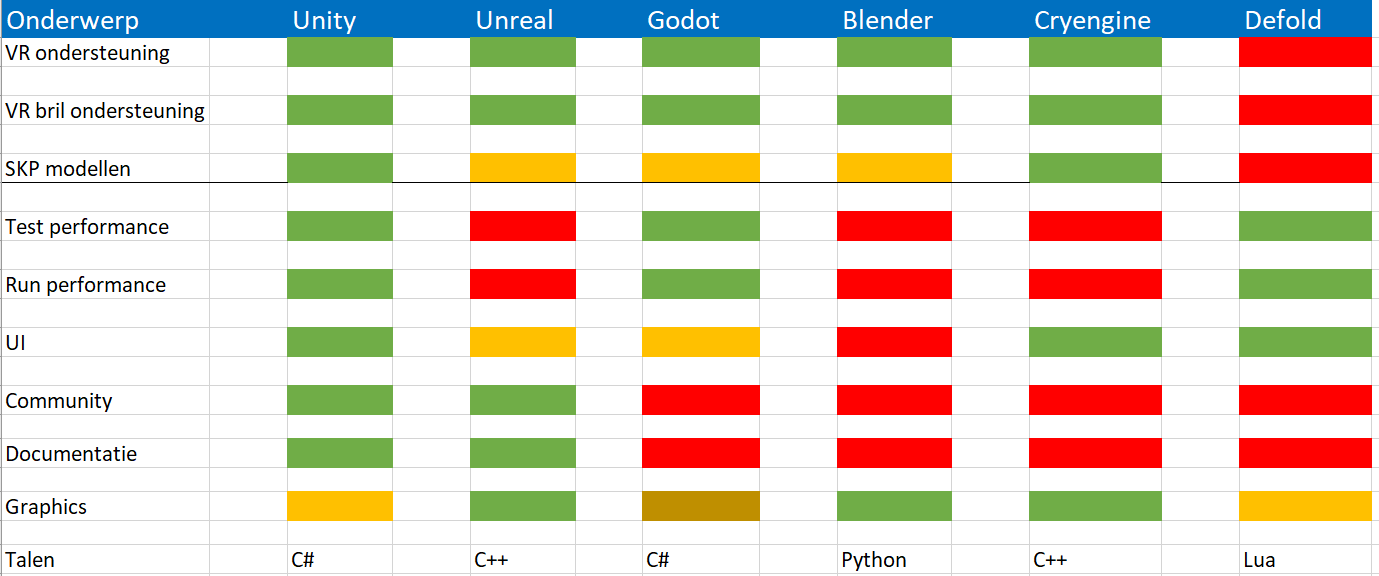
Doelstelling: Keuze maken voor een ontwikkelomgeving

### Onderzoek & ontwerp

Elk teamlid heeft onderzoek gedaan naar twee soorten VR engines. Uiteindelijk heeft elk lid doorgegeven welke engine hij goed vindt passen bij het project en uitgelegd waarom. Vervolgens hebben wij een keuze gemaakt uit drie verschillende engines die het beste zouden passen: Unity, Unreal Engine 4 en Cry Engine. Uiteindelijk hebben wij allemaal besloten in overleg met Peter om het project te maken met Unity.

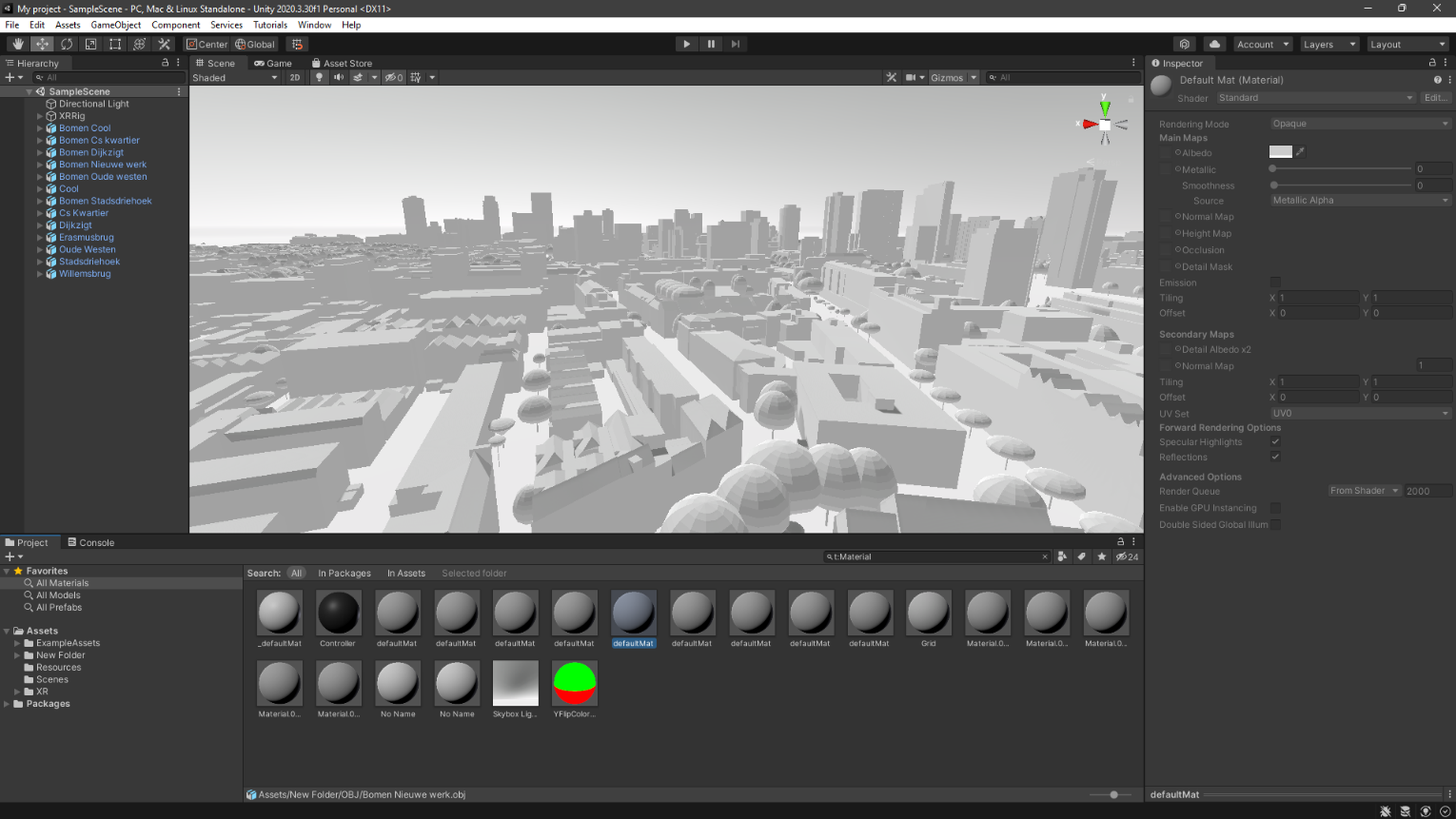
### Test & validatie

*Game Engines:*



### Leren & verbeteren

Met Unreal Engine 4 was er wel een verwachting dat we de SKP-modellen direct konden inladen al hoewel dit niet zo bleek te zijn in realiteit. In Unreal Engine 4 moet je helaas een plug-in gebruiken of een externe derde partij programma om de modellen eerst naar een ander type om te zetten en dan vervolgens in te laden in Unreal Engine 4. CryEngine heeft wel VR ondersteuning. Dit is alleen in een recente versie toegevoegd en zit dus nog in zijn beginfase qua functionaliteiten. Er zijn veel belangrijke dingen voor VR ontwikkeling die Unity en Unreal Engine 4 wel ondersteunen die niet aanwezig zijn in CryEngine 5.6. Met Unity kunnen wij de modellen gelijk in de omgeving importeren. Er is een probleem dat ze niet op de juiste positie direct werden geplaats. Dit is een niet al te groot probleem en kan handmatig opgelost te worden.



## Onderzoek Real Time RET-dienstregeling

Doelstelling: Onderzoek doen naar mogelijke verdere uitbreidingen

### Onderzoek & ontwerp

Een van de verdere uitbreidingen die Peter wil zien is dat er in de virtuele omgeving van Rotterdam real time ov-dienstregeling te zien is. Een van de mogelijkheden is om gebruik te maken van de data die OpenOV.nl (Stichting OpenGeo, 2014).

### Test & validatie

Wij hebben eerst de engines Unity en Unreal gedownload en vervolgens onderzocht naar Ondersteuning voor de gestelde eisen met behulp van libraries of plug-ins.

## Unity Colliders

Doelstelling: OV-model niet door andere objecten heen laten overlappen

Onderzoek & ontwerp

In Unity is er sprake van componenten. Een van de componenten categorie is colliders. Een collider zorgt ervoor dat een object met andere objecten kunnen interacteren op basis van botsingen. In de volgende afbeelding is de collider te herkennen aan het groen rechthoekig gedeelte.

### Test & validatie

Wij hebben de tram object een box collider gegeven. Er zijn meerdere colliders mogelijk namelijk: Sphere collider, Capsule collider, Mesh collider etc. Door ze allemaal te proberen kan er gekeken worden welke

### Leren & verbeteren

Box collider was handig toe te passen op de tram omdat de tram een rechthoekig vorm heeft waardoor de box collider beter aangepast kon worden. Zelf kan er ingesteld worden hoe groot of breed de collider zal zijn voordat het met een ander object zal botsen.

## Checkpoints

Doelstelling: OV-model laten stoppen bij het station

Onderzoek & ontwerp

Om ervoor te zorgen dat de OV-modellen stoppen bij Beurs kunnen we checkpoint functies gebruiken die Unity aanbied. Met deze checkpoints kunnen we aangeven waar de OV-modellen naar toe gaan en waar ze zullen stoppen.

Test & validatie

Na het plaatsen van een checkpoint en een OV-model daar naar toe laten rijden met dummy data blijkt het OV-model daar inderdaad langs te gaan en te stoppen voor een seconde.

### Leren & verbeteren

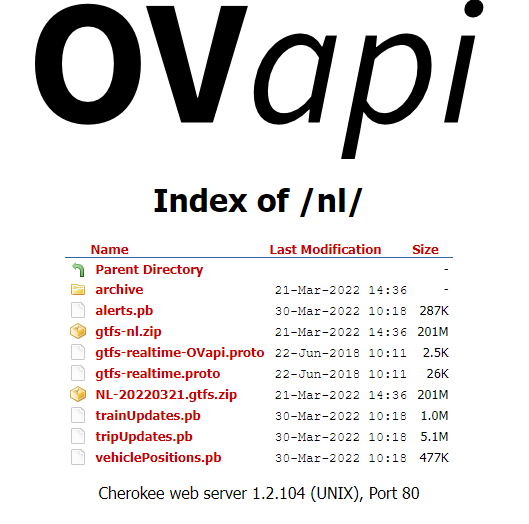
Er is tijd bested aan het onderzoeken welke collider het beste past bij onze objecten. Er is ook documentatie online beschikbaar van Unity zelf. Hier konden we zelf informatie vandaan halen en niet zelf er achter hoefde te komen.

## Werken met Protobuf en GTFS

Doelstelling: Realtime RET-dienstregeling in Visual Studio console applicatie ophalen

### Onderzoek & ontwerp

Protobuf is een bestands formaat waaruit wij onze benodigde data uit moeten halen. Hiervoor zijn er libraries voor ons beschikbaar gemaakt zodat wij makkelijk met Protobuf bestanden kunnen werken in Visual Studio. Naast Protobuf wordt er ook gebruik gemaakt van GTFS. GTFS (General Transit Feed Specification) is een API van google die zal communiceren met de Protobuf bestanden om zo de benodigde data op te halen.



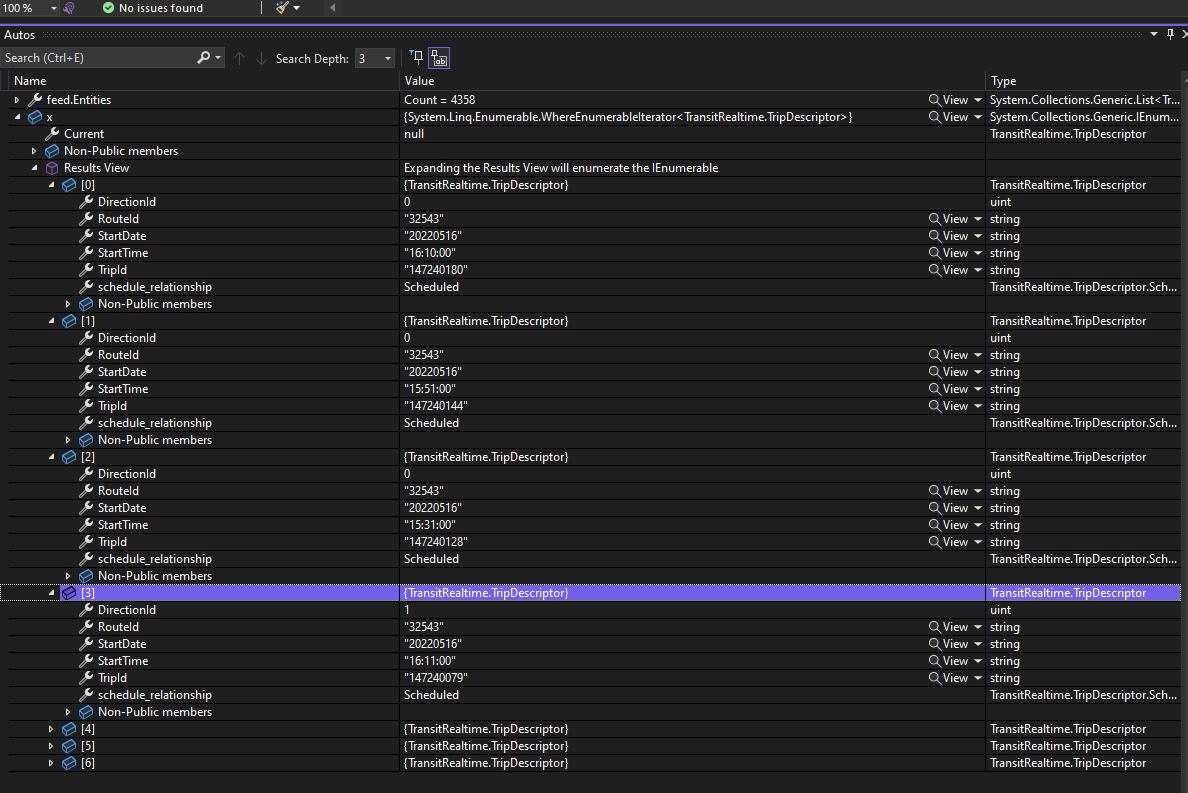
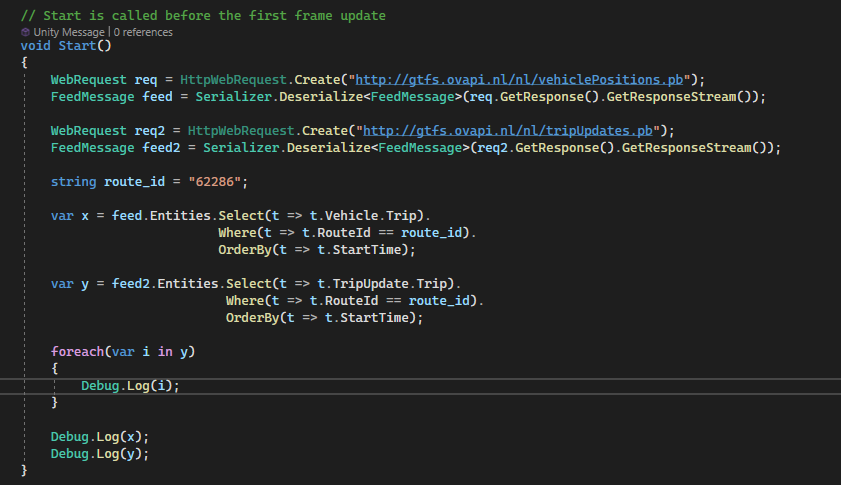
### Test & validatie

De Protobuf en GTFS API zijn allebei succesvol geïnstalleerd via de NuGet packagemanager van Visual Studio. Door nog wat zelfgemaakte code te schrijven hebben wij kunnen filteren om een overzicht te zien van tram 23

### Leren & verbeteren

In de files waaruit wij data terugkrijgen als wij gebruik maken van deze API zijn ‘.pb’ bestanden. Voor deze files moeten wij aparte plug-ins downloaden. Tijdens het downloaden en installeren van deze plug-ins hebben wij veel errors gekregen in Unity versie 2020. Deze hebben wij kunnen oplossen door de dll files van de plug-in aan de asset folder toe te voegen en Unity 2021 te gebruiken. De plug-ins werken alleen in Unity 2021 en niet in 2020.

Visual Studio. Naast Protobuf wordt er ook gebruik gemaakt van GTFS. GTFS (General Transit Feed Specification) is een API van google die zal communiceren met de Protobuf bestanden om zo de benodigde data op te halen.



## Code implementeren in Unity

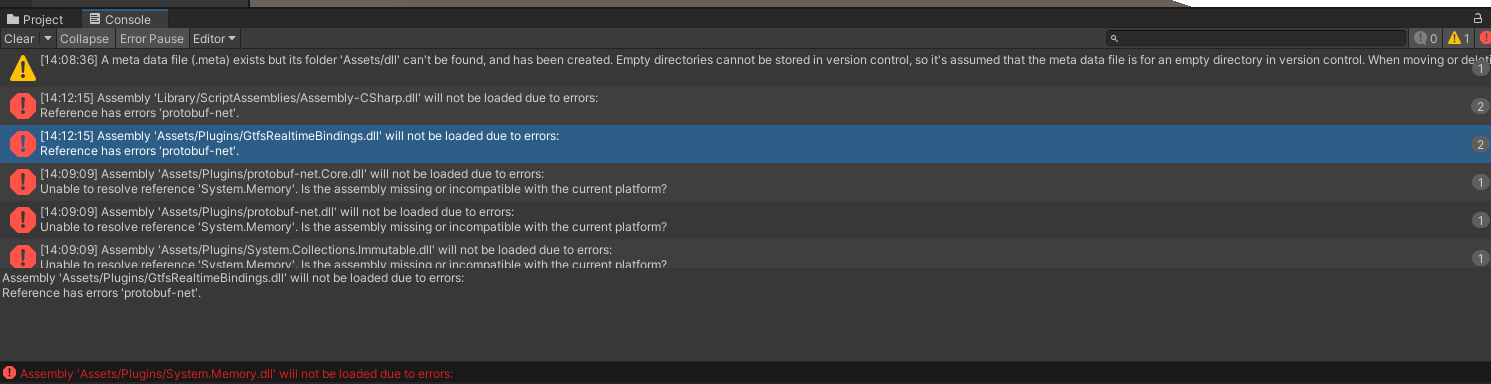
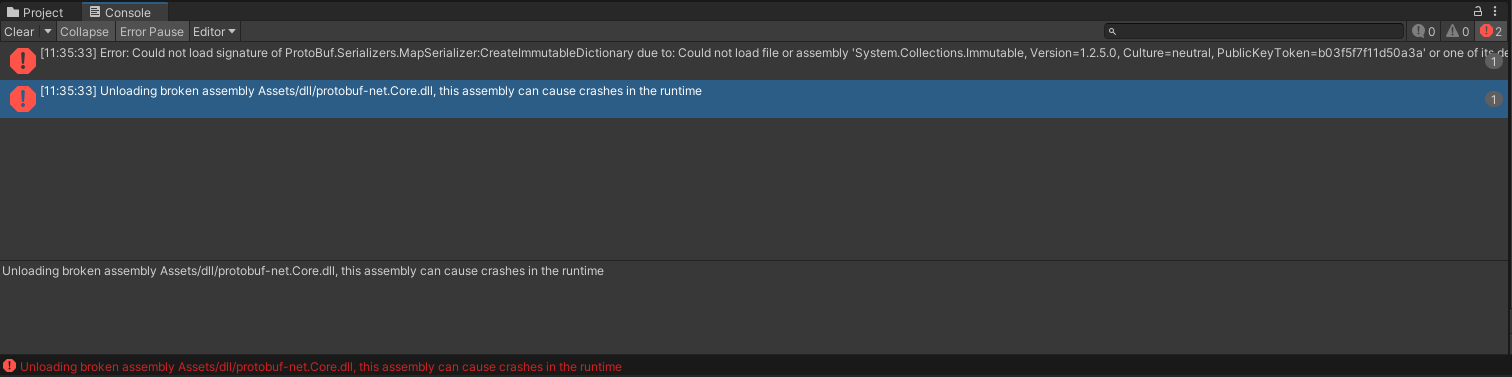
Doelstelling: Code die we in Visual Studio hebben overzetten naar Unity

### Onderzoek & ontwerp

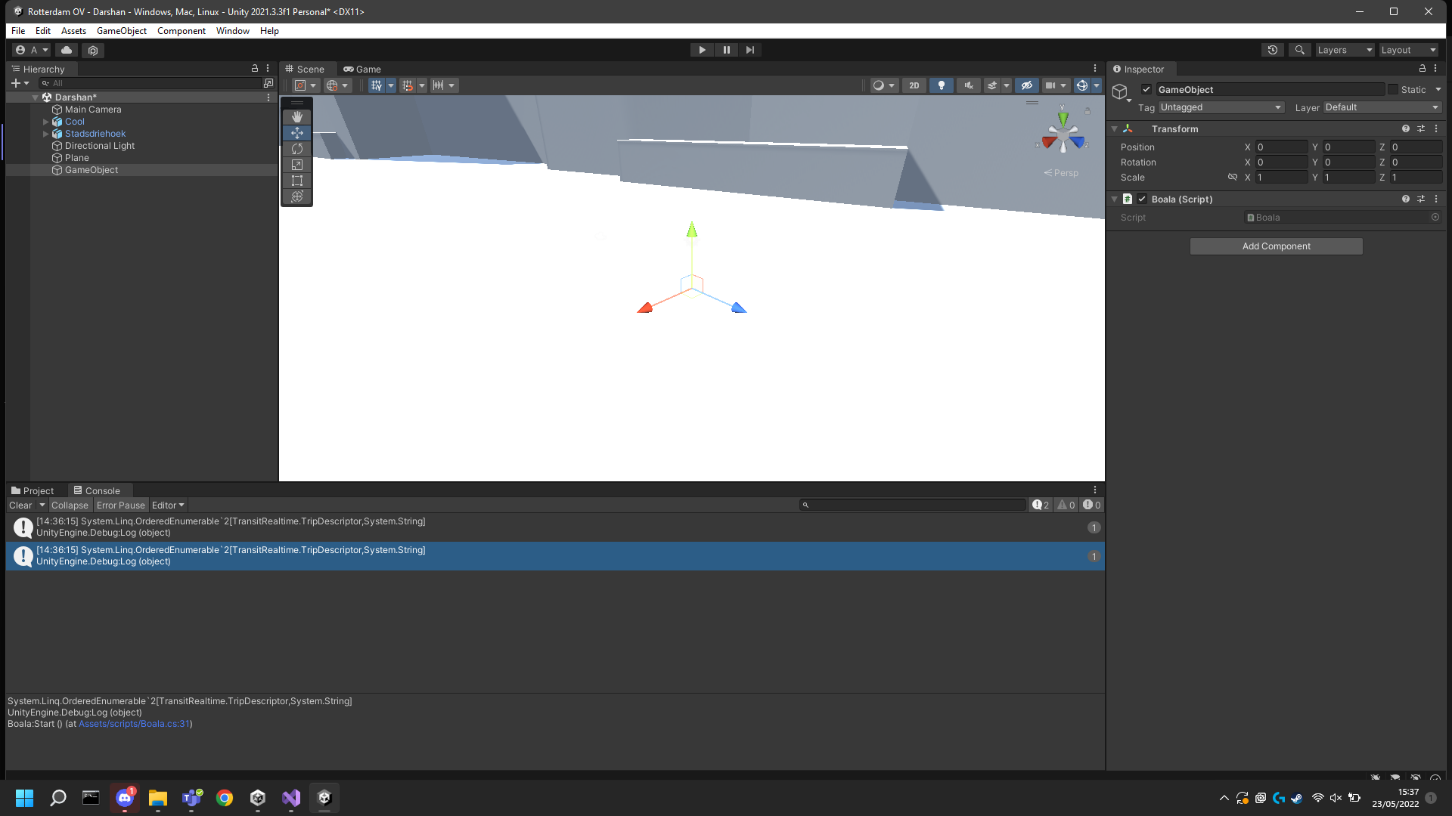
Unity heeft geen ingebouwde ontwikkelomgeving waarin wij code kunnen schrijven en is dus afhankelijk van een externe ontwikkelomgeving. Sinds wij al test code hebben geschreven in een aparte Visual Studio console applicatie hebben wij deze gekopieerd en aan ons Unity project gekoppeld. De externe ontwikkelomgeving waarin wij onze scripts zullen schrijven is Visual Studio.

### Test & validatie

Het overzetten van de code is niet goed gelukt. We kwamen vaak tegen dat de dll bestanden van de plug-ins niet gevonden konden worden en dat ze het daarom niet deden in Unity



Nadat we handmatig de dll's in een plug-in folder hebben gestopt heeft Unity de plug-ins wel kunnen vinden maar, dan geeft hij aan dat de plug-in kapot is en het project kan laten crashen. Dit hebben we met Jack overlegd en hij gaf als tip om naar een van zijn GTFS-projecten te kijken. Zijn GTFS Unity project draaide zonder enige problemen op onze pc's. Uiteindelijk was het probleem dat Jack een andere Unity versie gebruikte. Hij gebruikte 2021 en wij Unity 2020. Toen wij onze Unity hebben omgezet naar versie 2021 werkte alles prima.



## GitHub bij iedereen werkend krijgen

Doelstelling: Version control werkend krijgen

### Onderzoek & ontwerp

Sinds GitHub het meest gebruikte platform is voor version control en mensen in het team daar al ervaring Mee hebben zullen wij hiermee gaan werken. Als git client heeft het team besloten om met GitHub Desktop en Fork te werken. Deze keuzes zijn gemaakt op basis van eerdere ervaringen die het team met deze clients heeft en op aanbevelingen binnen de opleiding.

### Test & validatie

Eenmaal toen de clients gedownload waren werkte Fork prima maar GitHub Desktop bleek nog wat problemen op te leveren. Bij iemand in het team bleef GitHub Desktop maar grafische problemen opleveren waardoor het lezen van tekst erg moeilijk werd. Naast die grafische problemen heeft iedereen wel gewoon de remote master branch kunnen pullen en pushen naar hun eigen remote branches.

### Leren & verbeteren

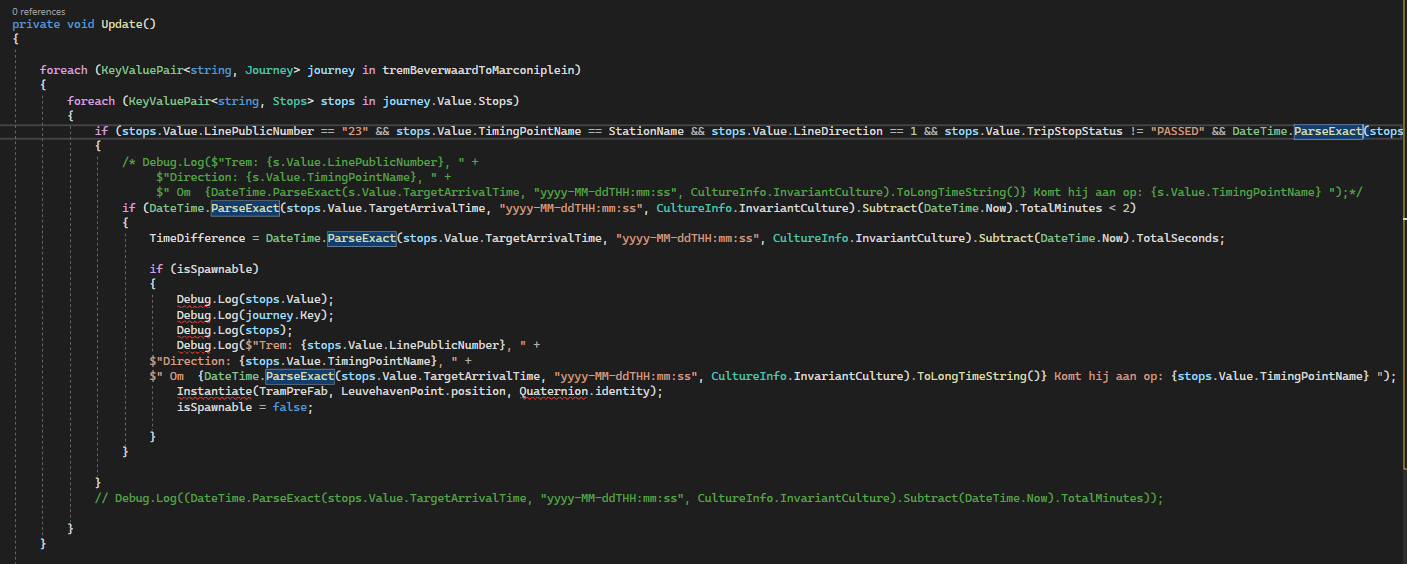
Er is wel ontdekt dat we bestanden in de gitignore bestand moeten zetten die wij niet willen meegeven. Dat zijn voornamelijk bestanden die als lokale database gezien worden en bestanden die hele specifieke gebruiker gegevens opslaan.

## KV78Turbo-OVAPI

Doelstelling: Nieuwe Realtime RET-dienstregeling API

### Onderzoek & ontwerp

Deze data worden opgehaald via v0.ovapi.nl/ waar met een http-get aanvraag een object of overzicht kan worden opgehaald. De gebruikte data komen specifiek van v0.ovapi.nl/journey/ en de vooraf bepaalde objecten hierin. Elk object verwijst naar een real-life twin. Dit wordt aangevraagd via een specifiek code verbonden aan het object. Om de juiste codes te achterhalen wordt er bij het opstarten eerst het overzicht opgehaald. Vervolgens wordt dit gefilterd op basis van lijn en vervoer beheerder wat in dit geval RET is. Om de data zo accuraat mogelijk te houden wordt de data met behulp van API-calls eens per 10 seconden aangevraagd. De API zelf update ongeveer eens per minuut wat het onmogelijk maakt gelijk de data op te halen zonder de API te overbelasten.



### Test & validatie

De OpenOV API data was niet heel duidelijk mee te werken. De meer we het zaten te gebruiken en implementeren de meer onduidelijkheid zich zat op te bouwen. Met de nieuwe Turbo app is de API sturen wij webrequests op dezelfde manier als OpenOV API. Het enige verschil ligt in hoe wij de data ontvangen. De data die wij krijgen van Open OV kwam in een .pb file formaat waarbij wij SQL queries konden schrijven om data te filteren. Het formaat van de data die wij krijgen van de Turbo API komt in een json formaat waarin alles een string of een int is. Wij filteren hier door de data heen door middel van een dictionary. Wij geven hier een value mee om alles met die specifieke value op te slaan.

Deze API werkt iets anders dan de OpenOV API om data uit de bestanden te krijgen. In plaats van queries te schrijven om de data uit de bestanden op te halen maken we nu eerder gebruik van dictionaries. Ook is te zien dat wij hieruit meer informatie kunnen halen. Wat wij voorheen niet uit de

### Leren & verbeteren

OpenOV API konden halen kunnen wij hier wel. Wij kunnen nu te zien krijgen langs welke haltes het OV al is geweest, beter realtime met wanneer hij op een station aankomt, of er vertraging is of niet en hij pakt gelijk een lijst met tijden van de hele dag in plaats van een uur of twee in de toekomst.

## NavMesh

### Onderzoek & ontwerp

Om de figuren die het OV zal representeren te laten bewegen gaan wij gebruik maken van een Unity functie genaamd NavMesh. Met NavMesh kunnen we de figuren apart selecteren en waar hun bestemming zal zijn om te stoppen.



### Test & validatie

Met de NavMesh kan er inderdaad aangegeven worden welk object waar welke richting op gaat. Soms gaat het object een beetje uit zijn baan als het een hele korte bocht moet maken maar voor het grootste gedeelte werkt het prima.

### Leren & verbeteren

NavMesh is een simpele hulpmiddel die ons heeft geholpen om een beter inzicht te krijgen hoe wij objecten kunnen laten bewegen in Unity. Dit is iets wat je in een dag kan op pakken en niet al te lang mee hoeft te zitten. Met NavMesh kunnen wij ook de checkpoints toepassen om ervoor te zorgen dat een object lang seen bepaald punt gaat in de applicatie.

# Conclusie:

Om 3D omgeving te implementeren vn Beurs met realtime RET-dienstregeling, hebben wij als ontwikkelomgeving Unity gekozen. Unity geeft het meeste flexibiliteit om te kiezen met wat voor 3D modellen wij willen werken, plug-ins, libraries en API’s.

De OV-data die wij ontvangen van de KV78Turbo-OVAPI werkt prima samen met Unity, nadat wij nog een externe library hebben gedownload. Deze library heet Newtonsoft Json.net. Met behulp van deze library wordt de data die wij binnen krijgen van de OV-API omgezet naar een bruikbare .net object.

Om de OV-modellen te laten bewegen is er gebruik gemaakt van de AI NavMesh. Dit is een simple hulpmiddel waarmee wij objecten kunnen laten bewegen. Met AI NavMesh kunnen we de objecten checkpoints geven waar ze naar toe moeten gaan en ook hun eindbestemming. De OV-data die ingeladen wordt kunnen we hie rook makkelijk meelinken, zodat de OV-modellen op de juiste aangegeven tijd op het station aankomen en vertrekken.

# Referenties:

Amazon. (2022). *Amazong Sumerian*. Opgehaald van Amazong AWS: https://aws.amazon.com/sumerian/

Blender. (2022, 06 19). *VR Scene Inspection*. Opgehaald van Blender 3.2 Manual: https://docs.blender.org/manual/en/latest/addons/3d\_view/vr\_scene\_inspection.html

CryEngine. (sd). *Virtual Reality*. Opgehaald van CryEngine Getting Started: https://docs.cryengine.com/display/CEMANUAL/Virtual+Reality

Defold. (2022). *The ultimate game engine for web and mobile*. Opgehaald van Defold: https://defold.com/

Gemeente Rotterdam. (sd). *Rotterdam 3D*. Opgehaald van Gemeente Rotterdam: https://www.3drotterdam.nl/#/

Godot. (2019). *VR started tutorial*. Opgehaald van Godot docs: https://docs.godotengine.org/en/3.1/tutorials/vr/vr\_starter\_tutorial.html

Google. (sd). *Build Virtual Worlds*. Opgehaald van Google Developers: https://developers.google.com/vr

Google. (sd). *GTFS Realtime Protobuf*. Opgehaald van Google Transit API's: https://developers.google.com/transit/gtfs-realtime/gtfs-realtime-proto

Google. (sd). *GTFS Static Overview*. Opgehaald van Google Transit API's: https://developers.google.com/transit/gtfs

Google. (sd). *Het Google Maps-platform Documentatie*. Opgehaald van Google Maps Platform: https://developers.google.com/maps/documentation?hl=nl

klaytonkowalski. (2021, 06). *Defold and 3D Support - Discussion*. Opgehaald van Defold Forum: https://forum.defold.com/t/defold-and-3d-support-discussion/68581

NDOV Loket. (sd). *NDOV Loket*. Opgehaald van NDOV Loket RET: http://data.ndovloket.nl/ret/

Oculus. (2016, 12 05). *Medium*. Opgehaald van Meta Quest: https://www.oculus.com/experiences/rift/1336762299669605/

SketchUp. (2022). *SketchUp Viewer for VR*. Opgehaald van Help Center: https://help.sketchup.com/en/sketchup-viewer/sketchup-viewer-vr

Stichting OpenGeo. (2014). *OpenOV*. Opgehaald van OpenOv: https://www.openov.nl/

Unity. (2022, 06 10). *Getting started with VR development in Unity*. Opgehaald van Unity Documentation: https://docs.unity3d.com/Manual/VROverview.html

Unreal Engine. (2022). *Virtual Reality Development*. Opgehaald van Unreal Engine Documentation: https://docs.unrealengine.com/4.26/en-US/SharingAndReleasing/XRDevelopment/VR/