

# Toegankelijkheidsonderzoek

10-06-2023

Versie 3.1

—

Kaan Gögcay

## Versiebeheer

| Versie | Datum      | Wijzigingen  |
|--------|------------|--|
| 1.0    | 07/02/2023 | Alles  |
| 2.0    | 18/04/2023 | Alles  |
| 2.1    | 05/06/2023 | Overige Vraag 2: Is dit wel de...: Voordelen en Nadelen toegevoegd per techniek  |
| 3.0    | 06/06/2023 | <p>Algemeen</p> <ul style="list-style-type: none"><li>- 2<sup>de</sup> hoofdvraag toegevoegd</li><li>- deelvragen anders genummerd</li><li>- minder spel-/grammaticafouten</li><li>- vaker op een nieuwe pagina beginnen waar logisch</li><li>- Bibliografie en bronvermeldingen toegevoegd</li></ul> <p>Deelvraag 2.1</p> <ul style="list-style-type: none"><li>- Compleet nieuwe inhoud</li></ul> <p>Deelvraag 2.2</p> <ul style="list-style-type: none"><li>- anders gestructureerd</li><li>- tekst omgezet in voordelen en nadelen</li><li>- Meer inhoud, nu ook voor- en nadelen van 2d en 3d-omgeving</li><li>- tabel toegevoegd</li></ul> |



|     |            |   |
|-----|------------|---|
| 3.1 | 10/06/2023 | Literatuurlijst<br>- gebruikte bronnen toegevoegd |
| 3.2 | 10/06/2023 | Inhoudsopgave toegevoegd                          |

## Inhoud

|  |    |
|--|----|
| .....  | 0  |
| Versiebeheer .....   | 1  |
| Inleiding.....   | 6  |
| Context .....  | 7  |
| Deelvragen .....   | 8  |
| Deelvraag 1.1: Hoe maken we de Level Editor zo toegankelijk mogelijk voor docenten?                      | 8  |
| Deelvraag 1.2: Hoe maken we het speelveld zo toegankelijk mogelijk voor studenten?                       | 8  |
| Deelvraag 1.3: Tot in hoeverre moeten en kunnen we rekening houden met studenten met een beperking?..... | 8  |
| Deelvraag 2.1: Wat is de meest geschikte codeertaal en framework om te gebruiken? .                      | 8  |
| Deelvraag 2.2: Is dit wel de beste oplossing voor het probleem?.....                                     | 8  |
| Deelvraag 1.1: Hoe maken we de Level Editor zo toegankelijk mogelijk voor docenten?.....                 | 9  |
| Wanneer is een applicatie gebruikersvriendelijk?.....  | 9  |
| Intuïtief en duidelijk ontwerp .....   | 9  |
| Consistente gebruikersinterface.....   | 9  |
| Duidelijke en beknopte taal.....   | 9  |
| Toegankelijkheid.....  | 9  |
| Foutafhandeling .....  | 10 |
| Gebruikersfeedback.....  | 10 |
| Hoe maken we de Level Editor gebruikersvriendelijk?.....   | 10 |
| Feedback Loop.....   | 10 |
| Deelvraag 1.2: Hoe maken we het speelveld zo toegankelijk mogelijk voor studenten? .....                 | 12 |
| Wanneer is een applicatie gebruikersvriendelijk?.....  | 12 |
| Hoe maken we het speelveld gebruikersvriendelijk? .....  | 12 |
| Deelvraag 1.3: Tot in hoeverre moeten en kunnen we rekening houden met studenten met een beperking?..... | 13 |

|   |    |
|---|----|
| Geen beperkingen op de opleiding .....  | 13 |
| Kleurenblind .....  | 13 |
| Deelvraag 2.1: Wat is de meest geschikte codeertaal en framework om te gebruiken .....          | 14 |
| Framework + Codeertaal .....  | 14 |
| WinForms + C# .....   | 14 |
| Phaser + Javascript .....   | 15 |
| Wellicht kan ChatGPT helpen? .....  | 15 |
| Unity + C# .....  | 15 |
| Deelvraag 2.2: Is dit wel de beste oplossing voor het probleem? .....                           | 16 |
| 3D Omgeving .....   | 17 |
| Denk bij een 3D-omgeving aan games zoals Zelda, Minecraft en Grand Theft Auto. ....             | 17 |
| Voordelen .....   | 17 |
| Nadelen .....   | 17 |
| 2D Omgeving .....   | 18 |
| Denk bij een 2D omgeving aan games zoals: Retro Pokemon Games, Tetris en Super Mario Bros. .... | 18 |
| Voordelen .....   | 18 |
| Nadelen .....   | 18 |
| Virtual Reality / Mixed Reality .....   | 19 |
| Voordelen .....   | 19 |
| Nadelen .....   | 19 |
| Augmented Reality .....   | 20 |
| Voordelen .....   | 20 |
| Nadelen .....   | 20 |
| Resultaten .....  | 21 |
| Literatuurlijst .....   | 22 |





## Inleiding

In dit document wil ik onderzoeken hoe ik mijn applicatie zo toegankelijk mogelijk kan maken voor alle eindgebruikers. Hieruit volgt de hoofdvraag van het onderzoek: Hoe kan ik de applicatie zo toegankelijk mogelijk maken voor alle eindgebruikers? Ook is er tijdens mijn stage een tweede hoofdvraag bijgekomen: Wat is de beste techniek en methode om de oplossing te realiseren?



## Context

In mijn opdracht moet ik een Serious-Game maken voor autotechniek studenten op het MBO. Deze Serious-Game wordt uiteindelijk bruikbaar voor zowel studenten als docenten. Docenten kunnen zelf levels bouwen voor hun studenten en studenten kunnen deze levels vervolgens spelen. Maar waarom doe ik dit?

De Serious-Game die ik ga maken is een oplossing op het probleem van mijn stakeholder Tim. Tim is een docent op de autotechniek MBO. Op deze MBO is er maar één praktijklokaal om met studenten één op één handelingen te beoefenen. Omdat er maar 1 praktijklokaal is duurt het lang om meerdere studenten één voor één een één op één les te geven.

Tim wil dit probleem oplossen door middel van een Serious-Game.



## Deelvragen

Om mijn hoofdvragen te beantwoorden, gaan we eerst een aantal deelvragen beantwoorden. Door het beantwoorden van deze deelvragen kunnen we uiteindelijk de hoofdvraag beantwoorden. De deelvragen van de hoofdvraag zijn: Hoe kan ik de applicatie zo toegankelijk mogelijk maken voor alle eindgebruikers?

Deelvraag 1.1: Hoe maken we de Level Editor zo toegankelijk mogelijk voor docenten?

Door deze vraag te beantwoorden, komen we erachter hoe we de applicatie zo toegankelijk mogelijk kunnen maken voor docenten.

Deelvraag 1.2: Hoe maken we het speelveld zo toegankelijk mogelijk voor studenten?

Door deze vraag te beantwoorden, komen we erachter hoe we de applicatie zo toegankelijk mogelijk kunnen maken voor studenten.

Deelvraag 1.3: Tot in hoeverre moeten en kunnen we rekening houden met studenten met een beperking?

Ik heb de mogelijkheid om extra functionaliteiten toe te voegen, zodat mensen met een beperking die normaal de Serious Game niet zouden kunnen spelen, wél kunnen spelen. Maar tot in hoeverre is dit nodig? Daar komen we achter door de huidige deelvraag te beantwoorden.

Deelvragen van de hoofdvraag: Wat is de beste techniek en methode om de oplossing te realiseren?

Deelvraag 2.1: Wat is de meest geschikte codeertaal en framework om te gebruiken?

Hier kijk ik naar verschillende codeertalen en codeeromgevingen en maak ik uiteindelijk een keuze die het meest geschikt is voor mijn project.

Deelvraag 2.2: Is dit wel de beste oplossing voor het probleem?

Misschien is er wel een betere oplossing. Als dat zo is, is het belangrijk om dit aan te kaarten aan mijn stakeholder. Zo niet, kun je een hele applicatie maken waar je uiteindelijk niets aan hebt.

Nu dat we weten waarom we deze vragen beantwoorden, kunnen we beginnen met het beantwoorden van de deelvragen.

## Deelvraag 1.1: Hoe maken we de Level Editor zo toegankelijk mogelijk voor docenten?

In deze deelvraag staan 3 termen die we eerst moeten begrijpen.

**Docenten:** MBO autotechniek docenten. Hieronder valt ook mijn stakeholder Tim.

**Level Editor:** Een omgeving waarin docenten levels kunnen maken voor hun studenten. In de praktijk zijn er ontzettend veel taken die we kunnen digitaliseren en kunnen opnemen in ons programma. Natuurlijk is het niet wenselijk dat je in een level elke mogelijke taak moet uitvoeren. Daarom leggen we de keus bij de docent, zodat de docent zelf kan kiezen wanneer zijn studenten een bepaalde taak leren.

**Zo toegankelijk mogelijk:** Dit houdt in zó gebruikersvriendelijk mogelijk. Maar wanneer is een applicatie gebruikersvriendelijk?

### Wanneer is een applicatie gebruikersvriendelijk?

Door middel van onderzoek ben ik tot het volgende gekomen.

#### Intuïtief en duidelijk ontwerp

Een gebruiksvriendelijke applicatie moet een ontwerp hebben dat intuïtief en gemakkelijk te navigeren is. Gebruikers moeten snel de lay-out en functionaliteit van de applicatie kunnen begrijpen.

#### Consistente gebruikersinterface

Een gebruiksvriendelijke applicatie moet een consistente gebruikersinterface hebben, met knoppen, menu's en andere elementen die op voorspelbare locaties in de applicatie zijn geplaatst.

#### Duidelijke en beknopte taal

De taal die in de applicatie wordt gebruikt, moet duidelijk en beknopt zijn en technisch jargon of verwarrende terminologie vermijden.

#### Toegankelijkheid

Een gebruiksvriendelijke applicatie moet toegankelijk zijn voor alle gebruikers, inclusief mensen met een handicap. Dit omvat functies zoals toetsenbordnavigatie, ondersteuning voor schermlezers en hoog-contrastmodi.

## Foutafhandeling

De applicatie moet fouten op een elegante manier afhandelen en duidelijke foutmeldingen geven die gebruikers helpen begrijpen wat er misging en hoe het te herstellen.

## Gebruikersfeedback

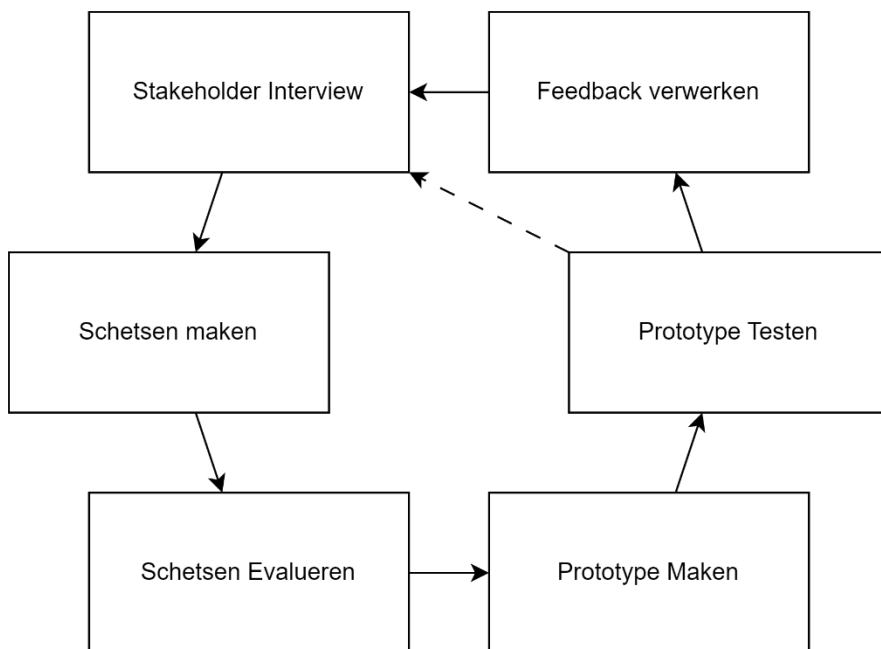
Gebruikersfeedback is essentieel voor het creëren van een gebruiksvriendelijke applicatie. Ontwikkelaars moeten actief feedback van gebruikers zoeken en deze gebruiken om de applicatie in de loop van de tijd te verbeteren.

(ChatGPT, What makes an application user friendly?, "What makes an application user friendly?")

## Hoe maken we de Level Editor gebruikersvriendelijk?

### Feedback Loop

Ik heb een proces dat ik wekelijks probeer te herhalen. In dit proces komen de 6 punten hierboven telkens ook weer terug om een bepaald niveau aan te houden. Dit proces kan het beste uitgelegd worden met een diagram.



### Stakeholder Interview

Het begint allemaal met een interview met de stakeholder. In dit interview probeer ik zoveel mogelijk te weten te komen over zijn wensen en ideeën. Ook probeer ik te begrijpen hoe bepaalde processen in de autotechniek in zijn werking gaan. Zo leer ik bijvoorbeeld de theorie over het verwisselen van een wiel van een auto.

### Schetsen Maken

In deze stap ga ik schetsen maken van nog niet toegevoegde functionaliteiten. Deze functionaliteiten zijn gebaseerd op wat ik te weten ben gekomen in het stakeholder interview.

### Schetsen Evalueren

Na het maken van de schetsen ga ik deze evalueren met mijn stakeholder. Uit deze evaluatie komt naar voren welke ideeën wel en niet bevallen. Ook geef ik de stakeholder ruimte om mee te denken met ideeën en functionaliteiten.

### Prototype Maken

De ideeën die uit de evaluatie zijn gekomen, ga ik vervolgens uitwerken in mijn applicatie.

### Prototype Testen

Nu ik een prototype heb, kan ik deze gaan testen met docenten. Tijdens deze test kijk ik naar het gedrag van de tester. Het doel van deze test is om te achterhalen of mijn applicatie duidelijk en begrijpelijk is zonder externe hulp. Wat ook helpt is na afloop vragen wat de gebruiker ervan vond en waar hij tegenaan liep.

### Feedback Verwerken

In dit proces ga ik de feedback die ik heb gekregen van mijn tester verwerken. Als ik twijfel over de feedback die ik heb gekregen, dan kan ik dit altijd in de volgende stap weer bespreken met mijn stakeholder in het stakeholder interview. Zoals je ziet, heb ik een stippellijn getrokken van *Prototype Testen* naar *Stakeholder Interview*. Dit houdt in dat we de stap feedback verwerken kunnen overslaan als er geen bruikbare feedback is gekomen uit de test.

Door rekening te houden met de 7 punten die ChatGPT heeft gegenereerd en de feedback loop kunnen we een gebruikersvriendelijke applicatie bouwen.

## Deelvraag 1.2: Hoe maken we het speelveld zo toegankelijk mogelijk voor studenten?

**Speelveld:** Het scherm waarin studenten de game spelen.

**Studenten:** MBO-autotechniekstudenten.

**Zo toegankelijk mogelijk:** Dit houdt in dat het zo gebruiksvriendelijk mogelijk moet zijn. Maar wanneer is een applicatie gebruiksvriendelijk?


De huidige deelvraag lijkt erg veel op deelvraag 1 en overlapt grotendeels.

### Wanneer is een applicatie gebruikersvriendelijk?

Zie deelvraag 1.

### Hoe maken we het speelveld gebruikersvriendelijk?

Dit is bijna exact hetzelfde als de vraag *“Hoe maken we de Level Editor zo toegankelijk mogelijk?”*, maar bij deze deelvraag gaan we alles testen op studenten.



### **Deelvraag 1.3: Tot in hoeverre moeten en kunnen we rekening houden met studenten met een beperking?**

Om deze vraag te beantwoorden heb ik een kort gesprek gehad met mijn stakeholder. Uit dit gesprek heb ik het volgende geconcludeerd.

#### **Geen beperkingen op de opleiding**

Op de opleiding Autotechniek komen bijna nooit mensen met een beperking voor. Dus het houdt eigenlijk al vrij snel op.

#### **Kleurenblind**

De enige beperking die af en toe voorkomt, is kleurenblindheid bij studenten. Maar zelfs hiervoor gaf de stakeholder aan dat we er geen rekening mee hoeven te houden, omdat de studenten in praktijklessen en in de praktijk geen hulpmiddelen hiervoor krijgen.

Dus kort gezegd, nee, we hoeven geen rekening te houden met studenten met beperkingen.

## Deelvraag 2.1: Wat is de meest geschikte codeertaal en framework om te gebruiken

**Codeertaal:** De taal waarin de code is geschreven. Het wordt meestal gebruikt in combinatie met een framework om het coderen sneller te laten verlopen.

**Framework:** Een codeerframework is een gestructureerde set van regels en richtlijnen die worden gebruikt om efficiënt softwarecode te schrijven. Het biedt ontwikkelaars een basis om mee te werken en helpt hen om consistente en betrouwbare code te schrijven. Een codeerframework kan ook standaardbibliotheken, hulpmiddelen en sjablonen bevatten om het ontwikkelingsproces te versnellen en te vergemakkelijken.

**Meest geschikt:** Dit hangt af van meerdere factoren. Een codeertaal en framework kunnen geschikt zijn voor mij als ik er veel ervaring mee heb. Maar je moet ook kijken naar eventuele beperkingen. Kan ik er alles mee doen wat ik moet doen? Is het een bekende codeertaal en framework? Als dat het geval is, is er veel informatie online beschikbaar en kan ik altijd hulp vinden.

### Framework + Codeertaal

#### WinForms + C#

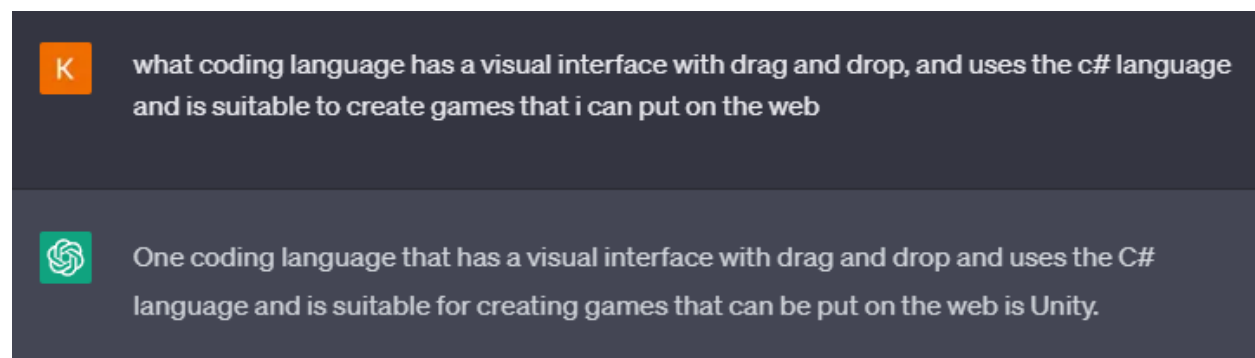
Een persoonlijke favoriet van mij is WinForms. Naast het feit dat ik er zelf veel ervaring mee heb, bestaat WinForms al voor een langere tijd. Hierdoor is het eenvoudig om zelf problemen op te lossen. Een ander groot voordeel is dat WinForms gebruikmaakt van C#. C# is ook een vrij toegankelijke codeertaal. Het wordt al jaren gebruikt en het is makkelijk om er online hulp over te vinden. Om de applicatie zo toegankelijk mogelijk te maken hebben ik en de stakeholder besloten om voor een website vorm te gaan in plaats van een desktop applicatie. Aangezien het onmogelijk is om WinForms te embedden, in een website valt WinForms weg.

## Phaser + Javascript

Veel games die op websites worden gehost, worden gemaakt in HTML5. Dit bracht mij bij het grootste HTML5-gameframework, Phaser. Phaser heeft een eigen Discord-server voor ondersteuning en biedt online documentatie. Het grote nadeel is dat ik geen ervaring heb met JavaScript. Desondanks heb ik het framework uitgeprobeerd. Het viel me op dat Phaser geen visuele interface heeft voor het ontwerpen van games. Dit vertraagt het ontwikkelproces, omdat het plaatsen van objecten op de juiste plek een kwestie wordt van trial-and-error.

## Wellicht kan ChatGPT helpen?

Ik heb gekeken naar de voor- en nadelen van WinForms en Phaser en heb zo de prompt opgesteld die een codeertaal en omgeving zou moeten geven die perfect geschikt is volgens mijn eisen.



*Figuur 1: Ik stel ChatGPT een vrij specifieke vraag.*

## Unity + C#

Unity heeft ook een Discord-server waar je altijd terecht kunt met al je vragen. Daarnaast is Unity erg bekend, dit betekent dat je buiten de Discord-server ook gemakkelijk antwoorden kunt vinden door te zoeken op het internet. Unity maakt gebruik van C#. C# heeft ook een grote community. Al met al is Unity + C# niet alleen technisch gezien een geschikte keuze voor mijn opdracht, maar ook een veilige keuze wat betreft community-ondersteuning..





## Deelvraag 2.2: Is dit wel de beste oplossing voor het probleem?

**Beste oplossing:** De oplossing waar we uiteindelijk het meeste aan zullen hebben, oftewel de meest nuttige oplossing.

**XR:** Een verzamelnaam voor VR, AR en MR.

Laten we eerst naar de vraag kijken van de stakeholder “Ik wil een 3D-omgeving waarin je de taken in kunt uitvoeren en wil er XR in verwerkt hebben”. Het verzoek van mijn stakeholder is erg specifiek. Hier zijn in het kort de voor- en nadelen van elke techniek:

## 3D Omgeving

Denk bij een 3D-omgeving aan games zoals Zelda, Minecraft en Grand Theft Auto.



*Figuur 2: Afbeelding uit de game Zelda: Majora's Mask*

### Voordelen

**Realistische visuele weergave:** In een 3D-omgeving kunnen objecten en omgevingen realistischer worden weergegeven, met behoud van diepte, textuur en belichting.

**Realistische interactie:** In een 3D-omgeving kunnen gebruikers objecten in de ruimte manipuleren en ermee interageren op een manier die nauwer aansluit bij de fysieke wereld.

### Nadelen

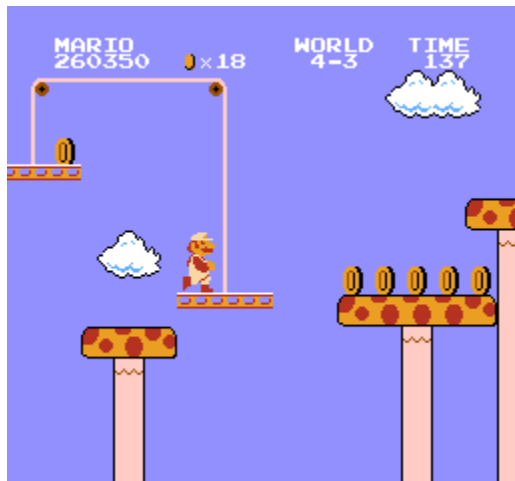
**Complexer ontwikkelingsproces:** Het maken van een 3D-omgeving vereist meer inspanning en expertise, omdat het omvat modellering, texturering, animatie en mogelijk complexere programmeerlogica.

**Hogere systeemvereisten:** 3D-graphics en interacties kunnen veeleisender zijn voor de hardware van een apparaat, waardoor gebruikers mogelijk krachtigere computers of grafische kaarten nodig hebben.

**Langere ontwikkelingstijd:** Het maken van 3D-assets en het ontwikkelen van functionaliteit in een 3D-omgeving kan meer tijd in beslag nemen dan in een 2D-omgeving, (ChatGPT, "Kun je ook de voor en nadelen geven van als we dit project in een 2d of 3d omgeving maken in unity?")

## 2D Omgeving

Denk bij een 2D omgeving aan games zoals: Retro Pokemon Games, Tetris en Super Mario Bros.



*Figuur 3: Afbeelding uit de game Super Mario Bros.*

### Voordelen

**Eenvoudiger ontwikkelingsproces:** Het maken van een 2D-omgeving kan over het algemeen eenvoudiger zijn dan het maken van een 3D-omgeving, omdat het minder complexiteit met zich meebrengt in termen van modellen, texturen en animaties.

**Lagere systeemvereisten:** 2D-graphics zijn over het algemeen minder intensief voor de hardware, waardoor het gemakkelijker toegankelijk is voor gebruikers met minder krachtige apparaten.

**Snellere ontwikkelingstijd:** Het maken van assets en het implementeren van functionaliteit in een 2D-omgeving kan vaak sneller worden gedaan dan in een 3D-omgeving, waardoor de ontwikkelingstijd mogelijk wordt verkort.

### Nadelen

**Beperkte visuele diepte:** In een 2D-omgeving ontbreekt de visuele diepte en het gevoel van diepte dat 3D-omgevingen bieden. Dit kan invloed hebben op de realistische weergave van objecten en omgevingen.

**Minder realistische interactie:** In een 2D-omgeving kan interactie met objecten beperkter zijn, omdat de beweging vaak beperkt is tot een enkel vlak.

(ChatGPT, "Kun je ook de voor en nadelen geven van als we dit project in een 2d of 3d omgeving maken in unity?")

## Virtual Reality / Mixed Reality

VR en MR delen functionaliteiten, voordelen en nadelen. Daarom heb ik ze samen gepakt.

### Voordelen

**Volledige immersie:** VR plaatst gebruikers in een volledig virtuele omgeving, waardoor ze zich volledig kunnen onderdompelen in een gesimuleerde wereld zonder afleidingen van de echte wereld.

**Realistische ervaring:** Met VR kunnen gebruikers virtuele objecten manipuleren en ermee interageren alsof ze fysiek aanwezig zijn, waardoor realistische trainings- en leerervaringen mogelijk zijn.

**Mogelijkheid tot simulatie:** VR stelt gebruikers in staat om complexe scenario's te simuleren die moeilijk te repliceren zijn in de echte wereld, zoals ruimtevluchten, medische procedures of industriële training.

### Nadelen

**Kosten:** VR-apparatuur, zoals headsets en controllers, kan duur zijn, vooral als er meerdere exemplaren moeten worden aangeschaft voor gebruik in educatieve instellingen.

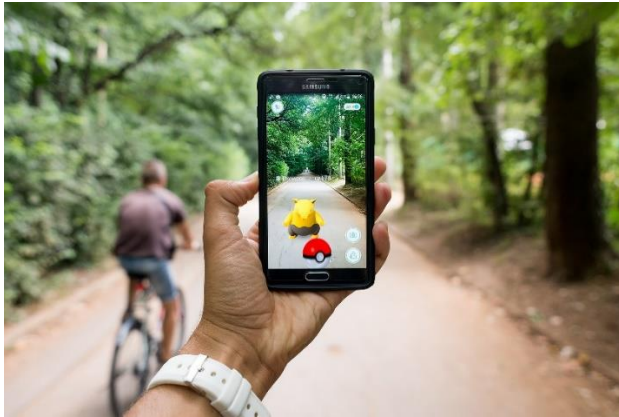
**Beperkte mobiliteit:** VR vereist vaak speciale apparatuur en vereist dat gebruikers zich binnen een bepaalde ruimte begeven, wat de mobiliteit kan beperken in vergelijking met AR, dat kan worden gebruikt met behulp van draagbare apparaten.

**Potentiële gezondheidsproblemen:** Sommige gebruikers kunnen last hebben van bewegingsziekte of vermoeidheid bij langdurig gebruik van VR-headsets, vanwege het verschil tussen de bewegingen in de virtuele en echte wereld.

(ChatGPT, "Kun je van de technieken AR en VR kort samenvatten wat de voor en nadelen zijn?")

## Augmented Reality

Denk bij AR aan games zoals: Pokemon GO, Ingress en Minecraft Earth



*Figuur 4: Foto waarin iemand Pokémon Go speelt op zijn telefoon*

### Voordelen

**Toegankelijkheid:** AR-applicaties kunnen worden uitgevoerd op smartphones, tablets of speciale AR-brillen, waardoor het toegankelijk is voor een breed publiek met deze apparaten.

**Lagere kosten:** In vergelijking met VR zijn AR-apparaten vaak goedkoper, omdat ze gebruikmaken van bestaande technologie zoals smartphones.

**Behoud van omgevingsbewustzijn:** AR overlapt digitale informatie met de echte wereld, waardoor gebruikers hun omgeving kunnen blijven zien en interactie kunnen hebben met de fysieke objecten om hen heen.

### Nadelen

**Beperkte immersie:** Hoewel AR digitale elementen toevoegt aan de echte wereld, is de ervaring minder meeslepend dan VR, omdat de echte wereld nog steeds zichtbaar blijft.

**Technologische beperkingen:** AR-technologie is nog in ontwikkeling, dus de precisie en betrouwbaarheid kunnen variëren. Soms kunnen trackingfouten optreden, wat kan leiden tot onnauwkeurige of ongewenste overlappende digitale objecten.

**Beperkte interactie:** In AR kunnen gebruikers digitale objecten vaak niet fysiek manipuleren of ermee interacteren, omdat ze alleen virtueel aanwezig zijn en geen tastbare eigenschappen hebben.

(ChatGPT, "Kun je van de technieken AR en VR kort samenvatten wat de voor en nadelen zijn?")

## Resultaten

|         | <b>Voordelen</b>   | <b>Nadelen</b>  |
|---------|--|---|
| 2D      | <ul style="list-style-type: none"> <li>- Snellere ontwikkelingstijd</li> <li>- Eenvoudiger ontwikkelingsproces</li> <li>- Lagere systeemvereisten</li> </ul> | <ul style="list-style-type: none"> <li>- Beperkte visuele diepte,</li> <li>- Minder realistische interactie</li> </ul>                                    |
| 3D      | <ul style="list-style-type: none"> <li>- Realistische visuele weergave</li> <li>- Realistische interactie</li> </ul>   | <ul style="list-style-type: none"> <li>- Langere ontwikkelingstijd</li> <li>- Complexer ontwikkelingsproces</li> <li>- Hogere systeemvereisten</li> </ul> |
| VR / MR | <ul style="list-style-type: none"> <li>- Volledige immersie</li> <li>- Realistische ervaring</li> <li>- Mogelijkheid tot simulatie</li> </ul>                | <ul style="list-style-type: none"> <li>- Kosten</li> <li>- Beperkte mobiliteit</li> <li>- Potentiële gezondheidsproblemen</li> </ul>                      |
| AR      | <ul style="list-style-type: none"> <li>- Toegankelijkheid</li> <li>- Lagere kosten</li> <li>- Behoud van omgevingsbewustzijn</li> </ul>                      | <ul style="list-style-type: none"> <li>- Beperkte immersie</li> <li>- Technologische beperkingen</li> <li>- Beperkte interactie</li> </ul>                |

## Literatuurlijst

ChatGPT. (2022, 30 november). OpenAI. <https://chat.openai.com/>

Join the phaser Discord Server! (z.d.). Discord. <https://discord.com/invite/phaser>

Phaser 3 API Documentation - Index. (z.d.). <https://photonstorm.github.io/phaser3-docs/>

Phaser - Phaser 3 API Documentation (beta). (z.d.). <https://newdocs.phaser.io/docs/3.55.2/Phaser>

Photon Storm - photonstorm.com. (z.d.). Making your first Phaser 3 game: Please select a language - Learn - Phaser.

<https://web.archive.org/web/20230330053710/http://phaser.io/tutorials/making-your-first-phaser-3-game>

Technologies, U. (z.d.). Unity - Manual: Unity User Manual 2021.3 (LTS).

<https://docs.unity3d.com/Manual/index.html>

Join the Official Unity Discord Discord Server! (z.d.). Discord. <https://discord.com/invite/unity>