# Documentatie Project WebTopics Basic

## Leerproces

Hieronder zal ik beschrijven hoe en wat ik geleerd heb en wat de struikelblokken waren in het proces voor het maken van het project.

### Het idee

Mijn idee was om een game te maken rond pokemon. In het start scherm kies je een pokemon die je wil vangen. Dan ga je over naar de AR-scene waar je pokeballs kan gooien naar de betreffende pokemon waarna deze dan verdwijnt in de pokeball zogezegd wanneer je raakt gooit.

Ik wist al dat A-frame een goede start positie was om het project te beginnen dus eerst ben ik wat beginnen spelen met de ‘getting started’ van op hun website en zo verder beginnen bouwen en andere dingen opzoeken.

## Problemen waar ik ben op gestoten

Hieronder som ik wat problemen op die ik tegenkwam, sommige al moeilijker op te lossen dan de andere.

* Ik wist niet dat met AR.js je enkel markers kon gebruiken om bepaalde objecten op het beeld te krijgen en aangezien ik gewoon een pokemon in de AR-scene wou krijgen was het dus niet nodig om met een marker te werken. Ook wou ik het echt zelf proberen om markerless te werken. Wat achteraf gezien misschien niet de makkelijkste keuze zou geweest zijn.
* Online had ik artikels gevonden over “AR hit testing”. Dit is wanneer je tikt op een vlak in de AR-scene (bijvoorbeeld de grond), dat de software dit ook ziet als een vlak en vervolgens je gewenste object hierop zal plaatsen. Dit is dus markerless. Het is echter niet zo simpel en als er al deftige voorbeelden online te vinden waren was het zeer moeilijk om deze te implementeren in mijn project. Via een ander project heb ik dan een library gevonden maar deze blijkt niet altijd heel goed te werken.
* Om het 3D-model van de pokemon in de scene te krijgen heb ik ook wat geleerd. Op de website van a-frame bevelen ze namelijk gltf-bestanden aan. Maar dit blijkt niet altijd te werken en is redelijk moeilijk om in je project te krijgen. De makkelijkste oplossing is via een website de gltf-bestanden om te zetten een glb-bestand. Deze zijn makkelijker te gebruiken.
* Met de nieuwe versie van a-frame zijn er wat aanpassingen gebeurd voor de physics component (waarbij objecten bijvoorbeeld een static of dynamic body krijgen zodat ze ergens kunne tegen botsen). Dit was voor mij belangrijk omdat ik de pokeballs zou moeten gooien naar de pokemon en deze zou moeten wegspringen bij contact. Het is me niet gelukt om dit tot een goed einde te brengen jammer genoeg. Wanneer in de scene krijg ik nooit pokeballs te zien en ik krijg foutmeldingen als ik een static of dynamic body wil toevoegen aan een object.

## Bronnen die me verder hebben geholpen

* <https://aframe.io/blog/arjs/>
* <https://jeromeetienne.github.io/AR.js/aframe/>
* <https://aframe.io/docs/1.2.0/guides/building-a-basic-scene.html>
* <https://aframe.io/docs/1.2.0/components/webxr.html>
* <https://aframe.io/docs/1.2.0/components/gltf-model.html>
* <https://jgbarah.github.io/aframe-playground/physics-01/>
* <https://github.com/immersive-web/webxr-samples/blob/main/hit-test.html>
* <https://web.dev/ar-hit-test/#entering-a-session>
* <https://medium.com/samsung-internet-dev/making-an-ar-game-with-aframe-529e03ae90cb>
* <https://learn.framevr.io/project1/part3>
* <https://github.com/AdaRoseCannon/basketball-demo>

Als ik terugkijk op het project had ik me misschien meer moeten focussen op het effectief gooien van de pokeballs en niet zo zeer aan de pokemon die aan de hand van ‘hit-testing’ op een bepaald surface kan weergegeven worden. Hierdoor heb ik wat tijd verloren en ben op het einde niet meer klaar geraakt. Ook het zoeken om te werken met markers in het begin heb ik toch nog op wat zitten zoeken om in orde krijgen en de pokemon in de scene te krijgen zonder marker.

## Stappenplan/tutorial

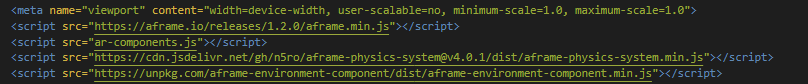
### Stap1: hoofdpagina

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Startpagina is simpele html en css waarbij foto’s en links worden gebruikt om door te gaan naar de pagina van de gekozen pokemon.

### Stap 2: scripts importeren



Importeer de nodige scripts. De eerste is de nieuwste versie van a-frame, dan een online gevonden script voor het ‘hit-testen’ (die niet helemaal naar behoren werkt). Vervolgens een script voor de physics en als laatste een script om een prebuild ruimte te maken.

### Stap 3: <a-scene> toevoegen aan de body

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Hier worden de verschillende objecten aangemaakt die in de scene moeten komen maar ook de nodige zoals een camera en licht. Door het gebruik te maken van assets wacht de scene op het laden van deze assets voordat er gerendert wordt.

### Stap 4: Javascript toevoegen

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Variabelen declareren

Afbeelding met tekst

Automatisch gegenereerde beschrijving

DOMoverlay is eig om bijvoorbeeld een 2D tekst te plaatsen in een 3D-scene. Hier wordt er gecheckt of dit kan worden gebruikt of niet.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Deze EventListeners worden gebruikt door de knoppen om de scene binnen te komen of ze te verlaten.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Een functie voor het plaatsen van de pokemon op de plaats waar de recticle zich bevindt.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

De bovenste Eventlistener wordt getriggerd bij een select, deze is bedoelt voor het gooien van de pokeballs maar spijtig genoeg gebeurdt er niets wanneer de pokemon zich op een plek bevindt. De onderste functie is om het spel te starten.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

CSS wordt gebruikt om vorm te geven aan de verschillende titels, buttons en tekst-vlakken.