

VR website with A-Frame

Leerlingen: Mathis Dewulf, Henrique Cavaleiro

Klas: 6DATA

Datum: 19/06/2025

Wat is A-Frame?

A-Frame is een **open-source webframework waarmee je eenvoudig Virtual Reality (VR) ervaringen kunt maken** die werken in de webbrowser.

Het is gebouwd op HTML en JavaScript, waardoor het toegankelijk is voor zowel beginners als gevorderde ontwikkelaars.

Waarom A-Frame gebruiken?

- **Eenvoudig te leren:** Je kunt 3D-werelden maken met simpele HTML-tags, net zoals je een gewone webpagina bouwt.
- **Werkt in de browser:** Je hebt geen speciale software nodig, alleen een moderne webbrowser.
- **Ondersteunt VR-brillen:** Werkt met VR-brillen zoals Oculus Quest, maar ook gewoon op je computer of smartphone.

Voorbeeld

Hier is een eenvoudig voorbeeld van een A-Frame scène:

```
<html>
  <head>
    <script src="https://aframe.io/releases/1.4.2/aframe.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-box position="0 1 -3" color="tomato"></a-box>
      <a-sphere position="2 1 -5" color="skyblue"></a-sphere>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="8" height="8"
color="#7BC8A4"></a-plane>
    </a-scene>
  </body>
</html>
```

Samenvatting

Met A-Frame kun je **snel en makkelijk VR-werelden maken met HTML**.

Het is ideaal voor beginners die willen experimenteren met 3D en VR op het web.

Stappenplan: Zelf een VR-school applicatie maken met A-Frame

1. Projectstructuur en voorbereiding

- Maak een nieuwe map aan voor je project, bijvoorbeeld `VRSchool-Aframe`.
- Voeg de volgende submappen toe:
 - `assets/` (voor afbeeldingen van de school)
 - `fonts/` (optioneel, voor lettertypes)
- Zet je 360°-foto's en andere afbeeldingen in de map `assets/`.

2. HTML-pagina opzetten

- Maak een `index.html` aan met de basisstructuur voor A-Frame:

```
<!DOCTYPE html>
<html lang="nl">
  <head>
    <meta charset="utf-8">
    <title>360° School VR A-Frame Demo</title>
    <script src="https://aframe.io/releases/1.5.0/aframe.min.js"></script>
    <script src="https://unpkg.com/aframe-gaze-component/dist/aframe-gaze-
component.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/aframe-keyboard@1.3.1/dist/aframe-
keyboard.min.js"></script>
    <script src="https://unpkg.com/aframe-look-at-component@0.8.0/dist/aframe-
look-at-component.min.js"></script>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <a-scene vr-mode-ui="enabled: true" embedded>
      <!-- Camera, sky, knoppen, toetsenbord, textfield -->
    </a-scene>
  </body>
</html>
```

3. 360°-beeld en camera toevoegen

- Voeg een `<a-sky>` toe met een startafbeelding:

```
<a-sky id="sky" src="assets/Overdekte.jpg" rotation="0 -90 0"></a-sky>
```

- Voeg een camera toe met een gaze-cursor:

```
<a-entity id="mainCamera" camera look-controls position="0 1.6 0">
  <a-entity
    cursor="fuse: true; fuseTimeout: 2000"
    position="0 0 -1"
    geometry="primitive: ring; radiusInner: 0.015; radiusOuter: 0.025"
    material="color: cyan; shader: flat"
```

```

    gaze
    raycaster="objects: .gaze-interactive"
  ></a-entity>
</a-entity>

```

4. Navigatie tussen ruimtes

- Maak een bestand `rooms.js` met een `window.roomMap` waarin alle kamers, afbeeldingen en uitgangen staan.
- Voeg in `index.html` een lege container toe voor knoppen:

```
<a-entity id="roomButtons"></a-entity>
```

- Maak een bestand `next-room.js` dat de knoppen dynamisch aanmaakt op basis van de kamer en de gebruiker laat navigeren.
- Voeg een extra knop toe voor "Stel een vraag" (zie voorbeeld in `next-room.js`).

5. Vragen en antwoorden toevoegen

- Maak een bestand `questions.js` met een `window.questions` object waarin per kamer een vraag, antwoorden (A/B/C) en het juiste antwoord staan.
- In `next-room.js` wordt bij het klikken op de vraagknop de vraag getoond en het toetsenbord geactiveerd.

6. Virtueel toetsenbord en tekstveld

- Voeg in `index.html` een onzichtbaar toetsenbord en tekstveld toe:

```

<a-entity id="keyboard" visible="false" position="0 0.6 -1.2" keyboard="inputEl:
#textfield" look-at="#mainCamera"></a-entity>
<a-entity id="textfield" geometry="primitive: plane; height: 0.2; width: 1"
material="color: #fff; opacity: 0.8" position="0 1 -1.5" visible="false"
text="value: ; color: #111; align: left; width: 0.95"></a-entity>

```

- Maak een bestand `keyboard-input.js` met een custom keyboard-component die knoppen voor A, B, C toont en het antwoord in het tekstveld zet.

7. Antwoord opslaan en feedback geven

- Voeg in `next-room.js` een "Beantwoord"-knop toe die het antwoord opslaat in `localStorage` en controleert of het juist is.
- Laat het tekstveld groen (juist) of rood (fout) oplichten na het beantwoorden.

8. Styling

- Voeg een eenvoudige `style.css` toe voor de achtergrondkleur:

```
html, body { height: 100%; margin: 0; }  
body { background: #111; }
```

9. Alles samenvoegen en testen

- Het project wordt bijgehouden in Github
 - Efficiëntere teamwork
 - Versiebeheer
- Zorg dat je scripts in de juiste volgorde worden geladen in `index.html`:

```
<script src="rooms.js"></script>  
<script src="questions.js"></script>  
<script src="gaze-listener.js"></script>  
<script src="keyboard-input.js"></script>  
<script src="next-room.js"></script>
```

- Test de applicatie in de browser en op een VR-bril.

Tip: Bekijk de bestanden `index.html`, `next-room.js`, `keyboard-input.js` en `questions.js` voor concrete voorbeelden van de implementatie.