ReactVR

Als tweede framework hebben we ReactVR genomen. Hierin hebben we een kleine beleving gemaakt en hebben we met de volgende elementen geëxperimenteerd: Animaties, panos, views, interacties, componenten en stylesheet.

Animaties:

Helaas zijn de animatie mogelijkheden in ReactVR niet heel uitgebreid. Op dit moment is het alleen mogelijk om View, Text en Image te animeren met Animated.View, Animated.Text, Animated.Image. Dit betekend dat je dus eigenlijk alleen containers, tekst en plaatjes kan animeren. Volgens de documentatie kan je zelf een animation maken met createAnimatedComponent, maar daar is verder geen tot weinig onduidelijke documentatie over. Animaties zijn op zichzelf niet heel moeilijk om te begrijpen als je eenmaal de syntax van ReactVR begrijpt maar als je dat nog niet snapt dan zijn het wel lastige obstakels om te overkomen.

Pano en views:

Pano image is een panorama foto die in een bol van 1000 meter om de gebruiker heen wordt gezet. Er zijn twee soorten: equirectangular panos en cubic panos. Equirectangular panos bestaat uit een foto met een aspect ratio van 2:1. De foto’s hiervoor zijn ook gemaakt met een speciale 360 camera. Deze foto’s worden dan in een bol vorm om de gebruiker heen gezet. Cubic panos bestaan uit 6 foto’s die in een kubus vorm om je heen gezet worden. Pano image is een van de makkelijker elementen van ReactVR. In ReactVR mag je maar 1 component in je render hebben maar vaak wil je meer dan 1 component om dit op te lossen heb je views. Views zijn eigenlijk gewoon containers waarin je dingen samen groepeert. Views zijn heel eenvoudig en de syntax is eigenlijk niet echt anders dan een simpele div tag van html.

Interacties

Bij ReactVR heb kan je in een component een onClick gebruiken en dan daarna een functie aan verbinden. In de kleine applicatie die ik had gemaakt werd een functie aangeroepen die plus 10 of min 10 bij een applicatie deed. De syntax is moeilijk om te bedenken en te schrijven als je geen ervaring er mee hebt. Maar als je het 1 keer gezien heb kan je het wel begrijpen om dat het best logisch gestructureerd is.

Components

Components zijn herbruikbare gebruikersinterface elementen die gebruikt kunnen worden als tags. Er zijn een aantal standaard ingebouwde componenten zoals Text en Image. Maar het is ook mogelijk om zelf een component te maken met React.Component. Elk gebruikers component heeft een render die child-components terug geeft. De standaard componenten zijn niet heel erg ingewikkeld omdat er duidelijke documentatie over is en het logisch gestructureerd is.

Stylesheet

Stylesheets wordt gebruikt om bijvoorbeeld rotatie positie en schaal mee te geven. Je kan ook al van te voren een stylesheet aanmaken en daar een naam aan geven. Als je dan die naam aan elementen geeft, dan hebben die elementen allemaal de stijl die bij de naam hoort. Het doet wel een beetje denken aan de manier hoe CSS werkt. Doordat het lijkt op de CSS structuur is het redelijk makkelijk te begrijpen en te maken.

Na deze specifieke punten binnen ReactVR zijn er nog een aantal algemene punten die opvallen. Zo is de documentatie niet heel erg uitgebreid en is het moeilijk om op het internet oplossingen voor je bugs te vinden omdat het een erg nieuwe techniek is. Daarnaast gebruikt ReactVR JSX syntax en dat maakt erg moeilijk om te begrijpen als je geen ervaring al hebt met JSX. Verder omdat het een 3d omgeving is het moeilijk om te begrijpen wat er ruimtelijk aan de hand is met je objecten voor dat je het ervaren hebt in de VR omgeving. Als laatste hoef je bij ReactVR niet nog aparte hosting te vinden voor het ontwikkelen van de omgeving omdat het deel is van de setup van ReactVR.