VNTANA Viewer Demo

This package comes with two scripts, each demonstrating different aspects of the viewer:

1. `simple` - adds the viewer to a web page, and sets it up to display a 3d model with the environment map through the attribute interface,
2. `integration` - fetches data from VNTANA Platform and applies it to the viewer through the property interface.

The accompanying `npm` package doesn't need any prior installation. In order to run the scripts it suffices to execute `npm run simple` or `npm run integration` from the package's root directory. Both scripts run the `http-server` and open the corresponding page in the browser.

Both examples utilize the content of the `shared` directory:

- `viewer.min.js` - ES module containing the VNTANA Viewer code,
- `viewer.umd.min.js` - UMD module containing the VNTANA Viewer code,
- `style.css` - styles used for this demo,
- `viewer.css` - default styles for the viewer and positioning of buttons.

Simple Example

Directory `simple` contains the `index.html` containing the page's HTML code, the `chair.glb` model, and `Neutral.hdr` environment map. The body of the document is:

```
1  <body>
2    <script type="module" src="../shared/viewer.min.js"></script>
3
4    <vntana-viewer src="chair.glb" environment-src="Neutral.hdr" tone-mapping="neutral">
5      <vntana-fs-button></vntana-fs-button>
6    </vntana-viewer>
7  </body>
```

Line 2 loads the ES module containing the viewer. Line 4 adds the viewer elements, sets the model through the `src` attribute and environment through `environment-src`. We also added the `tone-mapping` attribute to improve the lighting experience. `<vntana-fs-button>` is added as a child of the viewer, and toggles the viewer's fullscreen state when clicked.

Integration Example

Directory `integration` contains two files: `index.html` with HTML code for the page, and `platform.js` containing the function `getPlatformData`, which we will use to fetch the product data from VNTANA Platform.

The first part of the page's body loads the viewer with different buttons:

```
1  <vntana-viewer>
2    <vntana-fs-button></vntana-fs-button>
3    <vntana-qr-button class="expandable"></vntana-qr-button>
4    <vntana-ar-button></vntana-ar-button>
5    <vntana-center-button></vntana-center-button>
6    <div class="button-container zoom-buttons">
7      <vntana-zoom-in-button></vntana-zoom-in-button>
8      <vntana-zoom-out-button></vntana-zoom-out-button>
9    </div>
10 </vntana-viewer>
```

Classes `button-container` and `zoom-buttons` come as part of viewer's default styling. Elements `<vntana-qr-button>` and `<vntana-ar-button>` are mutually exclusive, so at most one of them will be visible at any time. Unless we provide the URL that will be encode in the QR, `<vntana-qr-button>` won't be visible.

Second part of the body handles the main purpose of this demo - loading the data from VNTANA Platform. We start by importing the `getPlatformData` function from file `platform.js` in the `integration` directory, and the `normalize` function to convert data stored in VNTANA Platform to viewer properties.

```
1  <script type="module">
2    import {getPlatformData} from './platform.js';
3    import {normalize} from '../viewer.min.js';
4
5    const platformData = await getPlatformData(
6      "asset-library",
7      "furniture",
8      "85a51c7b-07c1-4143-bd56-aa2a43acaa42"
9    );
10    platformData.config = normalize(platformData.config);
11
12   const config = {
13     src: platformData.src,
14     usdzSrc: platformData.usdzSrc,
15     poster: platformData.poster,
16     ...platformData.config,
17   };
18
19   const viewer = document.querySelector("vntana-viewer");
20
21   for (const [key, value] of Object.entries(config)) {
22     viewer[key] = value;
23   }
24
25   const qrButton = viewer.querySelector("vntana-qr-button");
26   qrButton.url = platformData.qrUrl;
27 </script>
28
```

The `getPlatformData` function accepts three parameters: `organizationSlug`, `clientSlug`, and `productUuid`. All three parameters can be easily obtained from VNTANA Platform links, since all platform links are of the form:

`https://platform.vntana.com/<organizationSlug>/<clientSlug>/products/edit/<productUuid>`

The function returns an object with the following properties:

- `src` - URL of the GLB model,
- `usdzSrc` - URL of the USDZ model,
- `poster` - URL of the poster/thumbnail,
- `qrUrl` - URL of the product's embed link with `autoAR` enabled,
- `config` - config data for the viewer without links.

The `qrUrl` should probably be replaced with a different URL for custom integrations. After obtaining the platform data in lines 5-9, we call the `normalize` function on the `config` obtained from `getPlatformData` in link 10. In lines 12-17 we merge all the data into one config containing a list of `(key,value)` pairs that will be passed to the viewer. In line 19 we obtain a reference to the viewer, and pass it these pairs in lines 21-23. We obtained the reference to the `<vntana-qr-button>` in line 25, and pass it the `qrUrl` string.