

LODE Three

This helper module simplifies loading lode-generated LOD artifacts in a Three.js application.

Usage

Import the module and the lode-manifest and create the `lodeContext` by calling the `createContext` method.

```
import * as lodLoader from '@kreativwebdesign/lode-three'
import manifest from "../lode-build/lode-manifest.json";
```

```
const lodeContext = lodLoader.createContext({
  basePath: "../lode-build",
  manifest,
});
```

and then load the artifacts:

```
lodLoader.loadModel({
  lodeContext,
  artifactName: 'path-to/artifact',
})
```

API

`createContext`

This method accepts the `basePath` to the lode-build output and the lode-manifest.

```
{
  basePath: "../lode-build", // Relative path to the lode-build output
  manifest, // the manifest generated by the lode-build, by default at lode-build/lode-manifest.json
}
```

`loadModel`

This method accepts the `basePath` of the generated artifacts where all the artifacts are and the specific artifact name.

```
{
  lodeContext, // Object returned from the createContext method
  artifactName: 'path-to/artifact', // path to the artifact base
}
```

This would work for the following folder structure:

```
| -index.js // where the code is
| -lode-build
|   | -lode-manifest.json
|   | -path-to
|     | -artifact
|       | -lod-0
|       |   | -artifact.gltf
|       |   | -artifact.bin
|       | -lod-1
|       |   | -artifact.gltf
|       |   | -artifact.bin
```

Outlook

- Add culled lod at the end of each lod if the last one is not set to infinity.
- Add helper for loading manifest async
- Add helper to progressively load LOD-artifacts

Dev

Run `yarn setup` to get up and running

Publish

- Prepare versions and ensure clean setup state
- `npm publish --access public`