

Visualizzatore 3D Additive (Vue.js)

- [Visualizzatore 3D Additive \(Vue.js\)](#)
 - [0](#) Prerequisiti
 - [1](#) Creare un'applicazione Vue.js
 - [2](#) Includere la libreria nel progetto
 - [3](#) Setup del componente vue
 - [4](#) Setup script per richiamare il codice di render
 - [5](#) Risultato finale
 - [6](#) Ulteriori funzioni

0 Prerequisiti

- Node.js installato
- file conf x accesso ai nostri repo nexus in cartella utente (.npmrc contenuto in IcarusView)

1 Creare un'applicazione Vue.js

Il primo passo è creare l'applicazione. Aprire il terminale e navigare fino alla directory nella quale si vuole operare e lanciare il comando:

```
> npm create vue@latest
```

Questo comando installerà ed eseguirà `create-vue`, lo strumento ufficiale per eseguire lo scaffolding del progetto Vue.

```
√ Project name: ... Web3Dviewer
√ Package name: ... web3dviewer
√ Add TypeScript? ... No / Yes
√ Add JSX Support? ... No / Yes
√ Add Vue Router for Single Page Application development? ... No / Yes
√ Add Pinia for state management? ... No / Yes
√ Add Vitest for Unit Testing? ... No / Yes
√ Add an End-to-End Testing Solution? » No
√ Add ESLint for code quality? ... No / Yes

Scaffolding project in .\Web3Dviewer...
```

N.B.: il nome `Web3Dviewer` è solo per esempio, si può inserire qualsiasi nome progetto si voglia.

Una volta creato il progetto, segui le istruzioni per installare le dipendenze e avviare il server di sviluppo:

```
> cd Web3Dviewer
> npm install
> npm run dev
```

In questo modo ora abbiamo un'applicazione web funzionante.

2 Includere la libreria nel progetto

Per includere la libreria nel progetto è necessario copiare la cartella fornita [WebGL](#) ed incollarla all'interno della cartella `Web3Dviewer/src/`.

 Alt text

3 Setup del componente vue

Usando il componente di esempio `App.vue` eliminare tutto il contenuto e sostituirlo con il seguente codice:

```
<template>
  <main>
    <div id = "WebGL-3mfExport">
    </div>
  </main>
</template>
```

Che permetterà di contenere il visualizzatore.

4 Setup script per richiamare il codice di render

Nella pagina `index.html` è necessario aggiungere il seguente codice per poter richiamare le funzioni della libreria:

```
<script type="module">
  import { webgl_original } from './src/WebGL/draw/webgl_drawVue.js';

  const WGL = new webgl_original();

  let options = {
    _dimX : 960,
    _dimY: 540,
    _fileName: "Cubo.3mf",
    _showCalcGrid: false
  }


  WGL.initcall(options);

</script>
```

```
<script src="./src/WebGl/draw/webgl_drawVue.js" type="module"></script>
```

La variabile `_showCalcGrid` definisce il tipo di griglia da mostrare (Infinita o Calcolata).

5 Risultato finale

 Gif che mostra il risultato finale

6 Ulteriori funzioni

La libreria presenta ulteriori funzioni quali:

- `.refreshCall(nStart, nEnd)`: permette di decidere quali layer dell'oggetto mostrare.
- `.resetCamera()`: gestisce il reset della camera alla posizione iniziale.
- `.setCamerType()`: permette di cambiare il tipo di camera tra:
 - Prospettica
 - ortografica