

Construção de uma biblioteca para mapas interativos com Three.js

Paulo E. R. Werle¹

¹ Ciência da computação – Universidade Federal Fronteira Sul (UFFS)
Chapecó, SC – Brazil

pauloeduardo840@gmail.com

Abstract. *This article describes the conception of an interactive library for map development, using Three.js technology. The aim is to provide developers with a flexible and robust tool to create detailed and immersive maps in their projects. The proposal aims to foster innovation and creativity, providing an engaging experience for end users. The library seeks to simplify the implementation of interactive maps, reducing the learning curve and development time, drawing inspiration from resources such as the League of Legends map.*

Resumo. *Este artigo descreve a concepção de uma biblioteca interativa para o desenvolvimento de mapas, utilizando a tecnologia Three.js. O objetivo é oferecer aos desenvolvedores uma ferramenta flexível para criar mapas imersivos em seus projetos. A proposta visa fomentar a inovação e a criatividade, proporcionando uma experiência envolvente aos usuários finais. A biblioteca busca simplificar a implementação de mapas interativos, reduzindo a curva de aprendizado e o tempo de desenvolvimento, inspirando-se em recursos como o mapa de League of Legends.*

1. Introdução

Neste artigo, apresentamos a idealização de um projeto para o desenvolvimento de uma biblioteca com o intuito de auxiliar no processo de construção de mapas interativos utilizando a biblioteca *Three.js* [Three.js 2024]. O objetivo é fornecer aos desenvolvedores uma ferramenta flexível e robusta que simplifique a incorporação de mapas detalhados e imersivos em seus projetos.

Nesse sentido, a biblioteca *ThreeJsMap* começou a ser desenvolvida para resolver os desafios técnicos envolvidos na criação de cenas 3D interativas no âmbito de criação de mapas. A ferramenta visa oferecer uma solução acessível e eficaz que permite aos desenvolvedores incorporar facilmente seus mapas personalizados em suas criações. A biblioteca abstrai muitos dos detalhes técnicos, permitindo que os desenvolvedores se concentrem mais na criatividade e inovação de seus projetos.

Ao notar a dificuldade de implementação de mapas, como no mapa de League of Legends [League of Legends 2024], surge-se a ideia de iniciar a projetar uma ferramenta para auxiliar os desenvolvedores a criar projetos com a mesma finalidade. A biblioteca *ThreeJsMap* foi desenvolvida para simplificar a criação de mapas 3D, proporcionando uma solução robusta e flexível. Com funcionalidades que incluem carregamento de modelos 3D, adição de iluminação, detecção de interações com objetos e renderização contínua, a biblioteca *ThreeJsMap* facilita significativamente o desenvolvimento de mapas em aplicações, incentivando a inovação e a criatividade.

2. Objetivos

Diante disto, tem-se como objetivo facilitar a implementação de mapas, reduzindo a curva de aprendizado e o tempo de desenvolvimento, oferecendo uma biblioteca intuitiva que permita aos desenvolvedores criar seus próprios mapas personalizados e envolventes de forma simples. Também pretende-se oferecer funcionalidades como renderização de modelos 3D previamente criados, controle de câmera para circulação no ambiente, animações e personalização na manipulação 3D, e demais ferramentas para atender às diversas necessidades dos desenvolvedores.

Com isso, a biblioteca poderá proporcionar uma experiência imersiva, tendo funcionalidades como renderizar modelos de mapas em 3D e 2D. Pretende-se também proporcionar uma experiência de usuário fluida e intuitiva, com possibilidade de adicionar animações e efeitos visuais para tornar os mapas mais atrativos e envolventes. Garantindo uma experiência visualmente estimulante, com elementos que adicionem vida e dinamismo aos mapas.

3. Funcionalidades implementadas:

A biblioteca *ThreeJsMap* foi desenvolvida para auxiliar desenvolvedores na criação de mapas interativos utilizando a *Three.js*. Abaixo estão descritas as funcionalidades implementadas até o momento, juntamente com uma explicação sobre como cada uma delas funciona.

Inicialização do canvas e renderização: A função implementado na biblioteca chamada *Canva* é responsável por localizar o elemento HTML onde o mapa será renderizado. Em seguida, a função *Render* cria o renderizador *Three.js*, ajustando o tamanho do elemento html canvas para garantir uma visualização adequada do mapa, sendo que propriedades como tamanho podem ser passadas como parâmetros;

Criação de cena e configuração de fundo: A função *Scene* cria uma nova cena *Three.js* e define a cor de fundo da mesma. Isso proporciona um ambiente tridimensional onde os objetos e mapas serão renderizados;

Configuração de Câmera: A função *Camera* inicializa a câmera que será utilizada para visualizar a cena. Ela configura parâmetros como posição, campo de visão (FOV), plano próximo e plano distante, garantindo uma perspectiva adequada para a visualização do mapa;

Controle de Navegação: A função *Control* utiliza a classe *OrbitControls* do *Three.js* para permitir a navegação interativa pelo mapa. Os controles configuram a rotação, zoom e pan da câmara, proporcionando uma navegação fluida e intuitiva. O *OrbitControls* foi configurado de modo que consiga proporcionar uma boa experiência, contendo propriedades como de limitação de área e de zoom;

Iluminação: A função *Light* adiciona uma luz direcional à cena onde simula o sol na cena, onde é possível configurar atributos como cor, intensidade e posição da luz. A iluminação adequada é crucial para a visualização clara dos modelos e mapas 3D;

Carregamento de Modelos GLTF: As funções *MapLoader* e *ObjectLoader* utilizam o *GLTFLoader* do *Three.js* para carregar modelos 3D no formato GLTF. Essas funções permitem configurar a posição, escala e callbacks para eventos de carregamento, progresso e erro, facilitando a inclusão de modelos personalizados no mapa;

Adição de Helpers para Desenvolvimento: As funções *GridHelper* e *AxesHelper* adicionam grades e eixos à cena, respectivamente. Esses helpers são úteis para desenvolvimento e depuração, oferecendo uma referência visual para a orientação e escala dos objetos no mapa.

Interatividade com Raycaster: As funções *RaycasterClick*, *RaycasterHover* e *RaycasterTouch* utilizam o Raycaster do Three.js para detectar interações do usuário com os objetos no mapa. Essas funções permitem configurar callbacks para eventos de clique, toque e hover, possibilitando a criação de interações dinâmicas.

4. Proximas funcionalidades:

A biblioteca *ThreeJsMap* já oferece diversas funcionalidades úteis para a criação de mapas interativos com Three.js. No entanto, ainda há funcionalidades importantes que pretende-se implementar para aprimorar ainda mais a experiência dos desenvolvedores e usuários finais:

Renderização de Mapas em 2D: Atualmente, a biblioteca suporta a renderização de modelos 3D. Mas, planeja-se adicionar suporte para a renderização de mapas em 2D, oferecendo uma visualização de cima sem distorção do modelo fornecido. Isso proporcionará flexibilidade na escolha da representação do mapa, permitindo aos desenvolvedores optar pela melhor abordagem conforme suas necessidades específicas.

Efeitos Visuais e Animações: Embora a biblioteca já ofereça funcionalidades básicas de interatividade, pretende-se expandir o suporte para efeitos visuais e animações dinâmicas. Isso incluirá elementos como mudanças de iluminação, transições entre áreas do mapa e animações específicas (Exemplo: nuvens em movimento, pássaros voando, aviões passando, entre outros).

Essas melhorias visam tornar a ferramenta ainda mais poderosa e versátil para a criação de mapas interativos, proporcionando aos desenvolvedores uma base robusta para a inovação e a criação de experiências imersivas e envolventes.

5. Conclusão

O objetivo deste projeto é fornecer uma solução acessível para o desenvolvimento de mapas interativos, capacitando os desenvolvedores a criar experiências imersivas para seus usuários. Ao simplificar o processo de criação de mapas detalhados e personalizados, espera-se estimular a inovação e a criatividade no desenvolvimento de jogos, aplicativos e outras aplicações interativas.

Acredita-se que o resultado final deste projeto será uma biblioteca completa e funcional, pronta para ser utilizada por desenvolvedores em uma variedade de projetos. Essa biblioteca pode permitir a criação de mapas interativos de forma simples e eficiente, proporcionando uma experiência imersiva para os usuários finais.

Assim, anseia-se para poder demonstrar o potencial total deste projeto por meio de exemplos de código e casos de uso que destacam suas diferentes funcionalidades. Espera-se que esses exemplos mostrem a versatilidade e a adaptabilidade que a idealização de novas funcionalidades juntamente com as que se tem presente possam ter em diferentes contextos de aplicação, inspirando os desenvolvedores a explorar novas possibilidades e criar experiências cativantes para seus usuários.

6. Projeto

Acessos a biblioteca:

Github: <https://github.com/paulo-werle/three-js-map>

Npm: <https://www.npmjs.com/package/three-js-map>

Referências

League of Legends (2024). Mapa de league of legends. https://map.leagueoflegends.com/pt_BR. Acesso em 07 de maio de 2024.

Three.js (2024). Three.js - javascript 3d library. <https://threejs.org/>. Acesso em 07 de maio de 2024.