

Assignment Nuxt.JS / Flask

Objectif

Développer une application web permettant d'afficher des objets 3D dans un navigateur grâce à Three.js. Un GUI développé avec Nuxt.JS et la librairie UI NuxtUI permet d'ajouter dans la scène trois types d'objets : une sphère, un cube et une pyramide à base triangulaire. Chaque modification est envoyée en temps réel à un serveur Flask via une WebSocket afin d'être persisté. Lorsque le serveur redémarre, il restaure la dernière configuration afin que tous les clients qui se connectent voient la même scène.

Tâches requises

1. Front-End:
 - Utiliser le seul framework **Nuxt.JS** pour le développement de l'application. Se baser sur les normes, standards et sur la documentation de Nuxt.
 - Pour la partie interface / design, se baser sur **NuxtUI Pro**. Aucune license n'est requise pour un développement local.
 - Utiliser **Three.js** pour le rendu 3D.
 - Implémenter un formulaire permettant d'ajouter à la scène un objet (un cube, une pyramide à base triangulaire ou une sphère), de pouvoir choisir sa taille et sa couleur et sa position dans la scène.
 - Établir une connexion WebSocket socketIO au backend pour recevoir et envoyer la configuration de la scène en temps réel.
 2. Backend
 - Implémenter un serveur **Flask**.
 - Utiliser **Flask-SocketIO** pour la gestion de la WebSocket socketIO.
 - Persister la configuration de la scène pour appliquer la même après un redémarrage du serveur.
 3. Communication WebSocket socketIO
 - Quand un client se connecte, lui envoyer la configuration de la scène.
 - Quand un utilisateur modifie la scène, envoyer et persister la configuration sur le serveur.
 - Quand le serveur reçoit une modification de la scène, s'assurer que cette modification soit reportée sur tous les clients connectés.
 4. Persistance
 - Persister la configuration dans un **fichier JSON**.
 5. Documentation
 - Ajouter un fichier **README** qui décrit l'application et comment l'utiliser, quelles dépendances ont été ajoutées.
-

Bonus (optionnel)

- Contrôles supplémentaires : texture, effets d'éclairage (spotlight, ambient), etc.
- Pouvoir déplacer les objets dans la scène avec la souris.
- Gestion d'un catalogue de configurations (save / load).

Critères d'évaluation

- Les fonctionnalités requises sont toutes implémentées et fonctionnent sans lag. La configuration à l'écran est la même pour tous les clients connectés.
- Le code source du frontend respecte les normes et standards les plus récents de Nuxt.JS & Vue.js.
- TailwindCSS est autorisé, tant que seuls les composants de NuxtUI sont utilisés pour la construction de l'interface.
- La configuration de la scène est persisté et transmise à tous les clients connectés.

Calendrier

- Le code doit être déposé sur un **repo Github public** et doit contenir le frontend et le backend. Le lien du repo devra être envoyé à **evan.schleret@windshape.com**.
- Cet assignment doit être complété d'ici au **jeudi 14 août 23h59**. Tout commit fait au-delà de ce moment ne sera pas comptabilisé.
- Documente ton raisonnement, ta recherche de solution, les choix que tu as fait pendant le développement de l'application, les problèmes que tu as rencontrés et comment tu les as surpassés, pour être prêt à nous présenter tout cela à une date ultérieure, à ta convenance.

Bonne réalisation et bon courage !