

Mis à jour le 24/08/2026

S'inscrire

Formation WebGPU

2 jours (14 heures)

Présentation

WebGPU est une API moderne qui permet d'exploiter la puissance des cartes graphiques depuis le navigateur pour concevoir des rendus 3D et des calculs parallèles performants. Avec WGSL, les pipelines de rendu et les compute shaders, vous construisez des expériences web graphiques plus fluides et plus ambitieuses.

Notre formation WebGPU vous permet d'acquérir les méthodes essentielles pour développer des applications graphiques interactives reposant sur une architecture GPU moderne. Vous apprendrez à initialiser un contexte WebGPU, à sélectionner un adaptateur, à créer un périphérique logique et à organiser les ressources nécessaires au rendu.

Vous maîtriserez la création de buffers, de textures, de bind groups et de pipelines, ainsi que les bases du langage WGSL. La formation aborde la programmation des shaders de sommets et de fragments afin de produire des scènes 2D et 3D affichées dans un canvas.

Vous apprendrez également à exploiter les compute shaders pour paralléliser des traitements, à gérer les commandes GPU et à structurer une boucle de rendu efficace. Les exercices mettent l'accent sur les choix d'architecture, la lisibilité du code et la réutilisation des ressources.

À l'issue de la formation, vous serez en mesure de concevoir un prototype WebGPU fiable, d'identifier les limites matérielles, de diagnostiquer les erreurs et d'optimiser les transferts entre le processeur et la carte graphique. Vous disposerez des bases nécessaires pour intégrer des usages graphiques avancés dans vos applications web.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Comprendre l'architecture et le modèle de programmation de WebGPU
- Initialiser un périphérique GPU et configurer un canvas de rendu
- Créer et exploiter des buffers, des textures et des bind groups
- Développer des shaders de rendu avec WGSLL
- Mettre en oeuvre un pipeline de calcul avec des compute shaders
- Diagnostiquer les erreurs et optimiser les ressources d'une application WebGPU

Public visé

- Développeur front end
- Développeur JavaScript
- Développeur 3D
- Ingénieur logiciel
- Architecte front end

Pré-requis

- Maîtrise des fondamentaux de JavaScript moderne
- Connaissances des balises HTML et de la mise en forme CSS
- Pratique des fonctions asynchrones et des promesses en JavaScript
- Notions de géométrie 2D ou 3D, vecteurs et matrices
- Expérience de développement avec un navigateur et les outils de débogage

Pré-requis techniques

- Poste sous Windows 11, macOS ou Linux avec pilotes graphiques à jour
- Carte graphique compatible avec l'accélération matérielle, mémoire vive de 8 Go minimum, 16 Go recommandés
- Navigateur récent compatible WebGPU, tel que Google Chrome ou Microsoft Edge
- Éditeur de code Visual Studio Code ou environnement équivalent
- Connexion Internet stable et accès autorisé aux outils de développement du navigateur

Programme de notre formation WebGPU

[Jour 1 - Matin]

Fondamentaux et architecture WebGPU

- Positionnement de WebGPU dans l'écosystème graphique web et différences avec WebGL
- Comprendre le modèle GPU : adaptateur, périphérique logique, file de commandes et ressources
- Découvrir les objets centraux : GPUAdapter, GPUDevice, GPUQueue et GPUCanvasContext
- Initialiser WebGPU dans une application JavaScript et gérer l'absence de compatibilité
- Configurer le canvas, le format de présentation et le cycle de préparation du rendu

- Atelier pratique : initialiser un projet WebGPU, détecter les capacités disponibles et afficher un canvas configuré

[Jour 1 - Après-midi]

Premier pipeline de rendu et shaders WGSL

- Comprendre les étapes d'un pipeline de rendu : sommets, primitives, rasterisation et fragments
- Prendre en main la syntaxe de WGSL : types, fonctions, attributs, structures et vecteurs
- Créer un module de shader et définir les points d'entrée vertex et fragment
- Créer un GPURenderPipeline et encoder une passe de rendu
- Utiliser des vertex buffers pour transmettre positions et couleurs à la carte graphique
- Atelier pratique : programmer et afficher un triangle coloré avec un vertex shader et un fragment shader WGSL

[Jour 2 - Matin]

Ressources GPU, transformations et textures

- Créer et alimenter les GPUBuffer selon leurs usages : vertex, index, uniform et storage
- Comprendre l'alignement mémoire, les structures de données et les échanges entre JavaScript et WGSL
- Organiser les ressources avec les bind groups et les bind group layouts
- Appliquer des transformations 2D et 3D avec vecteurs, matrices et uniform buffers
- Charger et exploiter des textures, des samplers et les coordonnées UV
- Atelier pratique : rendre une géométrie transformée et texturée à partir de buffers et d'un bind group

[Jour 2 - Après-midi]

Calcul GPU, diagnostic et optimisation

- Identifier les cas d'usage des compute shaders : traitements massivement parallèles, simulation et préparation de données
- Construire un GPUComputePipeline, définir les workgroups et déclencher un dispatch
- Lire les résultats d'un calcul GPU et synchroniser les buffers avec l'application
- Gérer les limites, les fonctionnalités optionnelles, les error scopes et la perte de périphérique
- Optimiser les performances : réduction des transferts CPU GPU, mutualisation des ressources et mesure des charges
- Atelier pratique : réaliser un calcul parallèle sur GPU, récupérer son résultat et intégrer le traitement dans une application de démonstration

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes,

souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.