

Formation WebGL

Description du besoin:

Le besoin du client est de former ses unités de travail à la technologie WebGL et aux frameworks utilisant cette technologie, dans le but de répondre à des besoins clients toujours plus innovants. Il a été établi en réunion de travail que les apprenants devront acquérir les bases de ce qui constitue un monde virtuel en 3D, ceci dans le but de mieux comprendre le fonctionnement des divers frameworks utilisés et le cas échéant, de pouvoir étendre leur fonctionnement si besoin est.

Dans cette optique, il a été recommandé de procéder à deux sessions de formations, la première axée sur l'acquisition des bases de la programmation graphique en 3D avec WebGL et une introduction plus ou moins poussée aux outils mathématiques utilisés. Cette première session sera accompagnée d'un exercice "fil rouge" que le formateur fera évoluer au fil de la formation, permettant ainsi aux apprenants de mettre en pratique leur savoir nouvellement acquis.

Dans l'optique de ne pas perdre de vue l'aspect "rentabilité à court terme" de la formation, des études de cas classiques seront insérées en fin de sessions, utilisant le framework **Three.js**, les apprenants seront amenés à déterminer la façon dont le framework utilise les notions abordées lors de la formation.

Une session de formation se compte par période de 5 jours ouvrés, chaque journée se réfère à un module, le détail des modules et thèmes abordés, ainsi que les exercices sont précisés plus loin dans le document.

Objectifs de la formation:

Le but n'est pas de faire d'un apprenant débutant un expert, mais de fournir toutes les briques de savoir indispensable à l'apprenant, afin que celui-ci puisse appréhender la grande majorité des situations qui se présenteront à lui lors de sa vie professionnelle.

L'apprentissage de la 3D étant un processus global, comprendre ce que l'on fait, même quand on utilise un framework propriétaire qui permet de s'affranchir de certaines connaissances, est essentiel. En effet, il suffit qu'un framework ne soit plus maintenu, ou qu'une contrainte client oblige le développeur à changer d'outil de travail pour que ce savoir, de nature empirique, devienne parfaitement inutile.

Il a donc été convenu avec les responsables de formation, que la formation devait surtout s'envisager sous un angle d'apprentissage global, permettant à l'apprenant de comprendre les fondamentaux de la programmation 3D avec WebGL.

Bien sur, le but n'est pas de rentrer forcément dans le détail de toutes les techniques décrites dans ce programme, mais bien d'en saisir l'essentiel, et d'être capable, à la sortie de la formation, de prolonger soi-même son apprentissage avec une certaine aisance. Les connaissances acquises lors de cette formation étant nécessaires à la conception d'un moteur 3D quelconque, l'apprenant retrouvera inévitablement ces différents concepts quelque soit le framework utilisé, lui faisant gagner un temps d'apprentissage précieux.

I Notions fondamentales

Jour/Module 1

Thèmes abordés:

- Evaluation des apprenants
- Introduction au vocabulaire de la programmation graphique
- Introduction aux cartes GPU: Une carte graphique c'est quoi ? A quoi ça sert ?
- Introduction aux vertices et à la géométrie de base en 3D
- Introduction à la notion de programme WebGL et aux buffers
- Introduction aux shaders

Exercice:

Mise en place de l'exercice "fil rouge", qui correspond à l'affichage d'une suite de primitives (triangles, carrés, sphère etc). Le formateur illustre chaque thème abordé par l'ajout de fonctionnalités correspondantes. Les apprenants sont invités à poser des questions et à participer à la mise en place de l'exercice.

Jour/Module 2

Thèmes abordés:

- Introduction aux vertex Array
- Introduction aux Meshes
- Introduction aux matrices 3x3
- Introduction aux matrices 4x4
- Créer et envoyer des données à un programme webgl

Exercice:

Continuité de l'exercice fil rouge, les apprenants sont amenés à comprendre les tableaux de vertices, la notion de mesh, de matrice et de programme. Ils devront créer leurs propres primitives de manière programmatique puis, charger des meshes décrits dans des fichiers .json qu'ils créeront à la main.

Jour/Module 3

Thèmes abordés:

- Introduction aux IndexBuffers
- Introduction aux textures
- Que sont les coordonnées UV et à quoi servent-elles ?
- Créer ses propres Shaders

Exercice:

Evolution de l'exercice fil rouge, les apprenants sont amenés en premier lieu à utiliser des shaders classiques, puis à habiller leurs meshes avec des textures. La notion de filtre de texture leur sera expliquée et pour finir, ils devront créer leurs propres shaders de manière autonome.

Jour/Module 4

Thèmes abordés:

- La gestion de la profondeur, l'algorithme du peintre et comment l'appliquer
- La gestion de l'alpha blending
- Apprendre à structurer son code pour gérer plusieurs objets

Exercice:

Toujours sur la base de l'exercice fil rouge, les problèmes de profondeur et de gestion des primitives transparentes seront abordés, avec pour chaque problème, une ou plusieurs solutions possibles. Puis l'apprenant sera amené à déplacer la caméra et opérer des transformations (scale, rotation, translation) sur des meshes.

Jour/Module 5

Thèmes abordés:

- Introduction aux lumières
- Rappel des notions introduites lors des 4 premiers modules
- Mise en relation des notions abordées et des fonctionnalités de frameworks tel que Three.js

Exercice:

Les apprenants sont invités à poser toute question en rapport avec les modules étudiés lors des 4 premiers modules. Si besoin, les notions seront approfondies ou réétudiées. Il sera introduit la notion de "lumière" par l'ajout d'une lumière globale au monde virtuel. Puis, ils seront amenés à réfléchir à l'implémentation des techniques abordées par l'étude d'un framework (Three.js), ce qui leur permettra de consolider leurs acquis et de poser les bases d'une logique programmatique WebGL qui leur permettra de traiter de gros volumes de primitives à afficher, ciblant avant tout la performance.

II Aller plus loin

Jour/Module 1

Thèmes abordés:

- Aller plus loin avec les lumières, les lumières ponctuelles
- Comprendre et ajouter une surbrillance spéculaire
- Gestion de plusieurs lumières

Exercice:

Reprise de l'exercice fil rouge de la première session, l'ajout de lumières ponctuelles et de surbrillance spéculaire sera introduit progressivement, en passant en détail les mécanismes utilisés. Le formateur détaillera en quoi ces lumières sont différentes des anciennes etc etc...

Jour/Module 2

Thèmes abordés:

- Comprendre le multi-texturing
- Comprendre les techniques d'optimisation (batch)
- Comprendre les algorithmes de tri (binary trees, quadrees etc...)
- Intégration des préceptes étudiés à code structuré

Exercice:

les apprenants sont invités à ajouter de plus en plus d'objets mobiles et à organiser leur code afin d'être capables de gérer un nombre d'objets virtuellement infini possédant un nombre de texture virtuellement infini. Avec le nombre d'objets à afficher, se feront sentir les besoins en performance, le formateur amènera donc les concepts d'algorithmes de tri, de batch etc..

Jour/Module 3

Thèmes abordés:

- Effectuer un rendu dans une texture et s'en servir comme d'un matériel
- Effectuer un rendu à travers un masque, le stencil buffer
- Création d'une skybox, théorie et pratique

Exercice:

Le but est d'introduire quelques concepts manquants encore à l'arsenal de l'apprenant dans l'optique de lui permettre de créer son propre ersatz de moteur 3D. Le formateur passera en revue quelques concepts déjà étudiés auparavant (texture, batch etc ...) et s'appuiera dessus pour introduire les nouvelles notions.

Jour/Module 4

Thèmes abordés:

- Création d'un moteur 3D basique regroupant l'ensemble des concepts passés en revue.
- Charger un monde en 3D créé à l'aide d'un logiciel populaire (Blender3D) et exporté au format.json

Exercice:

Lors de ce module, l'exercice fil rouge sera pris comme référence. L'apprenant devra recoder un début de moteur 3D en essayant d'incorporer à celui-ci un maximum d'éléments étudiés lors des sessions 1 et 2. Un monde basique créé sous Le logiciel Blender3D sera fourni à l'apprenant au format .json, celui-ci sera amené à charger le fichier fourni, et à convertir les données décrites dans le fichier en éléments / objets à l'aide de directives codées dans le moteur.

Thèmes abordés:

- Comprendre comment fonctionnent les moteurs populaires tels que Three.js, Babylon.js etc.
- Introduction théorique aux bones
- Introduction théorique à la fusion de géométrie
- Introduction théorique à l'animation de mesh
- Introduction théorique au bump mapping
- Introduction théorique au height mapping

Exercice:

Le formateur procédera à une étude du code source de différents moteurs 3D (Three.js, Babylon.js), les apprenants seront amenés à comparer le code de ces moteurs au leur, ainsi ils pourront comparer leurs solutions à celles des autres. Pour finir, il présentera la théorie de certaines techniques encore inconnues des apprenants.