OpenGL ES

Ref : 4-LC-OGES

Prochaines dates

Aucune date pour le moment

Durée : 3 jour(s)

Objectifs

Comprendre le rôle, les possibilité et les contraintes d'OpenGL ES dans le monde de la 3D temps réel embarquée
Comprendre les différences entre OpenGL et OpenGL ES, ainsi que la différence de vision entre OpenGL ES 1.X et 2.X.

Pré-requis

Connaissances de base en développement.
Les démonstrations seront réalisées à l'aide du langage C.

Plan de cours

1. Présentation d'OpenGL
place d'OpenGL sur le marché actuel de la 3D
rôle d'OpenGL et compléments nécessaires
ce qu'OpenGL n'est pas et ce qu'il ne fait pas
notions : rastérisation, vertex, fragment, pixel, texel, ...

2. OpenGL ES
différences et spécificités
OpenGL ES
évolution d'OpenGL ES par rapport à OpenGL
convergence avec OpenGL
gestion de la performances et de la mémoire, optimisations possibles
implémentations d'OpenGL ES
portabilité des applications
correspondances entre les versions d'OpenGL et d'OpenGL ES

3. OpenGL ES 1.x : fixed pipeline
espace de rendu 2D, framebuffer, buffering, ...
machine à états
matrices
espace de rendu 3D : frustum
géométries et modèles : meshes
vertex arrays, vertex buffers
éclairage, ombrages et ombres portées
blending, transparences, brouillard, lissage, ...
textures, multitexturing, mipmaps, compression, ...
tampons Z et stencil
skyboxes, systèmes de particules, ...

4. OpenGL ES 2.X : shaders
présentation, changement d'orientation
comment retrouver les fonctionnalités du pipeline fixe
gérer la compatibilité entre OpenGL ES 1.X et 2.X
impact sur les performances
portabilité des shaders
OpenGL ES Shading Language (GLSL)
vertex shader, fragment shader
multitexturing, stencil/depth test, per-pixel lighting, image space post-processing, ...
présentation d'autres utilisations avancées des shaders
évolutions probables

5. Autour d'OpenGL ES : conception d'applications complètes
intégrer les autres domainesentrées utilisateursons et effetsphysique
gérer les assets au sein du projetmodélisation 3D, textures (contraintes, règles, outils, ...)formats (performance ou standards ?)workflow caractéristique de conception (application et contenu)étapes du développement, maquettage, itérations
porter la logique et la structure de la scènescène graphsbibliothèques et moteurs existantsmoteurs 3Dmoteurs applicatifs dédiés
6. Bindings et intégration
quels langages ?
OpenGL et le web
intégration de contenu / rendu tiers (bitmap, vectoriel, vidéo, ...)
OpenGL en tant que système de fenêtrage