

DEVELOPPEUR Scol Bas niveau



OBJECTIFS :

La formation «Développeur Scol bas niveau » a pour objectif de permettre à l'apprenant de maîtriser le développement de la technologie SCOL à tous les niveaux de programmation (Machine Virtuelle (VM), Plugins SCOL, langage Scol) et de le rendre autonome autour de l'évolution de la technologie (Amélioration VM, Développement de plugins, Développement Serveur...).



Formacode : 46003

Durée : 3 jours minimum

Lieu : Sur site ou dans nos locaux

> PUBLIC CONCERNE

Développeur expérimenté, ingénieur.

> PRE - REQUIS

Formation SCOL haut niveau, Connaissances en développement (C, C++, Java, Python...) et Environnement de développement sous Windows (Visual Studio, API de programmation, multithread,..), Outils de suivi et de gestion de projet informatique (Subversion, trac...) Connaissances avancées en 3D temps réel et Réalité virtuelle.

> PEDAGOGIE

A partir d'un Template de plugin VM et à l'aide de la documentation existante, l'apprenant devra définir, développer et/ou améliorer un plugin permettant l'intégration d'une API dans la technologie SCOL et dans OpenSpace3D. Ce plugin pourra concerner des domaines relatifs aux IHM, à la 3D temps réel ou encore à des fonctions permettant d'ajouter de nouvelles fonctionnalités à OpenSpace3D.

CONTENU DE LA FORMATION

Module 1

La technologie Scol

Introduction : Historique de la Technologie, la communauté SCOL, particularités de la technologie, la programmation SCOL, le SCOL Voyager, les extensions du langage.

Les plugins SCOL : les partitions SCOL, la Machine virtuelle SCOL, généralités, représentation de la pile, fonctionnement de la VM

Introduction au moteur Ogre3D

L'API SCOL, SCOL SDK : Overview, les Objets SCOL, les fonctions pour SCOL, les Callbacks

Module 2

Outils et Licence

Les licences SCOL, versioning (subversion), Trac et Documentation

Module 3

Développement d'un plugin SCOL

L'environnement de développement, chartes de développements Scol, configuration d'un projet

Base d'un nouveau plugin : exemple : template.dll, chargement et Déchargement, Usm.ini : Fichier de configuration

Intégration d'une API, choix d'une API, méthode de développement, couche d'abstraction, wrapping, multithreading

Etude de cas

Présentation d'un plugin déjà développé, utilisation en SCOL, intégration à OpenSpace3D

