

Documentation J4K Java Library

Robin SHIN et Thibaud LEMAIRE

PACT 2015-2016

Installation

Tout d'abord, il est *impératif* d'avoir la version 1.8 du SDK Kinect et non la 2.0 car celle-ci est inexploitable (problèmes de pilotes).

Téléchargement du fichier .jar

Sur : <http://research.dwi.ufl.edu/ufdw/download.php>, télécharger ufdw.jar.

Intégration avec Eclipse

1. Ouvrir Eclipse, aller dans Project > Properties et aller dans l'onglet Java Build Path
2. Cliquer sur le bouton "Add External JARs..." et choisir le chemin vers le fichier ufdw.jar précédemment téléchargé.
3. Déplacer le fichier "ufdw_j4k_32bit.dll" ou "ufdw_j4k_64bit.dll" en fonction de la machine utilisée à la racine du projet Java pour éviter les problèmes de dll manquantes.
4. On peut désormais importer les fichiers de la librairie grâce à la commande "import edu.ufl.digitalworlds.j4k.*".

Recommandations pour l'intégration avec git

1. Tout d'abord, il ne faut surtout pas ignorer le fichier .classpath au risque d'avoir des problèmes avec git.
2. Dans le fichier .classpath, le chemin vers ufdw.jar doit être relatif et non absolu au risque de conflits dès qu'un utilisateur souhaite faire un push ou un pull.

Ajout d'un projet de démonstration sous Eclipse

1. Ouvrir Eclipse, aller dans File > Import... et sélectionner Git > Projects from Git
2. Sélectionner URI puis cliquer sur "Next"
3. Copier <http://research.dwi.ufl.edu/git/j4kdemo> dans l'espace dédié à l'URI et cliquer sur "Next" autant de fois que nécessaire, puis "Finish" : un nouveau projet "j4kdemo" est créé.

Création de l'objet Kinect

Initialisation

initKinectModule();

Initialise la Kinect, les données, le squelette, etc...

Méthodes

public void onSkeletonFrameEvent(boolean[] skeleton_tracked, float[] positions, float[] orientations, byte[] joint_status);

Méthode appelée lorsqu'un nouveau squelette est reçu. Cette méthode remplace le squelette de l'attribut associé de type Skeleton, crée un événement et l'envoie à tous les modules via un système de Listeners.

public void onDepthFrameEvent(short[] arg0, byte[] arg1, float[] arg2, float[] arg3);

Permet de fixer une date de lancement de la Kinect pour, entre autre, calculer le bpm. Cet Event est appelé lorsque le depthFrame est reçu.

public void setListener(KinectListenerInterface l); UV is a stream of 2D frames, which contain the U,V texture coordinate mapping for each pixel in the depth frame.

public static final int XYZ = 0x1000; XYZ is a stream of 2D frames, which contain the X,Y,Z coordinates that correspond to each depth pixel in the depth frame.

public int start(int flags); This method turns on the Kinect sensor and initializes the data streams specified by the input flags. The flags can be specified using the above types of streams according to your needs. For example flag=COLOR | DEPTH | SKELETON; initializes the color, depth, and skeleton streams.

public void stop(); This method turns off the Kinect sensor, and stops all the open streams.

Autres ressources disponibles sur ce site

D'autres méthodes

Disponibles sur : <http://research.dwi.ufl.edu/ufdw/j4k/J4KSDK.php>

Exemples de codes utilisant cette librairie

Disponibles sur : <http://research.dwi.ufl.edu/ufdw/j4k/examples.php>

Comment créer notre propre programme Java utilisant la Kinect ?

Tutoriel disponible sur : <http://research.dwi.ufl.edu/ufdw/j4k/examples.php#how>