Documentation J4K Java Library

Robin Shin et Thibaud Lemaire

PACT 2015-2016

Installation

Téléchargement des fichiers .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. On peut désormais importer les fichiers de la librairie grâce à la commande "import edu.ufl.digitalworlds.j4k.*".

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 crée.

Liste des méthodes de cette bibliothèque

Constructeurs

public J4KSDK(tex) ; The constructor of the J4KSDK class. It establishes connection with the native library, which uses the Microsoft's Kinect SDK. This constructor can automatically initialize the object based on the type of the Kinect sensor that is connected. If various types of sensors are connected priority is given to Kinect 1 devices.

public J4KSDK(byte kinect_type) ; Another constructor of the J4KSDK class. This constructor instantiates the object based the type of the Kinect sensor, which is passed as argument.

public J4KSDK(byte kinect_type, int id) This constructor instantiates the object based the type of the Kinect sensor and id, which are passed as arguments. For example if there are two Kinect 1 sensors, the first one corresponds to id=0 and the second one to id=1

Méthodes

public static final byte MICROSOFT_KINECT_1 = 0x1 ; public static final byte MICROSOFT_KINECT_2 = 0x2 ; These two constants specify the different types of Kinect sensors to be used in the following constructors.

public static final int COLOR = 0x1 ; public static final int INFRARED = 0x2 ; public static final int LONG_EXPOSURE_INFRARED = 0x4 ; public static final int DEPTH = 0x8 ; The following constants represent the different types of data streams.

public static final int PLAYER_INDEX = 0x10 ; public static final int SKELETON = 0x20 ; PLAYER_INDEX is a stream of 2D image frames, which contain the id of the depicted player in each pixel of the depth frame.

public static final int UV = 0x100 ; 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

Notre classe Kinect

Notre classe Kinect hérite de la classe mère J4KSDK.