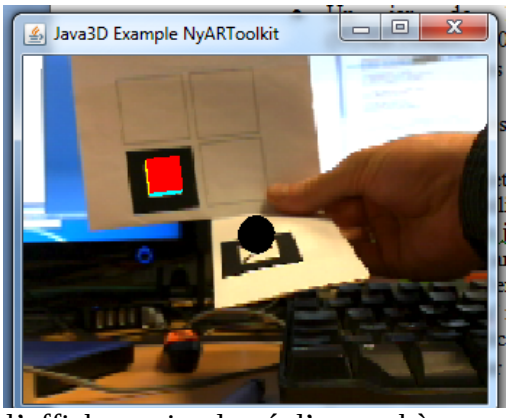
Master 2 Pro - IHM
Module Systèmes Mixtes - Sujet de TP (3h)
UPS – ENAC

Tous les fichiers à télécharger sont accessibles à cette URL :
http://www.irit.fr/~Emmanuel.Dubois/m2ihm.htm

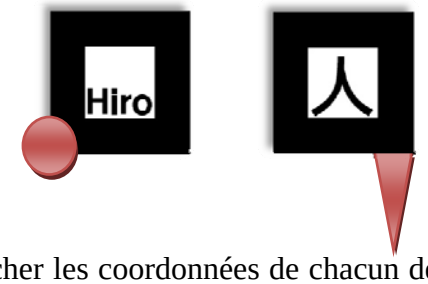
1) Brancher la webcam. Tester la WebCam en allant sur www.testmycam.com. Si nécessaire installez les drivers :

2) Assurez-vous que sont installés sur votre ordinateur :
   - Java JRE **32 bits**
     (c:\Program Files (x86)\Java\jre7 et c:\Program Files (x86)\Java\jdk1.7)
   - Java 3D 1.5.1 **32 bits**
     (c:\Program Files (x86)\Java\Java3D)
   - Java Media Framework
     (c:\JMF2.1.1e ou c:\Program Files (x86)\jmf)
     Si ce n'est pas le cas, veillez à l'installer une fois que la webcam est branchée et en fonctionnement)
3) Téléchargez le fichier Exo1_NyARToolkit300.zip et décompressez le dans le répertoire TP_NyARToolkit. Il contient :
   - Un jar de la librairie NyARToolkit adaptée au M2IHM (NyARToolkit3.0.0_M2PROIHM_2014_JRE7.jar) dans le répertoire lib
   - La description des patterns et des paramètres de la camera, dans le répertoire Data
   - Les fichiers sources de deux exemples, dans le répertoire src.
4) Créer un projet Eclipse
   - Configurer le projet :
     i. Forcer l'utilisation d'un **JRE7 - 32 bits**
     ii. Ajouter les jar de Java3D et de JMF
     iii. Insérer le jar de la NyARToolkit version M2ProIHM-2014.
     iv. Spécifier l'emplacement des librairies natives pour les JAR
         1. jmf : c:\windows\sysWOW64
         2. j3dcore-ogl : …java3D_32bits\bin
   - Exécutez le fichier NyARJava3D.java. Vérifiez qu'un cube s'affiche bien sur le pattern "Hiro".

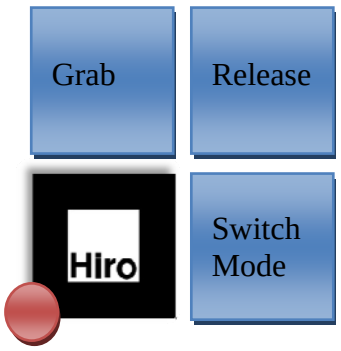


   - Exécutez ensuite le fichier MultipleJava3D_3_0.java.jar.
     Vérifiez qu'un cube s'affiche bien sur le pattern Hiro et qu'une série de 0 s'affiche dans la fenêtre de commande.
5) En partant du second exemple, rendez possible l'affichage simultané d'une sphère sur le pattern Kanji (sorte de Y) et du cube sur le pattern Hiro.

6) Modifiez alors le code de sorte que
   - Une sphère soit centrée sur le coin inférieur gauche du cube du Hiro ;
   - Un cône de 80 de haut et 10 de diamètre s'affiche sur le pattern Kanji (la hauteur du cône dans le plan du pattern, la pointe du cône sortant du pattern).



7) Faites afficher les coordonnées de chacun des patterns détectés et les coordonnées de la pointe du cône, dans une fenêtre WIMP contenant 3 x 3 Textfield en plus de la mise à jour du feedback graphique sur la vidéo.
8) Modifier alors le code afin de mettre en œuvre le pattern MVC et en particulier les échanges d'événements. L'instance chargée de la détection des patterns sera alors également en charge de l'émission d'un événement, capté par la fenêtre d'affichage des coordonnées **(NOTE : si trop complexe, passer à la question suivante.)**
9) Etablir la connexion avec la scène 3D élaborée en cous de Java3D (curseur + scène 3D) : très simple si la question 8 est faite ! Hiro pilotera alors la position du curseur de la scène 3D (la main).
   - L'instruction `try { Thread.sleep(2500); } catch (Exception e) { };` entre l'instruction d'ouverture de la fenêtre contenant la scène 3D et l'ouverture de celle contenant le feedback de la NyARToolkit-Java3D sera peut être requise.
   - Pensez à adapter le système de coordonnées de la scène 3D à celui de la NyARToolkit-Java3D (scale ?)
10) Faire en sorte de pouvoir accrocher un objet 3D pointé par la main en appuyant sur une touche clavier.
11) Ajouter autour du pattern Hiro trois parallélépipèdes rectangles définissant ainsi 3 zones autour de ce pattern : la détection de ce pattern entraîne alors l'affichage dans la zone graphique du cube et de la sphère initiale ainsi que de ces 3 zones.



12) Utiliser la présence de l'extrémité du cône associé au pattern Kanji dans une zone donnée autour du pattern "Hiro" pour :
   - Grabber l'objet pointé par Hiro ;
   - Relâcher l'objet grabbé.
   - ***Facultatif :*** Bascule entre le mode "Curseur absolu" / "Curseur relatif" (comme une souris)