

Screenful NiTETracker API (July 4th, 2014)

For readability, classes and methods that contribute to the library generally use “Hands” and “Bones” in their names when referring to “hand tracking” and “skeletal tracking” functionality to distinguish them from “Hand”/“User”/“Skeleton” used in NiTE naming scheme. Accessors will work as expected, ie. `NiTETracker.getUserTracker()` will return a `NiTEUserTracker`.

screenful.apps

- **TorsoPrinter**: Show tracked torso coordinates until Enter is pressed.
 - `main(String[] args)`
- **BonesAndHandsViewer**: Views both input methods (skeleton / hand) simultaneously.
 - `main(String[] args)`
- **HandsDirectionViewer**: Show general direction of tracked hand movement.
 - `main(String[] args)`

screenful.basic

- **BonesListener (Interface)**: Any class that wants skeleton data should implement this interface and add itself to NuiTracker’s bones listeners.
 - `onNewBonesFrame(UserTrackerFrameRef frame)`
- **HandsListener (Interface)**: Any class that wants hand data should implement this interface and add itself to NuiTracker’s hands listeners.
 - `onNewHandFrame(HandTrackerFrameRef frame)`
- **NiTETracker**: Tracker for NUI features (implements `HandListener` and `SkeletonListener` from the NiTE library).
 - `addHandsListener(HandsListener listener)`
 - `addBonesListener(BonesListener listener)`
 - `getBufferedImage(): BufferedImage`
 - `getHandFrame(): HandTrackerFrameRef`
 - `getUserFrame(): UserTrackerFrameRef`
 - `getHands(): List<HandData>`
 - * Get hand tracking data - see `HandsRenderer` for example of use.
 - `getHandTracker(): HandTracker`
 - `getBones(): List<UserData>`
 - * Get skeleton data - see `BonesRenderer` or `TorsoPrinter` for example of use.
 - `getUserTracker(): UserTracker`
 - `onNewFrame(HandTracker ht)`
 - `onNewFrame(UserTracker ut)`

screenful.detectors

- **ConsecutiveFrames**: A container object for passing two consecutive hand and user tracker frames.
 - `ConsecutiveFrames(HandTrackerFrameRef handsFrame, HandTrackerFrameRef previousHandsFrame, UserTrackerFrameRef bonesFrame, UserTrackerFrameRef previousBonesFrame)`
- **Detector**: Interface for detectors, ie. an object that determines whether the movement of something during two consecutive (hand and/or skeleton) frames was appropriate (true / false).

- **detected(ConsecutiveFrames frames): boolean**
- **getData(): GestureData**
- **DirectionDetector**: Detects hand point movement direction by calculating the displacement vector of each hand point (could also use skeletons) between two consecutive frames. When any tracked hand performs a big enough movement (sensitivity set in millimeters), the detected(..) method returns true.
 - **getSensitivity(): int**
 - **setSensitivity(int sensitivity)**
 - **DirectionDetector(int sensitivity)**
 - * constructor
 - **detected(ConsecutiveFrames frames): boolean**
 - **getData(): GestureData**

screenful.gestures

- **Gesture**: Gesture implements a generic gesture that notifies its listeners when a gesture has been detected for long enough.
 - **Gesture(Detector detector, int framecount, int cooldown)**
 - * constructor, cooldown is in milliseconds
 - **addListener(GestureListener listener)**
 - **onNewHandsFrame(HandTrackerFrameRef frame)**
 - **onNewBonesFrame(UserTrackerFrameRef frame)**
- **GestureData**: Extra data related to a detected gesture.
 - **GestureData(CardinalDirection direction)**
 - * constructor
 - **GestureData(Point3D directionVector, CardinalDirection direction)**
 - * constructor
 - **GestureData()**
 - * constructor
 - **getDirectionVector(): Point3D**
 - **setDirectionVector(Point3D directionVector)**
 - **getDirection(): CardinalDirection**
 - **setDirection(CardinalDirection dir)**
- **GestureListener**: A class that wants notifications when a gesture is detected should implement GestureListener and add itself to the gesture's listeners.
 - **onGesture(GestureData gesture)**
- **JointMetrics**: Static methods for getting skeleton measurements etc.
 - **jointToJointDistance(UserData user, JointType from, JointType to): double**
 - * Returns euclidian distance in space between two skeletal joints, eg. JointType.RIGHT_HAND, JointType.NECK etc. in millimeters
 - **elbowHandXOffset(UserData user, JointMetrics.Side side): double**
 - * Returns the X axis distance in millimeters between elbow and hand.
- **JointMetrics.Side**: Side.LEFT or Side.RIGHT
- **Poses**: Methods to detect skeletal poses based on joint positions, eg. "hands above neck" etc.
 - **dorkyClick(UserData user): boolean**
 - * Returns true if hands are above the neck and distance between the hands was less than 150 mm.

- **handsAboveNeck(UserData user): boolean**
 - * Returns true if a user's both hands are above the neck.
- **Utilities:** Some general utility methods
 - **convertPoint(Point3D nitepoint): javafx.Geometry.Point3D**
 - * convert a NiTE Point3D into javafx Point3D and round the coordinates to integers
 - **determineCardinalDirection(javafx.geometry.Point3D vector, int minSensitivity): CardinalDirection**
 - * return a cardinal direction that most closely matches the vector
 - **displacementVector(Point3D from, Point3D to): javafx.geometry.Point3D**
 - * calculate displacement vector between two points
 - **distance3d(Point3D from, Point3D to): double**

screenful.gui

- **GenericWindow:** Generic frame for displaying graphics
 - **run()**

screenful.gui.rendering

- **HandsRenderer:** Draw tracked hands (red rectangles when tracking) on top of depth image.
 - **onNewHandsFrame(HandTrackerFrameRef frame)**
 - **paint(Graphics g)**
 - * Draw depth image and tracked hands.
- **BonesRenderer:** Draw stick characters from skeleton data on top of depth image.
 - **onNewBonesFrame(UserTrackerFrameRef frame)**
 - **paint(Graphics g)**
 - * Draw depth image and skeletons.
- **DirectionRenderer:** Draw some feedback to directional gestures.
 - **onGesture(GestureData gesture)**
 - **paint(Graphics g)**
 - * Draw text to indicate directions.

screenful.gui.visualization

- **Visualization (Interface):** Interface for classes that provide some sort of graphical presentation of the sensor data. A visualization will take a NuiTracker in its constructor to start listening to events.
 - **show()**
- **DirectionVisualization:** Directional gesture visualization window
- **HandsVisualization:** Hand tracker visualization window
 - **show()**
- **BonesVisualization:** Skeleton tracker visualization window
 - **show()**