Screenful NiTETracker API (August 5th, 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 NiTE UserTracker.

Public class members:

screenful.server - main server program

- **GestureServer**: The main application for browser communication. Implements a WebSocket server for a browser to connect to and passes detected gestures to the UI.
  - **main(String[] args)**
    - Initializes NiTETracker and gesture detection and starts listening for web socket connections.
  - **onClose(WebSocket conn, int code, String reason, boolean remote)**
    - Actions to be done when a socket closes
  - **onError(WebSocket conn, Exception ex)**
    - Actions to be done when an error occurs with the WebSocket
  - **onMessage(WebSocket conn, String message)**
    - Actions to be done when a message is received
  - **onOpen(WebSocket conn, ClientHandShake handshake)**
    - Actions to be done when a connection is opened
- **GestureServer.Messenger**: Inner class of GestureServer sends the command to the browser via WebSockets.
  - **onGesture(Displacement gesture)**
    - When a detected gesture notifies the Messenger, it checks whether the gesture is assigned to exiting interaction or providing input and either stops hand tracking or sends the appropriate messages over WebSockets.
- **Settings**: A settings object for the GestureServer.
  - **Settings()**
    - Create default settings
    - Default filename "default.conf"
  - **Settings(String filename)**
    - Load settings from a configuration file
  - **save()**
    - Saves the current settings

screenful.basic - reading NiTE data

- **BonesListener (Interface)**: Any class that wants skeleton data should implement this interface and add itself to NiTETracker's bones listeners.
  - **onNewBonesFrame(UserTrackerFrameRef frame)**
- **HandsListener (Interface)**: Any class that wants hand data should implement this interface and add itself to NiTETracker's hands listeners.
  - **onNewHandFrame(HandTrackerFrameRef frame)**
- **NiTETracker**: Tracker for NUI features (implements HandListener and SkeletonListener from the NiTE library).
  - **addBonesListener(BonesListener listener)**
    - Add a listener for user tracking

- **addHandsListener(HandsListener listener)**
  - Add a listener for hand tracking
- **forgetHand(short id)**
  - Stop tracking a specific tracked hand ID
- **forgetHands()**
  - Stop tracking all tracked hands
- **getBones()**
  - Return currently detected skeletons
- **getBufferedImage(): BufferedImage**
  - Return the buffered depth image
- **getHandFrame(): HandTrackerFrameRef**
  - Return current hand tracker frame
- **getHandTracker(): HandTracker**
  - Return the hand tracker
- **getHands(): List<HandData>**
  - Get hand tracking data
- **getTrackedHands(): List<HandData>**
  - Get a list of hands that are being tracked (a hand ID can be present but not tracked)
- **getUserFrame(): UserTrackerFrameRef**
  - Return current user tracker frame
- **getUserTracker(): UserTracker**
  - Return the user tracker
- **getBones(): List<UserData>**
  - Get skeleton data
- **onDeviceConnected(DeviceInfo di)**
  - Handles OpenNI's device state changes
- **onDeviceDisconnected(DeviceInfo di)**
  - Handles OpenNI's device state changes
- **onNewFrame(HandTracker ht)**
  - Handles hand tracker frames
- **onNewFrame(UserTracker ut)**
  - Handles user tracker frames

screenful.detectors - movement detection in frames

- **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 detection based on movement

- **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)**
    - Create a new gesture using a specified Detector. Framecount is the amount of consecutive frames where the detector's condition should be true for the gesture to be recognized. Cooldown is in milliseconds and defines a period of inaction after a gesture has been successfully detected.
  - **addListener(GestureListener listener)**
    - Add a listener for the gesture
  - **onNewBonesFrame(UserTrackerFrameRef frame)**
    - Handle user tracker frames
  - **onNewHandsFrame(HandTrackerFrameRef frame)**
    - Handle hand tracker frames
- **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)**
    - Override to implement behavior
- **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
- **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**
    - Return distance between two 3D points

screenful.gestures.detectors

- **ConsecutiveFrames**: A container for passing two consecutive hand and user tracker frames.
  - **ConsecutiveFrames(HandTrackerFrameRef handsFrame, HandTrackerFrameRef previousHandsFrame, UserTrackerFrameRef bonesFrame, UserTrackerFrameRef previousBonesFrame)**
- **Detector**: An interface for defining frame-to-frame detection logic.
  - **detected(ConsecutiveFrames frames): boolean**
    - Override. Should return true when the difference between frames is appropriate. A Gesture will use this boolean value to determine if a frame contributed to the gesture.

- **getData(): Displacement**
  - Return last direction data.

- **DirectionDetector**: Detect movement in the cardinal directions.
  - **detected(ConsecutiveFrames frames): boolean**
    - Returns true when big enough movement is detected towards some direction.
  - **getData(): Displacement**
    - Return last direction data.
  - **getSensitivity(): int**
    - Return the chosen sensitivity.
  - **setSensitivity(int sensitivity)**
    - Set the sensitivity in millimeters.

- **Displacement**: Extra data related to a detected gesture.
  - **Displacement(CardinalDirection direction)**
  - **Displacement(Point3D directionVector, CardinalDirection direction)**
  - **Displacement(Point3D directionVector, CardinalDirection direction, short id)**
    - Constructors
  - **getDirection(): CardinalDirection**
    - Return cardinal direction of displacement (left/right/up/down/in/out)
  - **getDirectionVector(): Point3D**
    - Return 3D vector of displacement
  - **getId(): short**
    - Get the ID
  - **setDirection(CardinalDirection dir)**
    - Set the direction
  - **setDirectionVector(Point3D directionVector)**
    - Set the vector

screenful.gui - GUI windows

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

screenful.gui.rendering - graphics renderers for GUI windows

- **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 - containers for GUI windows

- **Visualization (Interface)**: Interface for classes that provide some sort of graphical presentation of the sensor data. A visualization will take a NiTETracker 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()**

screenful.testapps - applications for testing

- **BonesAndHandsViewer**: Views both input methods (skeleton / hand) simultaneously.
  - **main(String[] args)**
- **HandsDirectionViewer**: Show general direction of tracked hand movement.
  - **main(String[] args)**