LocalSkills-HelloWorldpart6_FaceDetection

In Part 6 of our *Hello World SDK* tutorial we'll learn how to use Misty's face detection abilities to trigger an event. If Misty detects a face she will play a sound, change her LED to white, and end the skill. If she does not detect a face within a reasonable amount of time, the LED will turn off, and the skill will end.

HelloWorldpart6_FaceDetection.json

```
{
    "Name": "HelloWorldpart6_FaceDetection",
    "UniqueId": "63b2cac5-4674-43ce-a048-670303a339ec",
    "Description": "Local 'Hello, World!' tutorial series, part 6.",
    "StartupRules": [ "Manual", "Robot" ],
    "Language": "javascript",
    "BroadcastMode": "verbose",
    "TimeoutInSeconds": 300,
    "CleanupOnCancel": false,
    "WriteToLog": false
}
```

HelloWorldpart6_FaceDetection.js

In order to tell if Misty has detected a face, we'll register an event to receive data from her occipital sensor. Call `misty.RegisterEvent()` and pass in a name for the event (to keep it simple, we'll use `FaceDetection`), the data stream we are subscribing to (`ComputerVision`), and the value `250` (how frequently we want to receive data -- in this case every 250ms).

```
misty.RegisterEvent("FaceDetection", "ComputerVision", 250);
```

Now that we have the event set up, we can send the command to start face detection. This command is different in that it starts the process for Misty to start *looking* for a face, while the event is only set up to trigger if a face is *detected*. Both parts are necessary to handle skills that include face detection.

```
misty.StartFaceDetection();
```

Within the callback (automatically named `_FaceDetection`) we should log a debug message to indicate that a face has been detected, send a command to play an audio clip, and another to change the LED. Then we can send a command to stop face detection. Once the code in this callback finishes, the skill will automatically end after 5 seconds of inactivity.

```
function _FaceDetection() {
   misty.Debug("Face detected!”);

   misty.PlayAudioClip("005-OoAhhh.wav");
   misty.ChangeLED(255, 255, 255); // white
   misty.StopFaceDetection();
};
```

With what we have so far, the skill will run indefinitely if no face is detected. To make the skill more complete, we need to write code to handle this “no face” situation. To accomplish this, let’s register for a timer event to trigger if no face was detected after 15 seconds. We register for this event just after we register for “`FaceDetection`” in our `RegisterEvent` method.

```
misty.RegisterTimerEvent("FaceDetectionTimeout", 15000);
```

Then within the callback (again, automatically named `_<event>`), we log a debug message to indicate the timeout was called, turn the LED off, and send the command to stop face detection. After this command has been issued, Misty will be inactive and the skill will automatically end after 5 seconds.

```
function _FaceDetectionTimeout() {
   misty.Debug("face detection timeout called, it's taking too long...");

   misty.ChangeLED(0, 0, 0); // black
   misty.StopFaceDetection();
};
```

See the complete file below for reference.

```
// Debug message to indicate the skill has started
misty.Debug("starting skill helloworld part6");

// Register for face detection event and timeout
misty.RegisterEvent("FaceDetection", "ComputerVision", 250);
```

```
misty.RegisterTimerEvent("FaceDetectionTimeout", 15000);

// Send command to start face detection
misty.StartFaceDetection();

// Face detection event callback
function _FaceDetection() {
   misty.Debug("face detection callback called..");

   misty.PlayAudioClip("005-OoAhhh.wav");
   misty.ChangeLED(255, 255, 255); // white
   misty.StopFaceDetection();
};



// Timeout callback
function _FaceDetectionTimeout() {
      misty.Debug("face detection timeout called, it's taking too long...");

      misty.ChangeLED(0, 0, 0); // black
      misty.StopFaceDetection();
};
```