Open *Source* | Open *Possibilities*

Image Capture Specification, Update

November 2013

QUALCOMM INNOVATION CENTER, INC.

Current Features

- Three main methods currently
  - takePhoto() - For photographs in the form of blob object
  - getFrame() – for RGBA data in the form of an ImageData object
  - setOptions() – for configuring capture device
- Single error event handler onerror

Finalizing Settings

- Current settings do not include two key camera options
  - Focus (autofocus)
  - Zoom
- Both are important for camera settings, but there is uncertaintly
  - Could these potentially be added as MediaStream constraints?
    - If so, could end up with a settings vs. constraints conflict
    - Would like a group decision one way or the other:
      - Should these adjustments be photo-specific for the foreseeable future?
- What are the options if these settings are not added?
  - Certain capture devices can offer these setting adjustment options directly to the end user
    - Developer cannot control them

Camera Preview

- Currently two known options for preview
  - MediaStream through <video>
    - May not *automatically* reflect current camera settings
    - Post-processing of MediaStream *may* be used to imitate camera settings
      - Imprecise
  - Create a preview stream out of getFrame()
    - Frame-by-frame capture-and-display could have efficiency problems on handheld devices
      - Repaint operations costly
      - Could be alleviated if browser triggers repaint operation only when the ImageData has changed
- Special MediaStreamTrack for camera has been considered by Task Force and rejected
  - Should it be reconsidered?

Camera Preview (cont.)

- How could a special MediaStreamTrack for preview work?
  - MediaStreamTrack.kind = “camera”
  - Constrainable interface should work as defined
    - May need to educate developer community that it is not really necessary to call applyConstraints() on a camera preview stream
  - Camera preview should not part of a valid argument for RTCpc.addStream()
- What are the cons of a MediaStreamTrack.kind = “camera” track?
  - Propagation of different types of MediaStreamTracks
    - Would it hamper implementations by adding complexity?
  - Some implementers may support preview directly on the MediaStreamTrack.kind = “video” itself
    - Display of a time-limited preview overlay when takePhoto() called

Broader Issues

- Use of DOM Promises
  - No real technical arguments exist against making getFrame() and takePhoto() promises-based
  - Several ongoing W3C specs have made the transition, e.g.
    - All SysApps Working Group specs
    - WebCrypto
  - Recommend redefining ImageCapture methods to promise-based