Assignment 10 (HF)

Due: Mon 19.01.2014; 16:00h (1 Week)

Solutions:

You can find exemplary solutions for all assignments on this site:
http://www.medien.ifi.lmu.de/lehre/ws1415/mmn/uebung/loesungen/

Do not share the password that your tutor gave you with others. Thanks to the students who let us publish their submissions.

If you did not attend the tutorial, you can request the password from your tutor via email or come to the next tutorial (highly recommended).

Task 1: Photobooth Webpage

Create a web page that simulates a photo booth. It allows the user to take photos through the webcam and to download them.

Your submission should meet the following requirements:

- The pictures should be square (600 x 600 pixels).
- **Webcam access**:
  Use the `navigator.getUserMedia()` method to access the webcam. Display the captured video to the user. Make sure that your app works on Firefox and/or Chrome. Please indicate in your files which browser you used.
- **Snapshot**:
  - Taking a snapshot should be done with a button.
  - Countdown. If the user clicks the button, start a countdown and display the remaining time as an overlay on the webcam stream. The countdown should be at least 3 seconds long. If you want, you can offer to adjust the countdown time.
  - Play a shutter sound when the snapshot is being taken.
- **Download**:
  Once a snapshot has been taken, the user can download the image as a PNG file by clicking a corresponding button. Make sure to pre-define a meaningful name for the file.

Please turn to the next page.

- Optional:
  - Filters
    Apply image filters to the video/image, e.g. a black-and-white or sepia filter. The user can activate a filter e.g. by clicking on the video or selecting from a drop-down list.
  - History
    Append thumbnails of all snapshots to the document. The user can re-select a snapshot and thereby download a previously taken snapshot.

Remarks:

- It is okay to use libraries for this submission. There are plenty on GitHub, e.g. Camera.js (https://github.com/idevelop/camera.js)
- Put your files into the subfolder 'task1'

Task 2: WebRTC

In the tutorial, you have shortly been introduced to WebRTC.

a) Briefly explain the benefits of WebRTC.
b) Find out through on-line research, why TURN and STUN servers are necessary for WebRTC. Try and explain what they do.
c) PeerChat is a nice little implementation of a video chat app that leverages WebRTC.
   a. Clone the project from GitHub: https://github.com/Hironate/PeerChat
   b. How does the app decide whether to create or join a conversation?
   c. So far, PeerChat only allows video chatting between a maximum of two users. What would you have to change to allow communications between more than 2 peers?
   d. PeerChat uses Socket.IO – what responsibilities does it have for the video call?

Put your files into a subfolder 'task2'.

Further notes:

- Please make sure to comment your code sufficiently to facilitate correction of your submission.
- Incomplete submissions are welcome. However, please make sure to include a "README" text file to tell us how far you have come.