Simon Johansson, Jonas Danebjer, Erik Bjäreholt, Valthor Halldórsson

Design

Camera module:
On the client there is one camera module for each server. Each camera has a Receiver thread that listens for messages from the servers, one Sender thread which sends anything from the camera's mailbox, and one Animator thread sending frames to be displayed to the SystemMonitor. Each camera contains a FrameBuffer with the most recently received frames.

SystemMonitor:
Handles the states of the system, namely the synchronization and Movie/Idle mode. Uses mailbox to communicate with cameras for modechange.

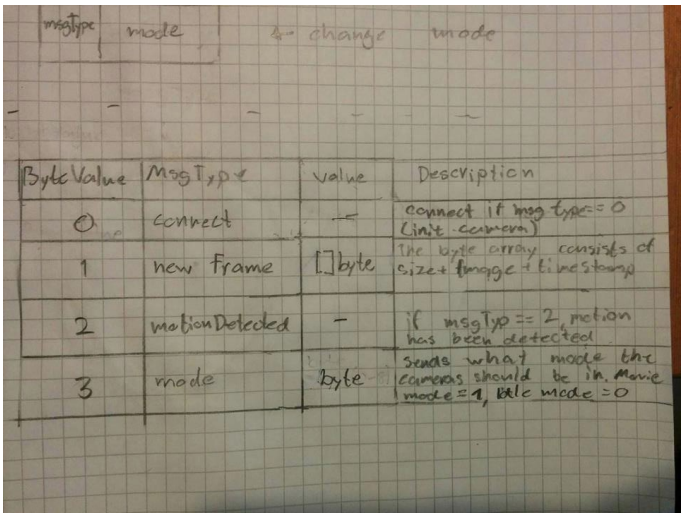
GUI:
The GUI is an Observer of the SystemMonitor.

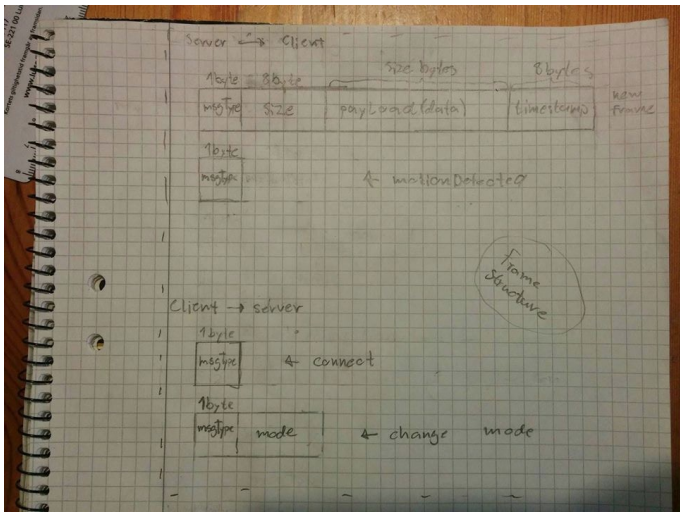
CameraServer:
The server consists of Sender and Receiver threads as well as an Updater thread that interacts with the hardware. Contains a monitor. The CameraServer itself is a thread which sets up connection and creates the rest of the camera.

Synchronous and asynchronous:
The animator sends frames to the SystemMonitor depending on the current mode, which is decided by incoming frames in the Receiver thread. The GUI observes changes in SystemMonitor and displays frames as fast as possible.

Messaging:
A message is sent from the Sender on the server-side and then received by the Receiver thread on the client-side. The structure of the messages are showed in the picture below. The receiver then adds the frame to the FrameBuffer while telling the SystemMonitor a frame has been received (for sync/async purposes). The Animator thread decides when a frame is to be displayed and then makes a copy to the SystemMonitor.

If the message is a motionDetected message it will instead instruct the SystemMonitor to change modes. The SystemMonitor handles all changes of modes (idle/movie and sync/async).





Simon Johansson, Jonas Danebjer, Erik Bjäreholt, Valthor Halldórsson

UML