Notes (Things to write in report)

* talk about Houdini's python api (hou) and how it works (maybe with a diagram?) and it's connection to the thumnail project
* talk about how the animated gif thumbnail is voluntary and can be disabled for better UX (since the thumbnail generation can take a very long time)
* the "-q" option is the antialias quality where
  + 0 = off,
  + 1 = low (2x),
  + 2 = medium (4x) and
  + 3 = high (5x or 6x)
* constraint, the library to be added to Houdini cannot be large (Pillow is about 1.2 MB which expands to \_\_\_\_)
  + this project also needs libjpeg for JPEG functionality and NumPy but this is already existing in Houdini so that is a bonus
  + PIL adds support for opening, manipulating and saving many different image file formats (support/development seems to be discontinued since 2009 with one last commit in 2011)
  + Pillow is a fork of PIL (with Python 3x support) "has been adopted as a replacement for the original PIL in linux distributions" and is officially the successor project for PIL
  + also considered
    - FFmpeg (convert images to video and convert avi to gif)
    - ImageMagick
    - mplayer (convert jpgs to gif and can specify the resolution, scale, and frame rate)
    - FFmpeg and mplayer are actual executables that are platform specific (not source code that can be imported as libraries) so they are not feasible; also, their features/options are overkill and not necessary for this functionality
    - gifmaker
      * cannot specify duration/delay
      * cannot loop in Pillow 2.7.0 (implemented for a future release) which was after the deadline
      * size is almost 3x as big (e.g. 6,271,716 bytes) for 10 frames
      * extremely blurry
      * on Github, it states gifmaker writes uncompressed GIF files only
* environment upon which the experiment data comes from:
  + Ubuntu 15.04
  + 64-bit linux
  + 101.1GB Disk
  + 15.7GiB memory
  + AMD Phenom (tm) II x6 1100T Processor (6 of them)
  + Quadro FX 3700/Pcle/SSE2 graphics card
* Task was to create animated thumbnails for the Pose Library in Houdini to help animate the characters
* amount of data stored in the JSON file (keep that as small as possible without compromising the quality of the playback)
* alter things like compression quality, resolution, playback image file size, FPS (difference between frame numbers and frame rate), frame delay -> find combination that gets the smallest size but best quality
* the images2gif.py script is better than the slower and inflexible gifmaker.py but it comes with its own compression method/strategy so there are no choices for compression quality there
* Look at code for scaling QMovie for first setting it and also for zoom (criteria is flexibility in scaling for zooming)
* Logics of the code:
  + calculate closest resolution with the same aspect ratio as viewport to 250x250 (but at least 250x250)
  + create JPGs and convert to GIF
  + delete those JPGs
  + load gif data into byte array and delete GIF
  + set GIF to thumbnail QMovie label
  + scale for zoom features
  + write byte data to JSON
  + load data back into byte array from JSON