Whats been implemented:

- Client that successfully downloads from .torrent file
- Client has successfully seeded to its self on local host
- Client that has basic UI that allows for multiple active torrents at once.
- Torrent data is checked for corruption when loaded (pieces checked with info dictionary)
- Bencoding class
- HTTP response parsing class
- UDP/DHT/HTTP support for trackers
- Partial completion support. (partial download system)
- LRU cache when downloading large files. (in theory should be able to download files above size of ram. However it seems I have a memory leak.... to be determined.)
- Partially supported magnet links. Coded but not debugged :-(Something seems to be wrong with the Extension handshake. So I stopped trying to support it.

Primary Resources:
(general)
https://wiki.theory.org/BitTorrentSpecification
(DHT)
http://bittorrent.org/beps/bep_0005.html
(EXTENSION PROTOCOL → implemented but not debugged had to leave it out of final build.)
http://www.bittorrent.org/beps/bep_0010.html
(META DATA DOWNLOAD)
http://www.bittorrent.org/beps/bep_0009.html

How to read the these disorganized document:

-Outline.pdf is probably a good place to start. Although no complete it gives a ground up view from primitives to the more complex data structures.

-Java Docs probably the next best place to try to figure out the layout of the code base.

-Observations.pdf and Designs.pdf were living documents that I was writing as I was going along.

A quick review on my thoughts:

I love the concept of magnet links. And the implementation was kinda interesting to. You get to wander around the DHT in search of peers that can give you the info dictionary.

While nothing in particular was a challenge to implementation (70% of this project was just parsing), there was a lot that had to be implemented. At ~5k lines of code its surprising just how little was actually implemented, I've certainly gained a lot more respect for my bittorent client.