Overview

Part 1 – Encoding

* Find the function that decodes .png (use ffmpeg function)
* Create our encoding to add to ffmpeg
* Incorporate the encoding into ffmpeg
* Video support for .utah
* Test
* Goal: .utah images display using ffplay. .utah images can be converted to video files.

Part 2 – Create the animator

* receive a png
* Use ffmpeg libraries to overlay sphere on the png
* Output n files as .utah
* UI
* Test
* Goal: It compiles without error. A WORKING APP. A+

Difficulties, Solutions, and Time Management

Part 1 –

Difficulties

1. Create encoding
   1. Understading how encoding/compression works (5-6 hours per)
      1. Reseach how encoding works
      2. Learn from other patterns
   2. Creating an efficient algorithm (10-20 hours total)
      1. Learn through examples
      2. Do it by hand.
      3. Test in another language if possible
2. Incorporate .utah support in ffmpeg (15hours total)
   1. Making it compile
      1. Complie frequently to check syntax and other small easy to miss errors.
   2. Functional makefile
      1. Fully understand the makefile
      2. Create a skeleton compile initially. Fill in details later
   3. Ensure **FULL** support. Everything is plugged in properly
      1. Ensure each component supports .utah. video play, overlay, reading, output, etc.

Part 2 –

Difficulties

1. ffmpeg Overlay (10hours, 3 hours per for reasearch)
   1. figure out how it works
      1. read and research the docs
      2. do a few test examples with native functions
2. outputting several .utah files with the overlay image (10-15 hours)
   1. outputting several files
      1. study and learn
      2. test examples
3. UI (5 hours)
   1. integrating the ffmpeg libraries with the app
      1. use the skeleton process
      2. understanding how to incorporate libraries

Summary

* bouncer exe (ui) C++
* bouncer app C++
* should not be creating lib but modifying
* will add our own .utah codec to avcodec lib
* Workflow:
  + UI will take care of communicating with the user and nothing else
  + UI sends params to the bouncer app
  + Bouncer will use various ffmpeg libs to create .utah images
    - Image overlay class
    - Output files class
    - Decode png image
    - Encode utah image
  + User then creates video with ffmpeg using .utah files
* IT COMPILES AND WORKS