

## 6.111 Final Project Checklist

Dimitri Turbine, Tyler Hutchinson, Bryan Newbold

*November 14, 2008*

**Project Title:** Realtime Visual Audio Composition

### **GUI, VGA Output, User Interaction**

**1024x768 VGA GUI display** resembling mockup in proposal/presentation

**ASCII text rendering**

**GUI sprite interaction** mouse button clicking

**Display of spectrogram with sweeping progress line**

*Scan line adjustment and shifting (if time)*

**Graph display of active FFT sample**

*Colorful FFT display (if time)* aka “Fire” effect

**Interactive bode plot** mouse-dragable point, display of resulting frequency response

**Appropriate taps written to filter stage**

**Image capture with minimum freeze time**

*Pixel edit mode with free-form line drawing capabilities (if time)*

**Individual Fourier transform to two-dimensional view**

*Selectable pole-zero filter diagram mode (if time)*

*Regression of bode plot to allow for line smoothing (if time)*

### **Camera Processing, Memory Management**

**No dropped VGA requests or NTSC input pixels** (everything stored in BZT RAM and all requests fulfilled)

**Processing of NTSC input into 8 bit pixel intensities**

*Configurable input processing (if time)*

**Provide samples to IFFT modules**

**Provide samples to GUI modules**

**Editing of sample memory on request from GUI**

## **Audio Output, Filtering, Sampling**

**Functioning audio output** with mute and volume control

**Discernible 128 tap filter functionality** at least with dummy tap values

**Recognizable figures in spectrogram of audio output** as measured with external signal analyzer or laptop. For example, smiley faces, straight lines, periodic dots

**Pausing/unpausing and restarting of IFFT module with no audio artifacts** (audio output muted until valid output available)

**Full 48 kilosample AC97 output (upsampled from 24 kilosamples)**

**Vector interpolation**

*Variable vector interpolation to adjust playback speed (if time)*

**720 sample to 1024 sample vector conversion**

*Stereo output using main/overlay as left/right channels (if time)*