Using Audio to Enhance Gameplay

RICE Unconventional Wisdom
Department of Electrical and Computer Engineering
Taylor Vaughn, Galen Schmidt, Jeremy Hunt

Goals and Motivation

- Develop a platform for generating audio driven events on the fly
- Apply this platform to a proof of concept game
- Develop methods of song analysis and coordinating game events

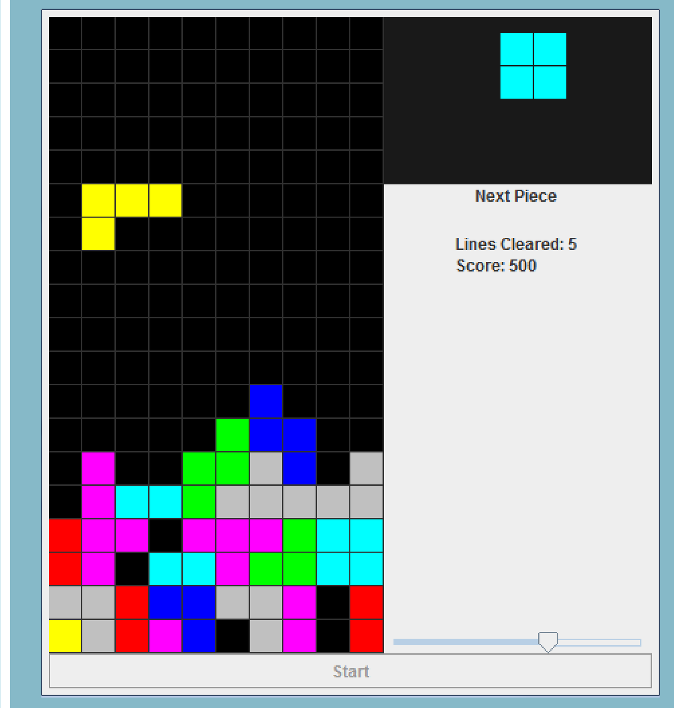
Game Play and Implementation

- Our proof of concept game is Tetris
- Three areas of signal and game play interaction

Signal Processing		Realization
Tempo Analysis	→	Pulsing Image
Frequency Analysis	→	Piece Selection
Amplitude Analysis	→	Fall Speed

Audio Tetris Strategy

In Tetris the next piece is displayed in the upper corner. The introduction of piece selection based on current frequency allows for new strategy. As the next piece segment updates the player decide whether or not to set down their current piece quickly or to hold on to it, hoping for a more desirable piece.



- At start, Audio Tetris will ask for an audio file (.mp3 or .wav)
- A progress bar indicates the time elapsed in both song and game
  - The game ends when the song end or game board is maxed out

Controls:
- ⇦ ⇨ Move piece
- ⇧ Rotate piece
- ⇩ Drop piece

Score = lines cleared * 100

Signal Analysis for Gameplay Interaction

Tempo Analysis

- Beats are detected by comparing low band instantaneous energy to low band local energy

$$E_{local} = \frac{\int [FFT(S_{local})]^2}{samples} \qquad Einstant = \int [FFT(Sinsant)]^2$$

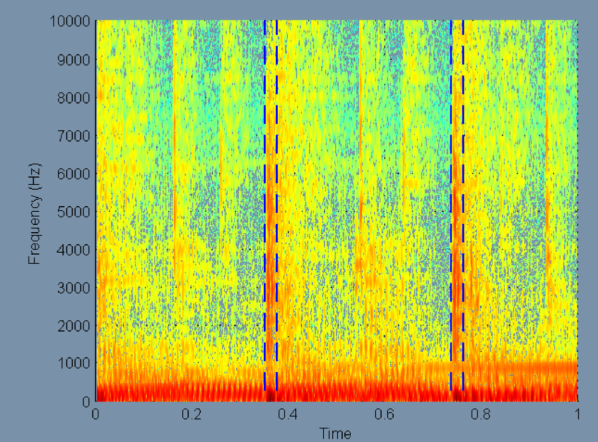
- If $E_{instant} > C{*}E_{local}$, a beat is recorded into a vector
  - C is computed from variance of the local energy

$$\sigma^2 = \frac{\int (Einst_{ant} - Elocal)^2}{samples}$$

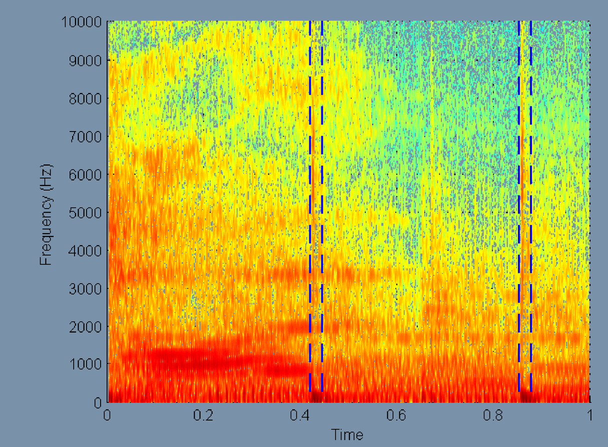
$$C = (-0.0025714 \cdot \sigma^2) + 1.5142857$$

- The vector is run through a filter to throw out double recorded beats
  - This occurs if the energy spike lasts for more than a single sample

Tempo Analysis of "Don't Shut 'Em Down" by Flogging Molly



Tempo Analysis of "Gimme Sympathy" by Metric



Frequency Analysis to Determine Next Piece

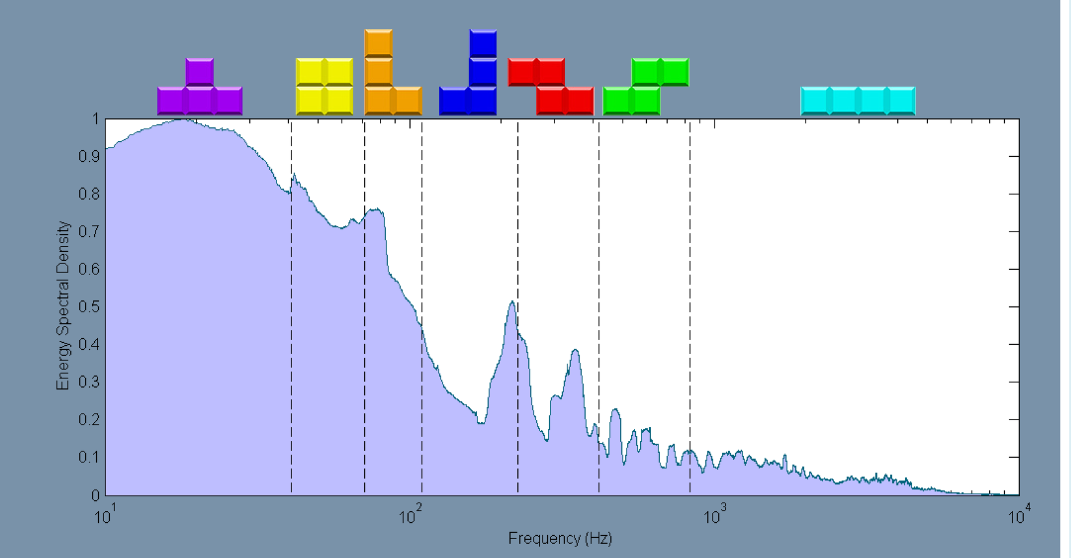
- The next piece is selected based on which frequency band contains the most energy
- The frequency spectrum is divided into seven segments with equivalent energy
  - For a chunk of time, I = vector of n frequency divisions

$$\int_{I[1]}^{I[2]} [FFT(S_{local})]^2 = \int_{I[2]}^{I[3]} [FFT(S_{local})]^2 = \ldots = \int_{I[]}^{I[n+1]} [FFT(S_{local}]^2$$

- The instantaneous energy in each segment is calculated
  - For instant in time dx

$$piece = \underset{i \in Pieces}{argmax} \left[\int_{I[i]}^{I[i+1]} [FFT(S_{instant})]^2 \right]$$

- This process is repeated for each time chuck and each instant in the song

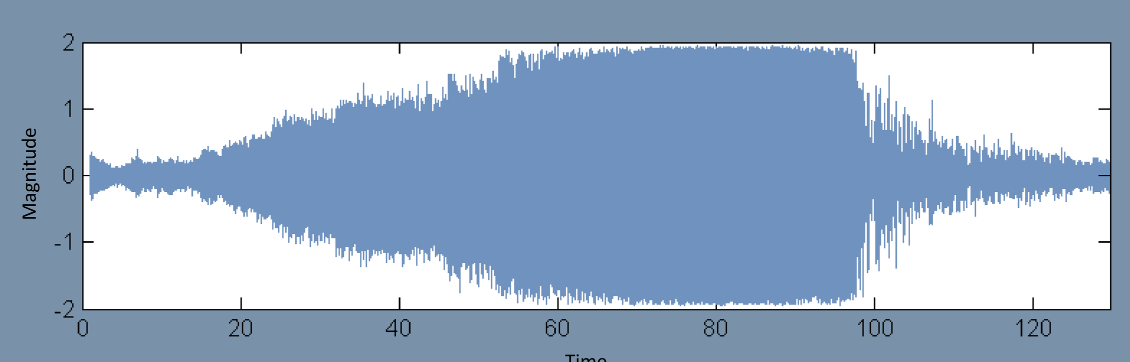


Amplitude Analysis to Determine Fall Speed

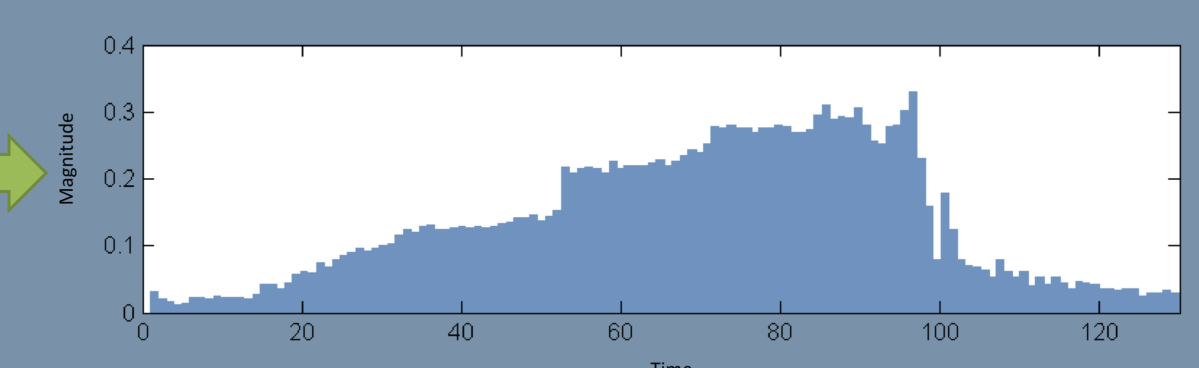
- The speed at which the current piece falls is dependent on the current relative amplitude
- Envelope at time τ for sample size $= \mathrm{C}_{speed}$

$$Envelope[\tau] = \frac{\sum_{t=\tau}^{\tau+S} abs(signal[t])}{S * \max[abs(signal)]}$$

$$speed_{cur} = speedmax \cdot Cspeed$$

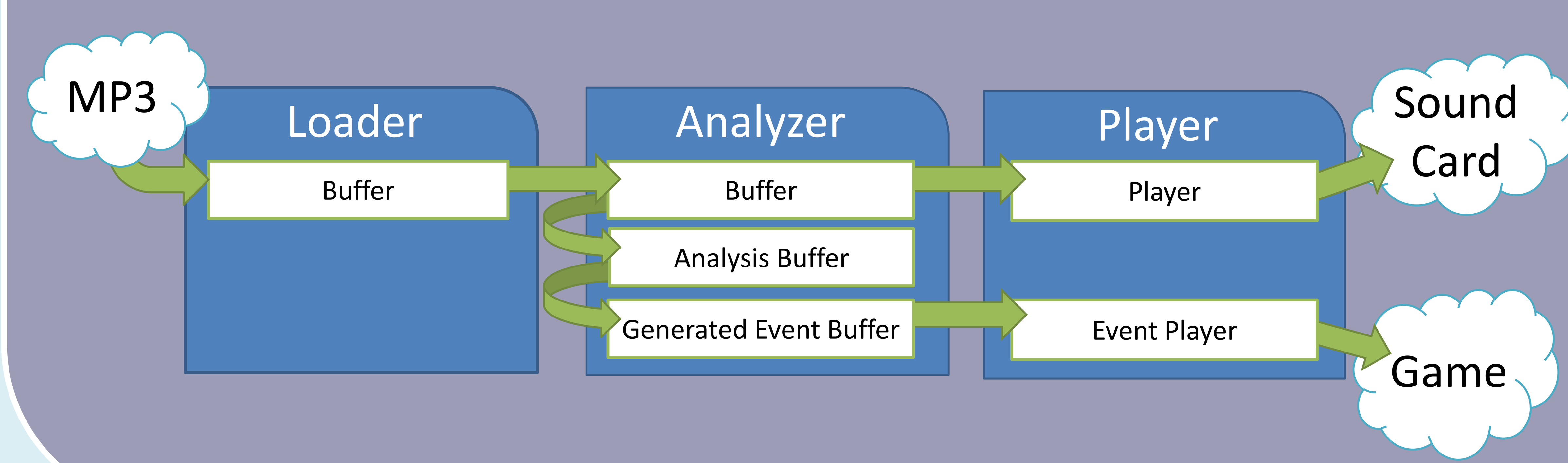


Raw Signal



Generated Envelope

Pipelined Processor for On-The-Fly Analysis and Play Back



3-Stage Parallel Pipe-lined Sound Engine

1. Loader
   - Translates input file
   - Generates buffer of sound to sent to other stages
2. Analyzer
   - Performs time and frequency domain analysis
   - Generates time-correlated events and corresponding song buffer
3. Player
   - Sends sound from buffer to sound card
   - Performs correlating events on game

Thanks to Dr. Richard Baraniuk, Raajen Patel, Eva Dyer, Dr. Tracy Volz, and Jianing Shi

Reference: "Beat Detection Algorithms" – Federic Patin *http://archive.gamedev.net/archive/reference/programming/features/beatdetection/index.html*