Intro to ChucK

Ge Wang
Fall 2008

Code == Musical instrument

ChucK

On-the-fly Programming

Audicle

>
=> syntax

- simple chuck: \( x \Rightarrow y \);
- chain chuck: \( w \Rightarrow x \Rightarrow y \Rightarrow z \);
- nested chuck: \( w \Rightarrow (x \Rightarrow y) \Rightarrow z \);
- un-chuck: \( x =< y =< z \);
- up-chuck: \( x ^= y ^= z \);

ChuckK Timing Constructs

- \textit{dur} is a native type
  - units:
    - \textit{samp}, \textit{ms}, \textit{second}, \textit{minute}, \textit{hour}, \textit{day}, \textit{week}
  - arithmetic:
    - \( 3:\text{second} + 100:\text{ms} \Rightarrow \text{dur}\) \(\text{quarter} \)
- \textit{time} is a native type
  - \textit{now} keyword holds current chuck time
  - arithmetic:
    - \( 5:\text{second} + \text{now} \Rightarrow \text{time}\) \(\text{later} \)

Advancing Time

- time stands still until you "advance" it
- you are responsible for keeping up with time
- time == sound

Concurrent Audio Programming

\begin{verbatim}
Impulse i => BiQuad f => dur;

// time loop
while( true )
{
  // impulse train
  1.0 => i.next;
  80::samp +=> now;
}

0.0 => float t;
while( true )
{
  // sweep center freq
  Math.sin(t) => f.freq;
  t + 0.01 => t;
  100::ms +=> now;
}
\end{verbatim}

Concurrency

- implemented using "shreds"
  - resemble non-preemptive threads
- automatically synchronized by time!
- can work at low and high level
- unifies control and audio rate

ChuckK Virtual Machine

- Code ("foo.ck", "bar.ck")
- Process
- Shred
- Scheduler
- Chuck Virtual Machine
On-the-fly Programming
(running with sonic scissors)

“Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.”

(Ada Lovelace, 1843)

on-the-fly programming:
(n.) the act of modifying the logic and structure of a program during runtime, for the purpose of rapid experimentation, and exerting expressive control. (also live coding)

Power Tools Can Maim
- power to spork many, many, many shreds
- power to precisely synchronize shreds
- edit and re-spork
- query for status...

But, Oops...
- which shred is which?
- which version of the edited code did I save?
- who is using all the processor cycles?
- what is the relative timing of the shreds?
- who is clipping?

The League of Automatic Composers (1974)

The Audicle
The Audicle

- visualization (audio, runtime stats, scheduling, etc.)
- insight into real-time, live programs
- different views of programs
  - syntax (code, objects)
  - concurrency (shreds)
  - time and timing (time, timing)
- different view of programming process

demo

http://chuck.cs.princeton.edu/

http://chuck.stanford.edu/