

Max @ LCM

Scott Hewitt

2012 - 2013

Outline

Revision Information

Build Date:

2013/01/09 04:52:24 +0000

Version Hash:

95ac419

95ac41932cd4bc4f4329412e0637ca628819d5fa

Scott Hewitt

scotthewitt.co.uk

Hello Max

Max @ LCM
Session 1
Scott Hewitt

Scott Hewitt

- ▶ scott@scotthewitt.co.uk
- ▶ scotthewitt.co.uk/cal

Expectations

- ▶ Assignment
- ▶ Attendance
- ▶ Contribution

Resources

- ▶ Lectures and Tutorials
- ▶ Max/MSP Forums cycling74.com
- ▶ Max/MSP Tutorials
- ▶ The Computer Music Tutorial - Curtis Roads
- ▶ Composing Interactive Music - Todd Winkler
- ▶ Electronic Music and Sound Design - Alessandro Cipriani

Max/MSP is

- ▶ A graphical program environment
- ▶ Uses patch cords
- ▶ to connect Max or MSP objects together
- ▶ Max for control data such as midi
- ▶ MSP for audio

Max/MSP need to know

- ▶ Max Modes Edit, Presentation, Locked
- ▶ Execution is right to left, top to bottom
- ▶ MSP objects have a ~
- ▶ MSP patch cords have stripes
- ▶ DSP can be turned on/off as needed
- ▶ Help files can be found by alt clicking on objects

DSP Settings

- ▶ Options → DSP Status
- ▶ Select the correct Audio Driver
- ▶ Also check the Sound system preferences

Example Delay Patch

Lets build and save a basic delay patch.
Its patching time!!!!!!

Delay patch Objects

- ▶ Max Object
 - ▶ *slider*
 - ▶ *flonum*
 - ▶ ***
- ▶ MSP Object
 - ▶ *ezdac*
 - ▶ *ezadc*
 - ▶ **~*
 - ▶ *delay~*

Max @ LCM Session 2 Scott Hewitt

Today

- ▶ Commenting
- ▶ Screen Feedback
- ▶ Bang

Commenting

- ▶ Important
- ▶ Direct
- ▶ Targeted
- ▶ Position
- ▶ *Comment*

Purpose

- ▶ Programmer
- ▶ User
- ▶ Assessment

For Programmer

- ▶ Highlighting Functionality
- ▶ Explanation of code
- ▶ Explanation of data
- ▶ Development Notes
 - ▶ to self
 - ▶ working / broken
 - ▶ questions

For User

- ▶ Instructions for use
- ▶ Suggested Values
- ▶ Credit
- ▶ Copyright

For Assessment

- ▶ Identify required elements
- ▶ Detail user actions
- ▶ Indicate understanding
- ▶ Demonstrate further reading

Example

On Screen Feedback

- ▶ Check code operation
- ▶ Confirm correct order
- ▶ Patch state

Feedback Objects

- ▶ *number*
- ▶ *flonum*
- ▶ *print*
- ▶ *message*

number / flonum

- ▶ Display last number
- ▶ flonum for floats
- ▶ Check maths
- ▶ (Also can be user input)

Example

Print Object

- ▶ *print* messages to Max window
- ▶ Useful for illustrating change in value
- ▶ Argument to differentiate

Example

Message Box

- ▶ Display max message
- ▶ Resend Value
- ▶ Store Value

Example

Bang

- ▶ Do it
- ▶ Causes objects to do and send output
- ▶ Message bang *button*

Example

Today

- ▶ *comment*
- ▶ *number*
- ▶ *flonum*
- ▶ *print*
- ▶ *message*
- ▶ *bang*

Task

- ▶ Commenting the bang
- ▶ Add print objects
- ▶ Adapt patch so either slide causes execution of code

Max @ LCM
Session 3
Scott Hewitt

MIDI Input

- ▶ MIDI Resolution
- ▶ MIDI Device

MIDI Input Objects

- ▶ *midiin*
- ▶ *notein*
- ▶ *bendin*
- ▶ *xbend*
- ▶ *ctlin*

Scale

- ▶ *scale*
- ▶ Scale input values as required
- ▶ Consider
 - ▶ Purpose
 - ▶ Required Range
 - ▶ Resolution

Non Continuous / Continuous Controllers

- ▶ Detect Movement
- ▶ Absolute Value

Desired Behaviour

- ▶ Automation Modes
- ▶ Pickup
- ▶ Jump
- ▶ Relative

Key

- ▶ Discrete
- ▶ Bang
- ▶ On / Off
- ▶ *toggle*
- ▶ Modifier Keys
- ▶ *key*
- ▶ *keyup*

Mouse

▶ *mousestate*

Input Filtering

- ▶ *mousefilter*
- ▶ *speedlim*

This Week

- ▶ Midi Input
- ▶ Key
- ▶ Mouse
- ▶ MIDI Input
- ▶ Key
- ▶ Mousestate

Todo I

- ▶ Panning Patch
 - ▶ Controlled by
 - ▶ Key
 - ▶ Mouse
 - ▶ Controller

Todo II

- ▶ Tutorials
 - ▶ MSP Tut 1, 2, 4, 5, 22
 - ▶ Max Tut 1 - 8

Max @ LCM Session 4 Scott Hewitt

Metro

- ▶ Trigger Events at set tempo
- ▶ bang message

Line

- ▶ Ramp from value to value
- ▶ number message
- ▶ float version
- ▶ Value, Value Time

Drunk

- ▶ Stagger between values

Random

- ▶ Range from 0 to value
- ▶ Consider *urn*

Loops

- ▶ Cause processes to loop

Rate of Control

- ▶ Max Rate
- ▶ MSP Rate

Line~

- ▶ Signal Rate
- ▶ Envelope
- ▶ *function*

Data Storage

- ▶ Application use
- ▶ User use

Basic

- ▶ Number Box
- ▶ User Interface
- ▶ int / float

Presets

- ▶ User Interface
- ▶ Include / Exclude

Lists

- ▶ A list of other Max data types
 - ▶ int, float, symbol
- ▶ Saves patch chords
- ▶ Conceptually cleaner
- ▶ Simplifies communication

List Objects

- ▶ *unpack*
- ▶ *pack*
- ▶ *pak*
- ▶ *bondo*
- ▶ *zl*

Objects

- ▶ *metro*
- ▶ *line*
- ▶ *line~*
- ▶ *drunk*
- ▶ *random*

ToDo

- ▶ Build automated system that moves between presets

Max @ LCM
Session 5
Scott Hewitt

User Interface

- ▶ Display Information
- ▶ Current Parameters
- ▶ Fault Testing
- ▶ Signal Levels
- ▶ Interaction
- ▶ Entice

Problems

- ▶ Misleading
- ▶ Inaccurate
- ▶ Crowded
- ▶ Unstable

Object Inspector

- ▶ Object attributes
- ▶ Appearance
- ▶ Range
- ▶ Inclusion in views

Hide on Lock

- ▶ Position effects execution
- ▶ Relates to program structure
- ▶ Older method
- ▶ Prevents interaction

Presentation

- ▶ Position does not effect execution
- ▶ Clean
- ▶ User interface
- ▶ Hard to debug
- ▶ Difficult to redevelop

Programmer

- ▶ Annotation
- ▶ Hint

Patcher Inspector

- ▶ Set patcher wide attributes
- ▶ Load in presentation mode option

Max @ LCM
Session 6
Scott Hewitt

An Object

- ▶ Name
- ▶ Argument
- ▶ Inlet
- ▶ Outlet
- ▶ Global Memory

Encapsulation

- ▶ Sub patcher *patcher*
- ▶ Tidy up code
- ▶ Encapsulated code with purpose
- ▶ Mapping to musical process

Object Style Building Method

- ▶ Identify smallest behaviour and build
- ▶ Reuse
- ▶ Quick

Abstraction

- ▶ Subpatchers designed for reuse
- ▶ Clear purpose

In use

- ▶ Seperate instance
 - ▶ play ~, delay /*sim*
- ▶ with Shared memory
 - ▶ Buffers, Sends, receives
- ▶ #tags

bpatcher

- ▶ Abstract interface
- ▶ Portable
- ▶ Managable
- ▶ Object inspector for arguments and patcher
- ▶ Patcher inspector for open in presentation

Why Abstract

- ▶ Unit test
- ▶ Document
- ▶ Inter-communication
- ▶ Speed

Tips

- ▶ Naming convention
- ▶ Save location
- ▶ Interface

This Week

- ▶ Encapsulation
- ▶ Abstraction

Max @ LCM
Session 7
Scott Hewitt

Message Routing

- ▶ Select
- ▶ Route
- ▶ loadbang

Audio Routing

- ▶ `matrix~`
- ▶ Amplitude change

Dynamic Routing

- ▶ *receive / send and forward*
- ▶ receive~ / send~
- ▶ Less patching
- ▶ Cross-patcher divide
- ▶ Post Load

Implementation


- ▶ Control channel
- ▶ Routed channels
- ▶ Audio degradation
- ▶ Crossfades

Combined with Abstraction

- ▶ Network-able
- ▶ Distributed
- ▶ Modular
- ▶ Mobile

Example

- ▶ A limitless ¹ delay

¹Of course eventually the computer will run out of resources. 

Max @ LCM
Session 8
Scott Hewitt

What is interactive art?

- ▶ Art that requires interaction
- ▶ Music that requires interaction

Interactivity

- ▶ Interaction means action
- ▶ A sense of participation
- ▶ Feedback
- ▶ Two way communication

Todd Winkler

... a music composition or improvisation where software interprets a live performance to affect music generated or modified by computers.

Composing Interactive Music

Todd Winkler (MIT Press, 1998)

An Extension/Progression of Collabration

Location

- ▶ Gallery
- ▶ Museum
- ▶ Schools
- ▶ Concert Hall

Implementation

Human Input -> Computer Listening -> Computer Interpretation
-> Computer Response/Composition -> Computer Performance ->
Human Response/Input

A question of Interface

Something Lost

- ▶ Pre Recording
- ▶ Pre Multitrack
- ▶ Live Performance

Historical Computer Music

- ▶ Non Real Time

What about Electronics?

- ▶ Live Electronics Systems
 - ▶ Cage
 - ▶ Mumma
- ▶ GROOVE (Max Matthews 1969)
- ▶ CV Control

Contemporary Computer Music

- ▶ Real Time expression through MIDI

Live Algorithms

- ▶ Voyager (G. Lewis)
- ▶ Swarms (M. Young)

Musical Models of Interaction

- ▶ Performance
- ▶ Instrument
- ▶ Composition

For Consideration

- ▶ What is of interest about interactivity?
- ▶ What interactivity do you miss?

Max @ LCM Session 9 Scott Hewitt

MIDI Fundamentals

- ▶ Note Message (0-127)
 - ▶ Pitch
 - ▶ Velocity
 - ▶ Aftertouch
- ▶ Note on, Note off
- ▶ Event in time

Typical Techniques

- ▶ Histogram (Tonality)
- ▶ Anal (Melodic Analysis)
- ▶ Borax (Rythm)
- ▶ Coll (Re-Synthesis)

Uses

- ▶ Playback
- ▶ Accompaniment
- ▶ Improvisor

Humanisation

- ▶ Random Elements
- ▶ Mistakes

This Week

- ▶ MIDI Analysis

Max @ LCM

Session 10

Scott Hewitt

Alternatives

- ▶ Live
- ▶ Reaper
- ▶ Csound
- ▶ PureData
- ▶ Supercollider
- ▶ ChuckK

Live

- ▶ Fast to work with
- ▶ Flexible
- ▶ Quick interface mapping
- ▶ <http://www.ableton.com>

ReNoise

- ▶ Strong Rhythm
- ▶ Tracker
- ▶ Powerful Audio Routing
- ▶ <http://www.renoise.com/>

Csound

- ▶ Powerful synthesis
- ▶ Flexible
- ▶ Non-real time
- ▶ Open Source
- ▶ <http://www.csounds.com/>

PureData

- ▶ Intergration of control rates
- ▶ Open Source
- ▶ Max descendant
- ▶ Not as supported
- ▶ Visual
- ▶ <http://puredata.info/>

SuperCollider

- ▶ Open source
- ▶ Text based
- ▶ Flexible powerful
- ▶ Built in polyphony
- ▶ <http://supercollider.sourceforge.net/>

Chuck

- ▶ Open source
- ▶ Text based
- ▶ Built in polyphony
- ▶ <http://chuck.cs.princeton.edu/>

Choices

- ▶ Chosen Tool
- ▶ One or Many