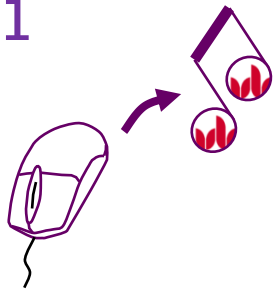


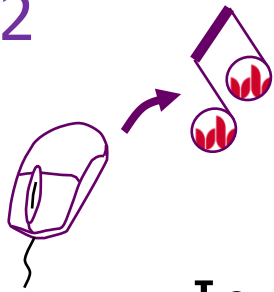
1



My Mouse – My Music



- Marc Conrad
- Tim French
 - Department of Computer Science and Technology,
University of Bedfordshire, UK.



History of the System

- In 2006: Attempt to create a production system for 'infinite' number of sonic events to be used for Authentication Systems.

musipass
Experimental musical password system

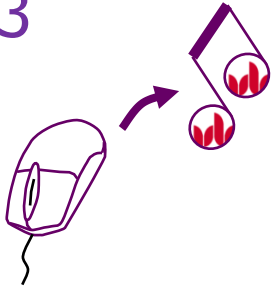
Home | Register | Log in | Credits | Help | Contact



References:

- Conrad, M., French, T., Gibson, M., *A Pragmatic and Musically Pleasing Production System for Sonic Events*, 10th International Conference on Information Visualization, London, 2006.
- Gibson, M., Renaud, K., Conrad, M., Maple, C., *Musipass: Authenticating me softly with "my" song*, NSPW, Oxford, UK, 2009.
- Gibson, M., Conrad, M., Maple, C., *"Accessible and Secure? Design Constraints on Image and Sound Based Passwords"*, i-society 2010, London, UK, 2010.

3

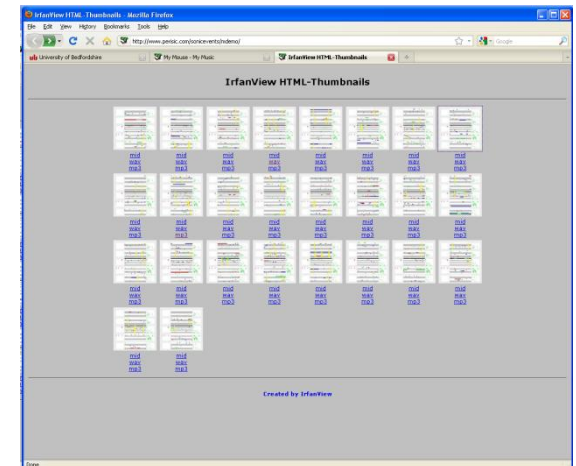


Sonic Events

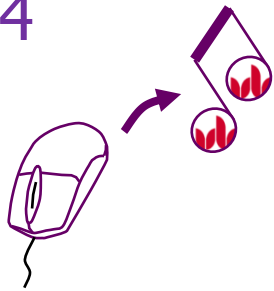
- Each musical (segment) of two seconds duration:
 - Distinguishable by a musically untrained ear.
 - Comprises multiple pre-defined MIDI "voices".
 - Randomly generated from initial "seed" values.
 - Virtually infinite number of permutations.

Pre-factored examples (mp3, wav) at:

<http://www.perisic.com/sonicevents/mdemo>



4



2006: Production System for Sonic Events

■ <http://perisic.com/sonicevents>

Sonic Events - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.perisic.com/sonicevents/

University of Bedfordshire Sonic Events

A Production System for Musically Pleasing Sonic Events

This web site contains additional information on [Marc Conrad's](#) and [Tim French's](#) Java production system for Sonic Events. Full details can be found in a [joint paper](#) with Marcia Gibson to be presented at [IV06](#).

This is the abstract:

We describe a novel application for sonic events namely their generation via mathematical functions implemented on a universal all purpose Java platform. Their design is driven by a set of requirements that arise in recognition-based authentication systems. We show that our approach has potential advantages as compared with traditional alphanumeric and other password systems. Our intention is to demonstrate that by leveraging familiar musical dimension and aesthetics human memorability, pleasure and pragmatics are enhanced. We demonstrate and briefly discuss one exemplar generative approach that has been specifically designed in order to fulfill the requirements implied by authentication systems. It is hoped that this work serves to stimulate debate and further activity in the field of computer generated sonics.


- [Sonic events demos](#) (in the formats midi, wav and mp3).
- [Download the Java package](#) (this includes source code). The jar file is executable, that means in a suitably configured system, double clicking on the package should show a simple user interface to produce sonic events.
- [The API documentation for the package.](#)
- [Miscellaneous information](#)

[\(last page\)](#)

The Java package is licensed under the [OpenSource fair licence](#):

Copyright: Marc Conrad, Tim French 2006. Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.
DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

The wav files have been produced with [TiMidity++ 2.13.0](#) using the [goemon patches](#). The conversion from wav to mp3 has been made with the [Eusing CD to mp3 converter](#) (Version 1.8). A big Thank You to all the folks that contribute free software to the community. Without you this project would not have been possible.

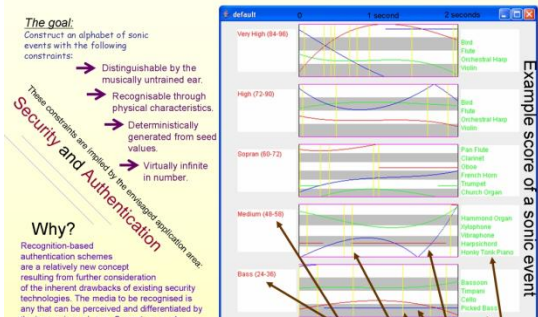
 © Marc Conrad, 2006-2009. The material on this page is presented "as is". There is no warranty implied by presenting this stuff. Feel free to use the material for your own research or teaching. When doing so please give a reference to this web site (<http://perisic.com/sonicevents/>) The webpage for this project is kindly provided by the [Perisic Guesthouse](#) (www.perisic.com). The author of this page is not responsible for external links.

Done

INSTITUTE FOR RESEARCH IN APPLICABLE COMPUTING

A JAVA PRODUCTION SYSTEM FOR MUSICALLY PLEASING SONIC EVENTS

Marc Conrad, Tim French, Marcia Gibson
Marc.Conrad@luton.ac.uk, Tim.French@luton.ac.uk, Marcia.Gibson@luton.ac.uk



The goal:
Construct an alphabet of sonic events with the following constraints:

- Distinguishable by the musically untrained ear.
- Recognisable through physical characteristics.
- Deterministically generated from seed values.
- Virtually infinite in number.

Why?
Recognition-based authentication schemes are a relatively new concept resulting from further consideration of the inherent drawbacks of existing security technologies. The media to be recognised is any that can be perceived and differentiated by the inexperienced user. Current research focuses on visual media, namely:

- Images
- Images-based passwords do however have inherent drawbacks of their own. Think about accessibility to visually impaired or situations where it is not possible to use a screen such as when authentication is required over a telephone. For these situations we need a better alternative, namely:

- ✓ Pitch
- ✓ Velocity
- ✓ Timbre

Each voice is fully determined by three mathematical functions:
 Red function: Midi pitch levels (middle C = 60)
 Blue function: Midi velocity
 Green function: Midi instruments
 Every yellow line fires a midi event

Implementation
 The application area of our sonic event production system is authentication: it does not exist as an isolated "music producing hand- or software" but is likely to be embedded as part of a wider system that meets industrial standards. Hence we use a modular, object-oriented code design that allows direct interaction with any other Java code.

Download sonic events on <http://perisic.com/sonicevents>

Supported by the Perisic Guesthouse <http://perisic.com/sonicevents>

University of Luton Education that works

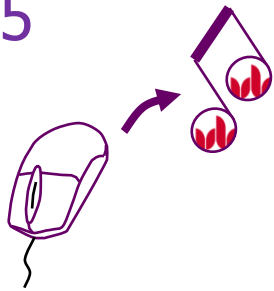
irac

10/09/2010 00:21:58

Marc Conrad
University of Bedfordshire

<http://perisic.com/mousemusic>

5



2006: Production System for Sonic Events

■ <http://perisic.com/sonicevents>

Sonic Events - Mozilla Firefox

http://www.perisic.com/sonicevents/

A Production System for Musically Pleasing Sonic Events

This web site contains additional information on [Marc Conrad's](#) and [Tim French's](#) Java production system for Sonic Events. Full details can be found in a [joint paper](#) with Marcia Gibson to be presented at [IV06](#).

This is the abstract:

We describe a novel application for sonic events namely their generation via mathematical functions implemented on a universal all purpose Java platform. Their design is driven by a set of requirements that arise in recognition-based authentication systems. We show that our approach has potential advantages as compared with traditional alphanumeric and other password systems. Our intention is to demonstrate that by leveraging familiar musical dimension and aesthetics human memorability, pleasure and pragmatics are enhanced. We demonstrate and briefly discuss one exemplar generative approach that has been specifically designed in order to fulfill the requirements implied by authentication systems. It is hoped that this work serves to stimulate debate and further activity in the field of computer generated sonics.

- [Sonic events demos](#) (in the formats midi, wav and mp3).
- [Download the Java package](#) (this includes source code). The jar file is executable, that means in a suitably configured system, double clicking on the package should show a simple user interface to produce sonic events.
- [The API documentation for the package.](#)
- [Miscellaneous information](#)

The Java package is licensed under the [OpenSource fair licence](#):

Copyright: Marc Conrad, Tim French 2006. Usage of the works is permitted provided that this instrument is retained with the works, so that any entity that uses the works is notified of this instrument.
DISCLAIMER: THE WORKS ARE WITHOUT WARRANTY.

The wav files have been produced with [TiMidity++ 2.13.0](#) using the [goemon patches](#). The conversion from wav to mp3 has been made with the [Eusing CD to mp3 converter](#) (Version 1.8). A big Thank You to all the folks that contribute free software to the community. Without you this project would not have been possible.

© Marc Conrad, 2006-2009. The material on this page is presented "as is". There is no warranty implied by presenting this stuff. Feel free to use the material for your own research or teaching. When doing so please give a reference to this web site (<http://perisic.com/sonicevents>) The webpage for this project is kindly provided by the [Perisic Guesthouse](#) (www.perisic.com). The author of this page is not responsible for external links.

INSTITUTE FOR RESEARCH IN APPLICABLE COMPUTING

A JAVA PRODUCTION SYSTEM FOR MUSICALLY PLEASING SONIC EVENTS

Marc Conrad, Tim French, Marcia Gibson
Marc.Conrad@luton.ac.uk, Tim.French@luton.ac.uk, Marcia.Gibson@luton.ac.uk

Security and Authentication

Why?
 Recognition-based authentication schemes are a relatively new concept resulting from further consideration of the inherent drawbacks of existing security technologies. The media to be recognised is any that can be perceived and differentiated by the inexperienced user. Current research focuses on visual media, namely **images**.

Image-based passwords do however have inherent drawbacks of their own. Think about accessibility to visually impaired or situations where it is not possible to use a screen such as when authentication is required over a telephone. For these situations we need a **better alternative**, namely:

- ✓ **Pitch**
- ✓ **Velocity**
- ✓ **Timbre**

Each voice is fully determined by three mathematical functions.

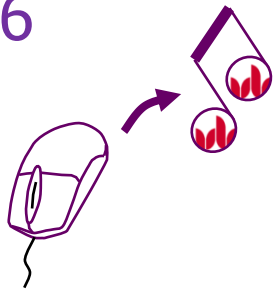
Implementation

100% Java!

Supported by the Perisic Guesthouse <http://perisic.com/sonicevents>

University of Luton Education that works

ILAC



Extensibility as a Key Factor

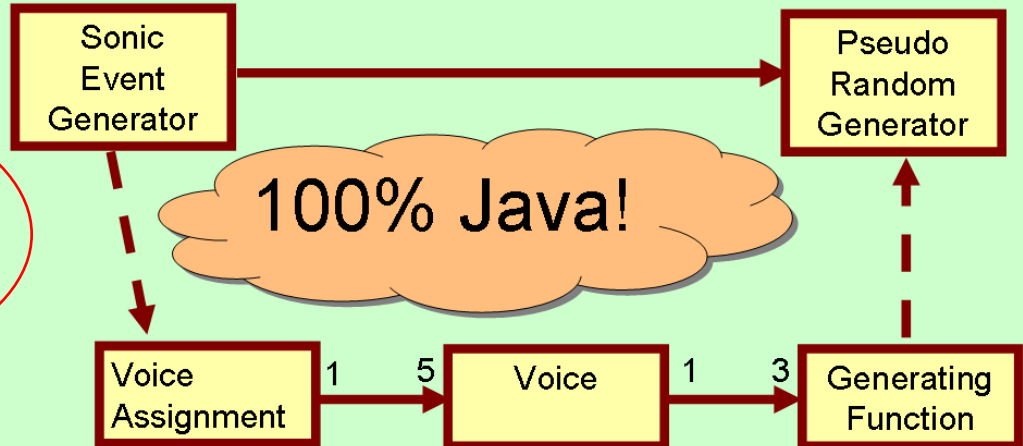
- The system of 2006 has been designed so that it can be used with other software

ver a
eed a

ely:

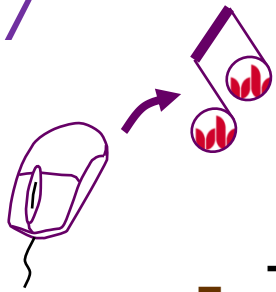
The application area of our sonic event production system is authentication. It does not exist as an isolated “music producing hard- or software” but is likely to be embedded as part of a wider system that meets industrial standards. Hence we use a modular, object-oriented code design that allows direct interaction with any other Java code.

Implementation



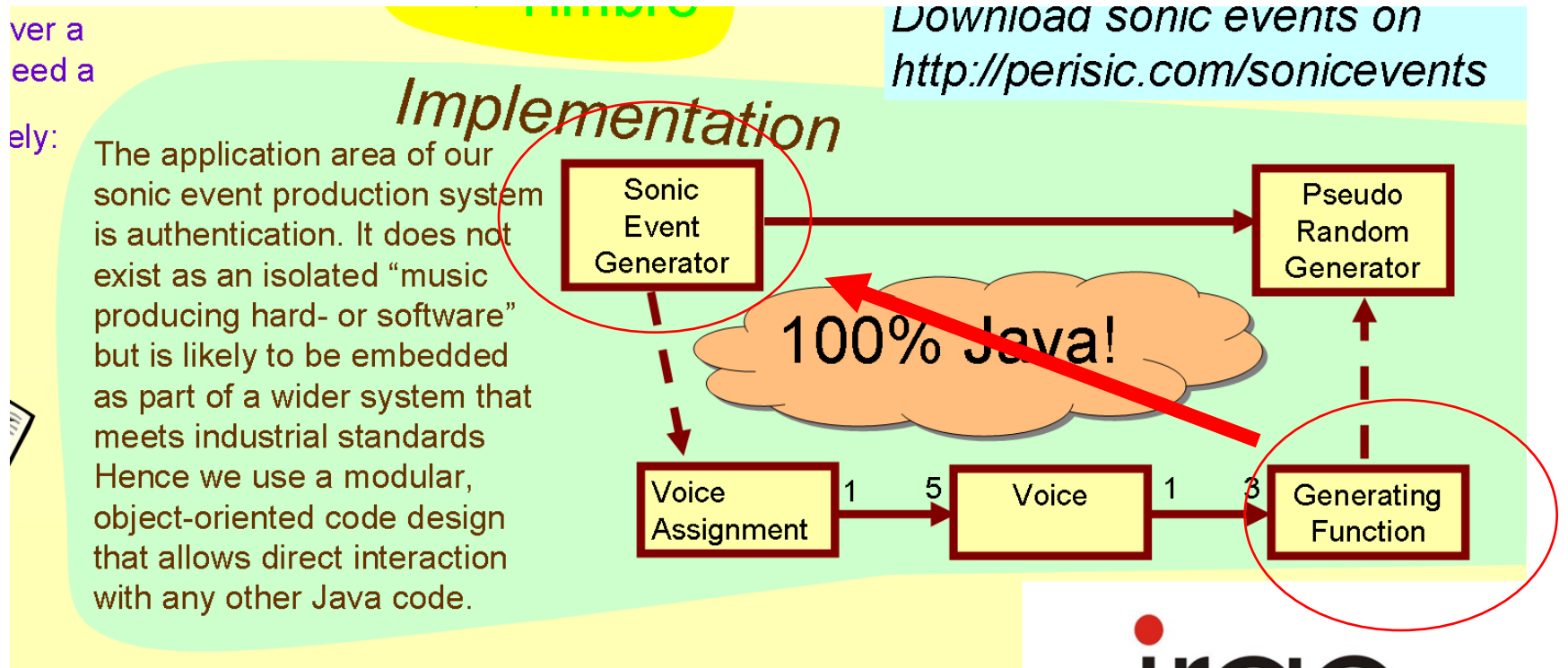
Download sonic events on
<http://perisic.com/sonicevents>





Sonic Events Driven by Mathematics

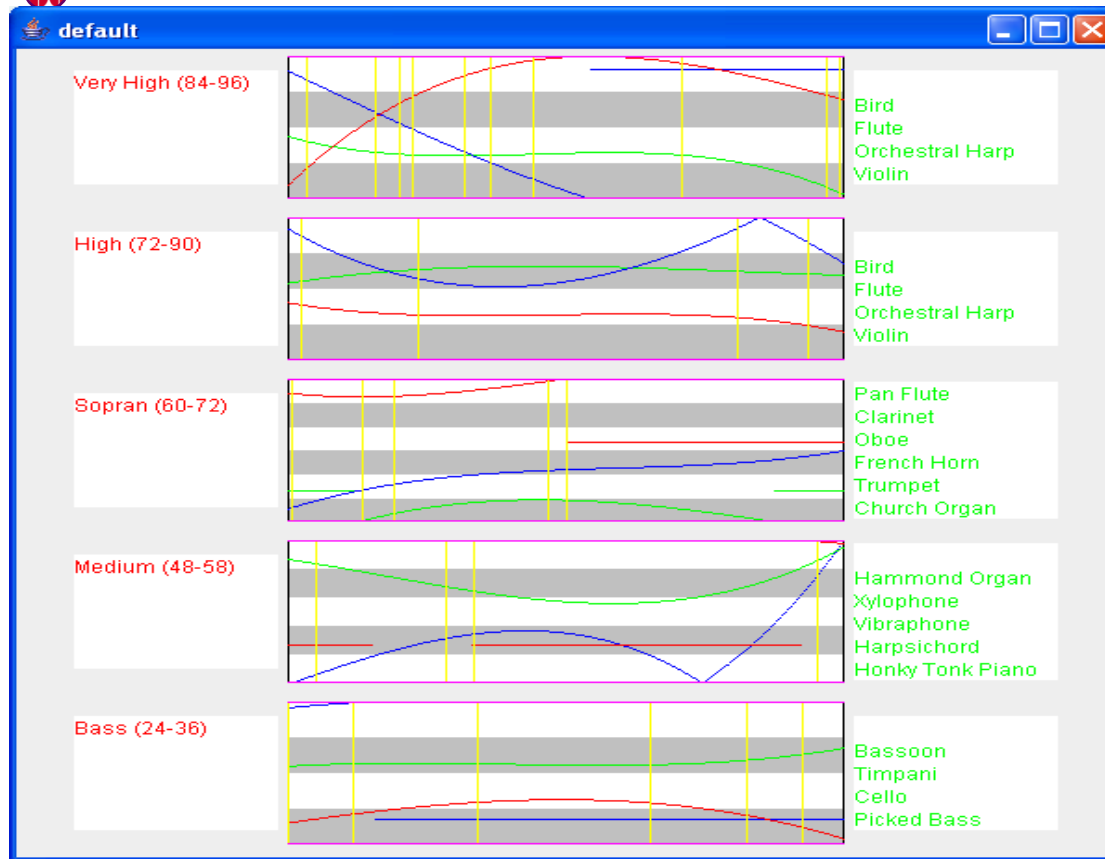
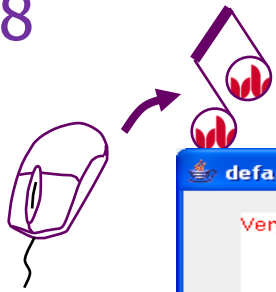
- The underlying mechanism doesn't use scores or musical notation but 'nice' mathematical functions.



8

Translate Mathematical Functions into Music

$16 = 5 * 3 + 1$ functions determine a sonic event.

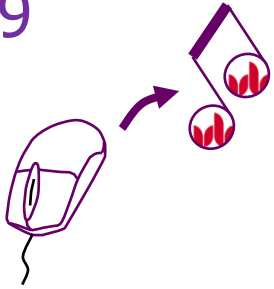


Green function:
(midi)
instruments

Blue function:
(midi)
velocity

Red function:
Midi pitch levels
(middle C = 60)

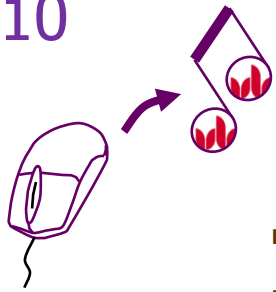
Every yellow line
fires a midi event
(one function to
determine these)



Mathematics and Music

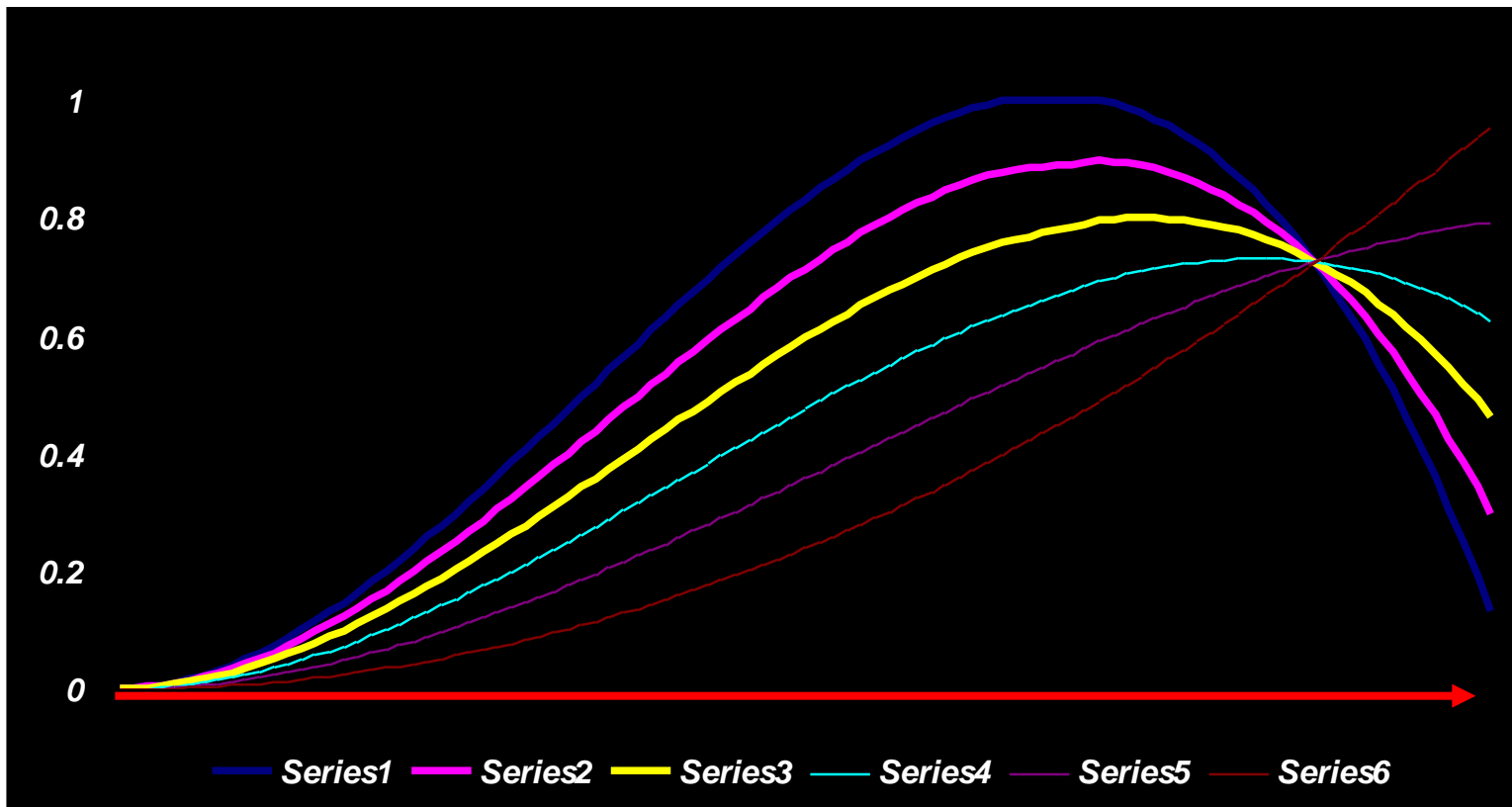
- If the music is to be defined by Mathematics then algebraic relations are used to relate sonic events to one other.
- The transition from one Mathematical function to another defines the transition from one musical segment to another.
- How can this be used for artistic expression?

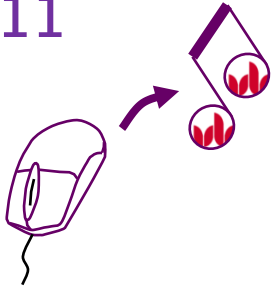
10



Transitions between Functions using Weighted Averages

- Picture shows weighted averages of $f_0 = \text{Series1}$ and $f_1 = \text{Series 6}$
- $\text{Series2} = 0.8 * f_0 + 0.2 * f_1$
- $\text{Series3} = 0.6 * f_0 + 0.4 * f_1$
- Etc.





Even More: Define Transitions between Four Functions.

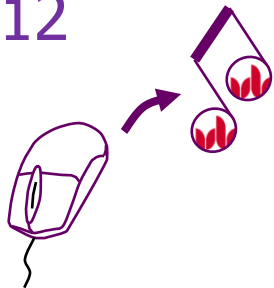
- Four functions: $f_{0,0}$, $f_{0,1}$, $f_{1,0}$, $f_{1,1}$
- For a point (x,y) define:
- $g_{x,y} = (1-y)(1-x) f_{0,0} + (1-y)x f_{1,0} + y(1-x) f_{0,1} + yx f_{1,1}$

■ Observe:

- $g_{0,0} = f_{0,0}$
- $g_{1,0} = f_{1,0}$
- $g_{0,1} = f_{0,1}$
- $g_{1,1} = f_{1,1}$

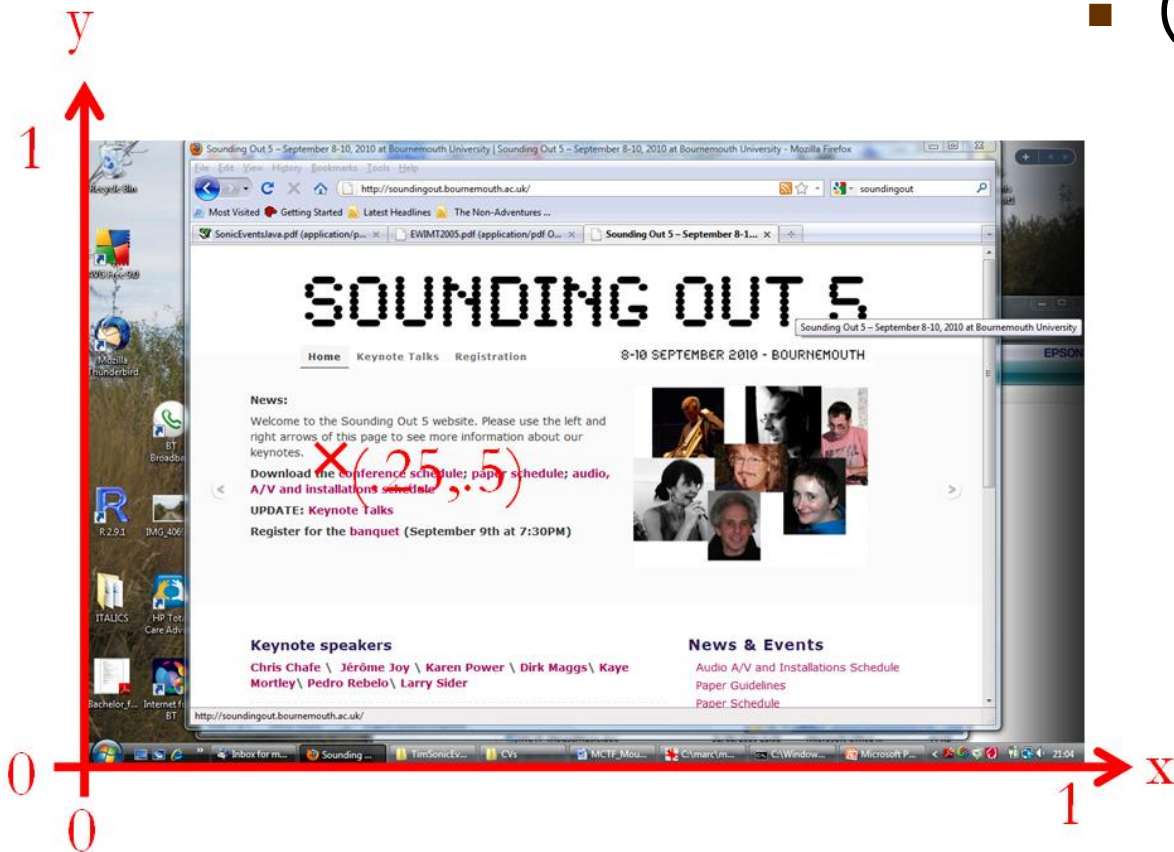
If $(x,y) \sim (u,v)$ are "close"
then $g_{x,y} \sim g_{u,v}$ are "close"

12

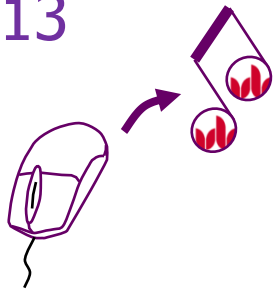


The Screen as a Coordinate System

- Example:
- $(x,y) = (0.25,0.5)$

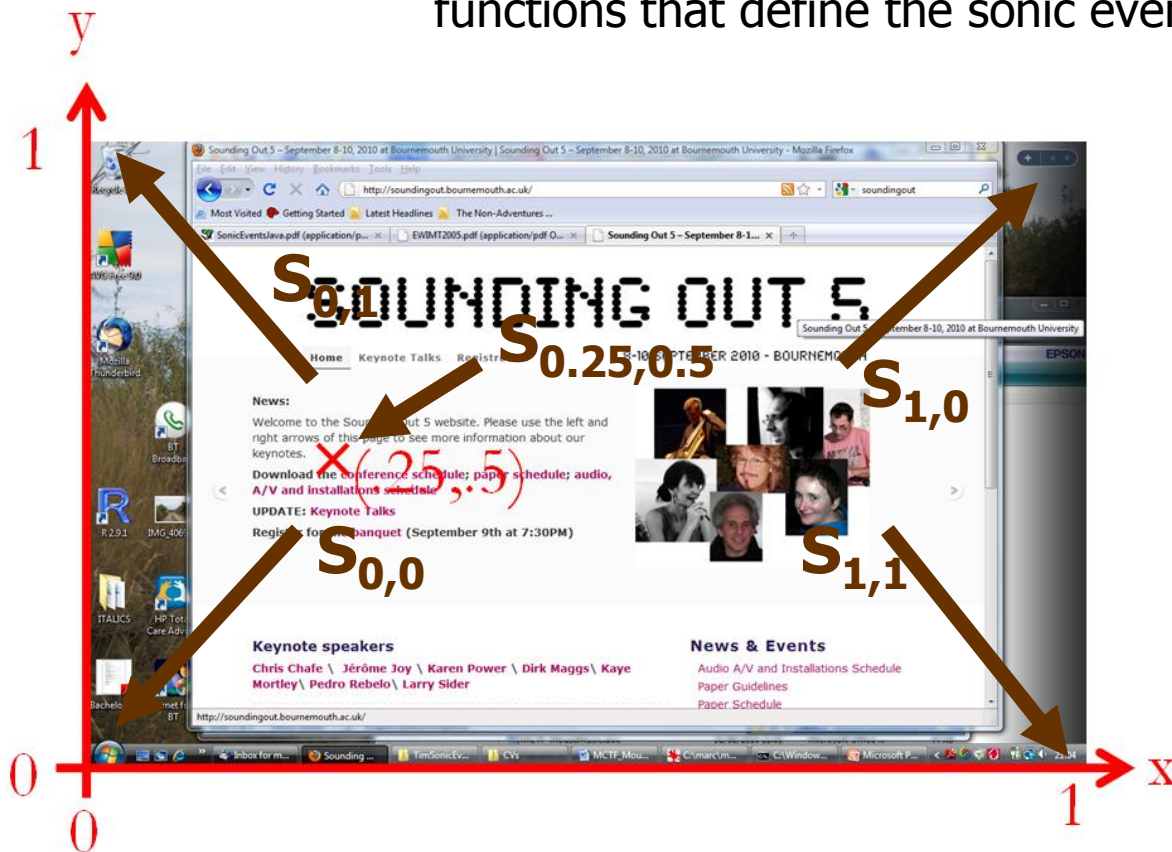


13

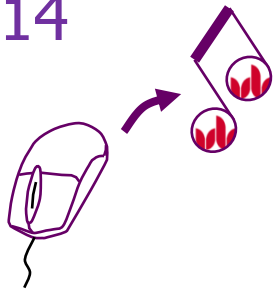


The Screen as a Coordinate System

- Define four sonic events $S_{0,0}$, $S_{0,1}$, $S_{1,0}$, $S_{1,1}$ for the four corners of the screen.
- Each sonic event is defined by 16 functions.
- Use the formula on previous slide to get the 16 functions that define the sonic event $S_{0.25,0.5}$,



14



Different 'backgrounds' act as *player stimuli* leading to aesthetic ways of exploring the resultant 'sound space'

1

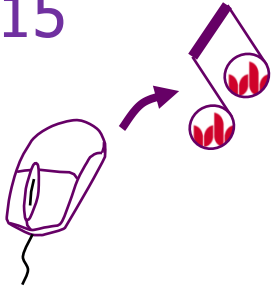
My Mouse – My Music

- Marc Conrad
- Tim French
- Depa University

10/09/2010 00:21:58

Marc Conrad
University of Bedfordshire

■ Compare with Xenaki's UPIC system and similar approaches (e.g. HighC)



Further Discussion / Questions

- Use of Java Midi output seems to be partly dependant on the choice of hardware platform?
- Which kinds of Sonic Event “seeds” should be assigned to the four corners of the screen?
- Current system allows to switch between three fixed settings. Should that be enhanced or is it already too much?
- Should “randomness” be added? How?
- Could similarly be used in 3-dimensional space.
- Also in 4+ (n) dimensions – but is there an application / interface?
- Etc.



Thank you! Any questions?