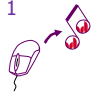




1 **My Mouse – My Music**


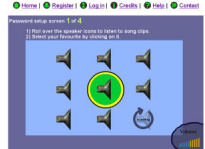
- Marc Conrad
- Tim French
 - Department of Computer Science and Technology, University of Bedfordshire, UK.




Marc Conrad
University of Bedfordshire
<http://perisc.com/mousemusic>

2 **History of the System**

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

- 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.

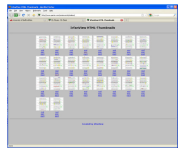
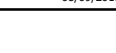


Marc Conrad
University of Bedfordshire
<http://perisc.com/mousemusic>

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.

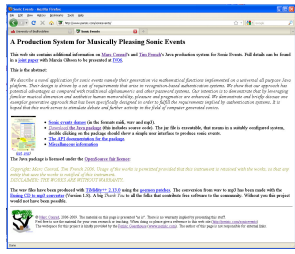
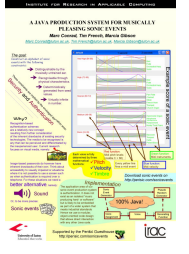

Pre-factored examples (mp3, wav) at:
<http://www.perisc.com/sonicevents/mdemo>

Marc Conrad
University of Bedfordshire
<http://perisc.com/mousemusic>

4 **2006: Production System for Sonic Events**

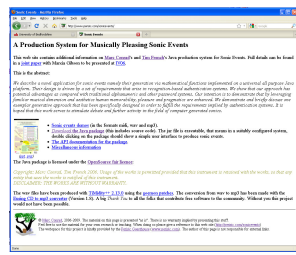
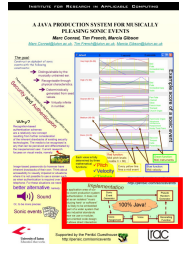
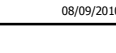
- <http://perisc.com/sonicevents>

Marc Conrad
University of Bedfordshire
<http://perisc.com/mousemusic>

5 **2006: Production System for Sonic Events**

- <http://perisc.com/sonicevents>

Marc Conrad
University of Bedfordshire
<http://perisc.com/mousemusic>

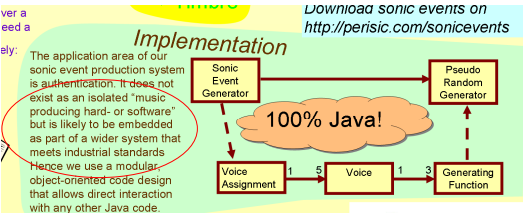

6 **Extensibility as a Key Factor**

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

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

Implementation

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.

Marc Conrad
University of Bedfordshire
<http://perisc.com/mousemusic>

7 Sonic Events Driven by Mathematics

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

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

Implementation

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.

100% Java!

Marc Conrad
University of Bedfordshire
<http://perisic.com/mousemusic>

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)

Marc Conrad
University of Bedfordshire
<http://perisic.com/mousemusic>

9 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?

Marc Conrad
University of Bedfordshire
<http://perisic.com/mousemusic>

10 Transitions between Functions using Weighted Averages

- Picture shows weighted averages of $f_0 = \text{Series1}$ and $f_1 = \text{Series6}$
- Series2 = $0.8 * f_0 + 0.2 * f_1$
- Series3 = $0.6 * f_0 + 0.4 * f_1$
- Etc.

Marc Conrad
University of Bedfordshire
<http://perisic.com/mousemusic>

11 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"

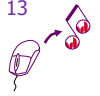
Marc Conrad
University of Bedfordshire
<http://perisic.com/mousemusic>

12 The Screen as a Coordinate System

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

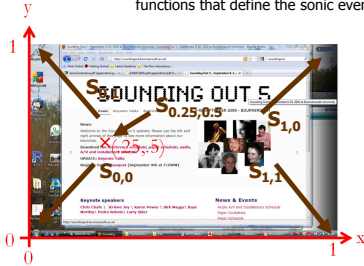
Marc Conrad
University of Bedfordshire
<http://perisic.com/mousemusic>

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 $S_{0.25,0.5}$



08/09/2010 11:49:13

Marc Conrad
University of Bedfordshire

<http://perisic.com/mousemusic>

14



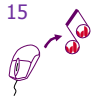
Different 'backgrounds' act as *player stimuli* leading to aesthetic ways of exploring the resultant 'sound space'

08/09/2010 11:49:13

Marc Conrad
University of Bedfordshire

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

15



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?

08/09/2010 11:49:13

Marc Conrad
University of Bedfordshire

<http://perisic.com/mousemusic>