Kevin Nolan MMT thesis outline

Kevin Nolan

1 Introduction

Purpose: introduce the topics under discussion, summarise work completed give thesis structure

1.1 Gentle introduction to the topics under discussion

1.2 Summarise the work undertook

1.3 Motivation

1.4 Thesis structure

2 Background to study

Purpose: establish the dominant analog DAW metaphors and broad approaches and tools in use.

2.1 Introduction

- Introduce what will be discussed in this section
- A brief historical perspective

2.2 The idea generation stage of music production

Introduction Give an overview of different stages of music production as outlined in (Duignan 2008) and end with focus on the initial idea generation

- Using an instrument
- Scoring software
- Quirky music tools such as "Loopy"
- Most commonly in electronic music, DAWS

2.3 Discussion of DAW Metaphors

1. Introduction
2. Piano roll
3. Mixing console
4. Plugins

2.4 Discussion of more open systems

1. Introduction
2. Textual programming systems
   - Supercollider
   - Csound
   - Commonmusic
   - Nyquist
   - Chuck
3. Graphical programming systems
   - Max MSP
   - Pure data
   - Ircam Open music
   - Plogue Bidule
   - Jeskola Buzz

2.5 Music in the browser: a new frontier

Discuss drum machines, etc available on the web. Note the accessibility that they provide. Perhaps introduce tone.js here.

2.6 Conclusion

While the dominant metaphors used in DAWs have their uses they can lead to limitations in the creative process particularly at the early stage of ideas creation. More open system give too much power and impede the creative process.

3 Similar work

Purpose: trace the alternative history of musical systems (visual music, graphic synthesis)

3.1 Introduction

3.2 Sonic sketching from a historical perspective

1. Visual music
   (a) Oskar Fischinger
   (b) Fantasia
2. Legacy synth systems
   (a) Oramics
   (b) UPIC
   (c) ANS synth

3.3 Sonic sketching in the twenty first century

1. Golan Levin (Levin 2000)
2. Music animation machine (Malinowski 2017)
3. SonicPainter by William Coleman (Coleman 2015)
4. Fischinger google doodle

3.4 Alternative music systems on the web

1. Pink trombone - https://dood.al/pinktrombone/
2. Tonehack
3. Auraglyph (although not web based)
4. http://arthurcarabott.com/mui-envelope/

3.5 Summary of currently available music creation systems

(Note that UPIC style sketch synths are not available online)

3.6 Conclusion

Both the twentieth and twenty first century have seen a great deal of experimentation with ideas of visualizing music and sketching. The audio processing capabilities available in modern browsers offers an opportunity to explore and refine less mainstream music creation metaphors.

4 My approach

Purpose: describe how I'm going about it and why. Describe the tools I'm using.

4.1 Introduction

4.2 Appraisal of options

- Availability
- Usage style - instrument like (Levin)

4.3 Approach - theory

1. HCI considerations, in particular NUI (Wigdor and Wixon 2011)
2. The Musical Interface Technology Design Space (Overholt 2009)
3. Research into cross modal perception (time x axis, pitch y axis) - https://www.researchgate.net/publication/280777718_Shape_drawing_and_gesture_Cross-modal_mappings_of_sound_and_music

4.4 Approach - practice

1. Introduction
2. Delivery on Web Browser
   (a) Modern Web Browser as a delivery platform :: discuss pros and cons and situations where it is likely to be a good option. I.e. prototyping where feedback is important. Disadvantages performance, can't be used with pro audio software such as ASIO. (Adenot 2017)
   (b) Benefits of using Tone.js (Mann 2015)
   (c) Paper.js for the graphics system
      - Scenegraph
      - Line smoothing
      - Vector system
3. FM synthesis :: Give a brief overview of FM synthesis and why it was a good choice for the application
4. Live coding workflow
   (a) Introduction
      - The morphic interface
      - Mention precedents such as smalltalk squeek
   (b) React.js framework to allow for a declaritive programming model as well as a live code reloading workflow
   (c) Clojurescript
      i. Relationship to clojure
      ii. Benefits of using clojurescript
         A. Immutable data structures (Binary tree)
         B. Functional programming paradigm
         C. Live code reloading (particularly when used in conjunction with react.js)
   (d) Managing state with Re-frame

- Describe programming model
- It's relationship to FRP

4.5 Conclusion

5 Execution

Purpose: give a detailed account of the build of the project.

5.1 Introduction

5.2 Early prototype work

1. Melodypainter
2. SonicSketch shape version
3. Porting William Coleman's SonicPainter

5.3 Actual implementation

5.3.1 Setting up the architecture

1. Clojurescript and javascript npm modules
2. Paper.js and react.js (paper.js bindings)
3. Tone.js and react.js
4. Reagent and react.js paper.js bindings

5.3.2 Core functionality - timeline events (or notes)

1. Introduction
   - Describe the core functionality
   - Describe core entities
2. Add timeline event
   - Business logic
   - UI
   - Audio
3. Add vibrato
   - Business logic
   - UI
   - Audio
4. Remove note
5. Move note
6. Change sound (preset system)

7. Probability

5.3.3 Secondary functionality

1. Introduction
2. Transport controls
3. Animation (current play position & notes)
4. Undo and redo
5. Fullscreen
6. Outer UI
7. Save and load file

5.3.4 Performance issues

5.4 Conclusion

- Summarise the resulting artifact

6 Evaluation

6.1 Introduction

6.2 Initial pilot test

Introduction Describe methodology used

- Describe results
- Describe comments and feedback
- Verified that it was approachable and basically worked as a NUI application

6.3 Exhibition

6.4 Conclusion

7 Conclusion and further work

7.1 Summary of work completed

7.2 Broader implications of development

- Incorporate ideas into DAWs (move away from tracks and mixers)

- Methodology used very successful for prototyping (delivery on web platform, live code reloading)

7.3 Future work

7.3.1 Performance improvements

- Try different audio engines, in particular wasm based
- Move graphics to GPU based system

7.3.2 Broaden visual language

- Incorporate and visualize more control data

7.3.3 Allow for larger structures

- Perhaps by scrolling
- Or multiple canvases/scenes
- "Smart brushes" that would draw arpeggios, or similar generative structures

7.3.4 3D spaces, VR, spatial audio

References

Adenot, Paul. 2017. "Web Audio API Performance and Debugging Notes." https://padenot.github.io/web-audio-perf/.

Coleman, William. 2015. *sonicPainter: Modifications to the Computer Music Sequencer Inspired by Legacy Composition Systems and Visual Art (MMT Thesis).*

Duignan, Matthew. 2008. "Computer Mediated Music Production: A Study of Abstraction and Activity."

Levin, Golan. 2000. "Painterly Interfaces for Audiovisual Performance." PhD thesis, Massachusetts Institute of Technology.

Malinowski, Stephen. 2017. "Music Animation Machine." http://www.musanim.com/.

Mann, Yotam. 2015. "Interactive Music with Tone. Js." In *Proceedings of the 1st Annual Web Audio Conference.*

Overholt, Dan. 2009. "The Musical Interface Technology Design Space." *Organised Sound* 14 (02): 217. doi:10.1017/S1355771809000326.

Wigdor, Daniel, and Dennis Wixon. 2011. *Brave NUI World: Designing Natural*

User Interfaces for Touch and Gesture. Burlington, Mass: Morgan Kaufmann.