Granny Synth

Adrian Ludwig

1st author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

adrian.ludwig@hs-augsburg.de

Codrin Podoleanu

2rd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

codrin.podoleanu@hs-augsburg.de

**ABSTRACT**

Kurze Zusammenfassung worum es geht und welche Ergebnisse erzielt wurden. Das Abstract schreibt man am besten als Letztes, wenn man den Inhalt des Papiers aufgeschrieben hat.

**Categories and Subject Descriptors**

H.5.2 [**User Interfaces**]*.* This is just an example, please use the correct category and subject descriptors for your submission*.* The ACM Computing Classification Scheme: [http://www.acm.org/about/class/ccs98-html](http://www.acm.org/class/1998/)

**General Terms**

Your general terms must be any of the following 16 designated terms: Algorithms, Management, Measurement, Documentation, Performance, Design, Economics, Reliability, Experimentation, Security, Human Factors, Standardization, Languages, Theory, Legal Aspects, Verification.

**Keywords**

Keywords are your own designated keywords. Z.B. “neue Interaktionstechnologien”

1. MOTIVATION

Problemstellung: Granulare Synthese

Bei der Erstellung elektronischer Musik kommt es darauf an stets eigene, neue Klänge zu gestalten. Eine sehr vielseitige, mächtige Möglichkeit Klänge zu gestalten und zu Formen it die sogenannte "Granularsynthese". Vor allem in der Improvisation lassen sich mit dieser Syntheseart stark eigenständige, eindrucksvolle Klänge erzeugen.

Da es sich um eine digitale Syntheseart handelt, gibt es bereits ein Vielzahl an Software-Synthesizern. Diese eignen sich für die Liveperformance nur bedingt, benötigen sie doch immer einen Rechner mit einer entsprechenden DAW. Vor allem in der Außenwahrnehmung durch Publikum entsteht dabei häufig der Eindruck, dass weniger Live auf der Bühne geschieht, als dass vieles vom Rechner abgespielt wird. Diese Wahrnehmung ist verständlich, sieht es manchmal doch tatsächlich so aus, als würde man seine Mails checken, wenn man angestrengt auf einen Bildschirm sieht.

Ziel von Granny Synth ist es, dieses Problem zu beseitigen, indem auf Basis eines Kleinstrechners und geeigneter Interfacegestaltung ein Liveperformance-geeigneter Granularsynthesesystem gestaltet wird.

Dieses Paper beschreibt die Erstellung eines Prototypen des Systems "Granny Synth". Folgende Schritte werden im folgenden durchlaufen:

* Motivation
* Verwandte Arbeiten
* Anwendungszenarion
  + Anwendungsfall allgemein
  + Konkreter Anwendungsfall
* Entwurf des Prototyps
  + Technischer Ansatz
  + Konzeptueller Aufbau des Prototypen
* Implementierung
  + Hard- und Softwareumgebung
  + Kern der Implementierung
  + Umfang des Prototypen
* Evaluation
  + Ziele der Evaluation und Methodik
  + Durchführung der Evaluation
  + Interpretation der Evaluationsergebnisse
* Zusammenfassung
  + Erzielte Ergebnisse
  + Erweiterungsmöglchkeiten

1. Verwandte Arbeiten

Wie in der Motivation beschrieben, gibt es bereits eine Vielzahl von Softwarelösungen für Granulare Synthese. Ein Prominentes Beispiel ist der in Ableton Live (Digital Audio Workstation) verfügbare "Granulator" https://www.monolake.de/technology/granulator.html

Diese Implementierung der Granularsynthese besticht durch eine vielzahl von Parametern und eine Verhältnismäßig einfache Bedienbarkeit. Dazu trägt insbesondere das sinnvoll gestaltete Bedieninterface entscheidend bei.

In der Welt der Hardware finden sich vergleichsweise wenig Implementierungen von brauchbarer Granularsynthese. Am häufigsten noch sind Lösungen im Bereich der Modularen Synthesizersysteme zu finden. Ein prominentes Beispiel hier ist Mutable Instruments "Clouds" Eurorack-Modul. https://mutable-instruments.net/modules/clouds/ Hier besteht das Problem, dass man darauf angewiesen ist, ein entsprechendes Rack mit allen weiteren nötigen Modulen aufzubauen. Das ist aufwändig und vor allem teuer.

Ein System, das aufgrund seines Aufbaus viele Arten der Synthese zulässt, indem es auf die Open Source Software "Pure Data" zurückgreift, ist Critter and Guitaris "Organelle". https://www.critterandguitari.com/organelle Dabei kommt es auf die Community an, die neue Patches auch frei zur Verfügung stellen kann. Dieses System funktioniert sehr gut, ist aber Aufgrund kleiner Auflagen eher teuer.

Aufgrund dieser vorhergehenden Arbeiten soll mit "Granny Synth" eine Lösung für entscheidende Probleme der genannten Beispiele geschaffen werden.

Der Klangerzeugungsaufbau und die Modulationsparamter des Granulator dienen als Grundlage zur Gestaltung der Softwareseite.

Als Basis für das System soll - wie bei Critter and Guitaris "Organelle" ein Raspberry Pi und Pure Data dienen. Im Gegensatz zur Organelle setzen wir explizit auf granulare Synthese, sowie auf niedrigere Kosten.

1. ANWENDUNGSSZENARIO & Use Case
   1. <mein Szenario>

Beschreiben Sie hier kurz, welches Anwendungssezario Sie vor Augen haben (gerne auch mit Skizze / Bild) Beispiele:

- Entwicklung einer Fernsteuerung mit der WiiMote. Nutzer sollen mittels Gesten Küchengeräte / Einrichtungsgegenstände ... bedienen

* 1. Definition der Use Cases

Beschreiben Sie hier die USE CASES, auf die Sie sich in der Studie konzentrieren. Zur Beschreibung eines USE CASES gehört:

* Die mit dem USE CASE verfolgte Zielsetzung (wozu dient er)
* die Angabe der beteiligten Akteure (d.h., ihre Rollen)
* die genaue Angabe der Bedienabläufe
* die Angabe relevanter Kontextbedingungen, die Sie voraussetzen, z.B. kein Hintergrundlärm, konstante Beleuchtung, etc.

1. ENTWURF EINES PROTOTYPS
   1. Technischer Ansatz

Beschreiben Sie hier kurz, worum es sich bei der von Ihnen eingesetzten Technologie handelt, z.B. welche physikalischen Effekte ausgenutzt werden (vgl. Foliensatz Sensoren) oder welche Algorithmen und Softwaretechniken zum Einsatz kommen.

* 1. Konzeptueller Aufbau des Prototypen

Beschreiben Sie hier aus welchen Systemkomponenten Ihr Prototyp besteht, welche Schnittstellen Sie zwischen den Komponenten festlegen. Evtl. ist es sinnvoll, weitere Unterpunkte zu bilden, etwa zur Beschreibung eines „Servers“ und eines „Clients“ oder zwischen „Sensorik“ und „Steuerung“ usw., je nachdem, welche Komponenten Ihr Konzept umfasst.

* + 1. Server

….

* + 1. Client

….

Beachten Sie, in diesem Abschnitt sollten Sie weitgehend von Implementierungsdetails abstrahieren. Diese bündeln Sie in einem separaten Abschnitt

1. IMPLEMENTIERUNG
   1. Hard- und Softwareumgebung

Beschreiben Sie hier kurz Ihre Entwicklungsumgebung (Hardware, Betriebssystem, Programmiersprachen / Bibliotheken, eingesetzte Tools etc.). Evtl. können Sie hier auch noch auf die Entwicklungsmethode eingehen, etwa, wenn Sie verschiedene Testaufbauten gemacht haben, die letztlich zum finalen Prototypen geführt haben.

* 1. Kern der Implementierung

Beschreiben Sie hier kurz den interessantesten Teil (= Herzstück) der Implementierung. Z.B. könnte hier ein eingesetzter Algorithmus in Pseudocode stehen.

Wenn Sie Programmiercode anführen wollen, bitte nur in Auszügen mit Erklärung. Ausführliche Code-Listing gehören in den Anhang bzw. auf CD.

* 1. Umfang des Prototypen

Fassen Sie hier kurz zusammen, was Sie vom Konzept implementiert haben und wozu Sie nicht mehr gekommen sind. Hier kann auch gerne ein Screenshot oder ein Foto des Prototyps im Einsatz gezeigt werden.

1. EVALUATION

Zu einer wissenschaftlichen Arbeit gehört stets eine Bewertung der erzielten Ergebnisse. Zunächst sollten Sie klarstellen, welche Ziele Sie mit der Evaluation verfolgen und welche Methodik Sie dazu verwenden möchten.

* 1. Ziele der Evaluation und Methodik

Je nachdem, worauf Ihre Studie abzielt, können Sie z.B. daran interessiert sein, ob Ihre Technik überhaupt nutzbar ist (d.h., kommen Nutzer damit zurecht oder kommt es häufig zu Fehlbedienungen). Das finden Sie z.B. heraus, wenn Sie Testpersonen mit Ihrem System arbeiten lassen.

Wollen Sie hingegen nachweisen, dass Ihr System besser ist, als ein bereits existierendes, so können Sie einen Systemvergleich anstellen.

* 1. Durchführung der Evaluation

Hier beschreiben Sie, wie Sie Ihr System evaluiert haben. Falls Sie eine Datenerhebung (Logging / Fragebogen / Interview) gemacht haben, geben Sie an, was dabei herauskam, z.B. die Anzahl der von einer Testperson gemachten Bedienfehler. Haben Sie mehrere Testpersonen involviert, können Sie auch eine statistische Auswertung durchführen.

Wichtig: hier geht es um die Angabe der objektiven Befunde, Spekulationen, warum ein Befund so oder so ausgefallen ist, haben hier nichts verloren!

* 1. Interpretation der Evaluationsergebnisse

Hier können Sie ggf. darüber spekulieren, wie die in der Evaluation gefunden Befunde zu erklären sind und durch welche Änderungen sich ggf. Optimierungen erzielen lassen könnten.

1. ZUSAMMENFASSUNG

Hier fassen Sie nochmal kurz zusammen, was Sie gemacht haben und welche Erkenntnisse aus Ihrer Arbeit hervorgegangen sind. Es bietet sich eine Gliederung an in die Teilabschnitte „Erzielte Ergebnisse“ und. „Ausblick“.

* 1. Erzielte Ergebnisse

z.B.

„Die vorliegende Studie beschäftigte sich mit der Fragestellung, inwiefern sich die „<X>-Technologie für die <Y>-Anwendung nutzen lässt. Dazu wurden zunächst für die <Y>-Anwendung wichtige Use Cases definiert, die dann der Entwicklung eines einfachen Prototypen zugrunde gelegt wurden. Der Prototyp besteht im Wesentlichen aus den Komponenten A B und C. Komponente B nutzt dabei eine leicht modifizierte Version der X-Technologie. Die Modifikation besteht aus der Erweiterung von X zu XX.“

In einer anschließenden Evaluation wurde der Prototyp hinsichtlich der Kriterien K1 bis Kk untersucht. Als Evaluationsmethode wurde eine Nutzerstudie mit n Testnutzern durchgeführt. Die Studie ergab, dass die erweiterte X-Technik für den Einsatz in einer Y-Anwendung zwar prinzipiell geeignet ist, jedoch noch weiteres Finetuning erforderlich ist.

* 1. Erweiterungsmöglichkeiten / Ausblick

Hier beschreiben Sie kurz, welche Verbesserungs- und Erweiterungsmöglichkeiten sinnvoll wären. Sofern Sie schon eine Idee zur Realisierung haben, können Sie diese in einem Satz skizzieren.

Da es in der Regel noch sehr viel zu tun gibt, müssen Sie eine Auswahl treffen! Konzentrieren Sie sich auf die unbedingt durchzuführenden bzw. interessantesten Erweiterungen.

DANKSAGUNG

Hier können Sie Menschen danken, die Sie bei der vorliegenden Studienarbeit unterstützt haben (z.B. auch Sponsoren).

REFERENZEN

1. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. ACM Trans. Program. Lang. Syst. 15, 5 (Nov. 1993), 795-825. DOI= <http://doi.acm.org/10.1145/161468.161471>.
2. Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.
3. Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM Press, New York, NY, 526-531. DOI= <http://doi.acm.org/10.1145/332040.332491>
4. Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.
5. Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
6. Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.
7. Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM Press, New York, NY, 1-10. DOI= <http://doi.acm.org/10.1145/964696.964697>
8. Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.
9. Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender, Ed. Acm Press Frontier Series. ACM Press, New York, NY, 19-33. DOI= <http://doi.acm.org/10.1145/90417.90738>

Columns on Last Page Should Be Made As Close As Possible to Equal Length