KARLSTAD UNIVERSITET

beatr

Real time, interactive Android app for creation and composition of electronic music.

Kristian Sällberg

8/3/2012

This document contains technical and information about the beatr project.

Index

INTRODUCTION ..... ERROR! BOOKMARK NOT DEFINED.
BACKGROUND ..... ERROR! BOOKMARK NOT DEFINED.
OVERVIEW ..... ERROR! BOOKMARK NOT DEFINED.
DETAILED DESCRIPTION ..... ERROR! BOOKMARK NOT DEFINED.
CONCLUSIONS/RESULT **4**
LIST OF REFERENCES ..... ERROR! BOOKMARK NOT DEFINED.
ATTACHMENTS **6**

Introduction

I have been interested in creating some kind of program for music generation for a long time. My personal background is as a professional Actionscript developer, so I am used to program user interfaces. Then I have studied java at school in many courses. I had not touched sound programming at all, so I realized early on that I would not have time to create the actual sound generating code myself. I needed to find a software synthesizer that would already work on the platform of choice.

For a long time, Google has done nothing at all to satisfy the developers community's desire to gain control of the sound card though the SDK. In 2010, Google embedded a library called SoniVox Jetcreator to make it easier to control and schedule sound interactively. With that API developers can schedule what midi files or slices of midifiles to play and what to play after that. And it's possible to change what midi file that plays interactively. But it has never been possible to feed sound data that you have generated yourself to the soundcard through the SDK. When writing this (2012) Google has just launched SDK version 4.1, ???????????????? har synthkod?????????. But anyway SDK version 4.1 will never be available for older/weaker devices and even if it is made available by hackers, most people will likely not run SDK version 4.1 in the coming year.

There is a possibility to write data to a stream that sends it to the sound card. Using the NDK (Native Development Kit), a few of the existing software synthesizers have ported their C libraries to work with the Android platform. They have successfully managed to connect a java interface to the C synth code.

These solutions are not easy to find, and when you find them, they are hard to compile and get running within the Android framework. It took me a good week of searching through music forums and even synth forums before finding a couple of Android ports of existing software synths that did not lag too much or that weren't too awkward to compile.

The combination of Android beeing an awkward platform to develop sound apps for (and it being reasonably difficult for developers to find the ported synths) has scared away most developers from this segment. Many sound developers have chosen to make their apps in the IOS platform instead. Even major players like Propellerhead (creator of Reason, Tractor etc) have chosen not to make an Android version of their IOS hit Propellerhead Figure. Their CEO stating the screen size fragmentation of the android ecosystem as a major obstacle for creating a good sound apps. ?????????????? KÄLLA ?????????????

Who are then left in the segment? There are a couple of developers who made the move and created interactive sound generation apps for the Android platform. I have been in touch with the creator of the most popular Android music generation app, Caustic. Caustic does not officially support the low end Android devices because the app is too advanced for Processors like the AMV6 ???????????? Caustic uses it's own synth code which is specifically tailored and optimized for the Android platform. I've also been in touch with a computer science professor who successfully compiled fluid synth to Android. He has not yet published an app based on the synth to the market.

I did not know about any of this limitations when I picked my project for this course. If I had – I would probably have chosen to do something else, or at least to do it in another platform.

So, all of the problems stated above combined, leave the market for real time, software synth backed, sound generating apps for low end devices pretty much empty. And that's what I wanted to change with beatr.

Background

I have wanted to some kind of program that generates sound, beats and music for many years. A couple of years ago I wanted to learn C++ and openFrameworks. During this period I drew a lot of controllers and instruments that I wanted to create. For instance I wanted a drum machine that could sequence beats and play them

Jag har länge velat göra någon form av applikation för musikgenerering. Jag har vänner som använder program för musikskapande på sina datorer. Det finns även enklare applikationer till de mobila plattformerna, främst IOS. När jag blev antagen till denna kursen såg jag möjligheten att äntligen få göra detta och samtidigt få betalt för det. Först hade jag tänkt skriva applikationen i C++ med hjälp av openFrameworks.

Overview

Applikationen är tänkt att fungera som en samling instrument som spelar samtidigt. Musiken man skriver ska helt och hållet spelas upp i realtid. Jag hittade efter ungefär en veckas letande ett bibleotek som är en software synthesizer, den kan alltså spela upp ljud direkt på telefonens ljudkort enligt data som skickas till bibleoteket. Instrumenten är förinställda att fungera på vissa sätt, t.ex. vill jag ha med en trummaskin, en melodisk synt och någon slags bassynt. Användaren väljer själv hur många instrument som ska användas, och kan dessutom gå in i ett edit-läge i varje enskild instrument och där välja i vilken takt och vilka toner instrumentet ska spela. Instrumenten användaren har valt att inkludera syns i en lista, i den listan kan man justera volymen på instrumentet. Det är så användaren ska kunna styra vilka instrument som låter när, det finns alltså ingen timeline där man anger exakt när något ska spela. Detta tycker jag är ett kul sätt att skilja appen från andra mer seriösa musikredigeringsprogram. Det gör delvis att appen blir mer fokuserad på att spela live, och gör att appen blir mer som ett interaktivt spel där användaren måste lyckas med kompositionen i realtid. [Se beatr/docs/design/beatr.png]

Detailed Description

Applikationen använder sig för nuvarande av ett mycket enkelt ramverk för MVC-modellen.

aMVC:

En vän har sammanställd detta MVC-ramverk som jag använder i min applikation. Det sätter controllern i centrum, controllern skapar views och models.

Views fungerar på så sätt att de är en samling som kan ta upp många andra nativa android-views i en aMVC-View. Detta för att man ibland i android behöver kunna nestla views i varandra för att uppnå vissa utseenden, och det skulle vara mycket opraktiskt att behöva ha en View för varje Layout i appen t.ex.

Model håller som vanligt på data som presenteras i vyn.

Pure Data:
Pure Data är en software synthesizer som finns tillgänglig i många olika plattformar. Den finns även portad till androidplattformen och det är alltså denna jag kommer använda. Man kan skapa instrument med "grafisk programmering" och sedan använda dessa instrument, och ändra parametrar i realtid för att kunna få en interaktiv upplevelse.

Pure data har inte stöd för att kunna ändra namn på objekt från kommandon från synthen, lösningen är att ändra namnen i .pd-filerna innan dom skickas in i pure data.

Conclusions/Result

List of References

Bilagor

Här vill jag t.ex. ha in bilderna på mitt fysiska anteckningsblock för projektet