Jam Sesh: Report

Music to Your Ears, From You


Table of Contents
Overview ..............................................................................................................................2
Prior Work ..........................................................................................................................2
APIs: .................................................................................................................................2
Goals ..................................................................................................................................3
Components .....................................................................................................................3
Flowchart ..........................................................................................................................4
Application: ......................................................................................................................5
Stephen, Ben, Edward ......................................................................................................5
Analysis: ............................................................................................................................6
James, Thomas, Stephen ..................................................................................................6
Synthesis ...........................................................................................................................6
James, Thomas, Edward, Ben ..........................................................................................6
Individual Reports ..........................................................................................................7
Ben ......................................................................................................................................7
Edward .............................................................................................................................7
James ....................................................................................................................................7
Stephen .............................................................................................................................7
Thomas ...............................................................................................................................7
End of the (Time)Line ......................................................................................................8
Summary ...........................................................................................................................8
References .........................................................................................................................9
Overview

The JamSesh computer application seeks to provide its user with real-time synthesised accompaniment to freestyled music. The aim of the project is to bridge the current void between music and technological aid. The end product is an application that will allow users to have their musical ability supported by the algorithms of JamSesh, along with other unique feature enabling more open opportunities to the user such as a variety of synthesised instruments, and the ability to import other music to play along with.

Prior Work

Auto-generated accompaniment programs have been made before, such as “Midi Utility” created by KH Midi Music Ltd. This program differs from our project goal, however, in that it requires the user to input the music as a midi file and select a number of options — the musical element is preloaded, not instantaneous. Our goal is to have a program respond in real time to live music, requiring only a few user options, including instrument selection. Unfortunately, we will not be able to use the prior work done in “Midi Utility,” as its code is not open source. Another important piece of prior work regarding real-time improvisation has been done by Al Biles of RIT; his project, GenJam, is based on a genetic algorithm that can learn to improvise jazz based upon the input of the user. This project is intriguing because it runs in real-time just as we hope to do, but our goal is to do so without a genetic algorithm and instead aim purely for improvisation. Another element of GenJam that is also seen in other works is the ability of the program to play a more melodic line back-and-forth “conversation” with the user. Our aim for this project is to empower the user and instead lean toward a user-centric application.

APIs:

To achieve the final product we are making use of two key API’s, namely TarsosDSP, and TheEchoNest. Tarsos is designed to find the Pitch of a microphone in real time. Our application of this API is a core part of the analysis work, as having the users pitch is vital to synthesising accompaniment for them. TheEchoNest is an API that can analyse a .wav file for key, pitch, tempo and beat, which does not allow for real-time switches for synthesis, but it does allow us to change slightly after-the-fact for large necessary changes on the users part such as a key shift.
Goals
The goals of Jam Sesh are simple to understand but complex to achieve. Overall, the framework of our project centers around processing sound inputted by a user, analyzing its properties, and synthesizing supporting harmonies and background music for the soloist. Some specific parts of this project include: a real-time element, allowing the selection of different instruments, synthesizing complex music structures (i.e. going beyond chord progressions)

Components
We have divided our project into three core subcategories, Application, Analysis, and Synthesis. Application is the overarching machine that interactive with the user and mediates between the other groups. Analysis is for answering the question, what is the user playing right now? Synthesis is for designing the harmony to play back to the user.
Flowchart

This is the flowchart of Jam Sesh. The computer application contains the User Interface and would gather information needed from the User (instruments, speed, main key, etc.). Then, the application will call Analysis. Analysis part uses microphone to gather information needed from the user input and send the information as a float array to Tarsos API (or EchoNest API).
Application:

Stephen, Ben, Edward

Application is responsible for the UI, and interactions between components. The Application will be responsible for finding and defining the globals in the project such as microphone and speakers.

Here is an example UI that we are currently working towards. This is the optimal result as it shows pitch for each instrument, including that of the users instrument. The user has easy and straightforward set up/interactions, with evident tools such as the play and pause buttons, the selection process for instruments. A more indepth look into the file menu enables users to import and export .wav files. At this current point Application is functioning as a generic go between for Analysis and Synthesis.
Analysis:
*James, Thomas, Stephen*

Analysis will then receive pitch information from the API and send interpreted data to the synthesis part. Synthesis will decide the pitch for each instrument to play, export the pitches into a midi file, and send the file back to application, which has a midi player.

Synthesis
*James, Thomas, Edward, Ben*

The synthesis element of our application is receiving the current pitch and successfully produces a triad. This triad consists of the root of the chord, the minor third, and the perfect fifth. This minor chord is formed using the current pitch from analysis as the root of the chord. Additional steps to be taken include transitioning between different kinds of chords and perhaps adding more musically complex pieces (such as suspensions extended chords).
Individual Reports

**Ben**
- Application and Synthesis, Liaison
- Given a hypothetical .wav file, synthesize output
- Researching sound classes
- Grasping prior work
- Spearheading written work

**Edward**
- Application and Synthesis
- Working on Java Applet
- Working on Synthesis solution and algorithm using Music Theory knowledge
- Research on MIDI file format

**James**
- Analysis and Synthesis
- Writing the interfaces and general classes for communication between group classes
- Researching procedural music generation
- Research into the MIDI file type and use in Java
- Working on getting a real time MIDI interface working

**Stephen**
- Application and Analysis
- In charge of general coordination and communication between groups
- Writing the general computer application

**Thomas**
- Analysis and Synthesis
- Pitch detection with Tarsos API
- Graphing live pitch
- Recording .wav file of live input
- Using EchoNest to determine key and tempo
End of the (Time)Line

Progress Report/Demonstration/Presentation 11/19-11/22
Enhancing Analysis 12/2
  ● Bridging the gap between real-time and precision
Synthesizing different tone qualities 12/2
  ● Our main aim is a guitar-like sound; then, a piano, etc (as time permits).
Functional Application UI 12/4
  ● Recording input/output; starting and stopping; saving files
Final Project Submission 12/6
Final Web Page 12/20

Summary

Overall, working on Jam Sesh has been a memorable experience for each group member. Combining the successes achieved along with the adversity we have faced, we have learned quite a bit about the collision of music and technology. Further, this project has shown us how unique an experience it is to work on a long-term project.
References


