Published in *Proceedings of the 1992 International Computer Music Conference*, San Jose, California

FollowPlay:
A MAX Program for Interactive Composition

Todd Winkler
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
Todd@CCRMA.Stanford.Edu

ABSTRACT

FollowPlay is a program for interactive music and a real-time environment for music composition. The program consists of a large, comprehensive collection of MAX objects organized into three functional types: Listener Objects analyze and record numerous aspects of a musician's performance, Composition Objects create MIDI data to send to synthesizers, samplers, or signal processors, and Interpreter Objects unify the entire collection with a score program and an intuitive user interface.

The proliferation of reliable interactive computer music systems has created an opportunity for performers to directly influence computer music processes. *FollowPlay* is a computer program for interactive music and a real-time environment for music composition. It was originally conceived as a way for musicians to communicate their highly refined interpretative skills to a computer, and elicit similarly musical results from it. Through their gestures, performers influence highly flexible compositional algorithms that are subtly responsive to musical nuances. These algorithms create MIDI data used to control synthesizers, samplers, and signal processors.

The program consists of a large, comprehensive collection of software modules, written in the MAX programming language. The modules are organized into three functional types: Listener Objects analyze and record numerous aspects of a musician's performance, Composition Objects create MIDI data to send to synthesizers, samplers, or signal processors, and Interpreter Objects unify the entire collection with an intuitive user interface and a score program that acts as the "brain" of the program, handling all inter-modular communications.

FollowPlay objects are grouped into three types: Listener Objects, Composition Objects, and Interpreter Objects. Listener Objects record and analyze performances from one or more MIDI instruments or acoustic instruments using pitch-to-MIDI converters. Some objects are designed to follow and store a specific aspect of a live performance, such as tempo, articulation, pitch, or dynamics. Other objects analyze musical changes over time, such as crescendo, accelerando, or phrasing. The listener package includes features for improving the incoming MIDI data signal by identifying and rejecting common performance errors, and spurious information from pitch-to-MIDI converters.

Composition Objects are algorithms designed to respond to the performance information gathered by Listener Objects, generating MIDI data that allow performers direct influence over music and timbres played on synthesizers and samplers, as well as control over the processing of their original sound. Multiple Composition Objects, each with a different technique, can be called simultaneously, and orchestrated on as many as 32 different MIDI channels. Composition Objects can create rhythm, specify the number of notes in a chord, influence the intervals and contour of a melodic line, and generate grace notes. Other objects alter synthesizer timbres via system exclusive information or send MIDI continuous controller data for interactive signal processing. Along with algorithmically generated music, MAX can import sequences from a standard MIDI file, and these sequences can be played back anytime during a performance, or piped back into the program. Since many of the variables can be used with absolute or constrained random values, it is possible to determine how much variation will occur from performance to performance. Composition Objects can also be played like musical instruments using the computer keyboard and mouse.

All objects in *FollowPlay* are controlled in some way by the Interpreter Objects. They initialize and alter all variables, setting up conditions to analyze incoming performance information, assigning Composition Objects to receive data from specific Listener Objects, and determining how music will be generated. The two main Interpreter Objects are the Interface Object, which offers on-screen access to all of the program's parameters, and the Score Object, which is a program that automates all the parameters presets and changes that will take place during the course of a piece. The Score Object can change all parameters immediately or slowly over time, as the music warrants, or set up flexible states where parameters respond continuously to performance data. Various strategies are used to advance through each section of a score, such as score following, keyboard triggering, or any condition set up in the performance data modules. All of the parameter changes that are automated with the Score Object are available via number boxes in the Interface Object, a single front panel where variables can be changed by simply typing in a number or by using the mouse as a slider to "click and drag" through the values. Any aspect of the program can be instantly changed while the music is playing.

Other Macintosh programs, running concurrently with *FollowPlay*, create a powerful working environment for composers. Music generated with *FollowPlay* can be recorded and edited with a sequencer, such as *Vision*. From there, it is possible to get a musical score using a notation program, such as *Finale*. The benefit of quickly seeing compositional algorithms as music on a page facilitates the careful fine-tuning of each algorithm, and is enormously helpful in discovering the musical capabilities of *FollowPlay*.

Interpreter Objects provide an intuitive platform for real-time exploration of musical processes. Composers can generate scores for acoustic works, improvise with other musicians, perform solo computer pieces, or shape interactive pieces on-the-spot during rehearsals or performances. This immediacy with generating and manipulating musical materials provides the composer with an interactive laboratory where musical ideas and time-varying compositional processes are quickly realized and refined.

FollowPlay also hold great promise as an interactive tool for teaching students about the fundamental parameters of music, and about compositional techniques. The benefit of getting immediate aural feedback from this kind of experimentation cannot be overemphasized. Although the program is admittedly biased towards processes that reflect a personal approach to composition, it is capable of producing a wide array of musical styles. Since the modular program structure is open-ended, it can expand indefinitely (up to the memory limit) and new objects will be added as problems are solved and processes invented. I look forward to other composers adding to the collection, and contributing their own musical viewpoints.

FollowPlay has already been used successfully for several interactive pieces which suggest a new paradigm for computer music; one where the very fabric of a composition is informed by the subtleties of detail displayed by a master performer (see this conference's concert program notes for a description of the author's *Snake Charmer*). Computer music based on musical gesture imparts a human musical sense to a machine, forming a bridge to the musical traditions of the past, while inviting our most highly esteemed musicians to participate in the process of making music in the computer age.

References

Puckette, Miller, " The Patcher," Proceedings of the 1988 International Computer Music Conference.

Rowe, Robert, "Feature Classification and Related Response in a Real-time Interactive Music System," Proceedings of the 1990 International Computer Music Conference.

Puckette, Miller and David Zicarelli. *MAX - An Interactive Graphic Programming Environment,* Opcode Systems, Inc., Menlo Park,1990.

Winkler, Todd, "Interactive Signal Processing for Acoustic Instruments," Proceedings of the 1991 International Computer Music Conference.