This summer I worked with another student, John Truskowski, Professor Stephen Majercik, and Professor Frank Mauceri to further develop research looking into group improvisation of music using Swarm-PI, an interactive computer system designed to create an experience in which a user can play a conventional instrument to which the computer can react. The computer’s output is not intended to mimic existing musical syntax; instead, this interaction is meant to harness swarm behavior.

The notion of swarm behavior is inspired by the interaction of birds within a flock, which fly together as a loosely connected unit and interact with other flocks. In the same way, streams of sound can be taken as input to create a swarm with which other swarms of sound can interact in specific ways determined by a set of parameters. The parameters specify the behavior of the flocks, which can be visually represented and also used to create output sound that correlate with the flocks’ behavior. For example, the location of a “boid,” the algorithm’s name for the birds within a flock, on the z-axis is used to determine volume.

The goal of the project is to build on the musician’s ability to interact with the computer, which responds to the music being played through the Swarm-PI algorithm. In the past, the work had been done using the language Java and a superclass of Java called Processing. That code was merged with Open Sound Control (OSC) on a software program called Max. With so many different elements, the code functioned but did not do so efficiently.

This summer, we rewrote the code so that it is all within the Max software. In doing so, we were able to clean up the code, reconsider statistics used to determine the interactions of the parameters, and more efficiently write functions, since the Max software was more compatible with the goals of the project.

Alongside the rewriting, we explored the notion of feedback from the human to the computer, asking, how does the computer decide what is “good” to play? We read a lot of papers exploring the notion of quantifying “interestingness” and “beauty” to determine a way to implement an algorithm that receives and reacts to feedback from the human. These weekly conversations were very stimulating and have sparked ideas for future work regarding feedback and memory in the program.

Working so closely with the code allowed me to become familiar with it from many different angles. In doing so, many questions arose, and as a result, there is a lot of room for future work and expansion of this project as a whole. I am very grateful for the opportunity to be here this summer and use my Computer Science major in such an interesting and creative way. I feel inspired by the interdisciplinary nature of this project, and I am excited to continue it upon my return to Bowdoin during the academic year as an independent study.

The software will be demonstrated in the fall at the President’s Science Symposium. The first time we hooked up the sound to the code, we combined sound granule of extinct birds with sound granules of voices speaking extinct languages. The product was really powerful, and it showed the diverse applications of the research.

Faculty Mentor: Stephen Majercik

Funded by the Edward E. Langbein Sr. Summer Research Award