[SQUEAKING] [RUSTLING] [CLICKING]

MICHAEL CUTHBERT: Hello, OpenCourseWare learners. This is Professor Cuthbert talking to you. And I just wanted to say a little bit about why the next two classes are not to be found on this site.

MIT and the music department was very fortunate to be able to hire a new professor in computational music that semester. And one of the things that we value as a department is people's ability to be good teachers. So as part of their interview, the candidates came and took over one of the classes to teach the content of the class and a little bit of their specialties, too. And two of those classes that were used were this class, 21m.383.

Both of the candidates, I can say, did amazing work, and I wish I could share them their videos with you. But we have a feeling that it's already stressful enough to be doing a job interview and to have your teaching being evaluated by people in the room, and then to know that whatever things you did in an unfamiliar classroom with unfamiliar students would also be put out online. It's just a little bit too much. So these-- to ask, too much to ask. So these videos are not available.

I want to say a little bit, though, what was in each of them. The first class was the continuation of our series on music, cognition, and computation, and how computers can aid the process of understanding how the brain processes music, and also, how historically, and I think in the future, insights from music cognition have improved the things that we do on the computer.

The most important example of this is Carol Krumhansl's probe tone experiments in understanding musical expectation in scales and keys. And these experiments, which help to understand how each note plays in our ability to understand what key we're in at this point underlies quite a number of algorithms being used today by computers to automatically detect the keys of compositions. And this includes Music 21's default analyze key method. And this is why the assignments on understanding key analysis went hand in hand with the section on cognition.

The lecture also covered and gave the opportunity to discuss some of the experiments reported by David Huron in his book *Sweet Anticipation,* of which the students were assigned to do readings. I would encourage, as a substitute for the content of this lecture, for you to look deeply at this book by a prominent music cognition and expert, and also one of the breakthrough founders of our field of computational music theory and musicology, David Huron. And doing so, I think, would give a good sense of the majority of what was contained in the lecture beyond what the candidate spoke about her own particular expertise.

The second guest lecturer covered the topic of the history of algorithmic composition from its beginnings, even before the computer, until today. And this lecturer covered the main reading for this week, which is on the "Illiac Suite," one of the first pieces to be composed by computers using algorithms and to generate notes, and continuing up till today with advances in AI, such as the music transformer by Ana Huang at Google and other new developments.

One of the reasons why it's so important to look at algorithmic music composition in a broader historical scope is it's too easy to get hung up on the particular problem that we have today, where we're getting closer and closer to having the computer produce musical scores that sound like other compositions, like by particular composers, and to work on that problem without asking. well, why? Why should we care?

Why do we want the computer to compose a new piece in the style of Johann Sebastian Bach, unless, by chance, you've already heard every piece by Bach, and you're just bored and you need more, which most of us haven't explored what's actually out there?

Instead, by looking back at the earliest history, when having a computer actually write music at that level of sophistication was so far from being possible and looking at what people wanted to do with the computer, it helps to answer questions. How can the computer be used to come up with ideas or take ideas that I have that will take too long for me to generate on my own and make that part more efficient? How can it be a tool to help me be a better human composer, or just better human?

So the class went through some of the earliest algorithmic compositions from Guido of Arezzo, a monk living in the 11th century, through Mozart's musical dice games, up until the "Illiac Suite" and then beyond. Some of what's covered in this class also appears in the various videos that accompany this class that weren't shown to students this year, but are available for you.

And other aspects touched on the guest speakers particular expertise, which was very valuable to be brought in. Anybody else teaching this class would bring in her or his own specialties that would make that class different from mine. And nobody should teach this class exactly the same way I would teach it, or have taught it.

So that gives a little bit of a nutshell of what's happening. The students then went on to write their own algorithms to generate compositions algorithmically, and they did a very good job of avoiding the problem of trying to get the computer to do something that we already do better, that we are not interested really in the computer's output, and that we just would only want to listen to if we knew a human created. And instead, they went beyond that and created a piece of music that were interesting in themselves and that no human would ever choose to write.

In addition to the two classes that job candidates covered, two other classes at around this time needed to be removed and the contents pushed into other classes because MIT was in the process of proposing to the full faculty a new master's program-- two new master's programs-- in music, technology and computation. And it happened that these proposals had to happen at the same time as this class. And so a little bit of time was lost on that. I'm hoping that if you've been interested in this class and want to go forward, you might look into these programs, which I assume will be live at the time that these videos are released.

I think that this is all that I'm going to be able to provide specially for the OpenCourseWare audience in this class, except at the very end, where I have a little message for you that I recorded at the time. Thank you.