MUSC 80L
Artificial Intelligence and Music
2018 Summer Session II
MW 01:00PM – 04:30PM
Music Center 138
Instructor David Kant

Contact
David Kant, email: dkant@ucsc.edu
Office Hours: Tuesday, 03:00PM – 05:00PM,
Digital Arts Research Center (DARC) 340

Overview
Music 80L is an introduction to the fundamental concepts of musical artificial intelligence and generative music systems. The course explores how musicians have engaged with the field of Artificial Intelligence throughout history, and examines the ways music has been codified into formal models and implemented by computers. The course is organized into four units, each focusing on a different approach to Artificial Intelligence and proceeding roughly in chronological order: (1) Expert Systems, (2) Emergent Systems, (3) Listening Systems, and (4) Deep Learning. Topics include Markov chains, rule-based systems, generative models, grammars, transition networks, genetic algorithms, swarm intelligence, machine listening, song generation, automatic radio, neural networks, machine learning, and interactive systems. Running themes include: how motivations, values, and cultural bias are embedded in AI systems; human versus machine agency; and what constitutes Artificial Intelligence.

Prerequisites
Music 80L covers basic introductions to related concepts in mathematics, computer science, linguistics, logic, cognitive science, and machine learning. There are no required prerequisites. Previous experience with computer programming is suggested but not required. Students will learn to use the Python computer programming language for programming music.

Learning Outcomes
As an 80-series Music Department course that satisfies GE designation MF: Mathematics and Formal Reasoning, this course identifies various learning outcomes pertaining to both history and culture as well as mathematics. Students will:

- develop a historical and cultural context for interpreting contemporary activity in the field of artificial intelligence and music
- learn to represent musical concepts within formal systems and computational models
- learn to reason about the behavior of formal systems and computational models
- become familiar with fundamental concepts of computer programming and learn to predict the results of computer code
- develop an understanding of how our prior emotions, beliefs, and values can distort or bias the reasoning and design of formal systems

Textbook
There is no textbook! All of the course readings and listenings are available online through Canvas or as electronic resources from the UCSC library.

Programming
Students will learn basic programming in **Python**. We will use **Google Colaboratory**, a Python environment (based on Jupyter notebook) that runs in the cloud. Notebooks are accessed, stored, and shared via Google Drive and run using a desktop web browser — I recommend Chrome. Access to either a personal computer or school computers is necessary. Prior programming experience is recommended but not required.

Format of the Course
The method of instruction is lecture-centric. Class meetings will include lectures and activities in which we, as a class, collectively design and implement musical artificial intelligences. In addition to readings and listenings, students will complete **tutorials** (via the Google Colaboratory) which teach programming concepts and allow students to interact with the formal systems and models studied.

Work
There will be weekly readings, weekly listening, and weekly quizzes, as well as projects and a final exam.

- **Readings and Listenings**: There will be weekly readings and listenings. The readings and listening list are posted on Canvas with links to electronic resources. Please don't fall behind!

- **Tutorials**: Weekly tutorials are interactive Python notebooks that teach and develop programming concepts. Essentially, these are readings with interactive code. Tutorials are to be completed online via Google Colaboratory. Please read **AND** execute the code in the code cells.

- **Quizzes**: There will be weekly quizzes, **assigned each Wednesday and due the following Monday**. The quizzes are brief multiple choice exercises that reinforce the concepts introduced in class, teach programming concepts, and review the reading and listening assignments. The quizzes are to be completed on Canvas. **Late quizzes will lose 10% for each day late**. I will review the solutions in class.

- **Projects**: There will be three projects throughout the term. The projects are creative exercises that ask you to work through a particular artificial intelligence algorithm. You will be asked to modify code and respond to short answer questions. This is the fun part of the class! Projects are to be submitted through Canvas. **Late project will lose 10% for each day late.**

- **Final Project**: You may either design and build a musical AI or write a paper. I will release the details the second week of class. There will be an option to work in groups.

Grading
Your grade will be determined by weekly quizzes, the three projects, and the final. I will include a grading rubric for each project specifying grading criteria and project expectations. Grade breakdown:

30% – weekly quizzes
40% – three projects
30% – final exam

Final grade scale

A+	98+
A	93–97
A-	90–92
B+	87–89
B	83–86
B-	80–82
C+	77–79
C	73–76
C-	70–72
D+	67–69
D	63–66
D-	60–62
F	0–59

Keeping up to date
Check the course Canvas site regularly! I will update the Canvas syllabus every week with an outline of what was covered in class, including links to readings, listenings, examples, slides, code, and additional resources. I will post class examples, class lecture slides, and additional resources (for supplemental information and tools).

Attendance
Attendance is mandatory! You are responsible for all information covered in class whether or not you are present — this includes both lecture content and course announcements. If you miss class, check in with a classmate, me, or the TAs to get caught up.

Getting Help
If you have a question, chances are one of your classmates does too. Please ask any questions that you are comfortable asking publicly on the Discussions widget of Canvas so that your classmates can see (and maybe even answer!). If you have a question that you would rather not ask publicly, email me or your TAs. I generally respond within 24 – 48 hours during the weekdays (to both email and Canvas Discussions). BUT please do not rely on a quick response the night or two before an assignment deadline — start your work early!

University Policies

Academic Integrity: Academic integrity is the cornerstone of a university education. Academic dishonesty diminishes the university as an institution and all members of the university community. It tarnishes the value of a UCSC degree.

All members of the UCSC community have an explicit responsibility to foster an environment of trust, honesty, fairness, respect, and responsibility. All members of the university community are expected to present as their original work only that which is truly their own. All members of the community are expected to report observed instances of cheating, plagiarism, and other forms of academic dishonesty in order to ensure that the integrity of scholarship is valued and preserved at UCSC.

In the event a student is found in violation of the UCSC Academic Integrity policy, he or she may face both academic sanctions imposed by the instructor of record and disciplinary sanctions imposed either by the provost of his or her college or the Academic Tribunal convened to hear the case. Violations of the Academic Integrity policy can result in dismissal from the university and a permanent notation on a student's transcript.

For the full policy and disciplinary procedures on academic dishonesty, students and instructors should refer to the Academic Integrity page at the Division of Undergraduate Education.

Title IX Policy: Title IX prohibits gender discrimination, including sexual harassment, domestic and dating violence, sexual assault, and stalking. If you have experienced sexual harassment or sexual violence, you can receive confidential support and advocacy at the Campus Advocacy Resources & Education (CARE) Office by calling (831) 502-2273. In addition, Counseling & Psychological Services (CAPS) can provide confidential, counseling support, (831) 459-2628. You can also report gender discrimination directly to the University's Title IX Office, (831) 459-2462. Reports to law enforcement can be made to UCPD, (831) 459-2231 ext. 1. For emergencies call 911.

Faculty and Teaching Assistants are required under the UC Policy on Sexual Violence and Sexual Harassment to inform the Title IX Office should they become aware that you or any other student has experienced sexual violence or sexual harassment.

Disability Accommodations and Resources: UC Santa Cruz is committed to creating an academic environment that supports its diverse student body. If you are a student with a disability who requires accommodations to achieve equal access in this course, please submit your Accommodation Authorization Letter from the Disability Resource Center (DRC) to me privately during my office hours or by appointment, preferably within the first two weeks of the quarter. At this time, I would also like us to discuss ways we can ensure your full participation in the course. I encourage all students who may benefit from learning more about DRC services to contact DRC by phone at 831-459-2089 or by email at drc@ucsc.edu.

Weekly Schedule

Week 1, Intro + Unit 1 — Mon., 7/30 & Wed., 08/1

Course Introduction • syllabus • course overview • a brief history of Artificial Intelligence **Practical 1: Intro to Python for Music** • using Google Colaboratory • Hello World! • variables, arithmetic, lists	**Readings** • Russell and Norvig – AI Textbook CH 1 **Tutorials** • Tutorial 1: Intro to Python for Music **Handouts** • How to Access Colaboratory Notebooks • Listening Week 1
Expert Systems I – Rule Sets • *Illiac Suite* • rule sets • digital representations of pitch and rhythm • symbolic AI "GOFAI" **Practical 2: Coding the Illiac** • random generate and test • conditionals and control flow	**Readings** • Hiller and Isaacson – Composition with a Computer **Tutorials** • Tutorial 2: Coding the Illiac **Assignment** • Quiz 1 due next Mon., August 6th, 1pm • Project 1 due next Wed., August 8th, 1pm

Week 2, Unit 1 — Mon., 8/6 & Wed., 8/8

Expert Systems II – Parameters • *The World's Longest Melody* • *Cybernetic Composer* • generative and parametric models • search and heuristics **Practical 3: Coding TWLM** • functions	**Readings** • Polansky – AMB Documentation • Ames – Cybernetic Composer **Tutorials** • Tutorial 3: Coding TWLM **Handouts** • Listening Week 2 **Deadline** • Drop, Mon., August 6th
Expert Systems III – Transitions • *Experiments in Musical Intelligence* • Markov chains • transition networks	**Readings** • Brooks – An Experiment in Music Composition • Cope – A Computer Model of Music

Practical 4: Markov Chains in Python • objects • parsing MIDI	**Tutorials** • Tutorial 4: Probability and Markov Chains **Deadline** • Grade Option, Friday, August 10th **Assignment** • Quiz 2 due next Mon., August 13th, 1pm • Project 2 due next Wed., August 15th, 1pm

Week 3, Unit 2 Mon., 8/13 & Wed., 8/15

Emergent Systems I – Artificial Life • *Trilobyte* • genetic algorithms • artificial life **Practical 5: Genetic Algorithms in Python**	**Readings** • Levy – The Genetic Algorithm • Corne and Bentley – CES CH1 **Tutorials** • Tutorial 5: Genetic Algorithms **Handouts** • Listening Week 3
Emergent Systems II – The Emergent Mind • *sunSurge Automata* • swarm intelligence • cellular automata • chaos • sync and entrainment • sub-symbolic AI **Practical 6: Coding Swarms and CAs**	**Readings** • Dunn – Thresholds and Fragile States • Scaletti – sunSurge Automata **Tutorials** • Tutorial 6: Coding Swarms and CAs **Handouts** • Listening Week 3 **Deadline** • Withdraw, Friday, August 17th **Assignment** • Quiz 3 due next Mon., August 20th, 1pm • Project 3 due next Wed., August 22nd, 1pm

Week 4, Unit 3 Mon., 8/20 & Wed., 8/22

Listening Systems I – Improvising Machines • *Voyager* • spontaneous and interactive systems • networks	**Readings** • Lewis – Too Many Notes • Erickson – Performing Algorithms **Tutorials**

• algorithmic bias **Practical 7: Making Music Collectively**	• Tutorial 7: Coding Voyager **Handouts** • Listening Week 4
Listening Systems II – The Artificial Ear • *Happy Valley Band* • *Shazam* • machine listening • timbre and perception **Practical 8: Teaching Python to Listen**	**Readings** • Kant – Introducing HVB • Mumma – Hornpipe **Tutorials** • Tutorial 8: Teaching Python to Listen **Assignment** • Quiz 4 due next Mon., August 27th, 1pm • Final due next Mon., September 27th, 1pm

Week 5, Unit 4 Mon., 8/27 & Wed., 8/29

Deep Learning I – The Return of Neural Networks • *Spring Spyre* • contemporary trends in musical AI • artificial neural networks • classification and the perceptron • recurrent neural networks **Practical 9: A Simple Neural Network**	**Readings** • Toivanien – Symbolic vs Connectionism • Karpathy – The Unreasonable Effectiveness of Recurrent Neural Networks **Tutorials** • Tutorial 9: A Simple Neural Network **Handouts** • Listening Week 5
Deep Learning II – Semantic Spaces • *MusicVAE* • autoencoders • latent models as semantic spaces **Practical 10: Navigating Semantic Spaces**	**Readings** • Google Magenta – MusicVAE: Creating a Palette for Scores with Machine Learning • Hu – Reshaping Not Destroying **Tutorials** • Tutorial 10: Navigating Semantic Spaces

This schedule is likely to change! Please <u>check the online Canvas syllabus for weekly updates.</u>

Projects

- Project 1 – Rules Rules Rules, Due Week 2
- Project 2 – Markov Madness, Due Week 3
- Project 3 – Evolved Sound, Due Week 4
- Final Project, Due Week 6

Quizzes

- Quiz 1, Due Week 2
- Quiz 2, Due Week 3
- Quiz 3, Due Week 4
- Quiz 4, Due Week 5

Tutorials

- Tutorial 1: Hello Python
- Tutorial 2: Coding the Illiac
- Tutorial 3: Coding The World's Longest Melody
- Tutorial 4: Probability and Markov Chains
- Tutorial 5: Genetic Algorithms
- Tutorial 6: Coding Swarms and CAs
- Tutorial 7: Coding Voyager
- Tutorial 8: Teaching Python to Listen
- Tutorial 9: A Simple Neural Network
- Tutorial 10: Navigating Semantic Spaces