CREATIVE AUTOMATA CLASS
Dr. Paul Fishwick, TA: Karen Doore
Fishwick (ATC 3.209): Paul.Fishwick@utdallas.edu
Doore: kdoore@utdallas.edu

GOALS:

Introduce new ways of seeing and making foundational concepts in computing. These concepts are listed below in the class schedule. We treat computing from its early days in mathematics (e.g., the act of counting, analog computing and the eventual transition to digital computing). Due to this breadth, the word “automata” is meant be both historical physical automata as well as automata as taught in Computer Science. Knowledge of classical automata in physical form is covered in signals and systems” courses, as well as in courses in continuous simulation, systems modeling and engineering. It is also found in areas relying on signal processing such as computer music. See the research lab blog for general philosophy and information about the research area of Creative Automata: http://www.creative-automata.com

MAKING the ABSTRACT CONCRETE:

With all of the discussion of “making” and “maker fairs”, and the emergence, or perhaps enhanced reemergence of making, Creative Automata as a theory and even a discipline is timely. The class covers various abstract topics beginning with Discrete Mathematics (ref. TOPICS section) and asks “how can we make this abstract topic concrete?” Making it concrete begins with a design phase. For us, that will be actual sketching as well as making interactive Javascript sketches just to flesh
out our concrete design ideas. Then, we can either “go virtual” with a game mod, game engine, or some other form of virtual reality. This is making but making using computer graphic and audio technologies. Or we can make tangible products that involve tactility. The class objective is then to get you to think beyond writing as a means for expressing abstract mathematical and computing concepts, and to instead think making.

**REQUIREMENTS:**

General: Willingness to learn whatever is required on your own, especially for projects beyond Javascript (e.g., interactive sandboxes, game engine or physical materials, including microcontrollers)

Programming: knowledge in at least one computer language - understanding of concepts

Art & Creativity: interest to be creative [art, design, going to art or design museums]

Mathematics: understanding of basic concepts in calculus (derivative, integration)

Physics: understanding of basic concepts

Computer graphics: basic understanding

Interaction: All students will be expected to interact in class, and perform impromptu assignments while in-class. Students
will be called upon randomly.

Cost of materials: You will be expected to pay for required materials as requested (all materials will be available, and all will cost less than $200 for the semester). A drawing sketch book is required (no larger than 8.5x11”).

**GRADING:**

Programming projects that center on creating sketches on paper, followed by prototypes in Javascript, with associated Javascript libraries. Also, other projects that go beyond Sketch level will require a proficiency in some other type of media such as a game engine, 3D graphics engine, or DIY projects involving electronics and physical materials.

Two exams will be given on material covered in the class.

Attendance will be taken at the start of each class and graded, unless you have an excused absence.

Some projects will be organized in teams, with each team composed of a mix of ATEC and CS students.

Expected grade distribution: A or A- (no more than 35% of class).

**SCHEDULE**

Object of the Day [Start of each day]
Each day will begin with an object from real life experience
where the information properties of that objected be elucidated. This will be “Learning to See Information”. We will discuss and interact during this time.

Overview of Creative Automata with Examples
   Core: Observation and Creation phases
   General discussion with online multimedia references
   Summation
   Subtraction
   Lotka Volterra model

The Art of Prototyping [~3 weeks]
   Learning how to use a sketch book [1/2 day]
Elements of Javascript and Select Libraries for e-sketching interaction [2.5 weeks]

Creativity in Practice:
   Drawing on right side of brain [drawing exercises]
Glass bead game, Zwicky’s Box,
Figures of speech, analogy, and metaphor
Mapping analog domains in mechanics (e.g., fluids, solids, position, angle)
The nature of Abstraction: art vs. mathematics

Foundational Concepts 1: Discrete Mathematics
   Number systems and encoding, basic patterns and ciphers
   Set theory and Boolean Logic
   Elements of language: syntax and semantics
   Basic data structures: arrays, trees, and graphs
   Finite State Machines, Push Down and Queue Automata
Foundational Concepts 2: Modeling
Methods of flow: control vs. data
Formulas: analog computing methods
  Flowcharting programs (from Javascript to Flow)
  System Dynamics Method
  Petri networks
  Logic networks
  Information modeling: semantic networks