LIVING COMPUTATION

WEEK 1/17

CS491.2/591.2 Fall 2017
UNM Computer Science

M,W 11:00-12:15 CEC-B146A

Dave Ackley
ackley@cs.unm.edu
Who are we?
Why are we here?
What do we believe in?
→ Notes to wiki at https://robust.cs.unm.edu/
1. **Report in (all, 15 min)**: Who are you? Why are you here? What do you believe in?

2. **Tell (Dave, all, 30 min)**: Introduction and overview. Course goals and ground rules. Discussion.

3. **Break (all, 5 min)**: Read, fill out & sign class form. 78% eclipse?

4. **Show (Dave, 15 min+-)**: Demo introduction to *ulam* and the Movable Feast.

5. **Report out (Dave, 5 min)**: Action items for Wednesday.
Who I Am
What I Offer To You
What I Ask Of You
Mechanics: Living Computation R&D
Theme: E Pluribus Unum
Discussion
Break
Directed individual research in robust living systems strategies applied to computational tasks, organized as a hands-on studio course and emphasizing awareness of the arc of creative processes involved in empirical research. In consultation with the instructor, the student will frame an empirical problem of interest, develop and implement computational strategies for it in Ulam, gather observations and data, and produce oral and written presentations of results, as well as providing constructive critiques of other's efforts.
Course mechanics

• No official heavyweight textbook, but material from the internet will be assigned as appropriate.

• Course work consists of:
  – Learning **ulam** via programming exercises and the project
  – Semester-long individual projects in living computation
  – Project progress presentations and critiques
  – Supporting each other in discussion, questions, and crits
  – Reading and discussion
  – Developing wiki content at robust.cs.unm.edu
  – Miscellaneous other assignments and challenges

• We all evaluate each other at midsemester and at end

• One on one meetings with instructor at midsemester at least
A studio class for computational research

- Research as art; research as science
- Science is something people do
- Coin of the realm
  - Jargon, attitude, rolling the log, results
- What counts as a result?
  - The unexpected consequence from simplicity
  - The 'U-shaped curve'
- The LPU and beyond
- → Living Computation
The two attractors of computing

Correct
Efficient

Robust
Correct

“Master of the Universe”
“Member of the Team”
1. Latin for "from many, one"

2. The relationship between individual and group, writ large, is at the heart of computer architecture, computational design, technology development and use, and society at large.

3. The relationship between individual and group is far too important and far too specific to be 'solved' once and for all (e.g., by the 'von Neumann architecture').

4. Our job is to design and build useful (prototypes of) mechanisms and computations for balancing the needs of the many and the needs of the few.
Key operations:

1. How to decide **who does what** (processor/memory)

2. How to **coordinate the work** (synchronization)

3. How to **exchange requests and information** (communication)

4. How to **innovate and improve the process** (search / adaptation)

5. How to **reward success and prevent or mitigate cheating** (credit assignment)
READ, fill out, and sign the declaration at the back of the opening day handout!
The Movable Feast Machine

'Moving artificial chemistry'
- Asynchronous cellular automata
- Massive neighborhood size
- Indefinitely scalable design

Software

```
module mfa 1;  
import Dockable;  
element Sample(bond prv, bond nxt, sbond dock, sbond tmp) = 0x123  
    
/* Behavior goes here as loop-free code, able to read and write only inside the event window */  
```

Element definitions
(compiled to non-volatile memory)

Hardware

Sites & atoms
(volatile memory)

Event window
(fixed size; may span tiles)

Active atom
(type-specific atomic update+ bond-aware diffusion)

Tiles
(memory+ processing+ local communications)
Demo

2. **Plan:** Think of TWO possible living computation areas that you might want to explore; prepare to discuss them briefly. (No obligation! Sky's the limit!)

3. **Do:** The tutorial introduction at https://github.com/elenasa/ULAM/wiki/Ulam-Programming-Language (Note: You are encouraged, but not required, to set up ulam/mfm on your own machine if you have an appropriate setup. But the CS machines have it available for use now.)