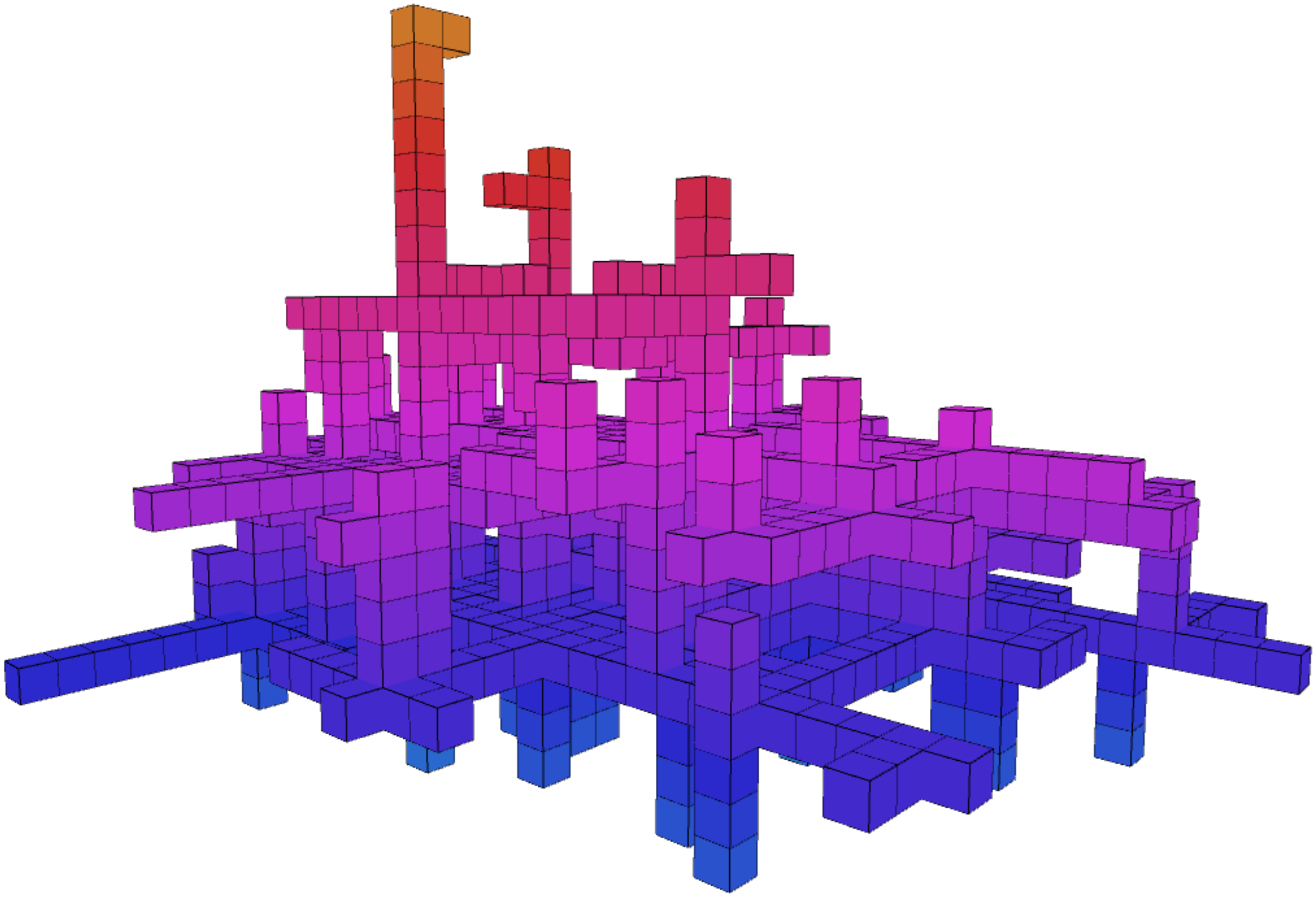
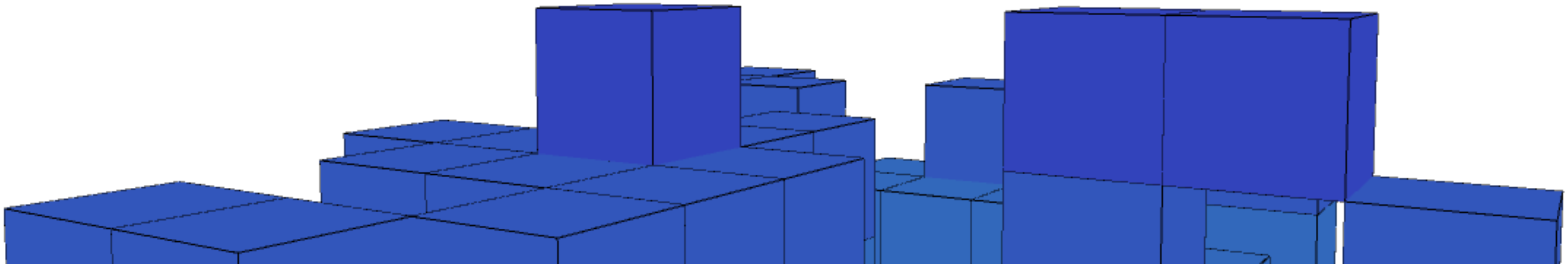
Levity: Building Cubic Playgrounds



Pehr Hovey MAT594CM Spring 09

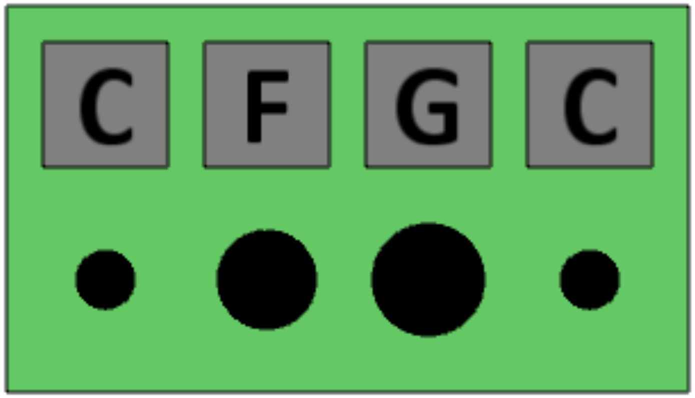
Cubic Origins

- Goal to create a system for creating generative 3D structures with a wide variety of shapes
- Want to specify complex processes with simple 'genes'
- Using a combination of probability and modified L-Systems to calculate geometry

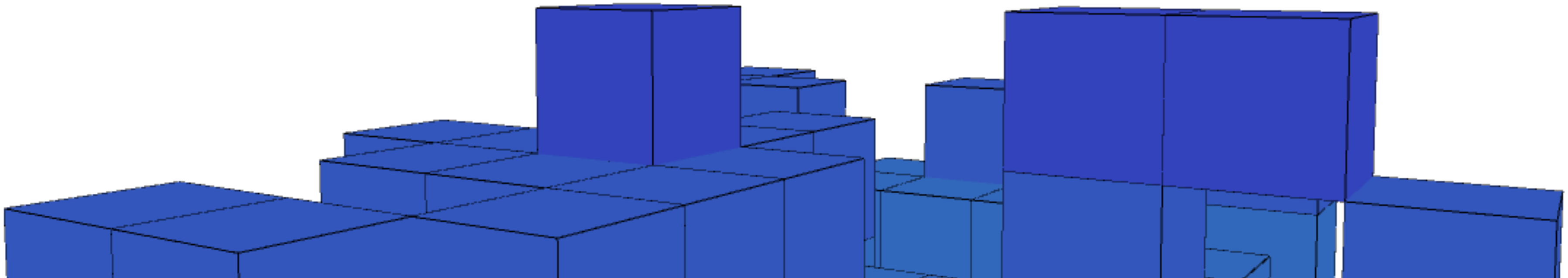


Cubic Genome

- The construct is specified with four genes:

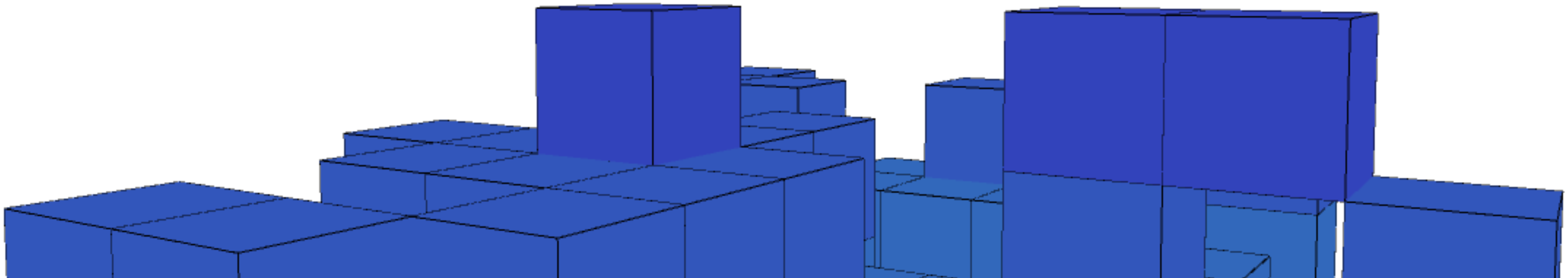


- Each Gene is an L-Code symbol between A-G which each has a float for basic parameter calculations



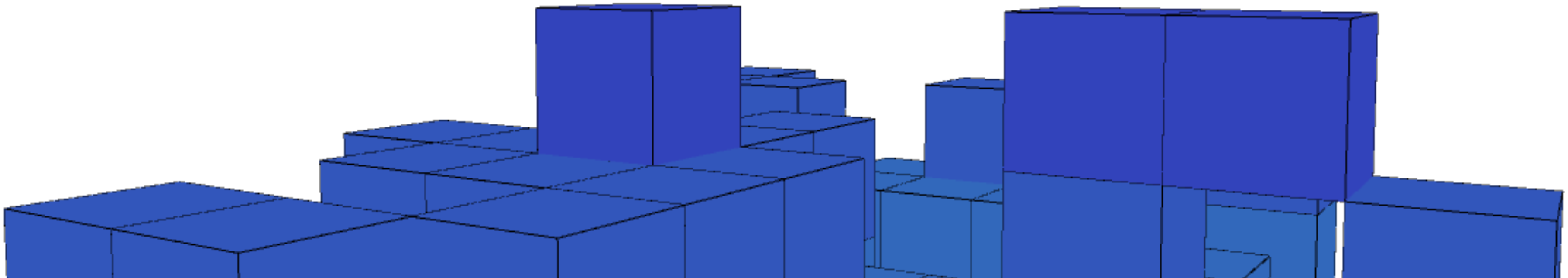
Cubic Genome

- Each slot affects different parameters:
  - L0: Coloration (hue) and spatial anchoring
  - L1: Maximum lateral size and mutation frequency
  - L2: Maximum height and growth speed
  - L3: Core geometric characteristics and initial root factors
- Changing any of the slots will re-generate the figure using the new parameters
- Random number generator is reset to the starting seed to help keep things semi-deterministic



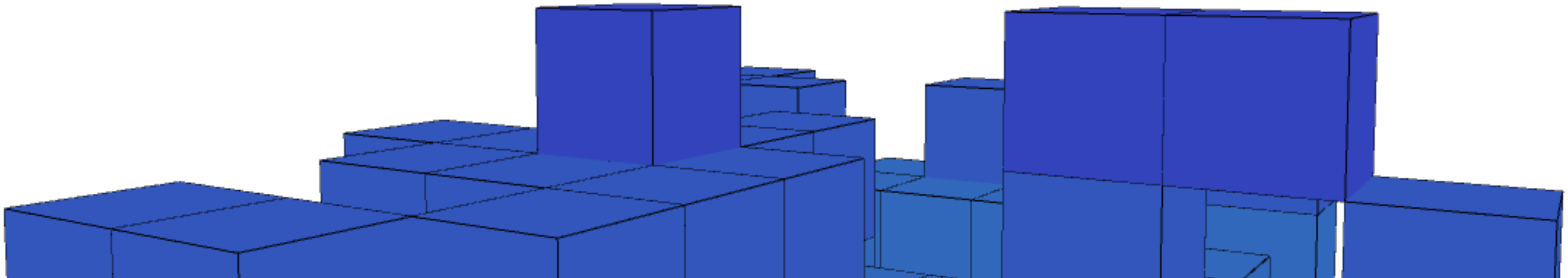
Generative Geometry

- Starting with the principles of L-Systems:
  - A specified set of valid symbols
  - Rules for expanding & re-writing the symbols
  - Actions for translating symbols into geometry
- Different from L-Systems:
  - No distinction between terminal & non-terminal systems
  - Entire L String is generated in advance and selectively applied
  - Probabilistic processes mean the resultant L-String is not completely deterministic and does not completely specify the resultant form



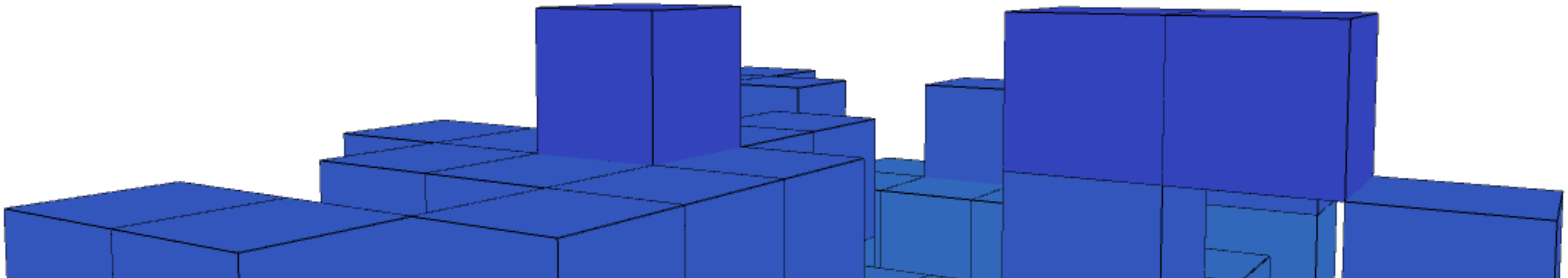
L-Code - > L-Symbols

- Each L-Code symbol (A→ G) has a float value (0.0→1.0)
  - Currently float is a linear function of the L-Code symbol index
- Each L-Code symbol also has a variable length L-String that is available for use in constructing geometry
- L-String for each Code symbol is generated at run-time and depends on the symbol’s Float Value and the state of the random number generator
- The length of the L-String is highly variable depending on the rules applied and the float value of the L-Code symbol



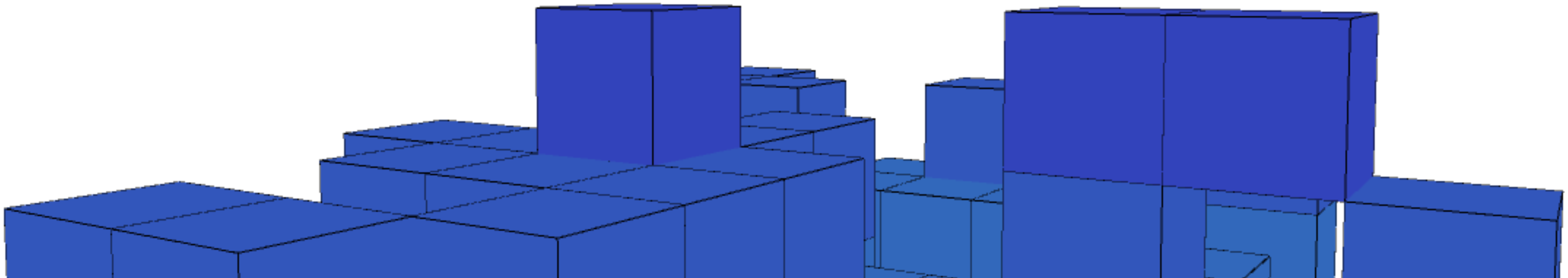
Simple L-Symbols

- Available Symbols:
  - START: The beginning of the form, not present in final string
  - MOVE: Go forward one cubic unit
  - FORK: Create one or more children on available surfaces
  - PAUSE: Do nothing for one iteration
- L-Symbol Rewriting Rules:
  - START → MOVE (Always)
  - MOVE → Combo of up to six FORK and MOVE (depending on F-val)
  - FORK → Either FORK or PAUSE (depending on F-val)
  - PAUSE→ Either MOVE or PAUSE (depending on random gen.)



L-Examples

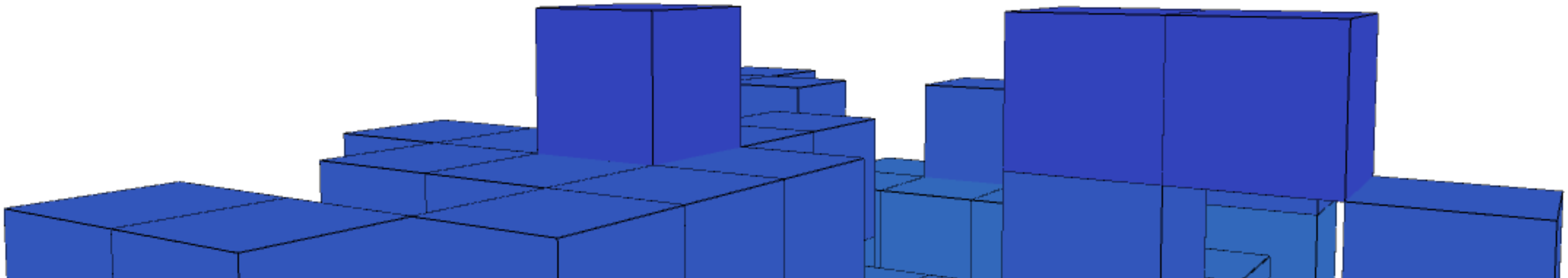
- B(0):->[START]
- B(1):->[MOVE]
- B(2):->[MOVE, FORK, MOVE]
- B(3):->[MOVE, MOVE, MOVE, PAUSE, FORK, MOVE, MOVE]
- B(4):->[MOVE, MOVE, MOVE, FORK, MOVE, FORK, FORK, MOVE, FORK, MOVE, PAUSE, FORK, MOVE, MOVE, FORK, FORK, FORK]
- C(0):->[START]
- C(1):->[MOVE]
- C(2):->[FORK, FORK, MOVE]
- C(3):->[PAUSE, PAUSE, FORK, MOVE, FORK]
- C(4):->[MOVE, PAUSE, FORK, FORK, FORK, MOVE, PAUSE]



Using the L-String

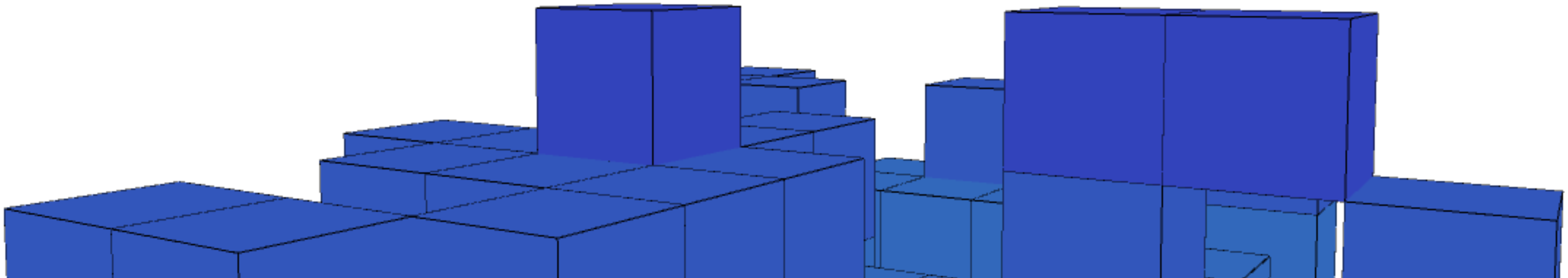
- Overall L-Structure stores the current 4-gene code and the current L-String for that code (based on Gene L3)
- Starting at each root cube pop one symbol and evaluate?

- MOVE: create new cube in forward direction (away from parent), give it the *remaining string* and stop growing.
- FORK: create some children (based on L1) and give them the *complete string* to start anew and keep growing
- PAUSE: based on Gene L2, skip one or more iterations



Challenges

- Tweaking the limits of the L-String generation system to keep things varied but bounded
- Minimizing number of cubes needed (collision detection)
- Keeping everything quantized to the grid for cubic effect
- Making good use of the expressiveness of 4-genes with 7 possibilities (7^4 = 2401 total)
- Keeping things generative and not rigged for aesthetics



L-End!