Competition

**Aim** To define and implement the most creative domain in a 64-bit word

**Genotype** Bit-string of length $b \leq 64$ bits

**Phenotype** Anything

**Key Measures**
- **Typicality:** Individuals represent domain
- **Quality:** Individuals valuable within domain
- **Diversity:** Scope for variation within domain

**Judging Criteria**
- Creativity
- Diversity
- Appeal
- Usage (of all 64 bits)

**Motivation** To help explore search dynamics in constrained creative spaces
Examples

**Biominoes** (64 bits) 8x8 icon design

![Biominoes](image)

**Road Block** (42 bits) Puzzle design

![Road Block](image)

**Harmonograms** (60 bits)
Fourier frequencies for harmonic curves

![Harmonograms](image)

**Long Life** (64 bits) 8x8 Game of Life

![Long Life](image)

**Shibumi** (64 bits) Game board states

![Shibumi](image)

**Loop** (64 bits) Game level design

![Loop](image)
Entries

Competition ran for four weeks prior to CEC (trial run)

Two entries:

1. **Land Bridge in a Word**  
   Thomas Palazzolo, Robert Reynolds, John O'Shea, David Warnke, Areej Salaymeh and Dustin Stanley  
   Wayne State University

2. **Cayley Graph Visualiser**  
   Miguel Nicolau and Dan Costelloe  
   University College Dublin
Entry #1: Land Bridge in a Word

**Stage 1.** User edits 8x8 bitmap (genotype).
Entry #1: Land Bridge in a Word

Stage 2. Bitmap processed to define “land bridge” landscape
Interactive predator-prey simulation

Typicality
Always 1

Quality
Based on play experience(?)

Diversity
Different landscape layouts

Literally creates a world with each word!
Entry #2: Cayley Graph Visualiser

**Genotype** 64 bit word

**Phenotype** Genotype $\rightarrow$ BNF grammar $\rightarrow$ Jenn3d input

Jenn3d

3rd party library

Visualises finite Coxeter groups on 4 elements

Genotypes are visualised as Cayley graphs
Entry #2: Cayley Graph Visualiser

**Typicality**
- 0.0  Not a Cayley graph
- 0.5  Doesn’t converge
- 1.0  Valid example

**Quality**
Based on user preference
User-guided evolution

**Diversity**
Visually diverse results
Winner

And the winner is:

**Cayley Graph Visualiser**
Miguel Nicolau and Dan Costelloe
University College Dublin


Runners up:

**Land Bridge in a Word**
T. Palazzolo, R. Reynolds, J. O'Shea, D. Warnke, A. Salaymeh and D. Stanley
Wayne State University


Thanks to all entrants!
8x8 Game of Life (B3/S23) with edge wraparound

Bitwise parallel update
Entire grid in single pass
82 binary ops (71 optimised) ≈1 op per cell(!)

Algorithm by Stephen Tavener

```java
// Edge cell masks (Left/Top/Right/Bottom)
static final long L = 0x8080808080808080L;
static final long R = 0x0101010101010101L;
static final long T = 0x00000000000000FFL;
static final long B = 0xFF00000000000000L;
static final long NL = ~L;
static final long NR = ~R;

// Bit plane adders
long bit1, bit2, bit3;

// Add neighbour count cXX to bit registers
void add(final long cXX) {
    final long carry1 = bit1 & cXX;
    final long carry2 = bit2 & carry1;

    bit1 ^= cXX;
    bit2 ^= carry1;
    bit3 |= carry2;
}

// Perform one step of 8x8 Life B3/S23 on state c11
long step(final long c11) {
    // Shift neighbors into position, with wraparound
    final long c10 = c11 >>> 8 | ((c11 & T) << 56);
    final long c12 = c11 <<< 8 | ((c11 & B) >>> 56);
    final long c00 = (c10 & NL) << 1 | ((c10 & L) >>> 7);
    final long c01 = (c11 & NL) << 1 | ((c11 & L) >>> 7);
    final long c02 = (c12 & NL) << 1 | ((c12 & L) >>> 7);
    final long c20 = (c10 & NR) >>> 1 | ((c10 & R) << 7);
    final long c21 = (c11 & NR) >>> 1 | ((c11 & R) << 7);
    final long c22 = (c12 & NR) >>> 1 | ((c12 & R) << 7);

    // Reset the bit registers
    bit1 = 0; bit2 = 0; bit3 = 0;

    // Accumulate live neighbor counts
    add(c00); add(c01); add(c02); add(c10);
    add(c12); add(c20); add(c21); add(c22);

    // Return live cases
    return ((c11 | bit1) & bit2 & ~bit3);
}
```