Workshop en simuación basada en agentes

Bogotá, Colombia – 12 y 13 de diciembre

Simulación basada en agentes

Una metodología para la investigación en ciencia e ingeniería

David Anzola Pinzón
Nelson Gómez Cruz
John Vargas
Daniel Rodríguez Cardenas

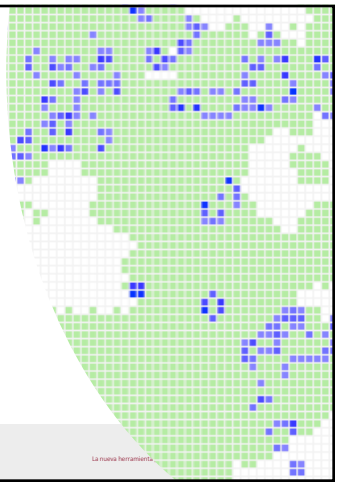
Centro de Innovación - INNOVA
Escuela de Administración
Universidad del Rosario

Laboratorio

Creando modelos basados en *patches*

El juego de la vida

- Motivado por el trabajo de Von Neumann.
- Implementado por Jhon Conway los años 60.
- Contiene un conjunto de reglas muy básicas
- Es un autómata celular de 2 dimensiones.

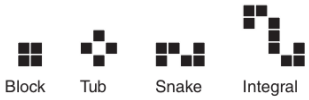


Conjunto de reglas

- **Soledad:** Si una celda está "viva" y tiene menos de dos celdas vivas en su vecindad, entonces muere.
- **Hacinamiento**: Si una celda viva tiene más de tres vecinos vivos, entonces muere.
- **Reproducción:** Si una celda NO viva tiene tres vecinos vivos, entonces vive
- **Estacionario**: En otro caso la celda se mantiene como está

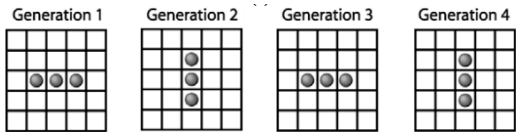
Patrones tipo I - estacionarios

- Patrones estables. Se mantienen sin cambios sin perturbaciones



Patrones tipo II - osciladores

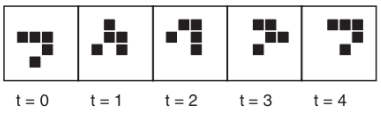
- Patrones que se repiten a sí mismos, luego de una secuencia de estados y retornan a su estado original



Blinker

Patrones tipo III – naves espaciales

- Patrones que se repiten a sí mismos después de una secuencia de estados y retornan a su estado original, pero se trasladan en el espacio.
- Patrones que se mueven a una velocidad constante

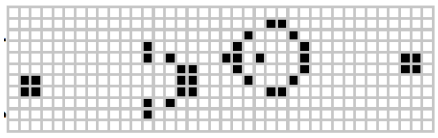


Glider

Patrones tipo IV Armas

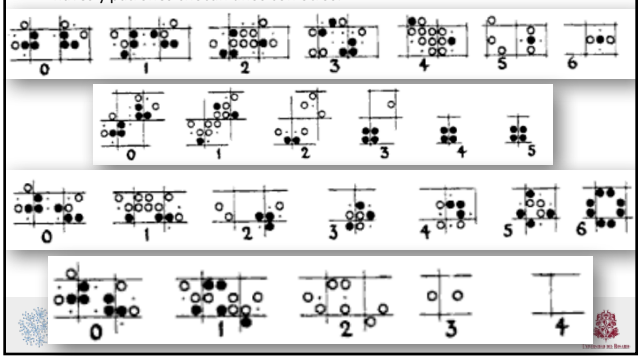
- Osciladores que emiten naves viajeras en cada ciclo

1. **Armas**: un arma que periódicamente dispara fuera de del patrón que lo emite.
2. **Disparadora de naves** (Glider Gun)



Reacciones en las colisiones

Podemos observar diferentes reacciones cuando las naves y patrones chocan unos con otros.



Implementando el modelo (1)

Uso de "*patches*" – (celdas)

- ¿Qué propiedades tiene cada celda?

```
patches-own [live-neighbors]
```

- ¿Cuál es el estado inicial del modelo?

```
to setup
  clear-all
  ask patches [
    ;; create approximately 10% alive patches
    set pcolor blue - 3 ;; dark blue cells are dead
    if random 100 < 10 [
      set pcolor green ;; green cells are alive
    ]
  ]
  reset-ticks
end
```


Implementando el modelo (2)

- ¿Qué debe realizarse en cada paso de la simulación?

```
ask patches [
  ;; each patch counts its number of green neighboring patche
  ;; and stores the value in its live-neighbors variable
  set live-neighbors count neighbors with [ pcolor = green ]
]
```

- ¿Cómo verificar las reglas de interacción y decisión?

```
ask patches [
  ;; patches with 3 green neighbors, turn (or stay) green
  if live-neighbors = 3 [ set pcolor green ]
  ;; patches with 0 or 1 green neighbors turn (or stay) dark blue (from isolation)
  if live-neighbors = 0 or live-neighbors = 1  [ set pcolor blue - 3 ]
  ;; patches with 4 or more green neighbors turn (or stay) dark blue (from overcrowding)
  if live-neighbors >= 4 [ set pcolor blue - 3 ]
  ;; patches with exactly 2 green neighbors keep their color
]
tick
```