

PRÁCTICA 6: NOTAS AUXILIARES PARA SU CORRECTO DESARROLLO E IMPLEMENTACIÓN

La práctica tiene por objetivo implementar un autómata celular bidimensional clásico donde los haya, conocido como el “Juego de la Vida”, propuesto por el matemático John H. Conway, y popularizado por A. K. Dewdney en su columna de “juegos matemáticos”, que publicó mensualmente durante muchos años en la revista *Scientific American* (publicada en España como “Investigación y Ciencia”).

¿Qué diferencia a un autómata celular 2-D de 1-D, además de la dimensión?

Pues dos cosas: la función de transición, y la vecindad; la primera depende de la segunda, y comenzamos por explicar esta última. Ahora, una célula dada no solo tiene vecinos a los lados, sino también arriba y abajo. Clásicamente, esto se tiene en cuenta mediante las vecindades de Moore y Von Neuman, ambas simétricas; pueden plantear alternativas asimétricas si se considera necesario, aunque la mayoría de fenómenos físicos, químicos o biológicos que se modelan con estos autómatas se adaptan mejor a una simulación con vecindad simétrica.

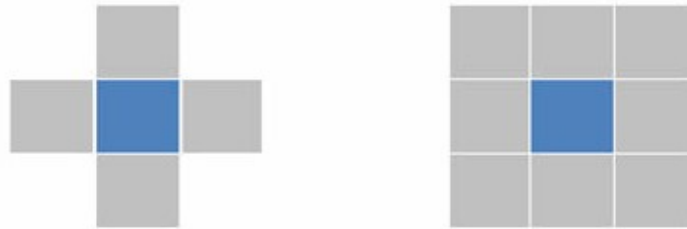


Figura: vecindades de Moore y Von Neumann.

La función de transición ahora tiene en cuenta para calcular el nuevo estado de una célula $a_i^{(t+1)}$ bien a las cuatro vecinas situadas en los puntos cardinales (subimagen izquierda de la Figura) y a ella, bien a las ocho vecinas que la rodean (subimagen derecha de la Figura) y a ella misma.

¿Qué función de transición utiliza el Juego de la Vida?

El Juego de la Vida trata de simular el comportamiento observado al microscopio de colonias de organismos bacterianos, teniendo en cuenta aspectos como la presencia de nutrientes, la relación entre bacterias, y la reproducción. Un punto de la retícula bidimensional (una bacteria) puede estar en dos estados posibles: viva y muerta; el cambio de estado viene descrito por una función de transición que se concreta mediante cuatro (o a veces tres equivalentes, depende de la bibliografía) reglas muy simples:

1. Cualquier bacteria viva con menos de dos vecinas vivas muere.
2. Cualquier bacteria viva con dos o tres vecinas vivas, sigue viva.
3. Cualquier bacteria con más de tres vecinas vivas muere por sobrepoblación.
4. Cualquier punto de la retícula vacío (donde murió previamente una bacteria o no había ninguna) con exactamente tres vecinas vivas se coloniza por una bacteria viva, por reproducción de sus vecinas.

¿Cómo fijo la configuración inicial de mi retícula?

Vuestro software debe permitir tres tipos configuraciones iniciales, que el usuario elegirá desde la interfaz:

1. Estado inicial de las bacterias aleatorio, dando lugar a una configuración inicial de la retícula estadísticamente homologable a una nube de puntos sin estructura. Luego, la evolución en el tiempo da lugar a patrones que se pueden parecer –o no– al siguiente:

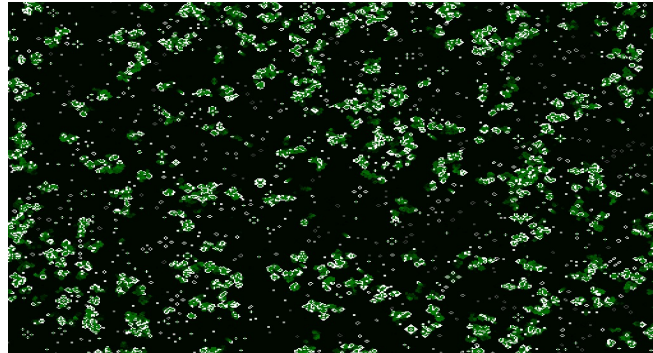


Figura: El Juego de la Vida en un punto de la simulación.

2. Sembrando la retícula de “islas de bacterias” separadas por zonas donde no hay actividad vital, para poder ver cómo se comportan esos núcleos de bacterias aislados, como en la Figura que sigue; no hace falta que las islas sean tan regulares ni tan estructuradas.

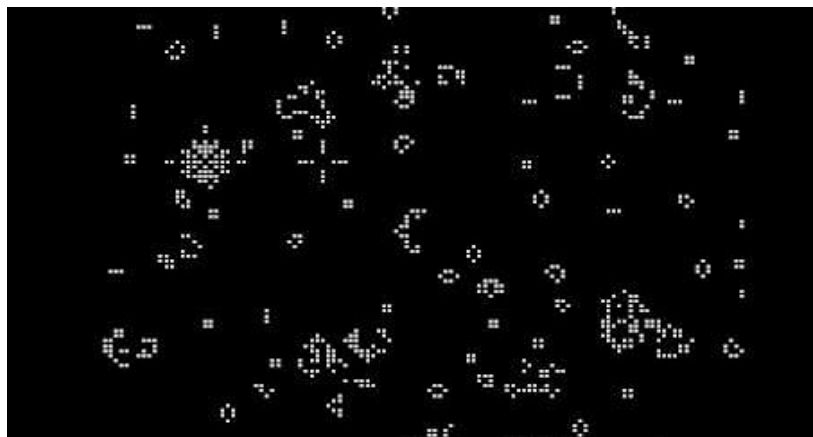


Figura: Núcleos bacterianos aislados.

3. Insertando en una retícula vacía varios núcleos bacterianos conocidos como cañones de planeadores (*gliders guns*), que son colonias metaestables (permanecen en el tiempo sin cambios) y que tienen la capacidad de “disparar” núcleos bacterianos que se propagan por la retícula (los planeadores). Más información sobre estos cañones y cómo configurarlos en <https://www.conwaylife.com/wiki/Glider>. En siguiente Figura, un cañón de planeadores “disparando”:



Figura: Un cañón de planeadores “disparando”; los planeadores se desplazan en diagonal hacia abajo.

¿Qué curvas debe graficar la aplicación?

Similares la que ya habéis desarrollado para el autómatas celulares 1-D para distancias de Hamming o entropía: se trata de una curva de población que muestre el número de bacterias vivas y el de huecos vacíos de la retícula, como una función del tiempo de simulación (número de generación). La siguiente imagen os da una idea de cómo integrar esas curvas en la interfaz de la simulación:

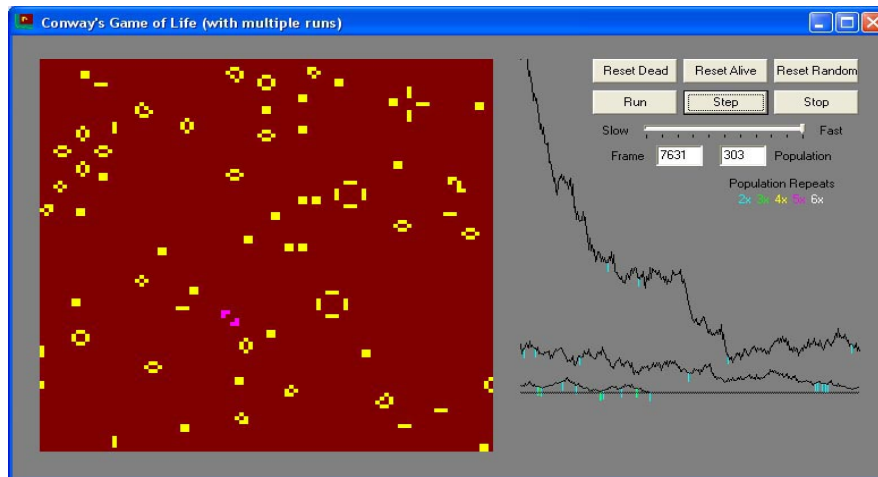


Figura: Posible interfaz para el Juego de la Vida, integrando el estado de la colonia de bacterias a la izquierda, los parámetros de configuración arriba a la derecha, y las curvas de población abajo a la derecha.

¿Qué inputs y outputs debe tener el GUI?

Inputs:

- Elección en un desplegable del tipo de estado inicial (aleatorio, islas de bacterias o cañones de planeadores)
- Número de generaciones a computar.

Outputs

- Evolución gráfica en el tiempo del estado de la retícula.
- Curvas de población celular

¿Qué elementos de control debe tener el GUI?

Únicamente dos: un botón de “Inicio” para activar la simulación una vez fijados los parámetros de entrada, y otro de “Reset” para limpiarlo todo y comenzar de nuevo.