

Trabajo Práctico Nº 1

El juego de la vida

Estructura de Datos – Plan 2013

Fecha de entrega:

Comisión martes: 12/09/2017

Comisión Jueves: 14/09/2017

Juego de la vida

El juego de la vida es un autómata celular (un sistema con un tablero o grilla donde se aplican reglas a una celda o célula y sus vecinos), diseñado por el profesor John Conway en la Universidad de Princeton en 1970, es un ejemplo de complejidad emergente o sistemas auto organizados, que sirven para estudiar cómo pueden surgir patrones y comportamientos complejos a partir de reglas simples. Ver:

https://es.wikipedia.org/wiki/Juego_de_la_vida

<http://www.math.com/students/wonders/life/life.html>

El objetivo de este trabajo práctico es implementar el juego de la vida en Python, usando objetos.

Cómo se juega

El juego de la vida es un juego determinado por sus propias reglas y por un patrón inicial. El juego comienza con un tablero de $n \times m$ celdas que pueden estar vivas o muertas.

El estado de todas y cada una de las $n \times m$ celdas forman un patrón.

Cada celda tiene 8 vecinos, salvo las que están sobre los bordes del tablero (Ver figura)

1	2	3
4		5
6	7	8

Figura 1: una celda y sus vecinos

Aplicando las reglas que definen el juego se obtiene la próxima generación del patrón. Las reglas son:

1. Una celda muerta con tres vecinos vivos revive.
2. Una celda viva con 2 o tres vecinos vivos permanece viva.
3. Una celda con menos de 2 o más de tres vecinos vivos muere.

Las reglas se aplican al tablero actual generando un nuevo patrón, como se muestra en la figura 2.

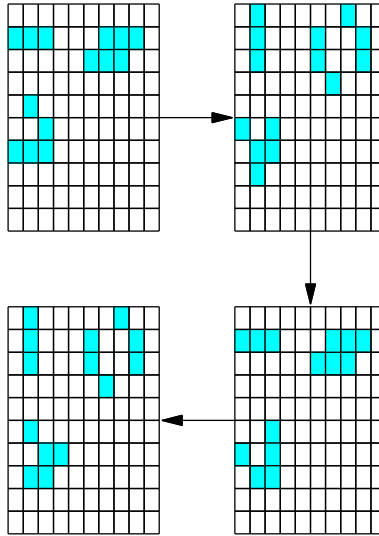


Figura 2: Aplicación de reglas – Generaciones

Especificaciones

1. Las celdas muertas se representan con un guión ('-') y las vivas con un asterisco (*). Por ejemplo:

Columna	0	1	2	3	4	5	6	7	8	9
Fila 0	-	-	-	-	-	-	-	-	-	-
Fila 1	*	*	*	-	-	-	*	*	*	-
Fila 2	-	-	-	-	-	*	*	*	-	-
Fila 3	-	-	-	-	-	-	-	-	-	-
Fila 4	-	*	-	-	-	-	-	-	-	-
Fila 5	-	-	*	-	-	-	-	-	-	-
Fila 6	*	*	*	-	-	-	-	-	-	-
Fila 7	-	-	-	-	-	-	-	-	-	-
Fila 8	-	-	-	-	-	-	-	-	-	-
Fila 9	-	-	-	-	-	-	-	-	-	-

Notar que las columnas empiezan en 0 desde la izquierda y las filas en 0 desde arriba.

2. Al iniciar el juego el usuario dispone de varias opciones:
 - a. Iniciar un nuevo juego. El usuario debe ingresar el tamaño del tablero.
 - i. Empezar con un patrón al azar. En cuyo caso se le debe preguntar al usuario con cuantas celdas vivas se quiere iniciar y generar el patrón.
 - ii. Introducir un patrón inicial indicando la fila y columna de cada celda viva.
3. Para el desarrollo del juego se deberá contar con un menú con las siguientes opciones:
 - i. Avanzar paso a paso, mostrando en cada paso el tablero completo. Al final de cada paso el usuario debe poder modificar el estado de una celda dada.
 - ii. Ejecutar el juego de la vida, mostrando a cada paso el tablero. El programa se detiene cuando encuentra un patrón vida estática, donde el tablero ya

no se modifica o un patrón oscilador de nivel 2, donde se alternan dos generaciones infinitamente.

4. Jugar en modo Vidas Estáticas. En este modo se deben encontrar todos los patrones vidas estáticas (still life, en inglés). Un patrón vida estática es aquel en donde las celdas no cambian de generación en generación. A medida que los va encontrando los debe mostrar por pantalla. Al iniciar le pregunta al usuario la cantidad de celdas vivas que tendrán los patrones. Por ejemplo en un tablero de 5x5 y patrones de 5 celdas vivas, se tienen combinaciones de 5 celdas en un total de 25 celdas, lo que da alrededor de 53000 tableros distintos para probar si son vida estática.
5. Guardar el juego actual para poder recuperarlo y continuar luego, ya sea en el modo normal o en el modo vidas estáticas. El usuario debe ingresar el nombre del archivo.
6. Recuperar un juego guardado en disco (se debe pedir al usuario el nombre del archivo).
7. En cualquier momento el usuario puede presionar Ctrl-C y volver al menú inicial que le permita guardar el juego para recuperarlo luego.

Se deberá usar la función dada *combination* que tiene dos parámetros: una lista de items y una cantidad N. La función enumera todas las combinaciones de los items en L tomados de a N. Por ejemplo, dada la lista $L = [0, 1, 2]$ y $N = 2$, la función *combination* devolverá todas las combinaciones de dos elementos: [0,1], [0,2] y [1,2]. En este caso se puede usar dicha función asignando a cada celda del tablero un número único. Por ejemplo en un tablero de 5x5 y patrones de tamaño 5 se pueden numerar todas las celdas desde 0 hasta 24 y generar todas las combinaciones posibles de 5 celdas (celdas vivas) y probar si son vida estática.