
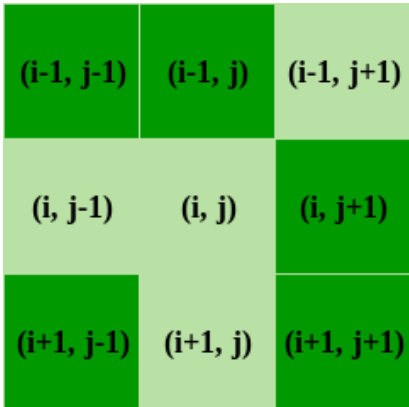
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		PRÁCTICA DE LABORATORIO
CARRERA: COMPUTACIÓN/INGENIERÍA DE SISTEMAS		ASIGNATURA: Simulación
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Frecuencias de lanzamientos
OBJETIVO ALCANZADO		
<p style="text-align: center;">ACTIVIDADES DESARROLLADAS</p> <p>Emplear el programa del juego de la vida de John Conway para realizar la siguiente simulación:</p> <ul style="list-style-type: none"> Determinar las Variables que rigen el sistema <p>Debido a que el Juego de la vida se basa en una cuadrícula de nueve cuadrados, cada celda tiene ocho celdas vecinas, como se muestra en la figura dada.</p> <p>Se accede a una celda dada (i, j) en la simulación en una cuadrícula [i] [j], donde i y j son los índices de fila y columna, respectivamente.</p> <p>El valor de una celda dada en un instante de tiempo dado depende del estado de sus vecinas en el paso de tiempo anterior. El Juego de la vida de Conway tiene cuatro reglas</p> <div style="text-align: center; margin: 20px 0;">  </div> <p style="text-align: center;">Variables Que Rigen:</p> <ul style="list-style-type: none"> ○ frameNum ○ img ○ grid ○ N <p>Nos basamos de en una cuadrícula con un tamaño de n x n (célula)(Variable N)</p> <p>Realizamos un método random (randomGrid) donde la célula debe cumplir con reglas para estar con vida que son:</p> <ol style="list-style-type: none"> 1. Muere si hay menos de dos vecinos vivos. 2. Continúe viviendo si hay exactamente dos o tres vecinos vivos. 		

3. Muere si hay más de tres vecinos vivos.

Tenemos el método update que nos permite copiar la cuadrícula ya que necesitamos 8 vecinos para el cálculo y vamos línea por línea.

FUNCIONAMIENTO

1. Primero, creamos dos listas vacías para contener las células que serán destruidas y las células que sobrevivirán o cobrarán vida.
2. Pasamos por la cuadrícula con un bucle anidado.
3. Usamos método check en la celda que estamos revisando.
4. Creamos una lista con células vivas del cheque.
5. Recorrimos la lista devuelta de la función y vemos cuántos de los vecinos están vivos.
6. Según las reglas del juego, la celda actual se coloca en una lista para matar o en una lista para mantener con vida / volverse con vida.

- **Diseñar un plan de simulación que permita llegar a una configuración en que los autómatas celulares no varían.**
 - Se debe generar una semilla con el siguiente plan:
 - El plan de vida de una célula debe tener más de tres vecinos para que pueda sobrevivir.
 - Si alguna célula está muerta debe tener tres vecinas nace una nueva
 - La célula debe tener más de tres células vecinas para sobrevivir o muere.
- **Diseñar un plan de pruebas automatizadas es decir que sea controlado por el ordenador y que permita lanzar una batería de experimentos.**

Se debe generar un método que cumpla con las leyes de vida que son :

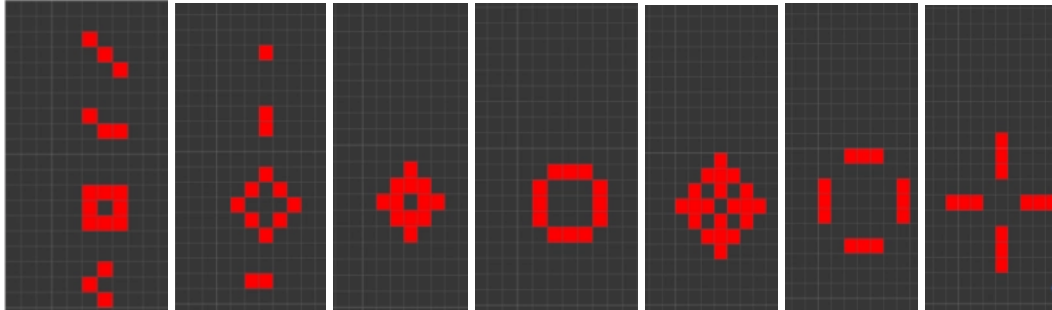
- Muere si hay menos de dos vecinos vivos.
- Continúe viviendo si hay exactamente dos o tres vecinos vivos.
- Muere si hay más de tres vecinos vivos.


Este método debe cumplir con estas leyes y utilizar un método randomico para lanzar diferentes muestras de experimento aleatorias.

A Través de este método ingresamos la cantidad de muestras que vamos a generar para nuestra célula inicial y obtener diferentes resultados.

```
def randomGrid(N):  
    return np.random.choice(vals, N*N, p=[0.2, 0.8]).reshape(N, N)
```

- Recopilar los resultados de estos planes.

Resultados**CONCLUSIÓN:**

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		