

# ESCUELA SUPERIOR POLITECNICA DEL LITORAL

## Proyecto Programación Orientada a Objeto

### 1er Parcial – 2017 II

Este proyecto modela la rebelión de una población subyugada contra una autoridad central usando un “Modelo basado en Agentes” (ABM). Es una adaptación del modelo de violencia civil de Joshua Epstein (2002).

La población deambula de forma aleatoria. Si su nivel de agravio contra la autoridad central es lo suficientemente alto, y su percepción de los riesgos involucrados es lo suficientemente baja, se rebelan abiertamente. Una población de oficiales de policía ("policías"), actuando en nombre de la autoridad central, busca reprimir la rebelión. Los policías deambulan aleatoriamente y arrestan a personas que se están rebelando activamente.

#### ¿Como funciona?

Cada agente (policías y población) se ubican de manera aleatoria en una casilla de un universo bidimensional.

Cada miembro de la población general tiene un nivel individual de “agravio” o “queja” hacia la autoridad central. Los “AGRAVIOS” están basados en el PERJUICIO-PERCIBIDO, el cual es asignado aleatoriamente en el momento de creación de los agentes, y la LEGITIMIDAD DEL GOBIERNO, la cual es global a través de todos los agentes.

$$AGRAVIO = PERJUICIO-PERCIBIDO * (1 - LEGITIMIDAD DEL GOBIERNO)$$

Cada miembro de la población también calcula un riesgo individual de rebelarse al comienzo de cada turno. Esta PROBABILIDAD DE DETENCIÓN ESTIMADA se basa en el número de policías y agentes rebeldes dentro de los parches VISION, concretamente

$$PROBABILIDAD DE DETENCIÓN ESTIMADA = 1 - \exp(-k * \text{round}(C / A))$$

Donde:

(C / A) es la proporción de policías por cada miembro de la población que se encuentre activamente rebelándose.

k = 2.3, es una constante para todo el universo que permite obtener un valor razonable cuando la proporción es 1 a 1

Con la **PROBABILIDAD DE DETENCIÓN ESTIMADA** se calcula el RIESGO NETO, dado que hay personas mas arriesgadas que otras el parámetro de AVERSION AL RIESGO es dado aleatoriamente a cada uno de los agentes al momento de nacer (un valor del 0 al 1).

$$RIESGO NETO = AVERSION AL RIESGO * PROBABILIDAD DE DETENCION ESTIMADA.$$

Finalmente, que un miembro de la población se rebele dependerá de:

$$AGRAVIO - RIESGO NETO > LIMITE$$

El límite es una constante para todo el universo y tiene un valor de 0.1

### Reglas del Modelo:

El modelo pasa por tres reglas diferentes, conocidas como M (movimiento), A (agentes) y C (policías):

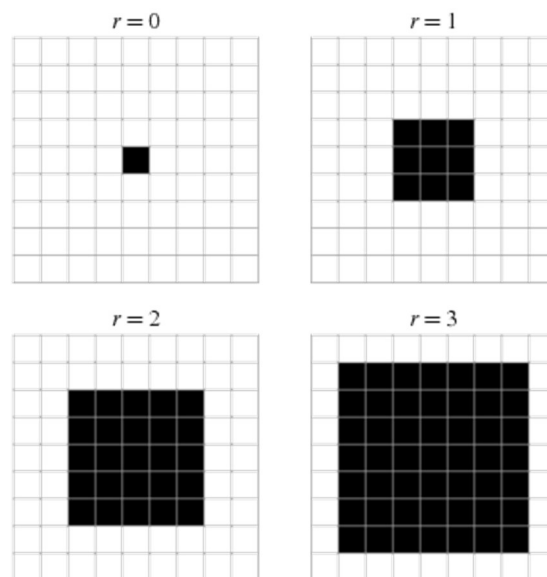
La regla de movimiento M dice que cada policía y miembro de la población que no esté encarcelado se mueve a un sitio aleatorio no ocupado dentro de los parches de VISION.

La regla A del agente dice que si el AGRAVIO O QUEJA de un miembro de la población excede el RIESGO NETO en un pequeño límite (establecido por defecto a 0.1), el agente decide rebelarse. Por ende, el agente o miembro pasa de inactivo a activo. Es importante mencionar que también podría pasar de activo a inactivo

La regla P de Policía dice que cada policía debe buscar agentes activos dentro de los parches de VISION. Si al menos uno existe, el policía selecciona aleatoriamente un agente activo y lo envía a prisión por un número de turnos entre 0 y MAX-TIEMPO-CARCEL y se mueve al parche del agente encarcelado. El parche de la persona encarcelada se considera desocupado.

### Campos de Visión y Vecindario

Los miembros de población y policías tienen un campo de visión, que determina su vecindario. En la figura 1, se puede observar la representación del vecindario dependiendo su campo de visión, si el campo de visión es 1, los policías buscaran agentes rebelándose solo en su vecindario. Por otra parte, los agentes calcularan su riesgo de ser arrestados evaluando cuantos policías existen en su vecindario.



### Interfaz de Usuario y Parámetros del Universo

La interfaz de usuario deberá proveer formas de setear los parámetros con que se correrán las simulaciones.

Se debe poder establecer

- % de densidad Inicial de Policías: 0-100%
- % de densidad Inicial de Agentes (miembros de la población general) 0-100%
- Tamaño del Universo (Numero de Celdas de alto y ancho)
- Máximo numero de turnos en la cárcel.
- Visión de policías y agentes
- Máximo Número de turnos (Cuantas veces se correrá la simulación)
- Legitimidad del Gobierno (Valor entre 0 y 1)
- Movimiento: On o Off (Si esta apagado los miembros de la población no se mueven)

Por cada turno debe presentar como el universo va cambiando, es decir como se mueven los agentes cuales están activos, inactivos o encarcelados. También, debe mostrar los policías. (Todo esto es por consola, se bajarán puntos para los que usen innecesariamente interfaz gráfica en este proyecto)

Adicionalmente, por cada corrida debe generar un archivo .csv que contenga un detalle de cada turno con:

*Numero de Turno, # de agentes Tranquilos, # de agentes rebelándose, # de agentes en la cárcel*

### **Diagrama de Clases (Avance 1)**

- Deberá ser presentado el día viernes 10 de noviembre de 2017

## **Modo de Calificación Proyecto:**

### **Funcionalidad (40 puntos)**

- Se valorará la apariencia del proyecto, sin que esto implique que el estudiante deba utilizar recursos adicionales a los aprendidos en clase. (El proyecto es en consola)
- Se verificará el cumplimiento de toda la funcionalidad requerida, así como las correctas validaciones de los datos.
- El proyecto debe ser debidamente probado antes de presentarlo al profesor. Por lo tanto Se espera que el programa no se caiga al ejecutarlo, ni tenga un comportamiento no esperado (Si se cae en la presentación, son 5 puntos menos por cada caída)

### **Documentación y Ejecutable(10 puntos)**

- El programa fuente deberá presentar un código documentado internamente (un correcto uso de comentarios). Deberá presentar Java Docs
- Realizar Diagrama de Clases
- Informe de su trabajo.
- Jar debe ser creado su proyecto debe correr con java -jar proyecto1.jar

### **Abstracción y Uso de Objetos (30 puntos)**

- Creación de Clases pertinentes con sus propiedades y métodos
- Correcta interacción de objetos. Que los objetos se comuniquen entre si y no solo sean llamados todos en el main.
- Usar recursos de herencia y/o interfaces para la solución.

### **Modularidad y Encapsulamiento (20 puntos)**

- Dividir el problema usando los métodos correspondientes. No escribir bloques inmensos de código.

- Encapsular correctamente el proyecto. Crear paquetes donde se agrupen clases que se relacionen.
- No olvidar que para acceder a las propiedades de las clases se debe proveer los métodos get y set.

**Penalidades:**

- *No uso de repositorio*

**Referencias:**

“Modeling civil violence: An agent-based computational approach”, Proceedings of the National Academy of Sciences, Vol. 99, Suppl. 3, May 14, 2002, and is available at <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC128592/>.