

M1. Actividad



Modelación de sistemas multiagentes con gráficas computacionales

Omar Jiménez Armendáriz	A01732097
Francisco Rocha Juárez	A01730560
Alejandro Alfonso Ubeto Yañez	A01734977

1.- Contexto

1.1.- Especificaciones del modelo

El modelo a crear debe simular una habitación de $m \times n$ espacios en el que, dado un porcentaje de suciedad, se generan aleatoriamente celdas que representan suciedad y polvo. Este modelo además debe contar con un determinado número de agentes limpiadores que tras haberse posicionado sobre una celda sucia, la limpian.

Finalmente, si antes de llegar a un step específico no se ha limpiado por completo el área, el programa se detiene al igual que cuando se limpia en su totalidad la superficie y por ende ya no quedan celdas sucias que limpiar.

Las **variables** de control a tomar en cuenta son:

- El **largo** de la habitación
- El **ancho** de la habitación
- El **número** de agentes limpiadores
- El **tiempo** máximo de ejecución del modelo

A continuación se muestran ejemplos de configuración del modelo desarrollado con el lenguaje de programación *Python*, más específicamente aprovechando su librería para programación y modelación de agentes, *Mesa*.

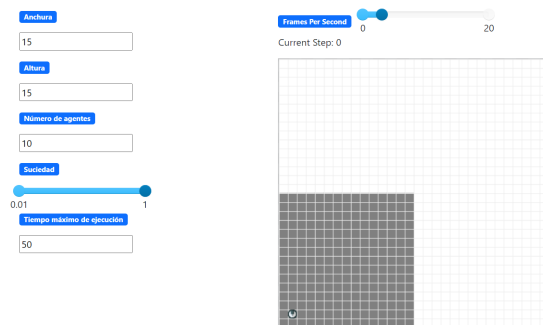


Imagen 1. Ejemplo de modelo: 15x15 densidad 100%

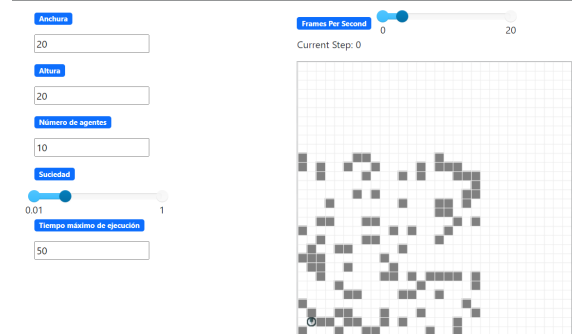


Imagen 2. Ejemplo de modelo: 20x20 densidad 25%

1.2.- Explicación la configuración del modelo base

La siguiente imagen representa el modelo de una habitación de 30 cuadrículas de altura por 30 cuadrículas anchura que cuenta con 10 agentes limpiadores los cuales parten de un mismo punto (Coordenadas [1,1]) con el **objetivo limpiar** las celdas sucias (celdas grises) que se encuentren en el camino.

Esta es la configuración del modelo que se utilizará para analizar el desempeño del producto final en el presente reporte. Se tomó esta decisión pues el tener una densidad del 50% es ideal para ver en partes iguales el cómo los agentes se desplazan y limpian celdas sucias en cada step.

Continuando con el diseño del modelo, se mantienen las medidas estándar de 30x30 para que el comportamiento de los agentes sea apreciable a simple vista y, finalmente, el tiempo máximo de ejecución será de 50 steps para apreciar el desempeño de 10 agentes.

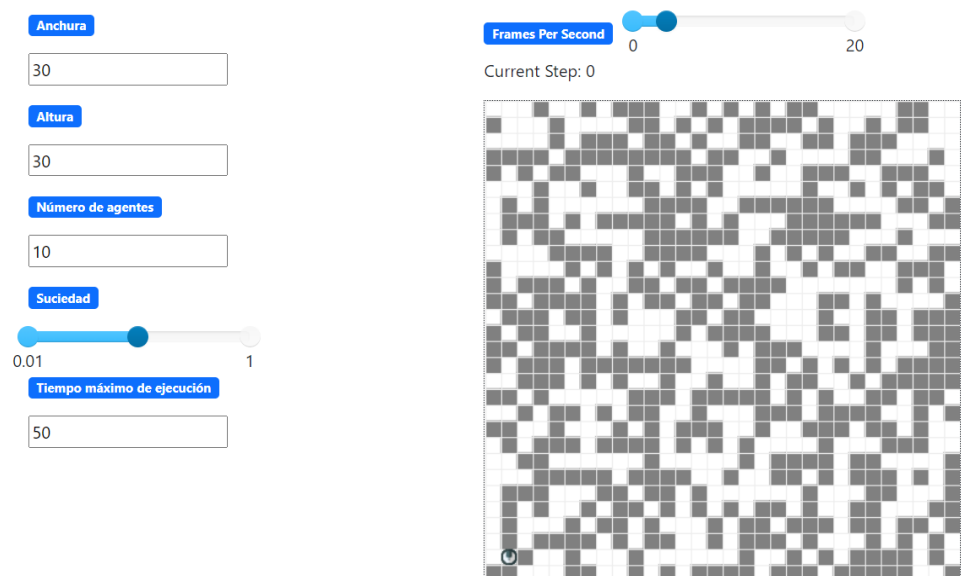


Imagen 3. Configuración base del modelo

1.3.- Comportamiento durante ejecución

El comportamiento de los agentes limpiadores es el siguiente: en cada paso de tiempo; si se encuentran sobre una celda sucia, la aspiran, si se encuentran sobre una celda limpia, eligen una dirección aleatoria para moverse hacia una de las 8 celdas vecinas, sin embargo, si esta celda está ocupada, el agente permanecerá en su lugar hasta el siguiente step en el que volverá a tomar una dirección aleatoria.

A continuación se muestra el resultado final de la ejecución del modelo con la configuración base. Se aprecia que solo llegaron a limpiar cerca de un 25% de las celdas sucias, esto ocurre por el factor aleatorio e incalculable de los movimientos de cada agente que los hace dirigirse a celdas que ya estaban limpias e incluso regresando a zonas donde ya habían limpiado.

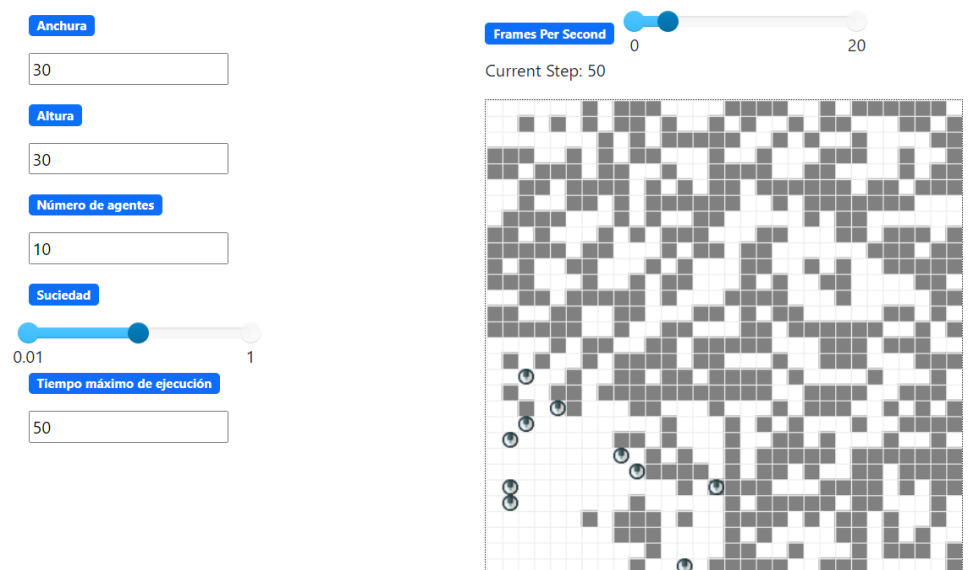


Imagen 4. Modelo base tras ejecución

Es visible el cómo a pesar de contar con una gran cantidad de agentes limpiadores, estos tienden a retroceder en su camino tanto como a avanzar en direcciones arbitrarias, por lo que se puede concluir que es necesario el programar un método de toma de decisiones más controlado y orientado a la **eficiencia** para abarcar un mayor número de celdas limpiadas en el menor tiempo posible.

Esto podría desarrollarse haciendo que los agentes no solo sean conscientes de la existencia de sus compañeros, sino también de colaborar entre sí para seguir una estrategia en equipo que les garantice un mejor desempeño. Para hacer a estos agentes más inteligentes una herramienta que se podría utilizar es la librería de *pathfinding* que les permitirá seguir una ruta hacia un objetivo determinado de manera lógica.

2.- Resultados

2.1.- Análisis de gráficos

Utilizando la herramienta Datacollector de la librería mesa, fue posible recopilar información del modelo en cada paso de tiempo, lo que permitió crear las siguientes gráficas que ilustran y generan conocimiento sobre el comportamiento de las múltiples variables del modelo. A continuación se muestran los resultados obtenidos.

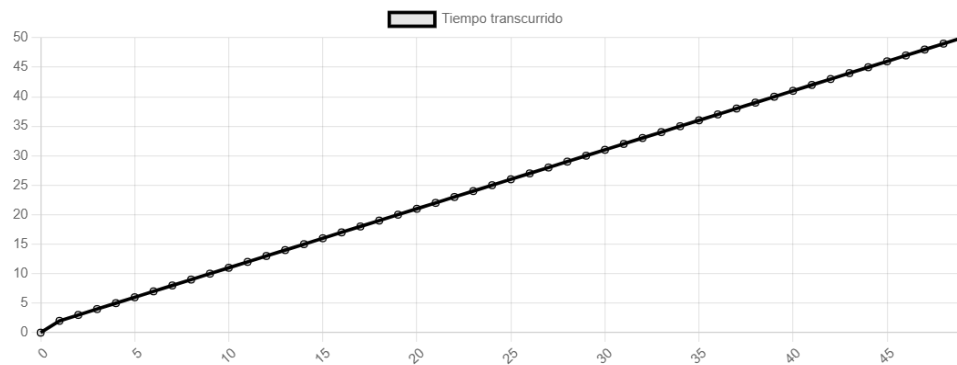


Imagen 5. Gráfico del paso del tiempo (steps)

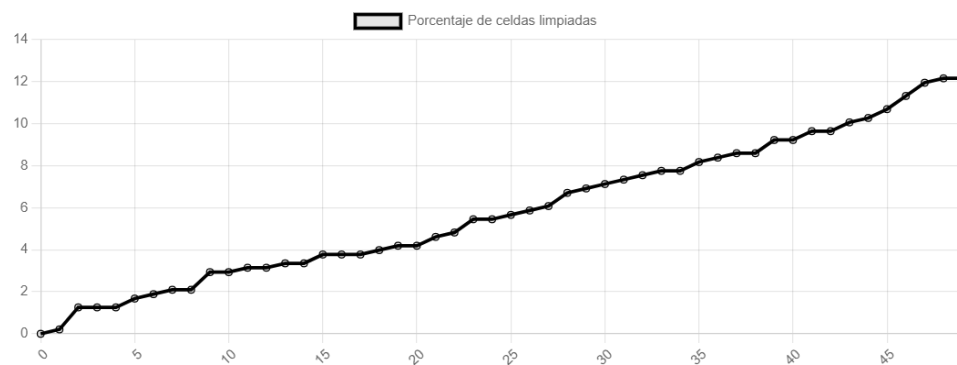


Imagen 6. Gráfico del porcentaje de celdas limpiadas con respecto al tiempo

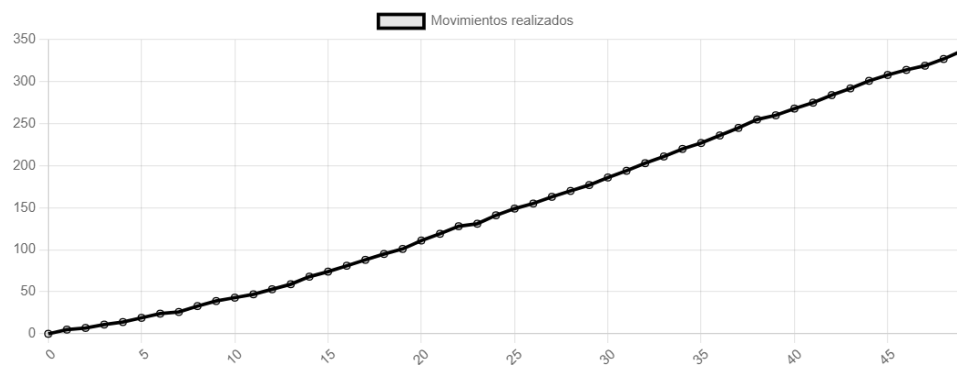


Imagen 7. Gráfico de sumatoria de movimientos realizados por todos los agentes del modelo

2.2.- Interpretación

La **imagen número 5** representa la evolución de los pasos de tiempo (steps) en cada paso de tiempo. Este gráfico crece de forma lineal de la forma $y = x$ ya que en cada paso de tiempo los steps han aumentado en 1 con respecto al paso anterior. Pero esto es algo evidente, lo verdaderamente interesante es lo que se expone en el siguiente gráfico.

La **imagen número 6** representa el número de celdas que han sido limpiadas con cada uno de los pasos de tiempo. Dada la naturaleza aleatoria de los agentes limpiadores, esta es una gráfica que tiende a tener un comportamiento impredecible pero que a juzgar la relación entre: el número de agentes, el espacio a limpiar y el tiempo disponible, llega a ser una línea con ligeras desviaciones que siempre va en ascenso o se mantiene, en ningún momento decrece ya que representa la cuenta de las celdas limpiadas en los steps anteriores más las limpiadas en el step actual.

Finalmente, la **imagen número 7** es muy interesante, ya que representa el cúmulo de movimientos dados por todos los agentes en cada uno de los pasos de tiempo. Al igual que su predecesora, tiende a ser una línea con muy ligeras desviaciones que siempre va en aumento solo que a diferencia del gráfico anterior es que el comportamiento de esta no es aleatorio.

Como se aprecia en el inicio de la línea, tiende a curvarse un poco nada más empezar la ejecución, esto ocurre ya que al iniciar la simulación, no todos los agentes son libres de moverse pues deben esperar a que sus celdas vecinas se desocupen para poder avanzar. Es por esto mismo que con el paso de unos pocos instantes, el gráfico se vuelve lineal, ya que todos los agentes limpiadores (o la gran mayoría) son libres de moverse a través del tablero.

2.3.- Conclusión

Tras haber realizado este ejercicio de programación de agentes, el equipo se queda deseoso de implementar un mayor poder de decisión para los mismos con el fin de orientarlos hacia una elección basada en eficiencia y cooperación que genere mejores y más interesantes resultados.

Cada uno de los integrantes del equipo reflexionan con asombro sobre la importancia del aspecto ético detrás de la programación de agentes que pueden moverse dentro de un espacio, ya que si se toma la irresponsable decisión de hacer que se muevan de forma aleatoria podría derivar en accidentes que involucren daños hacia los bienes e integridad física del consumidor e incluso terminar en resoluciones legales.

3.- Anexos

3.1.- Repositorio remotos

Accede al [repositorio en BitBucket](#) del proyecto y a su [versión de GitHub](#).