



INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Campus Estado de México

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 301)

M1. Actividad

Profesores:

Jorge Adolfo Ramírez Uresti

Sergio Ruiz Loza

Integrantes:

Carlos Iker Fuentes Reyes	A01749675
Santiago Chevez Trejo	A01749887

El presente reporte actúa como un registro y análisis del sistema de robots de limpieza programado con la librería de mesa, para simular un escenario en el que un número de N de agentes roombas vagan sin rumbo en un espacio de dos dimensiones ($N \times M$), en el cual existen algunas celdas sucias.

El código diseñado [[Link al repositorio](#)] cuenta con 3 componentes principales que serán tratados en el presente documento:

1. El modelo [RombaModel] que lleva registro de los datos de la simulación, instancia el ambiente y genera a los agentes.
2. Los agentes Roomba [RoombaAgent] que son los robots de limpieza, y las celdas [Cell], que incluso si están modelados como agentes en mesa, son un componente del ambiente con el que interactúan los agentes.
3. Un código de visualización que representa de manera gráfica la simulación del ambiente en un tiempo determinado. Éste hace uso de la librería de Mesa Visualization para iniciar un servidor que ejecuta una simulación del modelo.

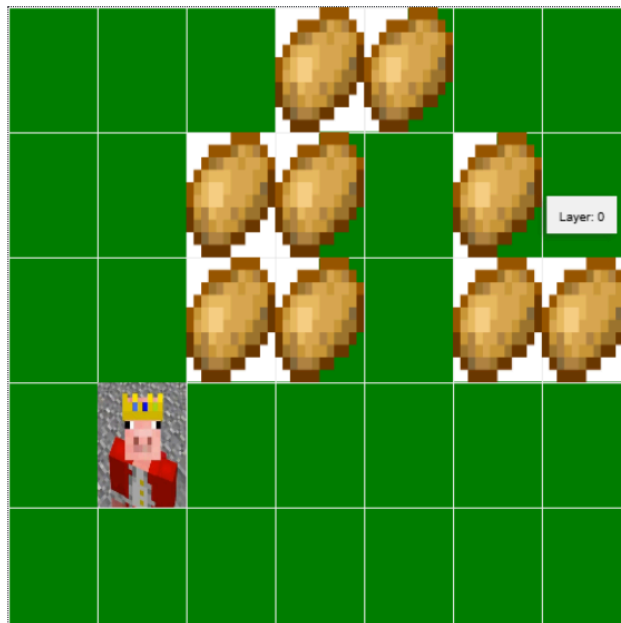
Los agentes Roomba son considerados agentes porque tienen las siguientes características:

1. Son autónomos: nadie le dice al agente a donde ir, su decisión de movimiento depende completamente de si la celda en la que está, se encuentra sucia o no. En caso de que se encuentre con tierra, la limpia y no se mueve sino hasta la siguiente simulación.
2. Son proactivos: toman decisión a donde ir con base en su ambiente.
3. Son reactivos: si encuentran algo sucio, inmediatamente lo limpian sin analizar más a fondo.
4. Son sociales: La realidad es que no son particularmente sociables [no hablan entre roombas], pero sí existe un intercambio de mensajes entre ellos y el ambiente de las celdas.

Asimismo, se contextualiza a los agentes Roombas en el siguiente ambiente:

1. El ambiente es un grid de dimensiones MxN.
2. El ambiente tiene celdas que pueden estar sucias o limpias.
3. En el ambiente las roombas existen y se mueven entre las celdas.
4. El tiempo máximo de ejecución es 115

La siguiente imagen muestra un ejemplo del ambiente:

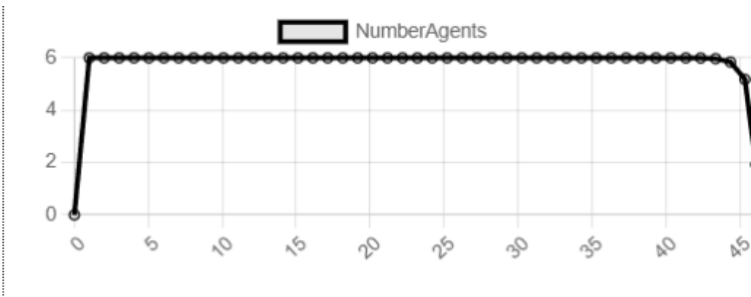


A continuación mostramos los resultados de diferentes escenarios:

Escenario 1:

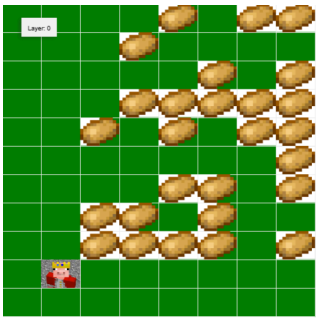


- Grid 7x5
- 9 celdas sucias (25.7%)
- 16 agentes



Steps	Cleaned Percentage	Total Movements	Time Taken
12	100%	167	11

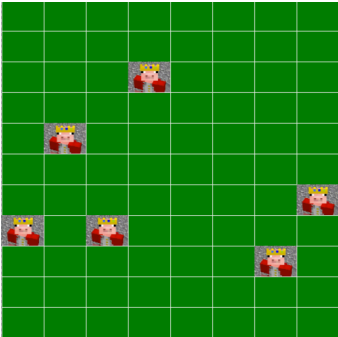
Escenario 2:



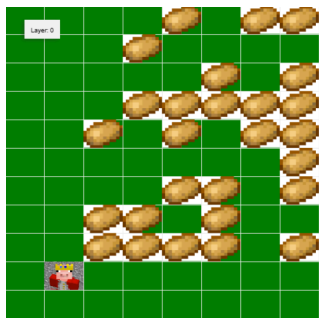
- Grid 12x8
- 26 celdas sucias (28.1%)
- 6 agentes



Steps	Cleaned Percentage	Total Movements	Time Taken
113	100%	639	112

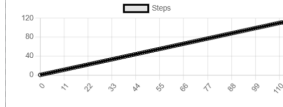

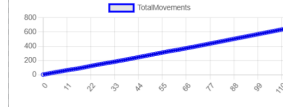



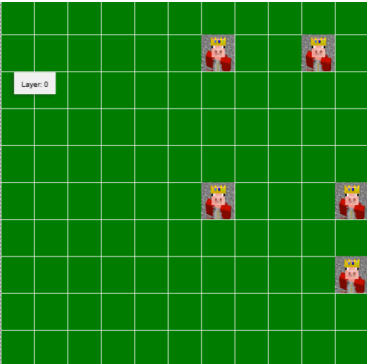
Escenario 3:



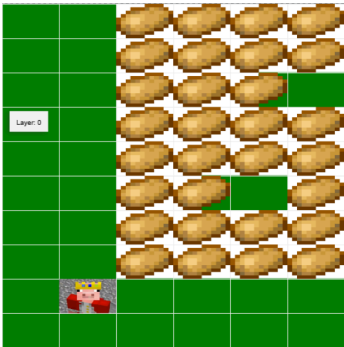
- Grid 11x10
- 53 celdas sucias (48.18%)
- 5 agentes



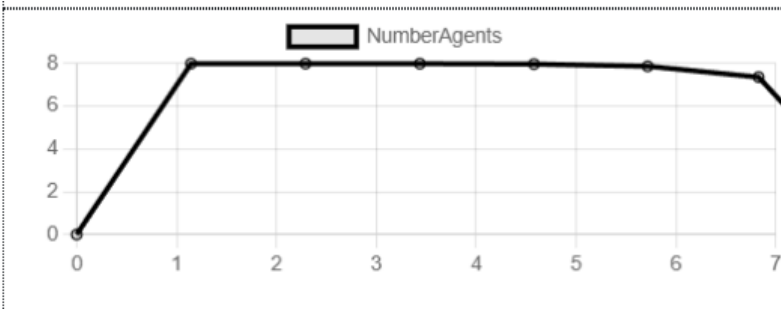
Steps	Cleaned Percentage	Total Movements	Time Taken
			
101	100%	452	101



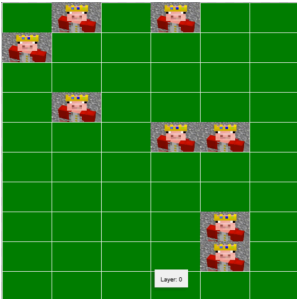
Escenario 4:



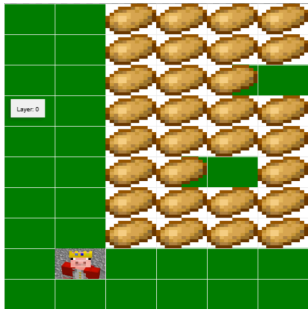
- Grid 6x10
- 26 celdas sucias (43.3%)
- 8 agentes



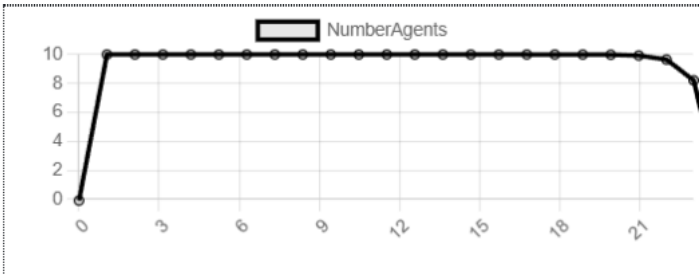
Steps	Cleaned Percentage	Total Movements	Time Taken
35	100%	250	34



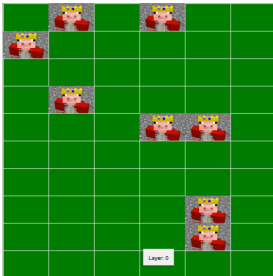
Escenario 5:



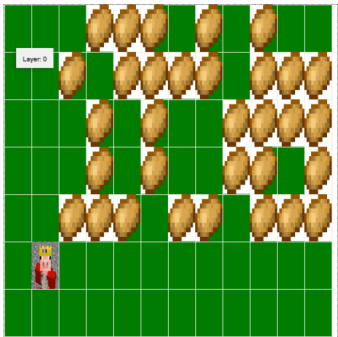
- Grid 12x4
- 14 celdas sucias (29.16%)
- 10 agentes



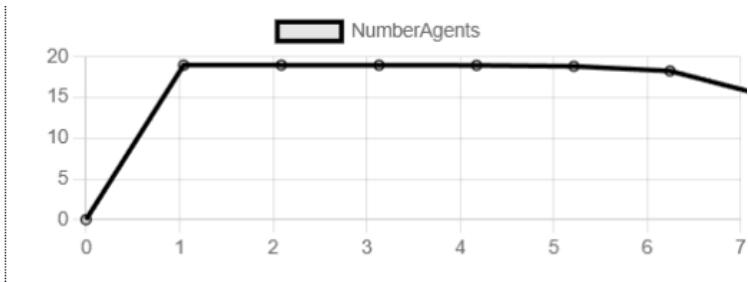
Steps	Cleaned Percentage	Total Movements	Time Taken
46	100%	446	45



Escenario 6:

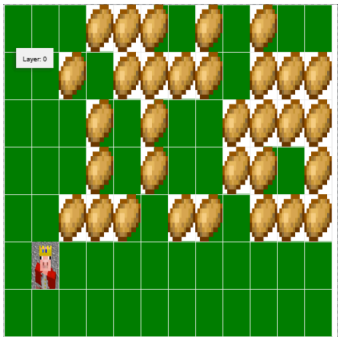


- Grid 12x7
- 32 celdas sucias (38.09%)
- 19 agentes

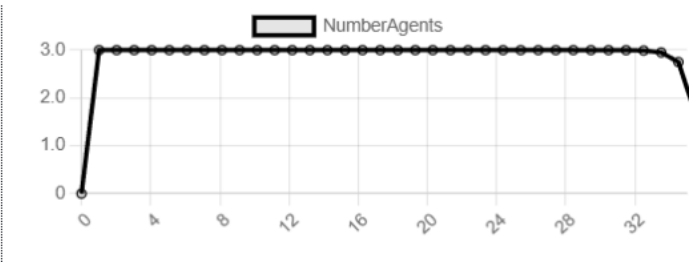


Steps	Cleaned Percentage	Total Movements	Time Taken
35	100%	633	34

Escenario 7:



- Grid 10x7
- 17 celdas sucias (24.2%)
- 3 agentes



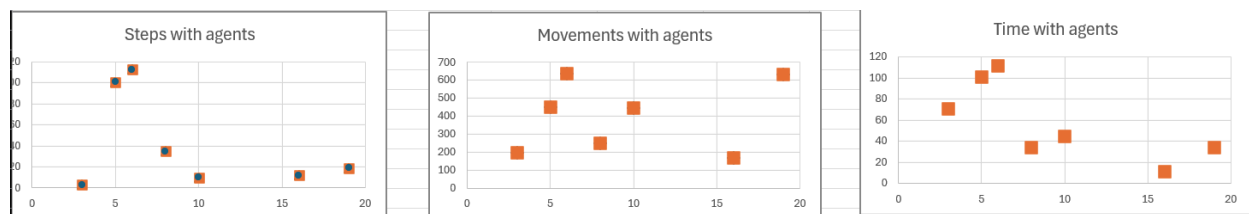
Steps	Cleaned Percentage	Total Movements	Time Taken
72	100%	199	71

Estos resultados se resumen en la siguiente tabla:

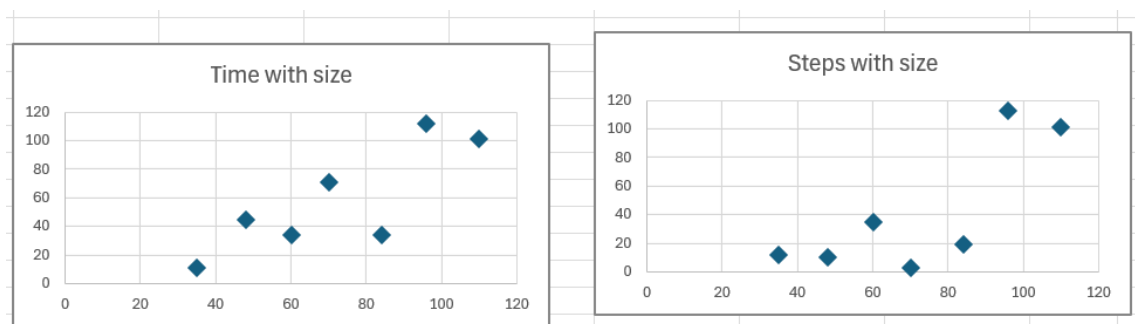
	Escenario 1	Escenario 2	Escenario 3	Escenario 4	Escenario 5	Escenario 6	Escenario 7
nAgentes	16	6	5	8	10	19	3
Steps	12	113	101	35	10	19	3
Cleaned%	100	100	100	100	100	100	100
Movements	167	639	452	250	446	633	199
Time	11	112	101	34	45	34	71
Size	35	96	110	60	48	84	70
Dirty	9	26	53	26	10	32	17

El comportamiento de los agentes es aleatorio, puesto que, elige una posición aleatoria dentro de una de sus 8 posiciones adyacentes. Por lo que uno debería esperar relaciones sin tendencia aparente al relacionar el número de agentes con los demás valores. La tendencia que debería afectar al tiempo, sería el tamaño del grid y la proporción de celdas sucias. En caso particulares de un número de agentes equivalente a una gran proporción del grid y a pocas celdas sucias, debería esperarse poco tiempo.

A continuación se muestran algunas tablas con las relaciones con el número de agentes:

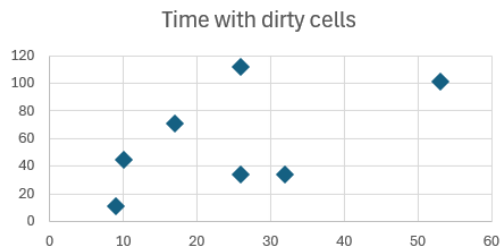


Como puede observarse, no hay una tendencia aparente entre las medidas de los pasos, tiempo y total de movimientos, con el número de agentes. Como ya se había predicho, el comportamiento aleatorio hace que no necesariamente un incremento en el número de agentes implique un menor número de pasos necesarios. No obstante, al probar con el tamaño del grid se obtiene lo siguiente:



Sigue sin existir una relación fuerte, pero es una tendencia más clara que con el número de agentes.

Haciendo lo mismo, pero ahora con el número de celdas sucias arroja lo siguiente:



La tendencia, con excepción de dos casos particulares, es mucho más clara. A mayor proporción de celdas sucias, más tiempo le tomará a los agentes limpiar el grid completo.

En resumidas cuentas, los datos muestran un comportamiento errático por parte de los agentes, pues de ser agentes con planeación o con alguna forma de optimización para mejorar su velocidad de limpieza debería observarse lo siguiente:

1. Si pudieran hablar entre ellos, podrían comunicarse y no ir hacia los mismos lados.
2. Si pudieran revisar su periferia, podrían ir hacia las celdas sucias.

Sin embargo, se observa un comportamiento errático que resulta en una limpieza poco eficiente del grid. Asimismo, al considerar el grid como toroidal, el tener siempre el mismo número de opciones genera la posibilidad de que la roomba oscile entre esquinas sin revisar el centro del grid en numerosas iteraciones. Por ende, un sistema de agentes reactivos en un grid de gran tamaño es bastante ineficiente para generar una limpieza en un tiempo determinado.