



Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

Escuela de Ingeniería y Ciencias

TC2008B.302

Modelación de sistemas multiagentes con gráficas computacionales

M1. Actividad

Alumnos:

Andrés Iván Rodríguez Méndez - A01754650

Arturo Montes González - A01798012

Profesor:

Jorge Adolfo Ramírez Uresti

Fecha de entrega:

09/11/2023

Testeo de tiempo para limpiar todas las casillas

Test	Tamaño de la matriz	Agentes de limpieza	Porcentaje de celdas sucias	Tiempo total
1	100 x 100	10	75%	2.60 segundos
2	200 x 200	50	50%	16.41 segundos
3	150 x 75	35	60%	4.10 segundos
4	500 x 500	100	80%	150.02 segundos
5	200 x 100	150	20%	23.82 segundos

En la anterior tabla podemos ver 5 ejecuciones con distintos tamaños del grid, diferente número de agentes de limpieza y diferente cantidad de basura, mediante estos datos podemos ver el tiempo que tarda en limpiar por completo el grid, analizando los resultados podemos decir que entre más agentes de limpieza tarda mas en limpiar todo el mapa, esto puede ser porque se pueden estorbar de cierta forma porque no pueden estar 2 agente que limpian en la misma celda, mientras que en la primera ejecución que se hizo un tamaño similar de grid pero con una cantidad menor de agentes comprueba que puede limpiar más rápido ya que se distribuyen de mejor manera.

Testeo de número de agentes vs model steps

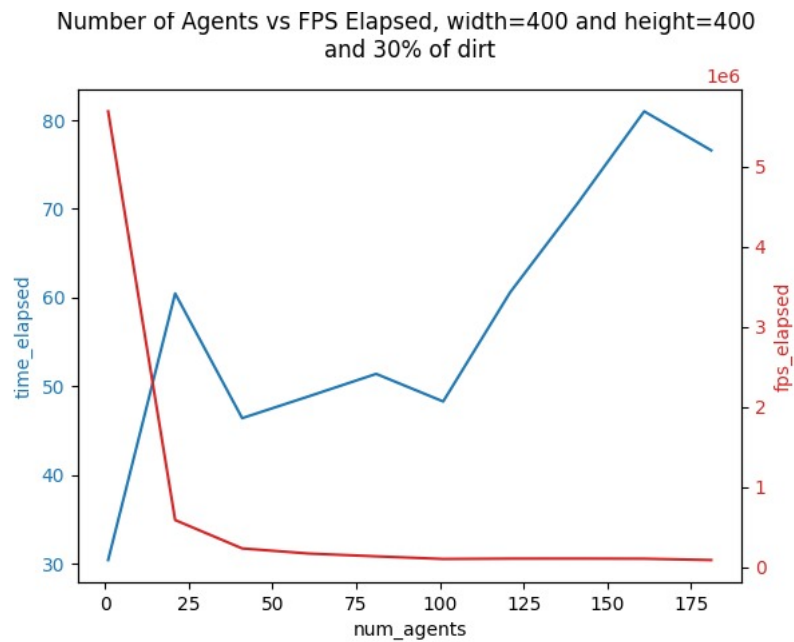


Figura 1. Tiempo de ejecución y fps(pasos) respecto al número de agentes para limpiar todas las celdas.

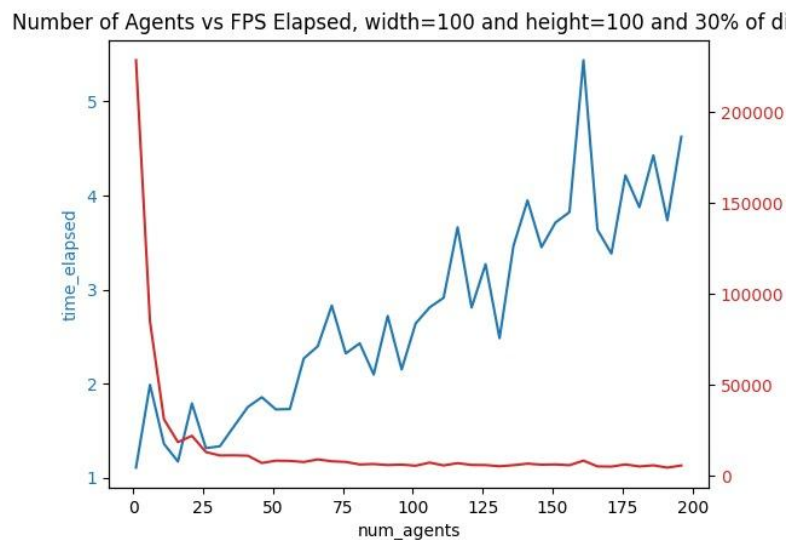


Figura 2. Tiempo de ejecución y fps(pasos) respecto al número de agentes para limpiar todas las celdas.

En la Figura 1 y Figura 2. podemos ver que, de manera antiintuitiva, conforme se aumentan los agentes, el tiempo de ejecución aumenta. Por el contrario, si analizamos el número de agentes respecto a los pasos totales del modelo, podemos notar que mientras más agentes hay, se necesitan menos pasos.

La observación anterior nos puede indicar que usar el tiempo para medir el rendimiento basado en la cantidad de agentes, es un error. Esto posiblemente se debe a que el CPU no puede manejar más agentes con la misma velocidad y, en consecuencia, es mejor usar una variable que no depende del CPU. Por ejemplo, podemos usar el número de pasos para medir el rendimiento de un sistema como el que estamos modelando.

Por otra parte, también podemos notar que hay un punto en el que aunque aumentemos el número de agentes, el rendimiento ya no va a mejorar. Esto es trivial si tomamos el siguiente ejemplo: supongamos que tenemos una red de $n \times m$ dimensiones y digamos que ponemos $n \times m$ agentes. Si los agentes se mueven de forma óptima, empezando en (0, 0), a lo más, les tomará $n \times m$ pasos limpiar todo. Ahora supongamos que tenemos $n \times m + 1$ agentes, podemos ver que esto no va a mejorar el rendimiento de $n \times m$ agentes y, por lo tanto, hay un límite en la mejora del rendimiento basado en el aumento de agentes.

Testeo de pasos para limpiar le grid completo

Test	Tamaño de la matriz	Agentes de limpieza	Porcentaje de celdas sucias	Pasos totales
1	100 x 100	10	75%	333730 pasos
2	200 x 200	50	50%	1893592 pasos
3	150 x 75	35	60%	443533 pasos
4	500 x 500	100	80%	20141259 pasos
5	200 x 100	150	20%	2183116 pasos

Analizando los resultados de la anterior tabla podemos ver que los pasos que dan cada aspiradora pueden variar, ya que si encuentran rápidamente con el método random la basura terminará antes, contemplado esto podemos ver que mientras más grande es el tamaño de grid más pasos tendrán que dar.

Conclusión

A lo largo del desarrollo de este trabajo notamos que es muy importante analizar nuestras variables y su incidencia en lo que queremos modelar, así como también programar bien el movimiento de los agentes y considerar muchos escenarios para que el funcionamiento sea el indicado. En consecuencia, es importante notar las limitaciones del hardware y software, por eso es importante encontrar directrices adecuadas que nos ayudan a analizar exactamente lo que estamos modelando.