



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:

Alan Alcántara Ávila - A01753505

Diego Manjarrez Viveros - A01753486

Carlos Alberto Sánchez Calderón - A01747433

Profesor:

Jorge Adolfo Ramírez Uresti

Fecha de entrega:

09/11/2023

Enlace al repositorio de Github del código

<https://github.com/Dmanjav/Tareas-SMA.git>

Análisis del problema planteado

Información recopilada

- Tiempo necesario hasta que todas las celdas están limpias (o se haya llegado al tiempo máximo)

```
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00100 segundos
La simulación tiene: 2 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 81, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00100 segundos
La simulación tiene: 2 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 51, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00100 segundos
La simulación tiene: 2 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 62, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> █
```

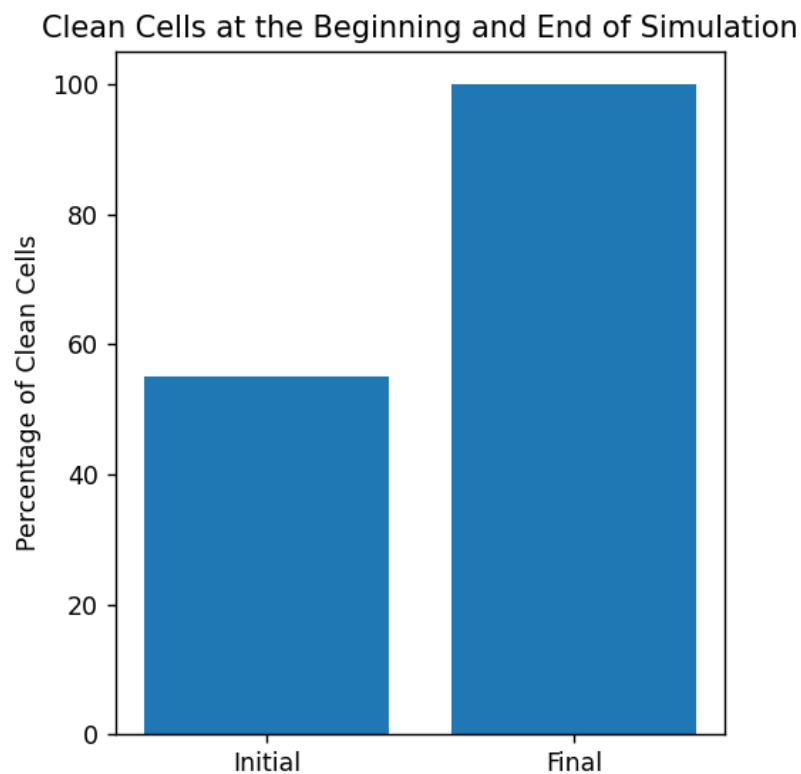
```
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00100 segundos
La simulación tiene: 5 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 29, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00100 segundos
La simulación tiene: 5 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 26, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00100 segundos
La simulación tiene: 5 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 20, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> █
```

```

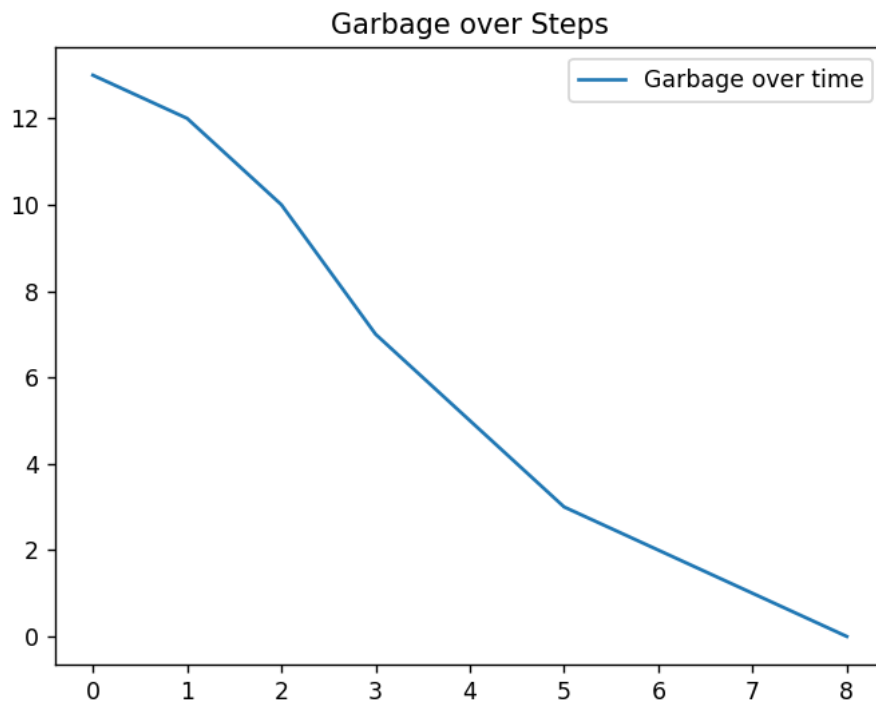
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00200 segundos
La simulación tiene: 55 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 11, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00200 segundos
La simulación tiene: 55 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 7, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad> python3.11 .\prueba.py
La simulación acabó en 0.00200 segundos
La simulación tiene: 55 aspiradoras, 55% de basura y tiene 100 pasos.
La simulacion acabo: framesTotales 9, basura restante: 0
PS C:\Users\mangu\Documents\GitHub\Tareas-SMA\M1 Actividad>

```

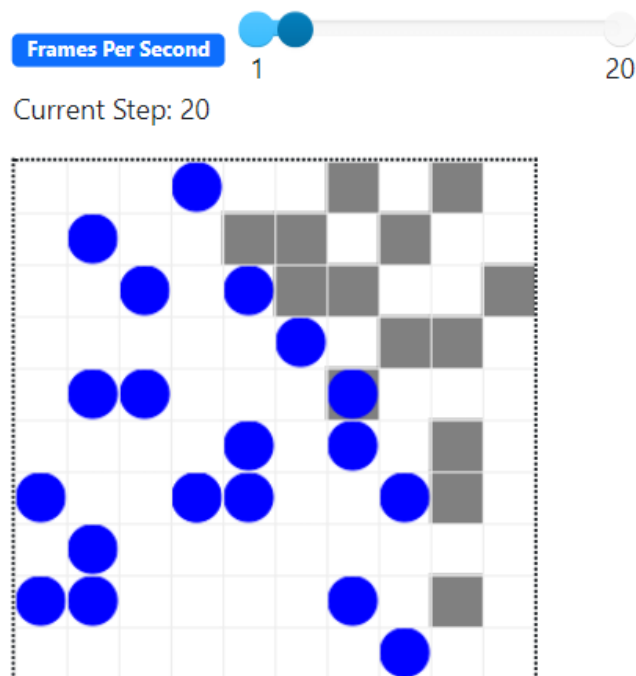
- Porcentaje de celdas limpias después del término de la simulación



- Cantidad de celdas sucias por cada frame



- Representación visual del modelo



- Número de movimientos realizados por todos los agentes

Número de aspiradoras	Steps (promedio)	Número de movimientos
2	64.66	129.32
5	25	125
55	9	495

Conclusiones

Temporalmente, se tuvo que posponer que no se puedan apilar elementos, en este caso aspiradores, debido a que encontrar la solución para eso fue demasiado debido a la curva de aprendizaje de MESA.

Aumentar el número de agentes tiende a reducir el tiempo necesario para limpiar el grid y por ello se alcanza un alto porcentaje de limpieza. Sin embargo, esto no es siempre igual, varía dependiendo del contexto completo, tamaño de grid, porcentaje de basura, etc.

Para mejorar la limpieza se tendría que tener en cuenta la coordinación entre las aspiradoras, para que no recorran los mismos lugares y el tiempo se reduzca mucho. En resumen, el rendimiento del robot de limpieza reactivo está relacionado con el número de agentes y el porcentaje de basura con el que se empieza. Un enfoque bien coordinado con múltiples agentes puede llevar a una limpieza más rápida y completa de la habitación.