



Actividad M1

Rodrigo Gomez Quiroz

A01660379

Campus Ciudad de México

## Introducción

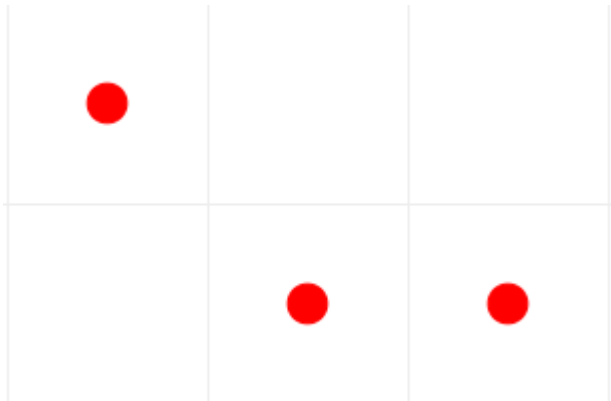
En este documento se describe un reporte de la actividad, en donde se describe el código del resultado logrado, incluyendo aspectos técnicos y estadísticos.

Se nos pidió una simulación en la cual distintos agentes interactúan entre sí, además de que estos mismos reaccionaran o aparecieran en distintas situaciones y posiciones, en la cuales algunos de estos tendrán distintas acciones.

## Código

### Celda sucias/Dirt

Se comenzó con la aproximación más sencilla. El ejercicio nos pide trabajar en un plano, en el cual este tendrá celdas, las cuales estarán divididas en tres categorías, sucias, vacías u ocupadas por un agente. Primero se comenzó en crear las celdas sucias, que a su vez también son agentes, estas aparecerán de forma aleatoria por todo el plano, con ayuda de la librería “*random*”. A cada una de estas se les asignó un único id y también un contador, esto para ayudar tanto al código funcional y las estadísticas a plasmar.



### Agente limpiador/VacumCleaner

A estos también se les categorizó como un tipo de celda, o en otras palabras, una celda ocupada, así mismo se les asignó un id único a cada uno de estos para los mismos fines ya mencionados. La principal diferencia de estos es que aquí se puede seleccionar la cantidad de agentes limpiadores que vamos a ver en la simulación. Estos a su vez pueden moverse libremente en las ocho direcciones posibles, excepto si alguna de las casillas está ocupada. Estos vagan sin rumbo fijo hasta encontrar alguna casilla “sucia”, entonces una vez en ella la limpian y siguen sin camino fijo, puede acabarse el tiempo y no haber encontrado alguna, por el momento todo se maneja de manera aleatoria pero se podría programar que estos limpiadores vayan a la casilla sucia más cercana.



## Modelo

La última parte del código fue establecer el plano en donde todos nuestros agentes estarán interactuando y moviéndose, una vez establecido el modelo se le añadieron las herramientas necesarias para hacerlo correr y poner funcional la simulación.

## Estadísticas

Podemos notar que la simulación solamente dura el tiempo máximo que nosotros le demos, así como podemos notar a continuación.

```
server = ModularServer(VacumModel,
                        [grid, chart],
                        "Vacum Model",
                        {"N": 10, "width": 5, "height": 5, "max_time": 10})
```

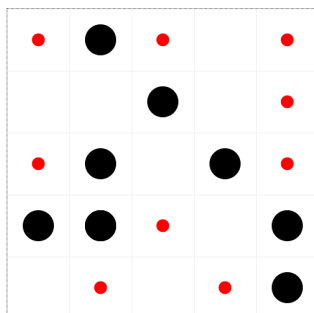
Aquí le damos tanto el número de agente que lo representamos como N y el tiempo máximo.

Estos a su vez son representados a manera de etapas en nuestra terminal de la siguiente manera.

```
{"type": "get_step", "step": 1}
{"type": "get_step", "step": 2}
{"type": "get_step", "step": 3}
{"type": "get_step", "step": 4}
{"type": "get_step", "step": 5}
{"type": "get_step", "step": 6}
{"type": "get_step", "step": 7}
{"type": "get_step", "step": 8}
{"type": "get_step", "step": 9}
{"type": "get_step", "step": 10}
{"type": "get_step", "step": 11}
{"type": "get_step", "step": 12}
```

Puede notarse que aunque le pusimos 10 segundos, aparecen 12 etapas, esto se debe a que se le agregan la etapa de inicio y final.

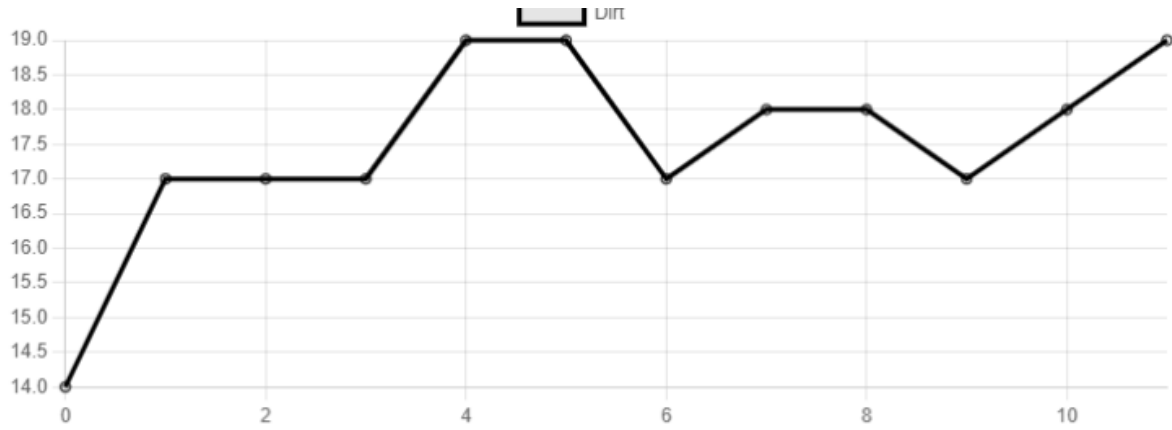
En cada etapa el tablero es bastante diferente esto se debe que todo es aleatorio por lo que aparecen nuevas celdas sucias y los limpiadores siguen vagando sin dirección alguna.



Para plasmar los cambios se puso una gráfica, esta cambia en cada simulación que le apretamos “Start”, pero cambia aún más si modificamos el tiempo y el número de agentes limpiadores

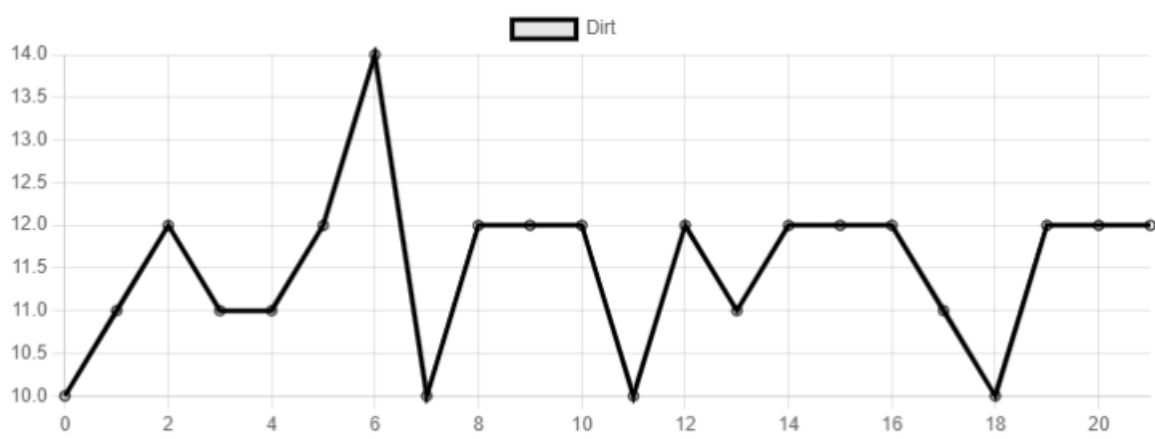
Ejemplo:

Limpiadores: 10 Tiempo:10



Se puede notar que hay actividad entre los agentes ya que los id y las acciones de los limpiadores se reflejan en la gráfica. Ahora modifiquemos los datos.

Limpiadores: 5 Tiempo:20



Efectivamente, los datos recolectados son muy diferentes, ya que al ser menos limpiadores y más tiempo, la gráfica muestra estas fluctuaciones interesantes.

Se puede concluir que en la simulación se plasma una interacción entre agentes en un plano, los cuales son capaces de enviar información a una gráfica, además de ser programados más autónomos y se evitó el menor control posible sobre estos.

<https://github.com/RODGom117/ActividadM1.git>