



**Tecnológico
de Monterrey**

**Instituto Tecnológico y de Estudios Superiores
de Monterrey**

**Modelación de Sistemas Multiagentes con
Gráficas Computacionales (Gpo 104)**

Evidencia 1 Actividad Integradora - Parte 1

Juan Carlos Martínez Zacarías - A01612967

Joaquín Aguirre de la Fuente - A01177479

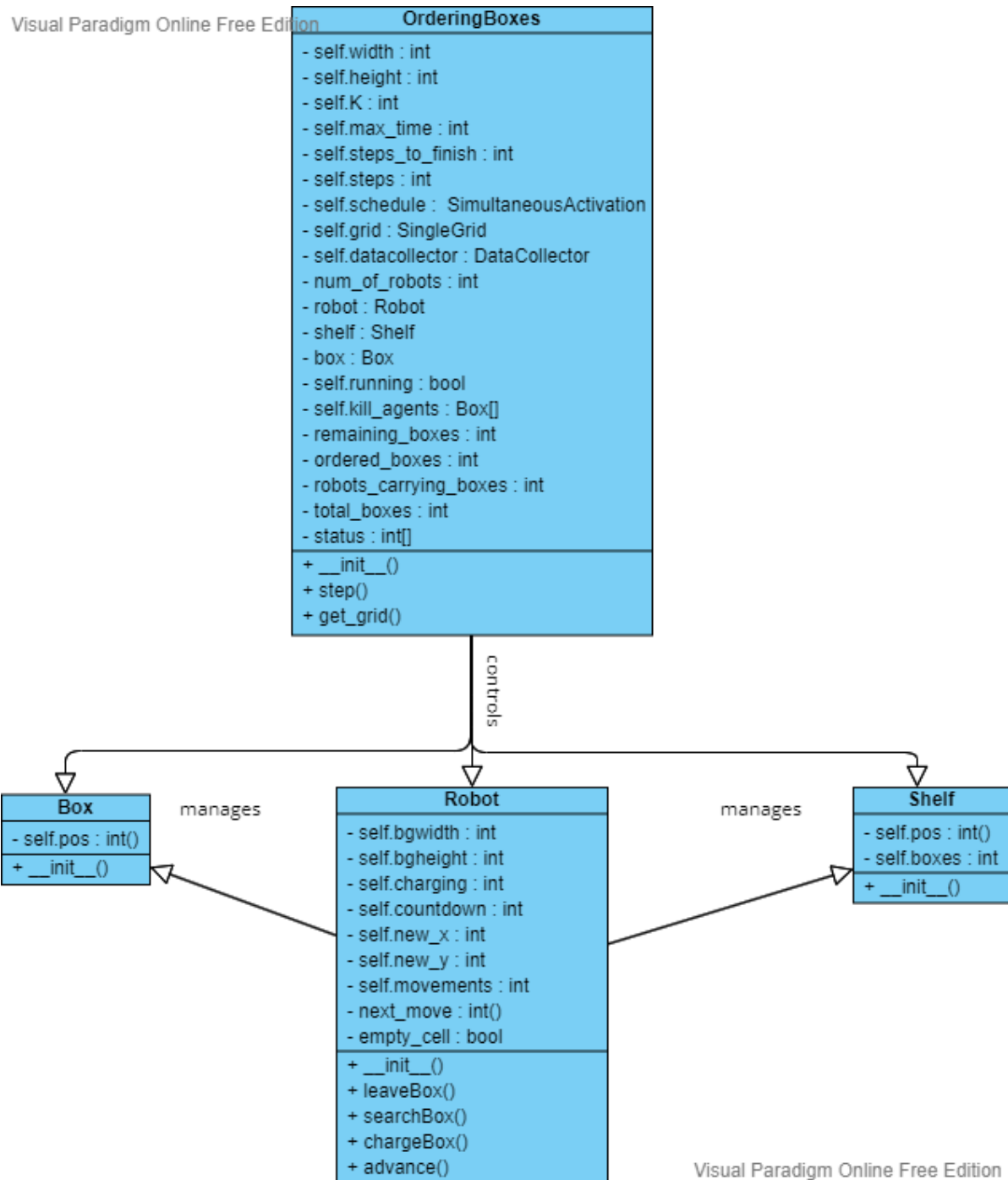
H. Daniel Martínez Rodríguez - A01177464

Sergio Alejandro Esparza González - A01625430

4 de septiembre de 2022

ACTIVIDAD INTEGRADORA - PARTE 1

Diagramas de clases



Protocolo de agentes

Agente “Box” (Caja):

- Cada caja es una instancia del total determinado al inicio del modelado del sistema.
- Cada caja se coloca en una posición aleatoria siempre y cuando no haya otro agente en dicha posición, y tomando en cuenta un rango que abarca todo el largo del espacio y a partir de la tercera fila (índice dos) en tanto a lo alto.

Agente “Shelf” (Estante):

- Cada estante es una instancia del total determinado por la longitud del espacio a utilizar para la construcción del modelo.
- Cada estante es colocado, en orden, a lo largo de la primera fila (índice cero) del espacio de ejecución del modelo.
- Cada estante es inicializado con una cantidad de cajas equivalente a cero.

Agente “Robot”:

- Cada robot es uno de los 5 robots disponibles.
- Cada robot es colocado en una posición aleatoria siempre y cuando no haya otro agente en dicha posición, y tomando en cuenta un rango que abarca todo el largo del espacio y a partir de la segunda fila (índice uno) en tanto a lo alto.
- Cada robot es inicializado con una cantidad de cajas equivalente a cero.
- Si el robot no está cargando una caja, este elige una dirección aleatoria y se mueve hacia ella siempre y cuando no haya agentes de por medio. A partir de este momento, el agente se moverá siguiendo la dirección anteriormente establecida a lo largo de cinco pasos. En caso de que su siguiente movimiento se vea obstruido ya sea por otro agente o por las “paredes” del espacio, el patrón de movimiento se reiniciará, es decir, el robot elegirá una nueva dirección y tratará de moverse en ella durante 5 pasos.
- Cada que el robot realice un movimiento que implique un cambio en su posición, a continuación, examinará su entorno en busca de cajas. Si encuentra una, observará el entorno de esta última esperando que no haya más robots que la puedan tomar. De no haber, la tomará. En el caso contrario, elegirá una posición aleatoria para moverse hacia ella siempre y cuando esté disponible con la finalidad de evitar un error en el que más de un robot tome una misma caja, considerando que todos los agentes llevan a cabo sus funciones al mismo tiempo.

- Si el robot está cargando una caja, este se moverá hacia la izquierda hasta llegar al extremo del espacio. Si hay un agente obstruyendo su camino, el robot se moverá hacia una posición aleatoria siempre y cuando esté vacía y tratará de continuar con la trayectoria anterior. Una vez que llegue al límite izquierdo del espacio, comenzará a descender hacia la zona en la que se encuentran los estantes. Si en su camino hay algún obstáculo, el robot se moverá hacia una posición aleatoria siempre y cuando esté vacía y tratará de continuar con la trayectoria anterior. Al llegar a la zona de estantes, examinará los que se encuentran a su alcance. Si el estante en turno tiene espacio disponible, en él se depositará la caja. Si no, el robot se moverá hacia la derecha hasta encontrar un espacio disponible. Así, en este último patrón de movimiento, si hay un obstáculo, el robot no se moverá para evitar cambiar dicho patrón.

Estrategia cooperativa para la solución del problema

El movimiento que realizan los agentes de tipo robot, que va un paso más allá en tanto a complejidad al tomar en cuenta que una vez que eligen una dirección tratan de seguirla por determinado tiempo en comparación con un simple movimiento aleatorio durante toda la ejecución, forma parte de una estrategia en la que además de abarcar más espacio, también se busca evitar en la medida de lo posible una colisión entre agentes, lo cual se ve reflejado en el cambio de dirección que experimentan los robots una vez que detectan un obstáculo en su camino, si bien existe la excepción que surge cuando un agente de este tipo se encuentra en la búsqueda de un espacio vacío para depositar la caja, de modo que simplemente evita el movimiento y espera a que su camino quede libre, reduciendo así la cantidad de movimientos realizados. Sin embargo, con un rango de entre 600 a 800 pasos requeridos para ordenar las cajas en 5 casos de prueba con 50 cajas iniciales en un espacio de 20 x 20, es factible una mejora en el diseño del sistema en la que, por un lado, los robots elijan una dirección a partir del análisis de su entorno probando distintos radios de manera incremental para que encuentren la correspondiente a la ubicación de la caja más cercana y que, de la misma manera, cambien su estado para que los demás robots no la consideren y se enfoquen en las demás; y, por otro lado, en lugar de reasignar la dirección de movimiento al detectar un obstáculo, los robots podrían tratar de rodearlo para, finalmente, seguir su camino, ya sea el de ordenar una caja o el de buscar alguna.

Todo lo anterior ahorraría movimientos innecesarios y, por ende, haría más eficiente el modelo, si bien elevaría su complejidad.

Consideraciones

Durante la ejecución del modelo se recopila información de manera constante sobre el estado de cada agente de tal manera que sea posible determinar el cumplimiento de su objetivo: acomodar todas las cajas en pilas de no más de cinco instancias. De esta manera, las condiciones determinantes para lo anterior son, por un lado, que no haya cajas desordenadas en el espacio y, por el otro, que el total de cajas ordenadas sea igual a la cantidad de cajas inicializadas al ejecutar el modelo, considerando además la posibilidad de encontrar robots cargando cajas. Sin embargo, al realizar pruebas y llevar a cabo el conteo final, en la gran mayoría de los casos las cajas acomodadas resultan sobrepasar a las inicialmente desordenadas en un rango de 1 a 5 unidades. Creemos que se trata de una situación que surge a partir del hecho de que todos los agentes llaman a sus funciones y actualizan sus variables al mismo tiempo a pesar del protocolo establecido, por lo que, por ejemplo, dos robots detectan una misma caja y ambos la toman, alterando el resultado final esperado. Por este motivo, encontramos una solución práctica al recalcular la cantidad total de cajas durante la ejecución, lo cual permite obtener los datos solicitados sobre el tiempo requerido para la realización de la tarea principal, así como el número total de movimientos de cada agente de manera precisa, pero con base en el valor recalculado, siendo información que se imprime en la terminal.

Enlaces

Modelo en Python:

<https://replit.com/@Juan-CarlosC392/TC2008B-ActIntegradora#agents.py>

Juego en Unity:

<https://drive.google.com/file/d/1pEXG8iTyK8cAgAnyAv9d0U7boqRWsCC/view?usp=sharing>