



INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE
MONTERREY

Ingeniería en Tecnologías Computacionales (ITC)

Modelación de sistemas multiagentes con gráficas
computacionales (Gpo 1)

Actividad Integradora

María José Burguete Euán

A01730344

1 de diciembre del 2021
Puebla, Pue.

Diagrama de clases:

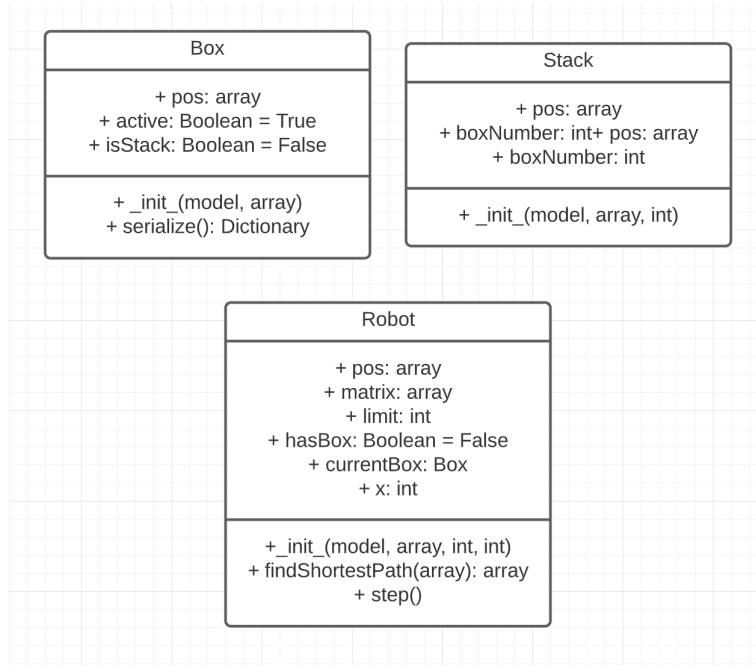
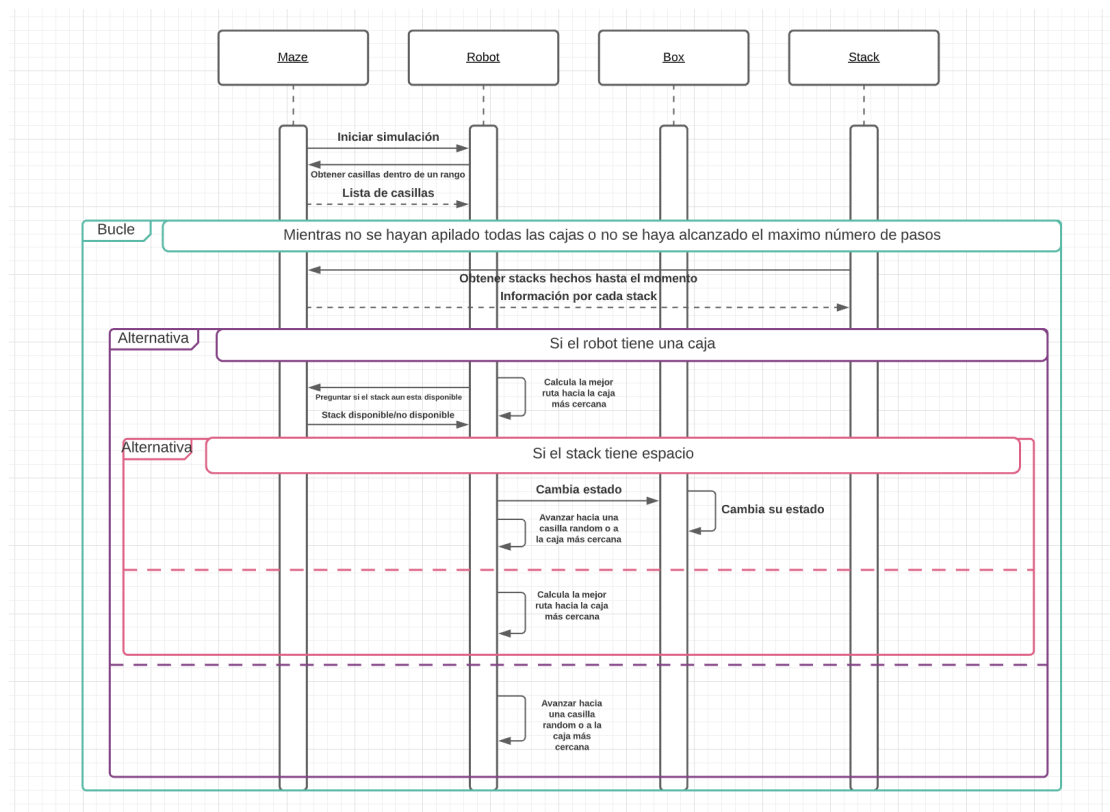


Diagrama de secuencia:



Para la actividad integradora se nos pidió hacer una simulación de robots que apilan cajas en un almacén, contando con 5 robots y N cajas, la tarea consistía en que los robots formarían stacks de 5 cajas máximo de una manera óptima y rápida. Para esto lo primero que hice fue identificar los agentes que necesitaba, los cuales fueron:

- Agente Robot
- Agente Box
- Agente Stack

Estos agentes interactúan entre sí en el modelo llamado Maze. Una vez que supe esto empecé a trabajar en la estructura y la lógica por medio de Mesa. En la primera parte del código creo mis agentes con posiciones aleatorias y los posiciono en el grid del modelo. Los agentes robot van obteniendo una lista de coordenadas alrededor de ellos en un radio de 10 casillas (proporcionada por el modelo) y cada paso van revisando si alguna de esas casillas contiene un agente caja, en caso de que sí, busca la ruta más cercana y empieza a seguirla, en caso de que no, escoge una casilla aleatoria para poder seguir moviéndose y seguir buscando. Otro aspecto a tomar en cuenta es cuando los robots están cargando una caja, estos agentes cuentan con una variable booleana que nos permite conocer si tiene o no una caja, en caso de tener una y encontrar otra casilla con una caja, revisa que el número de cajas colocadas ahí sea menor a 5 y si es así deja la caja y busca otra para poder seguir apilando, en caso de que ya no haya espacio sigue buscando la casilla más cercana con un agente caja (Independientemente de si es o no un stack o si ya no está disponible). En caso de no tener una caja sigue buscando una casilla que contenga un agente caja que pueda levantar, es decir, que no sea un stack. Por otro lado, cada paso de la simulación se va revisando si hay stacks creados y de qué tamaño es cada uno. Se van creando agentes stack por cada casilla que contenga más de dos cajas y se posicionan en la coordenada donde fue encontrado.

Por un lado, mi código le permite a los agentes estar conscientes de los demás agentes existentes sin necesidad de tener una referencia directa desde el principio ya que cada paso va actualizando su entorno, es decir, si hay cajas o robots alrededor. Sin embargo por otro lado considero que no es tan eficiente estar recorriendo todas esas casillas cada paso, por esto mismo otra estrategia que veo posible es pasar una referencia de las posiciones tanto de robots como de cajas desde un inicio, de esta manera, no tendría que iterar por tantas coordenadas ya que ya tendría una idea de donde se encuentra cada cosa en el grid, me pareció un poco directo pero también considero que podría optimizar muchísimo la cantidad de funciones e iteraciones que se hacen en el código.

Código en bitbucket: <https://bitbucket.org/MajoB/actividad-integradora/src/master/>

Código en Github (Unity):

<https://github.com/MajoBurguete/Actividad-Integradora-Unity>