

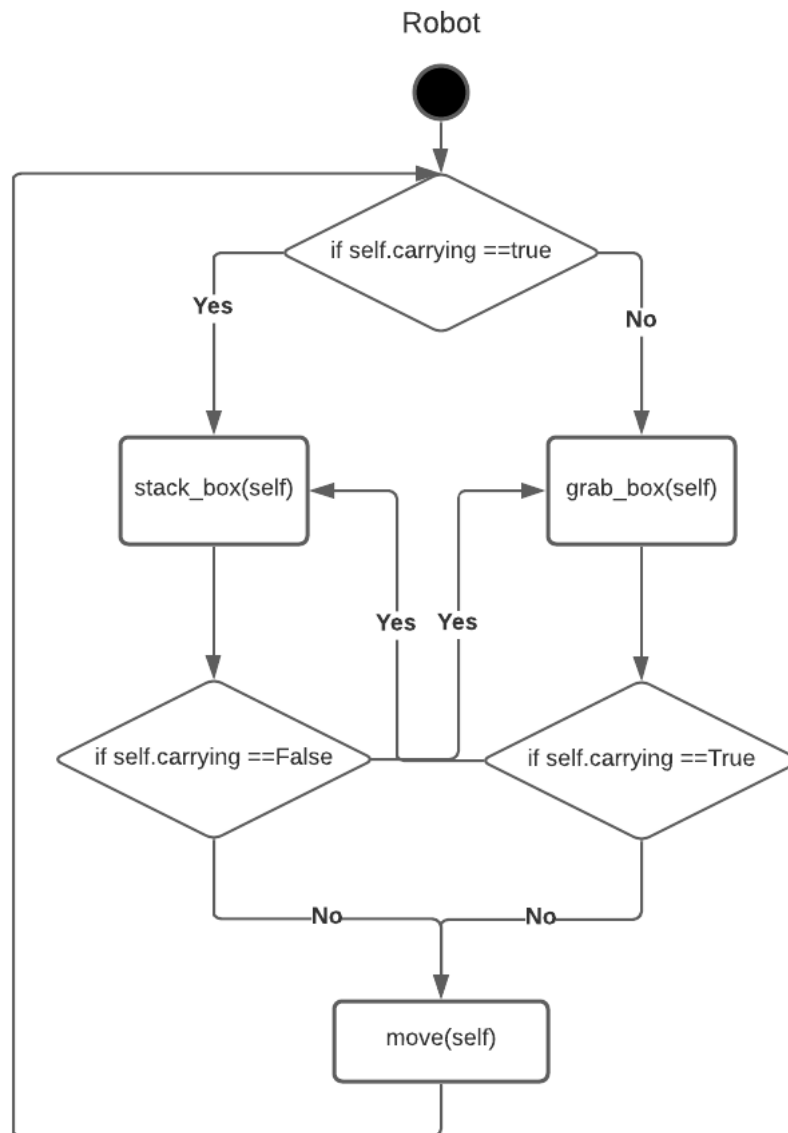
Funcionamiento de la solución:

Las clases que identificamos necesarias para representar a los agentes que intervienen en esta actividad son:

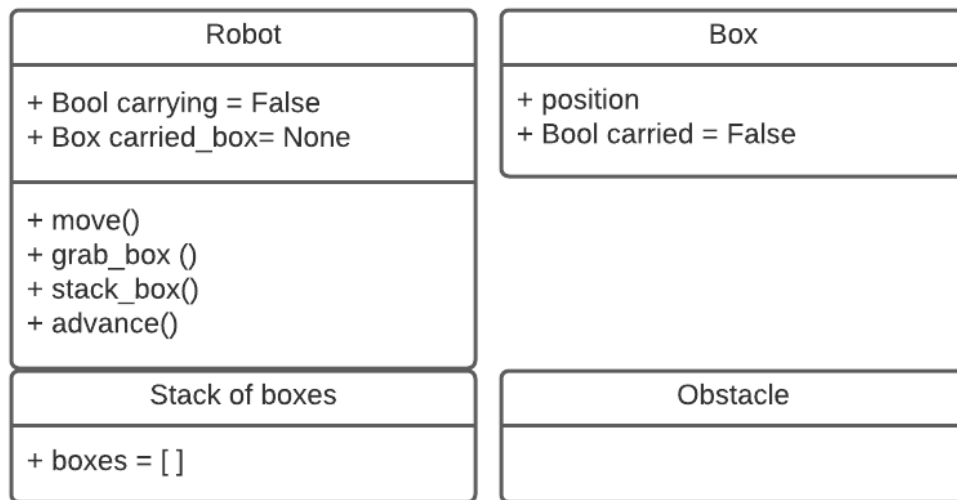
- **Cajas:**
 - Las cajas son representadas por agentes estáticos que aparecen en posiciones aleatorias del espacio. Mismos que deben de ser levantadas por los Robots.
- **Pila de cajas:**
 - Este agente se instancia cuando se coloca una caja sobre otra, en sus atributos sólo tiene un array donde se almacenan las cajas que están apiladas.
 - En la simulación de Mesa, el color del agente cambia dependiendo del número de cajas que tienen en su interior.
- **Obstáculos:**
 - Funcionan como variables para que los objetos no salgan del grid delimitado.
- **Robots:**
 - Los encargados de mover las cajas y, por lo tanto, solucionar la problemática presentada.
 - El protocolo que se implementó para su funcionamiento está basado en dos estados principales (cargando y no cargando). Cada robot es instanciado en modo no cargando, lo que implica que su primera acción es inspeccionar las celdas adyacentes en búsqueda de una caja que pueda recoger, en caso de que la encontrara, se debe mover hasta que pueda localizar una.
 - Para recoger una caja, el robot coloca el agente que la representa en su atributo Box y lo elimina del grid.
 - Cuando localiza y recoge una caja, su atributo cargando se vuelve positivo, por lo que ahora debe encontrar un lugar donde colocarla. Para hacerlo tiene que buscar en las celdas adyacentes si hay un pilar de cajas o una caja sola, priorizando a los pilares. En caso de no encontrar ninguna de las dos opciones se moverá hasta localizar una posición adecuada para apilar su caja.
 - Para colocar una caja dentro de un pilar existente, verifica si el tamaño del pilar es menor a 5, si es así la coloca.
 - En caso de encontrar una caja, la almacena dentro de la función `stack_box()`, elimina su agente del grid e instancia un pilar de

cajas, donde almacena la primer caja y luego la que llevaba cargando.

- A continuación se encuentra un diagrama del protocolo implementado:



Diagramas de clases:



El tiempo necesario para que todas las cajas estuvieran apiladas fue de 33 segundos y se requirieron un total de movimientos.

Creemos que nuestra solución es la más rápida, porque prioriza el acto de apilar las cajas, sin importar el lugar. Si de lo contrario, se buscara que las pilas de cajas se encontraran juntas, sería necesario, indicar a los robots una ubicación específica para que colocaran las cajas. Para lograrlo, también se debería implementar la comunicación necesaria para que los robots se ayudaran entre sí, evitando así quedar encerrados en cajas.

Nuestra implementación prioriza el movimiento de los robots que llevan cajas, por lo mismo cuando un robot que no esté cargando detecta a uno que sí, se detiene hasta que no lo detecte más. Para indicarlo en Unity, pensamos incluir una luz en los agentes que se encendiera cuando cargaran, pero no fue posible por dificultades complejas que no permitió mandar variables desde el modelo en mesa a unity.

Notamos que a veces algunos robots se movían en círculos, lo cual ralentiza el tiempo en el que ordenan cajas.