



**Instituto Tecnológico y de Estudios Superiores de Monterrey,  
Campus Estado de México**

**Escuela de ingeniería y ciencias**

**Modelación de sistemas multiagentes con gráficas  
computacionales**

**Actividad Integradora Parte 2**

**Alumno:**

José Benjamín Ruiz García A01750246

**Profesores:**

**Sergio Ruiz Loza**

**Jorge Adolfo Ramírez Uresti**

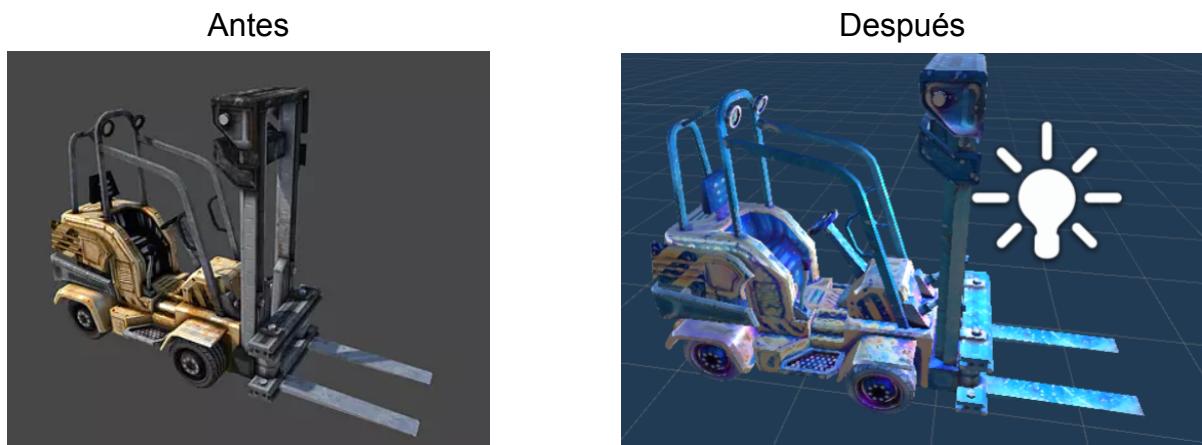
**Fecha:** 23 de noviembre de 2021

## Problemática a resolver

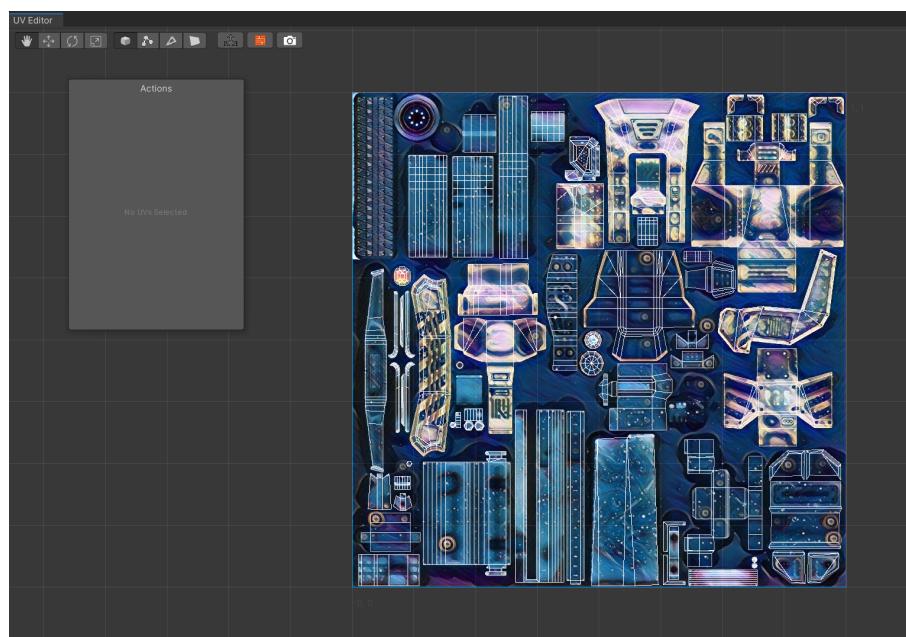
Modelar un sistema de multiagentes en donde los agentes deben acomodar las cajas de un almacén. Los agentes pueden recoger las cajas, moverse, detectar paredes, otros agentes, y cajas. Los agentes pueden hacer pilas de hasta máximo 5 cajas. Los agentes se pueden mover a cualquier celda adyacente siempre y cuando ésta esté vacía.

## Modelos con materiales (colores) y texturas (usando mapeo UV)

Para los modelos se utilizaron assets de la Asset Store de Unity; sin embargo uno de estos assets fue modificado para demostrar las competencias adquiridas en el curso:



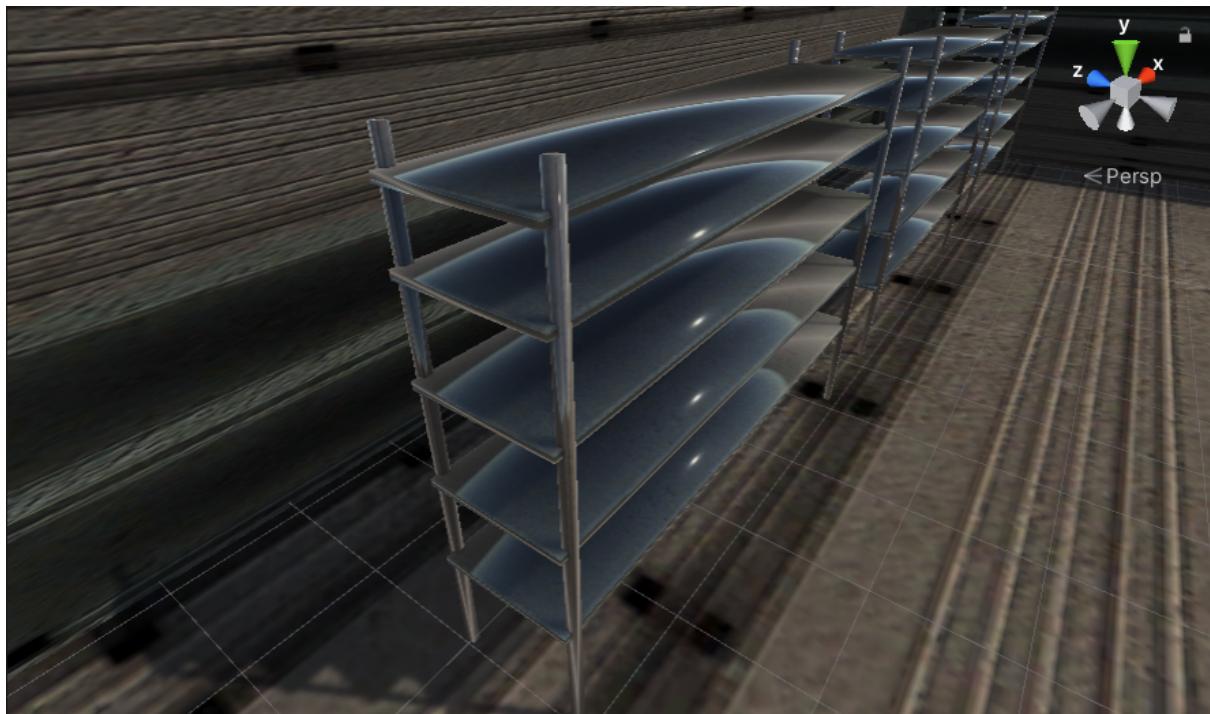
Para lograr esto lo que hice fue modificar las texturas del asset modificando el mapeo UV utilizando la herramienta ProBuilder de Unity:



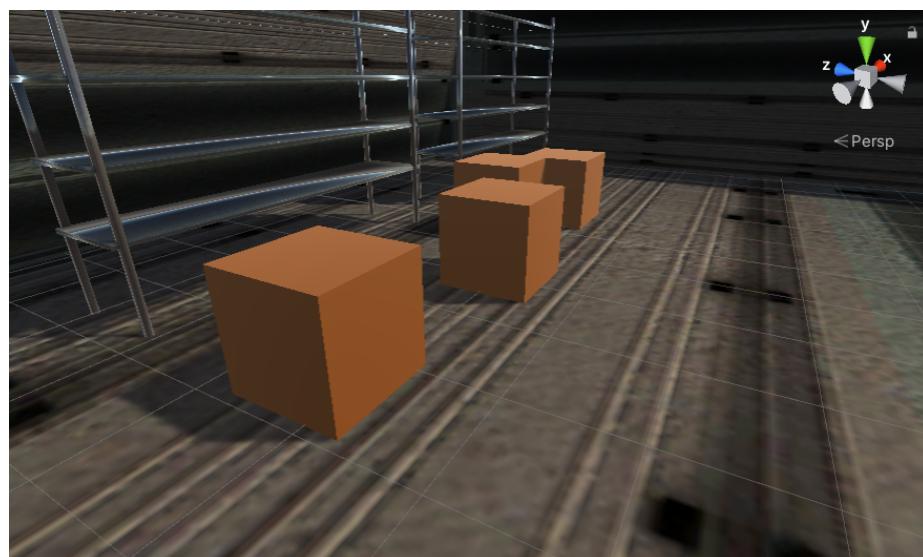
Luego de esto simplemente agregué la textura nueva a un material nuevo en Unity y agregué este material al modelo del montacargas. Con esto ya podemos utilizar el montacargas con la textura y mapeo UV generada por nosotros.

Para los demás elementos (excepto las cajas) se utilizaron assets de unity que ya traen texturas y mapeo UV:

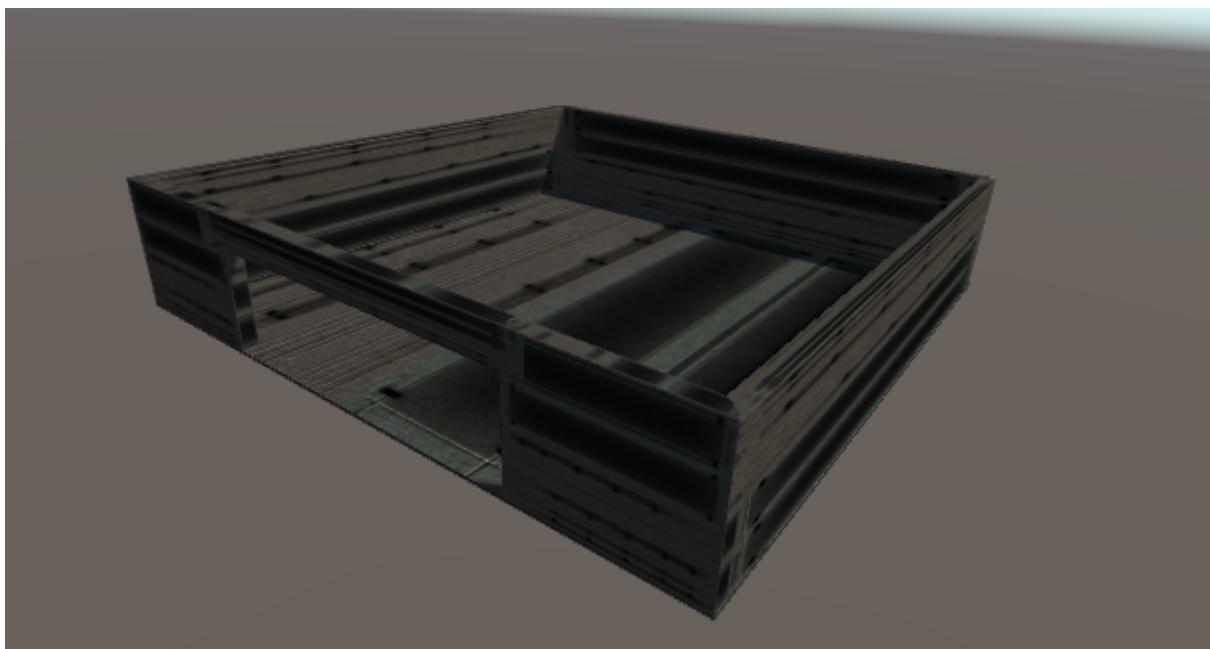
- *Estante*



- *Caja*

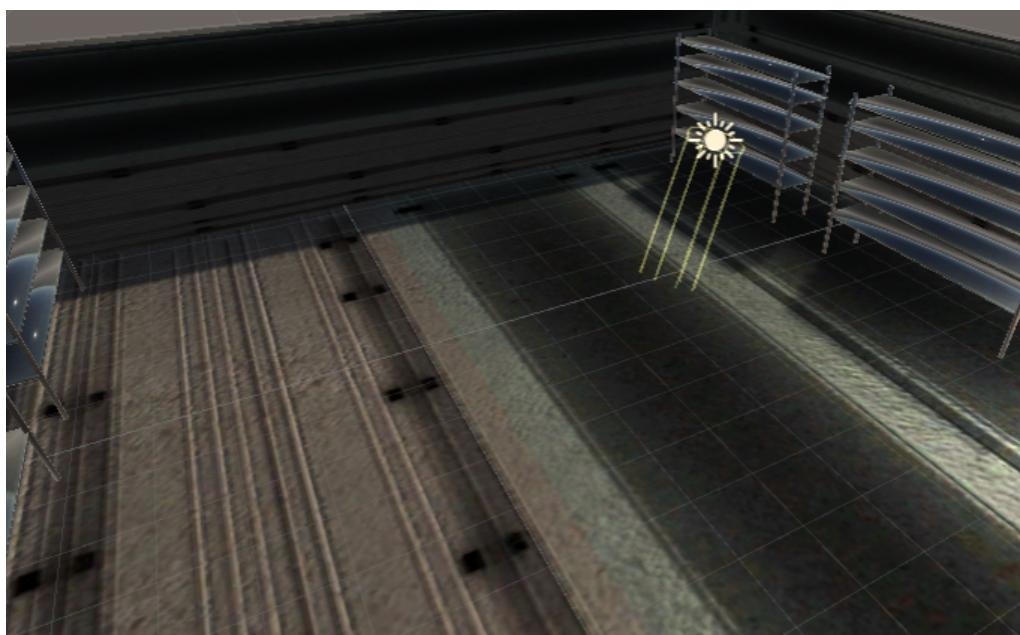


- *Almacén (piso, paredes y puerta)*



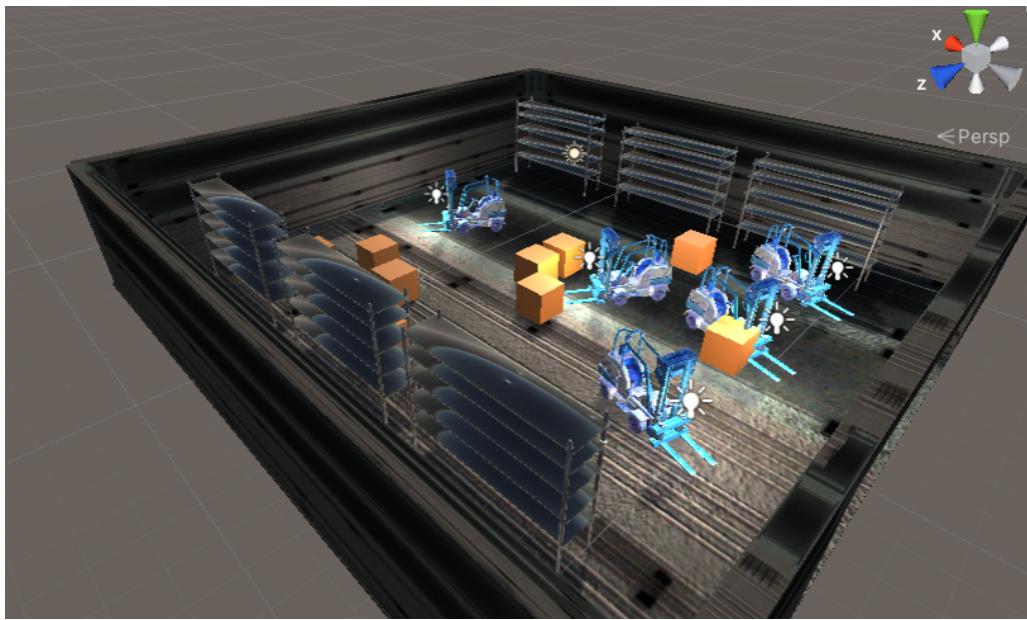
## Illuminación

- *Fuente de luz direccional*



- *Fuente de luz puntual para cada robot que se mueva con ellos*

Para lograr esto simplemente al prefab que se generó para el montacargas se le agregó una fuente de luz puntual con un rango de 10 y una intensidad de 5 para que se aprecie durante la simulación.



## Animación

Para lograr la animación se utilizó un servidor de Python que contiene un sistema de multiagentes que determina la posición en “x” y en “z” de los agentes. Una vez que el programa determina esto nuestro archivo “WebClient.cs” manda una petición de tipo GET al servidor y éste nos contesta con diferentes códigos de respuesta. Si el código es 200 significa que la petición fue exitosa y el servidor nos regresa un JSON que contiene las posiciones de cada uno de los agentes. Este proceso se hace varias veces durante la ejecución, lo que provoca que dé un efecto de movimiento a los montacargas.