Prueba de Aceptación: Sistema de Simulación SilkRoad

Objetivo General

Validar que el sistema SilkRoad gestiona correctamente la colocación de robots y tiendas, el cálculo de ganancias, los movimientos de robots y las restricciones de ubicación, garantizando la integridad de los datos y el cumplimiento de las reglas de negocio establecidas.

Casos de Prueba

1. Creación del Camino (Road)

PASO	DESCRIPCIÓN
1	María, administradora del sistema SilkRoad, necesita crear un nuevo camino para iniciar la simulación crea un camino de longitud de 2 posiciones
2	El sistema genera un error indicando que la longitud mínima del camino debe ser 4.
3	María intenta crear un camino con longitud de 150 posiciones.
4	El sistema genera un error indicando que la longitud máxima permitida es 100.
5	María crea correctamente un camino con longitud de 30 posiciones.
6	El sistema muestra visualmente el camino en espiral con 30 celdas grises válidas y las celdas restantes en negro
7	Tambien se muestra la barra de progreso (ProfitBar) inicializada en 0.

2. Colocación de Robots

PASO	DESCRIPCIÓN
1	María desea colocar un robot en la posición 35 del camino.
2	El sistema genera un error: Ubicación inválida: 35. Debe estar entre 1 y 30.
3	María coloca correctamente un robot Normal en la posición 5.
4	El robot se muestra visualmente en la celda 5 con un color único asignado por el ColorGenerator.
5	María intenta colocar otro robot en la posición 5.
6	El sistema genera un error: La posición 5 ya está ocupada por un robot.
7	María coloca exitosamente un segundo robot en la posición 10.
8	El sistema muestra ambos robots con colores diferentes.
9	María consulta la información de los robots usando robots(), y obtiene [[5,0],[10,0]].

3. Colocación de Tiendas

PASO	DESCRIPCIÓN
1	María desea colocar una tienda Normal en la posición 15 con -10 tenges.
2	El sistema genera un error: La cantidad de tenges debe estar entre 1 y 100.
3	María intenta colocar una tienda con 150 tenges en la posición 15.
4	El sistema genera un error: La cantidad de tenges debe estar entre 1 y 100.
5	María coloca correctamente una tienda Normal en la posición 15 con 50 tenges.
6	La tienda se muestra visualmente con base cuadrada y techo triangular en color único
7	María intenta colocar otra tienda en la posición 5 con 10 tenges (donde hay un robot).
8	El sistema genera un error: La posición 5 ya está ocupada por un robot.
9	María coloca exitosamente una tienda Autonomous en posición 20 con 40 tenges.
10	El método stores() retorna [[15,50],[posición aleatoria,40]].

4. Movimiento de Robots

PASO	DESCRIPCIÓN
1	María desea mover el robot ubicado en la posición 5 30 metros .
2	El sistema genera un error: "Movimiento inválido: el robot saldría del camino".
3	María mueve el robot de la posición 5 10 metros.
4	ProfitBar ahora muestra 40.
5	El robot se mueve visualmente a la celda 15. Como hay una tienda Normal con 50 tenges, el sistema calcula la ganancia: '50 - 10 = 40
6	La ProfitBar se actualiza mostrando 40 tenges de ganancia.
7	La tienda en la posición 15 cambia su color a negro indicando que fue usada y se vacía (tenges = 0).

5. Reinicio del Sistema (Reboot)

PASO	DESCRIPCIÓN
1	Después de este movimientos, María observa que hay una tiendas vacías (en negro) y un robot dispersos por elcamino.
2	María ejecuta el métodoresupplyStores()
3	Todas las tiendas recuperan sus tenges originales y vuelven a sus colores iniciales.
4	María ejecuta returnRobots()).
5	El robot regresa visualmente a sus posicion inicial de colocación.
6	María ejecuta (reboot()) que combina ambas acciones anteriores.
7	El sistema queda completamente reiniciado con tiendas llenas y robots en posiciones originales.
8	La ProfitBar mantiene las ganancias acumuladas.

6. Eliminación de Elementos

PASO	DESCRIPCIÓN
1	María desea eliminar el robot ubicado en la posición 25.
2	El sistema genera un error: "No hay ningun robot en la posición 25".
3	María elimina correctamente el robot de la posición 5.
4	El robot desaparece visualmente del camino.
5	La ProfitBar recalcula el máximo posible sin ese robot.
6	María intenta eliminar una tienda inexistente en la posición 7.
7	El sistema genera un error: "No hay ninguna tienda en la posición 7".
8	María elimina correctamente la tienda de la posición 15.
9	La tienda desaparece visualmente.

7. Movimiento Automático (moveRobots)

PASO	DESCRIPCIÓN
1	María tiene un robot en posiciones [10] y una tienda autonoma en la posicion [aleatorea] con tenges [40].
2	María ejecuta el método (moveRobots()).
3	El sistema calcula automáticamente la mejor tienda para cada robot maximizando (tenges - distancia).
7	María consulta (profitPerMove())
8	emptiedStores() = [[5,1],[12,1],[20,1]].

8. Simulación con SilkRoadContest

PASO	DESCRIPCIÓN
1	María prepara un arreglo de días: days = [[1, 5], [2, 10, 40], [1, 8], [2, 15, 60]]).
2	Día 1: coloca robot en posición 5.
3	Dia 2: coloca tienda en posición 10 con 40 tenges.
4	Día 3: coloca robot en posición 8.
5	Día 4: coloca tienda en posición 15 con 60 tenges.
6	María ejecuta solve(days)) sin visualización.
7	El sistema retorna:[5, 5, 5, 10] indicando: robot en 5 se mueve 5 metros, robot en 5 se mueve 10 metros.
8	María ejecutasimulate(days, false)con visualización rápida (1 segundo por día).
9	El sistema muestra animadamente cada día: colocación de elementos y movimientos automáticos.
10	Al finalizar, el sistema ejecutafinish()y muestra mensaje: " La simulación ha terminado".

9. Finalización y Bloqueo

PASO	DESCRIPCIÓN
1	María ejecuta finish()) en el sistema SilkRoad.
2	Aparece un mensaje: "La simulación ha terminado. Ya no se pueden realizar mas acciones".
4	María intenta colocar un nuevo robot.
5	El sistema genera un error: "La simulacion ya ha terminado. No se puede colocar un robot".
6	María intenta mover un robot existente.
7	El sistema genera un error: "La simulacion ya ha terminado. No se puede mover un robot".
8	María consulta (ok()) y obtiene false confirmando que el sistema está finalizado.

Resumen de Validaciones Implementadas

✔ Validación de longitud del camino (4-100)
✔ Validación de posiciones
✔ Validación de tenges (1-100)
✔ Validación de ocupación
✔ Validación de movimientos fuera del camino
✔ Validación de elementos inexistentes
✔ Cálculo correcto de ganancias
✔ Gestión de ProfitBar
✔ Generación de colores únicos
✔ Retroalimentación visual y animaciones

Conclusiones

El sistema SilkRoad cumple satisfactoriamente con todos los requisitos funcionales y de validación establecidos. Las reglas se aplican correctamente, las restricciones se validan apropiadamente y la interfaz visual proporciona una experiencia clara para el usuario.