

Este script simula un mercado donde diferentes tipos de agentes (que representan participantes en el mercado) compran y venden cartas según ciertas estrategias. La simulación sigue una serie de iteraciones, en cada una de las cuales los agentes toman decisiones que afectan el mercado y sus propios balances e inventarios.

Clases Principales

1. **Market:**

- Representa el mercado donde los agentes compran y venden cartas.
- El precio de las cartas fluctúa según las transacciones realizadas por los agentes.
- También tiene un stock de cartas disponibles para vender.

Atributos:

price: El precio de las cartas en el mercado.

stock: El número de cartas disponibles para la venta en el mercado.

Métodos:

buy(): Cuando un agente compra una carta, el stock de cartas del mercado disminuye y el precio sube un 0.5%.

sell(): Cuando un agente vende una carta, el stock de cartas aumenta y el precio baja un 0.5%.

Lógica:

- El precio y el stock de cartas en el mercado dependen de las transacciones realizadas por los agentes.

2. **Agent (abstracta):**

- Es la clase base para los diferentes tipos de agentes.
- Cada agente tiene un balance (dinero disponible) y un inventory (inventario de cartas).
- Tiene un método abstracto act(), que define la lógica de compra/venta de cada tipo de agente.

Atributos:

balance: La cantidad de dinero disponible para comprar cartas.

inventory: El número de cartas que el agente posee.

Métodos:

act(): Método abstracto que debe ser implementado por las subclases, donde cada agente define su comportamiento.

3. **RandomAgent:**

- Un tipo de agente que toma decisiones de compra, venta o no acción de manera aleatoria.

Método act():

Elige aleatoriamente entre comprar, vender o no hacer nada.

Si decide comprar y tiene suficiente dinero, aumenta su inventario y paga el precio actual del mercado.

Si decide vender y tiene cartas en su inventario, disminuye su inventario y recibe el precio actual del mercado.

4. **TrendFollowingAgent:**

- Un agente que sigue las tendencias del mercado.
- Compra cuando el precio sube y vende cuando el precio baja.

Método act():

Si el precio está subiendo, compra cartas.

Si el precio está bajando, vende cartas.

Si el precio se mantiene estable o no hay suficiente dinero, no hace ninguna acción.

5. **AntiTrendAgent:**

- Un agente que actúa en contra de la tendencia del mercado.
- Compra cuando el precio baja y vende cuando el precio sube.

Método act():

Si el precio está subiendo, vende cartas.

Si el precio está bajando, compra cartas.

Si el precio se mantiene estable o no tiene suficiente dinero o cartas, no hace nada.

6. **CustomAgent:**

- Un agente con un comportamiento mixto basado en el número de iteraciones restantes en la simulación.

Método act():

En las primeras iteraciones, actúa de manera similar a un TrendFollowingAgent.

En las últimas iteraciones, cambia su comportamiento para tomar decisiones más conservadoras.

7. **Simulation:**

- Esta clase se encarga de gestionar y ejecutar la simulación completa.
- Crea los agentes, gestiona las interacciones con el mercado y controla la ejecución de cada iteración.

Atributos:

iterations: El número de iteraciones o pasos que tomará la simulación.

agents: Una lista de todos los agentes que participan en la simulación.

Métodos:

create_agents(): Crea instancias de los diferentes tipos de agentes (random, trend-following, anti-trend, y custom).

run(): Ejecuta la simulación, en cada iteración hace que los agentes actúen y actualiza el estado del mercado.

Flujo General del Script:

1. Creación de agentes: La clase Simulation crea una lista de agentes con diferentes comportamientos (RandomAgent, TrendFollowingAgent, AntiTrendAgent, CustomAgent).
2. Ejecución de la simulación: La simulación se ejecuta durante un número determinado de iteraciones. En cada iteración, cada agente toma una acción (comprar, vender o no hacer nada) en función de su estrategia y el estado del mercado.
3. Interacción con el mercado: Durante cada iteración, los agentes compran o venden cartas, lo que afecta el precio de las cartas en el mercado. El precio sube un 0.5% cada vez que un agente compra una carta y baja un 0.5% cuando un agente vende una carta.
4. Resultados finales: Después de todas las iteraciones, se imprime el balance y el inventario final de cada agente, el precio final de las cartas y el stock restante en el mercado.

Resumen de la Lógica del Precio

- Precio inicial: El precio de las cartas comienza en \$200.
- Cambios en el precio:
 - Cuando un agente compra una carta, el precio sube un 0.5%.
 - Cuando un agente vende una carta, el precio baja un 0.5%.

Resumen de la Lógica del Inventario de los Agentes

- Inventario inicial: El inventario de cartas de cada agente comienza en 0.
- Cambios en el inventario:
 - Cuando un agente compra una carta, su inventario aumenta en 1.
 - Cuando un agente vende una carta, su inventario disminuye en 1.