

Simulación Cooperativa del Juego de Caramelos-Superviviente

Yonhel Mamani Cruz
Universidad Nacional del Altiplano de Puno

Julio 2025

Índice

1. Resumen	2
2. Introducción	2
3. Descripción del Juego	2
4. Planteamiento del Problema	2
5. Procedimientos y Desarrollo	3
5.1. Lógica de Grupos Incompletos	3
5.2. Entradas del Usuario	3
5.3. Fragmento de Código Utilizado	3
6. Resultados	3
7. Evidencias del Procedimiento	3
8. Conclusiones	6

1. Resumen

Se presenta una simulación cooperativa del juego de caramelos y chupetines. El juego fue implementado en Python utilizando la biblioteca gráfica Tkinter. Cada jugador inicia con caramelos aleatorios, y el objetivo es conseguir al menos un chupetín por persona. Se implementó una lógica cooperativa para asegurar que todos los participantes logren el objetivo, incluso en casos donde existan grupos incompletos o jugadores individuales.

2. Introducción

El juego de caramelos y chupetines fue diseñado con el objetivo de fomentar el pensamiento lógico, la estrategia cooperativa y el trabajo en equipo. Esta dinámica lúdica ha sido transformada en un programa interactivo que simula el proceso de distribución y combinación de caramelos. El sistema promueve la colaboración entre jugadores, permitiendo asegurar que todos obtengan al menos un chupetín, incluso cuando la formación de grupos sea desigual.

3. Descripción del Juego

Cada jugador recibe inicialmente dos caramelos, seleccionados aleatoriamente entre los siguientes tipos:

- Limón
- Huevo
- Pera

Para formar un chupetín, se requiere una combinación específica:

- 2 caramelos de limón
- 2 caramelos de huevo
- 2 caramelos de pera

Una vez formado un chupetín, el jugador recibe dos caramelos adicionales. Existe también la posibilidad de vender un chupetín a cambio de seis caramelos nuevos, lo cual puede ayudar en momentos estratégicos del juego.

4. Planteamiento del Problema

Durante la ejecución del juego, surgen varios desafíos:

- Asegurar que todos los jugadores obtengan al menos un chupetín.
- Manejo de grupos incompletos o jugadores individuales.
- Redistribución de caramelos para lograr combinaciones viables.

5. Procedimientos y Desarrollo

5.1. Lógica de Grupos Incompletos

En el juego pueden existir grupos que no tengan el número ideal de integrantes. Las estrategias aplicadas son:

1. Los jugadores individuales pueden integrarse temporalmente a otros grupos.
2. Los grupos incompletos pueden intercambiar caramelos con otros grupos.
3. Se garantiza equidad mediante reglas cooperativas programadas.

5.2. Entradas del Usuario

Al iniciar el programa, se solicitan dos datos fundamentales:

- Número de jugadores
- Número de integrantes por grupo

5.3. Fragmento de Código Utilizado

Listing 1: Validación para formar un chupetín

```
def can_make_chupetin(inv):  
    return inv['limon'] >= 2 and \  
           inv['huevo'] >= 2 and \  
           inv['pera'] >= 2
```

6. Resultados

A continuación se presentan los resultados obtenidos durante una ejecución de prueba:

- Se simularon 10 jugadores.
- Todos lograron formar al menos un chupetín.
- Se vendió un chupetín para obtener caramelos adicionales.
- Un grupo fue formado por un solo jugador, quien se integró temporalmente a otro grupo.

7. Evidencias del Procedimiento

A continuación, se incluyen capturas de pantalla como evidencia del desarrollo y ejecución del programa:

🍬 Juego de Caramelos y Chupetines 🍬

Número de personas: Tamaño por grupo:

Figura 1: Inicio de la simulación - ingreso de datos

```

▶ Total de personas: 10, tamaño de grupo: 3
⚠ El grupo 4 está incompleto con 1 personas.
📦 Se usará fondo común de caramelos para que todos tengan la misma oportunidad.

🎲 Reparto inicial:
Persona 1: ['limon', 'pera']
Persona 2: ['huevo', 'huevo']
Persona 3: ['huevo', 'limon']
Persona 4: ['pera', 'huevo']
Persona 5: ['huevo', 'huevo']
Persona 6: ['pera', 'limon']
Persona 7: ['huevo', 'huevo']
Persona 8: ['limon', 'limon']
Persona 9: ['limon', 'huevo']
Persona 10: ['pera', 'pera']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['pera', 'limon']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['pera', 'huevo']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'pera']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'limon']
Se vendió 1 chupetín para recibir 6 caramelos: ['pera', 'pera', 'limon', 'huevo', 'limon',
'limon']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['pera', 'pera']
Se vendió 1 chupetín para recibir 6 caramelos: ['huevo', 'huevo', 'huevo', 'limon',
'huevo', 'pera']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['pera', 'huevo']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'limon']
Se vendió 1 chupetín para recibir 6 caramelos: ['pera', 'pera', 'pera', 'huevo', 'limon',

```

Figura 2: Visualización de caramelos iniciales por jugador

```

Se vendió 1 chupetín para recibir 6 caramelos: ['pera', 'pera', 'pera', 'pera', 'pera',
'huevo']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'limon']
Se vendió 1 chupetín para recibir 6 caramelos: ['huevo', 'huevo', 'pera', 'huevo',
'limon', 'pera']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'limon']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['huevo', 'huevo']
Se vendió 1 chupetín para recibir 6 caramelos: ['limon', 'pera', 'huevo', 'pera', 'huevo',
'limon']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'pera']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'limon']
Se vendió 1 chupetín para recibir 6 caramelos: ['huevo', 'huevo', 'pera', 'pera', 'huevo',
'pera']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'limon']
Se vendió 1 chupetín para recibir 6 caramelos: ['huevo', 'pera', 'pera', 'huevo', 'limon',
'pera']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'huevo']
Se hizo 1 chupetín (usando 2 limón, 2 huevo, 2 pera) y se recibieron 2 caramelos extra:
['limon', 'limon']

📦 Inventario final de caramelos:
{'limon': 2, 'pera': 0, 'huevo': 0}
👤 Chupetines totales: 10
📦 Intercambios realizados: 11

🌟 Resultado final:
☒ ¡Objetivo logrado! Todos tienen al menos 1 chupetín.

```

Figura 3: Proceso de combinación y creación de chupetines

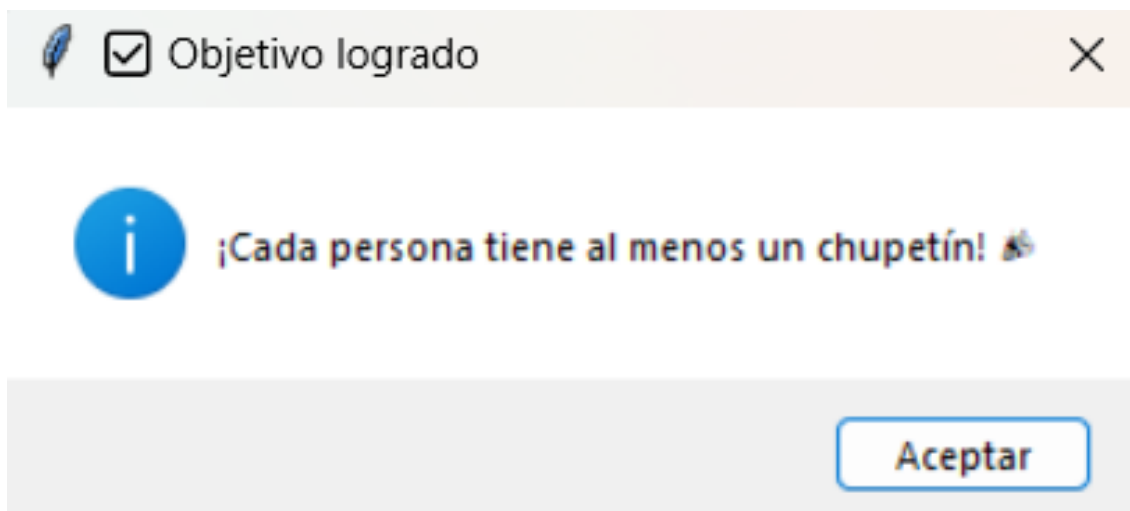


Figura 4: Resultado final con todos los jugadores ganando

8. Conclusiones

La simulación desarrollada del juego de caramelos y chupetines permitió abordar un problema de asignación y cooperación desde una perspectiva de optimización discreta. Se implementaron estrategias que buscan maximizar la obtención de chupetines por parte de los jugadores, bajo restricciones de recursos limitados (caramelos) y combinaciones específicas.

La lógica del sistema garantizó soluciones factibles incluso en escenarios con grupos incompletos o jugadores individuales, demostrando la eficiencia del algoritmo en la redistribución equitativa de recursos. Asimismo, se logró minimizar el desperdicio de caramelos mediante intercambios y decisiones cooperativas, lo cual refleja una optimización del uso de insumos.

En conjunto, el simulador constituye una aproximación lúdica a problemas clásicos de optimización combinatoria y asignación de recursos, con aplicaciones potenciales en el análisis de estrategias colaborativas, juegos serios y educación computacional.