ACKs & Exceptions

by

Particular Software

Overview

ACKs & Exceptions is a bluffing and deduction game for **4 players**.
Players enqueue hidden coasters representing messages that either succeed their processing attempt (**ACKs**) or cause failures (**Exceptions**), then issue challenges to prove how many messages can be successfully processed **without triggering an exception**.

The game continues until a player **wins two rounds**.

Components (per player)

Each player has **4 coasters**:

- **3 ACKs** (successful operations)
- **1 Exception**, labeled as one of:
    - **NullReferenceException**
    - **HTTP 500**
    - **SQL 1205**
    - **Timeout**

ACK
NullReference Exception
Timeout
HTTP 500
SQL 1205

Objective

Win **two rounds** by successfully completing challenges.

Setup

1. Each player takes their **4 coasters** into hand.

2. Choose a starting player.
3. Every player places **one coaster face down** in front of them (their stack).

Gameplay

1. Preparing Messages

Starting with the first player and proceeding clockwise:

- On your turn, **either**:
    - Put **one additional coaster face down** into your stack, **or**
    - Initiate a **challenge**

Players may build stacks of multiple face-down coasters.

---

2. Issuing a Challenge (Bidding Phase)

When a player initiates a challenge, they announce:

> **“I can process X messages without triggering an exception.”**

Then, clockwise, each player must:

- **Increase the number** (raise the bid), or
- **Pass**

Once a player passes, they are out of the bidding.

The player with the **highest bid** becomes the **Challenger**.

---

3. Processing

The Challenger now attempts to process the messages/coasters:

1. Each stack is turned into a queue: `var q = new Queue<Result>(stack)`
2. **First**, they must process their own queue: flip **all of their own coasters**, one by one.
3. If no exception is thrown, they may flip coasters from **any other players**, one at a time, in any order.
4. The attempt ends when:
    - The announced number of coasters is reached → **Success**
    - An Exception is thrown → **Failure**

Rules to remember:

- Coasters are flipped **one at a time**
- You may not skip your own queue
- Order of other players’ queue is your choice

Outcomes

✅ Successful Processing

- No exception was thrown.
- The Challenger **wins the round**.
- If this is their **second round win**, they **win the game**.
- Otherwise, a new round begins.

❌ Exception Thrown

- An Exception was thrown.
- The Challenger **loses one coaster** (chosen secretly from their remaining ones).
- That coaster is removed from the game.
- A new round begins.

If a player loses their **last coaster**, they are **SIGTERMed**.

Winning the Game

The game ends immediately when:

- A player **wins two rounds**, or
- Only one player remains active (all others eliminated)

Engineering Notes

- Bluff aggressively — confidence sells bad deployments.
- Early Exceptions can be powerful misdirection.
- Overbidding is risky, but hesitation is readable.
- Trust nothing. Especially ACKs.