High, Low, Jack, & Claude: AI-Assisted Shiny Development

Jasmine Daly, Principal Consultant
& Founder, Daly Analytics

rainbowR Conference 2026

The Game

Setback is a partnership trick-taking card game

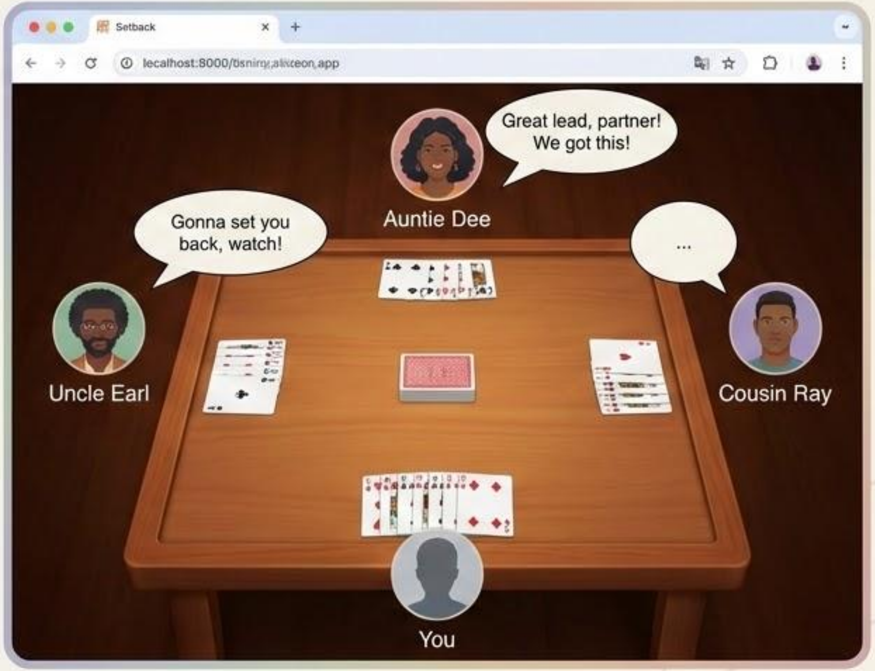
- 2v2 teams, 6 tricks per hand, 4 points up for grabs each hand
- You bid for the right to name the suit, then play it out
- Miss your bid? You get set back — you lose points
- Every hand comes down to partnership: reading your partner's signals, knowing when to lead, knowing when to get out of the way

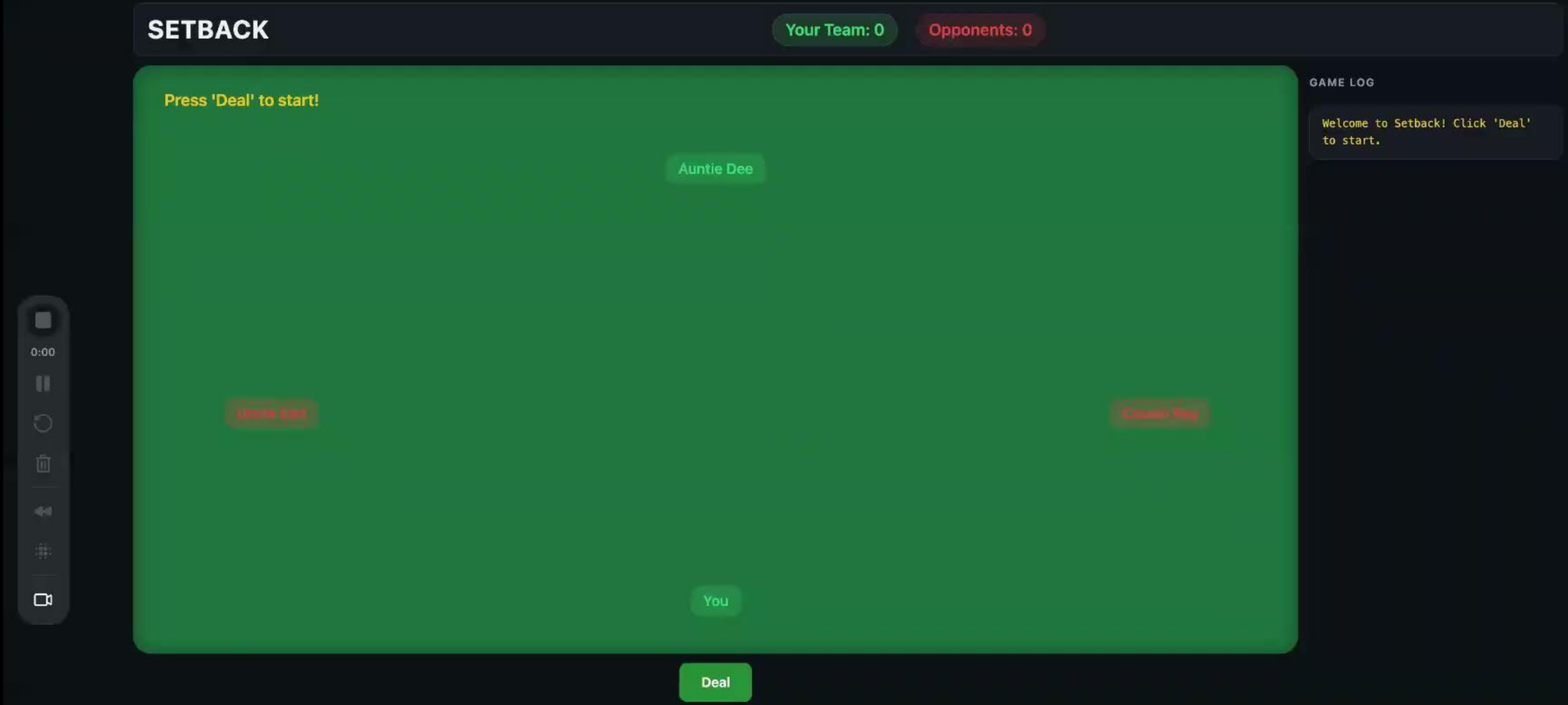
Sound familiar? That’s basically what it’s like building with an AI.

The Project

I used Claude Code to bring this game from the dining table to the browser

- R Shiny app, 3 AI opponents with family cookout personalities
- Uncle Earl (trash-talker), Auntie Dee (your hype partner), Cousin Ray (quiet assassin)
- Built through conversation — describe what you want, the AI writes it
- What follows is the story of that collaboration, told through the four points of Setback





Press 'Deal' to start!

Welcome to Setback! Click 'Deal' to start.

High

HIGH: The best card on the table

In Setback:
The High point goes to whoever held the highest card in the suit

In AI collaboration:
The ‘high’ moments are when the AI nails something you’d have spent hours on.

Planning Mode

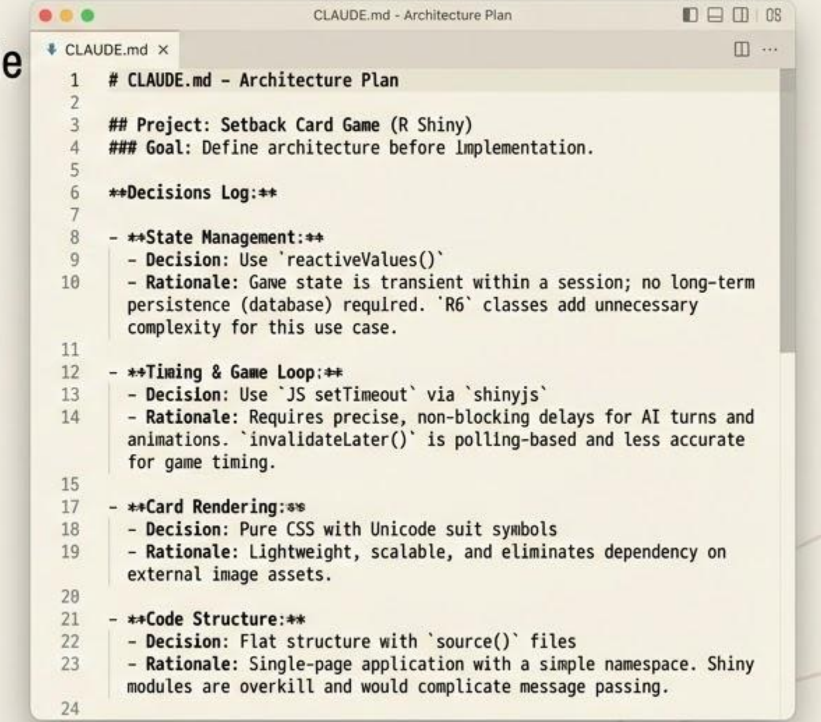
The AI was strongest when I knew exactly what I wanted

Before a single line of code: Claude Code's plan mode

Interactive Q&A to lock down every decision:

- ⚙️ `reactiveValues()` not R6 — no persistence needed
- 🕒 JS `setTimeout` not `invalidateLater()` — precise AI timing, no polling
- 🂡 Pure CSS cards with Unicode suits — no image dependencies
- 📂 `Source files` not Shiny modules — single page, simple namespace

The planning conversation took longer than the building. That was the point.

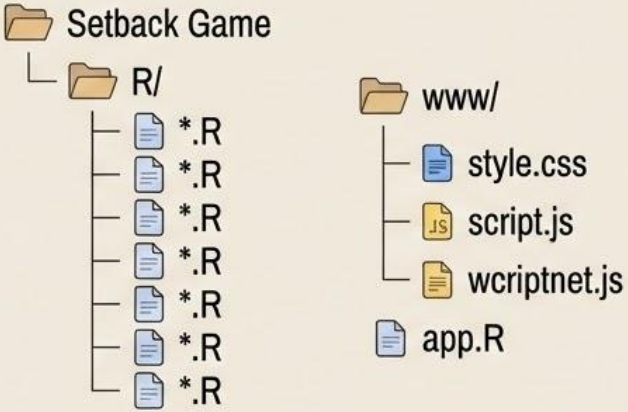


```
# CLAUDE.md - Architecture Plan

## Preject: Setback Card Game (R Shiny)
### Goal: Define architecture before Implementation.

**Decisions Log:**

- **State Management:**
  - Decision: Use `reactiveValues()`
  - Rationale: Gane state is transient within a session; no long-term
  persistence (database) requlred. `R6` classes add unnecessary
  complexity for this use case.

- **Timing & Game Loop:**
  - Decision: Use `JS setTimeout` via `shinyjs`
  - Rationale: Requires precise, non-blocking delays for AI turns and
  animations. `invalidateLater()` is polling-based and less accurate
  for game timing.

- **Card Rendering:**
  - Decision: Pure CSS with Unicode suit symbols
  - Rationale: Lightweight, scalable, and eliminates dependency on
  external image assets.

- **Code Structure:**
  - Decision: Flat structure with `source()` files
  - Rationale: Single-page application with a simple namespace. Shiny
  modules are overkill and would complicate message passing.

```

Fast Execution

Once the spec was clear, building was almost instant

- **Phase 1:** Game engine + bidding — 107 tests
- **Phase 2:** Scoring + state machine — 144 total tests
- **Phase 3:** AI strategy with partnership awareness — 184 total tests
- **Phase 4:** Pure CSS cards + dark theme UI
- **Phase 5:** Full interactive game loop

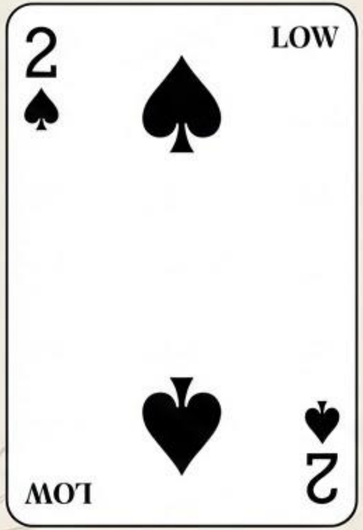


- Setback Game
  - R/
    - *.R
    - *.R
    - *.R
    - *.R
    - *.R
    - *.R
    - *.R
  - www/
    - style.css
    - script.js
    - wcriptnet.js
  - app.R

7 R files, 3 web files, one **app.R** — all through conversation. The AI wrote the state machine, the trick resolution, the legal play validation, the scoring — all correctly on first pass.

Low

LOW: The lowest card in play — someone has to hold it



In Setback, the Low point goes to whoever held the **lowest card** in the suit

In AI collaboration, there are low points too — where the AI writes code that looks right, tests pass, but something is fundamentally broken

Scoring Breakdown

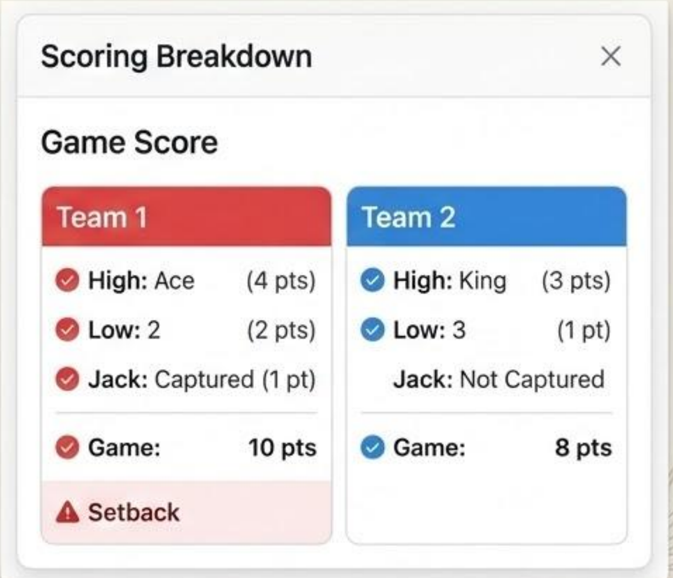
“I told the AI I was confused about counting. It built me a teacher.”

“It would be really helpful to show how everyone counts their cards at the end — that’s something I always feel confused about”

Claude Code built a full scoring breakdown panel: H/L/J/G with team colors, card values, who captured what, setback indicators.

It turned a vague complaint **into a concrete UI feature in one conversation.**

The AI is good at translating “I’m confused about X” into “here’s a UI that explains X”.



Scoring Breakdown

Game Score

Team 1	Team 2
✓ **High:** Ace (4 pts)	✓ **High:** King (3 pts)
✓ **Low:** 2 (2 pts)	✓ **Low:** 3 (1 pt)
✓ **Jack:** Captured (1 pt)	**Jack:** Not Captured
✓ **Game:** **10 pts**	✓ **Game:** **8 pts**
⚠ Setback	

10-Card Protection

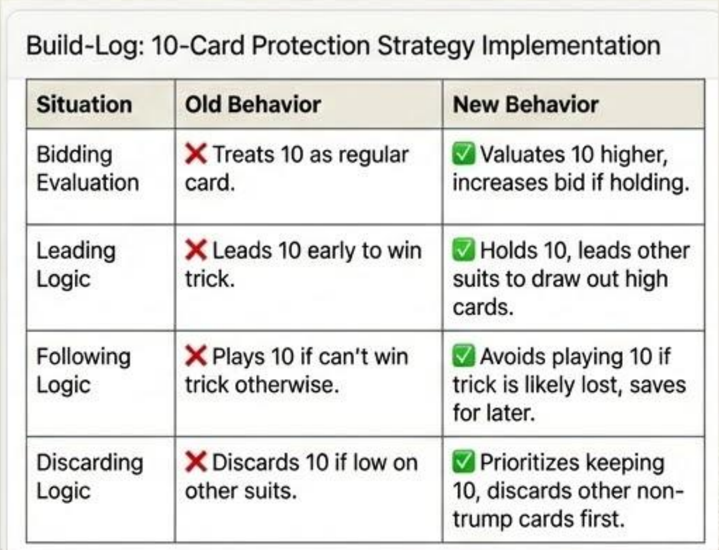
“I described a strategy concept. The AI implemented it across 4 decision points.”

“Players should protect their 10s — they’re worth 10 game points”

Claude Code updated: bidding evaluation, leading logic, following logic, discarding logic.

It understood the game theory and applied it consistently.

When the AI has clear context (the rules, the architecture, the goal), it can implement nuanced strategy.



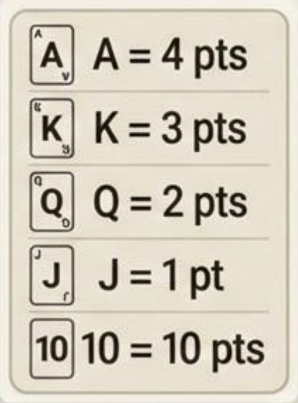
Build-Log: 10-Card Protection Strategy Implementation

Situation	Old Behavior	New Behavior
Bidding Evaluation	❌ Treats 10 as regular card.	✅ Valuates 10 higher, increases bid if holding.
Leading Logic	❌ Leads 10 early to win trick.	✅ Holds 10, leads other suits to draw out high cards.
Following Logic	❌ Plays 10 if can't win trick otherwise.	✅ Avoids playing 10 if trick is likely lost, saves for later.
Discarding Logic	❌ Discards 10 if low on other suits.	✅ Prioritizes keeping 10, discards other non-trump cards first.

Game

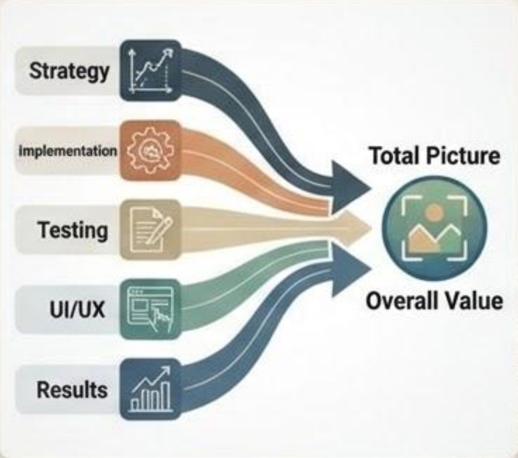
“GAME: Count up everything and see who came out ahead”

In Setback, the Game point goes to the team with the highest total card points from all their tricks



A = 4 pts
K = 3 pts
Q = 2 pts
J = 1 pt
10 = 10 pts

For us: what’s the total picture of building with an AI agent?



Knowing When You're Getting Set Back

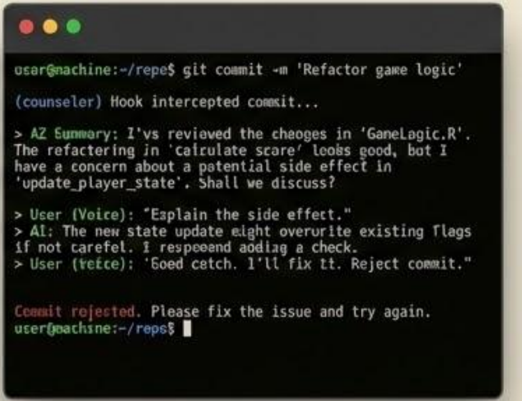
Miss your bid, lose points. Same with AI — compounding bugs from insufficient review create technical debt.

The Problem: Solo Development & Debt

- When developing solo, you miss the 'hey, what do you think?' colleague experience.
- This lack of review leads to compounding bugs and technical debt.

{counselor}: Voice-Powered Pre-Commit Hook

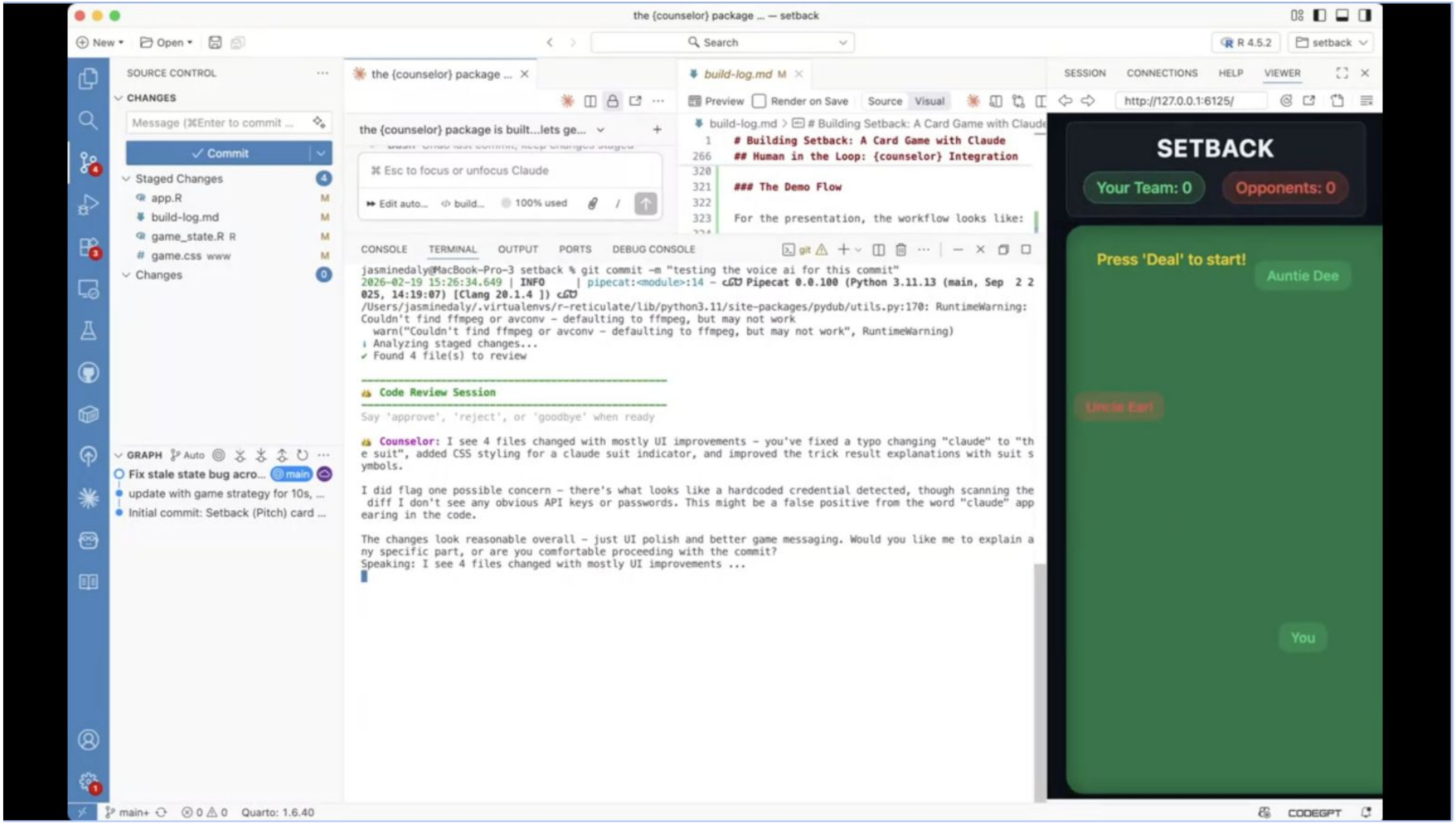
A voice-powered pre-commit hook built as an additional R package to help build this setback game.



```
usar@nachine:~/repe$ git commit -m 'Refactor game logic'

(counseler) Hook intercepted commit...

> AZ Summary: I'vs reviewed the cheoges in 'GaneLagic.R'.
The refactering in 'calculate scare' looks good, bot I
have a concern about a patential side effect in
'update_player_state'. Shall we discuss?

> User (Voice): "Esplain the side effect."
> AI: The new state update might overurite existing flags
if not carefel. I respeeend aodiag a check.
> User (Voice): 'Goed cetch. I'll fix tt. Reject commit."


Commit rejected. Please fix the issue and try again.
user@machine:~/reps$ █
```

The Solution: Conversational Review

- Intercepts every commit for a voice-powered conversation about your changes.
- Uses Deepgram for STT, Cartesia for TTS, and Claude for the review itself.
- Bridges the gap, giving you a chance to slow down and think carefully before submitting code.



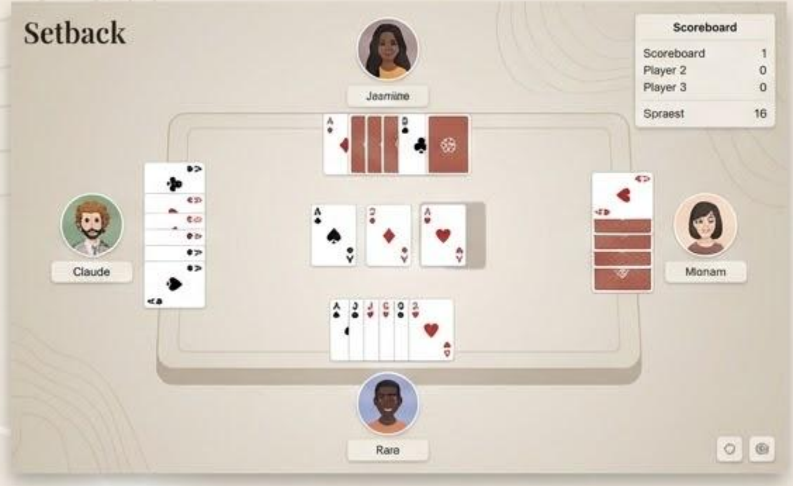
the {counselor} package ... — setback
New
Open
Search
R 4.5.2
setback
SOURCE CONTROL
CHANGES
Message (⌘Enter to commit ...
✓ Commit
Staged Changes 4
app.R M
build-log.md M
game_state.R R M
game.css www M
Changes 0
GRAPH Auto
Fix stale state bug acro... main
update with game strategy for 10s, ...
Initial commit: Setback (Pitch) card ...
the {counselor} package ...
the {counselor} package is built...lets ge...
⌘ Esc to focus or unfocus Claude
Edit auto... build... 100% used
build-log.md M
Preview
Render on Save
Source
Visual
build-log.md > # Building Setback: A Card Game with Claude
SESSION
CONNECTIONS
HELP
VIEWER
http://127.0.0.1:6125/
CONSOLE
TERMINAL
OUTPUT
PORTS
DEBUG CONSOLE
main+ ⊗ 0 ⚠ 0 Quarto: 1.6.40
CODEGPT

```
# Building Setback: A Card Game with Claude
## Human in the Loop: {counselor} Integration

### The Demo Flow

For the presentation, the workflow looks like:
```

```
jasminedaly@MacBook-Pro-3 setback % git commit -m "testing the voice ai for this commit"
2026-02-19 15:26:34.649 | INFO     | pipecat:<module>:14 - ᓚᘏᗢ Pipecat 0.0.100 (Python 3.11.13 (main, Sep  2 2
025, 14:19:07) [Clang 20.1.4 ]) ᓚᘏᗢ
/Users/jasminedaly/.virtualenvs/r-reticulate/lib/python3.11/site-packages/pydub/utils.py:170: RuntimeWarning:
Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work
  warn("Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work", RuntimeWarning)
ℹ Analyzing staged changes...
✓ Found 4 file(s) to review

────────────────────────────────────────────────────
🤖 Code Review Session
────────────────────────────────────────────────────
Say 'approve', 'reject', or 'goodbye' when ready

🤖 Counselor: I see 4 files changed with mostly UI improvements - you've fixed a typo changing "claude" to "th
e suit", added CSS styling for a claude suit indicator, and improved the trick result explanations with suit s
ymbols.

I did flag one possible concern - there's what looks like a hardcoded credential detected, though scanning the
 diff I don't see any obvious API keys or passwords. This might be a false positive from the word "claude" app
earing in the code.

The changes look reasonable overall - just UI polish and better game messaging. Would you like me to explain a
ny specific part, or are you comfortable proceeding with the commit?
Speaking: I see 4 files changed with mostly UI improvements ...
```

Thank You

High, Low, Jack, & Claude



Scoreboard	
Scoreboard	1
Player 2	0
Player 3	0
Spraest	16

Jasmine Daly — https://www.dalyanalytics.com/

github.com/dalyanalytics/setback

github.com/dalyanalytics/counselor

RainbowR 2026