Milestone 3: API Design Uno

Eric Ngo, Travis Ruso, Randy Chen, Luca Morgan

What is Uno?

The Core Concept

Our game is a two to four player, turn-based card game built on the ruleset

- Players match the last card played by its color or symbol

- Special cards (Skip, Reverse, Draw 2+ / 4+) change the game's flow

- Wild cards allow the player to change the active color

- The first player to empty their hand wins the game

API Endpoint Overview

Setup & Management

- POST /api/users/register

- POST /api/users/login

- POST /api/games (Create)

- POST /api/games/{game_id}/join

- POST /api/games/{game_id}/start

- POST /api/chat/message

- GET /api/chat/message

Gameplay & State

- GET /api/games/{game_id} (Public State)

- GET /api/games/{game_id} (Private Hand)

- POST /api/games/{game_id}/play_card

- POST /api/games/{game_id}/draw_card

- POST /api/games/{game_id}/end-turn

Deep Dive: POST /api/games/{game_id}/start

This is one of our endpoints. It's responsible for turning a lobby of waiting players into a live, playable game.

Key Validations

- Is the user still the host

- Is the game still in the lobby?

- Are there 2 to 4 players?

State Updates

- Deals 7 cards to each player

- Flips the first discard card

- Sets game status to 'active'

- Sets first player's turn

- Sets turn order

Socket.io events

- game:state:update (to all)

- game:hand:update (to sender)

Deep Dive: POST /api/games/{game_id}/play_card

This endpoint is the core game loop. It processes every player's move and validates the game rules on the server.

Key Validations

- Is it your turn?
- Do you have this card?
- Is the move legal? (matches color/symbol or is wild)

State Updates

- Card moves from hand to discard, becomes top discard/last played.
- Turn advances to next player
- Applies special logic (Skip, Draw 2, Reverse, etc.)

Socket.io events

- game:state:update (to all)
- game:hand:update (to sender
- game:ended (if last card)

Socket.io Event Summary

Event	Scope	Trigger	Data
game:state:update	All Players	After any action that changes the public game state (card played, card drawn, turn ended, game started).	{ id, player_turn, last_card_played, players }
game:hand:update	One Player	After a player draws a card, plays a card, or when the game starts (initial deal).	{ "hand": [{ "id": 10, "color": "red", "symbol": "one" }, etc.] }
game:player:joined	All Players	After a user successfully joins a lobby (POST .../join).	{ "user_id": 2, "username": "player_2", "turn_order": 2 }
game:ended	All Players	When a player plays their last card.	{ "winner_id": 1, "winner_username": "host_user" }
error:message	One Player	When a user's API call fails validation in a way that needs to be communicated (e.g., "Invalid move").	{ "message": "That is not a valid move. The last card was a Blue 5." }
chat:update	All Players	Update the local text inside of the in game chat	{"user_id": 2, "username": player2, "message": "This is a message"}

Questions?

Thank you for your time.