

Building a full competitive Magic: the Gathering (MTG) game engine + AI

Competitive Elder Dragon Highlander
(EDH) · Multiplayer · Imperfect Information
Richard Lam · 2026

Preface

Why MTG is a harder AI problem than Chess or Go

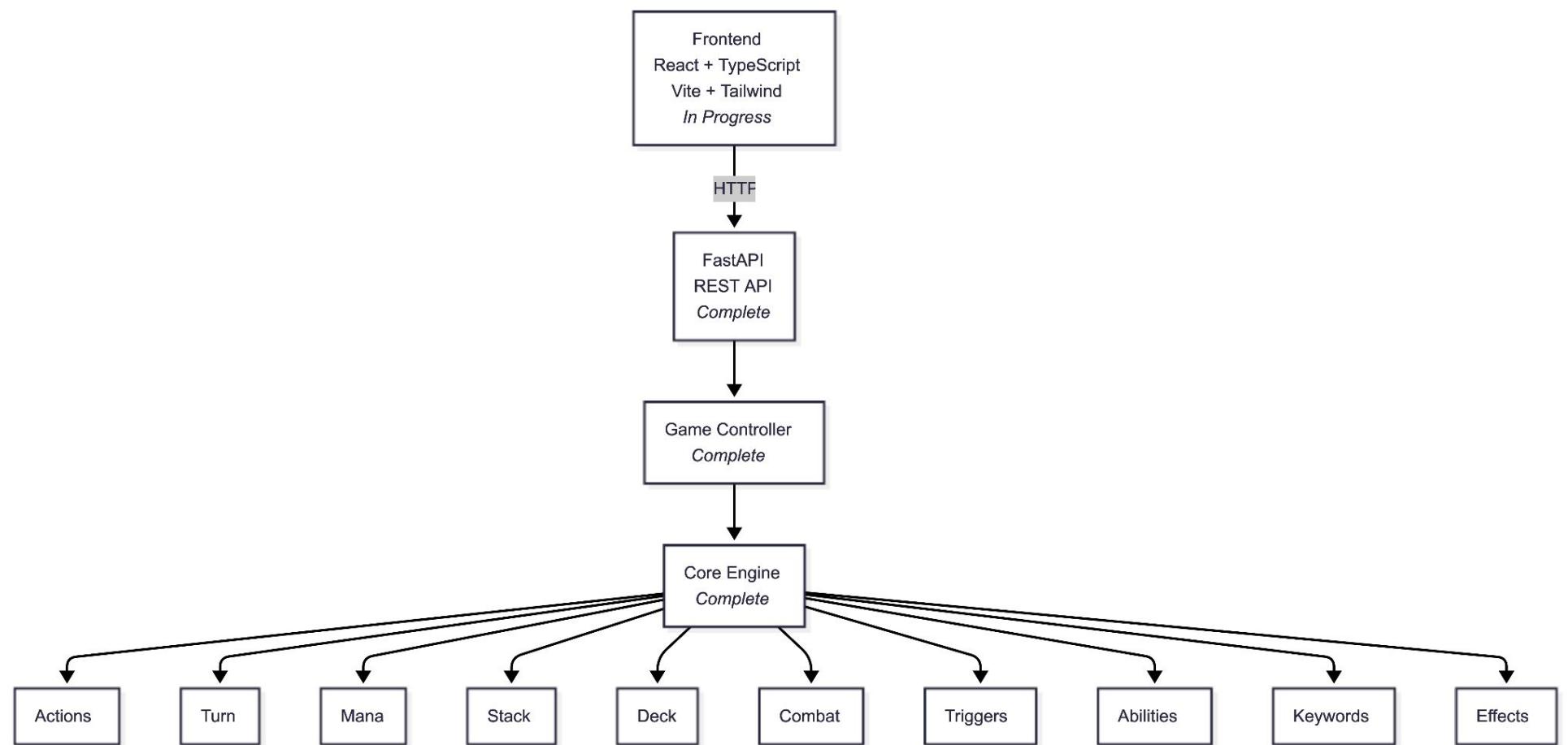
- MTG combines hidden information, multiplayer dynamics, and a complex rules engine; making it closer to a multi-agent system than a board game.

TABLE I
COMPARATIVE ANALYSIS OF CHESS, GO, AND MAGIC: THE GATHERING (MTG)

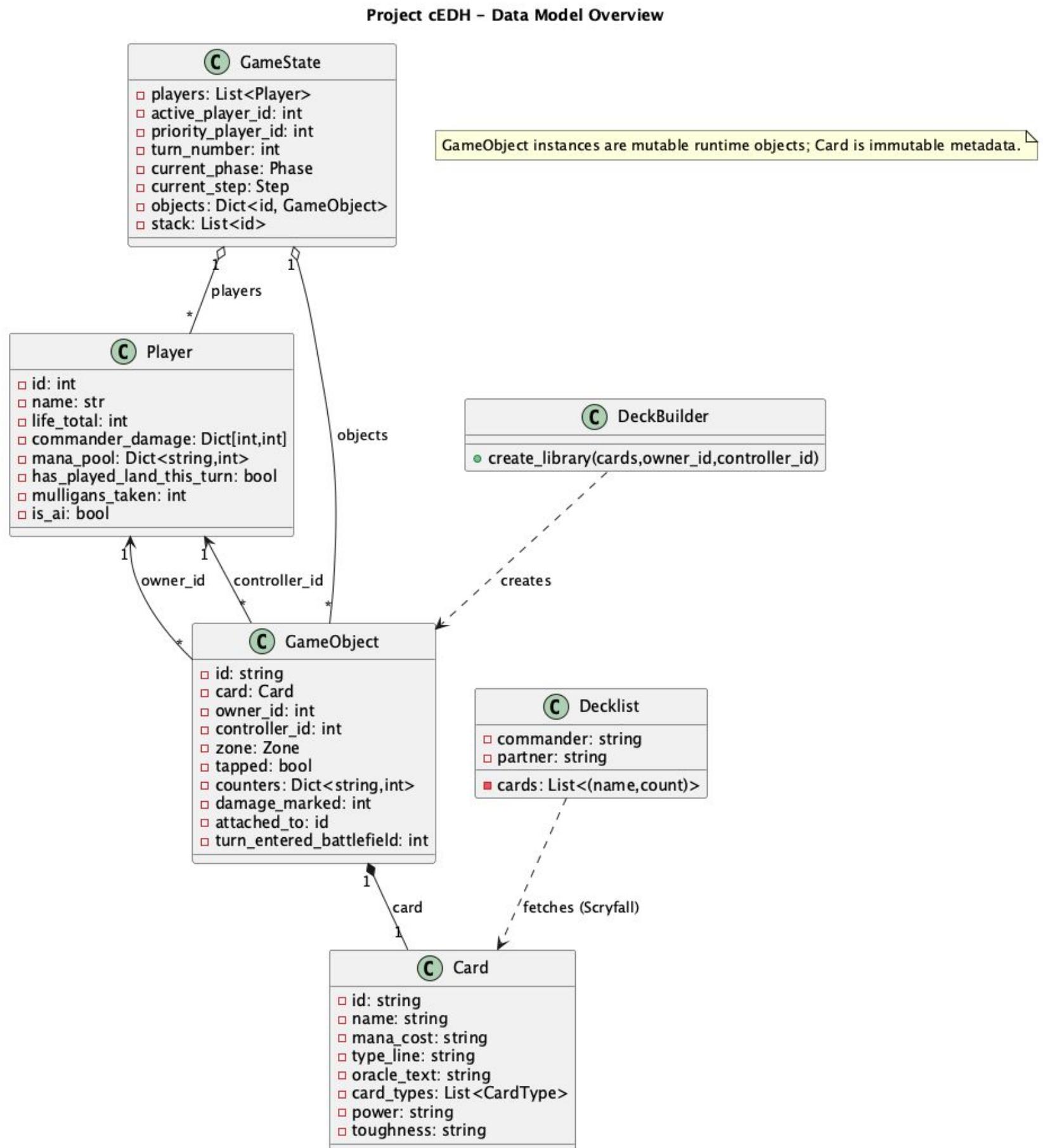
Aspect	Chess	Go	Magic: The Gathering (MTG)
Type	Abstract, perfect-information, two-player board game	Abstract, perfect-information, two-player board game	Collectible card game, multiplayer, imperfect information
Origin	6th century India / Europe evolution	China, 2000 BCE	1993, USA
Complexity	High tactical & positional depth, but finite	Extremely high; simple rules, vast strategy space	Extremely high; combinatorial depth, rules interactions, and emergent strategies
Game State Size	$\sim 10^{47}$ positions	$\sim 10^{170}$ positions	Effectively unbounded due to card combinations & interactions
Information	Perfect	Perfect	Imperfect (hidden cards, draws, opponent strategy)
Decision Type	Deterministic, tactical + strategic planning	Deterministic, long-term strategic planning	Probabilistic, adaptive, multi-layered strategic reasoning
Time to Master	Decades to become grandmaster	Lifelong mastery possible; very high ceiling	Lifelong mastery; constantly evolving meta, high variance
Recognition	Globally established, competitive tournaments	Culturally iconic in East Asia, growing worldwide	Recognized by MIT Technology Review as the world's most complex game; evolving competitive scene
Skills Required	Calculation, pattern recognition, strategy, endgame mastery	Pattern recognition, long-term planning, balance of influence	Systems thinking, probability, optimization, meta-analysis, adaptability, creativity

Software Architecture

UI → API → Game Controller → Core Rules Engine

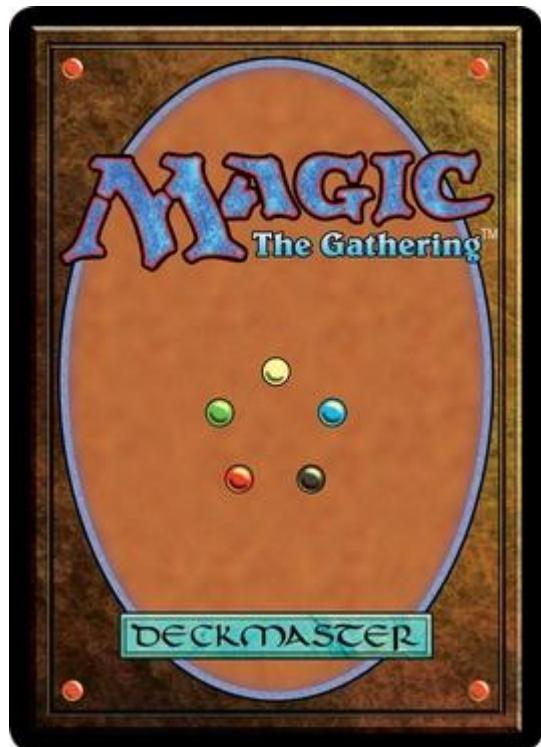


Game State / Data Model



Core Engine

- Turn & priority system
- Stack resolution & counterspells
- Mana costs (hybrid, phyrexian, X)
- Combat (multiplayer attackers / blockers)
- Triggers, abilities, keywords
- Oracle text execution



End-to-End Gameplay

Work in Progress

- ✓ Create game from decklists
- ✓ Commander London Mulligan
- ✓ Full turn cycle
- ✓ Cast spells & interact on stack
- ✓ Combat, triggers, state-based actions
- ✓ AI players auto-pass priority

Player 1 - Mulligan Decision

Current hand: 7 cards

Your Hand

Commander Mulligan Rules:

- Draw 7 cards for your opening hand
- Decide to keep or mulligan
- **First mulligan is FREE** (no cards to bottom)
- Subsequent mulligans: put X cards on bottom (X = mulligans - 1)

Mulligan (FREE) **Keep This Hand**

What's Next?

- Perfect backend mechanics
- Finish frontend polish
- Smarter AI (beyond auto-pass)
- Heuristic evaluation
- Monte Carlo Tree Search
- Self-play infrastructure

Demo

- Short engine demo in the comments ↓
- Headless execution + UI interaction
- AI currently pass-only