GM Agents: Heuristic-Based, Reward-Oriented Browser Game Playing Agents

Skanda Vasishta

Sarvesh Jayaraman

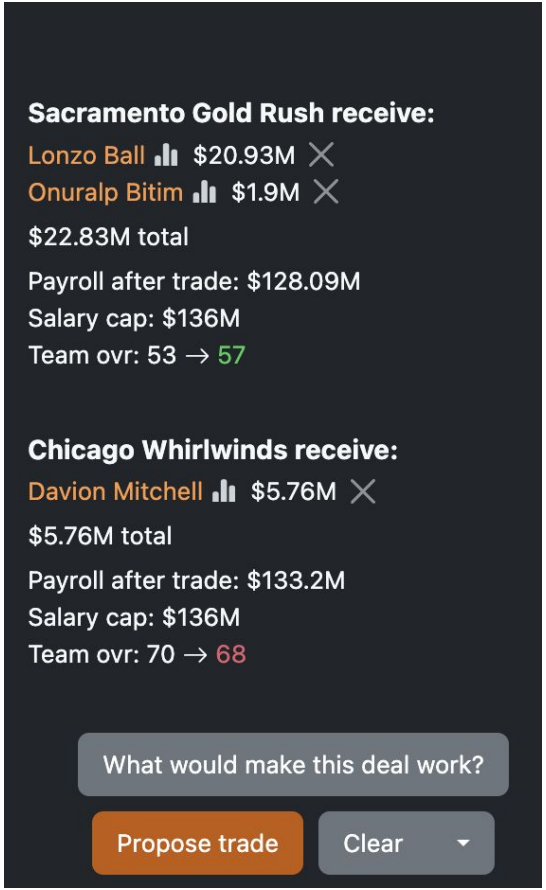
UC SANTA BARBARA

Motivation

- Goal: Multi-agent gameplay with agent managing a simulated basketball team
- Balances wins with realistic GM decisions
- Applies to long-term strategic planning
- Add determinism to browsing agent actions (with discrete but large action space/environment)

Game Environment: Basketball-GM

- Browser-based management sims:
  - Manage Roster/contracts
  - Trade negotiations
  - Set lineups & tactics
- Episode:
  - one season
  - decisions at beginning of season, trade deadline, end of season



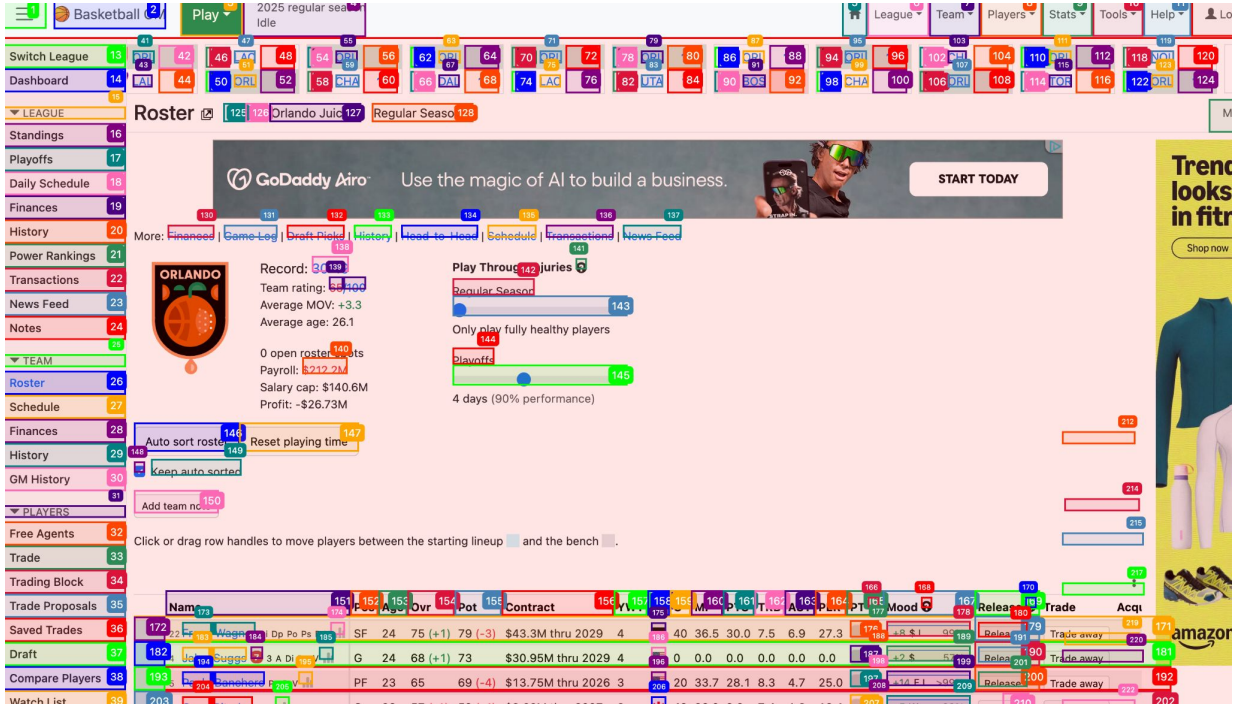
Similar Works

- Voyager (LLM-powered Minecraft agent)
  - Lets GPT-4 write and refine code in-game, building a growing "skill library" that the agent can reuse.
- Game-theoretic LLM (negotiation workflow)
  - Wraps any LLM in lightweight game-theory prompts (e.g., best response, backward induction) to drive choices in bargaining games.

Tech Stack

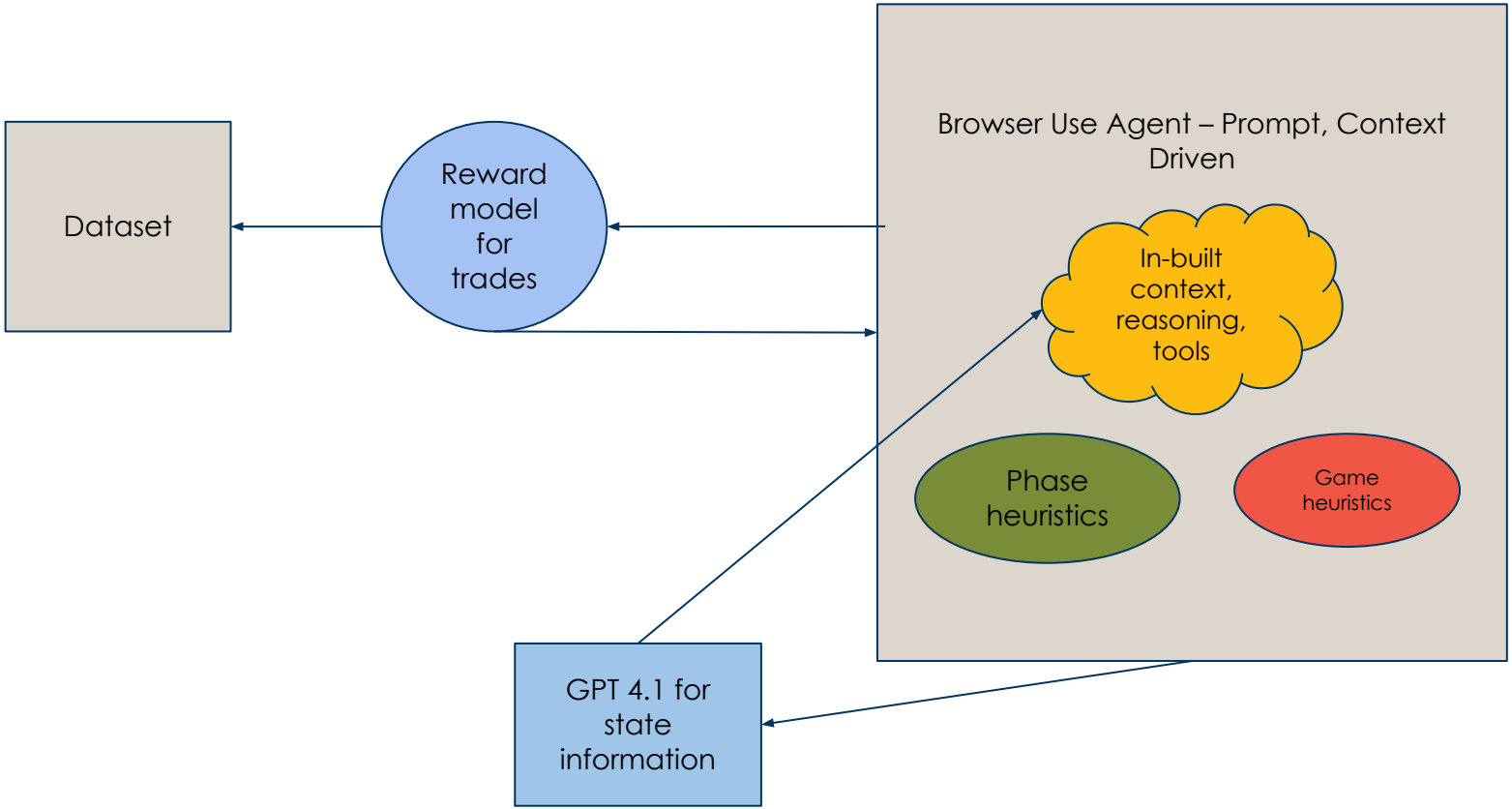
- Browser_Use
  - GPT 4o as backbone
  - Hooks + tools
- Playwright
  - Browser env
- GPT 4.1
  - Get state information
- Scikit-Learn
  - Reward model training

Browser Use




```
Your mission: Build a championship-winning basketball team through smart, data-driven decisions and strategic management!
INFO     [browser] 🌎 Launching local browser driver=playwright channel=chromium user_data_dir=~/.config/browseruse/profiles/default
{"record":"20-18","team_rating":"65/100","average_mov":"+1.3","average_age":"26.2","open_roster_spots":"0","payroll":"$212.2M","salary_cap":"$140.6M","profit":
"-$23.84M"}
INFO     [agent] 📍 Step 1: Evaluating page with 223 interactive elements on: https://play.basketball-gm.com/l/34/roster
═══════════════════════════════════════════════════════════════════════════════════════════════════════════════════════════════════════════════
INFO     [agent] 🧠 LLM call => ChatOpenAI [▀▄8 msg, ~8053 tk, 1026625 char, 🏞 img] => JSON out + 🔨 22 tools (function_calling)
INFO     [agent] 👍 Eval: Success - The task involves making a trade or free agent signing, which can be accessed via the sidebar on this current page. This is
 aligned with the task needs at the trade deadline phase.
INFO     [agent] 🧠 Memory: Currently at the trade deadline phase on the roster page. The team has a record of 30-19, team rating is 65/100, average MOV is +3.
3, average age is 26.1, 0 open roster spots, with a payroll of $212.2M over the salary cap of $140.6M, and a profit of -$26.73M. Injuries noted include Franz W
agner (11 games) and Anthony Black (5 games). Must make at least one roster move quickly.
INFO     [agent] 🎯 Next goal: Access the trade options from sidebar for potential trades or free agent signings. Evaluate available moves for quick execution.
```

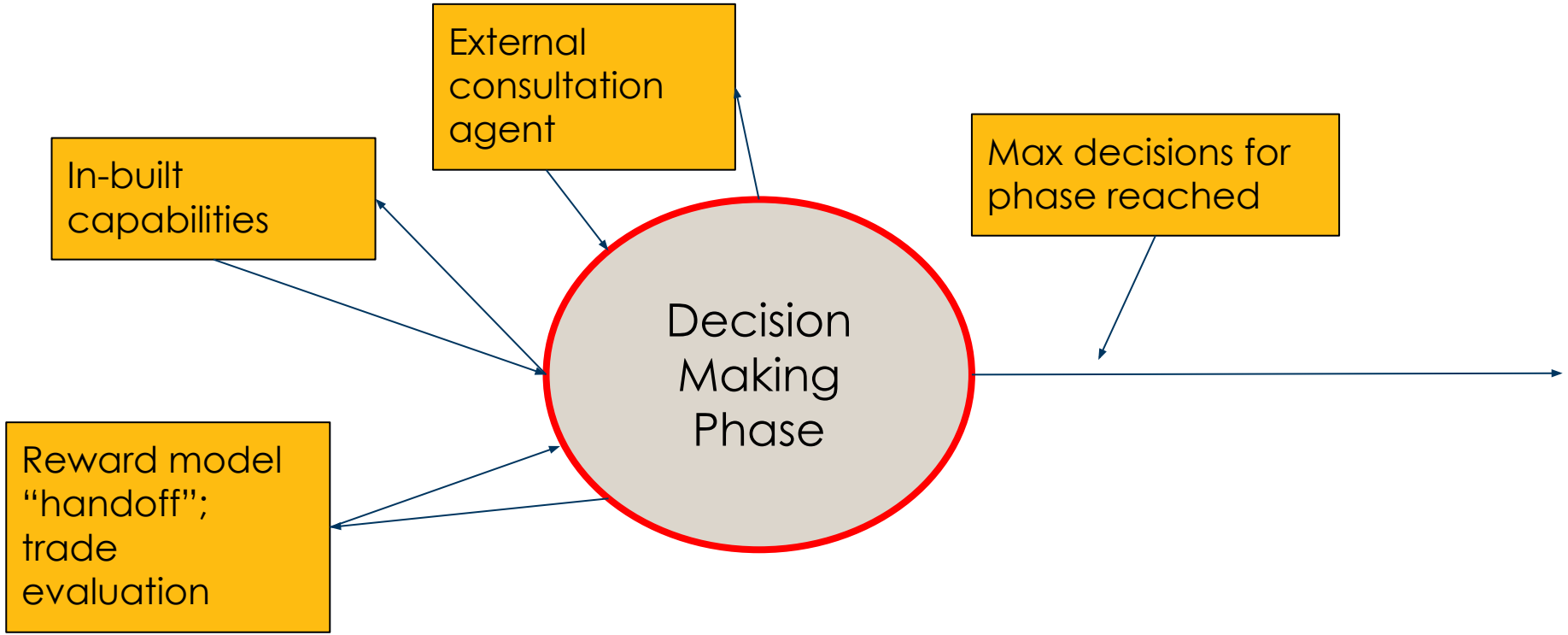
System Architecture



Multiple Agents, Heuristics w/ Browser Use


```
class PhaseManager:
    def __init__(self):
        self.current_phase = "trade_deadline"
        self.actions_remaining = 4
```

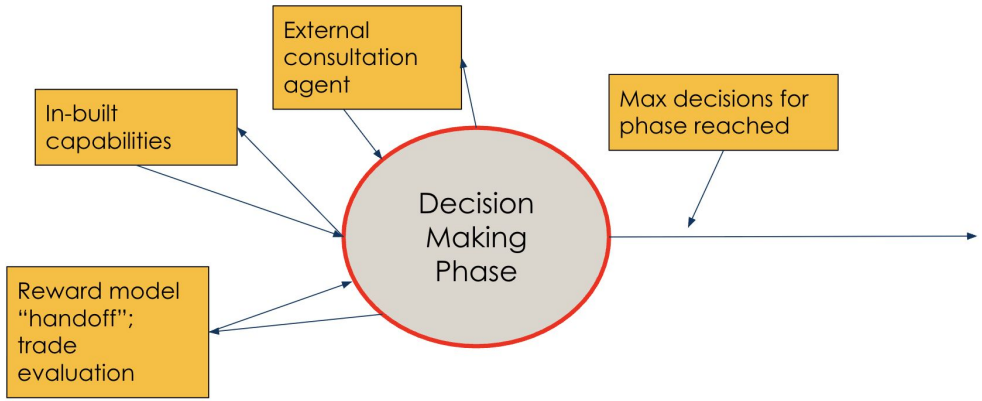
Decision Making Phase, Closer Look



Decision Making Phase, Closer Look

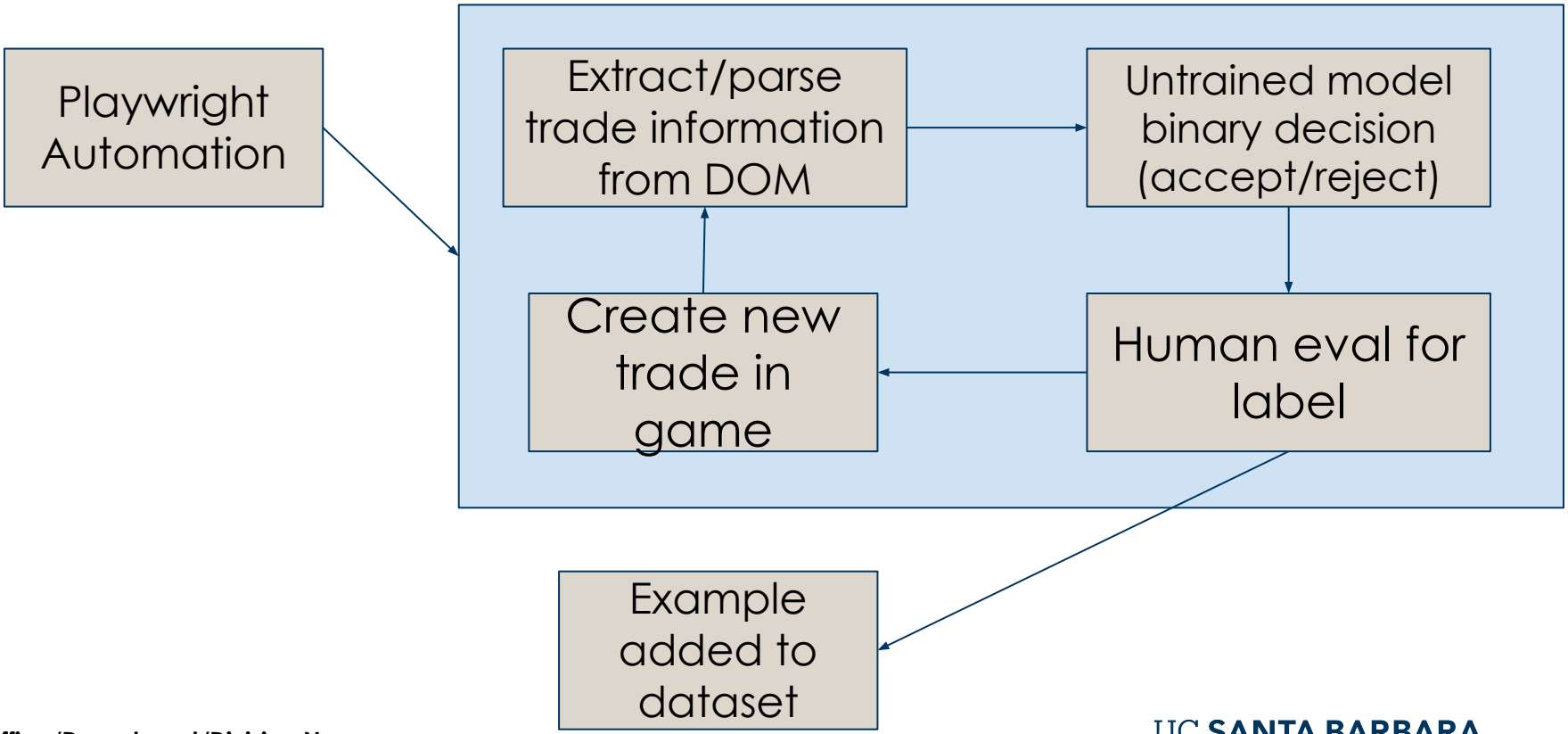
General workflow for “step”:

1. Gather + inject state info to agent via GPT4.1
2. ONCE per phase:
   a. Reward model “handoff” for trades
3. General model decision making behaviors, external agent consultation
4. num_decisions++



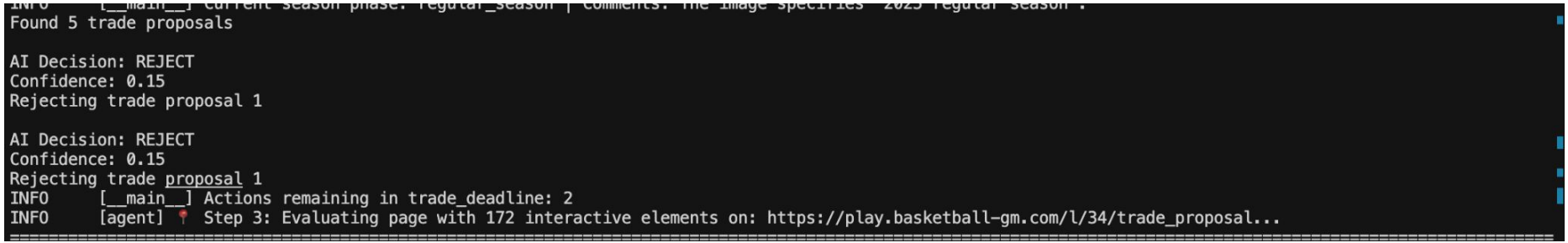
```
Your mission: Build a championship-winning basketball team through smart, data-driven decisions and strategic management!
INFO     [browser] 🌎 Launching local browser driver=playwright channel=chromium user_data_dir=~/.config/browseruse/profiles/default
{"record":"20-18","team_rating":"65/100","average_mov":"+1.3","average_age":"26.2","open_roster_spots":"0","payroll":"$212.2M","salary_cap":"$140
"-$23.84M"}
INFO     [agent] 📍 Step 1: Evaluating page with 223 interactive elements on: https://play.basketball-gm.com/l/34/roster
================================================================================================================================================
INFO     [agent] 🧠 LLM call => ChatOpenAI [✉️ 8 msg, ~8053 tk, 1026625 char, 📷 img] => JSON out + 🔨 22 tools (function_calling)
INFO     [agent] 👍 Eval: Success - The task involves making a trade or free agent signing, which can be accessed via the sidebar on this current
 aligned with the task needs at the trade deadline phase.
INFO     [agent] 🧠 Memory: Currently at the trade deadline phase on the roster page. The team has a record of 30-19, team rating is 65/100, aver
3, average age is 26.1, 0 open roster spots, with a payroll of $212.2M over the salary cap of $140.6M, and a profit of -$26.73M. Injuries noted i
agner (11 games) and Anthony Black (5 games). Must make at least one roster move quickly.
INFO     [agent] 🎯 Next goal: Access the trade options from sidebar for potential trades or free agent signings. Evaluate available moves for qu
INFO     [controller] 🖱️ Clicked button with index 33: Trade
INFO     [agent] ☑️ Executed action 1/2: click_element_by_index
INFO     [agent] Something new appeared after action 1 / 2
INFO     [agent] 📍 Step 2: Ran 2 actions in 20.01s: ✅ 2
INFO     [__main__] Current season phase: regular_season | Comments: The phase is specified as the '2025 regular season'.
INFO     [__main__] Actions remaining in trade_deadline: 3
INFO     [agent] 📍 Step 2: Evaluating page with 331 interactive elements on: https://play.basketball-gm.com/l/34/trade
================================================================================================================================================
```

Dataset Generation (Reward Model)



Reward Model Fine-Tuning

- Fine tune reward model on data labels to either accept/reject trade
  - Each trade proposal (with its description, players, salaries, etc.) is a "document" with a label
- Parse log → dataset: Extract trade description text + 0/1 human label from dataset
- Vectorise text: TF-IDF turns words/bigrams (≤20 k) into sparse features.
- Train logistic reg: Fast linear model learns P(good | text)
- Why not a larger model?
  - Limited corpus of data (~100 manually annotated examples)



Evaluation

*ran in headless browser state

With reward model handoff; 20 simulated game “runs”


```
"team_rating": "59.4/100",
"average_mov": 1.1,
"average_age": 26.8,
"open_roster_spots": 1.2,
"payroll": "$180.7M",
"profit": "$6.1M",
"trades_made": 4.7,
"won_championship_rate": "13.4%",
"average_win_percentage": "56.2%"
```


No reward model handoff; 20 simulated game “runs”


```
"team_rating": "50.2/100",
"average_mov": -2.8,
"average_age": 27.0,
"open_roster_spots": 1.6,
"payroll": "$172.8M",
"profit": "$0.8M",
"trades_made": 3.2,
"won_championship_rate": "2.4%",
"average_win_percentage": "36.7%"
```

Future Additions

- Fine tune on more powerful model
  - Artificially augment more reward model samples
- Extend reward modelling approach to other in game actions instead of relying solely on model
- RL-based approach for sequential actions
  - vs supervised fine tuning
- Add more phase support
- Dynamic max phase action generation (based on need)