A Magic: the Gathering Tournament Simulator in AnyLogic

Elia Togni[1] and Riccardo Conforto Galli[2]

[1,2]Dipartimento di Informatica, Università degli Studi di Milano.

We declare that this material, which we now submit for assessment, is entirely out own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of our work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should we engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study.

Abstract

This paper presents an agent-based model simulation of a Magic: The Gathering tournament, focusing on deck performance rather than individual player skill or in-game details. The tournament begins with deck submission, followed by multiple swiss-system rounds where players are paired based on their win-loss-draw records. Match results are simulated using matchup percentages derived from player data and online platforms, bypassing detailed gameplay simulation. Simplifications, such as assuming perfect adherence to rules and fixed decklists, were made to reduce player variance and focus on deck performance. These assumptions, along with the limitations of the model, are acknowledged to provide a clear focus on how different decks perform in the tournament environment.

Results highlight some discrepancies in deck performance, partly due to the absence of some winning decks in the analysis. These missing decks affect the frequency distribution, leading to imbalances in results. For example, decks like Turbo Fog and Grixis Affinity appear to overperform, while others, such as Golgari Gardens, perform worse. These variations stem from missing data for decks that would typically challenge the stronger-performing ones, thus affecting the overall results. Despite these limitations, the model provides valuable insights into deck performance within the tournament environment.

Acknowledgments

A heartfelt thanks to Brognoli Roberto and Donfrancesco Paolo for sharing data and insights with us.

Contents

List of Figures

List of Tables

I Introduction

1 What is Magic: the Gathering

Magic: The Gathering (MTG) is a paper and digital collectible card game created by Richard Garfield1 and first published by Wizards of the Coast in 1993. It is widely regarded as the first trading card game (TCG) and has grown to become one of the most popular and influential games of its kind. MTG combines strategy, deck-building, and fantasy lore, attracting millions of players worldwide and forming a robust competitive scene.

Figure 1: Richard Garfield, the father of Magic: the Gathering

1.1 Gameplay Basics

At its core, Magic: The Gathering is a game where players, referred to as **Planeswalkers**, use decks of cards to battle against each other. Each player begins with a life total, typically 20, and the objective is to reduce the opponent's life total to zero. The game is played with decks that consist of a minimum of 60 cards, including a mix of lands, creatures, spells, enchantments, artifacts, and planeswalkers.

Players construct their decks from a pool of available cards of different types. To cast spells and summon creatures, players need **mana**, an in-game resource which is produced by lands. Basic lands are cards that generate different colors of mana — white, blue, black, red, and green — and mana is needed to cast cards of the corresponding colors, each corresponding to a different play style and strategic approach. A deck may contain more than one color to gain more play style flexibility in change of **tempo**[1] and stability. The game also features other types of cards:

- **creatures**, which represent beings that can attack opponents and defend against attacks;

- **instants and sorceries**, cards that have a one-time effect when played;
- **enchantments**, which provide ongoing effects that can enhance creatures or offer other benefits;
- **artifacts**, items that provide various abilities and can be included in any deck.

The **rarity** of Magic: The Gathering cards adds a significant collectible aspect to the game. Cards are categorized into common, uncommon, rare, and mythic rare, with some sets also including special categories like promotional or limited-edition cards. Rarity influences not only the frequency of finding these cards in booster packs but also their power level and desirability among players and collectors.

Figure 2: The iconic Black Lotus card, one of the most sought-after and valuable cards in the game

1.2 Game Formats

Magic: The Gathering can be played in several formats, each with its own rules and deck-building constraints. Popular formats include:

- **Standard**: a format that uses the most recent sets of cards and rotates periodically;
- **Modern**: a format that allows cards from sets dating back to 2003;
- **Commander** (EDH): a multiplayer format where players build 100-card singleton decks led by a legendary creature;
- **Limited**: a format in which players build decks from a limited pool of cards, usually obtained through booster drafts or sealed deck events;

- **Pauper**: a format in which only cards that have been printed at the common rarity are allowed to be used. This format is popular for its budget-friendly nature and its ability to highlight the game's fundamental mechanics and interactions.

1.3 Competitive Scene

MTG has a vibrant competitive scene, ranging from local **Friday Night Magic** (FNM) events to large-scale tournaments like the Magic: The Gathering **Pro Tour** and **Mythic Championships**. Players can also compete in online platforms such as **Magic: The Gathering Online** (MTGO) and **Magic: The Gathering Arena** (MTGA).

1.4 Tournament Structure

A Magic: The Gathering tournament typically begins with players registering and submitting their deck lists. Players are then paired for the first round, where they play a **best-of-three** match within a set time limit, usually of 50 minutes. After each round, the results are reported and players are re-paired based on their win-loss-draw records, following a swiss system that ensures players with similar records face each other. To do so, each round won grants the players 3 points, each draw grants 1 point while each loss doesn't grant any point. This continues for several rounds, depending on the number of participants. It is important to underline that two players cannot play against each other more than one time in the swiss rounds. After the swiss rounds, the 8 top-performing players advance to a **single-elimination playoff**, culminating in a final match to determine the tournament champion.

2 MTGO Pauper Challenges

MTGO (Magic: The Gathering Online) challenges are competitive events held regularly on the platform, where players can test their skills and deck strategies against others in the Magic: The Gathering community. These challenges typically feature a structured format, such as Standard, Modern, or Pauper, and attract a wide range of participants, usually numbering between 32 and 128 players. To enter, players generally pay an entry fee, which can range from 10 to 20 event tickets, depending on the specific challenge. Participants compete in a series of matches, with top performers earning prizes such as booster packs, additional event tickets, and qualification points for larger tournaments. The prize distribution is tiered, with higher placements receiving more substantial rewards, such as significant quantities of event tickets and booster packs, while those finishing in lower top positions still earn valuable rewards.

3 The Paupergeddon

Our project takes virtually place at **Paupergeddon**, a popular series of Magic: The Gathering tournaments focused on the **Pauper** format, where only common cards are allowed. Originating in Italy, these events have grown significantly, attracting players from across the globe who appreciate the accessibility and strategic depth of Pauper. The format's affordability and emphasis on skill over card rarity make Paupergeddon a beloved fixture in the Magic community, fostering a competitive yet welcoming environment for both new and experienced players. With three editions held annually (Lecco, Pisa and Rome), Paupergeddon has become the biggest Pauper tournament in the whole world, with almost 800 players.

Figure 3: The Paupergeddon: Winter Edition in Lecco

4 Overview

II Simulation Model

The chosen simulation paradigm is **agent-based model**, a computational simulation technique used to model the interactions of autonomous agents (which can represent individuals, groups, or entities) within a given environment. These agents follow simple rules and make decisions based on their interactions with other agents and the environment, allowing the study of complex systems and emergent behaviors that arise from these interactions.

The logic of the agents in our model will be described in details in the following section.

We don't simulate every game in detail but we simulate only the result, based on matchup percentages obtained from data gathered by the players themselves and by online platforms. Therefore, we don't consider elements like player skill (as Magic is a game of incomplete information and so, the ability of one player to understand the other's intentions makes significant differences in terms of winrate). We also assume that the players behave strictly according to the rules of the game, which is often not the case (and is penalized with warnings and game losses). Another significant assumption is the existence of only the considered decklists and that, within the same decklist, no card modifications exist (which in real life happens quite frequently). This simplification was made purely due to the impossibility of obtaining such data.

Note that there could be more other implementation details about the players that were not considered. However, since our main focus was to observe and to analyze how each deck would perform, it seemed reasonable to us to reduce the variance on each player to the maximum.

III Model Implementation

1 Agents

1.1 Player

A **Player** is an agent representing an individual taking part in a Magic: the Gathering tournament. To keep track of its score, which is recorded by the structure WINS - LOSSES - DRAWS, the agent has the following variables: *game_won* and *game_lost*. The other variables, *matches_won*, *matches_lost* and *matches_draw* are used to sort the players in between the rounds. Finally, the last variable, *opponents*, is an ArrayList of Players which is used to avoid rematches before the reaching of the top 8.
Whenever a Player is created, a deck is assigned to it, following the probability distribution obtained through the matchup dataset (which will be explained in a following section). Also, each Player changes color to the one representing the deck the agent is going to use in the tournament. Furthermore, on the creation, the pie chart's player counter is increased by one.
A Player has two states: the first one, **idle**, represents the player at the beginning of the tournament, waiting for the first pairings to come up, or the player at a table, playing a match. The second one, **moving**, represents the transition of the player from a table to another between two matches. The transition from idle state to moving state is caused by the reception of a message from Companion at the end of the current match, containing the identifier of the next Table. When the agent reaches the new Table, it notifies that Table of its arrival.

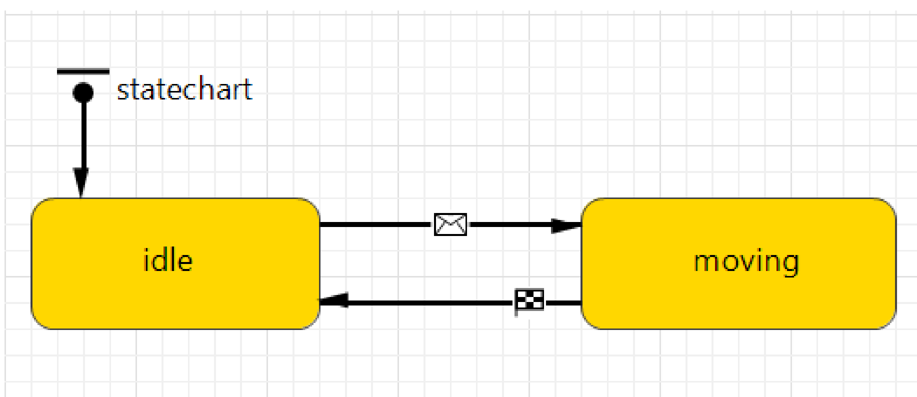

Figure 4: The Player Agent

1.2 Table

A **Table** is an agent used to model the match between two Players. In the entry state, **waitingForPlayer**, the Table awaits for the messages from the two Players assigned to it. The reception of the second one triggers a change of state, to the **waitStart** one. In this transition, the Table sends a message to Companion signaling the arrival of both Players. In the waitStart state, Table waits until Companion gives him the permission to start with the first game and then it moves to the **gameStart** state.

A game may end with the victory of one of the two Players or by timeout signal from Companion:

- in the first case, Table moves from gameStart to **gameResult**, in which the result of the game is generated through a Bernoullian extraction based on the matchup between the decks of the two Players. Next, the local variables containing the result of such game are updated. If no one has reached two victories, Table moves back to the gameStart state for the beginning of the next game. On the other hand, if a Player was able to win two games, table moves to the **sendResult** state;
- in the second case, the current game is interrupted and Table moves to the sendResult state.

On entering in sendResult, Table updates the score of each Player based on the result and sends it to Companion. Then, it returns into waitingForPlayer for the following turn.

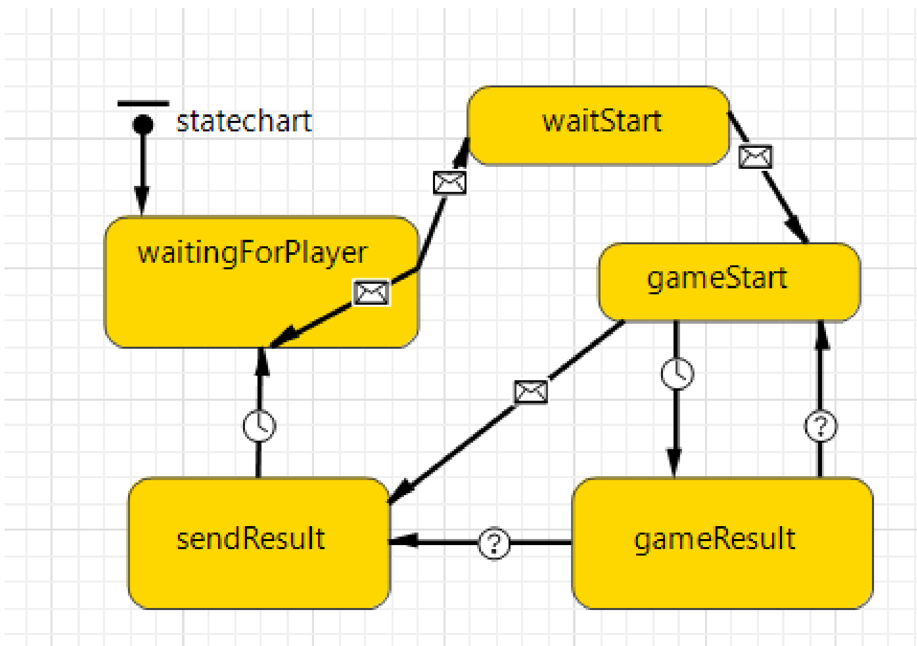

Figure 5: The Table Agent

1.3 Companion

This agent takes his name from the official mobile app that plays a crucial role in managing and tracking Magic: The Gathering tournaments. Like its real-world counterpart, this agent takes care of generating pairings for the swiss rounds, assigning tables to the players, granting the permission to start matches and keeping track of the time limit. Once finished the swiss rounds, the agent manages the direct elimination stage of the event by eliminating losing players at each turn and pairing the remaining ones following the tournament bracket structure described later. To simplify the description of the state chart, the two parts of the event will be explained separately.

First the agent needs to manage the swiss rounds: entering the **pairings** state, the agent generates pairings for the player sorting them using their score (and randomly in the first round). A player's score is derived from the following formula: $3 * n$ wins $+ n$ draws. Then, it pairs the first player with the next one in the rank that has not already played against, sends to each of them a message containing the id of the Table and removes the two players from the list. Once the list is empty, Companion enters the **waitPlayers** state. Here, each time this agent receives a message from a Table, a counter is incremented until all Tables have notified the players arrival, triggering the start of the turn clock and the start of the first game of the match for each Table. After that, Companion goes in the **waitResults** state where, each time it receives a message from a table, a counter is incremented until all the tables have notified the end of the match. When all the players' messages are received, if the actual swiss round is not the last one, the agent goes in the **pairing** state and the loop starts again.

In case the actual round is the last one, the agent goes in the first state of the top 8 phase.

When Companion reaches the **startTop** state, all the players except the top 8 players are removed from the simulation, all tables are removed except the first four and the top 8 players' decks are written as output of the simulation in the **Swiss Ladder** box contained in the Statistics view. Players are then paired as follows:

- the first player in the ladder is paired with last one at the first Table;
- the fourth player in the ladder is paired with the fifth one at the second Table;
- the third player in the ladder is paired with the sixth one at the third Table;
- the second player in the ladder is paired with the seventh one at the fourth Table.

The behaviour of the two states **waitTopPlayers** and **waitTopResults** is almost the same as **waitPlayers** and **waitResults**. The difference in this phase is that, after the reception of every result, the losing player is removed and the winning player is saved in a list with the corresponding next table: that is the same table if the player is on an even number table, otherwise the previous one. After receiving all the results, a message containing the destination table chosen in the previous step is sent to each player and all the unused tables are removed from the simulation. At the end of the top 8, the results of this final part of the tournament are written in the **Top Ladder** box contained in the Statistics view.

2 Single Tournament Experiment

The **Single Tournament Experiment** was made to model a paper or online Magic: the Gathering tournament. As already said, the paper tournaments differ from the online ones in rules about game time. In this experiment is thus possible to choose the paper setting instead of the online one: two players can draw a match if this is taking longer than the time limit of 50 minutes. Even though it is clear that a single run of the simulation of a tournament does not provide statistically relevant

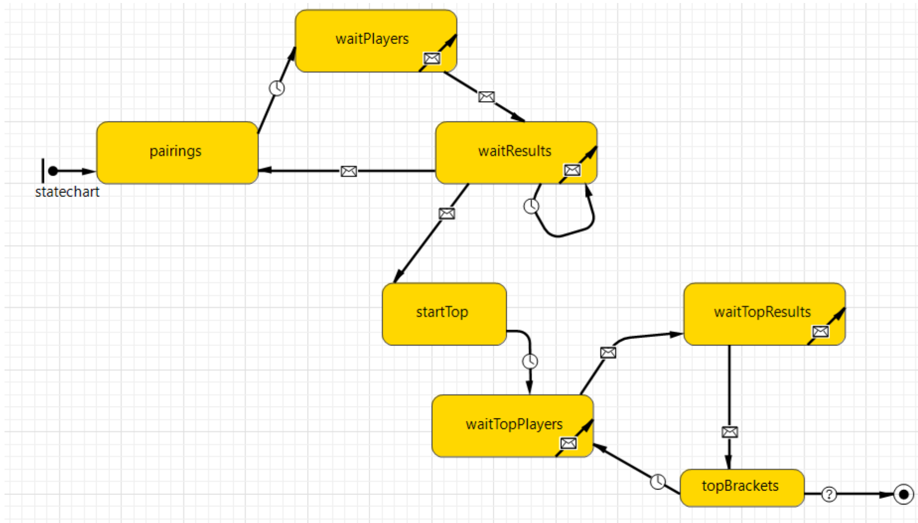

Figure 6: The Companion Agent

results, this experiment was made to show firstly how the tournament develops through time (while in the Monte Carlo experiment this is not observable). Secondly, while in the Monte Carlo experiment the final result of the top 8 is not shown (because of space matters), in the Single Tournament Experiment the top 8 happens and its results are recorded. Lastly, the single tournament permits to focus on a single player and to visualize its results.

From this experiment, we would like to gain insights about the probability of obtaining good results in a Paupergeddon-like competition.

In the initial view, the user is asked to upload a matchup and a metashare file in the model. They can also choose the number of participating players and the number of best performing decklists to visualize.

Pressing the button starts the model. The dimensions of the simulation view are computed to accomodate all the tables keeping always the same ratio between rows and columns.

When the tables are placed, the chosen number of players moves from outside the view to the assigned tables and the tournament starts. The number of rounds is simply computed as $\log_2 n\_players$. In this experiment, the user is able to navigate through three different view, a 2D one, a 3D one and a statistics view. While the 2D and 3D ones only shows the ongoing of the tournament, the statistics one contains the simulated metashare for the experiment, shown through a pie chart, and the swiss and top ladder.

This experiment, at the end of the swiss rounds, also simulates the top 8 and writes the results in a box in the Statistics view, next to the swiss ladder, as it is possible to observe in the figure 9.

3 Monte Carlo Tournament Experiment

The term Monte Carlo refers to a wide family of **estimation methods** based on the use of pseudorandom numbers. This approach relies on repeated random sampling to obtain numerical results.

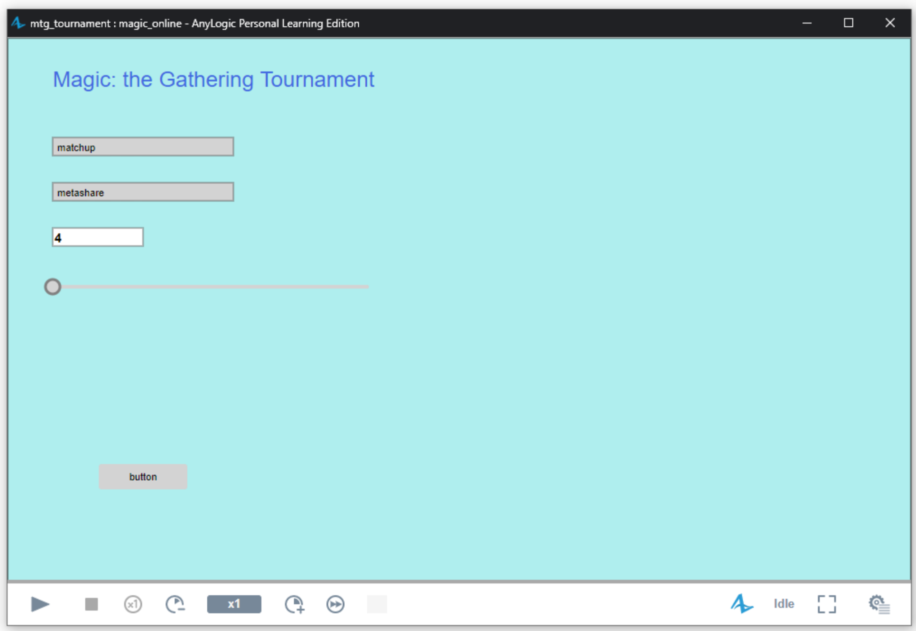

Figure 7: The initial screen of the experiment

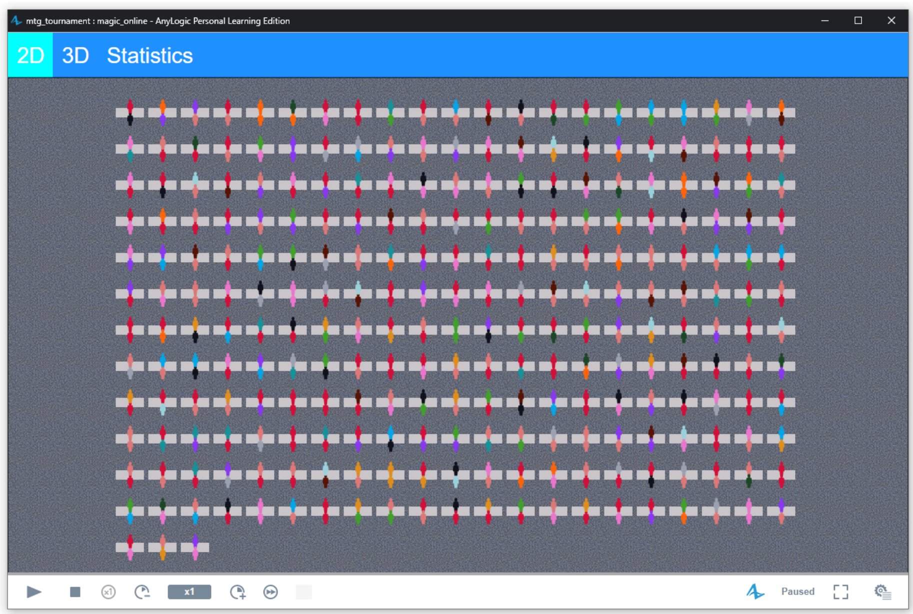

Figure 8: The 2D view of the experiment

Monte Carlo methods vary, but tend to follow a particular pattern:

- defition of a domain of possible inputs;
- generation of inputs randomly from a probability distribution over the domain;

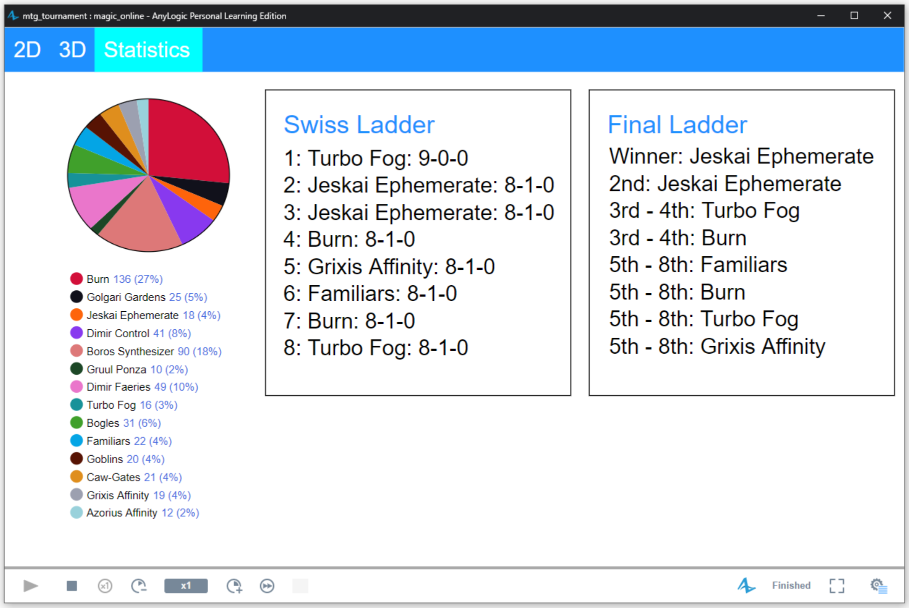

Figure 9: The Statistics view of the experiment

- deterministic computation of the outputs;
- aggregation of the results.

The **Monte Carlo Tournament Experiment** was made to model numerous Magic: the Gathering tournaments based on the same metagame. Starting from a fixed number of players, of size of the top and number of turns, this experiment takes as inputs a metashare (definition of a domain of possible inputs) and a matchup Excel files. Then, it computes a probability distribution from the metashare Excel file and associates a deck to each player, extracting each one of them from this distribution (generation of inputs randomly from a probability distribution over the domain). Hence, it simulate the swiss rounds (deterministic computation of the outputs).

For each of the 10.000 simulations of the tournament, this experiment reaches the end of the swiss and extract the list of the 8 decks that performed better. Then, it updates the pie chart adding, for each deck, the number of occurences that reached the top in this iteration. Then, it updates also the relative frequency of each one of them (aggregation of the results).

The final output of the simulation can be observed in 10

4 Key System Parameters

The main **key system parameters** are listed below:

- the **matchup** Excel file;
- the **metashare** Excel file;
- the number of players participating to the tournament;

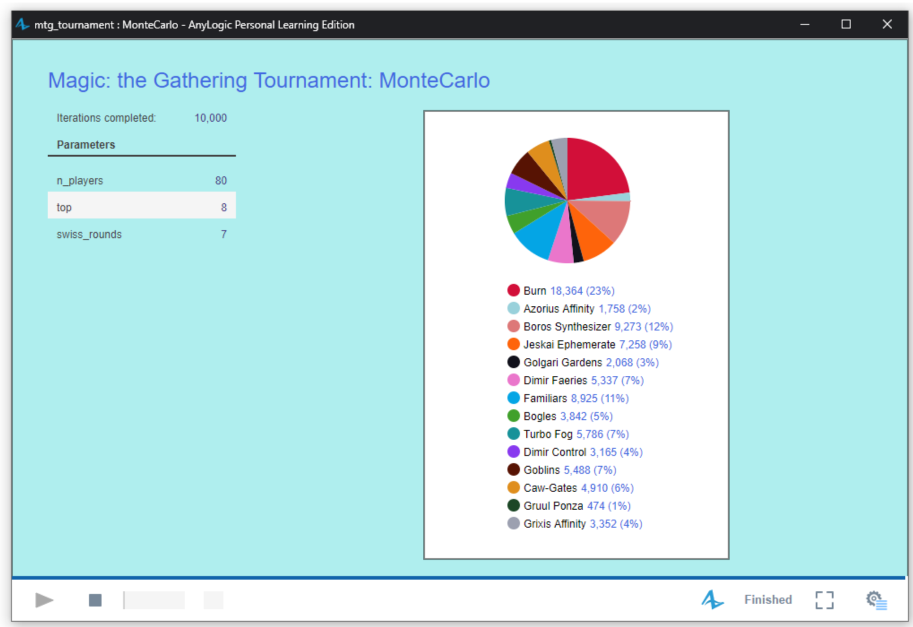

Figure 10: The view of the Monte Carlo experiment

- the timed boolean;

4.1 Metashare

The **metashare**1 Excel file contains three columns of data:

- **decklist**: represents the names of various decks;
- **presence**: indicates the number of occurrences or popularity of each deck;
- **id**: assigns a unique identifier to each deck.

The table includes 14 entries, where each row provides the deck name, its presence value, and a corresponding ID. For example, the deck Burn has a presence value of 260 and is assigned the ID 1, while Gruul Ponza has a presence of 31 and an ID of 14.

Decklist	Presence	ID
Burn	260	1
Boros Synthesizer	193	2
Dimir Faeries	118	3

Table 1: An example of the decks metashare

4.2 Matchup

The **matchup**2 Excel file contains a matrix that shows the winrates of different decks against each other. The rows and columns represent specific decks, while the values in the matrix indicate the percentage winrate of the deck in the row against the deck in the column.

	Burn	Boros Synthesizer	Dimir Faeries	Dimir Control
Burn	50.00	52.50	48.00	51.50
Boros Synthesizer	47.50	50.00	51.70	53.20
Dimir Faeries	52.00	48.30	50.00	59.40
Dimir Control	48.50	46.80	40.60	50.00
Caw-Gates	48.90	56.10	62.80	60.80

Table 2: An example of the decks matchups

5 Key Performance Indicator

- the **swiss ladder**;
- the **top ladder**;
- the **top decks frequency** in the Monte Carlo experiment.

IV Experimental What-if Analysis

1 MTGO

Since each Paupergeddon is subject to a different metagame, a statistical analysis of a single event is unlikely to yield a result similar to what actually occurred. Therefore, for this what-if analysis, we chose to analyze the results of the online Challenges in the period from April 1, 2024, to May 12, 2024, which is the period immediately following the Paupergeddon Winter Edition Lecco 2024 up to the day before the banlist of May 13. This period was free from the release of new sets, and its metagame is considered stable. This choice was made purely to avoid fluctuations caused by the tuning or discovery of new decks and strategies.

Specifically, we extracted data from all tournaments with more than 32 players held online on the MTGO platform[2]. This approach is due to the greater availability of online data compared to paper records and the difference in the implementation of the game in digital format: online, each player has their own clock indicating the total time they have available. This difference from the paper format is fundamental in preventing the so-called **slow play**, which is a penalizable practice where the leading player (presumably the winner of the first game) plays slowly to run out the opponent's available time. While in paper tournaments a judge can penalize a player practicing this strategy, online slow play only leads to the depletion of one's own clock time and thus to a loss. The fact that running out of available time results in a loss also implies that there are no draws in MTGO, making the overall structure simpler to model.

From these websites[3, 4], we extracted a table3 of win percentages for each matchup among the 14 most represented decks in the top 8 of the analyzed tournaments. Additionally, we counted the frequency with which each list was able to reach the top 8 of a tournament.

Deck	Count	Relative Frequency (%)
Burn	51	17.23
Boros Synthesizer	32	10.81
Dimir Faeries	25	8.45
Jeskai Ephemerate	23	7.77
Caw-Gates	22	7.43
Familiars	21	7.09
Golgari Gardens	17	5.74
Goblins	17	5.74
Dimir Control	14	4.73
Grixis Affinity	9	3.04
Turbo Fog	9	3.04
Izzet Terror	7	2.36
Ponza	5	1.69
Dredge	5	1.69
Bogles	4	1.35
Mono U Terror	4	1.35
Cycle Storm	4	1.35
Rakdos Madness	4	1.35
Mono U Faeries	4	1.35
Altar Tron	3	1.01
Walls	3	1.01
Orzhov Blade	3	1.01
White Weenie	2	0.68
Azorius Affinity	2	0.68
Flicker Tron	1	0.34
Tribe	1	0.34
Poison Storm	1	0.34
Jeskai Glitters	1	0.34
Turbo Initiative	1	0.34
Izzet Faeries	1	0.34
TOTAL	**296**	**100.00**

Table 3: Deck counts and relative frequencies in the MTGO Challenges top

Before proceding with the analysis, it's important to consider that a few decks which performed well in the collected data were so poorly represented in the metashare that the matchup data were totally unreliable. In fact, often these decks played against only a few of the other ones and only for a small number of matches. Therefore, we decided to remove their frequency in the final top count and redistribute their frequency to the other lists. Also, while the highlighted decklists may seem quite a few, they actually cover only a fraction of the total participating players and the 15% of the top decks.

The results of the Monte Carlo experiment are shown below 4, 5:

We decided to plot these results to better visualize the comparison 11.

Deck	Count	Frequency(%)
Burn	51	20,31
Boros Synthesizer	32	12,74
Dimir Faeries	25	9,96
Jeskai Ephemerate	23	9,16
Caw-Gates	22	8,76
Familiars	21	8,37
Golgari Gardens	17	6,77
Goblins	17	6,77
Dimir Control	14	5,58
Grixis Affinity	9	3,59
Turbo Fog	9	3,59
Ponza	5	1,99
Bogles	4	1,59
Azorius Affinity	2	0,80
TOTAL	**251**	**100,00**

Table 4: Deck occurrences and frequencies in the MTGO Challenges top 8

Deck	Count	Frequency (%)
Burn	18465	23,08
Boros Synthesizer	9245	11,56
Jeskai Ephemerate	7520	9,40
Familiars	6613	8,27
Goblins	5913	7,39
Turbo Fog	5795	7,24
Dimir Faeries	5378	6,72
Caw-Gates	5014	6,27
Grixis Affinity	4228	5,29
Bogles	3866	4,83
Dimir Control	3177	3,97
Golgari Gardens	2530	3,16
Azorius Affinity	1768	2,21
Ponza	488	0,61
TOTAL	**80000**	**100,00**

Table 5: Deck counts and relative frequencies in the Monte Carlo experiment

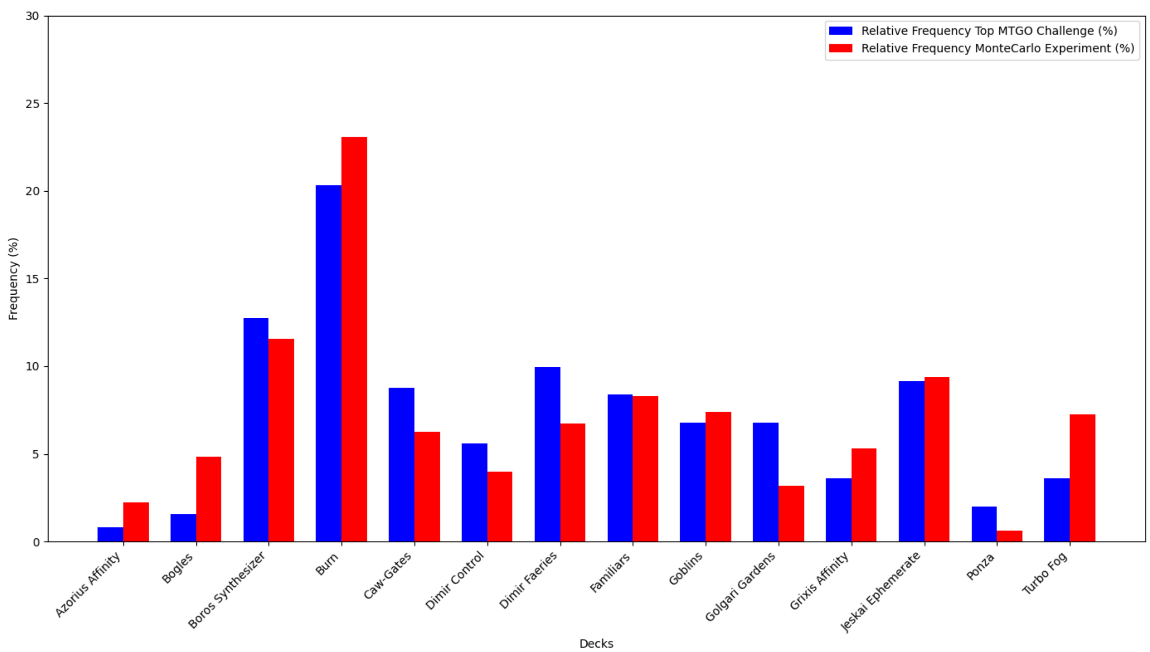

Figure 11: Comparison of decks frequency

V Conclusion

As it is possible to notice from the histogram above, there are quite some differences between the percentages of the decks which help in addressing some problems in the simulation itself.

Since some winning decks were not included in the analysis (as already discussed previously), the missing percentages of their frequency will be naturally distributed between the remaining ones. However, this redistribution will not be balanced, of course, and this is reflected on the results. As it is possible to notice, while it may seem that some decks perform way worse (Golgari Gardens, Dimir Faeries and Caw Gates) while other perform way better (Turbo Fog and Grixis Affinity) this is easily addressable to the decks that made results in the online Challenges that were not registered in the data. In fact, exempli gratia, Dredge and Rakdos Madness are two horrendous matchups for Turbo Fog and Grixis Affinity while they are really manageable for the others, with the exception of Golgari Gardens. This deck lost one bad matchup (Rakdos Madness) and a balanced one (Dredge) while it lost way more good matchups (Izzet Terror, Mono U Terror, Cycle Storm, Mono U Faeries, Altar Tron, Walls Combo, Orzhov Blade and White Weenie). This reflected on its results.

References

[1] Reid Duke. *Tempo*. Accessed: 2024-07-24. 2014. URL: `https://magic.wizards.com/en/news/feature/tempo-2014-09-22`.

[2] Wizards of the Coast. *Decklists*. Accessed: 2024-07-24. 2023. URL: `https://www.mtgo.com/decklists`.

[3] MTG Meta. *Pauper Metagame*. Accessed: 2024-06-10. 2023. URL: `https://mtgmeta.io/metagame/pauper?e=90`.

[4] Qonfused. *MTG Metagame*. Accessed: 2024-06-10. 2023. URL: `https://observablehq.com/@qonfused/mtg-metagame`.