# TAG: Pandemic Competition

Raluca D. Gaina, Martin Balla
Game AI Group, Queen Mary University of London, UK
{r.d.gaina, m.balla}@qmul.ac.uk

***Abstract*—Cooperation between humans and AI is an area of research explored more frequently in recent literature. Yet, environments used for this purpose are generally lacking in complexity. In this paper, we describe the first Tabletop Games Framework (TAG) competition designed around the Pandemic: a cooperative board game where players aim to cure the world of disease. We discuss the many AI challenges introduced through this environment, detail the competition setup, present baseline results for sample AI players and explore the game parameter space for interesting insights, such as the most dominant player roles: the Scientist and the Medic.**

***Index Terms*—Tabletop Games, Board Games, Pandemic, General Game Playing, Monte Carlo Tree Search, Game Analytics, MAP-Elites, N-Tuple Bandit Evolutionary Algorithm**

## I. Introduction

Modern tabletop games encompass a wide range of environments, with deep challenges for AI research. These are relevant to game-playing AI through imperfect information, strategic planning and dynamic action spaces, as well as to procedural content generation, due to the multitude of content often included in such games, from tokens to dice, cards, game boards and more. The Tabletop Games Framework (TAG) [1][1] provides access to various modern tabletop games implemented in Java. TAG aims to encourage and support research into a diverse set of environments. Games included range from simple abstract pen-and-paper games such as ”Tic-Tac-Toe” and ”Dots and Boxes”, to card games such as ”Exploding Kittens” [2], ”Poker” [3] and ”Dominion” [4], to strategic board games such as ”Settlers of Catan” [5] and ”Terraforming Mars” [6], [7].

One of the TAG games is Pandemic [8]: a collaborative 2-4 player game where players have to work together in order to cure 4 types of diseases which spread throughout the world, while racing against time. Designed by Matt Leacock and inspired by the SARS epidemic of 2003 – though the theme is still intriguing for current times of the Covid-19 pandemic – the game was reported by publisher Z-Man Games to have sold over 5 million copies worldwide by 2021[2]. Further, the game has won many awards and is one of the most successful cooperative games [9].

Due to its success, several human tournaments have been run for Pandemic[3]. Given its cooperative nature, a set of rules for competitive settings was put in place to allow for fair comparisons of teams of players by reducing the randomness affecting the games (Pandemic Survival series). We take inspiration from these events and bring the Survival series forward as an AI challenge, with the main aim of improving upon game-playing AI for Pandemic.

Its cooperative nature makes the game currently unique within the TAG framework, and also one of the most difficult games which the general-purpose AI players included in TAG cannot yet solve. With communication between players disabled in our setting, we focus on the main problem of adaptability and coordination of individual players in the complex Pandemic environment. Previous environments and competitions exploring player cooperation centred around the card game Hanabi [10], [11], having seen great progress in AI strength through the means of the competition. Pandemic adds more strategic depth and larger state and action spaces, raising the bar once again.

The summarised contributions of this paper are two-fold: first, we describe the first TAG competition which will be hosted at the IEEE Conference on Games 2022, using the board game Pandemic as a test-bed; we perform deep analysis of the environment and challenges present for AI game-playing research. Second, we discuss ongoing research into finding a spectrum of Pandemic game parameter configurations which allow for interesting games of varying difficulty.

## II. Related work

Pandemic was previously proposed as a challenge for human-AI cooperation by Sauma Chacón and Eger [12]. They acknowledge the limited cooperative environments for AI and discuss the difficulty of Pandemic resulting from its rule interactions. They largely focus on the problem of coordination with human players, when communication is limited: therefore, the AI players are required to plan strategies without knowledge of what the humans' goals may be, with a further need to balance between short-term and long-term objectives. The authors propose an approach focused on plan recognition to tackle the coordination issue (later expanded in [13]): the AI player attempts to infer possible future goals for the other player based on their action history, and adapts its own plans as a result to maximise the group outcome.

The authors expand their focus in follow-up work [14]. They use A* planning for effective search through the state space, selecting goals through a rule-based approach to inform the agent's priorities. They combine this with Monte Carlo simulations for trying out different potential card draws, and a complex heuristic for calculating the value of states based on several features: the agent's distance to highly infectious areas which need to be treated, the distance to the closest research

[1]http://tabletopgames.ai
[2]https://tinyurl.com/pandemic-5-million
[3]https://tinyurl.com/pandemic-survival-rules

station, the value of cards in hand and discarded, and the state of infections around the world (all concepts will be detailed later in the paper). Their results show their agent to be quite a strong player, managing to win up to a third of the games in the easiest setting tested, compared to $86.3\%$ reported by human players[4]. Particularly, their insights into the potentially best player role combinations are taken forward into our work to inform our competition configurations.

In experiments with human players, Sauma Chacón and Eger [15] report a correlation between the perceived skill level for the agent and the perceived helpfulness, aligning with its plan recognition abilities and highlighting the need for better coordination in such games.

Sfikas and Liapis [16] take a different approach, using a Rolling Horizon Evolutionary Algorithm [17] to create plans for all agents, with a centralised approach to decision-making. They further make use of abstractions to simplify the action selection process, by introducing macro-actions which group several low-level decisions into a more abstract goal (e.g. travel to a city and treat the infection there). They report up to $40\%$ win rate, especially with shorter horizons (shorter-term planning, allowing for more iterations of the algorithm and more accurate statistics). Later, they highlight that 2 of the 3 loss conditions are most likely to trigger, with short horizons most likely to lead to a scenario where players efficiently contain the infection, but run out of time before they cure all disease [18].

Although our own setup proposed is most similar to the work by Sauma Chacón and Eger and as presented previously by Gaina et al. [19], we expect a variety of methods inspired by previous work to be submitted to our Pandemic competition.

## III. Pandemic the Board Game

Pandemic is played with $2-4$ players who share the same goal: curing all diseases. We recommend the reader to check the full game rules[5]. The game is played over a board represented by a graph of major cities in the world (such as London, Delhi, Tokyo or Buenos Aires), separated into 4 regions with colours matching the 4 diseases in play. The diseases infect cities more and more over time, as decided by *infection deck* cards, with a limit as to how infected a city can get before causing an *outbreak* into adjacent cities. We refer to adjacent cities as cities connected by an edge in the board graph.

Players execute up to $4$ actions during their turn, and can move between cities, placing research stations, removing or curing diseases, and playing cards from their hands to execute actions, while making the most of their unique player role abilities (7 player roles available in total). Players draw cards at the end of their turn from the *player deck*, which includes city cards (with names matching cities on the board), Event cards (with special abilities, detailed below) and Epidemic cards (which trigger epidemics, detailed below). New cards are drawn from the *infection deck* afterwards, leading to new infections spreading throughout the world. To cure a disease, players must be at a research station and play 5 cards matching the colour of a disease not already cured. Diseases can be eradicated if they are cured and all infections are cleared from the board, which means no new infections for those diseases will appear on the board in the future.

There are multiple ways to lose the game: disease cubes used to infect cities may run out; the player deck of cards may run out; or there may have been too many outbreaks.

### A. Action space

The action space is dynamic: actions available to players vary every turn depending on the game state. The action types are as follows:

- **Move**: Move a player's pawn to an adjacent city. Cities with research stations are also considered adjacent to each other. Players may discard cards in hand to either move to the city named on the card, or to any city, if the card discarded matches the player's current location.
- **Build a Research Station**: Discard a city card in hand that matches the player's current location to place a research station at the current location. If the limited supply runs out, this action will instead move any other already built research station to the current location.
- **Treat Disease**: removes 1 infection from the current location of the player. If the disease is cured, all infections of that colour are removed with a single action.
- **Share Knowledge**: if 2 players are in the same location, they can trade (or exchange) 1 card which matches the city they are currently in.
- **Discover a Cure**: if the player is at a location with a research station, they may discard 5 city cards matching the colour of a disease not already cured to cure it.

The highest number of actions available for one decision we have observed so far[6] is 265, and the lowest is 11.

### B. Survival rules

In our competition setting we use Pandemic Survival rules, similar to human tournaments for this game. Several limitations are imposed as a result on the Event cards available. We limit the player roles to only 5 out of 7 options as well, removing "Contingency Planner" (who can play discarded Event cards; since we only use 2 such cards, the role has diminished benefits) and "Researcher" (who can trade cards more freely; sample AI players are not able to use this role to its full advantage, so we keep it turned off for this first edition of the competition). We use only 2 event cards: **Airlift** (allows to move any pawn to any city) and **Government Grant** (allows to add a research station to any city). Player roles either have an *effect* that modifies how a basic action is played, or how a game rule is applied; or allow the player to perform a special *action*:

- **Dispatcher**: *Effect* – may perform move actions with any player's pawn. *Action* – move any pawn to a city which already contains another pawn.

[4]https://tinyurl.com/pandemic-statistics
[5]https://tinyurl.com/pandemic-rules-pdf
[6]http://www.tabletopgames.ai/wiki/games/game_stats.html

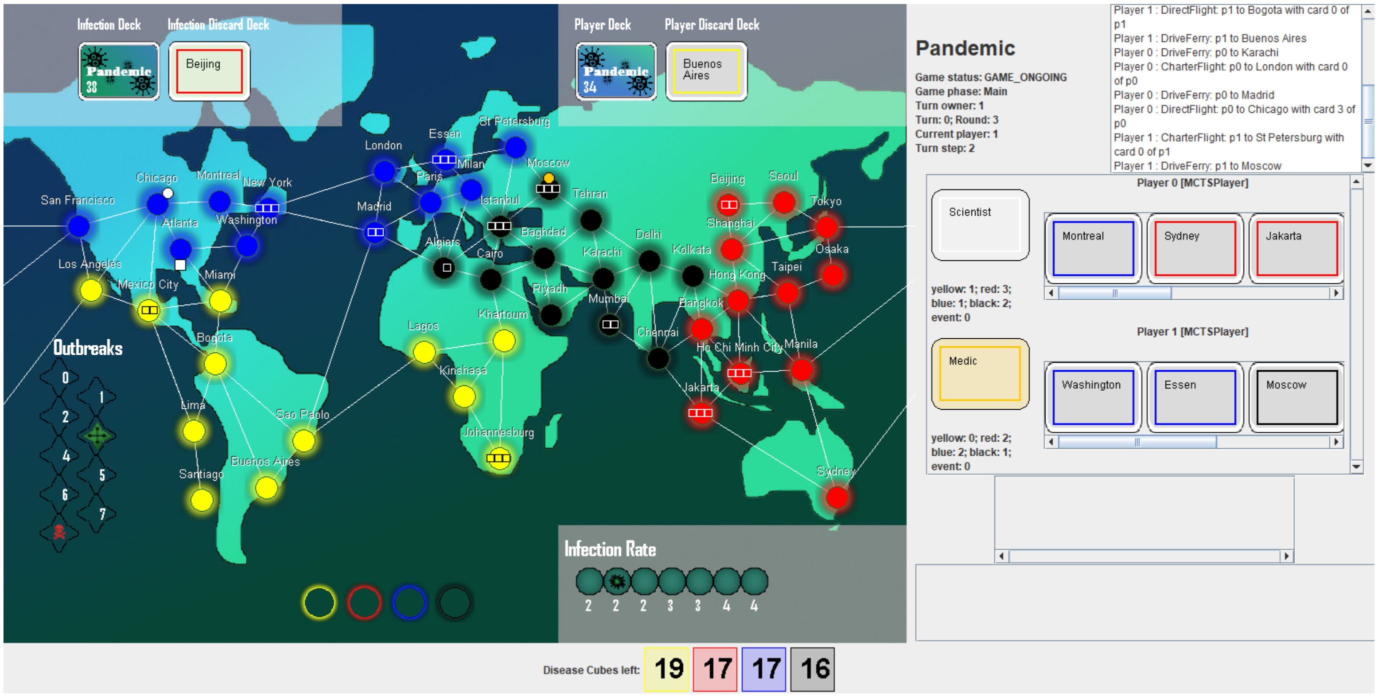


Fig. 1. Visualisation of the board game Pandemic in TAG

- **Medic**: *Effect* – remove all infection of one disease when treating disease in a city. *Effect* – infections from cured diseases are automatically removed (and further such infections blocked) from the city where the player is.
- **Operations Expert**: *Effect* – does not require a card to build a research station. *Action* – discard any city card in hand to move from a city with a research station to any other city.
- **Quarantine specialist**: *Effect* – prevents disease from infecting the city the player is currently in, and all connected cities.
- **Scientist**: *Effect* – requires 1 less card to cure disease.

## C. State representation

The Pandemic game state is made up of several components, the relevant ones for the competition being detailed next. **Graph board**: a collection of board nodes (cities), where each has a list of its neighbours, players currently in the city, level of infection from the 4 diseases, and whether a research station was built there or not. The graph information is read in automatically from a JSON file. **Epidemic flag**: informs whether an epidemic card was drawn by a player, triggering the epidemic branch of the rule graph (the infection rate increases; the city card at the bottom of the infection deck fully infects the corresponding city and is discarded; and all discarded infection cards are shuffled and placed back on top of the infection deck). **Research station locations**: a list of cities where research stations have been built. **Areas**: one per player, containing their role card and cards in hand. An additional area stores the following: the *player deck*, the *infection deck*, counters for the number of infection cubes remaining for each disease, the state of each disease (not cured, cured, or eradicated), the outbreak counter and the infection rate counter (which decides how many cards should be drawn from the *infection deck* at the end of each player's turn).

AI players receive a copied Java object containing all of this information, and they may access and modify the information freely. The copy received will be largely accurate to the real current game state, with the exception of the face-down decks of cards (*infection deck* and *player deck*), which are hidden information and are randomly shuffled in copies sent to players.

## D. Implementation specifics

The game is implemented within the Tabletop Games framework [1], using a modular graph-based structure for its rule definition. Node types included in its implementation are as follows. **Player action node**: expects a player action to be sent and executes it appropriately. **Condition node**: checks a condition (e.g. if enough actions have been executed by the current player, or they decided to end their turn). **Rule node**: executes one rule of the game (e.g. infect cities, draw cards for players, or decide next player).

The game engine then keeps track of the whole rule graph structure and its current position in the graph, executing rules appropriately. Each player receives this game model, and they may perform simulations of possible futures by passing an action and a game state through the model, which will return a possible next state resulting from the applied action. Internally, the rule graph is traversed from the current position until a node requiring a player decision is reached; therefore, multiple rules may be applied in-between player decisions.

Further, in our implementation, Event cards in the game are allowed to be played as regular actions during a player's turn only. However, they do not count towards the limit of 4 actions per turn (they remain free actions, but they are restricted to a player's turn, unlike original game rules which allows playing event cards at any point).

Lastly, we define a generic heuristic function which approximates the *goodness* of a game state. The value returned for any game state is a linear combination of several features, with tunable weights allowing to adjust how much a particular feature influences the final result. The weight values presented here are manually adjusted from expert knowledge and preliminary experiments. Equation 1 describes the resulting function, where $s_t$ is the current game state at time step $t$, $\phi_i$ is the value of feature $i$ and $w_i$ is the corresponding weight for feature $i$. The heuristic simply returns 10 if the game is won, or $-10$ if the game is lost.

$$h(s_t) = \sum_{i=1}^{7} \phi_i \times w_i \tag{1}$$

Features included are detailed next. $\phi_1$: the number of discovered cures, $w_1 = 0.6$. $\phi_2$: the number of disease cubes remaining (not on board), $w_2 = 0.1$. $\phi_3$: the number of player cards remaining in the draw pile, $w_3 = 0.1$. $\phi_4$: the number of cards in the current player's hand, $w_4 = 0.2$. $\phi_5$: the number of outbreaks which have occurred in the game so far, $w_5 = -0.2$. $\phi_6$: the number of research stations on the board, $w_6 = 0.2$. $\phi_7$: whether the player is currently at a research station or not, $w_7 = 0.3$.

## IV. Game Analysis & AI Challenges

Pandemic is a complex and deeply strategic board game, relying on not only individual high level of skill, but also player coordination to achieve success. As such, we identify several challenges which highlight Pandemic as an exciting environment for AI research.

**Coordination.** Pandemic is a cooperative game played by multiple players who must together achieve the winning condition. As a result, coordination of plans and adaptability to each other's plans and behaviours are key for a favourable outcome. This challenge is the main focus of the work by Sauma Chacón and Eger [12].

**Large and dynamic action space.** There are many actions available to players at any one time (as many as 265 options to choose from for one decision have been recorded so far). The resulting large branching factor raises the problem of focus, prioritisation and abstraction: which actions should be tried first, which can be directly discarded as not relevant in the current game state? Further, the dynamic nature of the action space means an adaptive algorithm must be used, which does not rely on fixed inputs or strong association between action index and actual action effect. Previous approaches which have shown great results in fixed-size action spaces, such as Rolling Horizon Evolutionary Algorithms with crossover [17], are not directly applicable here due to the disconnect between agent interface and action effects.

**Large state space.** Pandemic features a large state space, with a large world graph where each node contains detailed information about many other components in the game. A low bound estimate is $1.07 \times 10^{158}$ possible game states in the original game, with 170 components per state recorded in our implementation – each with several varying properties. This aspect adds another layer of complexity and again challenges the focus of players: being able to pinpoint the most important parts of the state space (e.g. the region in the world which has seen most infections and is in danger of resulting into outbreaks) is critical to performing well in the game.

**Conflicting objectives.** The game is designed with several conflicting objectives in mind, lending itself quite naturally to multi-objective optimisation approaches. The players must complete their winning condition, while juggling many other aspects of gameplay to avoid the many losing conditions: they must be able to manage cards effectively (in their own hand due to the limit of cards they may hold at one time; but also between each other, and trading efficiently); move around the board purposefully; take care of particularly infectious areas; build research stations strategically to allow for more flexibility and maximise winning chances; as well as attempting to maximise individual player effectiveness (e.g. avoiding grouping without an immediate gain).

**Asymmetric player roles.** Each player is assigned a unique role with special effects or actions they may perform. Players must be able to adapt their gameplay and strategy based on the roles currently in play, and use their abilities to the fullest. Some roles may be more beneficial than others in certain circumstances, thus it is important to learn how to manage potentially weaker player roles, or roles for whom their abilities mean a much smaller subset of actions is relevant (e.g. the Quarantine Specialist should focus more on protecting infectious areas, and less on travelling across the board).

**Long-term planning.** Unlike many other games used in research, Pandemic is a long board game, which takes human players hours to complete. This can be sped up in a digital implementation, where AI players are able to calculate possibilities and make decisions much faster than humans. However, there remains a considerable slow-down limiting the resources which can be spent efficiently for high performance. Most importantly, the large number of decisions required from AI players during the course of a game means that it becomes difficult to perform searches far enough into the future to obtain useful information for approaches such as statistical forward planning.

**Uncertainty and hidden information.** Finally, hidden information and the resulting uncertainty is a long-standing challenge for AI players, which appears often in many modern game environments. In Pandemic, the decks of cards which infect cities across the board, and which allow players to perform their actions, are both shuffled face-down decks of 48 and 50 cards, respectively. There are approximately $3.77 \times 10^{125}$ possible starting states in our setup, if player roles are fixed.

Being able to adapt to unexpected results, planning for the worst, and handling dangerous game situations efficiently is again of key importance.

## V. Competition Framework

The objective for the competition is to submit an agent which is capable of achieving high level of play across several versions of the Pandemic board game, with varying game parameters, aimed at providing different levels of difficulty. The framework code is open-source and all competition details are available online[7]. This includes data logging, video recording and optimisation tools usable by participants to analyse and improve their entries.

As previously mentioned, we are employing the Pandemic Survival rules; the same random seeds and game parameter configurations will be used to compare different entries. Due to its digital implementation, we are able to tweak some of the game parameters to obtain different game configurations of varying difficulty.

Participants submit 1 algorithm, in a `zip` archive including all files necessary to run, via email to `tagframework@gmail.com`. A $100ms$ decision making budget is imposed for all submissions. The same algorithm will be used for each player in (2-4)-player Pandemic games within the Tabletop Games framework. 4 training game configurations were made publicly available[8] to inform AI agent design, with $v0$ and $v2$ designed as *easier* variations (less cards needed to cure disease), and $v1$ and $v3$ as *harder* variations. All configurations have fixed player roles. We refer to parameters as in the original board game as *default* (with player roles randomly assigned at the start of the game).

We will test entries locally on 3 different game configurations, running 100 repetitions per submission per configuration, and rank all entries by total win rate, with several tiebreaks, following the official Pandemic Survival rules (averages across all games played): $\tau_1$ game duration in number of turns (minimum in winning games; maximum in losing games), $\tau_2$ maximum number of diseases cured, $\tau_3$ minimum number of outbreaks, $\tau_4$ minimum number of cities about to outbreak, $\tau_5$ minimum number of disease cubes on the map, and finally $\tau_6$ the maximum number of diseases eradicated.

The competition is hosted as part of the IEEE Conference on Games 2022 in August 2022, where the entry ranking highest across all test configurations will be declared the winner of the competition, and will receive $500 money prize sponsored by the IEEE Computational Intelligence Society. The results will be published on the competition website after the conference.

### A. Sample AI player performance

We select several of the general AI players included in TAG to present baseline results on several game configurations. Both OSLA and MCTS detailed below use the Pandemic heuristic function (see Section III-D) to evaluate game states.

TABLE I
Results for 2-player games, with 100 iterations per match. Players are ranked best to worst (top to bottom) using competition ranking criteria.

Default parameter configuration

| Player | Wins | $\tau_1$ | $\tau_2$ | $\tau_3$ | $\tau_4$ | $\tau_5$ | $\tau_6$ |
|---|---|---|---|---|---|---|---|
| OSLA | 0 | 47.4 | 0 | 8 | 16.2 | 37.5 | 0 |
| Random | 0 | 46.4 | 0 | 8 | 13.9 | 32.7 | 0 |
| MCTS | 0 | 46.2 | 0.02 | 8 | 13.1 | 31.9 | 0 |

Parameter configuration v0

| Player | Wins | $\tau_1$ | $\tau_2$ | $\tau_3$ | $\tau_4$ | $\tau_5$ | $\tau_6$ |
|---|---|---|---|---|---|---|---|
| OSLA | 100 | 28.8 | 4 | 0.02 | 1.63 | 13.1 | 0.33 |
| MCTS | 100 | 61.7 | 4 | 0.07 | 1.89 | 15.3 | 0.28 |
| Random | 0 | 153 | 0.47 | 1.53 | 11.8 | 79.3 | 0.01 |

Parameter configuration v1

| Player | Wins | $\tau_1$ | $\tau_2$ | $\tau_3$ | $\tau_4$ | $\tau_5$ | $\tau_6$ |
|---|---|---|---|---|---|---|---|
| Random | 0 | 46.7 | 0 | 8 | 14.9 | 34.4 | 0 |
| MCTS | 0 | 44.4 | 0.03 | 8 | 13.4 | 32.5 | 0.28 |
| OSLA | 0 | 29.1 | 0 | 8 | 14.1 | 33.3 | 0.01 |

Parameter configuration v2

| Player | Wins | $\tau_1$ | $\tau_2$ | $\tau_3$ | $\tau_4$ | $\tau_5$ | $\tau_6$ |
|---|---|---|---|---|---|---|---|
| OSLA | 100 | 38.3 | 4 | 0 | 2.13 | 36 | 0.11 |
| MCTS | 99 | 56.5 | 3.99 | 0.1 | 2.36 | 39.7 | 0.27 |
| Random | 0 | 151 | 0.48 | 4.5 | 15.3 | 95.8 | 0 |

Parameter configuration v3

| Player | Wins | $\tau_1$ | $\tau_2$ | $\tau_3$ | $\tau_4$ | $\tau_5$ | $\tau_6$ |
|---|---|---|---|---|---|---|---|
| Random | 0 | 47.7 | 0 | 8 | 14.3 | 35.9 | 0 |
| MCTS | 0 | 46.9 | 0.01 | 8 | 14.3 | 33.9 | 0 |
| OSLA | 0 | 32.6 | 0.02 | 8 | 16.1 | 37.8 | 0 |

- **Random**: Chooses actions to play uniformly at random out of those available.
- **One-step Look Ahead** (OSLA): Simulates the effect of all possible actions on the current game state. The action selected is that which leads to the most promising next state.
- **Monte Carlo Tree Search** (MCTS) [20]: Builds a tree of possible actions, choosing to explore the most promising parts of the search space first. Statistics about the most promising actions are recorded and used to ultimately decide which action to choose when the budget is exhausted. The search restarts at the next decision-making point with an empty tree. For details of MCTS parameters the reader is recommended to check the TAG documentation[9] and repository for default parameters[10].

Previous results for Pandemic in TAG [1] reported $0\%$ win rate for all agents. We expand previous results here with detailed analysis into their gameplay and performance, pertaining to key aspects as evaluated in the competition. Results for all public game configurations used in the competition, as well as the default parameter settings (as per the original board game) are presented in Table I.

In these initial experiments, OSLA appears to win games more efficiently than both MCTS and random in the easier game configurations. It manages to better keep infections under control, generally ending up with a better state of the board regarding the number of infections present and dangerous areas. We hypothesise that this is largely due to it finishing games

[7] http://www.tabletopgames.ai/competition/cog2022/
[8] https://tinyurl.com/pandemic-training-config
[9] http://www.tabletopgames.ai/wiki/agents/MCTS
[10] https://tinyurl.com/MCTS-params

much faster than MCTS, the game having less of a chance to spread the infection beyond control. This also happens in losing games – OSLA loses faster than MCTS.

On a closer inspection into gameplay styles emerging, OSLA appears very conservative with its cards and chooses to limit movement and rarely departs from its starting position. It relies instead on luck to draw the correct cards and makes use of the research station already built at the starting location to cure disease. As players are likely to be in the same location, they can also exchange cards more to easily gather the required number for curing disease; in fact, OSLA is 4 times more likely to use the Share Knowledge action during its games than MCTS. Due to the very short lookahead, OSLA is unable to see the advantage of moving around the board, therefore failing to control disease in more punishing game configurations.

On the other hand, MCTS performs many more movement actions (over $60\%$ of its actions are move actions, compared to $15\%$ for OSLA; half of these use cards in hand), the strategy resembling more that of human players. MCTS players spread out across the board more as well, visiting over $60\%$ of the cities throughout a game; approximately $40\%$ of the locations visited are infected cities, suggesting MCTS movement to be towards dangerous areas which should be visited and treated. However, MCTS is unable to manage its cards appropriately, and ends up discarding valuable cards which could be used for disease cures in order to execute its movement-heavy plans.

Despite using the exact same heuristic to inform their decisions, we see very different playstyles emerging due to the difference in planning methods and possible futures observed. In most cases, the players lose because of the outbreak loss condition triggering. However, there are cases for the random player in the 2 easier public game configurations ($v0$ and $v2$) where the game is lost due to no more player cards remaining in the deck. Overall, these findings suggest that efficient gameplay should avoid outbreaks, while maximising the benefit of each action taken to finish the game as fast as possible.

We finally note that changing the game parameters and overall difficulty (resulting from either reducing the pressure of loss conditions, or simplifying the win condition), our AI players managed to find situations not covered by the original ruleset. One such example is the need to add a $4^{th}$ loss condition: in very long games with low loss pressure, it could happen that the *infection deck* could run out of cards. A detailed study of edge-cases brought to light by different playing strategies is hereby noted as future work.

## VI. Experiments

Before concluding, we discuss a set of experiments employing MAP-Elites [21] and N-Tuple Bandit Evolutionary Algorithm (NTBEA) [22] to tune the game parameters automatically and explore the space of possibilities more in-depth.

### A. Map-Elites

Map-Elites is a Quality Diversity algorithm used to illuminate search spaces. Unlike other evolutionary algorithms, it keeps track of a variety of solutions which meet the required fitness functions, but which show interesting diversity in phenotypical (behavioural) space. The algorithm has been previously used to explore search spaces for game parameters [23], [24], agent parameters [25], level generation [26], game-playing [27] or heuristic weights [28], to name a few applications.

In our implementation, the algorithm runs for $1000$ iterations. It begins by randomly initialising 200 solutions ($20\%$ of the iterations), keeping the best ones in a 7-dimensional elite map. For each evaluation, 10 simulations are run, using OSLA to play 2-player games. A fitness function (detailed below) is used to give the parameter configuration a value, and 7 behaviours are tracked (detailed below). The cell in the elite map indexed by the behaviour values is filled in with the game parameter setting if either the cell is empty, or the fitness value previously recorded there is lower. After random initialisation, the process continues through evolution: 1 elite is randomly chosen from the map and mutated (parameter values are changed with $0.3$ probability). The new game parameter configuration is evaluated and inserted into the map as described above. The evolution concludes after the iteration budget is used up, returning the resulting elite map for analysis.

### B. Setup

*1) Game parameter space:* The parameter space for Pandemic is depicted in Table II. Not all of the parameters are modified, indicated with a dash symbol ('-') in the value range if fixed to the default value instead. The resulting search space size is 48,600 (2-player games) and 1,215,000 (4-player games). We note that with 1000 iterations we are able to only explore $2\%$ of the search space for 2-player games and $0.08\%$ of the search space for 4-player games. Future work will explore means of more efficient sampling of solutions for better coverage.

*2) Fitness function:* The fitness function used is the difference of the agent's win rate across all simulation games to a **target win rate**, bounded within [0,1]. We run 3 different sets of experiments, setting the target to $10\%$, $50\%$ and $100\%$ in order to explore a variety of difficulties.

*3) Behaviour Characterisation:* The behaviours recorded for each solution are averaged across the $10$ simulations performed during an evaluation. We record: $\beta_1$ the number of outbreaks that occurred during the game. $\beta_2$ the minimum number of disease cubes remaining. $\beta_3$ the number of cards remaining in the *player deck*. $\beta_4$ the number of cards remaining in the *infection deck*. $\beta_5$ the number of diseases cured or eradicated ($\beta_6$). $\beta_7$ the number of infection cubes on the board at the end of the game. All behaviour values are normalised and placed in bins in range [0.0, 1.0] with step of $0.2$.

### C. Results

We present a subset of the results in Figure 2. The rest of the results are made available online [11]. Overall, we observe low coverage of the entirety of the behaviour space ($0.18\%$), with the $10\%$ target win rate populating the least of the space available in the elite map. If we look at pairings of

[11]https://tinyurl.com/results-pandemic-cog-22

TABLE II
PANDEMIC PARAMETER SPACE.

| Idx | Parameter Name | Description | Type | Default Value | Value Range |
|---|---|---|---|---|---|
| 1 | nEpidemicCards | Number of epidemic cards in the Player deck | Numerical | 4 | 2, 4, 6 |
| 2 | loseMaxOutbreak | Maximum number of outbreaks before loss | Numerical | 8 | 6, 8, 10 |
| 3 | nCardsForCure | Number of city cards required to cure disease | Numerical | 5 | 2, 3, 4, 5 |
| 4 | maxCardsPerPlayer | Maximum number of cards in a player's hand | Numerical | 7 | 5, 7, 9 |
| 5 | maxCubesPerCity | Maximum amount of disease in a city before outbreak | Numerical | 3 | 3, 4, 5 |
| 6 | nInitialDiseaseCubes | Number of cube tokens in the game per disease | Numerical | 24 | 20, 24, 30 |
| 7 | nCardsDraw | Number of cards players draw at the end of turn | Numerical | 2 | 1, 2 |
| 8-11 | player0Role (or 1,2,3) | The role for players 0,1,2,3 | Categorical | Any | Scientist, Quarantine Specialist, Dispatcher, Medic, Operations Expert |
| 12 | nCubesEpidemic | Disease added to the city infected in an epidemic | Numerical | 3 | – |
| 13 | nInfectionsSetup | Number of steps during the infection setup | Numerical | 3 | – |
| 14 | nInfectionCardsSetup | Number of infection cards drawn during setup per step | Numerical | 3 | – |
| 15 | nCubesInfection | Infections added to a city on infection card draw | Numerical | 1 | – |
| 16 | survivalRules | Whether the survival rules are active or not | Boolean | True | – |

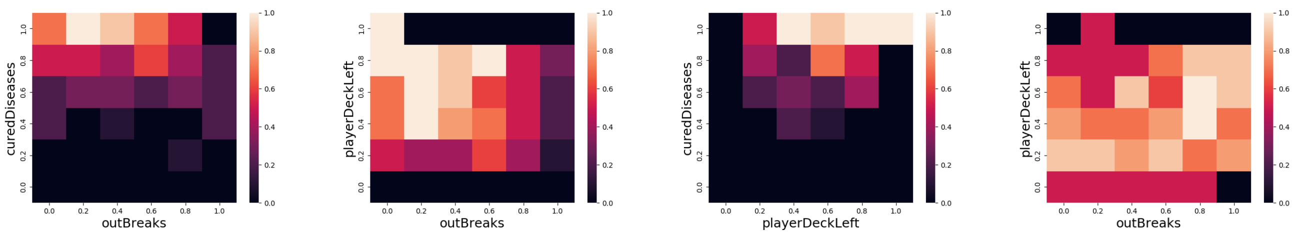


Fig. 2. Visualisation of 2D elite maps pairing 3 behaviours: $\beta_1$ outbreaks, $\beta_3$ cards remaining in the *player deck* and $\beta_5$ cured diseases (see Section VI-B3). Colours indicate fitness (the lighter the colour, the better the fitness). First 3 figures have target win rate 100%, the last figure has 50% target win rate.

the behaviours, we notice that $\beta_6$ is the behaviour for which variations are the hardest to find, as the agents do not manage to eradicate diseases often. We highlight in particular that games are generally easier to win early (as players cannot retrieve discarded cards, and it may happen that a game becomes unwinnable if enough cards matching one disease are discarded, a scenario likely to happen with low-skilled AI). However, MAP-Elites records games where the win appears later in the game (when under 40% of the cards in the *player deck* are left, which indicates timing). One such instance is the following combination of game parameters: {4, 10, 3, 9, 5, 20, 1, Operations Expert, Dispatcher} (order matches that in Table II). This is an easier variation, but using player roles which we analyse in the next section as not the best.

The last image in Figure 2 shows one example when optimising for 50% OSLA win rate. In this case, both all-wins and all-losses are given low fitness, which means maps are fuller but also have less variety in fitness, consistent with the baseline results presented in Section V-A. However, MAP-Elites is able to find some variants with high fitness (and therefore more balanced win rates, games of medium difficulty). The parameters which show up in these configurations most are 8 maximum outbreaks, 9 cards in hand maximum, 3 cards for cure and the Quarantine Specialist player role. This configurations favours OSLA's playstyle in particular, focusing on reducing the difficulty in curing disease, but loss conditions are kept as default, or do not have a big impact on the result.

### *D. Parameter statistics*

Lastly, we used NTBEA, a TAG built-in tool for noisy optimisation of agent, game and heuristic parameters, to gather statistics about the best and worst parameter settings. The same experimental setup as for MAP-Elites is employed.

Most notable in all experiments is the dominance of the Scientist player role, which allows to cure disease with 1 less card than normally required: this role achieves a fitness of $0.903 \pm 0.016$ as player 0 and $0.934 \pm 0.013$ as player 1 when optimising for 100% target win rate (maximum 1, minimum 0). The use of this role significantly reduces the difficulty of the game, with the Medic close behind (and the best partner for the Scientist), whereas there is no noticeable difference between the other player roles. Other parameters adjust the difficulty as expected (e.g. more outbreaks allowed before game end results in higher win rates). The $nCardsForCure$ parameter has the most impact on the game's outcome (with $0.948 \pm 0.056$ fitness with value 2 and 0 with value 5; standard deviation at 0.46) and $nCardsDraw$ the least (with only 0.006 standard deviation). We hypothesise that this is due to the *outbreak* loss condition triggering the most, meaning players benefit from being able to win quicker, but not necessarily from given more turns if the loss conditions are still very punishing.

Repeating the same experiments, with the MCTS player used for simulations instead, reveals the Dispatcher role as the weakest, averaging $0.722 \pm 0.043$ fitness as player 0 and only $0.310 \pm 0.087$ as player 1 (target win rate 100%). This particular role can be very strong due to the movement flexibility it offers, especially in combination with a Medic or Quarantine Specialist – however, the AI players used in these experiments are not skilled enough to be able to use it to its full potential and cannot make use of its advanced strategic mechanics.

More in-depth statistical analysis on parameter choices and exploration of parameters with more advanced AI players and different playstyles is noted as an area for future work.

## VII. Conclusions

This paper described the first Tabletop Games framework (TAG) competition hosted at the IEEE Conference on Games 2022, focused on the board game Pandemic. We discussed the challenges introduced in this environment relevant for game-playing Artificial Intelligence research, presented baseline AI player performance with in-depth analysis, and showed that interesting variations of the game parameters can be found by employing Quality Diversity algorithms such as MAP-Elites. We hereby conclude by inviting participation in this and future TAG competitions.

We recommend for participants, as a good place to start tackling the challenge, to reuse agents from the framework (and optimise their parameters); customise and tune heuristics; write rule-based agents; add new agents, e.g. Rolling Horizon Evolutionary Algorithms [17]; use macro-actions or goal-oriented planning; and finally to design new game configurations to augment the training set, employing the TAG analysis tools to optimise their submission.

Finally, we acknowledge the wider use of the Pandemic board game in other research or application areas, such as educational benefits for teaching cooperation, teamwork and team-building exercises [29], [30], and its potential application to handling real-world pandemics as emphasised by Larry Au [31]. With communication between players allowed, the study of language and emergent means of communications was explored by [32], the findings recommending the use of such games for learners of foreign languages. We end this paper with an optimistic statement on the value that the Pandemic board game and the proposed competition described in this paper can bring not only to the development of AI research, but to the world and society on a larger scale.

## References


[1] R. D. Gaina, M. Balla, A. Dockhorn, R. Montoliu, and D. P. Liebana, "Tag: A tabletop games framework." in *AIIDE Workshops*, 2020.

[2] M. Inman *et al.*, *Exploding Kittens*. Ad Magic, Inc., 2015.

[3] D. Billings, D. Papp, J. Schaeffer, and D. Szafron, "Poker as a testbed for ai research," in *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 1998, pp. 228–238.

[4] D. X. Vaccarino, *Dominion*. Rio Grande Games, 2008.

[5] K. Teuber, *The Settlers of Catan*. Mayfair Games, 1995.

[6] J. Fryxelius, *Terraforming Mars*. FryxGames, 2016.

[7] R. D. Gaina, J. Goodman, and D. Perez-Liebana, "Tag: Terraforming mars," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 17, no. 1, 2021, pp. 148–155.

[8] M. Leacock, *Pandemic*. Z-Man Games, Inc., 2008.

[9] C. Chabris, "The rise of cooperative games," April 2015. [Online]. Available: https://www.wsj.com/articles/the-rise-of-cooperative-games-1428676591

[10] J. Walton-Rivers, P. R. Williams, and R. Bartle, "The 2018 Hanabi Competition," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019, pp. 1–8.

[11] N. Bard *et al.*, "The Hanabi Challenge: A New Frontier for AI Research," *Artificial Intelligence*, vol. 280, p. 103216, 2020.

[12] P. Sauma Chacón and M. Eger, "Pandemic as a challenge for human-ai cooperation," in *Proceedings of the AIIDE workshop on Experimental AI in Games*, 2019.

[13] P. Sauma Chacón, "Improving cooperation between human and computer players through plan recognition in the game pandemic," Master's thesis, 2020.

[14] P. Sauma-Chacón and M. Eger, "Paindemic: A planning agent for pandemic," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, 2020, pp. 287–293.

[15] ——, "Evaluating a plan recognition agent for the game pandemic with human players," in *2021 IEEE Conference on Games (CoG)*. IEEE, 2021, pp. 1–7.

[16] K. Sfikas and A. Liapis, "Collaborative agent gameplay in the pandemic board game," in *International Conference on the Foundations of Digital Games*, 2020, pp. 1–11.

[17] R. D. Gaina, S. Devlin, S. M. Lucas, and D. Perez, "Rolling horizon evolutionary algorithms for general video game playing," *IEEE Transactions on Games*, 2021.

[18] K. Sfikas and A. Liapis, "Playing against the board: Rolling horizon evolutionary algorithms against pandemic," *IEEE Transactions on Games*, 2021.

[19] R. D. Gaina, M. Balla, A. Dockhorn, R. Montoliu, and D. Perez-Liebana, "Design and Implementation of TAG: a Tabletop Games Framework," *arXiv preprint arXiv:2009.12065*, 2020.

[20] C. B. Browne *et al.*, "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[21] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *ArXiv*, vol. abs/1504.04909, 2015.

[22] K. Kunanusont, R. D. Gaina, J. Liu, D. Perez-Liebana, and S. M. Lucas, "The n-tuple bandit evolutionary algorithm for automatic game improvement," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 2201–2208.

[23] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, "Talakat: Bullet hell generation through constrained map-elites," in *Proceedings of The Genetic and Evolutionary Computation Conference*, 2018, pp. 1047–1054.

[24] M. Balla, A. Barahona-Rıos, A. Katona, N. P. Pérez, and R. Spick, "Illuminating game space using map-elites for assisting video game design," in *11th AISB Symposium on AI & Games (AI&G)*, 2021, pp. 1–6.

[25] B. M. Viana, L. T. Pereira, and C. F. Toledo, "Illuminating the space of enemies through map-elites," *arXiv preprint arXiv:2202.09615*, 2022.

[26] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, "Empowering quality diversity in dungeon design with interactive constrained map-elites," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019, pp. 1–8.

[27] M. C. Fontaine, S. Lee, L. B. Soros, F. de Mesentier Silva, J. Togelius, and A. K. Hoover, "Mapping hearthstone deck spaces through map-elites with sliding boundaries," in *Proceedings of The Genetic and Evolutionary Computation Conference*, 2019, pp. 161–169.

[28] C. Guerrero-Romero and D. Perez-Liebana, "Map-elites to generate a team of agents that elicits diverse automated gameplay," in *2021 IEEE Conference on Games (CoG)*. IEEE, 2021, pp. 1–8.

[29] C. van Esch and T. Wiggen, "Can your students save the world? utilizing pandemic, a cooperative board game, to teach management," *Management Teaching Review*, vol. 5, no. 3, pp. 275–283, 2020.

[30] E. C. Anania, J. R. Keebler, K. M. Anglin, and J. P. Kring, "Using the cooperative board game pandemic to study teamwork," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 60, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2016, pp. 1770–1774.

[31] L. Au, "The board game pandemic: Cooperative sociotechnical imaginaries obscuring power relations," *Science as Culture*, vol. 30, no. 4, pp. 598–602, 2021.

[32] R. Masuda and J. deHaan, "Language in game rules and game play: A study of emergence in pandemic," *International Journal of English Linguistics*, vol. 5, no. 6, p. 1, 2015.