A case study in AI-assisted board game design

James Goodman
Game AI Research Group
Queen Mary University of London
james.goodman@qmul.ac.uk

Alan Wallat
holtandwallt.com
alanwallat@googlemail.com

Diego Perez-Liebana, Simon Lucas
Game AI Research Group
Queen Mary University of London
diego.perez@qmul.ac.uk, simon.lucas@qmul.ac.uk

***Abstract*—We use AI agents to play successive design iterations of an analogue board game to understand the sorts of questions a designer asks of a game, and how AI play-testing approaches can help answer these questions and reduce the need for time-consuming human play-testing. Our case study supports the view that AI play-testing can complement human testing, but can certainly not replace it. A core issue to be addressed is the extent to which the designer trusts the results of AI play-testing as sufficiently human-like. The majority of design changes are inspired from human play-testing, but AI play-testing helpfully complements these and often gave the designer the confidence to make changes faster where AI and humans 'agreed'.**

***Index Terms*—Game Design, AI Testing**

I. Introduction

Several AI techniques are used in the design of games. PCG approaches can be used to automate or assist the design of new levels or art assets, increasing the productivity of human team members and letting them focus on the higher value-added (and more interesting) activities [1]. Automatic AI testing agents can be used to find bugs in games, improving the quality of the final product and reducing the cost of repetitive human testing [2]. There are also AI approaches to balance games [3].

Much of this work is on digital video games, in which there is inherently a software artefact for the AI techniques to interface with or generate.

We present some preliminary results of using AI testing techniques in game design on an analogue tabletop board game. Specifically we use Monte Carlo Tree Search (MCTS) agents to play a game during the design process and speed up the design-test loop by reducing the reliance on human play-testing. The aim is not to auto-balance a game, but to provide actionable insights to the designer, who remains in control.

Our contribution is a case-study of using AI agents to play-test a tabletop board game during the design process. We present a pipeline of defining questions, automating metric reporting, and comparing to human play-test results with a designer-in-the-loop approach. We ascribe the design changes made to human or AI testing, to understand the relative impact of each. We also provide a typology of questions asked by game designers, and corresponding ways that AI agents can be used to answer these in a way that provides actionable insights to the designer.

A. Analogue versus digital

An analogue board game does not have any natural software representation as a video game must, and using AI techniques first requires the game to be digitally implemented, imposing an additional overhead. We use the Tabletop Games (TAG) Framework to make this process easier. TAG is a Java-based framework designed to support common patterns in modern tabletop board games, with over 20 implemented [4].

We make a distinction between *testing* and *play-testing*. The purpose of *testing* in this nomenclature is to find bugs in the implementation; this applies to video games but not to analogue games (although it does apply to a digital implementation). *Play-testing* is relevant to both and the focus is how far the game meets its design objectives.

Both analogue and digital games require alternating iterations of design and test [5]. Only when the target audience plays a game can the designer get any degree of confidence that it is hitting its design objectives. AI agents are never the target audience for a game.

However, AI agents, unlike humans, will not complain if asked to play a game ten thousand times and then another ten thousand times with one small rules tweak. This can provide detailed information on the relative values of different game elements or strategies which would be difficult to impossible to gain from the same volume of human games.

II. Previous Work

AI agents have been used to balance video games, either by recommending game parameter values that meet a pre-set balance goal [6], or letting AI agents play a game to see how quickly they achieve a goal-state. Using reinforcement learning for this was found to be too time-consuming and unstable in an FPS game, but simple A* worked well in an abstracted model of part of another game [7].

Different styles of agents have also been used to *play-test* a one-person dungeon exploration game, using evolved MCTS heuristics to emulate different anticipated human play-styles [8]. We distinguish this from using AI agents to *test* games for bugs and level glitches, such as [2].

AI play-testing has been applied to the analogue game Ticket to Ride, although not as part of the design process [9]. This was able to find a number of edge-cases not covered in the full rules, and provide relative values for different locations. This used hand-crafted heuristic agents, and the authors report that using MCTS failed to gain traction on the problem.

Beyer et al. 2016 usefully discuss the pitfalls of AI balancing of video games [3]. The agents may not be good enough for deep strategic play, or may be too good and play ‘perfectly’ in a very non-human fashion. To address these it is necessary for humans to be in the loop, either as human play-testers to complement any AI feedback or as the human designer playing games with parameters suggested by an AI optimisation approach to determine what it ‘feels’ like, and whether it is human-suitable.

Structured interview studies with board game designers have highlighted play-testing of games using AI agents as potentially beneficial by the designers [10], [11]. This work puts this proposal to the test in a real game design.

III. Game Description

Sirius Smugglers (SS) is a game for 2-5 players. Players move to one of five locations each turn. Players must first move to a new location, revealing their choices simultaneously, and then take actions at that location. The locations are:

- Ammonia moon. Players acquire Ammonia cards (values 1-3). There is a limited supply of cards each turn. If only one player is present, they gain all of them. Otherwise, they are divided equally, with extra cards going to players higher in the turn order.
- Contraband moon. As above, but Contraband cards (values 0-2) are gained.
- Smuggler moon. As above, but Smuggler cards are gained. There are 8 different types, each with a special ability. A Smuggler card ability can be used during a player’s turn by spending one Ammonia.
- Metropolis. This provides Favour cards on the same lines as above. A Favour card can be spent on a player’s turn to change the turn order, or with a Contraband card of any value to create a Cartel on the current location. A Cartel gives one bonus card of the the relevant type each turn, and only one Cartel can be present on a location.
- Sirius. This does not provide cards. A player can sell Ammonia or Contraband cards to gain points. This also increases respective progress tracks and the game ends *immediately* if one of these tracks reaches its end. These tracks each have four Medals that are gained by the player that causes the track to pass specific points. Medals increase in value, incentivising players to sell as late as possible.

Players may betray (discard) Smugglers to decrease a Corruption track, which otherwise increases through the game.This is a third way to trigger game end. The player who ends a game is not necessarily the winner, and the core game consists of tactical timing to gain the most useful cards and sell them at Sirius after the other players, to get the higher value medals, but before the game ends.

IV. Method

To test the game we need agents to play the game, a set of questions the designer wants answered, and a set of metrics to answer these questions. After each design iteration, rule changes were implemented in the digital version and 1000 games re-run to generate the automated metrics. This pipeline was streamlined with metrics added/subtracted at each iteration to the automated reporting in discussion with the designer; examples are shown in Figures 1 and 2. Human play-testing also took place between iterations, with 2-9 games generally being possible.

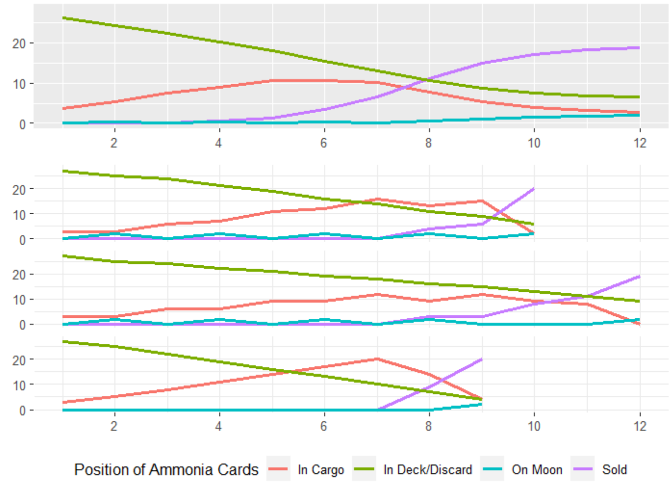

Fig. 1. Aggregate vs Sample trajectories of Ammonia. The top panel shows the average number of Ammonia cards in each possible location by game turn (averaged over 1000 games). Three sample single-game trajectories are then shown below. The aggregate plot shows clearly that Ammonia cards move from Moon to Hand to Sold over a game as expected. The individual game trajectories clarify that this is more abrupt than the smooth aggregate suggests, with Ammonia cards sold in the last couple of turns of a game.

A. AI Agents

MCTS parameters were tuned for the game at several different budgets; 8ms, 40ms, 200ms, 1s and 5s. A variety of different heuristics were used to give different styles of play; such as focusing on winning, or just final score.

B. Questions and Metrics

One challenge is generating results that are actionable by the designer. This involves an iterative dialogue starting with an initial set of exploratory statistics and graphs. With each design iteration, the set of reported data was adjusted. The starting set of metrics was:

- Game end conditions. How long does a game last, and how often are the different end-game conditions triggered. What are the final scores and the margin of victory?
- Benefit of player position. Is there any tendency for a player in first (or last) position to win?
- Tracking of selected game statistics over time; the aggregate average over 1000 games, plus sample trajectories of individual games. Both are important, as the average can hide important details as in Figure 1.
- Do agents with higher computational budgets defeat those with lower, and do they play differently?

Data were gathered and reported for all player counts to determine if these changed qualitatively between 2 and 5 players. This was useful for the designer as play-testing at higher player counts was more difficult to arrange. The overall pipeline was automated so that after each rule change a new set of AI play-tests was run with pre-specified agents and the current metrics generated in a user-friendly format (using R Markdown). The types of questions that arose as the process continued are reviewed in Section V-B.

V. Results

A. Agent verisimilitude

An important initial set of results was that agents played sufficiently like humans for the designer to broadly regard the data as useful. High-level agent behaviour was similar to that of human play-testers, with the behaviour of low-budget agents corresponded broadly with 'beginners', selling cards early, while the higher-budget agents had a more 'advanced' play style that focused on timing sales to be immediately before the end of the game to maximise medal points. Agents with a higher budget consistently won games when playing the lower budget ones.

B. Design goals and questions

Different types of question from the designer require different experimental techniques. This preliminary work provides some initial results on just the first two of these.

- Observational. Questions that can be answered by letting agents play the game. For example, how long an average game takes, whether initial player position makes a significant difference, is there a correlation between visiting the Metropolis and winning?
- Emulation. The AI play-style is reduced to simple heuristics implementable by a human player. This is then used in play-tests with a human emulating the AI policy and helps in understanding what it feels like to play in this way, and gain design intuition about why it works (or doesn't). This can help Where the designer is looking at tables of data and, "trying to understand how these feed into my design process".
- Interventions (hard). Questions on the lines of, 'Can you win without visiting the Metropolis?' This requires a controlled experimental set up, with agents constrained from not visiting the Metropolis. This was a natural follow-on question from the game designer, but not one answerable from a pure observational study. We propose to use the Restricted Play Framework with MCTS to be able to answer these questions in future work [12], [13].
- Interventions (soft). This is to address questions around effectiveness of different strategies, for example if a player focuses trading ammonia rather than contraband. The approaches proposed are progressive bias or widening in the selection phase [14], [15], and rollout policies that preferentially select a particular action type without imposing a hard constraint. This allows agents to ignore the heuristic in situations where it is clearly sub-optimal.
- Game Tuning. This was *a priori* expected to be a benefit of using a software framework like TAG, in which elements of the game can be parameterised and then auto-tuned to achieve a specified outcome. However, this was not found to be a helpful tool at this stage, as the designers were disinclined to have agency taken away from them. This is consistent with the results in [10].

Type	Human	AI	Joint	Designer	Total
Balance	1	1	2	1	5
Cosmetic	3				3
Simplification	8		1	3	12
Complexification	7	1	2	2	12
Total	19	2	5	6	32

TABLE I
Rule changes during three design iterations. 'Human' and 'AI' count changes inspired by the respective play-testing. 'Joint' stem equally from both. Cosmetic changes affect the display of the game to the players; simplification changes remove rules; complexification changes add rules; balance changes tweak rule parameters.

C. Impact on Design

The rule changes made over 3 iterations of the game were tracked and attributed to the results of human play-testing, AI play-testing or designer fiat outside of any direct link to play-tests. These results are summarised in Table I. The majority of changes are made due to the results of human play-testing. The impact of changes from AI play-testing is smaller, with more changes jointly supported by the two play-testing methods. The set of rule changes in Table I shows qualitatively the changes from human play-testing were to do with changes that simplified the game; by removing rules that confused players, amending the visual iconography of the game components, and standardising conventions across different aspects of the game. Example changes driven by human play-testing are the amendment of the corruption track to increase from zero to its max value in line with the other two game tracks, instead of starting at maximum value and decreasing to zero; or amendment of turn order to reduce interleaving of player turns across different locations, which was found to be confusing. AI play-test agents are blind to these sorts of issues.

The changes inspired by AI play-testing, jointly with the results of human play-testing were around the prevalence of Cartels. In the first iteration of the game implemented Cartels could be created on any location simply by playing a Favour card. This led to a dominant AI opening move of visiting the Metropolis to get a Favour card, and then competing for early-game Cartels. To address this a cost of a Contraband card was introduced to place a Cartel (complexification), and this was only permitted on the current location (balance). The effect of this on the game is illustrated in Figure 2. With the changes Cartels were still used, but moving the Metropolis as one's first move was no longer a dominant strategy.

AI play-tests also revealed unexpected edge-cases not found in human play-tests, as in [9]. For example an AI agent with an objective only of winning could prevent a game from ever ending by collecting a certain set of cards and refusing to sell them (which would cause it to lose the game).

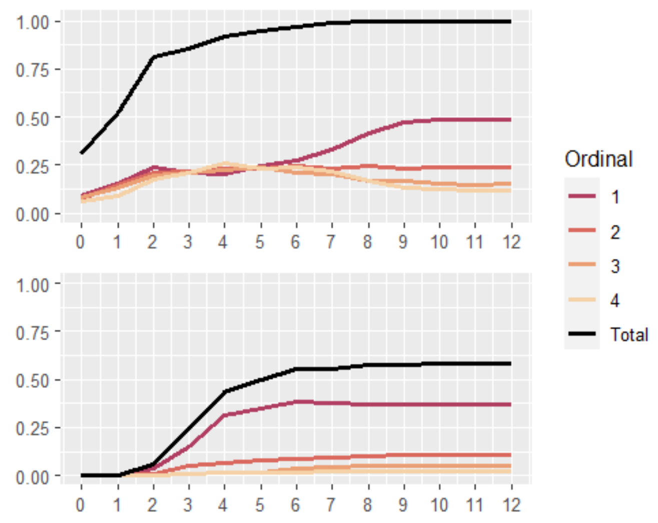

Fig. 2. Cartel prevalence on the Ammonia moon before and after rule changes. The top panel required only a Favour card to place a Cartel, and led to a dominant move to the Metropolis in the first turn. The bottom panel shows behaviour after increasing the cost of placing a Cartel. Ammonia Cartels now only appear from the 3rd turn (Turn 2), and are only used in about half of games.

These AI agents *cannot* tell us whether the rules are unclear, if the theme comes across satisfactorily in the mechanics, if the graphics iconography is confusing, or what it feels like to play the game. It is not straightforward to address these questions with AI agents, but future work that starts to do so even partially will make AI-testing increasingly useful to designers. Possible approaches are analysis of rules with large language models, and GAN-analysis of graphics.

D. Impact on Process

Games change a lot during design. In a video-game a decision to re-write a core part of game play discards an expensive set of software artefacts. In a tabletop game the artefacts are cheaper, and the designer is readier to go back to the drawing board given the lower cost. This needs to be taken into account, and it may be worthwhile waiting until a later stage of the design process before committing the game to code, although TAG helps reduce this load.

It is essential to physically play-test the game with the designer. This catches issues where the designer has mentally changed a rule, but the documentation may not have quite caught up. It also catches issues where the rules document was misinterpreted. This is no different to any software project and the designer as product-owner needs to be able to detect errors of implementation.

VI. Conclusions

We have presented some early results of a case-study of using AI agents to assist in the play-test of an analogue game in the design phase. To be useful the designer needs to trust that the agents can actually play the game, for example if the results of the AI experiments qualitatively agree with the results of human play-tests. To quote the designer, “I am glad this is the case. This corroborates player-test findings.”

AI game testing cannot remove the need for human play-testing. It can help find design issues that do not occur frequently, complement play-testing with additional data, and speed up the game development process. It allows human play-testers to focus on the questions only they can currently answer.

Acknowledgment

This work was funded by the EPSRC CDT in Intelligent Games and Game Intelligence (IGGI) EP/S022325/1.

References

[1] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation: A taxonomy and survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 172–186, 2011, publisher: IEEE.

[2] A. Sestini, L. Gisslén, J. Bergdahl, K. Tollmar, and A. D. Bagdanov, “CCPT: Automatic Gameplay Testing and Validation with Curiosity-Conditioned Proximal Trajectories,” *arXiv preprint arXiv:2202.10057*, 2022.

[3] M. Beyer, A. Agureikin, A. Anokhin, C. Laenger, F. Nolte, J. Winterberg, M. Renka, M. Rieger, N. Pflanzl, and M. Preuss, “An integrated process for game balancing,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 2016, pp. 1–8.

[4] 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.

[5] D. Davidson and G. Costikyan, *Tabletop: Analog Game Design*. Lulu. com, 2011.

[6] D. Hernandez, C. T. T. Gbadamosi, J. Goodman, and J. A. Walker, “Metagame Autobalancing for Competitive Multiplayer Games,” in *2020 IEEE Conference on Games (CoG)*. IEEE, 2020, pp. 275–282.

[7] Y. Zhao, I. Borovikov, F. de Mesentier Silva, A. Beirami, J. Rupert, C. Somers, J. Harder, J. Kolen, J. Pinto, and R. Pourabolghasem, “Winning is not everything: Enhancing game development with intelligent agents,” *IEEE Transactions on Games*, vol. 12, no. 2, pp. 199–212, 2020, publisher: IEEE.

[8] C. Holmgård, M. C. Green, A. Liapis, and J. Togelius, “Automated Playtesting with Procedural Personas through MCTS with Evolved Heuristics,” *arXiv:1802.06881 [cs]*, Feb. 2018, arXiv: 1802.06881. [Online]. Available: http://arxiv.org/abs/1802.06881

[9] F. de Mesentier Silva, S. Lee, J. Togelius, and A. Nealen, “AI-based playtesting of contemporary board games,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*. ACM, 2017, p. 13.

[10] V. Krainikova, “Prototyping of a Client for Board Games Automated Testing and Analysis,” in *Digital Transformation and Global Society: 6th International Conference, DTGS 2021, St. Petersburg, Russia, June 23–25, 2021, Revised Selected Papers*. Springer, 2022, pp. 347–361.

[11] L. Kougioumtzian, C. Lougiakis, and A. Katifori, ““Show your cards!”: What do creators need for the game design process?” in *Proceedings of the 18th International Conference on the Foundations of Digital Games*, 2023, pp. 1–6.

[12] A. Jaffe, A. Miller, E. Andersen, Y.-E. Liu, A. Karlin, and Z. Popovic, “Evaluating competitive game balance with restricted play,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 8, 2012, pp. 26–31, issue: 1.

[13] O. Keehl and A. M. Smith, “Monster Carlo: An MCTS-based Framework for Machine Playtesting Unity Games,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. Maastricht, Netherlands: IEEE, 2018, p. 8.

[14] G. M. J. Chaslot, M. H. Winands, H. J. V. D. HERIK, J. W. Uiterwijk, and B. Bouzy, “Progressive strategies for Monte-Carlo tree search,” *New Mathematics and Natural Computation*, vol. 4, no. 03, pp. 343–357, 2008.

[15] A. Couëtoux, J.-B. Hoock, N. Sokolovska, O. Teytaud, and N. Bonnard, “Continuous upper confidence trees,” in *International Conference on Learning and Intelligent Optimization*. Springer, 2011, pp. 433–445.