Game Element Design with Evolutionary Computation

Luke Aschenberg
lmaschenberg@knox.edu
Computer Science Department
Knox College, IL, USA

April 2022

1 Introduction

Designing a game is an extremely time-consuming process, and very, very few designers can support themselves on their designs alone, as a typical published game will bring the designer only $18,000 in royalties;[7] the average cost of living in America is $60132.[26] Automating this process can save immense amounts of time, and has the potential for immense innovation. Many people have attempted to automate game design to some extent or another, and I have compiled the following as a summary of their work.

As this problem has such a wide and unknown search space, Evolutionary Computationis a well-suited approach. Some papers directly tackle the problem, while others tackle adjacent problems like General Game Playing. I will be focusing on papers that use evolutionary computation, but this is not the only solution. Machine learning or other metaheuristics would also be well suited for the problem.

This summary has five major sections. The first four are generalized aspects of the process, and the fifth is an overview of the frontiers I have identified for each of those aspects. I will first examine the limitations each paper places on the domain to make the problem manageable. Then, I will describe the various encodings used, followed by the types of playtesting artificial intelligences and the fitness functions evaluated through their playtesting. The existing literature contains a wide variety of encodings and scope of search domain. This breadth means there is no significant synthesis that I can construct from the genetic operators each paper used.

Each of the four aspect sections will begin with an outline of Cameron Browne's work. Of each of the 13 most relevant projects(those included in the summary table 1), 61% cite him. His work forms the groundwork of many of the other papers, which makes him an excellent center-point to compare the other works to. After describing his project, I will describe how other authors have addressed the limitations of Browne's work and expanded the scholarly exploration of the total search space.

Additionally, I use a table at the bottom of the paper to summarize the papers I address for easy reference.1

2 Search Domains

Before identifying the domains that have been explored in the search space of designing games, it is important to identify the global domain of what games can be produced in general. Browne used the following definition of what a game can be:

> "A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome"[5][23]

This includes cooperative games, combat systems in TTRPGs, card games, arcade games, word games, and sports among others. Using this definition, there is a wide variety of games that have yet to be explored. Really, the global domain's frontiers depend on your definition of a game. Bernard Suits uses a wider definition that forgoes rules and quantifiable outcomes:

> "Playing a game is the voluntary attempt to overcome unnecessary obstacles"[22]

This definition includes TTRPGs even beyond their combat, tag, truth-or-dare, and competitions for the best hand-stand. Even this definition has been widely critiqued for Suits' claim that it represents all games possible [17]. For this paper, I will be assuming the definition Browne used is sufficiently broad, as rules and quantifiable outcomes make computational design significantly more within reach. Additionally, the scope of my research was limited to games that are simple enough to be represented with physical components, so I will not address frontiers in video games.

2.1 New Combinatorial Games

Browne focused on combinatorial games[5], which are antagonistic (non-cooperative), 2 player, perfect information games. Perfect information games are defined by their lack of hidden information and stochasticity, and their turn-based, finite-length nature. They are called perfect information because both players can see everything on the board, which makes these games intensely skill-based. Popular examples of games in this genre are Chess and Go. Browne published 2 of the 19 games his algorithm produced under the names Yavalath and Pentalath.[1] He is the first, and even after a decade, still the only author of a program that has wholly created publishable (and published) games.

His search space is further limited by his encoding, as not every conceivable combinatorial game can be produced. Browne's search space is any combinatorial game that can be constructed using the 200 ludemes he built into his encoding with a limited board size. Ludemes are rules that appear in one form or another in various combinatorial games, such as "in-a-row," which appears as a win condition in tic-tac-toe and connect 4, among others. [5] In other research, he found that many existing combinatorial games are simply recombinations of existing ludemes[11][19], and his representation is complex enough to encompass games as complex as chess. This means that restricting the search space to the specific ludemes he has formulated is a more minor limitation. The limited board size comes from an attempt to streamline the evolutionary process. Games were limited to a maximum of 5-length sides on a hexagonal board, 7-length for a trapezium, 8 for a square, 11 for a rhombus, and 11 for a triangle.[5]

2.2 New Card Games

In the literature reviewed, only one other than Browne aimed to fully create new rule sets. Jose Font et al. aimed to develop novel card games (games using one or multiple standard french decks of cards and sometimes tokens)[9]. Similar to combinatorial games, this is a decently limited search space, especially as they limited the search space within card games to only those using a standard deck of cards (excluding games like Pinochle and Schafkopf). This is a distinct search space from combinatorial games, however, as card games are built around the concept of chance. Multiple games may be played with the same strategy and have different outcomes, which makes gauging fitness more difficult in this domain.

However, within the search space of card games, their encoding limits them far more than Browne's encoding limited him. While they claim to be able to replicate Uno, their example encoding of Uno does not have any functionality of the special cards[8], and their description of their encoding does not suggest that it has the functionality to implement these cards. Card games that allocate specific functionality to particular cards other than score are not represented.

By limiting the search space so extremely, they have developed an algorithm that has developed dozens of games.[9] However, "these games are not entertaining for a human to play, and they do not include any particularly interesting novel mechanics"[9]. They have identified some changes that can be made to their fitness function to better select for more strategy-centric games, which suggests they believe that even in this limited search space, there is still plenty of room to explore.

2.3 Improving Existing Complex Board Games

The majority of the reviewed literature limited their search space to an extreme. Most papers chose a single board game, and attempted to identify imbalances[15], optimize existing components, or create variations of their chosen game[25][6][16][4][18]. In many of these cases, the chosen board game is even simplified farther[4][15], similarly to how Uno was simplified. Leonardo Bombardelli simplified the game RISK to remove the hidden-information aspects, and modifying the stochastic elements to compensate for this removal. Mahlmann et al. simplified the game Dominion by only looking at the 2 player variation of the game[15], and Bombardelli simplified RISK similarly[4]. Other papers are unclear about the number of players simulated, and may also be only testing 2 player variations of the game. Fernando Silva et al.

assumed that the original Ticket to Ride game was optimally balanced, and used statistics from that game as the basis of their algorithm. This is a simplifying assumption, as Mahlmann et al. was looking for imbalances in Dominion.

These assumptions are all necessary, however, as each of these three games have much more complex components than combinatorial or card games, and have relatively high branching factors even before introducing the stochastic/hidden information components of play. The choice to optimize a single game allows them to evaluate and/or generate variations of the components for these games, as evolving complete games with this level of complexity appears to be currently out of reach. Christian Sadi estimates the branching factor of a simplified variation of RISK is $10!(13^{10})$ [20], and Silva et al. estimate the branching factor of Ticket to Ride between 60 and 100[16]. While Ticket to Ride has a smaller branching factor than Go (250), Chess only has a branching factor of 35[20]. Non-combinitorial games with stochasticity tend to have high branching factors like these that make evaluating strategy a hard problem[20], and avoiding re-training General Game Players (GGPs) to fit each new game by limiting the search space to a single game is invaluable.

3 Combining Existing Board Games

As a midpoint between creating fully new rule sets and optimizing existing rules, Halim et al. chose to combine Chess and Checkers. As they both contain rules for capturing, moving existing pieces, and the same 8x8 checkerboard square tiled board, this gains some of the advantages of the limited search space that single-game papers enjoy[12]. Hom et al. chose a similar strategy with a broader search space, using Reversi, Checkers, and Tic-Tac-Toe, along with 8 rules they developed by hand. This is still smaller than card games or combinatorial games, but is a substantial space to search. Hom et al. calculated 5616 combinations of rules could be formed.[13] Browne's work is founded on the idea of making new games through combining others, which Halim's et al. and Hom's et al. projects can accomplish in a more narrow manner.

4 Encodings

Machado et al. claim that when evolving art, we have been more successful evolving music than visual art.[14]

> "The fact is that music theory is more developed and quantitative than theory in visual arts. In music we can "write down" a piece, in visual arts we have neither the alphabet nor the grammar to do so."

One of the first steps, then, in learning to evolve board games, is to develop a notation system, as there is not one standard notation created. Some of the following papers have used the same Game Description Language(GDL)[1], but most of them have created their own notation system to evolve games with.

4.1 Ludi–A Combinatorial GDL

Browne's work centers around a system called Ludi. Ludi includes many components, but the encoding of the GA is Ludi's GDL, and Ludi's rules parser. GDLs are text-based representations of board games that can be used as the encoding as Ludi does. The rules parser can understand the syntax of the GDL and translate it into a playable version with a Graphic User Interface(GUI). As described above, this GDL is formed of 200 ludemes (rule-concepts that exist across multiple combinatorial games). These high-level

[1]GDL is used to refer generally to any Game Description Langugage in this paper, not to Stanford's specific GDL[2][19]

concepts make Ludi's GDL relatively easy to read as a human, especially in comparison to lower-level GDLs like the one Stanford used for some of their GGP projects.[2]

Browne's GDL did have potential to create wholly invalid games, which the rules parser would be unable to generate, and eliminating these was the first stage new chromosomes went through. Browne also described how intron genes may have caused a degree of slowdown[5], but how they were necessary for the creation of the most complex and innovative mechanic Ludi produced[5]. Browne's first game to be published, Yavalath, has the end conditions (win(in-a-row 4))(lose(and(in-a-row 3)(not(in-a-row 4))))[5]. Before being fully formed, some of these conditions may have been meaningless or actively detrimental. The slowdown caused by keeping introns was ultimately worth it for Browne, but other papers decided that this potential is not worth it.

4.2 Non-Combinatorial GDLs

Font et al. developed a separate GDL for their card game evolution that uses a context-free grammar system. Their GDL is higher level than Browne's, building in a lot of structure to the encoding. This structure, and the grammar-basis of the system makes it impossible to generate unplayable games[8], although it can still create games that are non-finite, overly complex, or drawish.[9]

These introns Browne grappled with are likely not possible in grammar-guided genetic programming, which has the other side of the double-edged sword Browne encountered, where subtler rules combinations like Yavalath's may be difficult to evolve as they require multiple mutations to occur simultaneously without introns. The biggest drawbacks of Font's et al. card game GDL are in how they limit the search space. The potential drawbacks of grammar-guided genetics may be situational, and the advantages seem like they may be worth it.

GDLs have been developed for GGPs that include stochastic, hidden information games.[24][10] Michael Thielsher made one based on adding 2 keywords to an existing GDL originally designed for combinatorial games. A "sees" keyword allows some information to be hidden by prescribing what is shown, and a "random" keyword allows a simulation of most random elements in a game. The "sees" keyword adds a line of instruction to every rule that other players can see. The original GDL was very simple, with only 8 keywords used to define others specific to the game.[24] This ability to define other keywords is intriguing, as it easily allows for abstractions and full exploration of the domain the language is designed for. Additionally, this GDL is broad enough to encompass both combinatorial and card games, along with others.

However, This reduced keyword set makes for a lower-level GDL that uses many lines of language to describe a single mechanic. This lower-level language may not be as well-suited for evolution, as it takes a substantial amount of time to evaluate a single game, and mutations in this language would have a much smaller effect on the overall game.

In their card game GDL, Font et al. established zones of play that are always shown and zones that are always hidden.[8] Because Font et al. designed their language from the get-go with hidden information in mind, and with the idea of evolutionary generation in mind instead of GGP, their language is able to address a distinct domain from Browne more effectively than if they had used a modified version of Ludi's GDL,[19] or another combinatorial GDL.[24]

There are other GDLs that have been presented, including one made by Gaina et al. designed for GGPs to play "modern" board games, including Pandemic, Exploding Kittens, and Colt Express[10]. Gaina's et al. framework is extremely low level, requiring many files to represent a single game. Their framework allows them to calculate many attributes about the game's complexity and run the game using a variety of GGPs, but this extremely low level encoding makes using this framework infeasible for generation. Although Thielscher is much closer than Gaina et al. to having a usable alternative encoding, his is not usable wihtout adaptation. I have not sought out additional GDLs designed without evolution (or generation in general) in mind.

4.3 Hard-Coded Encodings

Evolving variations of a single game has the advantage of avoiding using a GDL and a rules parser[25][6][16][4][18][12 By directly representing the simulation of their chosen game, it is possible to further streamline the process of searching the already limited domain of a single game. As each hard encoding is unique to the specific games implemented, it is difficult to synthesize any meaningful comparison between them besides the efficiency of this general approach.

Mahlmann's et al. encoding falls in a grey area between a generalized encoding and a hard encoding. Mahlmann et al. also proposed a general representation of games. In their optimization of Dominion, they represented each set of decks with a 10 integer vector, with each integer representing one of the decks to be selected.[15] They believe that this could represent any number of games by assigning each rule in the game an integer, and using chromosomes as selections of rules that are in play at a time.[15] This encoding is specifically designed to optimize an existing ruleset, and is the highest level representation out of the four addressed. Sets of integers are very simple representations that are well suited to GAs. However, while this representation can work with any number of games, it does not work well for the generation of new games, or even game elements. This could be used in combination with a lower-level GDL like Thielscher's modified combinatorial GDL in an algorithm used to evolve novel rules, but does not work in isolation.

5 Fitness Functions

As with all artworks,[17] finding a fitness function that does not involve humans directly in the evaluation of a candidate is important. This is difficult, as the criteria for what makes a game good are mainly subjective.[25] Even something like the difficulty of a single-player puzzle can be difficult to approximate with a fitness function, as what is hard for humans and what is hard for computers can be substantially different.[3][13]

5.1 Generalizable Criteria Experiment

Browne used the genesis of board games as part of a larger experiment to demonstrate that there were quantitative criteria that could be used to determine the fitness of board games. 79 games were tested by humans, and 57 criteria were tested by a GGP on the same games. When the best 16 of the criteia were tested alone, there was a strong correlation between the GGP's rankings and the humans'.[5] These 16 were used as the fitness function for the genesis of the new board games, and when humans tested the generated board games, the fitness function still held a strong correlation. The best 6 of these criteria were uncertainty, lead-change, permanence, killer moves, completion, and duration.[5] Completion and duration are measures of the baseline playability of the game. Uncertainty and lead-change both measure the ability for players who are behind to return. Permanence measures the non-triviality of decisions made–can a skillful move be completely undone by your opponent on their next turn? Killer moves encourages the game to end quickly once a winner is in a definitive position.

5.2 Implementations of the Generalized Criteria

Several of the papers used variations of Browne's criteria as their fitness function. With 57 criteria measured, it is not a surprise that Browne established a solid groundwork in this area to take from.

Bombardelli and Mahlmann et al. used lead-change[4] and Mahlmann also used 'decisiveness,'[15] which is described in a similar manner to the killer moves criteria Browne measured. Bombardelli used Drama as their second criteria[4], which was one of the top 16 but not one of the top 6[5]. Font et al. used completion as a prerequisite for a game being fully evaluated, and duration as a part of their fitness function.[9]

Applying these criteria to non-combinatorial games can be difficult, however, as it is much more difficult to gauge the strategic value of a position in a previously unseen game with such complexity. Although Mahlmann et al. mainly used similar fitness functions to Browne, identifying who was in the lead for the purpose of tracking lead-change and decisiveness required many games to be played before fitness evaluation could begin.[15]

5.3 Simulationless Criteria

Simulating the playtime of a game is extremely taxing, and several papers have found methods to develop variants of existing games using fitness functions not dependent on a simulation. Tijben developed an algorithm that used entirely different criteria, and a different evaluation system as well. Tijben did not simulate games to playtest the dungeons, but instead just measured the complexity, theme, size, and clutter to develop a combined fitness score.[25] These criteria are very tailored to the specific project of Gloomhaven dungeons, and are not explicitly linked to gameplay or any human evaluation, but did allow for quicker generation of solutions. Humans did evaluate the dungeons afterwards and found positive results, although Tijben discusses several possible ways this could have been biased. [25]

Andrew Cardona et al. also used a simulationless fitness test for creating decks for the game Top Trumps. The fitness function is the sum of the percentage of other cards each card can beat under ideal circumstances. The balance of the game was validated later by human playtesters, who largely enjoyed the game and were more concerned with thematic issues than mechanical balance.[6]

Oranchak, while using a simulation to determine some of the fitness function, also uses several criteria that do not need to be simulated. To develop visually pleasing Shinro puzzle boards, constraints for patterns and symmetry were used.[18] The difficulty of the puzzle was also calculated, but these aesthetic criteria may have made human-enjoyable games have a higher fitness with a lower computational cost.

5.4 Other Simulation-Based Criteria

Of the simulation based approaches, however, Silva et al. also found additional criteria to measure a game by. They tracked the branching factor each turn of the simulated game to find the 'game arc,' which they attempted to fit to a curve as their primary fitness function.[16] This appears to be much more portable than Tijben's criteria, and served to effectively measure a game by itself whereas each of the other papers used several criteria in combination with each other. Additionally, this criteria is relatively easy to measure in comparison to lead-change, and avoids the difficulties Mahlmann et al. encountered. However, Silva's et al. criteria were reinforced by correlation with human evaluation, which is an area Silva et al. identified future research would be needed in.[16]

Halim et al. also identified additional criteria for their combined chess/checkers variants. Halim et al. measured the dynamism of each piece, punishing variations that had pieces that were mostly neglected by intelligent players.[12] Additionally, they measured whether each region of the game board saw equal use. These two criteria together encourage more efficient games that use all pieces in all parts of the board, and take full advantage of the resources given to the players by the rules. If pieces or spaces are not used, material is wasted.[12] Additionally, ensuring that all options are used equally is a good way to ensure that there is not a dominant strategy.[21]

Halim et al. and most of the other papers that used a simulation based evaluation used multiple types of players to determine skill level/strategic potential of the games. Halim et al. called this a fitness criteria called "Intelligence," while Font et al. used this as a basic evaluation rather than as part of the fitness function.[9][4][15][12] This is particularly valuable when paired with the Dynamism criteria, as encouraging all options to be equally good has the potential problem of making choices not impactful, which should be avoided even as much as a dominant strategy.[21]

6 Fitness Evaluators

6.1 Min-Max Search with Parent-Based Advisors

Ludi's GGP uses a Min-Max search algorithm with alpha-beta pruning, identifying the value of each situation by using a weighted sum of 20 different strategic advisor functions. Advisors may push for playing near the center, near the edges, or for a greater set of possible moves, among others. The weights for each advisor are set initially by averaging the weights used for the two parent games selected during crossover, then refined over the course of play.[5] This allows Browne to have decently skilled players testing the game with minimal training at the beginning of each round.

6.2 Multiple Simple Players

As with the encodings, many game-specific papers used policy-guided players[15][16][4][18] instead of a search algorithm. Mahlmann et al. used 2 simpler policy-guided players, and a neural network policy trained by Fynbo and Nelleman, all of which were better in some circumstances and worse in others.[15] Font et al. used simple random players for their card games for the most part, and a basic Monte Carlo Tree Search algorithm to determine strategic fitness.[9] Silva et al. used 4 simple hand-crafted policy-guided players,[16] and Bombardelli used 1. [4]

As with hard-encoding the specific games, using basic players allows these projects to run each iteration of the game faster, to the point where Bombardelli can evaluate a generation in 30 seconds[4], whereas Browne's algorithm can take several hours to find the fitness of a chromosome.[5] Beyond limiting the search space, these changes to the encoding and fitness evaluation have allowed for marked speedups.

7 Frontiers

This is a long paper, so I will encourage you to see the table listed in the final section of this paper as a quick summary reference as you consider the frontiers I describe below.1

7.1 Search Domains

Most literature focuses on optimizing or creating variants of specific board games, which cover very little of the domain. Browne has searched small-board combinatorial games, and Font et. al has searched a limited definition of card games.

The strengths of these approaches could be combined in future research. The variants that were focusing on specific games were extremely focused on exploitation of a limited search space, and with a very small degree of exploration added to the beginning, it is possible that an algorithm that creates wholly new games in a more efficient way could be formulated. Browne lamented that some of the games produced were a single mutation away from being excellent games, and thought evolutionary computation may not be the proper search algorithm.[5] Using a more focused search domain may allow evolutionary computation to remain an effective tool. The vast majority of the search space that has not been explored rests in stochastic, hidden information board games that have a similar complexity to the games that have been addressed in isolation. Using a combined approach to explore this search space could be very helpful.

As far as sub-domains that have not been touched at all, dexterity games like ERS, or simultaneous play games like Diplomacy have not been touched at all by any papers I have seen. Dexterity games may be able to use similar ideas to video games while utilizing the strength of relative simplicity that board games offer. Simultaneous play games like Diplomacy may require a newly composed GDL and rules-parser, but should not require any differences in the GGP or fitness function than other than the ones in other hidden-information games.

7.2 Encodings

Thielscher's modified combinatorial GDL was handy for expressing a very wide domain of games, but was a relatively low-level encoding. It may be possible to adapt this GDL for evolution by interpreting the game at a high-level in the rules-parser. For example, the rules-parser could extract a defined keyword and all lines of the language that describe it, and mutation and crossover could treat it as a single unit. However, this would add an $O(n^2)$ algorithm to the evaluation of every individual in an already slow algorithm.

7.3 Fitness Functions

Silva et al. identified a simple, general, and effective fitness function through game arcs that is not typically used. However, their understanding of what makes one arc better than another was limited, so they settled for what was typical to most games, and specifically to some editions of Ticket to Ride.[16] Gaining a better understanding of game arcs would allow this fitness function to be used in combination with other identified criteria for a stronger identification of fitness, which may lead to more efficient exploitation of search space.

7.4 Fitness Evaluators

Font et al. noted that much of the strategy in their card games was shallow, and suggested using three tiers of players to identify the strategic potential of a game instead of two. Using random agents initially to weed out poor candidates may make it more computationally efficient to use a more in-depth GGP like Browne's. However, if Ludi's GGP in particular is used, games that do not graduate from the random agents quickly should be discarded, as Ludi depends on the strategy of the parents of the game to determine a policy for the candidate chromosome.

8 Reference Summary Table

Author	*Domain*	*Encoding*	*Fitness Criteria*	*GGP*
Browne	New Combinatorial Games	Ludeme GDL and Rules Parser	Best of 57 determined by experiment	Min-Max Search with Inheritable Evaluation Policies
Font et al.	New Card Games	Stages and Areas GDL and Rules Parser	Completion, Complexity, Drawishness, Wingap	Simple Random, and Unmodified MCTS
Halim et al.	Combinations of Chess and Checkers	Integer Vector and Hard Coded Game	Intelligence, Space/-Piece Dynamism, and Duration	Simple Random, and Min-Max Search
Hom et al.	Combinations of Checkers, Reversi, and Tic-Tac-Toe	Zillions of Rules	Drawishness and First-Turn Advantage	Zillions of Rules
Silva et al.	Boards and Decks of Ticket to Ride	Graph, Boolean Vector, and Hard Coded Game	Branch Factor Curve	4 Handcrafted Policies
Bombardelli	RISK Maps	Graph and Hard Coded Game	First Turn Advantage, Drama, Lead Change, Duration	1 Handcrafted Policy
Cardona et al.	New Decks of Top Trumps	2D Array and Printed Game	Power of Card Ideally	N/A
Tijben et al.	New Gloomhaven Dungeons	Class and Printed Game	Complexity, Theme, Size, and Clutter	N/A
Mahlmann et al.	Balancing Dominion	Integer Vector and Hard Coded Game	Lead Change and Decisiveness	2 Handcrafted Policies and 1 Neural Network evolved with NEAT
Oranchak	Shino Puzzle Variations	2D Vector	Symmetry, Pattern Constraints, Moves to Solve	Handcrafted Policy
Thielscher	Combinatorial with Hidden Information and Stochasticity	Expanded Combinatorial GDL	N/A	N/A
Piette et al.	Combinatorial, Hidden Information, Stacking, and Boardless	Ludeme GDL and Rules Parser	N/A	N/A
Gaina et al.	Modern Board Games	Multiple Files per Game	N/A	N/A

Table 1: Summary

References

[1] URL: https://boardgamegeek.com/boardgame/33767/yavalath.

[2] URL: http://logic.stanford.edu/ggp/notes/gdl.html.

[3] Daniel Ashlock. "Automatic generation of game elements via evolution". In: *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. IEEE. 2010, pp. 289–296.

[4] Leonardo Boaventura Bombardelli. "Generating variations of the board game risk through evolutionary game design". In: (2022).

[5] C. Browne. *Evolutionary Game Design*. SpringerBriefs in Computer Science. Springer London, 2011. ISBN: 9781447121794. URL: https://books.google.com.br/books?id=tn2fE9USzZkC.

[6] Andrew Borg Cardona et al. "Open trumps, a data game". In: *Foundations of Digital Games, Ft. Lauderdale FL USA, aboard: Royal Caribbean's Liberty of the Seas (2014)*. Society for the Advancement of the Science of Digital Games. 2014.

[7] Emily. *How much board game designers make and where they work*. May 2022. URL: https://mykindofmeeple.com/how-much-do-board-game-designers-earn/.

[8] Jose M Font et al. "A card game description language". In: *Applications of Evolutionary Computation: 16th European Conference, EvoApplications 2013, Vienna, Austria, April 3-5, 2013. Proceedings 16*. Springer. 2013, pp. 254–263.

[9] José Marıa Font Fernández et al. "Towards the automatic generation of card games through grammar-guided genetic programming". In: (2013).

[10] Raluca D Gaina et al. "Design and implementation of tag: A tabletop games framework". In: *arXiv preprint arXiv:2009.12065* (2020).

[11] Connection Games. "Variations on a Theme". In: *AK Peters* (2005).

[12] Zahid Halim, Abdul Rauf Baig, and Kashif Zafar. "Evolutionary search in the space of rules for creation of new two-player board games". In: *International Journal on Artificial Intelligence Tools* 23.02 (2014), p. 1350028.

[13] Vincent Hom and Joe Marks. "Automatic design of balanced board games". In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Vol. 3. 1. 2007, pp. 25–30.

[14] Penousal Machado and Amılcar Cardoso. "Computing aesthetics". In: *Advances in Artificial Intelligence: 14th Brazilian Symposium on Artificial Intelligence, SBIA'98, Porto Alegre, Brazil, November 4-6, 1998. Proceedings 14*. Springer. 1998, pp. 219–228.

[15] Tobias Mahlmann, Julian Togelius, and Georgios N. Yannakakis. "Evolving card sets towards balancing dominion". In: *2012 IEEE Congress on Evolutionary Computation*. 2012, pp. 1–8. DOI: 10.1109/CEC.2012.6256441.

[16] Fernando de Mesentier Silva et al. "Evolving maps and decks for ticket to ride". In: *Proceedings of the 13th International Conference on the Foundations of Digital Games*. 2018, pp. 1–7.

[17] C Thi Nguyen. *Games: Agency as art*. Oxford University Press, USA, 2020.

[18] David Oranchak. "Evolutionary algorithm for generation of entertaining shinro logic puzzles". In: *Applications of Evolutionary Computation: EvoApplicatons 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part I*. Springer. 2010, pp. 181–190.

[19] Eric Piette et al. "General board game concepts". In: *2021 IEEE Conference on Games (CoG)*. IEEE. 2021, pp. 01–08.

[20] Christian Sadi. *Building an artificial intelligence for risk: Part 1 - minimax and the branching factor*. Dec. 2017. URL: https://medium.com/@christiansadi/building-an-artificial-intelligence-for-risk-part-1-minimax-and-the-branching-factor-83163a86289f.

[21] Christoph Salge and Tobias Mahlmann. "Relevant information as a formalised approach to evaluate game mechanics". In: *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. IEEE. 2010, pp. 281–288.

[22] Bernard Suits. *The grasshopper-: games, life and utopia*. Broadview Press, 2014.

[23] Katie Salen Tekinbas and Eric Zimmerman. *Rules of play: Game design fundamentals*. MIT press, 2003.

[24] Michael Thielscher. "A general game description language for incomplete information games". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 24. 1. 2010, pp. 994–999.

[25] CW Tijben. "Generating Gloomhaven dungeons through evolutionary game design". B.S. thesis. University of Twente, 2023.

[26] John Williams. *A look at the average American's monthly expenses*. Sept. 2022. URL: https://www.firstrepublic.com/insights-education/a-look-at-the-average-americans-monthly-expenses.