Towards a model to meet players' preferences in games

Mattia Bellini
University of Milan
Milan, Italy
bellinitia@gmail.com

ABSTRACT

Different have been the attempts to use Procedural Content Generation via Machine Learning in game development. Among the others, some researchers have tried to adapt a game, or some part of it, to the user playing it. This approach has been called "adaptive game design". Contrarily to what it may seem, apparently the most interesting findings in this field have been made for drama managers, *i.e.* for the artificial intelligences that procedurally generate story flow. The paper takes the move from what seems to be a missing in current literature and it is aimed at proposing and discussing a possible procedural content generation via machine learning model that takes the latest approaches in machine learning applied to drama managers and combine them with findings from adaptive game design. The objective of the proposed model is to give players the best possible gaming experience of a highly branched game, depending on their attitudes towards the gaming world.

KEYWORDS

Video games; Storytelling for video games; Procedural Content Generation via Machine Learning; PCGML; Adaptive game design; Drama manager.

INTRODUCTION

Procedural Content Generation via Machine Learning (PCGML) is a new paradigm for the self-driven creation of new content. The main difference with the mere procedural generation is the generally higher quality of the created content, achieved by integrating the procedural content generation (PCG) algorithm with a machine learning (ML) model trained on existing content.

PCGML has been applied to a variety of different content types and, by the time this paper is being written, it performed well particularly for the creation of images (*e.g.* using a Recurrent Neural Network [6]) and music (*e.g.* through «a probabilistic model based on distribution estimators conditioned on a recurrent neural network» [1]). Some of the techniques used in such studies have been applied in other domains, by knowledge transfer. The target[1] domains of these transfers included game development. More precisely, PCGML has been applied to game contents rather than games themselves, or, better told, on level design rather than game design. Researchers generally tried to automatically generate contents constituting levels, like maps, while the attempts to automatically generate or to adapt entire game environments have been less frequent. This is a direct reflection of the limits of present ML: the scarcity of available data for full game generation and the difficulty of creating a model able to generate an entire game from scratch.

As Summerville et al. [18] outline in their work, researchers of PCGML have applied different machine learning approaches in game studies, including artificial neural networks, Markov models, clustering and matrix factorization. Most of the works focused on the autonomous generation of levels, particularly for platformer games like *Super Mario Bros.* [11] (*e.g.* [18], [8]). There had been attempts to generate also contents different from mere game level, like *Magic: The Gathering* [24] cards [17] or stories for interactive fictions [7]. Some of the most interesting approaches in the field applied ML to drama managers (DM), to procedurally generate stories following players' behaviour. Similar studies have been made in the field of adaptive game design, *e.g.* to balance game difficulty to player's abilities. Following the idea of applying ML for a recognition of a player's attitudes, the focus of this paper is to discuss whether is possible an application of PCGML for the creation of a player-aware model capable of predicting user preferences and serving an adapted level progression, to maximize appreciation.

GHItaly19: 3rd Workshop on Games-Human Interaction, September 23rd, 2019, Padova (Italy)

ADAPTIVE GAME DESIGN

Using Procedural Content Generation, some researchers tried to adapt games to the player. Traditionally, these works were interested in modifying difficulty settings, using different techniques. Examples include the mechanics of different published games, like the aim assistant in *Max Payne* [13] that is more precise the less the player's abilities, or the opposing AI in *Mario Kart Wii* [12] that increases its skills when the player is performing too well. These studies are of little interest for the purposed of the model, since their aim is to adapt the game mechanics and difficulty but in no way the game itself, that is the aim of this research.

More interesting are semantic and declarative approaches, like the one found in Tutenel et al. [21]. Their model is based on a semantic definition of objects that includes all game-relevant information of a particular game object. These include functional information, possible relations to other objects and metadata of the game content. After having declared and assigned these data to the objects, it is possible to use them to better drive the new content creation process. Giving objects a semantic layer «helps convey the meaning and the role of an object in the virtual world, and consists of generic descriptions of classes of features, including attributes, properties, roles, relations, etc. This encourages the incorporation of further semantic information about player-dependent gameplay purposes, and how these can be used to control object generation» [10].

[10] points out also that a model that aims at providing a better game experience to the players by adapting the game to their play-time behaviour, needs a firm knowledge over what a player expects to play, to feel and in general to find in a product. This means, basically, three needings:

1. Have a solid player model and a way to capture player's expectation;
2. Quantify the expectation to a measurable level;
3. Process them and adapt the game consequently.

These three steps are of essential need for the creation of a model really capable of adapting contents to the player. Charles et al. [5] supports the idea of shaping a player model to capture her interests and playstyle, but also points a fourth need: the necessity of monitoring the player to constantly check for the effectiveness of a generated solution.

A last useful distinction made by [10] is the one between off-line customized generation and on-line adaptivity: the former is intended as a generation of contents while the game is not running, typically during the loading of a gaming session; the latter, on the contrary, describes the changes happening in run-time, just like the *Mario Kart Wii* [12] AI example seen above.

In the survey made by [10], however, an important lack in research emerges: the procedural generation of quests has been studied only from the point of view of placing goals and "keys" to reach them. Even though that is a relatively out-to-date survey, being dated back in 2011, this lack seems to be still present. Indeed, at the best of my knowledge, no new impactful studies have been conducted in this sense. On the contrary, many have been the works on improving the storytelling mechanisms on which DM are based, as we have already mentioned.

APPROACHES IN DRAMA MANAGERS

There have been several attempts in the field of DM to identify and classify a player in order to give her the story progression that best fits her tastes. In the survey made by Roberts and Isbell [14] we can find multiple examples, as also outlined by the more recent study conducted in [22]. For analytical purposes, the same four features presented in this lastly mentioned paper will be used to describe models found in literature. The four features are: *replayability* (possibility to play again the game without receiving the same gaming experience), *authorial control* (control over the game design process left in the hands of the author), *player autonomy* (freeness of the player in the gaming experience) and *adaptability* (capability of modifying the game to meet player's tastes). Later in the paper, we will also address the problem presented by a fifth feature, namely the *coordination*, *i.e.* the ability to orchestrate Non-Player Characters (NPCs) and other game elements to present specific experiences to the player.

Researches on DM present the closest approaches towards the model that this paper is aimed to propose. In particular, the approach of the *PaSSAGE* system [20] provides a good degree of adaptability through the identification of pre-defined players' styles. By assigning at each event of the game a weight for each style, the model chooses the most attractive event at every stage, ensuring autonomy to the player. To be noted is that this approach is deterministic, *i.e.* there is no degree of randomness. Thus, replayability is virtually zero, since the same actions will always result in the same reactions in the interactive fiction. This is also the reason for the high degree of authorial control of this model.

Very interesting is also the approach that emerges in *Implementation and Analysis of a Non-Deterministic Drama Manager* [22]. The aim of the authors here is to serve the best possible match between emergent player attitude towards the story and story progression, to provide a high degree of adaptability. The job is entrusted to a DM built on a genetic algorithm that ensure non-deterministic results, thus replayability, and the possibility to deal with a large number of blocks (here quests). To match player's preferences, the developers rely on a player model (PM) composed by a vector of three dimensions, representing the attitudes of the player towards the three regiments theorized by Durand in *The anthropological structures of the imaginary*. Quests are human-authored and described using similar vectors. This way, it becomes possible a computation of the concatenation, to evaluate the distance of the results of each possible sequence with player's preferences. After a probabilistic tournament selection, the tournament-winner concatenation is manipulated to minimalize the possibility of two identical results given the same premises. The model grants player autonomy by setting a minimum number of choices available to the player at each stage of the interactive fiction.

The model presented in [22] is an extremely useful starting point for the present theory. However, in the field of DM, the most interesting approach is the one showed in *Personalized Interactive Narratives via Sequential Recommendation of Plot Points* [26]. In the paper, the authors present a collaborative filtering approach, similar to the ones used in recommendation systems for services like Netflix and YouTube, applied to DM for the autonomous building of a story. Thanks to the collaborative filtering algorithm, the developers handed-off the complex problem of defining PMs to determine current users, as the different categories of users were grouped by the algorithm itself. An interesting advancement of this model is that it demonstrates how a progression-aware model has impactful benefits in the recommendation of subsequent plot nodes. On the other hand, a limit of the approach is that it is largely based on manual and explicit expression of positive and negative feedbacks via a review system, due to the non-pre-defined PM. The authors also implicitly pointed out a good practice to retain authorial control over story generation, *i.e.* the use of a branched scheme as a starting point from which to pick the blocks to be appended at each stage.

THE PROPOSED MODEL

The approach presented in [26] points out an extremely promising case of knowledge transfer: they demonstrated that sequential recommendation, frequently used to suggest complete fictional artefacts, is eligible to be applied also to shape only parts of a product, to best fit the tastes of audience in almost real-time. This promises to be a smart way to maximize player's appreciation of a game by adapting it at the levels of game design, story development and «*the logical flow of events and actions that follow*» [10]. However, Yu and Riedl's [26] decision to not rely on a pre-defined PM presents us a huge knowledge gap between what can be designed and what really players want. Notwithstanding the advantages of having self-built patterns that do not rely on any abstract theory, this approach is not returning any clearly readable data on user preferences but just, indeed, opaque patterns. For this reason, it might be quite more profitable to rely on a well-established PM. For the purpose of this model, I decided to base the PM on Stewart's theory[16], grounded in turn on Bartle's psychographic taxonomies [1]. Other theories might have been used, among the others Yee's model [25], or Bartle's three dimensions model [2]. I decided to discard Yee's because, with Bartle's word [3], «if you want a theory for [...] studying player psychology, then you may be better served by a straight taxonomy [...] such as Nick Yee's motivations». It should be clear that my intent is not to study the psychology of the players, but rather their preferred game style. On the other hand, Bartle's three dimensions model [2] presents a too-broad categorization, that becomes nearly impossible to handle in the development phase. Another possibility is to develop a custom categorization of players, but this is going further beyond the scope of the paper. I decided to base the model on Stewart's expansion of Bartle's taxonomies mainly for two reasons: firstly, it provides a clear and simple categorization of players that is well-grounded in literature and which validity has been tested different times. Secondly, describing the PM with four values guarantees an amount of information that is easily processable both by the algorithm and, more importantly, by the developers. However, the proposed model is not dependent on the way of describing the players and it is possible and easy to change the PM description if a better method is found.

Stewart in [16] describes the four taxonomies (Socializer, Killer, Explorer and Achiever) by binding them with specific actions performable in a generic game. On the basis of this theory, a player of the hypothetical single-player game based on the proposed model will be described by soft clusters, «in the sense that a particular player can have a degree of membership in each player type» [26], *i.e.* she will be characterized by a vector of four values ranging from 0 to 1, that quantify her degree of membership to each of the four taxonomies. The vector will then describe the *gaming persona* of the current player in a pretty accurate way.

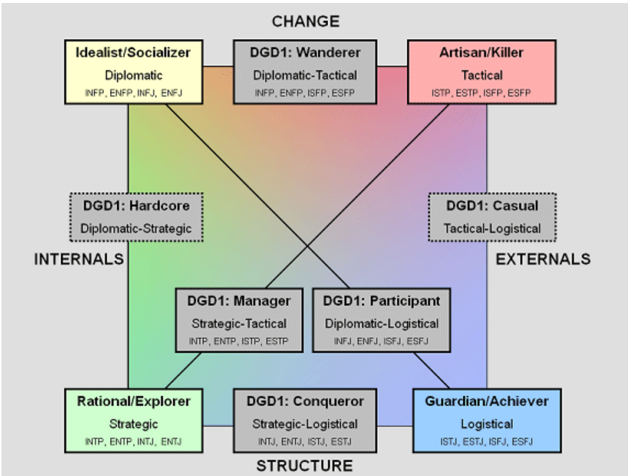

Figure 1 – Visual description of Stewart's expansion of Bartle's taxonomies [16]

Using such a model would surely require some care during level design, particularly for early-in-the-game levels. What would be needed are multi-solution problems to overcome. This might include both enemies and riddle to solve, obstacles, pathfinding moments, etc. The first step than is to assign a value for each of the taxonomies to the different possible actions, *i.e.* to the different ways of progressing in the events. Later, we can easily obtain a well-defined *gaming persona* of the player by adding those values to the player's profile. Indeed, to evaluate her disposition towards a taxonomy it will suffice to register the actions she performs and their semantic description, made using Stewart's expanded taxonomies. For example: a player that during the game tends to speak with all the NPCs and to solve problems in a "diplomatic way" might be labelled as a Socializer, while a player who tends to attack whatever is in sight might be labelled as a Killer. Again, this is an example: as said, the *gaming persona* are not defined as a set of mutually exclusive booleans, but rather as a set of values floating between 0 and 1. Indeed, a much more realistic representation of the player would be formed using "floats approach".

Furthermore, to better represent a player's attitude, it will be needed to weight different actions in a reasonable way. An accurate weighting is necessary to not overbalance an action regarding the others and thus to obtain a valid *gaming persona* of the player. Taking back the previous example: if actions are not differently weighted, a player that kills an evil slaver would end up with the same "Killer rate" as one who murders an innocent just for fun. In addition, a single action might have a (positive or negative) weight in two or more taxonomies, thus actions, too, need to be described as vectors of weights.

The proposed model relies therefore on PM recognition to procedurally generate content. The content is customized on player's profile and based on a PM built during an online (in-game) opaque survey: while normally playing, the algorithm registers player's path, formed by each action she chooses to perform, and updates in run-time her profile accordingly. However, the proposed procedural generation is offline (pre-game), meaning that the algorithm will select a block to present to the player as next level during the loading screen (*i.e.* in the time moving from one level to the subsequent) and not during an active game session, mainly to avoid slow-downs.

To decide which block to select, it will be used an approach similar to the one that can be found in [22]. The model will have a pool of developers-defined levels at each stage, in which are encapsulated *n* number of actions, having each a vector of traits corresponding to Bartle's taxonomies. A fitness function evaluates the levels in the pool by calculating the distance between available actions vectors and player's profile vector and return for each level a fitness rate. The fitness rate is then downscaled to a percentage, that is in turn used to probabilistically pick the level to show to the player as game progression. It might be more efficient to evaluate each block in advance, attaching a vector of properties to the levels and not to the single actions, in order to accelerate the concatenation process. This way, the algorithm would only have to evaluate levels as a whole, instead of each action separately. However, unfortunately, this approach probably presents its drawbacks, too: even if it is true that the algorithm could work faster, presenting the next level a few milliseconds in advance, on the other hand the blocks may need to be specifically designed to please a particular part of the audience. This might lead, in turn, to a regression to an almost deterministic model, especially after the PM will have reached a good level of precision. In addition, such an approach could mean more restricted possibilities in designing actions due to the specificity of the preferences of the audience for that block. This could make level blocks even more specific. In turn, it could lead to an even more specifically designed level, ending up with a system that, due to a "wrong" choice of the player in the early stage, could keep her in an unwanted path for the entire game. On the other hand, by evaluating each single action the designers can also include different taxonomies in each level. This way it will be possible to not present the player with actions belonging only to her preferred taxonomy, in order not to bore her with too similar tasks. To obtain the same result, if needed, it will also be possible to include a random factor during the pick of the blocks.

Picking from a set of author-made blocks, the algorithm will choose the one that would probabilistically provide the best possible experience for the player. Thus, for our previously instantiated player booleanly labelled as Socializer, the algorithm will more likely choose to concatenate levels with the most "problems" solvable via socializing, while for the Killer are more likely to be chosen the levels with the greatest number of possible enemies, and so on.

The advantages of the proposed approach are multiple and can be summarized as follows:

- Authorial control: using a defined pool of possible levels at each stage ensures a high degree of authorial control over the result. With this approach, each level is entirely created by authors: the PCG applies only on the concatenation. It generates the game, but not single levels. Authors can design the game and its story with a normal branching tree, just as [26] suggested;
- Player autonomy: the player keeps the autonomy she has in a general game, since there is no autonomy retention in the model itself. Constraints might be decided in the phase of actual development of game and levels;
- Adaptability: the whole model is intended to have a good level of adaptability, given the constraints of human capabilities to create levels. The proposed approach is not a PCGML model aimed at the autonomous generation of an uncountable number of games, or levels, of stories, but rather it is aimed at the maximization of the appreciation of a widely branched game. In addition, thanks to the evaluation of the single next step, players can change their attitude towards the game and its fictional world without being constraint in a narrow path, pre-determined by her early-in-the-game choices;
- Replayability: due to the probabilistic concatenation of levels, the model keeps a medium level of replayability, since the concatenation is not deterministic but, indeed, probabilistic;
- Coordination: coordination in the model is addressed incidentally, since there is no direct control of the model on the behaviour of NPCs and other game elements. The coordination arises here from the very fact that depending on the actions of the player, the concatenated levels will ideally be built to present a reaction of the environment to player's actions;
- Scalability: the evaluation of vectors of the individual actions found in a level ensures scalability, as it is possible to add or remove levels at each point of the game without impacting on the game progression, since the algorithm picks the best-fitting block in the provided pool. This being said, a *nota bene* is that the model does not evaluate the coherence of the game progression, that has to be addressed during the game design phase. This approach is scalable also in the sense that it is possible to modify the PM description to best fit the needing of each game built on the model;
- Single-person collaborative filtering: taking the example of the collaborative filtering approach found in [26], the proposed model will rely on a "single-person collaborative filtering". The model is based on a prediction of likeliness built on a series of positive and "non-positive" feedbacks. The feedbacks are given by player's choices of the actions to perform: the chosen action is a positive feedback, while all the other discarded possibilities are "non-positive" ones;
- Data scarcity: the main issue of the approach found in [26] is the heavy reliance on human-provided data quantity. For the model proposed in this paper, data scarcity is not an issue, mainly for two reasons: 1) the concatenation can be delayed until a certain amount of data over the player are collected, and 2) is virtually possible to design extremely dense levels that would give a relatively huge amount of information.

This model finds its place between level design and game design. Taking player's choices in levels as inputs, and outputting game design options through a recommendation system, might be a play-changing approach in PCGML applied to games. However, this approach shows us a challenge: on the one hand we would end up with a game that has the highest possible appreciation rate, due to the very fact that the game itself is shaped on the individual player attitudes. On the other hand, however, designing the proceeding of such a game requires particular care, above all for story progression. This is the main weak point of the model: it needs an expertise in storytelling and game design to keep the consistency of the story. To address this problem is probably preferable to keep levels relatively small-scaled: keeping in mind the Aristotelian unities of time, place and action when designing levels might be a good practice when this model is applied. Of course, constraints to the level concatenation can be applied in order to prevent a certain level being shown after another one that has nothing to do with the previous story. This does not mean, obviously, that the player cannot occur in major crossroads in the story.

FINAL CONSIDERATIONS AND FUTURE STUDIES

The current paper was aimed at proposing a PCGML model capable of adapting a game to players' attitudes and preferences. The starting point has been different models aimed at an adaptive game design. Notwithstanding some extremely useful best practices pointed out by such researches, no models actually similar to the one I was aiming to obtain have been found in this field. Instead, researches on DM seemed to be oriented more towards the direction my model was aimed at facing. Indeed, approaches that relied on a non-deterministic blocks concatenation [22] and on a collaborative filtering recommendation system [26] have been extremely useful for the theorization of the proposed model. This is based on the definition of a PM as a vector of four dimensions: each time the player performs an action, the PM is updated accordingly. At each new level, the algorithm probabilistically picks a subsequent block that is more likely to be interesting for the player, according to her current PM. A particular care is needed during the storytelling and game design phases, but every game that is not linear-paced needs expertise in game design and storytelling and the little additional care needed here is a little mite to be paid for what the model promises to do.

Further progress of the research will be, first of all, the development of the proposed model and its implementation in a game. However, there are also different other advancements that might be needed to obtain a completely valid result. Among the others, a better way of drawing a PM might be found. Indeed, as mentioned, the model is not dependent on the way of describing the players and it is always possible to modify the way the PM is calculated, described and stored. In addition, from the point of view of a storyteller, it might be very useful to conduct a proper research to analyse the new paradigm for addressing interactivity that emerges from the application of the proposed model. Also, it might be interesting to examine whether my approach presents restrictions in the stories or in the mechanics of a game based on it. Lastly, contrarily to what might be found in many researches on the field, I strongly believe that a PCGML model able to improve human design - rather than substituting it - can help the improvement of such approaches both in literature and in the industry. As of little help as it might be, I hereby encourage any studies aimed at this purpose.

REFERENCES

[1] R. A. Bartle. 1996. Hearts, clubs, diamonds, spades: Players who suit MUDs, in *Journal of MUD research*, Colchester.

[2] R. A. Bartle. 2005. Virtual Worlds: Why people play in *Massively Multiplayer Gamer Development*, 2, 1.

[3] R. A. Bartle. 2009. Understanding the Limits of Theory in *Beyond Game Design: Nine Steps to Creating Better Videogames*. Chris Bateman. Delmar.

[4] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription, arXiv preprint arXiv:1206.6392.

[5] D. Charles, A. Kerr, M. McNeill,M.McAlister, M. Black, J. Kücklich, A. Moore, and K. Stringer. 2005. Player-centred game design: Player modelling and adaptive digital games, in *Proc. DiGRA Conf. Changing Views—Worlds in Play*, Vancouver, BC, Canada, pp. 285–298.

[6] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. 2015. *Draw: A recurrent neural network for image generation* arXiv preprint arXiv:1502.04623.

[7] M. Guzdial, B. Harrison, B. Li, and M. O. Riedl. 2015. Crowdsourcing open interactive narrative in *the 10th International Conference on the Foundations of Digital Games*, Pacific Grove, CA.

[8] R. Jain, A. Isaksen, C. Holmgård, and J. Togelius. 2016. Autoencoders for Level Generation, Repair, and Recognition in *ICCC*.

[9] B. Li. 2015. Learning knowledge to support domain-independent narrative intelligence, Ph.D. Dissertation, Georgia Institute of Technology.

[10] R. Lopes, and R. Bidarra. 2011. Adaptivity Challenges in Games and Simulations: A Survey in *IEEE transactions on computational intelligence and AI in games*, vol. 3, no. 2.

[11] Miyamoto, T. Tezuka. 1985. *Super Mario Bros.*

[12] Nintendo EAD. 2008. *Mario Kart Wii.*

[13] Remedy Entertainment. 2001. *Max Payne*, Gathering of Developers.

[14] D. L. Roberts, and C. L. Isbell. 2008. A survey and qualitative analysis of recent advances in drama management. in *Transactions on Systems Science and Applications*, no. 3.

[15] M. Sharma, S. Ontañon, M. Mehta, and A. Ram. 2010. Drama Management and Player Modeling for Interactive Fiction Games in *Computational Intelligence Journal*, Volume 26 Issue 2

[16] B. Stewart. 2011. Personality and Play Styles: A Unified Model, Retrieved June 03, 2019 from: https://www.gamasutra.com/view/feature/134842/personality_and_play_styles_a_.php

[17] A. Summerville, and M. Mateas. 2016. Mystical tutor: A magic: The gathering design assistant via denoising sequence-tosequence learning.

[18] A. Summerville, and M. Mateas. 2016. Super Mario as a string: Platformer level generation via LSTMs, in *Proceedings of 1st International Joint Conference of DiGRA and FDG.*

[19] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. Hoover, A. Isaksen, A. Nealen, and J. Togelius. 2017. Procedural Content Generation via Machine Learning (PCGML) in *IEEE Transactions on Games.*

[20] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen. 2007. Interactive storytelling: A player modeling approach in *Proc. of the 3rd AIIDE07.*

[21] T. Tutenel, R. Bidarra, R. M. Smelik, and K. J. de Kraker. 2008. The role of semantics in games and simulations, in *Comput. Entertain.*, vol. 6, no. 4, pp. 1–35.

[22] M. N. R. Utsch, G. L. Pappa, L. Chaimowicz. 2018. Implementation and Analysis of a Non-Deterministic Drama Manager in *Proceedings of SBGames.*

[23] J. Valls-Vargas, S. Ontañon, J. Zhu. 2013. Towards Story-Based Content Generation: From Plot-Points to Maps, *IEEE Conference on Computational Inteligence in Games (CIG)*

[24] W. of the Coast. 1993. *Magic: The Gathering.*

[25] N. Yee. 2006. Motivations for play in online games in *CyberPsychology & Behavior*, 9, 6, pp. 772-775.

[26] H. Yu, and M. O. Riedl. 2014. Personalized Interactive Narratives via Sequential Recommendation of Plot Points in *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games.*