Game Artificial Intelligence: Challenges for the Scientific Community

Raúl Lara-Cabrera, Mariela Nogueira-Collazo, Carlos Cotta and Antonio J. Fernández-Leiva

Department "Lenguajes y Ciencias de la Computación", ETSI Informática, University of Málaga, Campus de Teatinos, 29071 Málaga – Spain
{raul,mnogueira,ccottap,afdez}@lcc.uma.es

Abstract. This paper discusses some of the most interesting challenges to which the games research community members may face in the area of the application of artificial or computational intelligence techniques to the design and creation of video games. The paper focuses on three lines that certainly will influence significantly the industry of game development in the near future, specifically on the automatic generation of content, the affective computing applied to video games and the generation of behaviors that manage the decisions of entities not controlled by the human player.
Keywords:Game Artificial Intelligence, Non-player characters, Procedural Content Generation, Affective Computing, Human-like behavior.

1 Introduction

According to the *Entertainment Software Association* [1] there are 155 million U.S. Americans that play videogames, with an average of two gamers in each game-playing U.S. households. The total consumer spend were 15.4 billion dollars in 2014 just in the United States. These figures show the good condition of the video game industry, which has taken the lead in entertainment industry. This situation has been a motivation for the research applied to video games, which has gained notoriety over the past few years, covering several areas such as psychology and player satisfaction, marketing and gamification, artificial intelligence, computer graphics and even education and health (serious games).

In the same way, the industry is beginning to adopt the techniques and recommendations academia offers. The reader interested in the current state of artificial intelligence techniques within the industry should refer resources such as the website AiGameDev[1], the *AI Summit* from the *Game Developers Conference*, the *AI Game Programming Wisdom* and *Game Programming Gems* book collections, or the book by Ian Millington and John Funge [22].

Research in artificial intelligence may take advantage of the wide variety of problems that videogames offer, such as adversarial planning, real-time reactive behaviors and planning, and decision making under uncertainty. For instance,

[1] http://aigamedev.com

real-time strategy games, which are a portion of the whole videogames, are being used as testbeds and frameworks for brand new artificial intelligence techniques, as stated by our previous study on real-time strategy and artificial intelligence [18].

This paper aims at some interesting trends that seems to guide the future of videogames, and the challenges that they offer to academia, focusing on the application of artificial intelligence and, more precisely, computational intelligence, i.e. bio-inspired optimization techniques and meta-heuristics [20]. We want to clarify that the universe of uses of optimization techniques on the development and game design is extremely broad and we do not pretend to make an exhaustive tour on it in this paper; in fact we recommend the interested reader a reading of other papers that have been published in the literature and serve as a basis to learn about the state of the art [37,43]. We focused, instead, on certain research areas that will influence significantly in the creation of commercial games over the next decade, we refer to the procedural content generation, affective computing, which has an impact in player satisfaction and the creation of behaviors or strategies of decision making for non-playable characters (NPC).

2 Procedural Content Generation

Procedural content generation refers to the creation of videogame content through algorithmic means, such as levels, textures, characters, rules and missions (the content may be essential to game mechanics or optional). The fact of procedurally generate the content may reduce the expenses of hiring many designers in order to create the content manually and even being a source of inspiration for them, suggesting novel designs. Moreover, it is possible to establish some criteria the generated content must meet, such as adjusting the created level to the player's game style in order to offer a continuous challenge to her. If the generation process is made in real time and the content is diverse enough then it may be possible to create real infinite games, which offers the player a brand new gaming experience every time she starts a new one. These benefits are well-known by the industry, as exposed by the use of these kind of techniques in successful commercial games such as *Borderlands* saga, *Skyrim*, *Minecraft* and *Terraria*.

Many distinctions may be drawn when dealing with procedural content generation and its procedures. Regarding when the content is generated, it might be during the execution of the game (online generation) or during the development (offline generation). If we speak about the main objective of the generated content, this could be necessary for the game progression, hence it is mandatory to ensure that the content is valid, or it should be optional, as the decoration of levels.

Another question is the nature of the generation algorithm, that is, if we have a purely stochastic algorithm, in which content is created from random seed, or, conversely, a deterministic algorithm, where the content is generated by a parameter vector. The third possibility is the hybridization of both perspectives, designing an algorithm with a stochastic and a deterministic component working together.

Looking at the objectives to be met, the creation process can be done in a constructive manner, ensuring the validity of the content throughout the process. The other option is to follow a scheme of generating and testing, where a lot of content, which goes through a validation phase and subsequent disposal of that which does not comply with the restrictions, is generated. The latter scheme is the most currently employed by the research community, and it is based on the search of the content in the space of possible solutions. The validation is done by assigning values to content so that its level of quality is quantified according to the objectives. Apart from maps and levels, there are other examples of content that may be generated procedurally such as music [7], stories for role-playing games [27], game rules [11] and quests [27].

These techniques are commonly used to generate maps and levels, as evidenced by the large number of papers devoted to this issue [14]. For instance, authors of [17] approach the problem of matchmaking in multiplayer videogames evolving maps for a first-person shooter in order to improve the game balancing for certain combinations of player skills and strategies. With a similar content type, authors of [10] presented a genetic algorithm for the generation of levels for the Angry Birds game whose objective is to minimize the elements' movement during a period of time, obtaining stable structures during the process.

3 Affective computing and player satisfaction

It was Rosalind Picard who in 1995 introduced the term Affective Computing and defines it as the computation which relates arises or influences emotions [30]. In the context of video games, there is still research on how to extrapolate the vast field of emotions onto a game's stage, the good reasons why to do it are clear [12], but the results obtained so far are usually modest compared to everything academia expect to achieve.

One of the earliest forms used to incorporate emotions in games was through the narrative, by generating situations that caught the player either by the characters, the incorporation of all kinds of conflicts and fantastic stories or by real life situations (*Final Fantasy* and *Resident Evil* sagas use this kind of emotional narrative). In a similar way, there are other games that stand out due to the high realism of their simulations and the incorporation of emotions to the main character (see Figure 1). This kind of affective focused on the main character requires a lot of artistic work in order to simulate the emotions in a realistic manner (it is common to use motion capture techniques and hire professional actors). For instance, the videogame *Beyond: Two Souls* heavily relies on the narrative and guides the player through a predefined story, hence limiting the possible actions the player may take so it is easier for the software to have control over the emotional flow of the main character. Generally, this approach to the implementation of emotion has as main objective to increase the user's immersion into the game and to do so, it sets emotional dynamics between the human player and the main character, so the player r empathizes with the actions her character conveys.

Fig. 1. Characters from *Heavy Rain* and *The Walking Dead*

There is another approach which aims to make the non-playable characters (NPCs) involved in the game behave as emotional individuals and, therefore, their emotions should influence the game whenever they take decisions. In order to achieve this, NPCs must have a high level of perception because they must react to every event that happens around them which could affects their emotional state, such as shrill sounds or fire. In this sense there exist interesting proposals that offer techniques and models devoted to implement emotional behaviors for the virtual agents [15,28,29].

The aforementioned approaches manages the affectivity through the virtual agents, focusing on what happens inside the videogame without taking into account the emotional relationships that players express while they are playing the game. This situation led us to a third approach, namely *self-adaption in affective videogames*, which is related to the field of modeling, evaluation and increase of player satisfaction [25,39]. Self-adaption refers to the ability of a game to take into account the preferences and gaming style of the player and react to these features increasing the player satisfaction and making a unique gaming experience for each player. In the case of affectivity, the game should self-adapt depending on the emotions the player expresses while she is playing, establishing emotional links between human and NPCs.

The most interesting papers in this field are focused mostly on creating a formal model that represents the behavior of the player so it is possible to evaluate her level of satisfaction, all based on psychological research on satisfaction [8,21,35]. Then one has to consider the employment of this model to determine what level of satisfaction experiences the player and then proceed to readjust the game to maintain or raise that level. Some successful proposals in this sense are [13,40,41,42]

Each approach mentioned here is an open research field having several lines that also demand new solutions, and optimization is one of them. Most successful proposals out there, some of which have been cited here, consider a search process, based on meta-heuristics, to explore the broad search space that is generated from two inherently complex contexts: videogames and emotions.

4 Behaviors

Traditionally the Artificial Intelligence (AI) of a game has been coded manually using predefined sets of rules leading to behaviors often encompassed within the so called *artificial stupidity* [19], which results in a set of known problems such as the feeling of unreality, the occurrence of abnormal behaviors in unexpected situations, or the existence of predictable behaviors, just to name a few. Advanced techniques are currently used to solve these problems and achieve NPCs with rational behavior that takes logical decisions in the same way as a human player. The main advantage is that these techniques perform automatically the search and optimization process to find these "smart" strategies.

Bio-inspired algorithms are the basis of many of these advanced methods, as they are a suitable approach in this regard, because they are able to produce solutions of great complexity as an emerging result of the optimization process, and its adaptive capacity allows them to incorporate information provided by the user. Due to this, there are several successful proposals that follows this approach. For instance, co-evolution [36] is one of the heuristic techniques inspired by the natural evolution principles that has been widely used in videogame AI programming. In [34] the author presents a research that were capable of evolving the morphology and behaviour of virtual creatures through competitive co-evolution that interact in a predator/prey environment. Other interesting papers used co-evolution to obtain game strategies for artificial players of a war game called Tempo [2,3,16,26].

Machine learning is used as well when modeling the behavior of artificial players. Authors of [31] have used self-organizing maps in order to improve the maneuvering of platoons in a real-time strategy game. By analyzing data obtained from the sensors, the authors of [32] have developed an algorithm for an artificial pilot so it is able to learn the race track and drive through it autonomously. In [9], the authors obtained several features from the maps of a real-time strategy game and use them to determine a NPC's behavior.

4.1 Notable challenges

We are immersed in an era of resurgence of artificial intelligence that directly influences the development of game AI and, as a consequence, the generation of decision making mechanisms for NPCs provides an exciting challenge to the scientific community as this goal can be conducted from many different points of views. In the following we enumerate some of the most exciting directions in which the development of game NPC behaviors can be done.

Human-like behaviors Current players demands highest quality opponents, what basically means obtaining enemies exhibiting intelligent behavior; in addition, especially, in on-line games, it is well-known that players enjoy playing against other human players, however it is also known that many of the players involved in Massively multiplayer online (MMO game) are bots created by the game developers and this can reduce the immersion of the player in the game. Therefore, developers make a significant effort (in terms of funds) to generate bots that simulate to play in a human style with the aim of providing human players the sensation of being facing other 'human' players (that in fact might be non-human).

In a more wide context, this can be translated to an adaptation of the Turing Test [38] to the field of the videogame development. The basic fundamental is that an NPC that plays like a human might be considered as a human player, as the NPC would not be distinguished from a human player (assuming the judge to assess the humanity of the bot is not allowed to see -physically speaking- the players).

However, one of the main problems that developers find to cope with the objective of generating human-like bots is that it is not easy to evaluate what a 'human-like intelligence' means for a bot in videogames. This precisely is one of the main problems, a hard problem, and moves the context to a psychological scenario that introduces more complexity to its solving.

In this context, the "2k bot prize" is a competition that proposes an interesting adaptation of the Turing test in the context of the well-known FPS game Unreal Tournament 2004, a multi-player online FPS game in which enemy bots are controlled by some kind of game AI. More information on this is provided below.

General Game Playing (GGP) Can a bot play different games without being previously specifically trained for them? This is basically the question that underlies the research to generate automated general game players. In some sense, this issue is related to the creation of human-like behaviors, as a general player mimics a human that learns the rules of a game and subsequently is able to play it without being previously trained on it. The skill to play would be acquired with the game experience and this is other of the fundamentals under the GGP concept.

As said in [4], "A general game player (GGP) is a computer program that plays a range of games well, rather than specializing in any one particular game"

Recently, GGP obtained public recognition via *DeepMind*, an artificial intelligence - developed by a private company associated to the giant Goggle- that was able to master a diverse range of Atari 2600 games; this general player consists of a combination of Deep Neural Networks with Reinforcement Learning [23].

GGP opens grand challenges not only for the community of development of games, but for the society in general.

Other issues the recent boom of casual games played in mobile devices provokes that both the Design and Gameplay of games demand resources that appear and evaporate continuously during the execution of a game. This precisely occurs in the so-called pervasive games (i.e., "games that have one or more salient features that expand the contractual magic circle of play spatially, temporally, or socially" [24]) where the gaming experience is extended out in the real world. Playing games in the physical world requires computations that should be executed on-the-y in the user's mobile device and having into account that players can decide to join or drop out the game in each instant. The application of AI techniques can help to improve the immersion of the player by generating automatically new objectives or imposing constraints to the game. This is an open issue that demands more research in a future close.

4.2 Competitions

Over the past few years different competitions where researchers have the opportunity to compare their strategies and algorithms in specific scenarios and games have appeared. Some examples of the most important are listed along with a brief description:

- **2K BotPrize[2]:** The objective is to develop an Unreal NPC capable of tricking human players to believe it is human as well
- **Starcraft AI Competition[3]:** Anual competition of Starcraft NPCs that fight each other to be the winner
- **Simulated Car Racing Competition[4]:** The objective is to develop an artificial car driver that competes in a virtual racing championship
- **GVG-AI[5]:** The *General Video Game AI Competition* is a competition where the artificial players might be capable of playing several game genres, trying to be as generic as possible

There is a problem with these competitions: the challenges are very specific and closely linked to the game on which they are played. Thus, strategies for winning over-specialize in exploiting the features of the game itself, but throwing a poor performance when they are used in another game. Therefore, another possible challenge is to design generic competitions to be able to give good results not only in a single game or environment, but in several of them, something that is already being tackled in the previously outlined *GVG-AI* competition.

[2] http://botprize.org/
[3] http://webdocs.cs.ualberta.ca/~cdavid/starcraftaicomp/
[4] http://cig.dei.polimi.it/
[5] http://www.gvgai.net/

5 Tools and frameworks

This section is devoted to present the tools or frameworks that the scientific community has at its disposal for testing and validation of the results obtained during the research. Nowadays there are many tools freely available, so following there is a collection of the most used with their main features, to serve as a reference list to researchers in artificial intelligence and videogames.

Fig. 2. A screenshot of Starcraft

ORTS (Open Real-Time Strategy) [5] is a real-time strategy game designed specifically as a research tool and published under the GLP (GNU public license). It features an open message protocol and the client application let the researchers analyze the performance of their algorithms playing games in a controlled environment where the simulation takes place on the server side. Another strategy game that is widely used as a research tool is Starcraft[6], which features a software library (BWAPI[7]) that helps to connect the game engine with AI strategies (see Figure 2. Furthermore, RoboCode[8] is a platform whose objective

[6] `http://us.blizzard.com/en-us/games/sc/`.
[7] `http://bwapi.github.io`
[8] `http://robocode.sourceforge.net/`

is to develop a combat robot using Java or .NET to fight against other robots in real time.

Planet Wars[9] y ANTS[10] are two games developed for the AI competition hosted by Google in 2010 and 2011, respectively. The former is a space conquest game for many players whose objective is conquer all the planets in a map, the latter is a multiplayer game as well where every player represents a set of ants whose objective is gather food and conquer their opponents' anthills.

In Vindinium[11] the player has to take the control of a legendary hero using the programming language of her choice to fight with other AI for a predetermined number of turns. The hero with the greatest amount of gold wins.

Eryna[12] [6] is another tool created to support the research on AI applied to videogames. It is a real time multiplayer game that lets the user launch games between several NPCs and evaluate the results. Their fundamental components are: the game engine that follows an authoritative server architecture and process concurrently several connections and processes, an AI module that is fully customizable and lets the researcher develop her own NPCs and a module for procedural content generation capable of generate new maps.

SpelunkBots[13] is a tool-set developed from the source code of the platform game Spelunky. It provides the researchers an easy way to code an artificial player for this game. The tool has been developed by Daniel Scales [33].

6 Conclusions

The challenges in the research lines that we have mentioned throughout this paper are huge and certainly affect other areas beyond the field of video games. For instance, the generation of quasi-human behavior is something that is being already investigated and traditionally have their seed in the famous "Turing test". The possibilities opened up by applying science to videogames are vast: the integration of feelings in artificial players and the option to build a direct channel between them and the sentimental perception of the player through the so-called Affective Computing.

Regarding procedural content generation, it has been shown that is a hot field in academia, with a large number of papers related to it. Moreover, the videogame industry is successfully using many of the advances obtained by academia, although there are many non-tackled challenges in this sense.

We end this paper mentioning that there are many areas related to the use of Computational/Artificial Intelligence that have not been specifically described here, where researches might find additional challenges such as player modeling, computational narrative and AI-assisted game design among others. We are dealing with stimulating challenges not only for the near future, but the present.

[9] `http://planetwars.aichallenge.org/`
[10] `http://ants.aichallenge.org/`
[11] `http://vindinium.org/`
[12] `http://eryna.lcc.uma.es/`
[13] `http://t2thompson.com/projects/spelunkbots/`

Acknowledgements

This work has been partially supported by Junta de Andalucía within the project P10-TIC-6083 (DNEMESIS[14]), by Ministerio de Ministerio español de Economía y Competitividad under (preselected as granted) project TIN2014-56494-C4-1-P (UMA-EPHEMECH), and Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

References

1. Association, E.S.: Essential facts about the computer and video game industry (2015)
2. Avery, P., et al.: Coevolving a computer player for resource allocation games: using the game of Tempo as a test space. Ph.D. thesis, School of Computer Science University of Adelaide (2008)
3. Avery, P.M., Michalewicz, Z.: Static experts and dynamic enemies in coevolutionary games. In: IEEE Congress on Evolutionary Computation. pp. 4035–4042 (2007)
4. Browne, C., Togelius, J., Sturtevant, N.R.: Guest editorial: General games. IEEE Trans. Comput. Intellig. and AI in Games 6(4), 317–319 (2014)
5. Buro, M.: ORTS: A hack-free RTS game environment. In: Schaeffer, J., et al. (eds.) Computers and Games. Lecture Notes in Computer Science, vol. 2883, pp. 280–291. Springer (2002)
6. Collazo, M.N., Leiva, A.J.F., Porras, C.C.: Eryna: una herramienta de apoyo a la revolución de los videojuegos. In: Camacho, D., Gómez-Martín, M.A., González-Calero, P.A. (eds.) Proceedings 1st Congreso de la Sociedad Española para las Ciencias del Videojuego, CoSECivi 2014, Barcelona, Spain, June 24, 2014. CEUR Workshop Proceedings, vol. 1196, pp. 173–184. CEUR-WS.org (2014)
7. Collins, K.: An introduction to procedural music in video games. Contemporary Music Review 28(1), 5–15 (2009)
8. Csikszentmihalyi, M., Csikzentmihaly, M.: Flow: The psychology of optimal experience, vol. 41. HarperPerennial New York (1991)
9. Fernández-Ares, A., García-Sánchez, P., Mora, A.M., Merelo, J.: Adaptive bots for real-time strategy games via map characterization. In: Computational Intelligence and Games (CIG), 2012 IEEE Conference on. pp. 417–721. IEEE (2012)
10. Ferreira, L., Toledo, C.: A search-based approach for generating angry birds levels. In: Computational Intelligence and Games (CIG), 2014 IEEE Conference on. pp. 1–8. IEEE (2014)
11. Font, J., Mahlmann, T., Manrique, D., Togelius, J.: A card game description language. In: Esparcia-Alcázar, A. (ed.) Applications of Evolutionary Computation, Lecture Notes in Computer Science, vol. 7835, pp. 254–263. Springer Berlin Heidelberg (2013)
12. Freeman, D.: Creating emotion in games: The craft and art of emotioneering™. Comput. Entertain. 2(3), 15–15 (Jul 2004)
13. Halim, Z., Baig, A.R., Mujtaba, H.: Measuring entertainment and automatic generation of entertaining games. International Journal of Information Technology, Communications and Convergence 1(1), 92–107 (2010)

[14] http://dnemesis.lcc.uma.es/wordpress/

14. Hendrikx, M., Meijer, S., Van Der Velden, J., Iosup, A.: Procedural content generation for games: A survey. ACM Trans. Multimedia Comput. Commun. Appl. 9(1), 1:1–1:22 (Feb 2013)
15. Johansson, A., Dell'Acqua, P.: Emotional behavior trees. In: Computational Intelligence and Games (CIG), 2012 IEEE Conference on. pp. 355–362. IEEE (2012)
16. Johnson, R., Melich, M., Michalewicz, Z., Schmidt, M.: Coevolutionary Tempo game. In: Evolutionary Computation. CEC'04. Congress on. vol. 2, pp. 1610–1617 (2004)
17. Lanzi, P.L., Loiacono, D., Stucchi, R.: Evolving maps for match balancing in first person shooters. In: Computational Intelligence and Games (CIG), 2014 IEEE Conference on. pp. 1–8. IEEE (2014)
18. Lara-Cabrera, R., Cotta, C., Fernández-Leiva, A.J.: A Review of Computational Intelligence in RTS Games. In: Ojeda, M., Cotta, C., Franco, L. (eds.) 2013 IEEE Symposium on Foundations of Computational Intelligence. pp. 114–121 (2013)
19. Lidén, L.: Artificial stupidity: The art of intentional mistakes. AI Game Programming Wisdom 2, 41–48 (2003)
20. Lucas, S.M., Mateas, M., Preuss, M., Spronck, P., Togelius, J. (eds.): Artificial and Computational Intelligence in Games, Dagstuhl Follow-Ups, vol. 6. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2013)
21. Malone, T.W.: What makes things fun to learn? heuristics for designing instructional computer games. In: Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems. pp. 162–169. ACM (1980)
22. Millington, I., Funge, J.: Artificial intelligence for games. CRC Press (2009)
23. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature 518(7540), 529–533 (2015)
24. Montola, M., Stenros, J., Waern, A.: Pervasive Games. Morgan Kaufmann, Boston (2009)
25. Nogueira, M., Cotta, C., Leiva, A.J.F.: On modeling, evaluating and increasing players' satisfaction quantitatively: Steps towards a taxonomy. In: et al., C.D.C. (ed.) Applications of Evolutionary Computation - EvoGAMES. Lecture Notes in Computer Science, vol. 7248, pp. 245–254. Springer (2012)
26. Nogueira, M., Cotta, C., Leiva, A.J.F.: An analysis of hall-of-fame strategies in competitive coevolutionary algorithms for self-learning in RTS games. In: Nicosia, G., Pardalos, P.M. (eds.) Learning and Intelligent Optimization - 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7997, pp. 174–188. Springer (2013)
27. Onuczko, C., Szafron, D., Schaeffer, J., Cutumisu, M., Siegel, J., Waugh, K., Schumacher, A.: Automatic story generation for computer role-playing games. In: AIIDE. pp. 147–148 (2006)
28. Peña, L., Ossowski, S., Peña, J.M., Sánchez, J.Á.: EEP - A lightweight emotional model: Application to RPG video game characters. In: Cho, S., Lucas, S.M., Hingston, P. (eds.) 2011 IEEE Conference on Computational Intelligence and Games, CIG 2011, Seoul, South Korea, August 31 - September 3, 2011. pp. 142–149. IEEE (2011)
29. Peña, L., Peña, J.M., Ossowski, S.: Representing emotion and mood states for virtual agents. In: Klügl, F., Ossowski, S. (eds.) Multiagent System Technologies -

9^{th} German Conference, MATES 2011, Berlin, Germany, October 6-7, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6973, pp. 181–188. Springer (2011)
30. Picard, R.W.: Affective computing. MIT Media Laboratory (1995)
31. Preuss, M., Beume, N., Danielsiek, H., Hein, T., Naujoks, B., Piatkowski, N., Stür, R., Thom, A., Wessing, S.: Towards intelligent team composition and maneuvering in real-time strategy games. IEEE Transactions on Computational Intelligence and AI in Games 2(2), 82–98 (2010)
32. Quadflieg, J., Preuss, M., Kramer, O., Rudolph, G.: Learning the track and planning ahead in a car racing controller. In: Computational Intelligence and Games (CIG), 2010 IEEE Symposium on. pp. 395–402. IEEE (2010)
33. Scales, D., Thompson, T.: Spelunkbots API-an AI toolset for spelunky. In: Computational Intelligence and Games (CIG), 2014 IEEE Conference on. pp. 1–8. IEEE (2014)
34. Sims, K.: Evolving 3D Morphology and Behavior by Competition. Artificial Life 1(4), 353–372 (1994)
35. Sweetser, P., Wyeth, P.: Gameflow: a model for evaluating player enjoyment in games. Computers in Entertainment (CIE) 3(3), 3–3 (2005)
36. Thompson, J.N.: The geographic mosaic of coevolution. University of Chicago Press (2005)
37. Togelius, J., Yannakakis, G.N., Stanley, K.O., Browne, C.: Search-based procedural content generation: A taxonomy and survey. IEEE Trans. Comput. Intellig. and AI in Games 3(3), 172–186 (2011)
38. Turing, A.: Computing machinery and intelligence. Mind 59(236), 433–460 (1950)
39. Yannakakis, G.N.: How to model and augment player satisfaction: a review. In: WOCCI. p. 21 (2008)
40. Yannakakis, G.N., Hallam, J.: Towards capturing and enhancing entertainment in computer games. In: Advances in artificial intelligence, pp. 432–442. Springer (2006)
41. Yannakakis, G.N., Hallam, J.: Towards optimizing entertainment in computer games. Applied Artificial Intelligence 21(10), 933–971 (2007)
42. Yannakakis, G.N., Hallam, J.: Real-time game adaptation for optimizing player satisfaction. Computational Intelligence and AI in Games, IEEE Transactions on 1(2), 121–133 (2009)
43. Yannakakis, G., Togelius, J.: A panorama of artificial and computational intelligence in games. Computational Intelligence and AI in Games, IEEE Transactions on PP(99), 1–1 (2014)