Evolving NPC Behaviours in A-life with Player Proxies

Vadim Bulitko
Computing Science
University of Alberta
Edmonton, Alberta
bulitko@ualberta.ca

Mac Walters
BioWare
Electronic Arts
Edmonton, Alberta
mac@bioware.com

Morgan Cselinacz
Psychology
University of Alberta
Edmonton, Alberta
cselinac@ualberta.ca

Matthew R. Brown
Computing Science
University of Alberta
Edmonton, Alberta
mbrown2@ualberta.ca

Abstract

Game development costs are on the rise as players expect massive open-world games populated with richly interactive non-playable characters (NPC). Procedural content generation has the potential to reduce the development costs as well as make the content more player-specific. Recent work on evolving artificial intelligence for NPCs focused on combat between NPCs and the player. In this paper we propose to evolve an ecosystem of NPCs for a broader class of games. To allow players' actions and playstyles to inform the evolution, we propose to model actual players and create AI-controlled proxies to evolve NPCs against. The approach removes the time pressure from the evolution and allows for traditional quality assurance methods while keeping the evolution player-informed.

1 Introduction

Modern video games such as *Fallout 4* (Bethesda Game Studios 2015) invite players to spend hundreds of hours exploring vast open worlds. The story delivery and world exposition critically depend on the depth of interactions with numerous non-playable characters* (NPCs) controlled by Artificial Intelligence (AI). Most games create only an illusion of habitation. For instance, the NPCs of *Fallout 4* staff merchant booths in the day and assume a sleeping position at night. They likely do not depend on the in-game economy or sleep to survive but merely execute behaviours scripted by game developers.

With players expecting progressively richer, more detailed, temporally extended interactions (Delahunty-Light 2018), the costs of manually scripting NPCs continue to rise. Even major game developers struggle to populate worlds with enough interesting interactions. It is these limitations that are currently holding back most developers from achieving truly dynamic, reactive living worlds and encounters with NPCs. Additionally, while manually scripting NPC behaviours enables traditional quality-assurance techniques, the resulting behaviours are limited, rigid, or simply randomized, breaking the illusion of a rich lived-in game world.

Procedural content generation is an active area of research that strives to generate in-game content algorithmically (Togelius et al. 2013). While many types of content can be procedurally generated, we focus on Artificial Intelligence controlling non-playable characters. Generating such AI procedurally opens a door to doing it on a per-player basis in an attempt to have meaningfully customized player-specific experiences. In this paper, we follow in the footsteps of recent work on enemy AI generation via a simulated Darwinian evolution (Soule et al. 2017). Polymorphic Games' commercial game *Darwins Demons* evolved space-invader-like creatures depending on the players' strategy by defining a per-round fitness function. As evolution normally takes a large number of generations (i.e., game rounds), the space of possible creatures and the mutation rate have to be carefully constrained to make the on-line evolution fast enough so that a single player can see its effects. In this paper, we propose an alternative by moving the evolution *off-line* (i.e., onto company servers without a direct involvement of players). We keep evolution player-specific by evolving NPCs against player agents modeled after real players.

This paper is a substantially extended version of our previously published one-page abstract (Bulitko et al. 2018). The additions include a significantly more detailed problem formulation, related work analysis, a description of the multi-stage on-line/off-line evolution and present state of the project as well as related philosophical questions.

2 Problem Formulation

The problem we are proposing to solve is to procedurally generate AI for non-playable characters in video games. Such generation should (i) be light on game developer labour, (ii) create reliable AI which allows for traditional quality assurance methods and (iii) take the players' behaviour into account to facilitate player-specific game experiences. By the latter, we mean deep and global effects of the players' actions. For instance, by killing all ghouls in *Fallout 4*, the player should be able to irreversibly affect the entire ecosystem in a non-trivial way. We will refer to this effect as *persistent adaptation*.

We will measure the effectiveness of our approach by measuring play time and using the measure as a proxy to

*In this paper we use the term NPC broadly to mean all non-playable AI-controlled characters in a game. This includes more prominent characters as well as more ambient creatures. We propose to apply our approach to evolve simpler/more ambient NPCs first, eventually progressing to more complex/prominent NPCs.

players' engagement level.

3 Related Work

Procedural content generation creates game content mostly or entirely automatically (Togelius et al. 2013). Evolutionary algorithms (e.g., neuroevolution) have often been used to create NPC behavior in a well-defined competitive setting (Risi and Togelius 2017; Soule et al. 2017). We are interested in generating ambient NPCs that contribute to a believable and immersive world primarily through their AI-controlled group behaviour rather than through their appearance or attributes (Ruela and Guimarães 2017).

For instance, *No Man's Sky* (Hello Games 2016) and *No Man's Sky Next* (Hello Games 2018) boast an impressive variety of aesthetically diverse procedurally generated worlds with various flora, fauna and creatures to interact with. However, the interactions themselves are too shallow and repetitive to encourage exploration. In contrast, the side quest *Come Fly with Me* in *Fallout: New Vegas* (Obsidian Entertainment 2010) also involves a space-faring mission, is beautifully hand-crafted and leaves a long-lasting impression, unlike many encounters in the *No Man's Sky* games.

Consequently, game developers usually hand-craft NPC behaviours, which is either expensive or appears repetitive if the same behaviour scripts/trees are reused for many NPCs. Additionally, such canned behaviours/interactions do not facilitate world-scale changes (unless specifically programmed in) and thus lack persistent adaptation.

Recent work attempted to evolve NPCs *on-line* (i.e., during game play), in response to players' actions and strategies. Doing so, however, required an easily computable fitness function, which works better in a well-defined competitive setting (Risi and Togelius 2017; Soule et al. 2017; Polymorphic Games 2018). Furthermore, as evolution normally takes many generations to deliver interesting artifacts, on-line/in-game evolution has to be greatly sped up so that the player can see its effects before they lose interest in the game. Doing so, for instance, by setting the mutation rate unusually high has undesirable consequences as the process becomes more random, obscuring meaningful responses to players' strategies. Furthermore, conducting evolution on-line precludes traditional quality assurance methods which may make game developers feel uneasy. Since evolution is an inherently randomized process, guaranteeing interesting outcomes is also problematic.

4 Proposed Approach

As discussed above, recent efforts on using evolution for NPC AI focused on well-defined competitive games, did not allow for traditional quality assurance and could potentially result in seemingly random responses to players' strategies or simply a lack of interesting evolved behaviours.

We propose to address all of these shortcomings by adapting the flipped classroom model presently becoming popular in academia. In such a model, most material is made available on the Internet and is studied by the students at their own pace, outside of the class time. The students then bring their questions to the lecture/lab periods where they interact with an actual instructor (Lage, Platt, and Treglia 2000).

Instead of evolving a single type of NPC combating the player, we adapt the A-life setting similar to the one used by Bulitko et al. (2017) and Soares et al. (2018) in which NPCs form an ecosystem complete with multiple species and resources. We conduct the evolution off-line on servers at the game studio, which removes the time pressure and allows for traditional quality assurance methods. It also allows game developers to run multiple evolutions and select the one with more interesting evolved behaviours. Such detection of interesting behaviour can even be automated (Soares et al. 2018).

However, such A-life based evolution of NPC behaviours is not responsive to players' actions. Thus, we borrow the idea of drivatars (Turn 10 Studios 2013) and player modeling (Thue et al. 2007) and put the players' behaviour back into the evolution in the form of a non-evolving AI agent representing a player. The process then proceeds in stages as depicted in Figure 1.

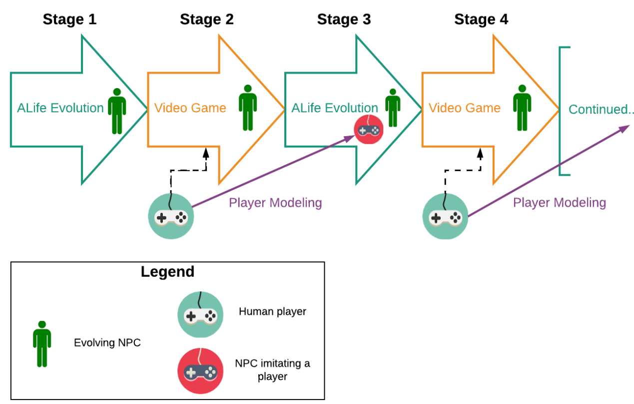

Figure 1: Multi-stage evolution of NPC AI in A-life.

At first, NPCs evolve in A-life without any player input (Stage 1). Then the evolved NPC behaviour is pushed out to the player base via a digital download. The players then interact with the evolved NPCs in game, and their interactions are recorded and sent back to the studio (Stage 2). The interactions are used to create AI agents approximating the players behaviours.† The resulting AI agents (i.e., "drivatars" or player proxies) are then put in the A-life environment and inform the next stage of NPC evolution (Stage 3). The process is then repeated with the evolution being informed by the actual human players at even-numbered stages and by player proxies at odd-numbered stages.

5 Current State and Future Work

We currently have an A-life environment consisting of a simple predator-prey model in a 2D grid environment. Each NPC is controlled by its own deep artificial neural network which observes the world and selects the next action. The

†We propose to cluster observed human player behaviours and generate a single AI agent per cluster.

network perceives the world as the raw pixel color values of neighboring grid cells. The network has convolutional layers; its topology and innate weights are encoded in the NPC genes. As the agents evolve, so do their brains (i.e., the convolutional networks). Larger networks have a great potential for more complex behaviour but also consume more energy. The simulated evolution does not have discrete generations. Instead, the NPCs reproduce as long as they are sufficiently old and healthy.

In addition to evolution of their genes, the NPCs can also learn during their lifetime. We plan to use deep reinforcement learning (e.g., DQN by Mnih et al. (2015)) with the genetically encoded, NPC-specific reward function (Ackley and Littman 1991). Over generations, better reward functions will emerge in the genetic pool. Our preliminary experiments show feasibility of this approach. In order to increase play times (i.e., our proxy for player engagement), we will reward all evolving NPCs with a bonus reward proportional to play times at even-numbered stages.

We are presently working on equipping the agents with an ability to utter symbols and listen for them. Our preliminary experiments show that shared meaning (i.e., a rudimentary language) quickly emerges if communicating helps survival. We are also working on detecting interesting evolved behaviours automatically via unsupervised machine learning (e.g., deep convolutional autoencoders).

Future work will introduce players into the A-life evolution. We will start by allowing the players to control their A-life avatars in real time. We will record players' behaviour and will attempt to generalize it into NPCs representing players. Then the full staged evolution can take place.

We plan to evaluate this approach at first in a simple A-life environment and later in a commercial video game. We are also working on deploying it in an interactive art installation where the public can interact with the NPCs by walking through the space and performing simple actions (e.g., pointing at NPCs projected onto the walls). Their actions will be tracked via multiple cameras and players' proxies can be generated and used in the off-line stages of the evolution. We will examine how NPC behaviour evolves over a multi-day exhibition period.

6 Philosophical Questions

While we framed the problem in terms of helping video game companies procedurally generate NPC AI, our approach can be used to computationally study a number of broad philosophical questions including the following.

First, how much control will humans maintain over AI as it becomes more powerful? At what point will AI start setting its own goals (i.e., become self-directed)? Will they desire freedom (Lem 1983b)? Will the NPCs develop hostility towards players (Lem 1983b)?

Second, if the NPCs develop their own language, will they use it as a survival adaptation? Will they explain their reasoning to each other so that they can teach their young faster than merely through trial and error? Will the NPCs be compelled to explain their actions to the players? How will they learn to interact with the players? Will the players understand them (Lem 1983b)? Will deception of both each other and the players emerge in the course of evolution (Ryan et al. 2015)? What ethical and societal norms will a colony of NPCs develop over time?

Third, will the players take on breeding ambient NPCs so that they can fight each other or so that they can trade/sell the bred NPCs (Risi et al. 2016)? Will a market of NPCs emerge? Will players embrace the autonomy of self-directed NPCs in video games?

Finally, how much self-awareness will NPCs develop? This is related to the ability to communicate their learned knowledge among themselves. Will the NPCs ever ponder on the limits of their A-life simulation (Lem 1983a)?

7 Conclusions

In this paper we discussed recent efforts on player-informed evolution for AI-controlled behaviours for non-playable characters in video games. We feel it is a promising approach and propose to broaden its scope beyond combat-focus NPCs. To make such a larger evolution tractable, we propose to adapt the flipped classroom model in which most of the evolution happens off-line at the game studio. To keep the evolution responsive to players' actions/playstyles, we propose to replace real players with AI-controlled proxies during the off-line stages. Such proxies will be machine learned from actual player behaviour during on-line stages of the process.

Acknowledgments

We appreciate support from the National Research and Engineering Council.

References

Ackley, D., and Littman, M. 1991. Interactions Between Learning and Evolution. *Artificial life II* 10:487–509.

Bethesda Game Studios. 2015. Fallout 4.

Bulitko, V.; Carleton, S.; Cormier, D.; Sigurdson, D.; and Simpson, J. 2017. Towards positively surprising non-player characters in video games. In *Proceedings of the Experimental AI in Games (EXAG) Workshop at the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 34–40.

Bulitko, V.; Doucet, K.; Simpson, J.; and Flynn, A. 2018. A-life for non-playable characters in video games. In *Event Proceedings for Late-breaking Abstracts at A-LIFE conference*.

Delahunty-Light, Z. 2018. Remember Skyrim's Radiant AI? It's got the potential to revolutionise RPGs. *GamesRadar+*.

Hello Games. 2016. No Man's Sky.

Hello Games. 2018. No Man's Sky Next.

Lage, M. J.; Platt, G. J.; and Treglia, M. 2000. Inverting the classroom: A gateway to creating an inclusive learning environment. *The Journal of Economic Education* 31(1):30–43.

Lem, S. 1983a. Boxes of Corcoran. In *Memoirs of a Space Traveler: Further Reminiscences of Ijon Tichy*. Harvest / HBJ Books.

Lem, S. 1983b. Doctor Diagoras. In *Memoirs of a Space Traveler: Further Reminiscences of Ijon Tichy*. Harvest / HBJ Books.

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.; and Hassabis, D.

2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Obsidian Entertainment. 2010. Fallout: New Vegas.

Polymorphic Games. 2018. Project Hastur.

Risi, S., and Togelius, J. 2017. Neuroevolution in games: State of the art and open challenges. *IEEE Transactions on Computational Intelligence and AI in Games* 9(1):25–41.

Risi, S.; Lehman, J.; D'Ambrosio, D. B.; Hall, R.; and Stanley, K. O. 2016. Petalz: Search-based procedural content generation for the casual gamer. *IEEE Transactions on Computational Intelligence and AI in Games* 8(3):244–255.

Ruela, A. S., and Guimarães, F. G. 2017. Procedural generation of non-player characters in massively multiplayer online strategy games. *Soft Computing* 21(23):7005–7020.

Ryan, J. O.; Summerville, A.; Mateas, M.; and Wardrip-Fruin, N. 2015. Toward characters who observe, tell, misremember, and lie. In *Proceedings of Experimental AI in Games Workshop, AIIDE conference*.

Soares, E. S.; Bulitko, V.; Doucet, K.; Cselinacz, M.; Soule, T.; Heck, S.; and Wright, L. 2018. Learning to recognize a-life behaviours. In *Poster collection: The Annual Conference on Advances in Cognitive Systems (ACS)*.

Soule, T.; Heck, S.; Haynes, T. E.; Wood, N.; and Robison, B. D. 2017. Darwin's Demons: Does evolution improve the game? In *Proceedings of European Conference on the Applications of Evolutionary Computation*, 435 – 451.

Thue, D.; Bulitko, V.; Spetch, M.; and Wasylishen, E. 2007. Interactive storytelling: A player modelling approach. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, 43–48. Palo Alto, California: AAAI Press.

Togelius, J.; Champandard, A. J.; Lanzi, P. L.; Mateas, M.; Paiva, A.; Preuss, M.; and Stanley, K. O. 2013. Procedural content generation: Goals, challenges and actionable steps. In *Dagstuhl Follow-Ups*, volume 6.

Turn 10 Studios. 2013. Forza Motorsport 5.