Acting with Style: Towards Designer Centred Reinforcement Learning for the Videogames Industry

Batu Aytemiz*
University of California, Santa Cruz
Santa Cruz, California, USA
baytemiz@ucsc.edu

Mikhail Jacob
Microsoft Research Cambridge
Cambridge, UK
t-mijaco@microsoft.com

Sam Devlin
Microsoft Research Cambridge
Cambridge, UK
sam.devlin@microsoft.com

ABSTRACT

In recent years reinforcement learning (RL) techniques have been successful in solving complex problems, especially in video games. However, this rapid progress has not yet translated into mass adoption of RL techniques in the video games industry. We believe there isn't enough focus on being able to specify not only what goal our agents achieve, but also how they achieve it and also how reinforcement learning techniques fit into pre-existing workflows and constraints. We offer three suggested methods to alleviate these problems: Using preference learning to specify agent styles, using Potential-based Reward Shaping to make combining multiple sources of reward more robust and using an automated reward ratio scheduler to allow designers to work at a more meaningful abstraction level. Finally, we present a set of questions that we as a research community should answer to make reinforcement learning more approachable by the widest audience of potential RL users.

CCS CONCEPTS

• **Human-centered computing** → *User centered design.*

KEYWORDS

Reinforcement Learning, Human-Computer Interaction, Preference Learning

ACM Reference Format:
Batu Aytemiz, Mikhail Jacob, and Sam Devlin. 2021. Acting with Style: Towards Designer Centred Reinforcement Learning for the Videogames Industry . In *CHI 2021 Workshop on Reinforcement Learning for Humans, Computer, and Interaction (RL4HCI), May 08–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recent results have proven, given ample computation resources, that Reinforcement Learning (RL) based approaches can solve highly complicated problems [3]. Even though many of these results have been in the games domain [12], when we look to the games industry (and many other customer-facing industries for that matter), we see scant adoption of RL techniques. We think HCI research could be applied to understand why this is the case and improve RL adoption. For example, Jacob et al's 2020 interview-based study of game designers and developers around the challenges and opportunities for RL in commercial games revealed two key challenges — the lack of expressive control and inadequate designer workflows [8].

RL agents deployed in the real world need to respect a wide range of constraints on *how* they achieve their goals, whether based on safety, cultural expectation, fairness, justice–RL is still maturing and so researchers are currently focused on ensuring that RL agents can achieve their goals at the expense of most other considerations. The prevailing approach of writing an all-encompassing single scalar reward, also referred to as the reward hypothesis [13], to train the reinforcement learning agent makes it difficult to respond to all these different constraints. We claim that writing a single mathematically-specified reward is not a suitably efficient way to capture the rich, nuanced set of constraints that an RL agent needs to respect in real-world tasks while also fitting into the workflow of the widest audience of potential RL users. Novel techniques that enable system designers to be more expressive and guide autonomous agent behaviour are vital. We argue, however, that if we want reinforcement learning techniques to be more widely adopted then simply introducing new techniques isn't sufficient. We also need to investigate what types of practices are effective and how our novel techniques integrate into preexisting workflows in training the artificial agents.

The existing body of research on interactive reinforcement learning [2] have given us a strong foundation but unless reinforcement learning techniques are improved with a user-centered design focused on eventual RL users rather than RL researchers, it is unlikely we will see anything but results out of specialized research labs. The video game industry is a good example of this untapped potential where tooling and workflow integration for RL haven't kept up with the developments in research. This is preventing RL techniques from being widely adopted by those who create games, even though video games are a common testbed for RL research. We believe reinforcement learning can be utilized in video games for much more than benchmarking research.

Video games allow people to immerse themselves in whatever stories they so desire [1], from the exhilarating adventures of a wizard to the gut-wrenching experiences of a horror story protagonist. These stories are only as impactful as the fantasy of the world they manage to evoke in the player's mind [5]. The non-player characters that populate these worlds are crucial in developing the game's fantasy—from the companions to fight alongside, to the

*Work done while at Microsoft Research Cambridge.

Yokohama '21, May 08–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

monster that hunts the player, each computer-controlled agent has a very specific role to play in bringing these game worlds to life.

Having agents that learn by themselves how to behave in these varied stories is a long-lasting vision of videogame developers [15]. After all, wouldn't it be incredible if the agents we have in our games could surprise even their creators, instead of always behaving in structured ways following hand-crafted rules? Reinforcement learning, in theory, offers such a framework where agents can act in a given environment towards a provided goal and learn from their own experience. Unfortunately in practice, we are still far from realizing this vision. The tension between keeping the aforementioned "fantasy of the world" consistent, and the freedom of the agents to act as they see fit is one of the aspects that hold reinforcement learning back from game industry adoption.

To keep the fantasy of the game worlds alive it is essential for the AI agents to be controllable and act in accordance with the style of the world around it. When the player encounters any aspect that seems out of place from this fantasy—whether it be a misplaced prop, an unexpected piece of dialogue, or an AI agent acting erroneously— the illusion is broken [14]. Due to these constraints, videogame designers often lean towards more hand-crafted, rule-based techniques such as finite state machines [4] and behaviour trees [7]. Designers who are tasked with weaving the fantasy of the game world often must spend a significant amount of time tweaking the parameters of the AI systems just to ensure the agent acts as befitting the game's chosen style under different circumstances.

On the other end of the control – autonomy continuum there are end-to-end model-free reinforcement learning methods. These techniques can deliver truly innovative or unexpected behaviours that can surprise and delight players [9] but they care not for the fantasy of the game, nor the aesthetic constraints valued by the designer. This makes utilizing these agents in real video games rather difficult [8]. Current learning techniques don't lend themselves well to abiding by the stylistic constraints of the game world.

Merely discovering techniques that afford more control over the behavioural styles of our agents will not be sufficient to increase adoption of reinforcement learning techniques in creative industries. We also need to be cognisant of the people who would actually use these techniques, and how they would use them.

The designers who decide and tweak a game's aesthetic aspects are often separate from the engineers who implement the underlying AI behaviours. Requiring designers to become proficient enough in reinforcement learning to adjust the AI codebase to achieve the desired style is unrealistic. Instead, we need to better understand how designers work, and ensure the techniques we develop can fit into their workflows to empower them in realizing their vision.

Thus, we claim substantially more research is necessary towards improving how humans' control, guide and interface with reinforcement learning techniques before learning models can be used widely in creative industries. More specifically, we identify two areas that we claim need more attention:

- The ability to specify not only what goal our agents achieve, but also how they achieve it.
- The experience of training and iterating on reinforcement learning agents, and how it fits into pre-existing workflows and constraints.

In the following section we are going to describe our example contributions towards these outlined problems.

2 PREFERENCE LEARNING TO SPECIFY STYLE

We implemented a preference-based learning method (from [6]) to allow designers to specify their desired style through a simple interface. We used a simplified 2D navigation task to understand the previously raised problems better. Our proposed method works as follows:[1]

(1) The game agent designer, in consultation with an RL engineer designs the task for the agent to complete in a given part of the game and creates a set of high level capabilities and behaviours for the agent.
(2) The reinforcement learning engineer trains the agent to solve the given task using traditional reinforcement learning approaches.
(3) The designer is shown short clips of the agent acting to accomplish the goal.
(4) The designer then selects the clips where the agent exhibits more of the desired style, and doesn't take "fantasy-breaking" actions.
(5) The preferences of the designers are then fed back into the agent as a "style-reward" for it to optimize.
(6) After the designer is happy with the behaviour of the agent, they can move on to the next part of their game creation process.

This approach has several important benefits: First, it is much easier to recognize a desired style compared to describing it in mathematical terms. Furthermore, picking a desired behavior requires no coding experience. This gives more agency to the designers as compared to both existing practice [8] and the traditional RL creation pipeline. This allows them to be more expressive when creating agents in the game world.

Second, this approach decouples the process of training the agent to accomplish the task, from training the agent to act in the desired style. This decoupling results in a workflow better fit for multidisciplinary teams: The engineers can do the brunt of the reinforcement learning training, and then the designers can focus only on shaping style of the agent.

Third, this method drastically reduces the computation necessary to iterate on different styles. Traditionally, in order to make a change to a reinforcement learning agent–however small—the whole training process needs to be restarted from scratch. This in practice results in the need for a sizeable time and computation budget and limits the number of iterations that can be done by the time in a fixed schedule. In our method, because the designer is not starting from scratch, but rather iterating on an already trained agent, the computation and time requirements are much less, resulting in higher iteration count.

Preference learning, of course, is not the only way to allow designers to express themselves—on the contrary each preference only contains one bit of information which is not a lot at all. What

[1]For a more detailed breakdown of our system refer to https://www.microsoft.com/en-us/research/blog/designer-centered-reinforcement-learning/

other techniques can we imagine that are even richer in information, and even better support designer creativity and expression?

- Can we use rankings, instead of binary preferences to allow designers to specify the style in further detail?
- Can we use designer demonstrations to allows designers to directly act out the desired style?
- Can we utilize user data and isolate the stylistic choices our users have?
- Can we extract style information from other forms of input such as videos or key framed animations?

3 WORKFLOW IMPROVEMENTS

Even after specifying a reward that captures the desired style, referred to as style reward, it is not trivial to integrate it into the agent's preexisting task reward. If the ratio of the style reward to the task reward is too high, style reward overwhelms the task reward and the original task performance suffers. However, if the ratio is too low, then there is no observable behavior change. The default approach to solving this problem is to iterate—tweak the reward function slightly and run another experiment. This workflow is not efficient as "it's unwieldy and it takes a lot of time" [8], sometimes taking up to weeks of distributed training. This makes retraining with incremental changes to a reward function very expensive and finding an effective way to combine style reward with tasks rewards highly important.

The workflow of tweaking style reward and the task reward ratio to achieve the right balance is made further difficult as there is no inherent meaning to these values. What does it mean to have the style reward be two times the task reward? Ideally, we like the designers to be able to work with tuning knobs that are more meaningful from a design perspective, instead of fiddling around with arbitrary and hard-to-reason hyperparameters.

In theory, nothing is preventing the designers from running thousands of experiments to find the perfect balance of task reward to style reward. In practice fixed deadlines, computational budgets, and waning motivation makes this naïve search a daunting task.

We contribute two example interventions to make introducing style to an already trained agent easier:

(1) Using Potential-based Reward Shaping [10] to make the original task performance more robust to the introduction of a new style.
(2) Using an automated scheduler to balance the style to task reward ratio in order to abide by the designer provided constraints.

Potential-based reward shaping (PBRS) is a technique that is initially used to ensure the addition of shaping rewards do not negatively influence the overall task performance. Shaping rewards are additional rewards that are separate from the main task reward. They help guide the agent into accomplishing the main task. They have the potential, however, to degrade the maximum original task performance. For a navigation task where the goal is to reach the goal, a shaping reward can be given whenever the agent gets closer to the goals. We repurposed PBRS to make it easier to introduce the style reward into the preexisting task reward. While using PBRS doesn't change resulting agent behavior, it makes reaching that behavior a whole lot easier.

With the automated scheduler the designer specifies a minimum acceptable task threshold. The scheduler then tries to maximize the style reward while also keeping the task performance above the specified threshold. The designer can interface with the system by specifying nn acceptable reward thresholds and letting the system optimize the style to task ratios, instead of directly manipulating the style to task reward ratio to hit the desired reward threshold.

This method allows the designer to work in an abstraction level that is more meaningful from a design perspective. Consider an example where the designer is trying to shape a car racing agent to drive in a specific, reckless style (which is suboptimal when it comes to finishing the race as fast as possible) while also ensuring the agent finishes the race in an allocated time. Previously the designer would have to find the correct style (reckless driving) to task (finishing the level fast) reward ratio to fulfill these constraints—if the agent was too slow, the task reward weight would have to be increased, if the driver didn't seem reckless enough, the style reward weight would have to be increased, iteratively, and over multiple experiments. With our system we allow the designer to specify a minimum task reward which makes it easier to implement a design idea such as "finish the level in 30 seconds while maximizing the reckless driving style" and the automatic scheduler ensures this is the case. We claim the latter is a more effective workflow for designers.

The pain points identified and addressed here are two among many. While there is some previous work [11], not enough focus is given to investigating the reality of how reinforcement learning is used by its users. Human-computer interaction research methodologies and user-centered perspectives are essential for us to make reinforcement learning a more approachable and impactful technique. We still need further insights into:

- What are the most commonly faced issues when training reinforcement learning agents?
- What are the correct abstraction levels to structure reinforcement learning problems?
- What are the assumptions novice users make about reinforcement learning that hamper initial progress?
- How do we ensure that the benefits of RL distributed across a more diverse range of stakeholders?
- How do we evaluate RL agents with an emphasis on real world performance and constraints over training graphs?

4 CONCLUSION

With each new paper reinforcement learning techniques are getting more and more capable of solving increasingly complex tasks. Unfortunately, however, the conversation around how we harness this increasing capability isn't keeping up with the progress in these developments. This situation speaks to the urgent need for HCI research and methodologies to understand who ultimately will use RL and how RL can be built for them. In this paper we described two obstacles around adoption of reinforcement learning techniques (insufficient techniques for designer expression and inefficient workflows) and described our example solutions. It is our hope that we can expand the conversation around how reinforcement learning agents are trained and ensure the potential of reinforcement learning doesn't stay locked away in specialized research labs.

REFERENCES

[1] Anna Anthropy and Naomi Clark. 2014. *A game design vocabulary: Exploring the foundational principles behind good game design.* Pearson Education.

[2] Christian Arzate Cruz and Takeo Igarashi. 2020. A Survey on Interactive Reinforcement Learning: Design Principles and Open Challenges. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, New York, NY, USA, 1195–1209.

[3] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, and Charles Blundell. 2020. Agent57: Outperforming the Atari Human Benchmark. (March 2020). arXiv:2003.13350 [cs.LG]

[4] Daniel Brand and Pitro Zafiropulo. 1983. On Communicating Finite-State Machines. *J. ACM* 30, 2 (April 1983), 323–342. https://doi.org/10.1145/322374.322380

[5] Douglas Brown and William. 2012. *The suspension of disbelief in videogames.* Ph.D. Dissertation. Brunel University School of Arts PhD Theses. https://bura.brunel.ac.uk/handle/2438/7457

[6] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems 30*, I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett (Eds.). Curran Associates, Inc., 4299–4307.

[7] Michele Colledanchise and Petter Ögren. 2018. *Behavior Trees in Robotics and AI: An Introduction.* CRC Press. https://play.google.com/store/books/details?id=YVOWDwAAQBAJ

[8] Mikhail Jacob, Sam Devlin, and Katja Hofmann. 2020. "It's Unwieldy and It Takes a Lot of Time"—Challenges and Opportunities for Creating Agents in Commercial Games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 16. 88–94.

[9] Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter J Bentley, Samuel Bernard, Guillaume Beslon, David M Bryson, Patryk Chrabaszcz, Nick Cheney, Antoine Cully, Stephane Doncieux, Fred C Dyer, Kai Olav Ellefsen, Robert Feldt, Stephan Fischer, Stephanie Forrest, Antoine Frénoy, Christian Gagné, Leni Le Goff, Laura M Grabowski, Babak Hodjat, Frank Hutter, Laurent Keller, Carole Knibbe, Peter Krcah, Richard E Lenski, Hod Lipson, Robert MacCurdy, Carlos Maestre, Risto Miikkulainen, Sara Mitri, David E Moriarty, Jean-Baptiste Mouret, Anh Nguyen, Charles Ofria, Marc Parizeau, David Parsons, Robert T Pennock, William F Punch, Thomas S Ray, Marc Schoenauer, Eric Shulte, Karl Sims, Kenneth O Stanley, François Taddei, Danesh Tarapore, Simon Thibault, Westley Weimer, Richard Watson, and Jason Yosinski. 2018. The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities. (March 2018). arXiv:1803.03453 [cs.NE]

[10] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, Vol. 99. 278–287.

[11] Ariel Rosenfeld, Moshe Cohen, Matthew E Taylor, and Sarit Kraus. 2018. Leveraging human knowledge in tabular reinforcement learning: a study of human subjects. *The Knowledge Engineering Review* 33 (2018).

[12] Kun Shao, Zhentao Tang, Yuanheng Zhu, Nannan Li, and Dongbin Zhao. 2019. A Survey of Deep Reinforcement Learning in Video Games. (Dec. 2019). arXiv:1912.10944 [cs.MA]

[13] Richard S Sutton. 1998. Reinforcement learning: Past, present and future. In *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 195–197.

[14] Kimberly Voll. 2016. Less is More: Designing Awesome AI for Games. https://www.youtube.com/watch?v=1xWg54mdQos

[15] Alan Zucconi. 2020. The AI of "Creatures". https://www.youtube.com/watch?v=Y-6DzI-krUQ