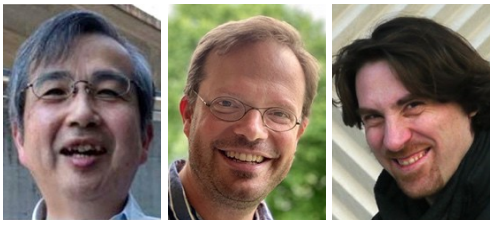
Interaction With a Memory Landscape

Tatsuo Unemi
Soka University, Hachioji, Japan
unemi@soka.ac.jp

Philippe Kocher
Zurich University of the Arts, Switzerland
philippe.kocher@zhdk.ch

Daniel Bisig
Zurich University of the Arts, Switzerland
daniel.bisig@zhdk.ch



Abstract

The touchscreen-based application *Greedy Agents and Interfering Humans* addresses the coexistence of humans and an AI system by providing an interactive environment where users can interact with a learning agent. The application is based on the well-known paradigm of reinforcement learning, a framework to model learning mechanisms based on the modification of behaviour through experience.

Instead of being a black box, the learning process is rendered perceivable for the user. The learning agent's memory is interpreted as a vector field, which is visualised by a particle flow. And the learning process is exposed to interaction to make it even more palpable. The user can not only observe the advancement of the learning process but also actively disturb it by placing obstacles in the environment or directly modifying the agent's memory. In addition, whenever the user touches the screen, a sonification of the simulation's state can be heard through headphones.

1. Introduction

This paper describes a touchscreen-based application entitled *Greedy Agents and Interfering Humans* in which a user can interact with an agent that learns to navigate through an environment. By means of visualisation and sonification, the agent's learning process is made perceivable. This application is the downscaled version of a tabletop installation that the authors developed earlier, in which up to three visitors, whose hands were tracked by a distance camera, could interact at the same time [1].

Our application employs reinforcement learning, a well-known paradigm with a long-standing history. Reinforcement learning provides a framework to explain how an agent's behaviour is changed through experience.

It is essentially based on a trial-and-error approach in which behaviour that leads to a successful outcome is rewarded and thus reinforced. Our application makes an attempt to render this underlying algorithm perceptible and interpretable through interaction and thus tries to open the black box.

Furthermore, as it turns the algorithm into an aesthetic expression by means of visualisation and sonification, our application also offers the opportunity to engage with contemporary computational art. And finally, it exemplifies the creative coexistence between humans and an AI system. In this sense, it aligns with the authors' earlier works dealing with the coexistence of human and non-human actors [2–5].

2. Reinforcement Learning

The process on which the application *Greedy Agents and Interfering Humans* is based is reinforcement learning. This learning paradigm models how an agent learns to make decisions by interacting with an environment. It is inspired by behavioural psychology and focuses on training intelligent agents to take action in a way that maximises a cumulative reward. Reinforcement Learning is widely used in applications like robotics, game playing, autonomous vehicles, recommendation systems, and more.

This learning paradigm has been researched for more than a hundred years in the fields of psychology and ethology. In the late 19th century, Thorndike [6] initiated scientific research into these phenomena in the context of social psychology. Several decades later, Skinner [7] conducted systematic experiments on pigeons and rats following a Behaviorism approach. In these experiments, the animals changed their behaviour to increase positive experiences (being fed) and avoid negative experiences (electric shocks). In the 1980s, with the advent of powerful computational resources, it became feasible to adopt principles of reinforcement learning in the context of machine learning. Since the early 1990s, substantial research on computational forms of reinforcement learning has been conducted by Sutton and Barto [8].

The key components of reinforcement learning are an agent, an environment with which the agent interacts, a set of possible actions and a reward in the form of a numerical value that indicates how good or bad each chosen action was. It is the agent's goal to maximise the cumulative reward over time. In order to learn effectively, the agent must be able to build up a kind of memory. It must be able to associate past actions with the rewards it has received to select future actions accordingly.

The memory keeps track of the experience made at every point in the environment. It is thus closely related to the environment’s topology and can consequently be described as a memory landscape.

3. Implementation

3.1 Simulation

The simulation on which the application *Greedy Agents and Interfering Humans* is based is an implementation of a Q-learning algorithm [9]. The environment in which the agent moves about consists of a small two-dimensional grid world of 6 x 11 cells (fig. 1). Each cell represents either an empty space or an obstacle. On

each cell, the agent can choose among four discrete actions (up, down, left, right) to move to an adjacent cell, provided it is not an obstacle. The agent has a simple navigation task: find the shortest path from a start to a goal location, i.e., reach the goal location with a minimum number of actions. In the beginning, the agent does not know anything about the environment and moves randomly from cell to cell. As the learning progresses, the probability that the agent chooses a random action (exploration) instead of an optimal action (exploitation) gradually decreases [10].

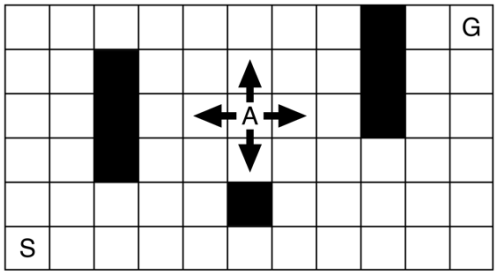


Fig. 1. *The 6 x 11 simulation grid. S: start, G: goal, A: agent, black squares: obstacles.*

When the learning simulation starts, two cells at opposite sides of the grid world are defined as start and goal positions. There are no obstacles at this stage, as these are only defined later by the user's interaction. Each learning episode starts by placing the agent on the start position and ends when the agent has either managed to reach the goal position or exceeded a maximum number of actions. Upon reaching the goal location, the agent receives a reward depending on the efficiency of its search. Then it is put back to the start position, and the search begins again. The learning process continues until the number of simulation steps or the number of times the goal was reached exceeds a predefined value. In that case, the memory is reset and a new simulation cycle is started.

During a learning simulation, the agent memorises the value of an action at each cell, i.e., how fruitful it is to continue in a specific direction to obtain the highest possible reward. The higher the value of an action, the more likely the agent will take this action. When the agent eventually reaches the goal, the value of the last action taken is propagated backwards from the goal to previous positions along the agent's path. In order to accelerate this propagation, a replay mechanism is employed that causes the agent to randomly recall previous navigation steps from a memory pool. This mechanism is similar to the Dyna architecture proposed by Sutton [11].

3.2 Visualisation

The simulation's state is visualised on the screen. The agent's position is shown as a white circle. The user can observe how the agent searches for the goal position and how this endeavour becomes increasingly effective. However, to make this learning progress perceivable in real-time, the learning episodes have to be executed and iterated at a much faster rate in the background. The agent's movements displayed on screen represent one single learning episode taken at a certain point in time and slowed down.

The agent’s memory is interpreted as a vector field in which each vector represents the preferred direction of movement for the corresponding cell. The direction and length of each vector are calculated as the sum of the four orthogonal directions of the discrete actions the agent can perform, scaled by

their respective value. A particle flow animation consisting of some hundred thousand short line segments moving across the screen visualises this vector field. The movements of these line segments result from the forces exerted on them by the vector field. Each line segment is drawn in a colour that changes according to its moving speed. The agent's learning process and build-up of memory become perceivable as the visualisation changes its appearance during a simulation cycle. While the particle flow is not yet pronounced at an early stage of learning (fig. 2), it becomes more clearly directed towards the goal as the simulation advances (fig. 3).

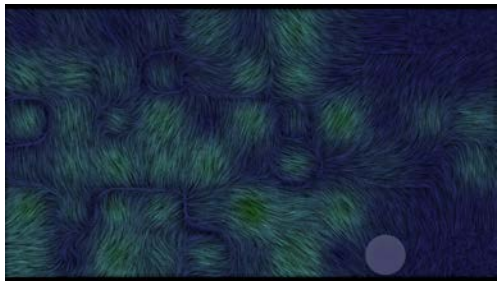


Fig. 2. *The visualisation of the simulation at an early stage of learning.*

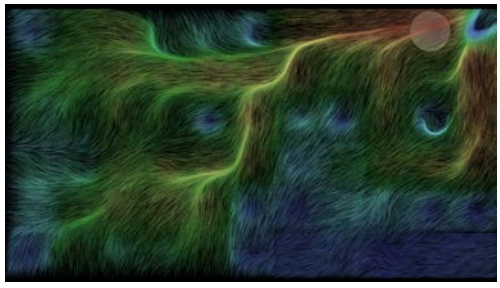


Fig. 3. *The visualisation of the simulation at a later stage of learning.*

3.3 Sonification

The sound played back via headphones also reflects the state of the agent's memory and the dynamics of the simulation. Unlike the visualisation, the sound is only activated when the user touches the screen. The user's finger becomes a stethoscope, as it were, with which the memory landscape can be acoustically examined. Since every touch simultaneously leads to an interaction with the simulation, it is impossible to listen to the sonification without altering it.

Each time the agent moves to a different cell, a short, high-pitched sound is audible. The regular repetition of this sound reflects the rendering steps and, thus, the discretisation of time, which is constitutive for the simulation. It also indicates whether the agent is moving at all and not currently stuck in a dead end of obstacles erected by the user.

The agent's memory is turned into sound by using the values of the actions for each cell as sound synthesis parameters: the higher the values, the brighter and denser the sound. The sound design uses a granular synthesis approach, thus creating a strong link to the particle-based aesthetics of the visualisation. Above all, the notion of the movement of particles is doubled in the sound. In respect to the user's viewpoint, a flow of particles in a horizontal direction is matched by a movement of the sound in the stereo field; a flow in a vertical direction by a continuous movement in pitch.

The position of the touch determines the cells whose values are made audible. To enable the perception of a continuous sound, the values of the cells next to the touch position are interpolated. The sound of a cell containing an obstacle is

strongly attenuated and superimposed by a low-frequency sound.

3.4 Interaction

The interaction is based on detecting a visitor's touch on the screen. Up to two touch events can be detected simultaneously. The positions of the touch events are mapped into the grid world of the simulation.

Two mutually exclusive forms of interaction exist. The first form of interaction becomes active when the touch remains stationary. If the touch persists for longer than a predefined amount of time, an obstacle is added to the simulation at the location of the touch. If the touch is subsequently removed, the obstacle continues to exist for the same amount of time as the touch has previously been applied. Since the agent cannot traverse this obstacle, it has to learn how to navigate around the obstacle to reach the goal position. As a result, the agent's memory is indirectly modified through interaction.

The second form of interaction becomes active if the touch changes position. As a result of this interaction, the agent will increase its preference for those movement directions aligned with the touch's travel direction and decrease its preference for the other movement directions. Accordingly, the agent's memory landscape is directly modified through interaction.

4. Conclusion

The application *Greedy Agents and Interfering Humans* is an attempt to make a machine-learning algorithm accessible to the user not only by rendering its principles visible and audible but also by opening them up for human interaction. In a time in which technology, notably AI, enters people's daily lives more and more, it becomes increasingly necessary to find ways to make its principles explainable and thus open up the black boxes. It is our conviction that computational art can be a valuable way to provide this kind of understanding.

Our objective was to create an artistic realisation of reinforcement learning rather than further develop it in an engineering context. From an artistic viewpoint, we believe that creativity is always a joint effort of human and non-human actors connected in a network. Our application deals explicitly with this concept of collaboration, albeit not in the only conceivable way. A wide and promising field for future work still remains open.

References

[1] Unemi, T., Kocher P., and Bisig, D. 2023. “Greedy Agents and Interfering Humans”. In *Proceedings of the Conference on Computation, Communication, Aesthetics & X,* Weimar, 363–367.

[2] Bisig, D. and Kocher, P. 2015. “DRIFT – Virtual Sand in Augmented Space”. In *Proceedings of the 18th Generative Art Conference,* Venice, 51–64.

[3] Bisig, D. and Unemi, T. 2009. “Swarms on Stage – Swarm Simulations for Dance Performance”. In *Proceedings of the 12th Generative Art Conference,* Milano, 105–114.

[4] Bisig, D. and Unemi, T. 2010. “Cycles: Blending Natural and Artificial properties in a generative artwork”. In *Proceedings of the 13th Generative Art Conference,* Milano, 140–154.

[5] Bisig, D. and Unemi, T. 2011. “From Shared Presence to Hybrid Identity”. In *Proceedings of the Consciousness Reframed Conference,* Lisbon, 48–53.

[6] Thorndike, E. L. 1898, 1911. “Animal Intelligence: an Experimental Study of the Associative Processes”. In *The Psychological Review: Monographs Supplements,* 2(4), i–109.

[7] Skinner, B. F. 1953. *Science and Human Behavior*, New York: MacMillan.

[8] Sutton, R. S. and Barto, A. G. 1998, 2/2018. *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press.

[9] Watkins, C. J. C. H. 1989. *Learning from Delayed Rewards*, Ph.D. thesis. University of Cambridge.

[10] Unemi, T., Nagayoshi, M., Hirayama, N., Nade, T., Yano, K., and Masujima, Y. 1994. “Evolutionary Differentiation of Learning Abilities: a Case Study on Optimizing Parameter Values in Q-Learning by a Genetic Algorithm“. In *Artificial Life IV.* MIT Press, 331–336.

[11] Sutton, R. S. 1990. “Integrated architectures for learning, planning, and reacting based on approximating dynamic programming”. In *7th International Conference on Machine Learning*, 216–224.