Boats and Time

What can reinforcement learning tell us about the present moment?

Machine learning project final report by Sean Brantley

Abstract

One enduring question about reality concerns the qualitative nature of time. Thinkers throughout history have explored the relationship between the external flow of physical reality and human experience of past, present, and future. In this report, I present a modern experimental investigation of a proxy for the present moment—the provenance state of a video game—using a virtual task environment and reinforcement learning techniques. I describe how I adapted a previously developed Bomberman-like video game to incorporate machine learning functionality and detail the iterative process of developing the agent-environment pair. This process included selecting a learning model, refining the reward function, and configuring hyperparameters to guide an autonomous virtual agent in completing a waypoint-finding task within a T-maze setup. My results suggest that the agent was consistently able to learn how to collect the reward from its starting block, a spacetime point at (0,0,0), narrowing its actions within the possibility space to achieve goal-directed behavior over time. I also outline future design paths, such as increasing task complexity, and address stakeholder perspectives on the potential uses and misuses of the data analysis scheme employed. Ultimately, I argue that this work demonstrates the existence of a set of possible outcomes emerging from a single present moment, bridging philosophical inquiry with contemporary experimental methodology.

Background

Space, time, possibility, and value are, from my perspective, the foundational aspects of human experience—concepts that are inextricably linked and converge at the ever-fleeting present moment. The idea that we navigate our surroundings by processing a constant flow of information from the external world, selecting a singular "best path" as the future transitions into the past, is not new. Immanuel Kant famously argued that space and time are essential frameworks for thought (Kumar, 2023), while also emphasizing how we ought to behave—what we should value—within a social world (Rueter, 2023). But what then is this point of convergence? What is the *present moment*?

Modern inquiry has largely turned to physics to explain this concept, given its ability to answer questions about the physical world and produce technologies like GPS, superconductors, and nuclear energy as a result. Yet, as scientists have made great strides, they have also grown hesitant to acknowledge that all experimental observations are mediated by human perception. As Kant proposed, we can never experience the "thing-in-itself"—the objective reality independent of our senses and cognition (Kumar, 2023). If this is true of objects, why should space or time be any different? I propose that our inability to resolve foundational questions in quantum mechanics,

such as the measurement problem, stems in part from a failure to account for the distortion that our subjective lens imposes on objective reality. By revisiting and updating philosophical notions of the relationship between spacetime and human experience, I aim to integrate an empirically grounded cognitive science perspective into the conversation.

For this reason, in past work, I have consistently turned to the incorporation of virtual environments (VEs) as a way of improving psychological investigations (Brantley, Wilkinson, & Feng, 2021; Wilkinson, Brantley, & Feng, 2021; Cecchini, Brantley, & Dubljević, 2023). These environments provide a controlled yet dynamic platform, facilitating near-perfect replicability of a starting state for complex tasks while offering a myriad of tunable variables. In some contexts, a specific type of VE, real-time strategy games, has even been touted as a standard task environment for cognitive science, comparable to the role of *Drosophila melanogaster* in genetics and biology (Thompson, Blair, Chen, & Henrey, 2013). Altogether, VEs offer modern scientists experimental opportunities that were unimaginable to our predecessors.

For this project, I aim to expand on this idea by integrating machine learning (ML), with a focus on what reinforcement learning (RL) can reveal. RL, a subset of ML, has emerged as a powerful tool for training autonomous agents in virtual environments. Inspired by behavioral psychology, RL involves tasks that challenge agents to navigate these environments by interacting with them, receiving feedback in the form of rewards or penalties, and iteratively refining their strategies to maximize cumulative rewards.

From a single starting state, an RL agent, like any other entity, traverses a path through an enormous possibility space, exploring configurations of spacetime shaped by its pursuit of specific goals. The trajectories produced by the agent—akin to the notion of a path function that describes the evolution of particles or systems over time in traditional chemistry and physics (A, 2023)—reflect not only its attempts to achieve a defined objective but also the iterative process of learning, where each step informs the next. This refinement of behavior in the face of uncertainty gives rise to optimized spacetime patterns, illustrating not only how agents adapt and improve over time but also that the space for such adaptation inherently exists. By studying these patterns, we can gain practical insights into decision-making processes while also speculating on the abstract concept of a singular provenance point, the proxy for a moment. Examining the spacetime trajectories of the agent highlights that entities experience only a subset of the vast possibility space embedded within the environment. However, with the computational power enabled by high-level RL, it may become possible to acquire enough data points to infer characteristics of the entire set.

Project Preparation (Data Analysis / Augmentation Replacement)

Unlike a typical ML project, where the dataset for training is prepared beforehand, RL generates its data dynamically as the agent explores its environment. In this context, data analysis corresponds to the state of the virtual world, while data augmentation involves tuning and refining that world. My exploratory data analysis process revolved around developing a functional virtual task to test the agent, *Boats*, by repurposing the Mazerunner game I had created prior to this course.

Before I could begin modifying the agent-environment pair, several preparatory steps were required. First, I updated my Unity version to meet the compatibility requirements for the latest Unity ML-Agents toolkit. I also set up a Python environment configured with the correct Python version and essential ML libraries, such as PyTorch, to enable communication between Unity and the training scripts.

Once I had a compatible version of my game, I prepared the player avatar to interact autonomously with the environment. This involved attaching key ML components to the avatar (Figure 1), including a decision requester, a Ray Perception Sensor 3D, and behavior parameters scripts. These components collectively enable the agent to make decisions on a consistent time scale, perceive tagged elements in its surroundings, and execute actions based on its understanding of the virtual space. Additionally, I developed a Player Agent script — an extension of Unity's Agent class — that integrated with the preexisting game framework and allowed for customized behavior specific to my project.

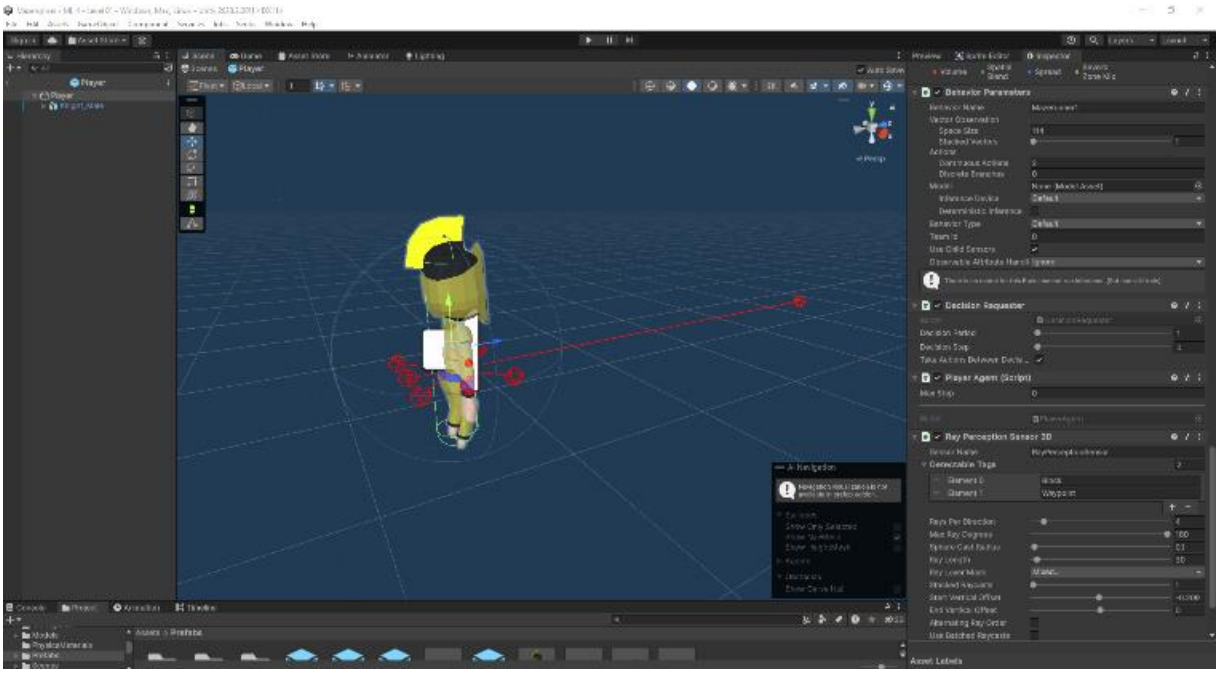

Figure 1: Screenshot of Player prefab with ML scripts visible in the inspector in the right-most column of the image. The avatar model and ray perception sensors can be seen in the middle section of the image.

Unlike a human with a top-down view of the maze, Boats perceives its surroundings through a Ray Perception Sensor, which simulates a limited field of view by detecting tagged objects like the blocks and waypoints within a specified radius and angle. For example, the sensor is configured with a ray length of 70 units and a 180-degree detection field, allowing the agent to “see” only immediate obstacles and goals. Boats lacks memory or abstract reasoning, so each decision is based solely on its current perception and learned policy. However, with each fixed update Boats pings out into its surroundings to mimic a first-person view of the environment.

At the end of this foundation laying stage, the project became very iterative. I cycled through figuring out what the agent was capable of, adjusting the environment (Figure 2), and tweaking the reward

function. While this section focuses on the agent-environment pair preparation, the reward function's development occurred concurrently and will be detailed later.

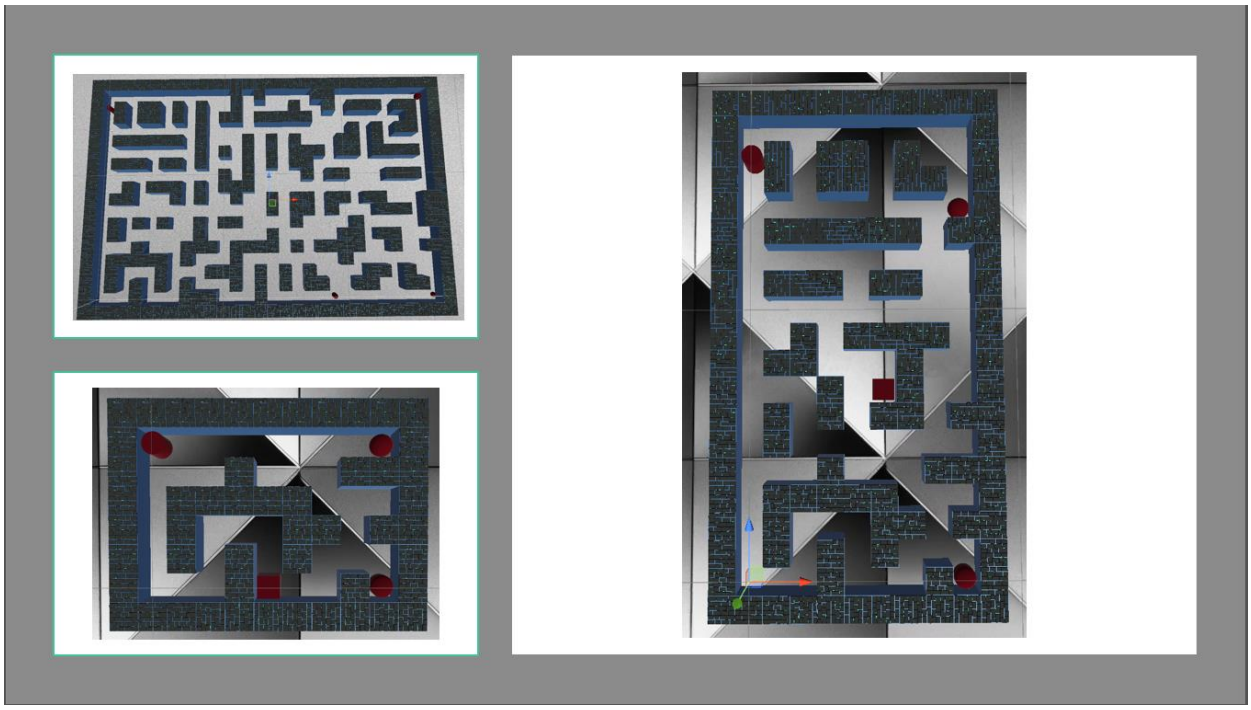

Figure 2: The top left image shows the first iteration, which removed the bomb, enemies, and destructible objects, replacing them with waypoints. The right and bottom left images show subsequent iterations, culminating in a reduced maze size with the T-m

In the first integration attempt, I discovered that combining the bomb placement mechanic and player spawning functionality with the ML decision-making process caused an infinite loop, crashing the game. The agent's high-speed training, near-instantaneous decision to drop a bomb, inability to escape the bomb's radius, and repeated spawning in the same location created an unmanageable feedback loop. To address this, I shifted from an elimination task to a location-based task.

In the second iteration, I limited Boats' actions to movement on the X and Z planes and removed enemies and bomb functionality entirely, scattering four waypoints across the maze. However, the maze was so large that it became difficult to discern patterns in the agent's failures, as no two runs were similar. This prompted two successive reductions in maze size. At the smallest size (bottom left of Figure 2), a consistent failure pattern emerged: the agent would often get stuck in the alcove of the waypoints surrounded by three blocks, making no further progress during any sequential episodes and repeatedly returning to the same spot at an increasingly rapid pace.

To address this issue, I experimented with several reward structures, but none successfully resolved the problem. After submitting the Part-Three check-in, I discussed the issue with instructor Embry, who advised simplifying the project. His guidance, "just show that your model could learn something," led me to focus on consistently collecting a single waypoint. This approach also reminded me of T-maze experiments, a staple of behavioral biology research I had encountered during my undergraduate studies, so I modeled the task in that manner.

This refinement process is conceptually similar to exploratory data analysis practices commonly employed prior to traditional machine learning projects. I iteratively refined the “dataset”—the functional player and level pair—to produce a scenario in which my model, combined with the reward function, could achieve measurable learning outcomes, akin to how one of my peers’ models might learn to classify an image.

Model Selection

The model I used to teach Boats how to complete the task was the Proximal Policy Optimization (PPO) algorithm. Admittedly, this selection was made based on a suggestion from ChatGPT early in the project. As someone with no prior experience in machine learning or reinforcement learning, I initially did not fully understand that PPO was the underlying mathematical model. This realization is honestly only fully taking hold as I write this final report.

PPO is a reinforcement learning algorithm designed to balance exploration—trying new paths and strategies—and exploitation—optimizing known successful behaviors. In the context of my project, this meant the agent, Boats, alternated between testing new routes through the maze and refining its movements toward the waypoint as it learned which actions yielded the highest rewards. Another key component of PPO is the clipped surrogate objective, which ensures that policy updates remain stable and avoid drastic changes. This stability was crucial in preventing Boats from overcorrecting its behavior after each episode, enabling the agent to steadily improve its performance without erratic or counterproductive shifts in strategy.

During the meeting with instructor Embry, mentioned previously, one of the main pieces of feedback I received was to consider experimenting with alternative learning models. However, given that the T-maze and radar system were only finalized two days before the video portion of the project was due, I did not have time to investigate or implement a different model. I leave that exploration as a direction for future work.

Reward Function and YAML File (Training Methodology Replacement)

As previously discussed, the creation of the reward function was an iterative process. Over roughly 150 individual training runs, the reward structure was adjusted to encourage or discourage specific patterns of navigation. Initially, the goal was to collect all waypoints as quickly as possible, but by the end, the objective shifted to consistently navigating to a single waypoint.

Rather than listing every adjustment made, I want to focus on one key finding: simpler reward structures performed better. At the start, the reward function included only a small positive reward for moving, a small negative penalty for running into walls, and a large positive reward for finding waypoints. Surprisingly, this basic setup almost succeeded in solving the first large maze, with the agent finding three of the four waypoints in one run. However, as the maze size was reduced and failure patterns emerged, I attempted increasingly complex reward structures to encourage additional behaviors. Unfortunately, each added complexity introduced new local minima, limiting the agent's ability to explore effectively.

By the end of the development process, I simplified the reward function significantly, ending with a structure close to the original. Two key additions were made:

1. A list of 1x1 unit squares in the virtual space, representing normalized grid locations used to track areas the agent had discovered.
2. A periodic calculation of the distance between the avatar and the waypoint.

In the final version, the avatar received:

- **0.1 points** for moving,
- **30 points** for discovering a new location, and
- **1000 points** for reaching the waypoint.
  Negative rewards included:
- **-0.2 points** for colliding with a wall, and
- A penalty equal to the distance (in units) for every second spent more than four units away from the waypoint transform (Figure 3).

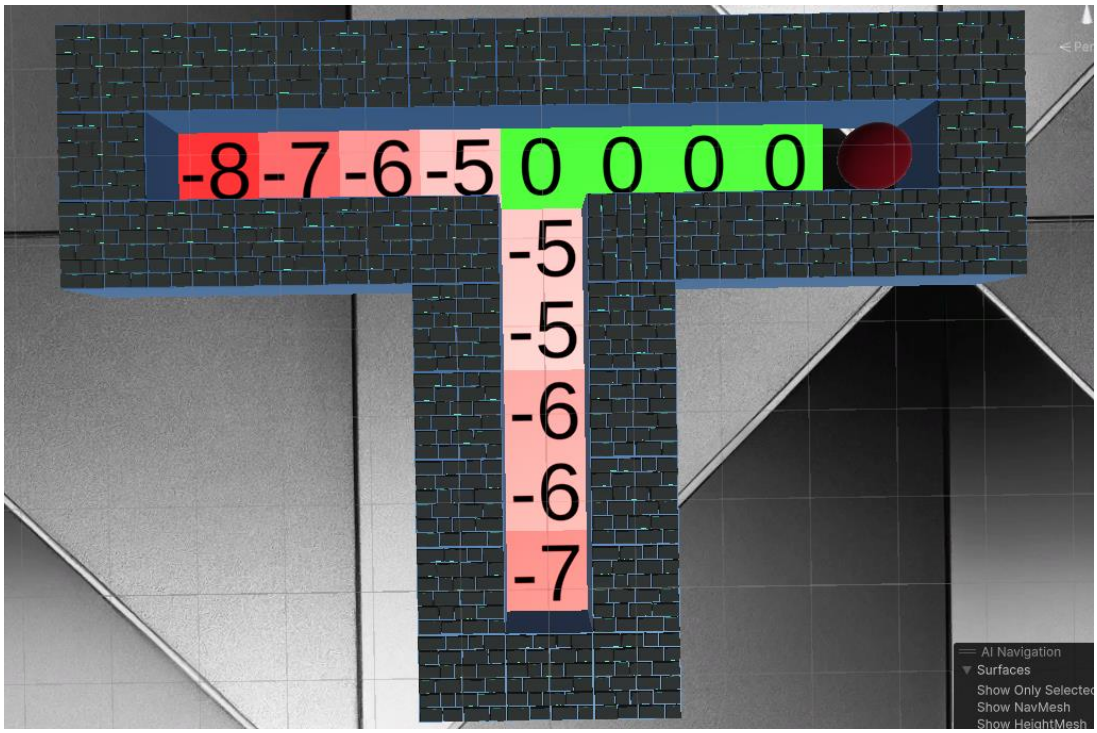

Figure 2: Image of final T-maze, with color coded and numerically labeled tiles, indicating deduction penalties.

This arc—from simplicity to complexity and back—also extended to experimentation with hyperparameters in the YAML configuration file. Like the model selection process, this aspect of the project felt abstract, likely due to my limited experience with machine learning. I relied on ChatGPT for guidance as I adjusted parameters such as the learning rate, batch and buffer sizes, beta, and checkpoint intervals. Among these, the most impactful change was to the beta parameter. Initially, a low beta value caused premature convergence, restricting the agent's ability to explore effectively. Increasing beta encouraged broader exploration, allowing the agent to escape the left arm of the maze by increasing the likelihood of moving into unfamiliar areas.

In reinforcement learning, traditional concepts of underfitting and overfitting are less prominent than in supervised learning, as the agent dynamically generates its own data through interactions with the environment. To mitigate potential overfitting to specific environment configurations, I iteratively refined the environment design, such as reducing maze complexity. Progress was monitored using cumulative reward trends displayed in the command line and qualitative observations of agent behavior during training episodes. While no explicit validation set was used, consistent performance across multiple episodes and the emergence of the intended waypoint-collection behavior served as indicators of the agent's ability to generalize within the defined environment.

Results

To evaluate the project's performance, I conducted an experiment on Boats, recording the last ten trials and reporting the findings regardless of success (Table 1). The parameters were straightforward: Boats had a maximum of 500,000 steps to locate the waypoint in five consecutive episodes. Each episode ended either when Boats found the waypoint or when the three-minute game timer elapsed. For clarity, an episode is defined as a single attempt by Boats to navigate the maze, while the maximum step count represents the total steps allocated for all episodes within a trial.

Run #	Right / Left off the start	First Waypoint Find Time (Minute)	Last Waypoint Find Time (Minute)	Win or Lose
1	Left	110	113	Win
2	Left	21	27	Win
3	Right	7	10	Win
4	Right	7	10	Win
5	Left	141	144	Win
6	Left	116	118	Win
7a	Left	1	8	Win
7b	Right	17	20	Win
8	Left	5	14	Win
9	Right	1	10	Win
10	Left	10	14	Win

Table 1: Results table showing the run number, starting direction, time it took to find the waypoint for the first time in each (minutes), time to find the waypoint in the fifth back-to-back trial, and whether the trial was considered a win or a loss. The first greyed out run number 7a was not recorded, but as it was observationally the most interesting run, I chose to report on it anyway.

Surprisingly, given enough time, Boats successfully reached the waypoint 11 out of 10 times—a result that includes an unrecorded but particularly noteworthy run (#7a displayed as the grey line item in Table 1). In that run, Boats sprinted up the leg of the T-maze, ran into the left arm, bounced out, and entered the right arm, finding the waypoint within the first two minutes. It then proceeded to locate the waypoint again in the second episode. However, in the third episode, Boats tried the left arm again but got stuck in there for the full three minutes. After that mistake, it immediately redeemed itself by quickly finding the waypoint in the remaining five episodes, achieving its best time in this set.

Overall, I would rate Boats performance as **7.5**/**10**. While finding a single waypoint in a T-maze of this size is an extremely simple task—one that a human child could likely complete in under 20 seconds—Boats' longest attempt took 141 minutes. This result skirts the edge of what I would consider success. However, I am very satisfied with Boats' ability to overcome the critical issue of getting stuck in the left arm of the maze. A second notably success is that, in 7 out of 10 trials, Boats found the first waypoint in less than 22 minutes, suggesting that the system has merit. These achievements would likely not have been possible without the radar system, as Boats would likely have failed 6 out of the 10 attempts without it. This improvement, in my view, earns both Boats and myself an above-average rating.

It is also worth noting that my results section fell short of my original proposal's plan to capture and graph the spacetime trajectories produced by the agent. I mention this for two reasons. First, I believe these visualized lines are the cornerstone of my overarching idea, the best way of conveying the way I visualize human experience of time. While I could not create these visualizations for this project, they remain a critical direction for future work. Second, the omission of these lines underscores the challenges I faced throughout this project. Their absence is a reminder of the steep learning curve I encountered, but it also highlights the progress I made. Looking back, I am proud of the results I was able to achieve despite this shortcoming.

Future Work

There are two avenues of project-specific future work I would like to discuss. First, regarding reinforcement learning itself, there are numerous opportunities to improve both my understanding of machine learning and the project in its current form. Second, I could work toward restoring the original complexity of the Mazerunner game, potentially enabling a direct comparison between the behavioral patterns of human participants and RL agents.

To the first point, a key area for future exploration would be dissecting the Proximal Policy Optimization (PPO) algorithm in greater depth. Understanding how each variable in the algorithm impacts learning, stability, and performance would be critical for creating an agent capable of completing the task in a more optimized and efficient manner. Another potential direction involves investigating how to store the trained model's state and redeploy it in the same or slightly modified environments to assess its ability for continuous refinement and adaptability. Additionally, as suggested by instructor Embry, experimenting with alternative reinforcement learning models, such as Deep Q-Learning (DQN) or Soft Actor-Critic (SAC), could reveal whether these algorithms are better suited for navigation tasks of this type.

To the second point, I could reintroduce elements of complexity that were stripped away during the refinement process. This might involve gradually restoring features such as multiple waypoints, destructible objects, and bomb placement mechanics to create a more challenging environment. Ultimately, the goal would be to develop an agent capable of completing the original Mazerunner game. Such an agent could then be compared to human participants in terms of efficiency, strategy, and behavioral patterns, potentially providing insights into differences between artificial and human problem-solving approaches.

Stakeholder Acknowledgements

At its heart, this project represents a broad philosophical idea of exploring the present moment of human experience—one that intersects fields like psychology, data analysis, and ethics—channeled through the framework of a ML project. While this idea could be explored from the perspective of many different stakeholders, I will limit this discussion to three areas that are most important to me: the methodological implications for human psychology research, the potential applications for esports analytics, and the ethical considerations surrounding irresponsible use cases.

Implications for Human Psychology Research

This project highlights how RL paradigms, when combined with VE, could serve as powerful tools for studying human psychology. By simulating decision-making processes and capturing patterns of behavior, these experimental setups can reveal how agents navigate uncertainty and optimize their actions over time. Translating this to human participants could provide new insights into cognitive processes such as learning, problem-solving, and adaptation. For example, comparing the trajectories of RL agents with those of humans in similar virtual tasks could help researchers identify heuristics or strategies that align with different cognitive profiles. This paradigm offers a unique blend of ecological validity and methodological control, making it an attractive addition to the experimental toolkit for cognitive and behavioral psychology.

Applications for Esports Analytics

In the realm of esports, understanding player behavior is critical for both competitive performance and audience engagement. The methodological insights gained from this project could inform novel approaches to esports analytics. Specifically, the idea of analyzing behavioral trajectories as patterns in spacetime could provide players and coaches with valuable feedback on decision-making and optimization. For instance, a data visualization tool could compare the motility patterns of professional players with those of less experienced players, revealing areas for improvement or moments of strategic divergence. This type of analysis could also be extended to develop artificial intelligence (AI) training partners that emulate or challenge human strategies, further bridging the gap between artificial and human intelligence in competitive gaming.

Irresponsible Use Cases

While the potential applications of this paradigm are exciting, they also raise important ethical questions. The ability to analyze behavioral trajectories with such granularity could be misused in ways that harm individuals or infringe on their privacy. For example, this technology might be exploited to create manipulative training systems, invasive surveillance tools, or AI algorithms designed to predict and influence human behavior for unethical purposes. As researchers and developers, I believe it is imperative to clearly define the ethical boundaries of these technologies, ensuring their implementation serves the greater good rather than enabling harm. As Kant emphasized in his Categorical Imperative, we must act according to universal moral principles and never treat people merely as a means to an end (Rueter, 2023).

Conclusion

At the beginning of this project, I set out to use RL to explore a concept I call *thick time*—the spread of spatial arrangements that can potentially exist at any given moment. To pursue this idea, I

equipped Boats with sensory capacities, tuned the environment it would navigate, and refined the brain it would use. By the end, Boats was able to consistently find the waypoint within half a million steps, demonstrating ten separate trajectories through the T-maze that all led to the same metric of success. In my view, because these ten routes all spring from the same controlled virtual spacetime origin they embody the existence of thick time.

While I do not claim to know whether the paths we take are deterministic or indeterminate, or whether every possibility actualizes or not, I feel confident in saying this: as time progresses, the space before each of us is open, ready to be filled with whatever we choose to do next. This project provided me with the opportunity to refine both my technical skills and philosophical intuition, offering me a chance to try to produce new insights into the intersection of machine learning, human experience, and the nature of reality.

Citations

A, H. (2023, September 24). *What is the Difference Between State Function and Path Function*. Retrieved from Pediaa: https://pediaa.com/what-is-the-difference-between-state-function-and-path-function/

Brantley, S., Wilkinson, M., & Feng, J. (2021). Beat the Bots: Exploring the Effects of Placebo Manipulation on Performance During Video Gameplay. *Proceedings of Human Factors and Ergonomics Society Annual Meeting*.

Cecchini, D., Brantley, S., & Dubljević, V. (2023). Moral judgment in realistic traffic scenarios: moving beyond the trolley paradigm for ethics of autonomous vehicles. *AI & SOCIETY*.

Kumar, P. (2023, October 30). *Kant's Idealistic Theory of Space and Time*. Retrieved from Philosophy Institute: https://philosophy.institute/philosophy-of-science-and-cosmology/kants-idealistic-theory-space-time/

Rueter, S. (2023, August 5). *A Comprehensive Overview Of Kant's Categorical Imperative*. Retrieved from Philosophos: https://www.philosophos.org/metaphysical-theories-kant-s-categorical-imperative?

Thompson, J. J., Blair, M. R., Chen, L., & Henrey, A. J. (2013). Video Game Telemetry as a Critical Tool in the Study of Complex Skill Learning. *PLOS ONE*.

Wilkinson, M., Brantley, S., & Feng, J. (2021). A Mini Review of Presence and Immersion in Virtual Reality. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*.

Acknowledgements

I would like to express my gratitude to the following individuals, organizations, and tools for their invaluable contributions to the success of this project:

- **Brittani**, for listening to all my challenges, and specifically for engaging with the issue of the agent getting stuck in the left arm and recommending the radar system that ultimately led to achieving 10/10 wins.
- **Stout**, for tinkering with the project while I was at work, figuring out how to correctly use the command line functions, and implementing the steps-without-progress functionality—an essential predecessor to the three-minute episodes that kickstarted Boats' learning process.
- **Ziencik**, for being a squared away sailor and giving me the name "Boats."
- **Austin**, for suggesting the combination of reinforcement learning with my line visualization idea, effectively setting the stage for this project.
- The **members** of the Applied Cognitive Psychology Laboratory and NeuroComputational Ethics Research Group, for teaching me how to perform research, communicate my ideas, and temper my claims.
- My **mom**, **dad**, and **all my friends**, for encouraging my ramblings about the nature of reality and offering feedback on what parts do not make sense.
- **NCSU Data Science Academy**, for providing such an open-ended learning opportunity and **Instructor Embry**, for the supportive comments, accurate guidance, and logistical flexibility.
- **ChatGPT**, for serving as an editor, sounding board, and debugging assistant throughout the project.

Thank you for your time.