ML Project Proposal – Time, Probability, and Motility

Sean Brantley

Question:

What is now? Is it the continuation of the past into a deterministic future or is the world being built anew every moment? I propose using reinforcement learning, a contemporary computer science strategy, to explore whether the present is merely the continuation of the past or if reality is being built as we go. My approach frames the problem as teaching a virtual avatar to complete a video game task as quickly as possible. By framing this problem in the context of a video game, where the avatar's actions unfold over time, I can create a controlled environment to simulate and investigate this philosophical question.

Background:

Since single moments can never be replicated, from an experimental perspective the exploration of this concept is physically impossible. However, virtual task environments (VTE - video games and virtual reality-based experiments) may provide the best proxy for exploring 'the now' as the state of any virtual task can be saved and replayed. Since a time slice can be stored and redeployed (replicability) it mimics a starting position of the world happening multiple times (a seeming impossibility). This provenance, while being only virtual, could provide insight into how reality functions at each moment. Whereas moments are irreversibly lost to the past, VTEs allow for an almost paradoxical exploration where identical moments can be replayed by the same participant and across different participants. Thus, granting insight into the continuous nature of time through the lens of experiencers, all starting from a common origin. In the following sections, I will layout an argument for this project and a plan to complete it.

My Game, Research, and Local environment:

To learn Unity real-time software development and create an experimental platform for personal use, I bought a game design course from Udemy and have developed a fully functioning Bomberman-like game using their template. The game has two levels and the player's goal is to eliminate all the mutant animals running around the map using a supply of bombs (Figure 1). Throughout the map, there are non-destructible and destructible blocks. Underneath the destructible blocks, three different power-ups (increases to avatar movement speed, bomb supply, and projectile distance) have a random chance to spawn. The player wins when all enemies are eliminated.

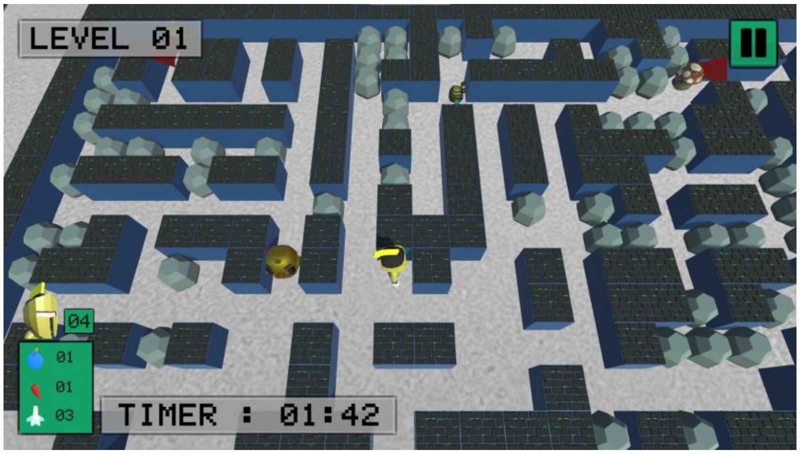

Figure 1 - Screenshot of my working MazeRunner game. Design instructions and development support – Learning Unity and C# for Complete Beginners by MetalStorm Games on Udemy.com

In addition to the course design, I've implemented two extra components; a timer and a world line tracker (grabs the avatar's position in 2D space and plots it over a 3rd temporal dimension). I added these pieces because of my profound interest in using ethological methodologies to explore human behaviors (Figure 2). As can be seen at least for animals, geometric motility patterns can be useful visualizations of decision-making and environmental interactions, and if used correctly might provide insights into time itself. While I plan to perform my future work with human participants, I think that learning how to create an AI that can emulate successful behaviors, would be a great comparison data set as at the end of the day the decision trees for both humans and AI can be represented with spacetime lines.

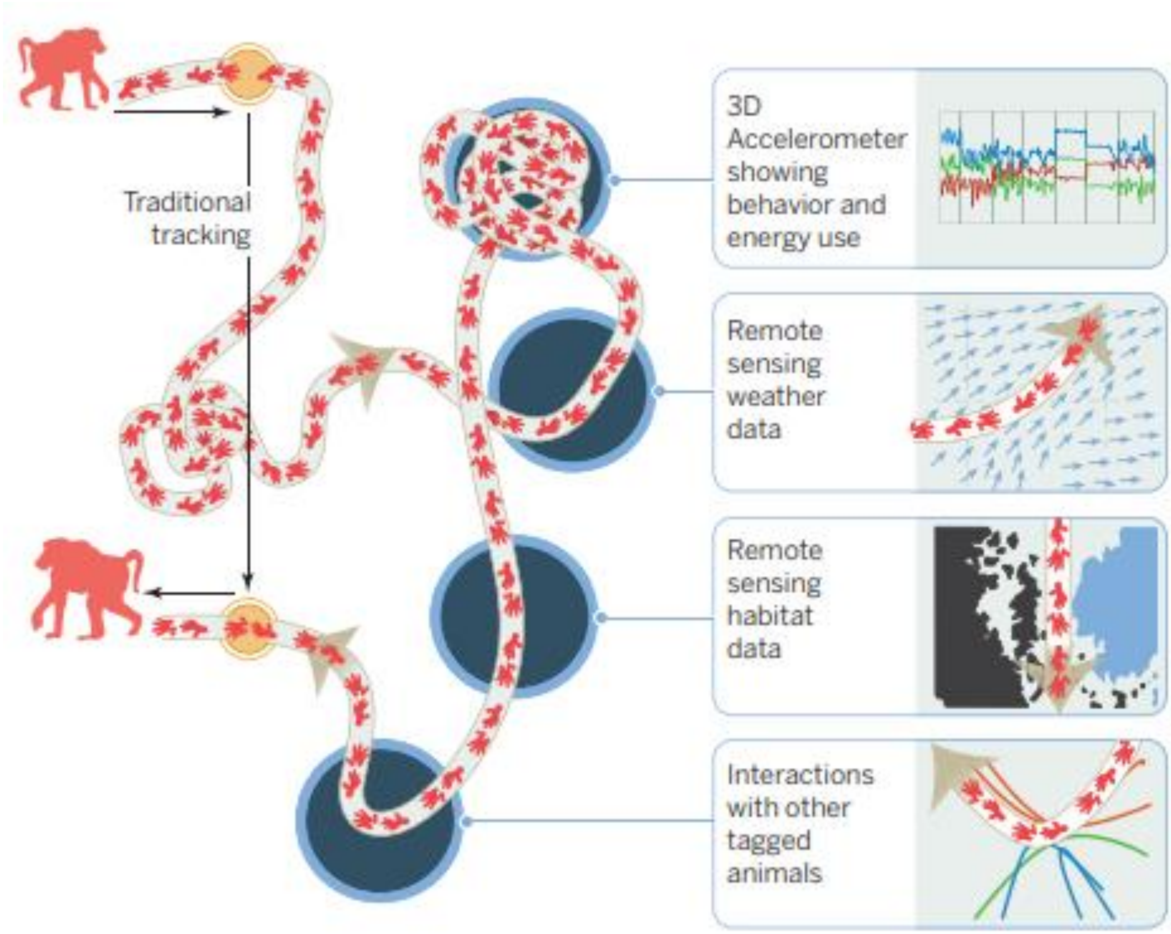
Traditional tracking
3D Accelerometer showing behavior and energy use
Remote sensing weather data
Remote sensing habitat data
Interactions with other tagged animals

Figure 2 – Traditional Animal Tracking (Yellow Dots) vs. High-Resolution BigData Animal Tracking (Grey and Red Line). Reproduced from "Terrestrial animal tracking as an eye on life and planet," by R. Kay, 2015, Science, 348(6240). DOI: 10.1126/science.aaa2478.

Finally, I would like to do this project completely from my local machine. Since I enjoy gaming and wanted to learn Unity, I invested in a setup that has a high-performance graphics card a 6-core, 6-thread processor with 16 GB of RAM, and currently have 243GB of disk space on an external hard drive. Overall, this system should be well-equipped to handle the reinforcement learning tasks locally.

Twofold Project Aim:

1. I propose setting up a reinforcement algorithm to train my in-game avatar, 'boats,' allowing it to learn how to complete and, hopefully, speedrun a single level. If successful, other avatars will be trained to generate both within-participant and between-participant data. This combination of data will permit an investigation into a proxy for the same time slice, examining how an avatar's motility over time represents the vast spread of potential outcomes from some initial moment.

2. As a presentable, I intend to collect the lines of every trial (or every nth trial) and combine them in a meaningful graphic to showcase the various world lines (Figure 3) available from a single starting spacetime position. Given the scope of this class, I only plan on evaluating performance in terms of speed and not strategies. However, alternate strategies are something I am interested in categorizing in the future. In terms of what I am looking for, I hope to see the lines get shorter as trials progress and that the avatars create different successful trajectories. While it is the randomness in the underlying algorithm producing the variation in motility, I believe it provides strong evidence that the space exists for near infinite amount of successful and unsuccessful trajectories, each representing a potential that could emerge from the isolated moment, reinforcing the idea that every moment contains a vast range of possible futures (thick time).

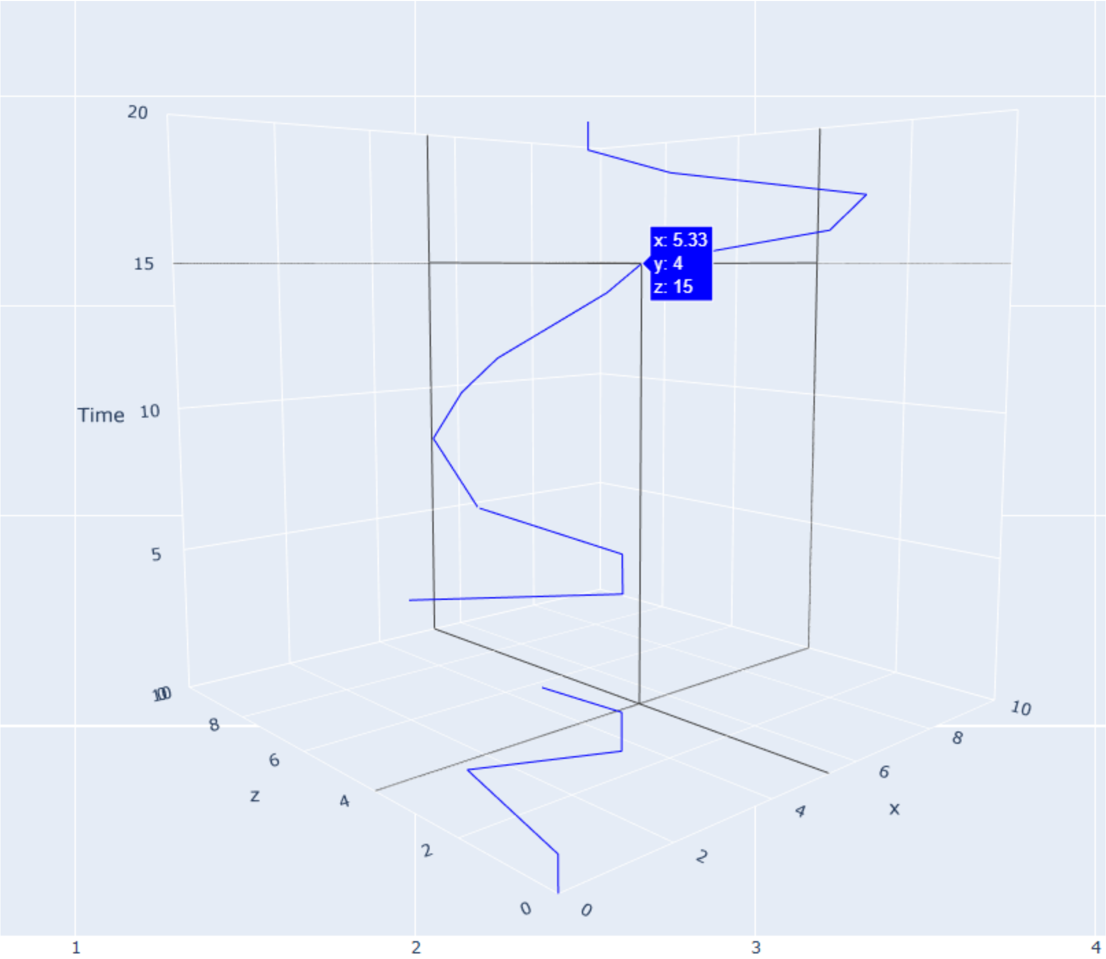

Figure 3 – Virtual world line. This graphic shows the position of the avatar in virtual space over a period of twenty seconds.

Design Proposal:
For this I provided ChatGPT with a draft of the previous sections of the proposal and the prompt "Alright, since the piece that I am the worst at is the ML aspect. Please walk me through an approach to this problem that is simple and reliable for creating an avatar that will learn to play my game."

It recommends the following steps:

Step 1 - Research and Set Up ML-Agents in Unity
Unity already provides a ML-Agent toolkit that is very likely compatible with my game. It has provided a link to the installation guide and given me a starting script for my agent class. However, to ensure that I understand the code I am implementing and the underlying machine-learning algorithm I would create a short write-up on the library and used functions.

Step 2 - Design a reward function.
For this, I intend to strip the game down to a single level and remove all powerups to simplify the game and remove randomness. The following is the proposed reward function parameters.

- Running into destructible or non-destructible blocks -1
- Dying to bomb -1
- Standing still -2
- Dying to an enemy -5
- Losing the game – 10

- Taking a step +1
- Placing a bomb +2
- Destroying a destructible block +2
- Blowing up an enemy +5
- Winning the game +10

- Winning the game quicker than it did previously +100

These positive and negative rewards are designed to keep the avatar moving through the maze and placing bombs. Penalizing the avatar for touching walls should prevent it from getting stuck, while rewarding movement encourages exploration. The remaining rewards similarly guide the algorithm’s learning, with the +100 reward for speed running designed to prioritize finding the optimal path through the maze. By assigning such a high value, I aim to encourage aggressive exploration and optimization of the avatar's movements.

Step 3 - Train the Agent
In this section, it first shows me how to begin the training using the command mlagents-learn in the CLI. Second, it outlines how to set up training configurations using a YAML file provided by the toolkit. Finally, it shows me how to monitor the training using TensorBoard integration.

Step 4 - Debug and Tune
The three suggestions it says are to adjust hyperparameters, improve observations, and tweak the reward function. There will likely be significant debugging but given that it is my code base I feel moderately confident I can integrate this new component.

Step 5 - Collect and Visualize data

Here I have thought of two ways of representing the data but haven't thought of a great way yet. The simple way would be to use the line tracker I already have in place and merely combine all the graphs at the end maybe color-coding the trials and add the minimap as the background. The second which might be more difficult would as a density aggregation of positions like what has been done in the geospatial analytics lab here at NC State (Figure 4).

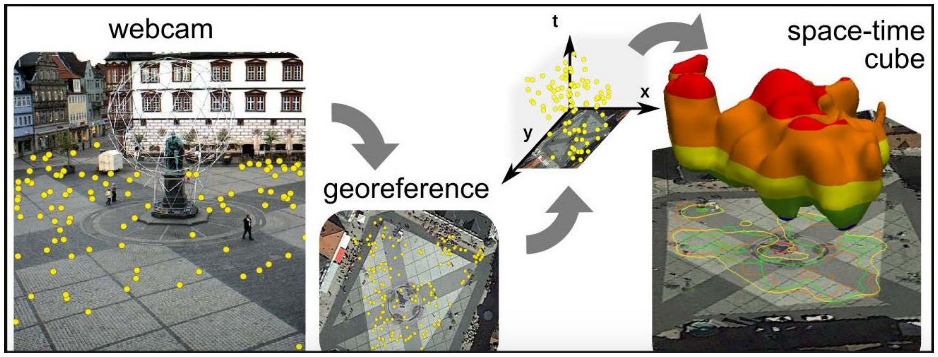

Figure 4 Reproduced from Petrasova, A., Hipp, J. A., & Mitasova, H. (2019). "Visualization of Pedestrian Density Dynamics Using Data Extracted from Public Webcams." Geo-Inf. doi.org/10.3390/ijgi8120559

Answering Unaddressed Milestone 2 Questions:

Potential Stakeholders?

Gamers/gaming industry, military, researchers, advertisers, app developers, and philosophers. However, I am pursuing this project out of a personal interest in understanding reality. While these fields could find applications for the data collection methods used, I would personally prefer that this approach not be employed by the military or advertisers.

Obstacles?

- I feel like the code integration could be difficult, especially some of the I/O components like collecting and processing the data.
- The logistical framework to support the project could be more difficult than I understand. I might need to swap to cloud computing storage, buy a new hard drive, or free up memory to make space for data collection.
- The algorithm might be bad, and I might not ever figure out how to get boats to beat the game.

Long-term aim:

I want to push the idea with my life that time is thick, meaning there is a spread of reality available at every moment of human experience. For me, this project represents an opportunity to build on the foundation of my life's work. By running this experiment with people, I hope to demonstrate that neurophysiological patterns correlate with these geometric patterns. This connection may help explain the neural correlates of behavior. Tangibly this stance **might** provide a physical mechanism for free will, bridge the subjective-objective gap, aid in understanding the measurement problem of Quantum Mechanics, and provide a deeper metaphysical explanation for time and inextricably linked probability.